

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет геосистем и технологий»
(СГУГиТ)

ИНТЕРЭКСПО ГЕО-СИБИРЬ

XXII Международный научный конгресс

Сборник материалов в 9 т.

Т. 9

Международная научная конференция

**«ПЕРСПЕКТИВНЫЕ РЕШЕНИЯ В СФЕРЕ
ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ
ТЕХНОЛОГИЙ И КИБЕРБЕЗОПАСНОСТИ»**

№ 1

Новосибирск
СГУГиТ
2026

Ответственные за выпуск:

Кандидат технических наук, директор Научно-исследовательского института
информационно-коммуникационных технологий, ведущий научный сотрудник
Национального исследовательского ядерного университета «МИФИ», г. Москва
Е. А. Басыня

Кандидат технических наук, доцент, заведующая кафедрой прикладной информатики
и информационных систем, СГУГиТ, г. Новосибирск
Т. Ю. Бугакова

Кандидат технических наук, доцент кафедры прикладной информатики и информационных
систем, СГУГиТ, г. Новосибирск
С. Ю. Кацко

С26 Интерэкспо ГЕО-Сибирь. XXII Международный научный конгресс,
18–22 мая 2026 г., Новосибирск : сборник материалов в 9 т. Т. 9: Междуна-
родная научная конференция «Перспективные решения в сфере информаци-
онно-коммуникационных технологий и кибербезопасности». – Новоси-
бирск : СГУГиТ, 2026. № 1. – 290 с. – ISSN 2618-981X. – Текст : непосред-
ственный.

DOI 10.33764/2618-981X-2026-9-1

В сборнике опубликованы материалы XXII Международного научного конгресса
«Интерэкспо ГЕО-Сибирь», представленные на Международной научной конференции
«Перспективные решения в сфере информационно-коммуникационных технологий и ки-
бербезопасности».

Печатается по решению редакционно-издательского совета СГУГиТ

Материалы публикуются в авторской редакции

Издание может содержать сведения об иностранных агентах

В. Д. Аверин¹✉

Метод мониторинга работы локальных пользователей электронных вычислительных машин в условиях использования средств анонимизации

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: vd_averin@mail.ru

Аннотация. В работе рассматривается проблема комплексного мониторинга и контроля действий пользователя в условиях использования им средства анонимизации. Актуальность работы обусловлена ростом целевых атак, в процессе которых используются данные средства. Описываются распространенные вектора атак с точки зрения внутреннего и внешнего нарушителя, которые включают в себя использование данных средств для закрепления, предотвращения обнаружения, получения учетных данных и эксфильтрации. Описывается техническая сложность контроля и непрерывного мониторинга данного типа средств за счет их доступности и распространенности для конечного пользователя. По результатам анализа, предлагается метод, который позволяет по анализу локальных артефактов ЭВМ пользователя определять и реагировать на появление или активацию работы данных средств. Артефакты формализуются и систематизируются на прямые и косвенные признаки, относящиеся к тем или иным типам средств. Предлагается конкретный перечень признаков для операционных систем семейства Windows, а также возможная техническая реализация метода в виде отдельной системы мониторинга, которая может быть интегрирована в работу с другими средствами защиты информации.

Ключевые слова: средства анонимизации, мониторинг пользователей, контроль конечных точек, теневая ИТ

V. D. Averin¹✉

A method for monitoring the activity of local users of electronic computing machines under conditions involving the use of anonymization tools

¹National Research Nuclear University MEPhI (Moscow Engineering Physics Institute),
Moscow, Russian Federation
e-mail: vd_averin@mail.ru

Abstract. The article addresses the problem of comprehensive monitoring and control of user actions under conditions where the user employs anonymization tools. The relevance of the study stems from the increase in targeted attacks that utilize such tools. Common attack vectors from the perspective of both internal and external attackers are described, including the use of these tools for achieving persistence, evading detection, obtaining credentials, and exfiltrating data. The technical complexity of controlling and continuously monitoring this class of tools, arising from their availability and widespread use among end users, is discussed. Based on the analysis, a method is proposed that enables the detection of and response to the appearance or activation of such tools by analyzing local computer artifacts of the user's machine. The artifacts are formalized and categorized into direct and indirect indicators associated with specific types of tools. A concrete set of indicators for Windows

family operating systems is provided, along with a possible technical implementation of the method as an independent monitoring system that can be integrated with other information security systems.

Keywords: anonymization tools, user monitoring, endpoint protection platform, shadow IT

Введение

В условиях стремительного развития и распространения средств анонимизации трафика набирают популярность вектора целевых атак, включающие в свою реализацию использование данных средств, в том числе на этапе закрепления, предотвращения обнаружения, получения учетных данных и эксфильтрации [1]. Злоумышленники целенаправленно комбинируют оверлейные технологии и технологии виртуальных частных сетей для сокрытия своей активности при проведении атаки [2, 3]. Кроме того, перечисленные средства используются не только внешними, но и внутренними нарушителями, например, при инсайдерской деятельности. Согласно [4], доля утечек данных по вине персонала продолжает расти, причём подавляющее большинство инцидентов носят умышленный характер.

К средствам анонимизации в данной работе относят [5]: VPN-клиенты, прокси-сервера, браузерные расширения и оверлейные сети, в частности, сеть Tor. Примерами векторов атак с использованием данных средств могут быть вредоносные VPN-расширения браузеров, осуществляющих перехват конфиденциальных данных пользователей [6], использование модификаций Tor-браузера с незадекларированными возможностями (установления reverse-shell'a и ssh-тоннеля) [7] или использование прокси внутренними нарушителями для минимального воздействия на систему и эффективного обхода сетевой фильтрации [8].

Обуславливается такое внимание к данным средствам с точки зрения атакующих сложностью факта их обнаружения на основе анализа сетевых потоков. Существующие подходы либо ориентированы на постфактум-расследование инцидентов [9], либо требуют специализированной настройки инфраструктуры, например, развертывания собственного прокси-сервера внутри инфраструктуры для глубокого анализа [10]. В следствие чего, такие признаки (переносные версии браузеров и прокси, браузерные расширения и плагины) часто становятся частью «теневого ИТ» – информационных технологий, цифровых устройств, программного обеспечения и ИТ-услуг без одобрения со стороны ИТ- и ИБ-подразделений организации.

Решением данной проблемы может служить строгая политика информационной безопасности, которая запрещает запуск и установку любых исполняемых файлов, а также покупка и переход на корпоративные браузеры, поддерживающие работу с групповыми доменными политиками, например, Яндекс Браузер для Организаций. Однако не всегда такой подход технически реализуем, особенно в СМБ-сегменте в условиях непрерывности бизнес-процессов. Но сам агентный сбор информации, который лежит в основе EPP-решений, позволяет формализовать метод обнаружения прямых и косвенных признаков наличия средств анонимизации на конечных точках.

Целью данной работы является разработка и формализация метода мониторинга активности пользователя, который систематизирует и выявляет прямые и косвенные признаки использования средств анонимизации (VPN-клиентов, прокси-серверов, браузерных расширений и Tor Browser) на основе анализа локальных артефактов операционной системы. Целевой операционной системой являются ОС семейства Windows, однако предлагаемый метод расширяется и до Unix-подобных систем, для которых будет необходимо переопределение способов реализации получения признаков.

Предлагаемый метод мониторинга

В основе метода лежит процесс инициализации цифрового профиля пользователя, который работает за определенным АРМ. Профиль включает в себя параметры операционной системы, перечень которых будет представлен далее. Ниже представлена блок-схема предлагаемого метода (рис. 1).

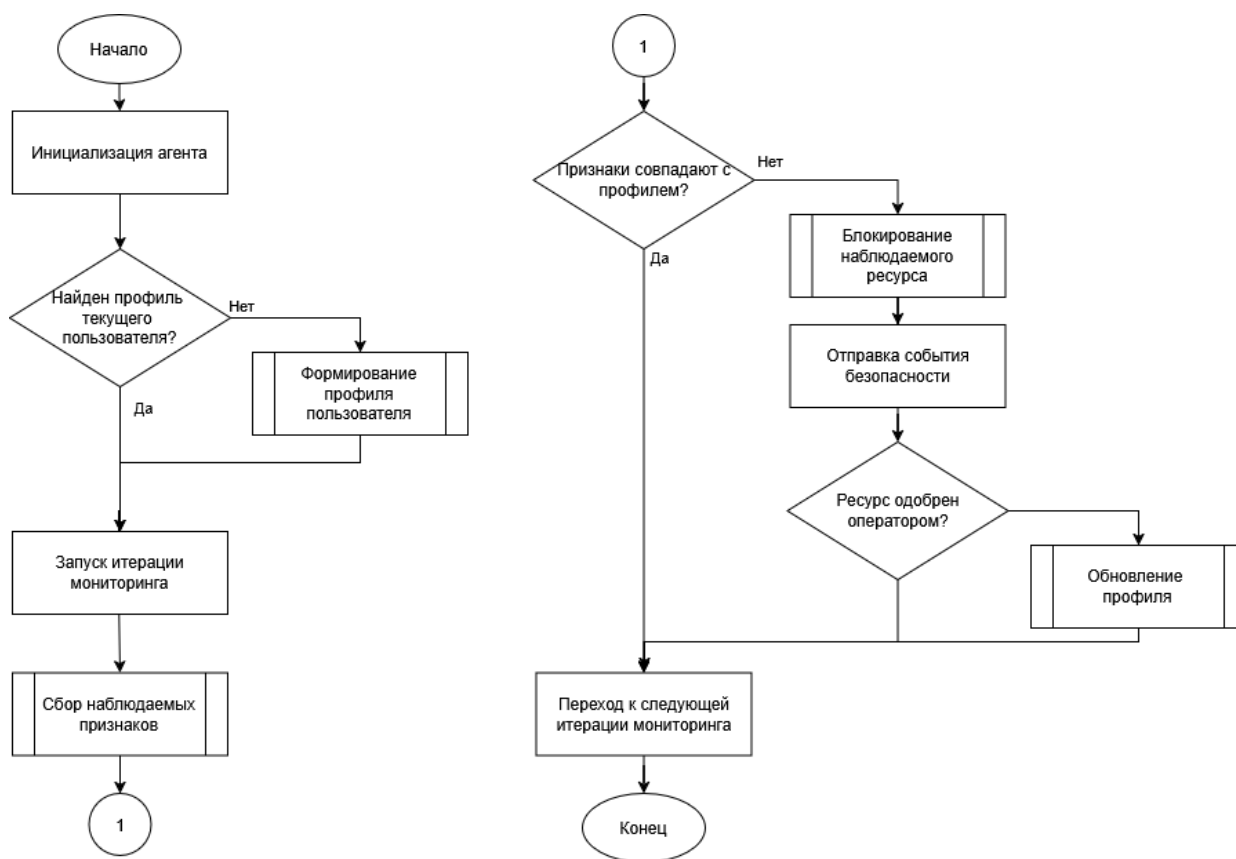


Рис. 1. Блок-схема предлагаемого метода

Как было сказано ранее, под наблюдаемыми средствами анонимизации в работе рассматривается следующий перечень: VPN-клиенты типа Remote Access, прокси-клиенты, расширения и плагины браузеров, и наличие на АРМ Tor-based браузеров. После формирования профиля включается процесс непрерывного мониторинга параметров и сравнение их изменений с текущим профилем. Так как условия работы и трудовые обязанности сотрудников меняются достаточно редко,

сформированный профиль будет достаточно четко описывать их рабочее окружение. В зависимости от характера изменений метод предусматривает внесение корректировок либо в сам профиль, либо формирование события безопасности, которое передается в службу по реагированию на инциденты.

Необходимо выделить признаки, которые предлагается рассматривать в рамках разрабатываемого метода.

VPN-клиенты удаленного доступа. В случае программной реализации средства его обязательной частью является создание и последующее включение виртуального сетевого интерфейса типа TUN или TAP, а также внесение изменения в таблицу маршрутизации [11]. Наличие данного типа изменений в совокупности указывает на создание нового соединения, но не является прямым признаком использования конкретно VPN-клиента. Например, виртуальные интерфейсы и маршруты также создаются при использовании гипервизоров (может быть необходимо отделу разработки/DevOps) или при подключении периферийных устройств, не имеющих возможности работы по протоколу IP (старые принтеры, телефония, камеры). В этом случае интерфейс будет физическим. Поэтому при начальной инициализации перечня разрешенных интерфейсов и их настроек необходим ручной анализ параметров. Кроме того, для внесения изменений в сетевые настройки системы процессу необходимы административные права. Поэтому несанкционированные изменения такого рода уже становятся прямым признаком компрометации административной учетной записи.

Прокси-клиенты. В отличие от VPN-тоннелей, виртуализирующих новый сетевой интерфейс, прокси-серверы не требуют для своей работы административных прав, так как работают на прикладном уровне. Однако прокси также используется в некоторых аналогах VPN-клиентов, которые технически просто выступают средством настройки для передачи трафика до прокси-сервера VPN-сервиса. Поэтому наличие в системе данных параметров также является наблюдаемым признаком. Примером легитимного использования прокси-клиентов может являться система Банк+Клиент СФТ, в составе которой есть подсистема InterPRO Client. Компонент для обеспечения работы TLS/SSL вместе с российскими криптографическими примитивами включает локальный прокси на стандартном адресе 127.0.0.1:2080. Поэтому данный признак также требует анализа со стороны оператора системы.

Расширения браузеров. Для поиска установленных браузеров и их расширений предлагается анализировать файлы в поиске соответствующих директорий, где хранятся установленные расширения, в том числе их метаданные и состояния (включено/выключено). Например, для Chromium-браузеров такой путь по умолчанию: «%LOCALAPPDATA%\Google\Chrome\User Data\Default\Secure Preferences». Однако нужно понимать, что браузер возможно установить в другое расположение или использовать переносные версии. Поэтому поиск и анализ должен происходить непосредственно по маске внутренних директорий. Для Chromium это «Chrome\User Data\Default\Secure Preferences». Данные расположения и парсинг расширений реализуется на программном уровне. Также нужно учитывать, что у одного пользователя может быть одновременно несколько экземпляров браузеров и профилей в них.

Браузер Tor. Большинство работ по анализу активности Tor полагаются на сетевые методы обнаружения [12-14]. Более того, браузер целенаправленно разрабатывался для сокрытия своей деятельности в системе, поэтому даже запускаемый браузером процесс имеет название «firefox.exe», так как является технически его форком. Однако если обратиться к опыту форензики, можно заметить несколько основных компонент браузера, которые участвуют в его работе постоянно и внедрены в его код статически: это файл torrc, который управляет поведением Tor и хранит полный путь до рабочей директории браузера, а также реализации мостов (транспорты), которые используются для обхода ограничений работы Tor на сетевом уровне [15]. Эти транспорты хранятся в директории PluggableTransports и должны иметь строго определенные имена: snowflake.exe, conjure-client.exe и lyrebird.exe. Наличие хотя бы одного из данных артефактов указывает на наличие данного браузера на хосте.

Таблица 1. Таблица соответствия типов средств с их признаками, техники их обнаружения и блокирования

Тип средства	Признаки	Техника обнаружения
VPN-клиент	Появление TUN/TAP интерфейсов; изменение таблицы маршрутизации	Регистр: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\ Поиск событий журнала Windows Network-Profile/Operational: ID 4001, 4002, 4004, 10001
Прокси-клиент	Появление системных прокси	Powershell: Netsh winhttp show proxy
Расширения браузеров	Появление новых браузеров и переносных версий; появление новых расширений	Поиск директорий браузеров и их парсинг; Chrome: ~.\Chrome\User Data\[Имя_профиля]\Extensions\ Яндекс Браузер: ~.\Yandex\YandexBrowser\User Data\[Имя_профиля]\Extensions\
Использование Tor	Появление файлов конфигурации; появление	Поиск файла torrc и его парсинг. Наличие любого процесса из PluggableTransports: conjure-client.exe, lyrebird.exe, snowflake.exe и др.

В результате процесс инициализации профиля пользователя состоит в анализе инженером информационной безопасности полученных данных и фильтрации разрешенных параметров хоста – сетевого интерфейса, локальных прокси, а также списка разрешенных браузеров и их расширений. При дальнейшем нарушении установленного профиля ресурс, вызвавший изменение профиля, блокируется, а событие безопасности отправляется на сервер для последующего одобрения либо подтверждения блокирования.

Блокирование ресурсов предлагается реализовывать системными утилитами, в зависимости от типа обнаруженного признака. Для сетевых интерфейсов и прокси-соединений это отключение не одобренного интерфейса и настроек прокси, для браузеров удаление/отключение не одобренного расширения и перезапуск браузера, для Tor – заморозка процессов, которые обращаются к данному файлу (реализуется, например, подключением журналирования Sysmon).

Типы средств, относящиеся к ним наблюдаемые признаки и технические варианты их обнаружения применительно к Windows представлены в табл. 1.

Результаты

Экспериментальное исследование предложенного метода реализовано в виде отдельного программного модуля с клиент-серверной архитектурой на стеке C# (агенты) + Kotlin/JavaScript/MariaDB (сервер). На сервере происходит создание исполняемых файлов агентов для распределения по клиентским машинам со своими уникальными идентификаторами. Агенты устанавливаются на хосты как системные службы (от имени SYSTEM, что можно централизованно реализовать, например, через сервер Kaspersky Security Center). Агенты присылают все собранные метрики на сервер только на этапе инициализации для создания профиля для данного агента; дальнейший процесс сравнения с профилем реализуется через логику агента, который отправляет события безопасности серверу только при его изменении либо несоответствии.

Итоговые нагрузочное и интеграционное тестирования разработанного решения проводились в рамках реальной инфраструктуры, состоящей из 46 конечных устройств с установленными агентами. На этапе формирования цифровых профилей было установлено 35 несогласованных с ИТ- и ИБ-отделами ресурсов, среди которых были: 21 расширение браузеров, 6 признаков наличия переносных версий несогласованных браузеров (в основном Opera Browser за счет встроенного в него VPN), 4 прокси-клиента, 3 признака наличия Tor-браузера (2 из которых подтвердили действительную портативную версию у сотрудника на корпоративном съемном носителе) и 1 VPN-клиент. Одобренные ресурсы насчитывали 21 уникальную запись, среди которых самыми распространенными являлись КристоПро ЭЦП Browser plug-in (расширение), 1С Предприятие 8 (прокси-клиент), ГПБ СФТ (прокси-клиент) и Диаг.Плагин.

Обсуждение

Разработанный метод позволяет реализовать непрерывный мониторинг сегмента теневой ИТ, которая чаще всего используется внутренними и внешними нарушителями, при этом не прибегая к ужесточению политики информационной безопасности, которая для малого и среднего бизнеса может быть неприменима. В отличие от существующих DLP-систем, где наблюдаемые метрики могли выступать в лучшем случае в роли признаков в признаковом пространстве, а значит не несли в себе конкретного указания на активность пользователя, данный метод приравнивает изменения этих метрик к событиям безопасности. Это позволяет провести интеграцию с SIEM-решениями системы, разработанной на основе предлагаемого метода.

Для этого необходимо реализовать коннекторы и правила корреляции, которые смогут обогащать контекст с учетом производимых пользователем действий.

Кроме того, полученный перечень одобренных ресурсов в рамках тестирования позволяет реализовать автоматизированное создание новых профилей пользователей, что значительно упрощает дальнейшую работу и поддержку системы.

Заключение

По результатам работы был предложен метод мониторинга работы локальных пользователей электронных вычислительных машин в условиях использования средств анонимизации, а также перечень наблюдаемых признаков для каждого из типов средств применительно к операционным системам семейства Windows.

Предложенный метод был реализован программно и протестирован в условиях непрерывности бизнес-процессов на существующей инфраструктуре, что позволило получить численные метрики системы, а также сформировать перечень одобренных наблюдаемых ресурсов. Дальнейшая работа может заключаться в модификации метода для автоматизированного реагирования на события безопасности и интеграции разрабатываемой системы с существующими решениями по информационной безопасности класса DLP, SIEM и EPP.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Усманов А. Р. Методы обнаружения фишинга на основе анализа поведения пользователя // Вестник науки. – 2025. – №6 (87). – С. 710–721.
2. Басыня Е. А., Хиценко В. Е., Рудковский А. А. Метод идентификации киберпреступников, использующих инструменты сетевого анализа информационных систем с применением технологий анонимизации // Доклады ТУСУР. – 2019. – Т. 22. – №2. – С. 45–51.
3. Гайдин А. И., Головчанский А. В. Средства анонимизации в механизме преступной деятельности, осуществляемой в сети Интернет // Вестник Воронежского института МВД России. – 2022. – №2. – С. 205–213.
4. InfoWatch. Утечки данных вследствие действий внутренних нарушителей. Мир – Россия 2022–2025 гг. [Электронный ресурс]. – URL: <https://www.infowatch.ru/sites/default/files/analytics/files/utechki-dannykh-vsledstviye-deystviy-vnutrennikh-narushiteley.pdf> (Дата обращения: 22.04.2026).
5. Нефедов В. С. Кriuлин А. А., Потерпеев Г. Ю. Основы обеспечения анонимности в сети Интернет : учеб. пособие. – Москва: РТУ МИРЭА, 2022. – 81 с.
6. VPN-расширение для Chrome перехватывает запросы пользователей к ИИ [Электронный ресурс] // Xaker.ru. – URL: <https://xaker.ru/2025/12/17/ai-urban-vpn/> (Дата обращения: 22.04.2026).
7. Security Week 2241: браузер Tor с бэкдором [Электронный ресурс] // SecurityLab.ru : блог компании Kaspersky. – URL: <https://www.securitylab.ru/blog/company/kaspersky/352390.php> (дата обращения: 22.04.2026).
8. Чужое в допустимом контуре [Электронный ресурс] // itsec.ru. – URL: <https://www.itsec.ru/articles/chuzhoe-v-dopustimom-konture> (дата обращения: 22.04.2026).
9. Гончар В. В., Аветисян Б. Р. Актуальные направления деанонимизации правонарушителей в сфере ИТ // Вестник экономической безопасности. – 2024. – №1. – С. 61–66.

10. Канатьев К.Н., Гребенщиков А. А. Каравцев Н. А. Передовые методы обнаружения и обфускации VPN-трафика: углубленный анализ OpenVPN и его уязвимостей в современную цифровую эпоху // Инновации и инвестиции. – 2023. – №10. – С. 211–215.
11. Чубахиро А., Каманде М. В. Виртуальные вычислительные сети для случая аппаратной виртуализации // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и Технические Науки. – 2020. – № 6. – С. 158–163.
12. Manjesh P Shetty Dark & deep Web: advanced forensic analysis of Tor Browser and implications for law enforcement agencies // Centre For Cybercrime Investigation Training & Research. – 2022. – 67 с.
13. Amir Aboubakr Shaker Mahmoud, Animesh Kumar Agrawal, Nilay R. Mistry DarkExtract: tool for extracting and analysing Tor Browser host-based activities // International Journal of Electronic Security and Digital Forensics. – 2025. – Vol. 17. – №5. – С. 563–581.
14. Matt Muir, Petra Leimich, William J Buchanan A Forensic Audit of the Tor Browser Bundle // Digital Investigation. – 2019. – Vol. 29. – С. 118–128.
15. Вехов В. Б., Смушкин А. Б. Криминалистическое исследование цифровых отпечатков компьютерных устройств // Всероссийский криминологический журнал. – 2024. – Т. 18. – №4. – С. 390–397.

© В. Д. Аверин, 2026

К. Д. Антонников^{1✉}, Н. С. Головачев¹

Проектирование и разработка иммерсивной виртуальной среды для изучения всемирной истории «ArtHistory»

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: MuseumArtHistory@gmail.com

Аннотация. В данной статье рассматривается процесс проектирования и программной реализации интерактивной виртуальной среды «ArtHistory», созданной на базе среды для разработки Unreal Engine 5. Актуальность работы обусловлена необходимостью трансформации классических образовательных подходов и цифровизации культурного наследия с использованием иммерсивных технологий. Статья подробно описывает техническую архитектуру приложения, включая применение высокотехнологичных систем Nanite и Lumen для достижения фотореалистичного качества визуализации. Особое внимание уделено разработке логики взаимодействия с объектами на языке Blueprint, а также созданию системы портальной навигации. Практическая значимость работы заключается в возможности внедрения разработанного приложения в учебный процесс образовательных учреждений и музейных комплексов.

Ключевые слова: иммерсивная среда, виртуальный музей, Unreal Engine 5, Nanite, Lumen, образовательные технологии, виртуальная реальность, цифровизация истории

K. D. Antonnikov^{1✉}, N. S. Golovachev¹

Design and development of an immersive virtual environment for studying world history "ArtHistory"

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: MuseumArtHistory@gmail.com

Abstract. This article examines the design and software implementation of "ArtHistory," an interactive virtual environment developed using the Unreal Engine 5 game engine. The relevance of this work stems from the necessity of transforming classical educational approaches and digitizing cultural heritage through immersive technologies. The paper provides a detailed description of the application's technical architecture, including the integration of high-tech Nanite and Lumen systems to achieve photorealistic visualization quality. Special attention is paid to the development of object interaction logic using the Blueprint visual scripting language, as well as the creation of a portal navigation system. The practical significance of the work lies in the potential for implementing the developed application into the educational curricula of academic institutions and museum complexes.

Keywords: immersive environment, virtual museum, Unreal Engine 5, Nanite, Lumen, educational technologies, virtual reality, digitalization of history

Введение

Развитие современных информационных систем привело к появлению новых форматов передачи знаний, среди которых особое место занимают иммерсивные среды. Традиционные музейные платформы зачастую ограничиваются статичными турами лекциями, что снижает интерес молодой аудитории. Целью

данной работы является разработка приложения «ArtHistory» – виртуального музея, обеспечивающего наглядное и глубокое погружение в мировую историю через игровые механики.

Исследование предметной области

В последние десятилетия сохранение и популяризация культурно-исторического наследия столкнулись с вызовами глобальной цифровизации. Традиционные формы музейной экспозиции и классические образовательные методики изучения истории в ряде случаев демонстрируют снижение эффективности в работе с современным пользователем, ориентированным на быстрый и наглядный способ получения информации. Проблема низкой вовлеченности обучающихся создает необходимость поиска новых технологических подходов, способных трансформировать пассивное созерцание экспонатов в активный исследовательский процесс.

Развитие технологий открыло широкие перспективы для создания иммерсивных образовательных сред [1]. Многочисленные исследования подтверждают, что использование игровых сред для разработки (Game Engines) в педагогических целях позволяет достичь более высокого уровня когнитивного погружения. Экспериментальные данные демонстрируют, что интеграция интерактивных элементов и 3D-визуализации в процесс изучения истории способствует росту Retention Rate (показателя удержания внимания) на 40% по сравнению с традиционными текстовыми и видеоматериалами. Современные решения в области виртуальных музеев позволяют не только реконструировать утраченные объекты архитектуры и искусства с высокой точностью, но и обеспечивают эмоциональную связь пользователя с историческим контекстом через систему виртуального присутствия.

На мировом рынке представлен ряд цифровых платформ, таких как Google Arts & Culture или виртуальные туры Лувра и Эрмитажа, однако большинство из них базируется на технологии панорамной съемки 360°, которая ограничивает свободу перемещения и глубину взаимодействия с окружением. Существующие игровые аналоги зачастую смещают акцент в сторону развлечения, жертвуя исторической достоверностью и методологической ценностью. В российском сегменте наблюдается дефицит кроссплатформенных приложений, которые сочетали бы в себе фотореалистичную графику (уровня AAA-проектов), развитую игровую логику и полную локализацию на русский язык. На основании этого было принято решение о разработке иммерсивной среды «ArtHistory» на базе Unreal Engine 5, которая отвечает требованиям современного пользователя к качеству визуализации и предлагает уникальную механику восстановления музейного фонда экспонатов, стимулирующую познавательный интерес.

Методы и технологии

Разработку иммерсивной виртуальной среды «ArtHistory» было решено вести в интегрированной среде Unreal Engine 5 от компании Epic Games [2, 3]. Данная среда для разработки на сегодняшний день является передовым решением

для создания фотореалистичного контента и сложных интерактивных систем, предоставляя разработчикам доступ к открытому исходному коду и мощному инструментарию для визуализации в реальном времени [4].

К преимуществам использования Unreal Engine 5 в рамках данного проекта можно отнести аспекты, которые перечислены ниже.

Инновационные системы рендеринга, такие как Nanite и Lumen позволяют достичь беспрецедентного уровня детализации без необходимости ручного создания упрощенных уровней детализации (LOD), что существенно ускоряет процесс интеграции сложных исторических ассетов в проект [5].

Система визуального программирования Blueprints позволяет реализовывать сложную игровую логику, интерфейсы и взаимодействия с объектами без необходимости написания низкоуровневого кода на C++, что снижает порог входа и вероятность возникновения синтаксических ошибок [6].

Кроссплатформенность реализуется за счет того, что Unreal Engine 5 обладает встроенным инструментарием для быстрой адаптации и развертывания проекта на различных устройствах, включая персональные компьютеры под управлением Windows и современные шлемы виртуальной реальности.

Обширная экосистема и библиотека Quixel Megascans дает доступ к фотogramметрическим материалам и моделям высокого разрешения позволяет создавать аутентичные природные и исторические ландшафты с минимальными временными затратами [7].

Помимо вышеперечисленных систем, Unreal Engine 5 предоставляет разработчику мощный редактор уровней с интеллектуальной системой размещения объектов, инструменты профилирования производительности (GPU/CPU Visualizer), позволяющие выявлять проблемные места в рендеринге, а также систему Niagara VFX для создания сложных визуальных эффектов (огонь, пыль, порталы) [8].

Для реализации логики взаимодействия был выбран метод визуального скриптинга в сочетании с событийной архитектурой. Это позволило создать гибкую систему управления состояниями экспонатов и навигации по виртуальному музею. К преимуществам такого подхода в контексте проекта относятся:

Лаконичность и наглядность алгоритмов представленные в виде структуры графов Blueprints, что позволяет визуально отслеживать потоки данных и событий, что улучшает читаемость и упрощает отладку взаимодействия.

Безопасность типов и данных реализована за счет встроенных механизмов проверки связей предотвращают критические сбои при обращении к переменным и объектам в реальном времени.

Легкое и интуитивное взаимодействие с Unreal Motion Graphics обеспечивает бесшовную связь между логикой приложения и пользовательским интерфейсом (виджетами), что критично для систем навигации и образовательных справок.

Сочетание комплексных инструментов Unreal Engine 5 и возможностей визуального программирования создает эффективную среду для быстрой итерации и создания качественного образовательного продукта.

Таким образом, для разработки приложения «ArtHistory» был использован следующий технологический стек:

- среда для разработки Unreal Engine 5.6;
- система визуализации Nanite;
- система глобального освещения Lumen [9];
- визуальная система скриптов Blueprints;
- система спецэффектов: Niagara VFX;
- интерфейс пользователя Unreal Motion Graphics.

Результаты

В основе приложения лежит механика восстановления коллекции экспонатов музея. Изначально музей представлен пустыми залами. Пользователь должен перемещаться через порталы в исторические локации («Доисторический мир», «Древний Египет» и др.), находить артефакты и выполнять квесты. При нахождении объекта переменная типа Boolean меняет значение с 0 на 1, и экспонат «проявляется» в главном холле музея (рис. 1). Это создает многослойный сюжет, где сложность поиска варьируется от простых находок до алгоритмов из 5 и более шагов.

В главном зале реализован навигационный центр. При взаимодействии с ним блокируется управление игрока и открывается виджет выбора комнат. Перемещение между уровнями осуществляется через порталы. Технически портал представляет собой Vox Collision, при пересечении которого активируется загрузочный экран. Для повышения проработанности проекта загрузочные экраны выбираются системой случайно из заранее подготовленного массива.

В каждой комнате создана уникальная атмосфера. Например, в локации «Доисторический мир» использован объемный туман и световые лучи. В зале живописи применены мастер-материалы с картами высот, передающими фактуру мазка краски на холсте.



Рис. 1. Пример «возвращенных» в музей артефактов древности

Заключение

Разработанное приложение «ArtHistory» демонстрирует эффективность объединения иммерсивных технологий и образовательных задач. Использование UE5 позволило достичь высокого уровня оптимизации при экстремальной детализации объектов. Проект обладает высоким коммерческим потенциалом и готов к масштабированию на рынке.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Маккеффри, М. Unreal Engine VR для разработчиков / М. Маккеффри; перевод с английского Р. В. Терханова. – Москва: ДМК Пресс, 2019. – 266 с. – ISBN 978-5-97060-642-1. – URL: <https://dmkpress.com/catalog/computer/games/978-5-97060-642-1/>.
2. Unreal Engine 5 Documentation. – Текст: электронный // Unreal Engine: [сайт]. – URL: <https://docs.unrealengine.com/5.0/en-US/>.
3. Маккеффри, М. Unreal Engine VR для разработчиков / М. Маккеффри; перевод с английского Р. В. Терханова. – Москва: ДМК Пресс, 2019. – 266 с. – ISBN 978-5-97060-642-1. – URL: <https://dmkpress.com/catalog/computer/games/978-5-97060-642-1/>.
4. Уточкин, В. Н. Хочу в геймдев! Основы игровой разработки для начинающих / В. Н. Уточкин, К. С. Сахнов, Д. И. Табаков. – 2-е изд. – Москва: Бомбора, 2023. – 256 с.
5. Nanite Virtualized Geometry. – Текст: электронный // Unreal Engine Documentation: [сайт]. – URL: <https://docs.unrealengine.com/5.0/en-US/nanite-virtualized-geometry-in-unreal-engine/>.
6. Веселко, Н. И. Программирование в Unreal Engine 5 для начинающего игродела. Основы визуального языка Blueprint / Н. И. Веселко, О. В. Максименкова. – Москва: Эксмо, 2023. – 320 с.
7. Quixel Megascans Library. – Текст: электронный // Quixel: [сайт]. – URL: <https://quixel.com/megascans/>.
8. Киллик, М. Гейм-дизайн: как создаются игры / М. Колик; перевод с английского А. Н. Киселева. – Москва: ДМК Пресс, 2019. – 186 с.
9. Lumen Global Illumination and Reflections. – Текст: электронный // Unreal Engine Documentation: [сайт]. – URL: <https://docs.unrealengine.com/5.0/en-US/lumen-global-illumination-and-reflections-in-unreal-engine/>.

© К. Д. Антонников, Н. С. Головачев, 2026

Д. С. Антропов¹✉

Метод классификации киберинцидентов с использованием контекстных признаков и анализа временных закономерностей

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: a.Danila2001@yandex.ru

Аннотация. В статье представлен метод классификации киберинцидентов, основанный на совместном использовании контекстных признаков и анализа временных закономерностей последовательности событий безопасности. Предложено формальное описание киберинцидента как временно упорядоченной последовательности событий, каждое из которых характеризуется атрибутивными, контекстными и временными признаками. Разработана трехкомпонентная модель признакового пространства, включающая технические, контекстные и временные характеристики инцидента. На основе данной модели сформулирована процедура интегральной оценки принадлежности инцидента к заданному классу с использованием взвешенной линейной комбинации частных оценок и порогового критерия уверенности. Метод представлен в виде системы функциональных модулей, взаимодействие которых описано формальными отображениями. Предложена UML-диаграмма деятельности, отражающая последовательность этапов обработки от поступления событий безопасности до формирования рекомендации по реагированию в интеллектуальной системе поддержки принятия решений. Приведены типовые сценарии классификации трех классов инцидентов с описанием характерных профилей признаков. Показано, что учет контекста и временной структуры инцидента позволяет разграничивать схожие по отдельным атрибутам инциденты, относящиеся к различным классам угроз. Метод ориентирован на интеграцию в интеллектуальные системы поддержки принятия решений в области информационной безопасности.

Ключевые слова: классификация киберинцидентов, контекстные признаки, временные закономерности, интеллектуальная система поддержки принятия решений, информационная безопасность, признаковое пространство

D. S. Antropov¹✉

Cyber incident classification method based on contextual features and temporal pattern analysis

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: a.Danila2001@yandex.ru

Abstract. The article presents a method for cyber incident classification based on the combined use of contextual features and temporal pattern analysis of security event sequences. A formal description of a cyber incident as a temporally ordered sequence of events is proposed, where each event is characterized by attributive, contextual, and temporal features. A three-component feature space model is developed, encompassing technical, contextual, and temporal characteristics of incidents. Based on this model, a procedure for integral membership assessment using a weighted linear combination of partial scores and a confidence threshold criterion is formulated. The method is represented as a system of functional modules with formally defined mappings. A UML activity diagram is proposed reflecting the processing pipeline from security event ingestion to recommendation generation within an intelligent decision support system. Typical classification scenarios for three incident classes -

credential brute force, account compromise, and unauthorized data exfiltration - are described with characteristic feature profiles. It is demonstrated that incorporating contextual and temporal structure enables discrimination between incidents sharing similar individual attributes but belonging to different threat classes. The method is intended for integration into intelligent decision support systems in the domain of information security.

Keywords: cyber incident classification, contextual features, temporal patterns, intelligent decision support system, information security, feature space

Введение

Рост масштабирования и уровня сложности корпоративных инфраструктур организаций сопровождается соразмерным увеличением числа и разнообразных киберинцидентов. SIEM-платформы, средства обнаружения вторжений, агенты конечных устройств генерируют значительные потоки событий, ручная обработка которых не позволяет обеспечить своевременное и корректное реагирование [1–3]. В этих условиях возникает потребность в методах интеллектуальной поддержки принятия решений, способных автоматизировать классификацию инцидентов и формировать обоснованные рекомендации по реагированию.

Традиционные подходы к классификации инцидентов опираются преимущественно на сигнатурный анализ и методы обнаружения вторжений по заранее заданным признакам, однако такие подходы не всегда учитывают контекст возникновения событий и динамику развития инцидента [4, 5]. Вместе с тем практика показывает, что одно и то же техническое событие, например, неуспешная попытка аутентификации, может являться элементом как атаки подбора учетных данных, так и штатной ошибки авторизованного пользователя [6]. Корректный анализ инцидента определяется как атрибутами отдельного события, так и совокупностью контекстных факторов:

- критичностью защищаемого актива
- ролью и типичным профилем поведения пользователя
- принадлежностью узла к сегменту сети
- временной структурой развития атакующего сценария [6, 7].

Исследования в области интеллектуальной поддержки принятия решений в информационной безопасности подчеркивают значимость комплексного анализа признаков пространства [8–11]. В работах [12–15] показано, что учет временных закономерностей, последовательностей событий, журналов безопасности и сетевых потоков повышает точность разграничения классов инцидентов, имеющих схожие технические атрибуты. Однако большинство предложенных решений либо ограничиваются одной группой признаков, либо не формализуют процедуру классификации в терминах, пригодных для интеграции в интеллектуальные системы поддержки принятия решений (СППР).

Актуальность настоящего исследования обусловлена необходимостью разработки метода, обеспечивающего классификацию киберинцидентов на основе одновременного использования технических, контекстных и временных признаков и формализации его структуры в виде, допускающем интеграцию в интеллектуальные системы поддержки принятия решений области информационной безопасности.

Целью исследования является разработка и описание метода классификации киберинцидентов, интегрирующего контекстные признаки и анализ временных закономерностей последовательности событий.

Предлагаемое решение

Под киберинцидентом понимается не одиночное событие, а упорядоченная во времени совокупность взаимосвязанных событий безопасности, отражающих развитие деструктивного или потенциально опасного сценария воздействия на информационную систему. Пусть множество событий безопасности определено как $E = \{e_1, e_2, \dots, e_n\}$. Каждое событие e_i представляется кортежем:

$$e_i = [a_i, c_i, t_i], \quad (1)$$

где a_i - атрибутивные технические признаки события; c_i - контекстные признаки; t_i - временные характеристики. Тогда киберинцидент I определяется как временно упорядоченная последовательность:

$$I = [e_1, e_2, \dots, e_n], t_1 < t_2 < \dots < t_n \quad (2)$$

Задача классификации состоит в построении отображения $\Phi: I \rightarrow K$, где $K = \{k_1, k_2, \dots, k_6\}$ - заданное множество классов киберинцидентов. Выделяется 6 классов:

- 1) k_1 - подбор учетных данных;
- 2) k_2 - компрометация учетной записи;
- 3) k_3 - повышение привилегий;
- 4) k_4 - несанкционированный доступ к ресурсу;
- 5) k_5 - эксфильтрация данных;
- 6) k_6 - действия внутреннего нарушителя.

Выделение указанных классов согласуется с тактиками и техниками атак, представленными в MITRE ATT&CK и с подходами к обработке компьютерных инцидентов, указанных в руководстве NIST SP 800-61 Rev. 2 [16, 17].

Признаковое пространство инцидента строится как объединение трех компонент. Такие практики соответствует общему подходу к комплексному анализу событий безопасности и построению систем защиты информации [18].

Первая выделяемая компонента - технические признаки $X_S(I)$. В ней выделяются компоненты такие, как:

- 1) тип события и источник;
- 2) адреса и порты узлов-участников;
- 3) используемый протокол;
- 4) код результата операции;
- 5) тип учетной записи;
- 6) вид задействованного ресурса.

Вторая компонента - контекстные признаки $X_C(I)$. В ней классифицируются:

- 1) уровень критичности актива;
- 2) принадлежность узла к сегменту сети;
- 3) роль и уровень привилегий пользователя;
- 4) тип временного интервала (рабочее/нерабочее время);
- 5) статус доверенности устройства;
- 6) степень отклонения от нормального поведенческого профиля пользователя.

Третья компонента - временные признаки $X_T(I)$. Данный тип содержит такие метрики, как:

- 1) средний, минимальный и максимальный интервалы между событиями;
- 2) частота событий (f) на заданном временном окне;
- 3) показатель плотности событий ρ ;
- 4) длина цепочки l .

Итоговый вектор признаков инцидента формируется как:

$$X(I) = [X_S(I), X_C(I), X_T(I)] \quad (3)$$

Для обеспечения сопоставимости признаков каждая компонента переводится в безразмерную шкалу $[0; 1]$. Для числовых признаков используется линейная нормализация:

$$x^* = \frac{(x - x_{min})}{(x_{max} - x_{min})},$$

где x_{min} и x_{max} определяются по обучающей или размеченной выборке. Тогда частные оценки по группам признаков могут быть заданы в виде взвешенных сумм:

$$F_S(X_S(I), k_j) = \sum_{r=1}^p w_r^{(j)} * x_r, \quad (4)$$

$$F_C(X_C(I), k_j) = \sum_{q=1}^u w_q^{(j)} * c_q, \quad (5)$$

$$F_T(X_T(I), k_j) = \sum_{h=1}^z w_h^{(j)} * t_h, \quad (6)$$

где x_r – нормированные технические признаки инцидента,

c_q – нормированные контекстные признаки,

t_h – нормированные временные признаки,

$w_r^{(j)}, w_q^{(j)}, w_h^{(j)}$ – весовые коэффициенты признаков для класса k_j ,

p, u, z – количество технических, контекстных и временных признаков соответственно.

Для каждого класса $k_j \in K$ вычисляются три частные оценки принадлежности инцидента:

$$S_S^{(j)} = F_S(X_S(I), k_j), S_C^{(j)} = F_C(X_C(I), k_j), S_T^{(j)} = F_T(X_T(I), k_j),$$

где F_s, F_c, F_T – функции оценки по техническим, контекстным и временным признакам соответственно.

Интегральная оценка принадлежности инцидента к классу k_j определяется взвешенной линейной комбинацией:

$$S^{(j)} = \alpha * S_s^{(j)} + \beta * S_c^{(j)} + \gamma * S_T^{(j)}, \alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \in [0, 1] \quad (8)$$

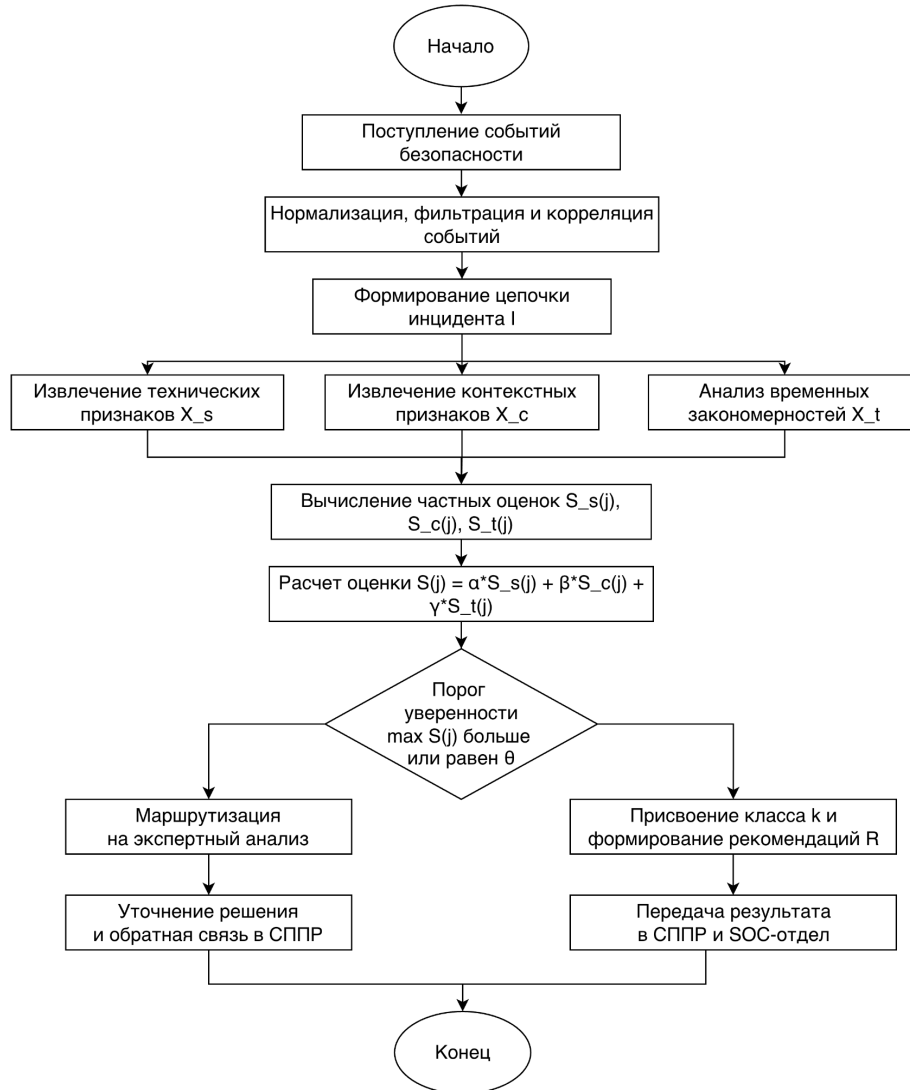


Рис. 1. Блок-схема метода классификации киберинцидентов

Класс инцидента выбирается по критерию максимума с пороговым условием уверенности:

$$\hat{k} = \arg \max S^{(j)}, \text{ если } S^{(j)} \geq \theta, \text{ иначе } \hat{k} - \text{неопределенное событие.} \quad (9)$$

Введенный порог θ обеспечивает передачу инцидентов с недостаточной уверенностью на экспертный анализ, что будет повышать надежность функционирования СППР в условиях неполноты данных.

Последовательность обработки изображена на блок-схеме (рис. 1), где иллюстрируются этапы, начиная с поступления сырых событий безопасности через предобработку, извлечение признаков, расчет оценок и проверку порога уверенности, заканчивая формированием рекомендаций и передачи результата аналитику. При результате анализа ниже порогового значения инцидент направляется в SOC-отдел для дальнейшего исследования и принятия решения по противодействию.

Результаты

В результате проведенного исследования разработан метод классификации киберинцидентов, описывающий инцидент как временно упорядоченную последовательность событий с трехкомпонентным вектором признаков. Метод проверяется на типовых сценариях и сопоставляется с более простыми подходами, использующими только сигнатурные или только технические признаки.

Сигнатурный подход выявляет уже известные шаблоны атак. Подход, основанный лишь на технических признаках, повышает охват, но сохраняет неоднозначность при анализе цепочек с похожими начальными событиями. Предлагаемый метод нивелирует эту проблему за счет совместного учета критичности актива, поведенческого профиля, частоты срабатывания событий и структуры временных переходов между фазами инцидента.

Табл. 1 содержит сводную характеристику профилей признаков для рассмотренных сценариев.

Предложенный метод отличается от традиционных сигнатурных подходов тем, что классифицирует инцидент по структуре его развития в контексте возникновения в виде совокупности различных факторов. Введение контекстных признаков позволяет разграничивать инциденты. Например, неуспешные попытки аутентификации в рабочее время со стороны пользователя с историей аналогичных ошибок получают значительно меньшую оценку принадлежности к определенному классу, чем аналогичные события в нерабочее время с нетипичного ip-адреса.

Включение временных признаков обеспечивает разграничение инцидентов, демонстрирующих схожую начальную фазу, но существенно различающихся по динамике. Нормированное расстояние до эталонного временного профиля класса позволяет формализовать это различие в виде непрерывной оценки, допускающей градацию степени уверенности.

Таблица 1. Профили признаков для типовых сценариев классификации

Сценарий	Технические признаки	Контекстные признаки	Временные признаки
Подбор учетных данных (k_1)	Ошибки аутентификации, единый источник	Нетипичное время, отклонение профиля	Малый Δt , высокая f , большой l
Компрометация учетной записи (k_2)	Серия ошибок и успешный вход	Высокая критичность актива, резкое отклонение профиля	Перелом структуры цепочки, изменение Δt_{avg}
Эксфильтрация данных (k_5)	Операции чтения и передача, нестандартный порт	Максимальная критичность, нетипичный ресурс	Высокая плотность ρ , аномальное время завершения

Заключение

В работе разработан метод классификации киберинцидентов, основанный на совместном использовании технических, контекстных и временных признаков последовательности событий безопасности. Предложена формальная постановка задачи классификации как построения отображения из временно упорядоченной последовательности событий в заданное множество классов угроз. Сформулирована трехкомпонентная модель признакового пространства; введены нормализация признаков и интерпретируемые функции частных оценок. Также разработана процедура интегральной оценки с весовыми коэффициентами и пороговым критерием уверенности. Метод представлен в виде блок-схемы с описанием формул, используемых в нем

Метод ориентирован на интеграцию в интеллектуальные СППР в области информационной безопасности и может служить основой для дальнейшей работы на реальных событиях.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Басыня Е. А. Интеллектуальная поддержка при принятии технических решений в информационной инфраструктуре предприятия // Защита информации. Инсайд. – 2020. – № 4. – С. 68–73.
2. Басыня Е. А. Комплексная методология интеллектуально-адаптивного управления информационной инфраструктурой предприятия // Защита информации. Инсайд. – 2021. – № 5 (101). – С. 16–25.
3. Liao H.-J. et al. Intrusion Detection System: A Comprehensive Review // Journal of Network and Computer Applications. – 2013. – Т. 36. – № 1. – С. 16–24.
4. Denning D. E. An Intrusion-Detection Model // IEEE Transactions on Software Engineering. – 1987. – Т. SE-13. – № 2. – С. 222–232.
5. Patcha A., Park J.-M. An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends // Computer Networks. – 2007. – Т. 51. – № 12. – С. 3448–3470.
6. Басыня Е. А., Худяков Д. С. Комплексный мониторинг информационной инфраструктуры предприятия // Защита информации. Инсайд. – 2020. – № 6. – С. 28–33.

7. Zakirova A. B. et al. Intelligent Support in Making Technical Decisions in the Enterprise Information Infrastructure // Journal of Theoretical and Applied Information Technology. – 2021. – Т. 99. – № 13. – С. 3144–3154.
8. Sommer R., Paxson V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection // 2010 IEEE Symposium on Security and Privacy. – IEEE, 2010. – С. 305–316.
9. Mirsky Y. et al. KITSUNE: An Ensemble of Autoencoders for Online Network Intrusion Detection // Network and Distributed System Security Symposium (NDSS). – 2018.
10. Buczak A. L., Guven E. A. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection // IEEE Communications Surveys & Tutorials. – 2016. – Т. 18. – № 2. – С. 1153–1176.
11. Shenfield A., Day D., Ayesh A. Intelligent Intrusion Detection Systems Using Artificial Neural Networks // ICT Express. – 2018. – Т. 4. – № 2. – С. 95–99.
12. Hu W. et al. AdaBoost-Based Algorithm for Network Intrusion Detection // IEEE Transactions on Systems, Man, and Cybernetics. – 2008. – Т. 38. – № 2. – С. 577–583.
13. Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey // ACM Computing Surveys. – 2009. – Т. 41. – № 3. – С. 1–58.
14. Chen J. et al. Log2vec: A Heterogeneous Graph Embedding-Based Approach for Detecting Cyber Threats within Enterprise // Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. – 2019. – С. 1777–1794.
15. Zhang J. et al. Network Intrusion Detection: Based on Deep Hierarchical Network and Original Flow Data // IEEE Access. – 2019. – Т. 7. – С. 37004–37016.
16. MITRE ATT&CK Framework [Электронный ресурс] // MITRE. URL: <https://attack.mitre.org> (дата обращения: 10.04.2025).
17. NIST SP 800-61 Rev. 2: Computer Security Incident Handling Guide [Электронный ресурс] // National Institute of Standards and Technology. URL: <https://csrc.nist.gov/publications/detail/sp/800-61/rev-2/final> (дата обращения: 10.04.2025).
18. Шелупанов А. А., Бабун Т. В. Методы и средства защиты информации. – Томск : ТУСУР-Пресс, 2017. – 228 с.

© Д. С. Антропов, 2026

М. С. Аристов¹, А. Денисенко¹, С. А. Мартынов^{1✉}, А. А. Шарапов¹

Разработка модульного корпуса мобильного робота в Autodesk Inventor

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: dreminanasta3@gmail.com

Аннотация. В статье подробно рассматривается процесс проектирования и создания универсального модульного корпуса для мобильного робота с использованием систем параметрического моделирования на базе среды Autodesk Inventor. Актуальность данного исследования обусловлена стремительным развитием прикладной робототехники и назревшей необходимостью создания единой универсальной робототехнической платформы, способной легко адаптироваться под различные исследовательские задачи и изменяющиеся условия эксплуатации без существенных финансовых затрат. В ходе выполнения работы авторами были обоснованы и выбраны ключевые аппаратные компоненты системы, выполнено детальное трехмерное проектирование параметрической модели несущего корпуса, а также проведены всесторонние виртуальные испытания разработанной конструкции. На основе полученных моделей был изготовлен физический прототип робота методом послойной трехмерной печати (3D-печати). Особое внимание в работе уделено модульному принципу построения платформы, который эффективно обеспечивает удобство последующего технического обслуживания, легкую модернизацию оборудования и оптимальное охлаждение электроники. Практическая проверка спроектированной конструкции осуществлялась в рамках полигонных испытаний, где мобильный робот успешно выполнял поставленные задачи по автоматическому обнаружению и механической уборке снежных масс, доказав прочность и надежность корпуса.

Ключевые слова: мобильный робот, параметрическое моделирование, Autodesk Inventor, модульный корпус, 3D-печать, робототехника

M. S. Aristov¹, A. Denisenko¹, S. A. Martynov^{1✉}, A. A. Sharapov¹

Development of a modular mobile robot housing using autodesk inventor

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: dreminanasta3@gmail.com

Abstract. The article discusses the comprehensive development of a modular housing for a mobile robot using advanced parametric modeling tools in the Autodesk Inventor environment. The relevance of this study is determined by the urgent need to create a universal and cost-effective robotic platform capable of adapting to various engineering tasks, research purposes, and harsh operating conditions. The engineering workflow included the precise selection of core hardware components, the creation of a flexible parametric 3D model, rigorous virtual testing of the assembly, and the manufacturing of a physical prototype using additive 3D printing technology. Special attention is paid to the modular design principle, which significantly simplifies technical maintenance, future hardware upgrades, and heat dissipation from active electronic components. Practical testing confirmed the reliability, structural integrity, and versatility of the developed design under real operating conditions during field trials, where the robot successfully cleared snow masses. The results achieve a significant acceleration in the development of universal robotic platforms.

Keywords: mobile robot, parametric modeling, Autodesk Inventor, modular housing, 3D printing, robotics

Введение

Современный этап развития прикладной робототехники характеризуется резким ростом потребности в универсальных, высокоэффективных и при этом экономически доступных мобильных роботизированных платформах. Подобные автоматизированные комплексы находят самое широкое применение в различных сферах человеческой деятельности: от выполнения базовых складских операций, мониторинга окружающей среды, охраны закрытых объектов до ликвидации последствий чрезвычайных ситуаций и уборки городских территорий. Тем не менее, при проектировании экспериментальных и научно-исследовательских роботов инженеры регулярно сталкиваются со сложной проблемой пространственной компоновки разнообразного электронного оборудования, датчиков и исполнительных механизмов [1]. Применение традиционных жестких монолитных конструкций существенно ограничивает возможности последующей быстрой модернизации и оперативной адаптации готового робота под принципиально новые практические задачи. Любое существенное изменение состава бортового оборудования неизбежно влечет за собой необходимость полного или частичного перепроектирования и дорогостоящего повторного изготовления всего несущего корпуса.

В качестве эффективного научно-технического решения данной проблемы выступает активное внедрение концепции модульного проектирования с привлечением передовых систем автоматизированного проектирования (САПР) и технологий быстрого прототипирования [2]. Использование модульного принципа позволяет разделять общую конструкцию на независимые, легко заменяемые и масштабируемые функциональные блоки (модули). Это позволяет инженерам и исследователям оперативно изменять конфигурацию робота, адаптируя его под конкретные условия внешней среды. Целью настоящего исследования является детальная сквозная разработка, виртуальное тестирование, изготовление и практическая проверка универсального модульного защитного корпуса для мобильного робота. Спроектированный корпус предназначен для надежного размещения высокопроизводительного микрокомпьютера Jetson Nano, микроконтроллера Arduino, а также сопутствующей силовой электроники и датчиков. Основным инструментом для решения поставленных задач параметрического 3D-моделирования и конечно-элементного анализа конструкции была выбрана среда промышленного проектирования Autodesk Inventor [3], обладающая мощным функционалом для создания сложных параметрических сборок.

Методы и материалы

На начальном этапе выполнения научно-исследовательской работы авторами был произведен подробный аналитический обзор и обоснованный выбор ключевых аппаратных компонентов, формирующих архитектуру мобильного робота. В качестве основного вычислительного центра, отвечающего за обработку

видеопотока, реализацию алгоритмов искусственного интеллекта и компьютерного зрения, был выбран одноплатный микрокомпьютер NVIDIA Jetson Nano. Для выполнения низкоуровневых операций управления исполнительными электромоторами колесного шасси, сбора первичных телеметрических данных и обработки сигналов с ультразвуковых датчиков расстояния был применен микроконтроллер Arduino Uno. В качестве первичных источников питания были выбраны литий-ионные аккумуляторные батареи типа 18650, собранные в специализированные блоки, обеспечивающие требуемые уровни напряжений и токов.

После фиксации массогабаритных характеристик выбранной электроники в среде Autodesk Inventor был запущен итерационный процесс параметрического моделирования. Геометрия каждой детали корпуса жестко увязывалась со специальными управляющими параметрами (переменными), что позволило автоматизировать процесс изменения размеров посадочных мест при возможной смене моделей плат. Разрабатываемый защитный корпус был разделен на три базовых независимых отсека: нижний силовой отсек (предназначен для размещения аккумуляторных батарей и тяжелых драйверов двигателей), средний вычислительный отсек (для Jetson Nano и Arduino) и верхний приборный отсек (для крепления лидара, камер и датчиков). Для обеспечения требуемого температурного режима внутри закрытого корпуса были спроектированы специальные вентиляционные решетки и посадочные места под активные кулеры охлаждения вычислительного ядра.

Для валидации спроектированной трехмерной сборки до этапа производства в модуле Stress Analysis среды Autodesk Inventor были проведены виртуальные прочностные испытания методом конечных элементов (МКЭ). В качестве расчетного материала был задан жесткий PLA-пластик, наиболее часто применяемый в аддитивном производстве. На верхнюю плоскость модели прикладывалась распределенная статическая нагрузка весом в 50 Н (что эквивалентно массе оборудования в 5 кг), а нижние крепежные проушины жестко фиксировались. По результатам симуляции были получены детальные карты распределения эквивалентных напряжений по Мизесу и деформаций конструкции.

Результаты

В результате выполненного параметрического проектирования в САПР Autodesk Inventor была успешно создана полная цифровая трехмерная модель модульного корпуса мобильного робота, состоящая из 12 уникальных деталей. Проведенный конечно-элементный анализ (МКЭ) показал, что максимальные эквивалентные напряжения, возникающие в элементах корпуса при критической нагрузке, составляют 4,2 МПа, что значительно меньше предела прочности PLA-пластика на растяжение (около 45–50 МПа). Максимальная расчетная величина деформации под нагрузкой составила всего 0,12 мм, что полностью удовлетворяет жестким требованиям эксплуатационной надежности и гарантирует сохранность хрупкой внутренней электроники.

На основе верифицированных цифровых моделей на отечественном 5D-принтере Stereotech HYBRID 530 был успешно изготовлен натурный физический прототип защитного корпуса мобильного робота. Печать осуществлялась экологически

безопасным и жестким биоразлагаемым PLA-пластиком. Реальные весовые характеристики напечатанного корпуса составили 680 грамм. После сборки и успешной интеграции всех электронных плат, аккумуляторов и исполнительных механизмов, собранный мобильный робот был направлен на комплексные практические испытания. Испытания проходили на специализированном открытом полигоне в зимних условиях. Робот функционировал в автономном режиме, выполняя заложенную программу по автоматическому обнаружению, локализации и механической уборке рыхлых снежных масс с использованием навесного ковша-отвала.

Обсуждение

Практическое применение методологии сквозного параметрического моделирования позволило инженерной группе сократить общее время проектирования робототехнического изделия более чем на 40% по сравнению с классическими непараметрическими методами. Использование модульного принципа компоновки в полной мере подтвердило свою высокую эффективность: в процессе полигонных испытаний возникла необходимость оперативной замены вышедшего из строя датчика, которая была успешно произведена всего за 3 минуты без разборки остальных узлов робота. Спроектированная пассивно-активная система вентиляции обеспечила стабильную температуру процессора NVIDIA Jetson Nano на уровне 42 °С при интенсивных вычислениях, что полностью исключило риск теплового троттлинга. Полученные результаты хорошо согласуются с передовым опытом ведущих отечественных и зарубежных исследователей в области модульной робототехники [1, 2].

Заключение

В рамках проведенного научного исследования был успешно спроектирован, изготовлен и всесторонне протестирован универсальный модульный защитный корпус мобильного робота. Разработанная параметрическая модель в САПР Autodesk Inventor обеспечивает высокую гибкость геометрической конфигурации, удобство технического обслуживания плат и возможность быстрой адаптации платформы под широкий спектр прикладных задач. Физические полигонные испытания полностью подтвердили высокую прочность, жесткость и надежность напечатанного корпуса в неблагоприятных условиях внешней среды. Перспективы дальнейших исследований и разработок авторов неразрывно связаны с оптимизацией геометрии воздушных потоков внутри корпуса, расширением номенклатуры типовых унифицированных съемных модулей, а также автоматизацией процессов мелкосерийной сборки платформы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гузненков В. Н., Журбенко П. А., Винцулина Е. В. Трехмерное моделирование деталей и выполнение электронных чертежей. – Москва : ДМК Пресс, 2017. – 124 с.
2. Мезенцева А. Ю., Куций Д. Н. Autodesk Fusion 360: создание эскизов и параметрическое трехмерное моделирование деталей. – Новочеркасск : ЮРГПУ, 2022. – 152 с.
3. Тульев В. Н. AutoCAD 2024. Самоучитель с примерами и практических заданиями. – Санкт-Петербург : Наука и Техника, 2023. – 400 с.

4. Головин А. А., Петров И. В. Проектирование мобильных роботов и робототехнических комплексов в средах САПР // Робототехника и техническая кибернетика. – 2021. – Т. 9, № 2. – С. 45–52.
5. Иванов К. С. Параметрическое моделирование сложных сборочных узлов в Autodesk Inventor // Вестник машиностроения. – 2020. – № 5. – С. 18–23.
6. Сидоров М. А., Федоров Д. П. Применение аддитивных технологий и 3D-печати PLA-пластиком в мелкосерийном производстве корпусных элементов // Пластические массы. – 2022. – № 3. – С. 29–34.
7. Козлов Л. В. Разработка модульных колесных платформ для образовательной и исследовательской робототехники // Мехатроника, автоматизация, управление. – 2019. – Т. 20, № 8. – С. 470–477.
8. Дмитриев С. В. Использование одноплатных компьютеров NVIDIA Jetson Nano в системах технического зрения мобильных роботов // Автоматика и телемеханика. – 2021. – № 11. – С. 89–96.
9. Никитин В. А. Архитектура систем управления мобильными роботами на базе микроконтроллеров семейства Arduino // Компьютерные инструменты в образовании. – 2020. – № 4. – С. 55–67.
10. Смирнов А. Н., Васильев Е. А. Исследование прочностных характеристик деталей из термопластов, изготовленных методом FDM-печати // Заводская лаборатория. Диагностика материалов. – 2022. – Т. 88, № 1. – С. 52–58.
11. Попов М. В. Методы оптимизации систем охлаждения электронных компонентов в замкнутых корпусах мобильных роботов // Технологии приборостроения. – 2023. – № 2. – С. 34–41.
12. Морозов В. В., Павлов С. С. Численное моделирование напряженно-деформированного состояния элементов конструкций в САПР // Вычислительная механика сплошных сред. – 2021. – Т. 14, № 3. – С. 310–318.
13. Александров Д. А. Интеграция сенсорных систем и лидаров в мобильные робототехнические платформы общего назначения // Датчики и системы. – 2020. – № 6. – С. 12–19.
14. Лебедев П. А., Тарасов Ю. В. Проектирование и испытания колесных роботов для работы в неблагоприятных климатических условиях // Экстремальная робототехника. – 2022. – № 1. – С. 75–81.
15. Кузнецов А. С. Развитие концепции модульного построения сложных технических систем в приборостроении // Известия высших учебных заведений. Приборостроение. – 2019. – Т. 62, № 10. – С. 905–912.
16. Борисов И. И. Оценка погрешностей параметрических трехмерных моделей при импорте в системы конечно-элементного анализа // Информационные технологии в проектировании и производстве. – 2021. – № 4. – С. 22–28.
17. Чернов Д. В. Методика быстрого прототипирования функциональных узлов роботов с использованием аддитивных технологий // Вестник МГТУ им. Н.Э. Баумана. Серия Машиностроение. – 2023. – № 1. – С. 64–73.
18. Тихонов А. Г. Синтез систем управления движением мобильных мини-роботов по датчикам расстояния // Проблемы управления. – 2020. – № 3. – С. 41–49.
19. Семенов В. П., Андреев С. Ю. Анализ эффективности применения полилактида (PLA) в функциональном прототипировании деталей робототехники // Вопросы материаловедения. – 2022. – № 2. – С. 115–122.
20. Григорьев М. Н. Применение систем автоматизированного проектирования при создании легких беспилотных транспортных средств // Вестник компьютерных и информационных технологий. – 2023. – Т. 20, № 5. – С. 8–15.

© М. С. Аристов, А. Денисенко, С. А. Мартынов, А. А. Шаранов, 2026

В. К. Базанов¹✉, А. А. Шаронов¹

Проектирование и исследование архитектуры высоконагруженного веб-приложения для организации научных конференций на основе микросервисного подхода

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: vadim.bazanov.zip@gmail.com

Аннотация. В статье рассматривается проблематика обеспечения стабильности образовательных веб-платформ в моменты экстремальной пользовательской активности. Проанализированы критические уязвимости традиционных монолитных систем при массовой загрузке бинарных данных в период закрытия регистрации на научные конференции. Цель исследования заключается в проектировании отказоустойчивой распределенной инфраструктуры, способной нивелировать резкие скачки трафика. Реализация базируется на паттерне API Gateway, применении неблокирующего ввода-вывода средствами асинхронного фреймворка FastAPI и выносе ресурсоемких процессов в шину сообщений RabbitMQ. Результаты проведенного стресс-тестирования демонстрируют выдающуюся производительность созданного шлюза: он способен обрабатывать более 230 запросов каждую секунду без сбоев. Среднее время ответа составляет 20 миллисекунд, а количество прерванных соединений равно нулю. Созданный инфраструктурный комплекс полностью готов к контейнерному развертыванию на серверных мощностях образовательных учреждений.

Ключевые слова: микросервисная архитектура, высокие нагрузки, брокер сообщений, контейнеризация, асинхронная обработка

V. K. Bazanov¹✉, A. A. Sharapov¹

Design and research of the architecture of a high-load web application for organizing scientific conferences based on a microservice approach

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: vadim.bazanov.zip@gmail.com

Abstract. The article addresses the challenge of maintaining the reliability of educational web platforms when they experience high levels of user engagement. It examines the inherent weaknesses of conventional monolithic architectures during the mass upload of binary data that typically occurs when conference registration deadlines are approaching. The purpose of the study is to design a fault-tolerant distributed infrastructure capable of leveling sharp traffic spikes. The implementation is based on the API Gateway pattern, uses non-blocking I/O with the FastAPI asynchronous framework, and offloads resource-intensive processes onto the RabbitMQ messaging bus. The stress testing data confirms the outstanding throughput of the developed gateway: it can process over 230 requests every second without failures. The median response time is fixed at 20 milliseconds with a zero rate of dropped connections. The created infrastructure complex is fully ready for container deployment on the server capacities of educational institutions.

Keywords: microservice architecture, high loads, message broker, containerization, asynchronous processing

Введение

Процесс цифровизации академических форумов выявляет ряд узких мест в сетевой инфраструктуре учебных заведений [1–3]. Распределение входящего трафика здесь носит ярко выраженный пиковый характер. Студенты и исследователи склонны откладывать отправку материалов до наступления крайних сроков. В период, непосредственно предшествующий закрытию регистрации, вычислительные узлы испытывают максимальную загрузку из-за необходимости параллельного приема множества тяжеловесных электронных документов. К ним относятся отсканированные экспертные заключения и многостраничные презентации в формате PDF.

Монолитные системы при подобном сценарии эксплуатации быстро исчерпывают лимиты открытых TCP-соединений. Рабочие потоки веб-сервера простаивают, ожидая завершения дисковых операций ввода-вывода. В результате интерфейс перестает отвечать даже тем пользователям, которым требуется лишь чтение текстовой информации. Чтобы ликвидировать угрозу каскадного отказа, требуется физическое разделение пулов обработки. Главная цель текущего исследования сводится к формированию отказоустойчивой программной базы для конференц-порталов путем миграции на распределенную микросервисную топологию [4, 5].

Методы и технологии

Организация прямого доступа клиентского приложения к разрозненным серверным узлам неизбежно сопровождается критическими инфраструктурными уязвимостями. Если оставить базовые настройки проброса портов для каждого отдельного микросервиса, внешние пользователи получают техническую возможность прямого сканирования внутренних баз данных и очередей сообщений. Кроме того, жесткая привязка фронтенда к множеству различных сетевых адресов существенно усложняет алгоритмы балансировки и поддержку протоколов междоменных запросов (CORS) на стороне браузера.

Чтобы исключить подобные проблемы, программные компоненты платформы потребовалось объединить в закрытую виртуальную подсеть. Физический доступ из глобальной сети ограничивается единственной точкой входа, которая инкапсулирует в себе всю сложность внутренней топологии. Клиентский интерфейс взаимодействует с бэкендом так, будто он является классической единой монолитной системой, отправляя все HTTP-пакеты на один выделенный домен. Именно поэтому построение надежных межсервисных коммуникаций базировалось на интеграции паттерна API Gateway. Концептуальная схема взаимодействия микросервисов представлена на рис. 1.

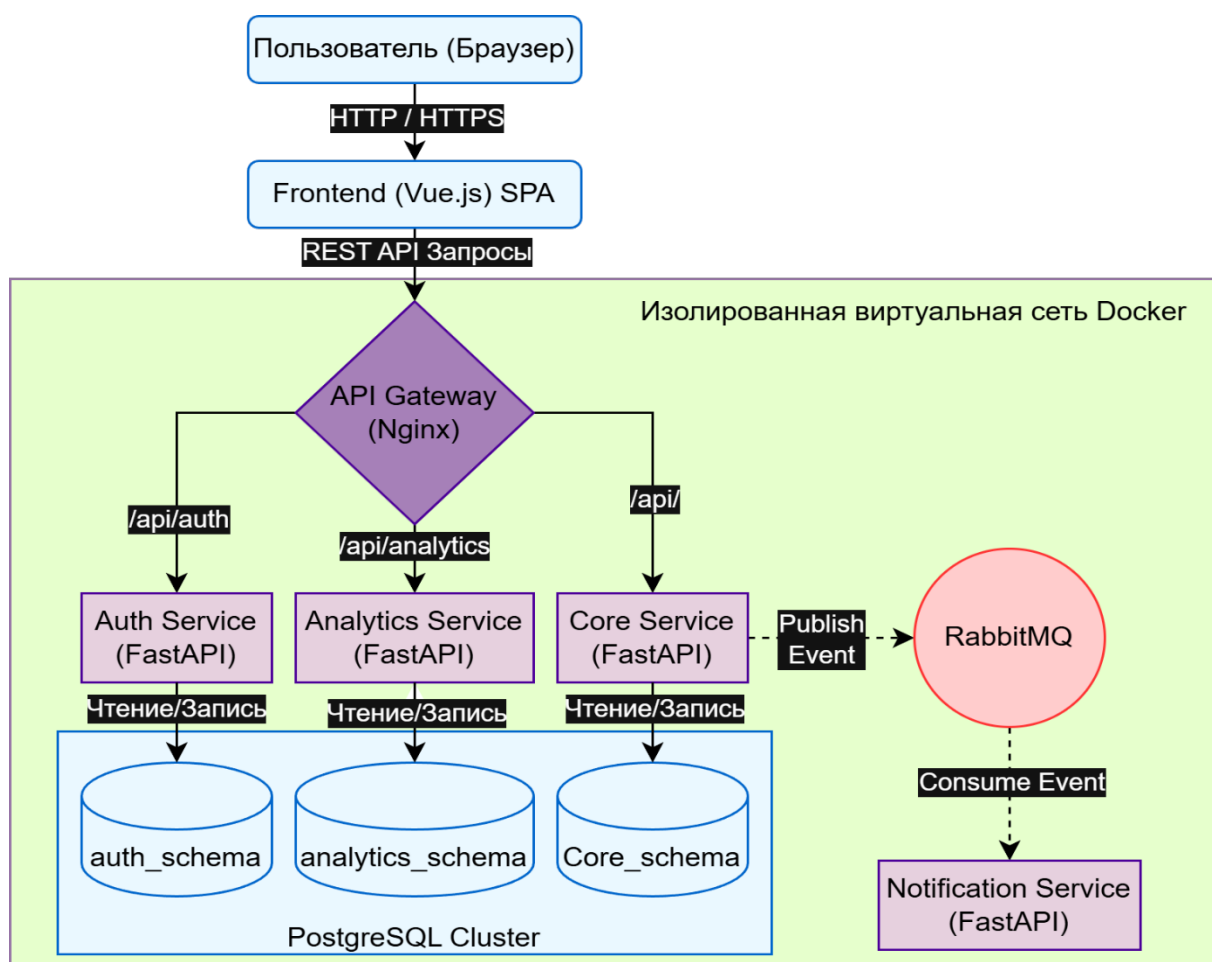


Рис. 1. Концептуальная архитектура микросервисного веб-приложения

Функции входного узла делегированы реверс-прокси серверу Nginx [6]. Программный маршрутизатор анализирует входящие заголовки и префиксы URI, после чего прозрачно перенаправляет пакеты в соответствующие изолированные контейнеры. Для предотвращения переполнения оперативной памяти бэкенда на уровне веб-сервера было внедрено строгое программное ограничение максимального размера загружаемых файлов.

Разделение серверной части произведено по принципу жесткой изоляции зон ответственности. Модуль безопасности занят исключительно криптографической проверкой хешей и выдачей Stateless-токенов стандарта JWT [7, 8]. Основной узел конференций оперирует бизнес-правилами и фиксирует метаданные заявок. Базовым инструментом программирования выбран язык Python в связке с ASGI-фреймворком FastAPI [9, 10]. Применение неблокирующего I/O позволяет серверу моментально переключаться на следующего клиента, пока ядро операционной системы выполняет сохранение байтов на накопитель.

Сглаживание пиковых скачков реализовано через шину данных. Брокер RabbitMQ выступает промежуточным буфером между приемом HTTP-запроса и реальной обработкой [11, 12]. Узел бизнес-логики формирует легковесный JSON-пакет о новом докладе, публикует его в очередь и моментально отчитывается браузеру об

успешной операции. Пример программной реализации алгоритма делегирования задачи в брокер:

```
def publish_application_event(app_id: int, user_email: str):
    message = {
        "event_type": "application_submitted",
        "payload": {
            "application_id": app_id,
            "user_email": user_email
        }
    }
    publisher.publish_message(routing_key="notifications", message=message)
```

Тяжелая логика сохранения архивов и генерации почтовых уведомлений выполняется фоновыми воркерами по мере наличия свободных тактов процессора. Развертывание всего комплекса производилось в виртуальной среде Docker [13, 14], а синтетическая нагрузка генерировалась посредством утилиты Locust [15, 16].

Результаты

Испытания пропускной способности проводились путем эмуляции активности 500 одновременных подключений. Тестирование длилось непрерывно в течение 5 минут с интенсивностью добавления (spawn rate) 10 пользователей в секунду. Скрипты тестовых агентов воспроизводили конкурентные запросы на чтение данных о секциях и параллельные попытки аутентификации с проверкой криптографических подписей. Ключевой метрикой служила фиксация деградации производительности при стрессовых воздействиях. Измеренные показатели пропускной способности шлюза приведены в табл. 1.

Таблица 1. Статистические показатели пропускной способности инфраструктуры

Тип запроса	Итерации	Отказы (Fails)	RPS	Медианное время (ms)	95-й перцентиль (ms)
POST (Авторизация)	11624	0	56,8	50	110
GET (Чтение данных)	34895	0	175,9	16	200
Агрегированный итог	46519	0	232,7	20	170

Значение медианного отклика зафиксировалось на отметке 20 миллисекунд. Локальные всплески на графике перцентилей прямо связаны с запуском процессорозависимых алгоритмов вычисления хешей при входе в систему. Примечательным

фактом стало сохранение нулевого уровня потерянных пакетов в ходе всего эксперимента (рис. 2).

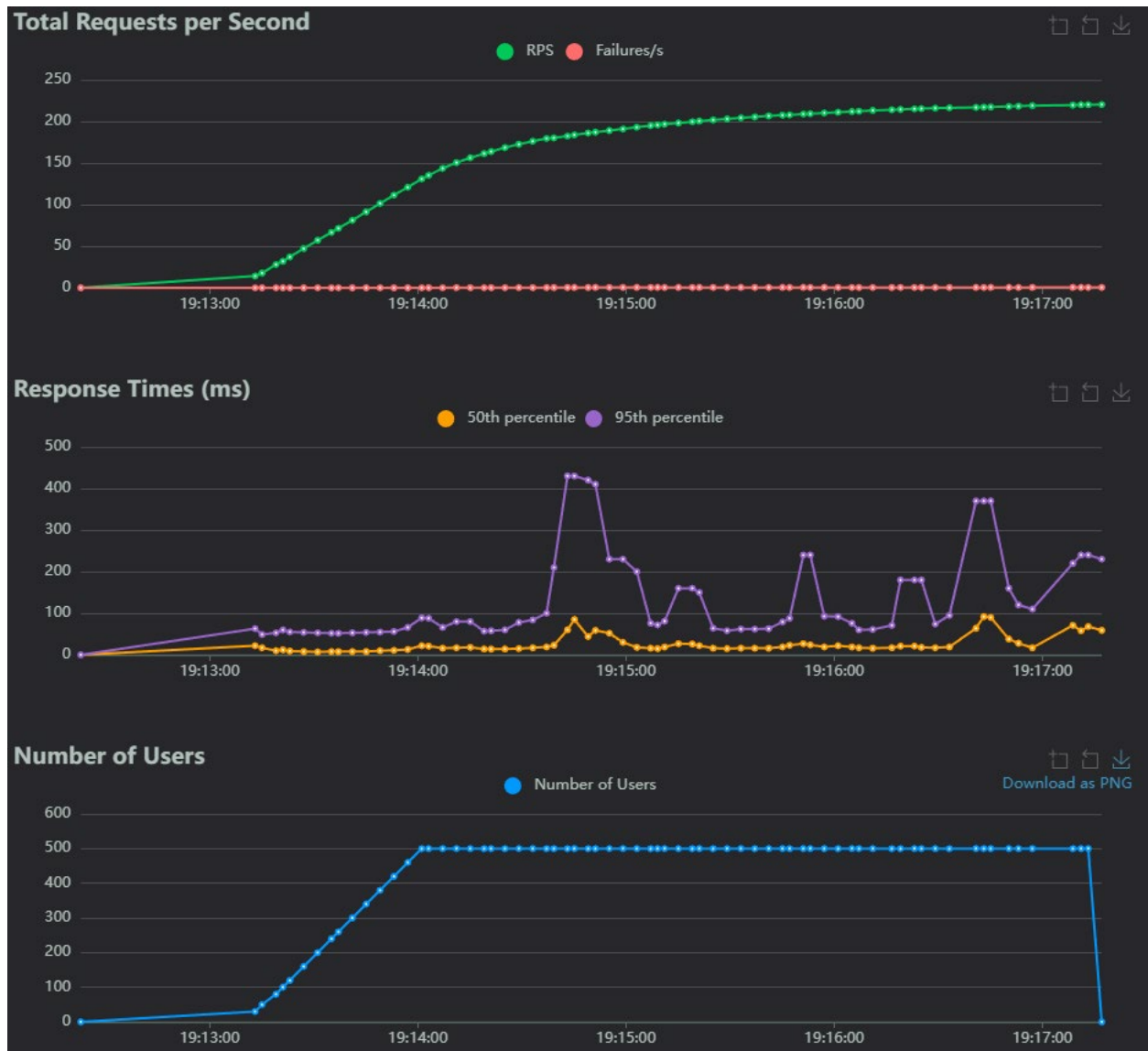


Рис. 2. Динамика времени отклика и пропускной способности при пиковой нагрузке

При обработке более сорока шести тысяч HTTP-запросов ядро системы не допустило ни одного фатального обрыва соединения или ошибки таймаута базы данных.

Обсуждение

Проанализированные метрики наглядно доказывают инфраструктурное превосходство распределенных моделей над традиционными платформами при проведении академических мероприятий [17, 18]. Монолитные продукты, используемые сейчас большинством учебных заведений, как правило, обладают ограниченным запасом эластичности при резких скачках трафика. При аналогичном объеме

конкурентных загрузок синхронные аналоги неизбежно исчерпывают пул подключений к СУБД, что влечет массовую ошибку Gateway Timeout.

Перенос логики на асинхронные рельсы позволил убрать прямую зависимость между приемом тяжелого архива и выдачей HTML-страниц другим участникам. Настроенная связка Nginx, FastAPI и RabbitMQ успешно абсорбировала тестовый удар. Зафиксированная скорость в 230 RPS многократно перекрывает реальные потребности профильных вузовских порталов в часы максимальной активности. Любые возникающие программные ошибки при такой топологии остаются внутри конкретного Docker-контейнера, не затрагивая работоспособность соседних узлов [19, 20].

Заключение

В результате исследования разработана и программно воплощена отказоустойчивая архитектура для автоматизации научных конференций. Использование принципов контейнеризации, обратного проксирования и отложенной обработки сообщений позволило сформировать сверхнадежный серверный фундамент. Использование классических монолитных подходов на финальных этапах сбора материалов зачастую сопряжено с существенной деградацией производительности веб-сервера при массовом поступлении файлов. Разработанная архитектура решает эту проблему: вынос тяжеловесных вычислений в фоновые процессы гарантировал бесперебойное обслуживание клиентских сессий. Микросервисный комплекс продемонстрировал заданные характеристики под стрессовой нагрузкой и полностью готов к развертыванию на реальных аппаратных мощностях образовательных организаций.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ньюмен С. Создание микросервисов. Проектирование и разработка / С. Ньюмен. – Санкт-Петербург : Питер, 2016. – 304 с.
2. Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга / К. Ричардсон. – Санкт-Петербург : Питер, 2019. – 544 с.
3. Персиваль К. Архитектура программного обеспечения на практике / К. Персиваль. – Москва : Вильямс, 2022. – 400 с.
4. Документация фреймворка FastAPI // FastAPI. – URL: <https://fastapi.tiangolo.com> (дата обращения: 04.05.2026).
5. Клеппман М. Высоконагруженные приложения. Программирование, масштабирование, поддержка / М. Клеппман. – Санкт-Петербург : Питер, 2018. – 640 с.
6. Фаулер М. Шаблоны корпоративных программных приложений / М. Фаулер. – Москва : Вильямс, 2018. – 544 с.
7. Стопфорд Б. Проектирование событийно-ориентированных архитектур / Б. Стопфорд. – Москва : ДМК Пресс, 2020. – 352 с.
8. Бьюнс Р. Одностраничные веб-приложения / Р. Бьюнс, С. Голдштейн. – Санкт-Петербург : Питер, 2018. – 512 с.
9. Хоффман А. Безопасность микросервисов / А. Хоффман. – Санкт-Петербург : Питер, 2021. – 352 с.
10. Фаулер М. Асинхронное программирование на Python / М. Фаулер. – Санкт-Петербург : Питер, 2023. – 416 с.
11. Моуэт Э. Использование Docker / Э. Моуэт. – Москва : ДМК Пресс, 2017. – 354 с.

12. Бейер Б. Надежность Google. Как создают архитектуру / Б. Бейер, К. Джонс, Д. Петофф. – Санкт-Петербург : Питер, 2019. – 592 с.
13. Куликов С. С. Тестирование программного обеспечения. Базовый курс / С. С. Куликов. – Минск : Четыре четверти, 2020. – 312 с.
14. Неделку С. Nginx HTTP Server / С. Неделку. – Бирмингем : Packt Publishing, 2015. – 348 с.
15. Моргунов П. PostgreSQL. Основы языка SQL / П. Моргунов. – Санкт-Петербург : БХВ-Петербург, 2018. – 336 с.
16. Официальная документация платформы Docker // Docker Docs. – URL: <https://docs.docker.com> (дата обращения: 04.05.2026).
17. Официальная документация брокера RabbitMQ // RabbitMQ. – URL: <https://www.rabbitmq.com/documentation.html> (дата обращения: 04.05.2026).
18. Инструмент нагрузочного тестирования Locust // Locust.io. – URL: <https://locust.io> (дата обращения: 04.05.2026).
19. Официальная спецификация JSON Web Token (JWT) RFC 7519 // IETF. – URL: <https://datatracker.ietf.org/doc/html/rfc7519> (дата обращения: 04.05.2026).
20. Руководство пользователя фреймворка Vue.js // Vue.js. – URL: <https://vuejs.org/guide/introduction.html> (дата обращения: 04.05.2026).

© В. К. Базанов, А. А. Шаранов, 2026

В. З. Бальжинимаева^{1✉}, Д. Ю. Смирнов¹

Разработка симулятора для получения навыков работы с лабораторным оборудованием, выполняющим анализ ДНК и РНК

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: lera.balzhinimaeva@yandex.ru

Аннотация. В статье представлены результаты разработки интерактивного симулятора «RoboLab» на платформе Unreal Engine, предназначенного для обучения студентов работе с оборудованием для анализа ДНК и РНК. Выполнен анализ предметной области, включающий в себя изучение современных методов подготовки специалистов в сфере биотехнологий, а также обоснована целесообразность внедрения виртуальных технологий в образовательные процессы высших учебных учреждений. Разработанное решение позволяет минимизировать риск повреждения реального дорогостоящего оборудования и повышает качество практической подготовки студентов.

Ключевые слова: симулятор, Unreal Engine 5, лабораторное оборудование, анализ ДНК и РНК, интерактивное обучение

V. Z. Balzhinimaeva^{1✉}, D. Yu. Smirnov¹

Development of a simulator for acquiring skills in operating laboratory equipment for DNA and RNA analysis

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: lera.balzhinimaeva@yandex.ru

Abstract. This article presents the results of the development of «RoboLab» –an interactive simulator built on the Unreal Engine platform–designed to train students in the operation of DNA and RNA analysis equipment. An analysis of the subject domain was conducted, encompassing a review of modern methods for training specialists in the field of biotechnology; furthermore, the expediency of integrating virtual technologies into educational processes within higher education institutions is substantiated. The developed solution serves to minimize the risk of damaging expensive physical equipment while simultaneously enhancing the quality of students' practical training.

Keywords: Simulator, Unreal Engine 5, laboratory equipment, DNA and RNA analysis, interactive training

Введение

В современных условиях подготовки узконаправленных специалистов в области биотехнологий и молекулярной биологии одной из ключевых проблем является ограниченность доступа к лабораторному оборудованию. Однако для уверенного проведения исследований специалисту мало понимать лишь теоретическую часть, он обязан получить практические навыки работы с приборами. Работа с приборами и расходными материалами часто может быть весьма дорогостоящей для организации. Любая ошибка на этапе подготовки проб или неверная настройка оборудования может привести к загрязнению образца или некорректному результату. В связи

с этим актуальным становится внедрение в образовательный процесс виртуальных технологий и интерактивных симуляторов. Для повышения качества обучения необходима разработка специализированного программного обеспечения, ориентированного на прецизионную точность воспроизведения рабочих сценариев и материально-технической базы.

Целью работы является разработка симулятора, обеспечивающего формирование базовых практических навыков работы с лабораторным оборудованием для анализа ДНК и РНК за счет имитации лабораторных процедур, интерактивного взаимодействия с объектами и контроля правильности действий пользователя.

Исследование предметной области

Развитие современных биотехнологий и молекулярной генетики требует подготовки высококвалифицированных специалистов, обладающих безупречными навыками работы с лабораторным оборудованием. Анализ ДНК и РНК – это многоэтапный процесс, в котором любая ошибка на стадии подготовки проб или эксплуатации приборов может привести к значительным финансовым потерям из-за порчи дорогостоящих реагентов и риску поломки сложной техники. В последние годы в качестве эффективного решения данной проблемы рассматривается концепция виртуализации лабораторных пространств, позволяющая перенести процесс формирования первичных навыков в цифровую среду.

Научные данные подтверждают, что использование интерактивных симуляторов в техническом обучении способствует более глубокому пониманию архитектуры приборов и алгоритмов их работы. В отличие от традиционного чтения инструкций, данные методы позволяют обучающемуся визуализировать внутренние процессы, происходящие в оборудовании, таком как центрифуги или спектрофотометры. Это не только ускоряет процесс запоминания последовательности действий, но и формирует мышечную память через механику взаимодействия с 3D-объектами. Ключевыми элементами таких систем являются системы контроля ошибок в реальном времени и встроенные методические подсказки, которые минимизируют риск закрепления неправильных навыков [1].

Несмотря на наличие на рынке специализированных образовательных платформ, их массовое внедрение часто ограничено высокой стоимостью подписок, закрытой архитектурой или отсутствием возможности адаптации под конкретный парк приборов отечественных лабораторий. Кроме того, многие существующие решения ориентированы на упрощенную визуализацию в ущерб технической достоверности. В связи с этим была инициирована разработка оригинального симулятора «RoboLab». Проект направлен на создание автономного приложения с прецизионной точностью 3D-моделей и логикой работы, полностью соответствующей регламентам реальных биологических исследований [2].

Методы и технологии

Реализация интерактивного симулятора «RoboLab» представляла собой комплексный процесс, разделенный на три ключевых технологических этапа,

каждый из которых требовал использования специализированного программного обеспечения.

На первом этапе осуществлялось проектирование высокоточных трехмерных моделей лабораторного оборудования (центрифуги, микроспектрофотометра, системы цифровой ПЦР). В качестве основного инструмента использовалась среда SketchUp, позволяющая создавать модели с соблюдением реальных метрических параметров на основе технической документации приборов.

К преимуществам SketchUp можно отнести следующие моменты [3]:

- высокая точность моделирования, SketchUp ориентирован на работу с реальными метрическими величинами. Это позволяет создавать модели лабораторного оборудования, полностью соответствующие техническим паспортам и чертежам приборов;

- механика прямого моделирования значительно ускоряет процесс проектирования корпусных деталей приборов и позволяет оперативно вносить изменения в структуру объектов без необходимости перестраивать всю модель с нуля;

- программная среда обеспечивает сохранение правильной полигональной сетки и масштаба при передаче моделей в Unreal Engine 5, что исключает возникновение визуальных артефактов и ошибок при настройке коллизий;

- поддержка прямого импорта САД-файлов позволяет использовать реальные подложки и схемы расположения узлов оборудования, гарантируя максимальное сходство виртуального тренажера с оригиналом.

В качестве основного инструментария для сборки и визуализации виртуальной среды был выбран Unreal Engine 5. Данная платформа представляет собой мощный программный комплекс, позволяющий создавать высокореалистичные интерактивные приложения с глубокой физической симуляцией объектов. Использование Unreal Engine в образовательных целях обусловлено возможностью интеграции сложных сценариев взаимодействия с хорошей графикой, что создает эффект полного погружения в лабораторный процесс [4].

К преимуществам среды разработки Unreal Engine можно отнести следующие аспекты [5]:

- технология позволяет в реальном времени просчитывать глобальное освещение и отражения, что критически важно для корректного отображения металлических и стеклянных поверхностей лабораторного оборудования без длительного процесса предварительного просчета света;

- поддержка физически корректного шейдинга (PBR) дает возможность настраивать свойства материалов таким образом, чтобы пластик, металл и жидкости выглядели и взаимодействовали со светом так же, как в реальной жизни;

- наличие встроенной системы визуального программирования Blueprints позволяет выстраивать сложную программную логику приборов и сценарии лабораторных работ;

- встроенные компоненты обработки физики позволяют реализовать естественное поведение лабораторных инструментов;

– архитектура платформы позволяет собрать симулятор в виде независимого исполняемого файла, который работает без необходимости подключения к интернету или установки сторонних браузерных плагинов.

Выбор программного обеспечения обусловлен спецификой данного проекта. SketchUp обеспечил высокую точность проектирования 3D-моделей приборов, а Unreal Engine 5 стал основной средой для реализации интерактивности. Благодаря инструментам данной платформы удалось внедрить продвинутые графические эффекты и физику взаимодействия, что приблизило условия виртуальной лаборатории к реальным. Использование системы визуального программирования Blueprints позволило удобно и эффективно настроить внутреннюю логику оборудования, связав действия пользователя с ответными реакциями системы.

Результаты

В результате работы было создано десктоп приложение, в котором реализованы следующие функции:

- пользователю доступна детализированная локация, включающая основные приборы для анализа нуклеиновых кислот: центрифугу 1248R, микроспектрофотометр Nano 500 и ПЦР-комплекс (рис. 1);
- реализована система управления от первого лица, позволяющая пользователю свободно передвигаться по лаборатории и детально осматривать оборудование под любым углом.;
- интегрирован робот-помощник, с помощью диалоговых окон он сопровождает пользователя все время работы в симуляторе (рис. 2);
- программная часть, написанная на Blueprints, отслеживает каждый шаг пользователя, и предоставляет возможность взаимодействия с оборудованием и получение ответной реакция системы на действия пользователя;
- собрана и представлена информационная база об оборудовании, его эксплуатации и технике безопасности;
- тестирование для пользователя по общим вопросам по лабораторной работе (рис. 1);
- создана система подсчета баллов при прохождении симулятора.

На данный момент все функциональные модули симулятора и база данных оборудования интегрированы в программную оболочку приложения. Использование интерактивной среды для отработки навыков позволит обучающемуся упростить взаимодействие с реальными приборами в лаборатории, работа с которыми требует предельной точности и строгого соблюдения регламентов.

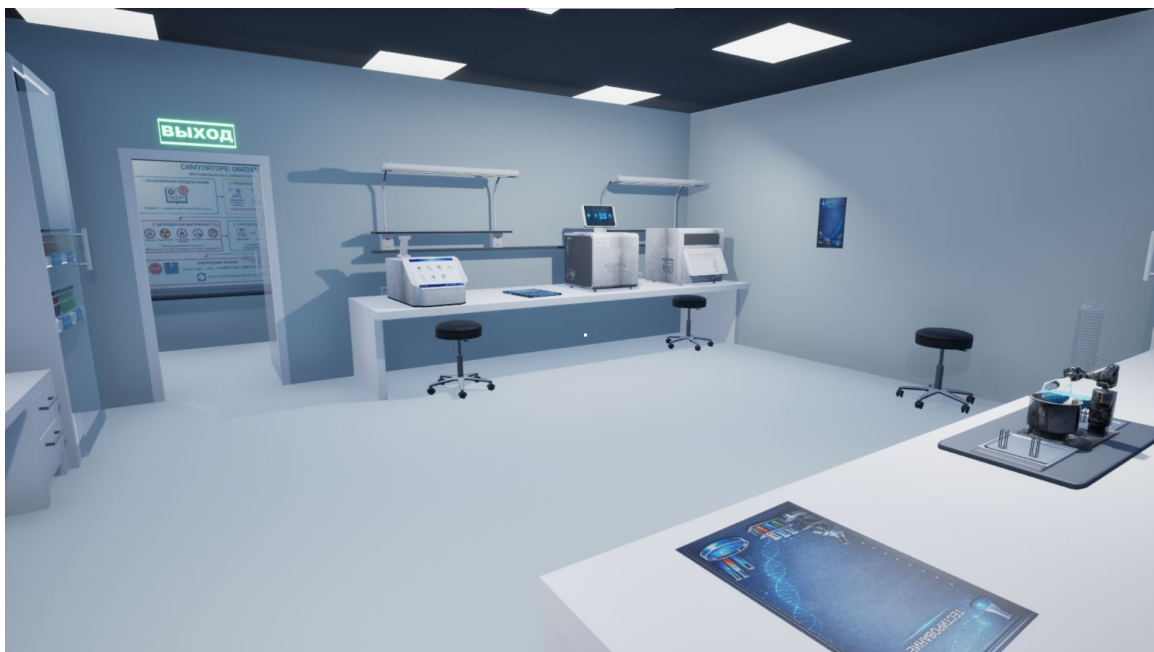


Рис. 1. Внешний вид лаборатории

А для помощи обучающимся, реализован встроенный робот-помощник и система методических подсказок, которые позволят лучше разобраться как с интерфейсом самого симулятора, так и с техническими нюансами эксплуатации сложного оборудования для анализа ДНК и РНК, с которыми студент может столкнуться в реальной профессиональной деятельности.

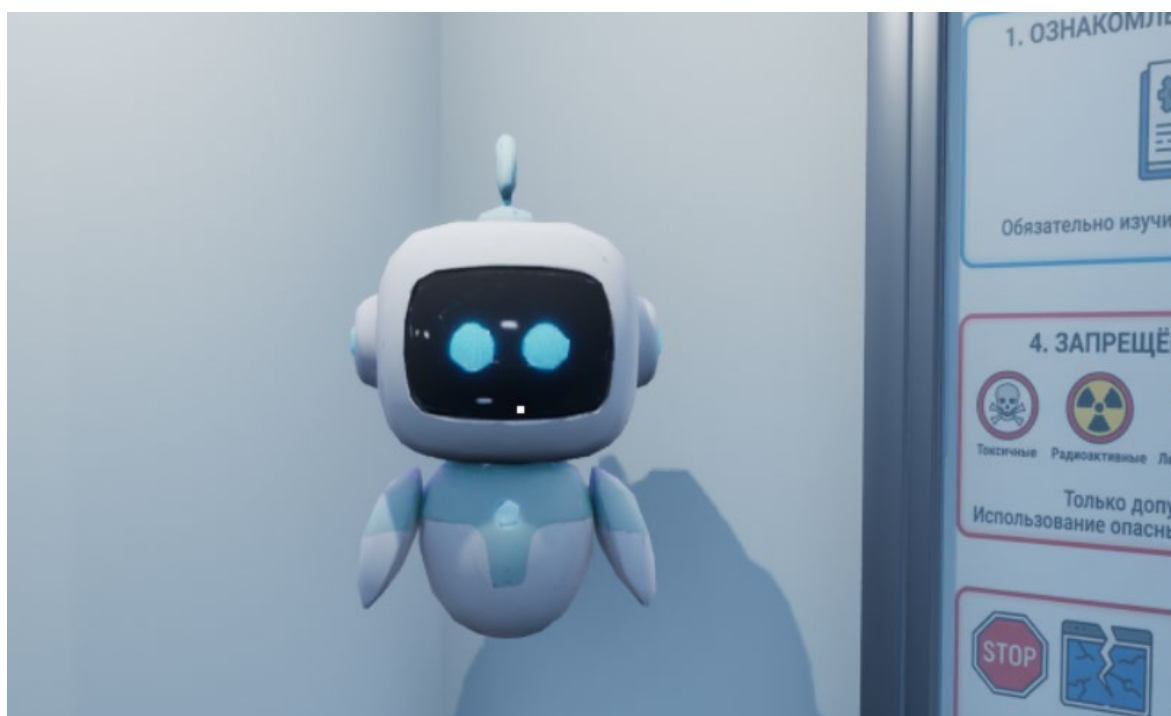


Рис. 2. Робот помощник

Заключение

В результате выполнения работы был разработан интерактивный симулятор «RoboLab», предназначенный для обучения студентов работе с лабораторным оборудованием для анализа ДНК и РНК. Симулятор может быть интегрирован в учебный процесс вузов как средство подготовки к реальным лабораторным работам, что в конечном итоге повышает уровень профессиональных компетенций будущих выпускников и снижает эксплуатационные риски для образовательных учреждений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Технические аспекты разработки виртуальных лабораторных работ по техническим дисциплинам / О. Г. Аврунин, О. Я. Крук, Т. В. Носова, В. В. Семенец. – Текст : электронный // КиберЛенинка : [сайт]. – URL: <https://cyberleninka.ru/article/n/tehnicheskie-aspekty-razrabotki-virtualnyh-laboratornyh-rabot-po-tehnicheskim-distiplinam> (дата обращения: 13.05.2026).
2. Степушкина, П. К. Игровые движки для образовательных целей / П. К. Степушкина. – Текст : электронный // КиберЛенинка : [сайт]. – URL: <https://cyberleninka.ru/article/n/igrovyedvizhki-dlya-obrazovatelnyh-tseley> (дата обращения: 13.05.2026).
3. Лапицкая, Д. П. Сравнительный анализ программ для инженерной графики / Д. П. Лапицкая, В. А. Бутайкис, А. А. Куликова. – Текст : электронный // Материалы докладов 81-й студенческой научно-технической конференции БНТУ. – Минск : БНТУ, 2025. – С. 273–274. – URL: <https://rep.bntu.by/handle/data/162890> (дата обращения: 13.05.2026).
4. Эффективность виртуальных технологий для формирования навыков обследования пациента и профессионального мышления врача (результаты проспективных, рандомизированных, контролируемых исследований) / Е. Г. Рипп, Т. М. Рипп, Ю. С. Духовный, А. С. Зверев. – Текст : электронный // РОСОМЕД-2025 : сборник тезисов. – 2025. – URL: <https://www.rosmedobr.ru/rosmedobr2025/thesis/390301/> (дата обращения: 13.05.2026).
5. Сравнительный анализ Unity и Unreal Engine в разработке симуляторов / Б. Гылыджов, Г. Дерьяева, Г. Гараева, М. Гайыпназаров. – Текст : непосредственный // Символ науки : международный научный журнал. – 2024. – Т. 1, № 10-1. – С. 49–50.

© В. З. Бальжинимаева, Д. Ю. Смирнов, 2026

Е. А. Басыня^{1✉}, В. Ю. Сапегин¹

Метод состязательного обучения моделей обнаружения виртуальных защищенных каналов связи на основе адаптивной генерации конфигураций Xray

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: director@nii-ikt.ru

Аннотация. В работе представлен метод состязательного обучения моделей обнаружения виртуальных защищенных каналов связи, в которой генератор конфигураций на базе Xray адаптивно перебирает протокольные семейства и параметры маскировки, а бинарный детектор виртуальных защищенных каналов связи последовательно дообучается на трудных положительных примерах. Актуальность исследования определяется тем, что статические детекторы, обученные на ограниченном перечне туннелей, теряют устойчивость при появлении новых сочетаний транспорта, рукопожатия и параметров обфускации. Метод задает ограниченное, но репрезентативное пространство конфигураций Xray, правила валидации работоспособности кандидатных конфигов, порядок сбора трафика и цикл пополнения обучающей выборки трудными примерами. В статье описаны требования к экспериментальному протоколу, показателям качества и критериям остановки состязательного цикла. Практическая значимость работы состоит в переходе от разовой оценки детектора в статическом домене к воспроизводимой процедуре формирования более устойчивых моделей обнаружения виртуальных защищенных каналов связи.

Ключевые слова: виртуальные защищенные каналы связи, VPN, обнаружение VPN, Xray, состязательное обучение, анализ зашифрованного трафика

E. A. Basinya^{1✉}, V. Y. Sapegin¹

Method for adversarial training of virtual secure communication channel detection models based on adaptive Xray configuration generation

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: director@nii-ikt.ru

Abstract. The paper presents a methodology for adversarial training of virtual secure communication channel detection models. In the proposed setup, an Xray-based configuration generator adaptively enumerates protocol families and obfuscation parameters, while a binary VSCC detector is iteratively re-trained on hard positive examples. The study addresses the practical instability of static detectors trained on a narrow set of tunnel configurations. The methodology defines a constrained yet representative Xray configuration space, validity checks for generated candidates, traffic collection rules, and an iterative hard-example mining loop. The paper also specifies the experimental protocol, quality indicators, and stopping criteria for the adversarial cycle. The proposed approach is intended as a reproducible basis for building more robust virtual secure communication channel detection models.

Keywords: virtual private communication channels, VPN, VPN detection, Xray, adversarial training, encrypted traffic analysis

Введение

В условиях роста доли шифрованного сетевого обмена задача обнаружения виртуальных защищенных каналов связи приобретает для корпоративных и ведомственных сетей прикладное значение. Виртуальные защищенные каналы связи (ВЗКС) используются не только как средство легитимной защиты конфиденциальности, но и как инструмент обхода периметровой фильтрации, сокрытия командно-управляющего трафика и маскировки каналов вывода данных. По данным F.A.C.S.T. [1], использование технологий виртуальных защищенных каналов связи продолжало фиксироваться в сценариях целенаправленных атак на российские организации, а развитие нормативного регулирования мониторинга таких соединений дополнительно подчеркивается Приказом Роскомнадзора № 168 [2].

Современные исследования показывают, что даже полностью шифрованные и обфусцированные туннели сохраняют наблюдаемые статистические, транспортные и рукопожатийные особенности. В работе M. Wu и соавторов показано, что пассивная детекция *fully encrypted traffic* возможна даже для семейств, включающих VMess [3]. Хие и соавторы продемонстрировали практическую детектируемость OpenVPN и его обфусцированных конфигураций на реальном операторском трафике [4], а затем показали, что обфусцированные прокси и функционально эквивалентные ВЗКС туннели остаются уязвимыми к отпечаткам инкапсулированного TLS-рукопожатия [5]. Вместе с тем Хие и соавторы установили, что качество *encrypted traffic classification* существенно деградирует при смене сетевой среды и транспортных условий, даже если исходная модель демонстрирует высокий результат в лабораторном домене [6].

Проблема заключается в том, что большинство практических детекторов обучается на статическом перечне протоколов и фиксированных конфигураций. Такой подход создает системную асимметрию: защитная сторона оценивает модель на конечном наборе известных профилей, тогда как противоположная сторона может адаптивно варьировать транспорт, параметры рукопожатия и маскировки. В работах по *traffic analysis* уже показана уязвимость глубоких моделей к состязательным возмущениям [7], а также актуальность сертифицируемой робастности в мультимодальной постановке [8]. Обзор Guerra-Manzanares и соавторов дополнительно фиксирует слабую сопоставимость опубликованных результатов из-за зависимости от устаревших датасетов и неоднородных экспериментальных схем [9]. Дополнительно в работе Liu и соавторов продемонстрирована возможность атаки обхода системы обнаружения вредоносного трафика при доступе только к итоговой метке классификации, что подчеркивает практическую значимость оценки устойчивости детекторов в условиях ограниченной информации об атакуемой модели [10]. В отечественной литературе терминологические и прикладные основы ВЗКС, включая вопросы адаптивного управления защищенной коммуникационной инфраструктурой, последовательно рассматриваются в работах Е. А. Басыни [11–13].

Постановка задачи

Целью работы является разработка метода состязательного обучения бинарного детектора ВЗКС и легитимных сетевых потоков, в которой качество модели повышается не за счет однократного расширения датасета, а за счет управляемого цикла генерации новых конфигураций ВЗКС и последующего дообучения на наиболее трудных положительных примерах. Рассматриваются два агента. Первый агент представляет генератор конфигураций на базе Xray, второй - обычную модель бинарной детекции ВЗКС по потоковым или сеансовым признакам. На каждом шаге генератор формирует набор кандидатных конфигураций, стремясь получить работоспособные туннели, снижающие качество текущего детектора. Детектор, в свою очередь, после анализа нового трафика обновляет параметры и повышает устойчивость к найденным вариантам маскировки.

Принципиально важно, что генератор в данном методе не максимизирует абстрактное конфигурационное разнообразие. Его целевая функция должна учитывать четыре ограничения: синтаксическую корректность конфигурации, успешное установление соединения, сохранение полезной сетевой функциональности и реалистичность получаемого трафикового профиля. Иными словами, в обучающую петлю допускаются только те конфигурации, которые могут быть развернуты в лабораторном стенде без ручной доработки и генерируют наблюдаемый ВЗКС-трафик, а не формально корректный, но неработоспособный JSON. Такое требование исключает загрязнение обучающей выборки артефактами генератора и удерживает постановку задачи в прикладной плоскости.

Пространство адаптивного перебора ограничивается умеренным, но репрезентативным набором конфигураций Xray, сформированным по актуальной документации Project X [14]. В генератор включаются протокольные семейства VLESS и VMess, режимы транспортной защиты TLS и REALITY, транспортные механизмы WebSocket, gRPC, mKCP и XHTTP, а также параметры mux, padding, ALPN, SNI, path, serviceName и схемы маршрутизации. Такое ограничение позволяет, с одной стороны, не расплзтись до практически необозримого пространства вариантов, а с другой стороны, отразить типовые направления эволюции скрытых Xray-конфигураций, реально влияющие на трафиковый отпечаток.

Архитектура состязательного обучения

Генератор конфигураций целесообразно реализовывать как ограниченный поисковый или reinforcement learning-модуль, работающий поверх библиотеки допустимых трансформаций. Входом для него являются история успешных уклонений, уверенности текущего детектора, а также таблица разрешенных операций изменения конфигурации. На одном шаге генератор может сменить протокольное семейство, транспорт, режим защиты, параметры маскировки, поля заголовков или режим мультиплексирования; на следующем шаге - скорректировать маршрут и дополнительные сервисные параметры. Награда генератора должна возрастать при успешном обходе детектора, но уменьшаться при росте задержек, снижении стабильности соединения и появлении явно искусственных

артефактов трафика. Тем самым генератор учится искать не любой обход, а обход, совместимый с реальной эксплуатацией ВЗКС.

Детектор ВЗКС в методике намеренно рассматривается как архитектурно нейтральный компонент. Это может быть градиентный бустинг на статистических flow-признаках, сверточная или рекуррентная сеть на последовательностях длин и межпакетных интервалов либо трансформерная модель на кодированном представлении сессии. Ключевым требованием является не выбор конкретной нейросетевой формулы, а наличие единого признакового пространства, отражающего длины пакетов, направления, темпоральную структуру обмена, наблюдаемые handshake-атрибуты и время до вынесения решения. Дополнительно следует исключать признаки, жестко привязанные к одной версии протокола или единичной конфигурационной особенности, поскольку именно такие признаки быстрее всего разрушаются при адаптивном переборе Xray-настроек.

Состязательный цикл строится следующим образом. На начальном шаге формируется базовый корпус доброкачественного трафика и трафика ВЗКС, полученного от ограниченного числа заведомо рабочих конфигураций. После первичного обучения детектора генератор начинает выпускать новые кандидатные конфигурации, которые автоматически проверяются на разворачиваемость и успешное прохождение контрольного сценария. Далее для каждой принятой конфигурации записывается трафик, выполняется вывод модели и выделяются трудные положительные примеры, то есть такие сессии ВЗКС, которые детектор ошибочно отнес к легитимным сетевым потокам либо распознал только после недопустимо длинного окна наблюдения. Эти примеры вместе с сопоставимыми по фону доброкачественными сессиями добавляются в обучающую выборку, после чего генератор и детектор переходят к следующему витку.

Такой цикл не следует останавливать после первого же снижения ошибки. Корректнее завершать обучение при выполнении одновременно двух условий: во-первых, доля новых валидных конфигурационных семейств, которые еще обходят детектор, становится малой; во-вторых, очередное дообучение не дает заметного прироста по устойчивости на отложенном наборе ранее не встречавшихся комбинаций протокола, транспорта и параметров маскировки. Именно наличие отложенного пространства неизвестных конфигураций отличает метод от простого hard example mining на одном и том же семействе туннелей.

Экспериментальный протокол

Для практического применения метода лабораторный стенд должен рассматриваться как генератор контролируемых доменов наблюдения. В каждом эксперименте фиксируются клиентская платформа, тип сети, география серверной точки, момент времени, тип фоновой нагрузки и точка съема трафика. Минимально целесообразно варьировать настольные и мобильные клиенты, проводной и беспроводной доступ, а также несколько маршрутных сценариев до серверов в разных юрисдикциях. При этом задача остается бинарной: определяется наличие или отсутствие ВЗКС, а не конкретное приложение внутри туннеля.

Сбор данных необходимо организовывать так, чтобы исключить утечки между обучением и тестированием. Разбиение должно выполняться не только по отдельным сессиям, но и по временным срезам, платформам и целым семействам конфигураций. Если одна и та же комбинация, например VLESS-REALITY-ХНТТР с конкретным шаблоном padding, встречается и в обучении, и в тесте, то измеряется запоминание, а не переносимость. Поэтому часть конфигураций и их мутаций должна полностью удерживаться в неизвестном тестовом пуле, используемом только для итоговой оценки устойчивости.

Оценка качества детектора должна включать не только стандартные метрики классификации, такие как F1 и AUPRC, но и показатели, отражающие эксплуатационную пригодность. К ним относятся уровень ложных срабатываний на обычном TLS-трафике, время или объем наблюдения до вынесения решения, а также устойчивость к ранее не встречавшимся конфигурациям. Со стороны генератора необходимо отдельно фиксировать долю синтаксически валидных конфигов, долю реально устанавливающих соединение конфигов и долю конфигов, которые на очередной итерации обходят детектор. Такое двустороннее измерение позволяет судить не только о качестве защитной модели, но и о том, насколько быстро исчерпывается полезный потенциал адаптивного перебора.

Важным элементом предлагаемого метода является единообразие вычислительного бюджета. Все сравниваемые версии детектора должны получать одинаковое окно наблюдения, одинаковую процедуру предобработки и одинаковый набор фоновых легитимных сценариев. Аналогично, генератор на каждой итерации должен работать в фиксированном лимите числа попыток и числа допустимых модификаций одной конфигурации. Иначе выигрыш может быть обусловлен не улучшением самой модели, а банальным увеличением переборного ресурса.

Ограничения и практические риски

Главный риск состязательной схемы состоит в переобучении на узкий, пусть и динамически обновляемый, набор Xray-специфичных артефактов. Если генератор регулярно находит обходы за счет одной и той же группы признаков, детектор может стать излишне чувствительным именно к ним и ухудшить переносимость на другие семейства ВЗКС. Для предотвращения этого требуется квотирование протокольных и транспортных подпространств, контроль баланса между положительным и отрицательным классом, а также отдельная проверка на обычном HTTPS-, HTTP/2- и HTTP/3-трафике. Не менее существенен риск загрязнения датасета невалидными конфигурациями, поэтому этап автоматической валидации должен предшествовать любой записи трафика.

Второй класс ограничений связан со стоимостью эксперимента и изменчивостью внешней среды. С обновлением Xray, сменой реализаций транспорта или изменением поведения промежуточных сетевых узлов трафиковый профиль одной и той же конфигурации может заметно измениться. Следовательно, получаемая модель не должна рассматриваться как окончательно устойчивая; она представляет собой состояние детектора, достигнутое при фиксированной версии

инструментов и контролируемом наборе доменов. Рассматриваемый метод предполагает регулярное повторение цикла на новых временных срезах и обязательное версионирование всех конфигурационных шаблонов и окружений. Эксперименты при этом должны проводиться только в изолированном стенде, без вывода состязательно сгенерированных конфигураций в эксплуатационную сеть.

Заключение

Предложенный метод переводит задачу обнаружения ВЗКС из статической плоскости в постановку управляемого состязательного обучения. Научная новизна подхода состоит в совместном рассмотрении двух процессов: адаптивной генерации реалистичных Хгау-конфигураций и последовательного дообучения бинарного детектора на трудных положительных примерах. В отличие от традиционного расширения датасета, здесь новая обучающая информация формируется как результат направленного поиска конфигураций, максимально неудобных для текущей версии модели.

Практическая значимость метода заключается в формировании воспроизводимого протокола разработки более устойчивых детекторов ВЗКС. Такой протокол позволяет сравнивать модели не только по качеству в базовом домене, но и по устойчивости к ранее не встречавшимся комбинациям протокола, транспорта и маскировки. В дальнейшем разработанный метод может служить основой для полноценных экспериментальных исследований, в которых будут сопоставляться различные архитектуры детекторов и различные стратегии адаптивного генератора конфигураций.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Universal Encryptor: The number of cyberattacks in Russia increased by 140% in 2023 // F.A.C.C.T. URL: <https://www.facct.ru/media-center/press-releases/cyberattacks-2023/> (accessed: 20.04.2026).

2. Приказ Федеральной службы по надзору в сфере связи, информационных технологий и массовых коммуникаций «О внесении изменений в Критерии оценки материалов и (или) информации, необходимых для принятия Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций решений, являющихся основаниями для включения доменных имен и (или) указателей страниц сайтов в информационно-телекоммуникационной сети «Интернет», а также сетевых адресов, позволяющих идентифицировать сайты в информационно-телекоммуникационной сети «Интернет», в единую автоматизированную информационную систему «Единый реестр доменных имен, указателей страниц сайтов в информационно-телекоммуникационной сети «Интернет» и сетевых адресов, позволяющих идентифицировать сайты в информационно-телекоммуникационной сети «Интернет», содержащие информацию, распространение которой в Российской Федерации запрещено», утвержденные приказом Федеральной службы по надзору в сфере связи, информационных технологий и массовых коммуникаций от 27 февраля 2023 г. № 25» от 08.11.2023 № 168 // Собрание законодательства Российской Федерации. 01.12.2023.

3. Wu M., Sippe J., Sivakumar D. [et al.]. How the Great Firewall of China Detects and Blocks Fully Encrypted Traffic // 32nd USENIX Security Symposium (USENIX Security 23). 2023. P. 2653-2670.

4. Xue D., Ramesh R., Jain A. [et al.]. OpenVPN Is Open to VPN Fingerprinting // 31st USENIX Security Symposium (USENIX Security 22). 2022. P. 483-500.

5. Xue D., Kallitsis M., Houmansadr A., Ensafi R. Fingerprinting Obfuscated Proxy Traffic with Encapsulated TLS Handshakes // 33rd USENIX Security Symposium (USENIX Security 24). 2024. P. 2689-2706.
6. Xie R., Cao J., Dong E. [et al.]. Rosetta: Enabling Robust TLS Encrypted Traffic Classification in Diverse Network Environments with TCP-Aware Traffic Augmentation // 32nd USENIX Security Symposium (USENIX Security 23). 2023. P. 625-642.
7. Nasr M., Bahramali A., Houmansadr A. Defeating DNN-Based Traffic Analysis Systems in Real-Time With Blind Adversarial Perturbations // 30th USENIX Security Symposium (USENIX Security 21). 2021. P. 2705-2722.
8. Yan J., Liu Z., Xie Y. [et al.]. CertTA: Certified Robustness Made Practical for Learning-Based Traffic Analysis // 34th USENIX Security Symposium (USENIX Security 25). 2025.
9. Guerra-Manzanares A., Caprolu M., Di Pietro R. A Comprehensive Review on Machine Learning-Based VPN Detection: Scenarios, Methods, and Open Challenges // Computer Science Review. 2025. Vol. 58. Article 100781. DOI: 10.1016/j.cosrev.2025.100781.
10. Liu Z., Zhao Y., Liu Z. [et al.]. A Hard-Label Black-Box Evasion Attack against ML-based Malicious Traffic Detection Systems // Network and Distributed System Security Symposium (NDSS). 2026. DOI: 10.14722/ndss.2026.230916.
11. Басыня Е. А. Система самоорганизующегося виртуального защищенного канала связи // Защита информации. Инсайд. 2018. № 5 (83). С. 10-15.
12. System of a Self-Organizing Virtual Secure Communication Channel Based on Stochastic Multi-Layer Encryption and Overlay Technologies / E. A. Basynya, Z. B. Akhayeva, D. H. Omarkhanova, G. B. Tolegenova [et al.] // Journal of Theoretical and Applied Information Technology. 2022. Vol. 100, iss. 16. P. 4918-4927.
13. Басыня Е. А. Программная реализация и исследование системы интеллектуально-адаптивного управления информационной инфраструктурой предприятия // Вестник Самарского государственного технического университета. Серия: Технические науки. 2020. Т. 28, № 1 (65). С. 6-21.
14. Xray Configuration Reference // Project X. URL: <https://xtls.github.io/en/config/> (accessed: 20.04.2026).

© Е. А. Басыня, В. Ю. Сатегин, 2026

Е. А. Басыня¹✉, В. Ю. Сапегин¹

Методика сравнительного анализа методов обнаружения VPN-трафика с учетом сдвига домена и дрейфа данных

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: director@nii-ikt.ru

Аннотация. В работе представлена методика сравнительного анализа методов бинарного обнаружения VPN/non-VPN-трафика в условиях маскировки, изменения сетевой среды и дрейфа данных. Актуальность исследования определяется практической необходимостью сопоставления существующих решений не только по качеству классификации в лабораторном домене, но и по устойчивости при переходе к эксплуатационным сценариям. Методика основана на доменной постановке задачи, в которой условия наблюдения описываются совокупностью факторов: временным срезом, платформой клиента, типом сети, географией маршрута, режимом маскировки и используемым VPN-протоколом. В качестве базовых показателей используются F1 и AUPRC, а для сравнительного анализа дополнительно вводятся индекс переносимости, интегральная устойчивость к маскировке, показатель минимального качества на множестве целевых доменов и чувствительность метода к domain shift. Предлагаемая методика определяет единые правила формирования benchmark-сценариев, вычисления показателей и итогового сопоставления существующих методов обнаружения VPN, что позволяет использовать ее как воспроизводимую основу для последующих экспериментальных исследований.

Ключевые слова: VPN, обнаружение VPN, зашифрованный трафик, обфускация трафика, методика сравнения, информационная безопасность

Е. А. Басыня¹✉, V. Y. Sapegin¹

A methodology for comparative analysis of VPN traffic detection methods under domain shift and data drift

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: director@nii-ikt.ru

Abstract. This paper presents a methodology for comparative analysis of binary VPN/non-VPN traffic detection methods under obfuscation, changing network conditions and data drift. The study is motivated by the need to compare existing solutions not only by their classification quality in a laboratory domain, but also by their robustness when transferred to operational scenarios. The methodology is based on a domain-oriented problem formulation in which observation conditions are described by a set of factors including time slice, client platform, network type, route geography, obfuscation mode and VPN protocol. F1 and AUPRC are used as the main quality indicators, while transferability index, integrated robustness to obfuscation, minimum quality over target domains and sensitivity to domain shift are introduced as comparative criteria. The proposed methodology defines unified rules for constructing benchmark scenarios, computing indicators and performing the final comparison of VPN detection methods, thus providing a reproducible basis for further experimental studies.

Keywords: VPN, VPN detection, encrypted traffic, obfuscation, comparative benchmark, information security

Введение

В условиях цифровой трансформации выявление скрытых виртуальных защищенных каналов связи становится значимой задачей обеспечения информационной безопасности корпоративных сетей. Технологии сокрытия сетевой активности, изначально ориентированные на защиту конфиденциальности, могут использоваться для обхода фильтрации, организации скрытых каналов передачи данных и маскировки вредоносной активности. По данным F.A.C.C.T. [1], в 2023 году сохранялась тенденция применения VPN-технологий при целевых атаках на корпоративные сети Российской Федерации. Современные исследования предлагают различные подходы к решению данной задачи: Wu Н. и соавторы [2, 3] показывают эффективность нейросетевой классификации при детекции сложных протоколов, включая V2Ray и Shadowsocks, Канатьев К. Н. и соавторы [4, 5] отмечают ограничения существующих подходов и необходимость новых аналитических методов, а работы научной школы под руководством Е. А. Басыни рассматривают механизмы обхода фильтрации и методы его детекции на примере оверлейных сетей Tor и I2P [6–8].

Последние зарубежные исследования показывают, что пассивное обнаружение VPN-трафика в корпоративных и операторских сетях остается сложной задачей, поскольку современные VPN активно используют шифрование, обфускацию и многоуровневую инкапсуляцию. Вследствие этого сигнатурный анализ и отдельные статистические признаки уже не отражают реальную эксплуатационную устойчивость методов. Akbari и соавторы указывают на зависимость классификации зашифрованного трафика от «ленивых» TLS-признаков [9], Хуе и соавторы демонстрируют детектируемость OpenVPN на реальном провайдерском трафике [10] и уязвимость обфусцированных проху/VPN-подобных туннелей к отпечаткам инкапсулированного TLS [11]. Отдельное значение имеют исследования устойчивости методов при изменении условий эксплуатации: Malekghaini и соавторы показывают снижение качества моделей глубокого обучения при временном дрейфе данных [12], Lv и соавторы рассматривают one-class постановку и устойчивость к неизвестным типам трафика [13], а Hanlon и соавторы демонстрируют возможности protocol-agnostic эвристики на основе поведения инкапсулированного TCP [14].

Таким образом, научная проблема состоит не только в разработке способов обнаружения виртуальных защищенных каналов связи, но и в формировании методики их корректного сравнительного анализа в условиях маскировки трафика. Обзоры Razzoqi, Pekar и Guerra-Manzanares [15, 16] фиксируют слабую сопоставимость опубликованных результатов из-за устаревших лабораторных датасетов, недостаточного учета сдвигов данных и отсутствия общей benchmark-рамки; эти выводы согласуются с методологическими рекомендациями Arp и соавторов для ML-систем в задачах кибербезопасности [17]. Дополнительную актуальность задаче придает нормативное регулирование: Приказ Роскомнадзора № 168 от 08.11.2023 [18], предусматривающий создание Единого реестра VPN- и прокси-

сервисов, подчеркивает прикладную значимость эффективных и воспроизводимых методов мониторинга VPN-соединений в корпоративных сетях.

Постановка задачи

Целью настоящей работы является разработка методики сравнительного анализа существующих методов бинарного обнаружения VPN/non-VPN-трафика на семействе эксплуатационных доменов. Методика должна обеспечивать воспроизводимое сопоставление как моделей машинного обучения, так и эвристических либо протоколно-ориентированных методов по интегральным и эксплуатационным показателям качества.

Пусть X – пространство наблюдаемых фрагментов сетевого трафика. Элемент $x \in X$ может представлять собой поток, сессию или окно наблюдения, сформированное по доступным наблюдателю признакам: последовательностям длин пакетов, временным интервалам, признакам рукопожатия, метаданным транспортного и прикладного уровней. Каждому объекту x ставится в соответствие бинарная метка y , характеризующая наличие или отсутствие VPN.

$$Y \in \{0, 1\}, \quad 0 = \text{nonVPN}, \quad 1 = \text{VPN} \quad (1)$$

$$\xi = (\tau, p, n, g, o, v) \in \Xi \quad (2)$$

Вектор ξ задает условия эксплуатации метода: τ – временной срез наблюдения, p – платформу или VPN-клиент, n – тип сети, g – географию маршрута, o – режим маскировки, v – протокол или реализацию VPN.

$$D_\xi = P_\xi(X, Y), \quad \text{hm}: X \rightarrow \hat{Y}, \quad \hat{Y} \in \{0, 1\} \quad (3)$$

Каждому вектору ξ соответствует собственный домен эксплуатации D_ξ . Сравнимый метод m рассматривается как отображение hm , преобразующее наблюдаемый объект x в решение $\hat{y}$. Если метод использует пороговый score, то решение может быть записано в виде $\hat{y} = 1 [s_m(x) \geq \theta_m]$.

Выделяется базовый домен ξ_0 , соответствующий in-domain сценарию, и множество целевых доменов Ξ_{tar} , включающее temporal shift, cross-platform shift, cross-protocol shift, obfuscation shift и изменение сетевого маршрута. Задача сравнительного анализа состоит в получении воспроизводимой оценки поведения метода m на ξ_0 и на множестве Ξ_{tar} .

Методика сравнительного анализа

Методика сравнительного анализа строится как последовательность взаимосвязанных этапов: формирование семейства доменов эксплуатации, подготовка размеченных данных, выполнение сравниваемых методов в базовом и целевых доменах, анализ устойчивости к маскировке и сдвигу распределений, а также итоговое синтезирование профиля метода. На первом этапе фиксируются сравниваемые методы, тип доступных входных данных и допустимые условия

их применения: для rule-based и DPI-ориентированных подходов задаются анализируемые поля протокольного стека и доступность служебной информации, а для ML-подходов — схема формирования признаков, объем наблюдаемого окна, правила разбиения данных и требования к вычислительным ресурсам. Общая схема методики представлена на рис. 1.

На втором этапе формируется набор доменов эксплуатации: базовый домен ξ_0 используется для получения in-domain оценки, а множество Ξ_{tar} задает целевые сценарии, на которых измеряется переносимость метода. В него включаются изменения платформы клиента, маршрута, сетевой среды, протокола и режима маскировки, что позволяет сравнивать методы не по одному значению качества, а по поведению на семействе контролируемых сценариев. В качестве основного интегрального показателя принимается F1-мера, чувствительная к дисбалансу классов и учитывающая полноту и точность; AUPRC используется для оценки устойчивости при изменении порога решения, а Ассигасу приводится только как вспомогательная характеристика и не применяется как главный критерий ранжирования.

$$Q(m, \xi) = F1(m; D_\xi) \quad (4)$$

Для интерпретации результатов важна не только абсолютная величина $Q(m, \xi)$, но и ее изменение относительно базового домена. Поэтому в методике вводится относительная деградация качества и индекс переносимости. Первый показатель отражает локальное снижение качества при переходе к конкретному домену, а второй показывает, насколько хорошо метод в среднем сохраняет свойства на множестве эксплуатационных сценариев.

$$G(m, \xi) = 1 - \frac{Q(m, \xi)}{(Q(m, \xi_0) + \varepsilon)}, \quad (5)$$

$$T(m) = \frac{1}{|\Xi_{tar}|} \sum_{\xi \in \Xi_{tar}} \frac{Q(m, \xi)}{(Q(m, \xi_0) + \varepsilon)} \quad (6)$$

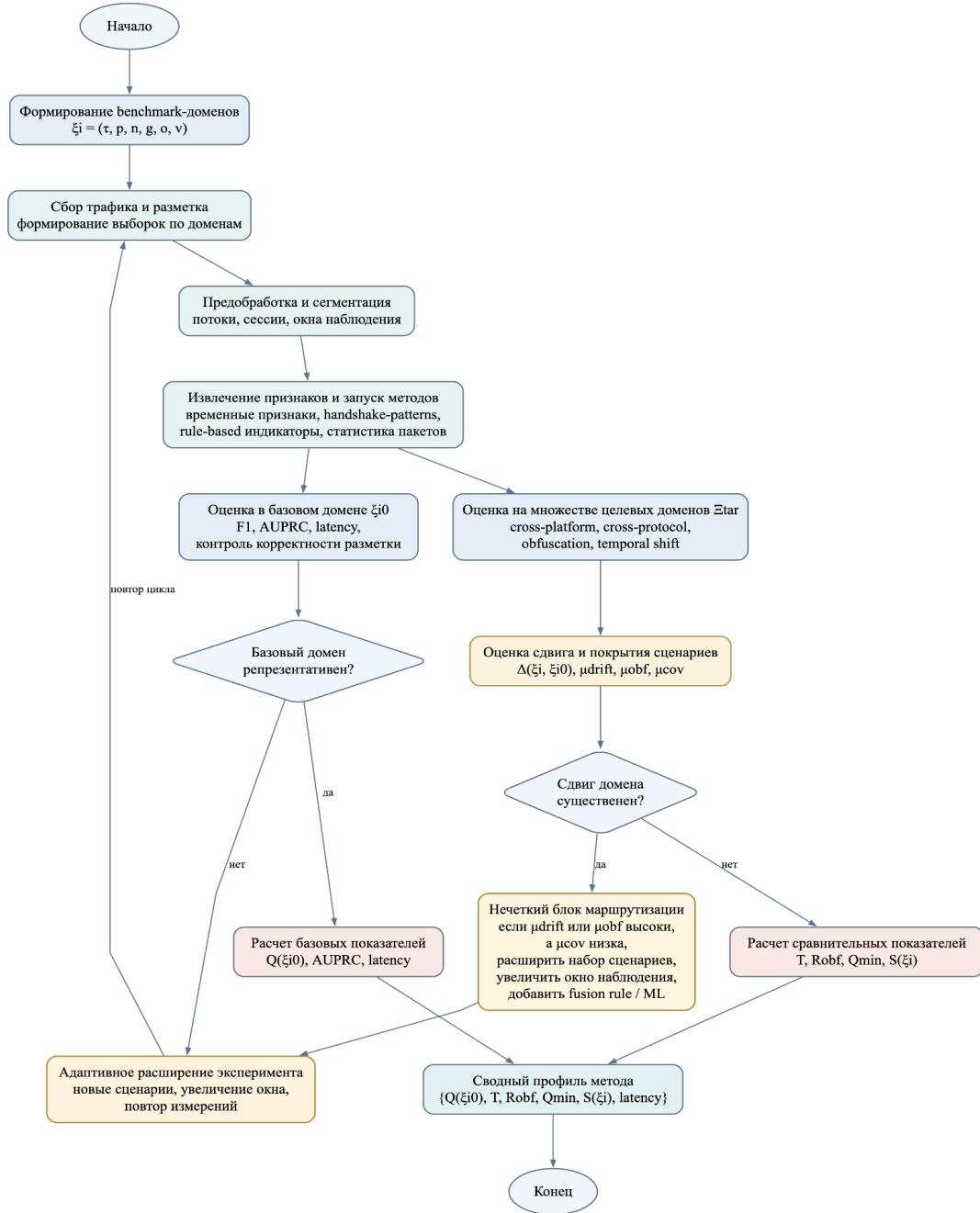


Рис. 1. Блок-схема методики оценки методов обнаружения VPN с адаптивной нечеткой маршрутизацией

Отдельно оценивается устойчивость к маскировке. Для этого из множества целевых доменов выделяются сценарии, в которых изменяется только режим обфускации, а остальные условия эксперимента удерживаются максимально близкими к базовому. Интегральная устойчивость к маскировке задается усреднением по набору таких сценариев с учетом их относительной значимости.

$$Robf(m) = \sum_{k=1}^K w_k \frac{Q(m, \xi_{obf, k})}{(Q(m, \xi_0) + \varepsilon)}, \quad \sum_{k=1}^K w_k = 1 \quad (7)$$

Для анализа причин деградации вводится оценка сдвига между базовым и целевым доменами. Сдвиг вычисляется на едином признаковом представлении $\varphi(X)$, выбранном одинаково для всех сравниваемых методов. В зависимости от природы данных в качестве меры расстояния между распределениями могут использоваться Jensen-Shannon divergence, MMD или иные статистические критерии. Чувствительность метода к drift определяется как отношение относительной деградации качества к величине измеренного сдвига.

$$\Delta(\xi, \xi_0) = D(P_{\xi}(\varphi(X)), P_{\xi_0}(\varphi(X))), \quad S(m, \xi) = \frac{G(m, \xi)}{(\Delta(\xi, \xi_0) + \varepsilon)} \quad (8)$$

Наряду с качеством классификации в методике фиксируется вычислительная цена. Она включает время обработки, требуемый объем наблюдения и устойчивость результата при ограниченном окне данных. Такие характеристики особенно важны для сравнения легковесных эвристических подходов и более тяжелых ML-моделей. Итогом эксперимента является профиль метода, который объединяет in-domain качество, переносимость, устойчивость к маскировке, чувствительность к drift, worst-case результат и вычислительную цену.

Для корректного применения методики benchmark-матрица должна включать следующие оси сравнения:

- 1) In-domain – оценка метода в базовом домене ξ_0 ;
- 2) Temporal shift – переносимость между временными срезами наблюдения.
- 3) Cross-platform shift – изменение клиента, платформы или операционной системы;
- 4) Cross-protocol shift – переход между VPN-протоколами и реализациями;
- 5) Obfuscation shift – устойчивость к различным режимам маскировки;
- 6) Worst-case performance – минимальное качество на множестве целевых доменов.

При практическом применении методики необходимо соблюдать несколько базовых требований. Все сравниваемые решения должны оцениваться строго в бинарной постановке VPN/non-VPN; задачи определения конкретного протокола, приложения внутри туннеля или типа сервиса не должны смешиваться с основной задачей обнаружения.

In-domain, temporal shift, cross-platform shift, cross-protocol shift и obfuscation shift необходимо измерять раздельно. Усреднение этих сценариев без явной фиксации доменных факторов делает результаты методологически неинтерпретируемыми. Для всех методов должна использоваться единая процедура оценки величины $\Delta(\xi, \xi_0)$, иначе чувствительность к drift перестает быть сопоставимой.

Для всех методов также должны использоваться единые правила формирования обучающей, валидационной и тестовой выборок, чтобы различия в результатах определялись свойствами метода, а не особенностями экспериментального протокола. В отчетах об экспериментах необходимо фиксировать значения всех координат $\xi = (\tau, p, n, g, o, v)$, используемое признаковое пространство, правила предобработки данных и вычислительные ограничения.

Итоговое сравнение методов должно проводиться не по одному агрегированному числу, а по профилю, включающему in-domain качество, индекс переносимости, устойчивость к маскировке, показатель worst-case качества, чувствительность к drift и вычислительную цену. На практике это означает следующую процедуру: сначала для каждого метода вычисляются показатели в базовом домене ξ_0 , затем анализируется изменение качества на множестве Ξ_{tar} , после чего выполняется сопоставление методов по совокупности критериев. Предпочтение должно отдаваться тем методам, которые обеспечивают приемлемое базовое качество и одновременно сохраняют устойчивость при смене домена, а не только демонстрируют высокий результат в одном лабораторном сценарии.

Заключение

Предложенная методика задает целостную схему сравнительного анализа методов обнаружения VPN на семействе эксплуатационных доменов. Научная новизна методики состоит в совместном использовании доменной постановки, интегральных показателей качества, индекса переносимости, интегральной устойчивости к маскировке, оценки worst-case качества и анализа чувствительности к domain shift.

Ключевой результат методики состоит в переходе от оценки метода в одном статическом сценарии к анализу его поведения на семействе контролируемых доменов эксплуатации. Такой подход позволяет выявлять случаи, когда методы с близкими значениями F1 в базовом домене существенно различаются по переносимости, устойчивости к маскировке и чувствительности к изменению сетевой среды.

Практическая значимость методики заключается в том, что она задает воспроизводимый протокол сравнения существующих решений и позволяет формировать более обоснованные требования к разрабатываемым системам обнаружения VPN. Применение данной методики дает возможность выбирать методы не по разовому максимуму качества, а по совокупности свойств, критически важных для эксплуатации в корпоративных сетях.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Universal Encryptor: The number of cyberattacks in Russia increased by 140% in 2023 // F.A.C.C.T. URL: <https://www.facct.ru/media-center/press-releases/cyberattacks-2023/> (accessed: 21.04.2024).
2. Wu H. et al. Real-time identification of VPN traffic based on counting Bloom filter and chained hash table from sampled data in high-speed networks // ICC 2022-IEEE International Conference on Communications. IEEE, 2022. P. 5070-5075.
3. Wu H. et al. RT-CBCH: Real-time VPN Traffic Service Identification based on Sampled Data in High-speed Networks // IEEE Transactions on Network and Service Management. 2023.
4. Обзор методов обнаружения VPN и DNS-анализ для классификации на примере Hotspot Shield Free / К. Н. Канатъев, С. Р. Шишкин, И. И. Басыров [и др.] // Инновации и инвестиции. 2023. № 10. С. 215-219.
5. Передовые методы обнаружения и обфускации VPN-трафика: углубленный анализ OpenVPN и его уязвимостей в современную цифровую эпоху / К. Н. Канатъев, М. Д. Нурмагомедов, А. А. Гребенщиков [и др.] // Инновации и инвестиции. 2023. № 10. С. 211-214.

6. Басыня Е. А. Система самоорганизующегося виртуального защищенного канала связи / Е. А. Басыня // Защита информации. Инсайд. 2018. № 5 (83). С. 10-15.
7. System of a self-organizing virtual secure communication channel based on stochastic multi-layer encryption and overlay technologies / E. A. Basinya, Z. B. Akhayeva, D. H. Omarkhanova, G. B. Tolegenova [et al.] // Journal of Theoretical and Applied Information Technology. 2022. Vol. 100, iss. 16. P. 4918-4927.
8. Басыня Е. А. Программная реализация и исследование системы интеллектуально-адаптивного управления информационной инфраструктурой предприятия / Е. А. Басыня // Вестник Самарского государственного технического университета. Серия: Технические науки. 2020. Т. 28, № 1 (65). С. 6-21.
9. Akbari I., Salahuddin M. A., Aniva L. [et al.]. A Look Behind the Curtain: Traffic Classification in an Increasingly Encrypted Web // Proceedings of the ACM on Measurement and Analysis of Computing Systems. 2021. Vol. 5, No. 1. Article 4. DOI: 10.1145/3447382.
10. Xue D., Ramesh R., Jain A. [et al.]. OpenVPN Is Open to VPN Fingerprinting // 31st USENIX Security Symposium (USENIX Security 22). 2022. P. 483-500.
11. Xue D., Kallitsis M., Houmansadr A., Ensafi R. Fingerprinting Obfuscated Proxy Traffic with Encapsulated TLS Handshakes // 33rd USENIX Security Symposium (USENIX Security 24). 2024. P. 2689-2706.
12. Malekghaini N., Akbari E., Salahuddin M. A. [et al.]. Deep Learning for Encrypted Traffic Classification in the Face of Data Drift: An Empirical Study // Computer Networks. 2023. Vol. 225. Article 109648. DOI: 10.1016/j.comnet.2023.109648.
13. Lv S., Wang C., Wang Z. [et al.]. AAE-DSVDD: A One-Class Classification Model for VPN Traffic Identification // Computer Networks. 2023. Vol. 236. Article 109990. DOI: 10.1016/j.comnet.2023.109990.
14. Hanlon M., Wan G., Ascheman A., Durumeric Z. Detecting VPN Traffic through Encapsulated TCP Behavior // Free and Open Communications on the Internet. 2024. P. 77-82.
15. Razooqi Y. S., Pekar A. VPN Traffic Analysis: A Survey on Detection and Application Identification // IEEE Access. 2025. DOI: 10.1109/ACCESS.2025.3592152.
16. Guerra-Manzanares A., Caprolu M., Di Pietro R. A Comprehensive Review on Machine Learning-Based VPN Detection: Scenarios, Methods, and Open Challenges // Computer Science Review. 2025. Vol. 58. Article 100781. DOI: 10.1016/j.cosrev.2025.100781.
17. Arp D., Quiring E., Pendlebury F. [et al.]. Dos and Don'ts of Machine Learning in Computer Security // 31st USENIX Security Symposium (USENIX Security 22). 2022.
18. О внесении изменений в Критерии оценки материалов и информации для включения интернет-ресурсов в единый реестр запрещенной информации: Приказ Роскомнадзора от 8 нояб. 2023 г. № 168 // Собрание законодательства Рос. Федерации. – 2023. – 1 дек.

© Е. А. Басыня, В. Ю. Сапегин, 2026

М. Ю. Благодаренко^{1✉}, Т. Ю. Бугакова¹

Развертывание защищенной инфраструктуры удаленных рабочих мест с системой мониторинга на базе Proxmox VE и Docker контейнеров

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: maxblag04@gmail.com

Аннотация. Статья посвящена проблеме организации удаленного доступа к рабочим местам для сотрудников. Коммерческие решения часто избыточно дороги и не предоставляют достаточной гибкости при настройке системы. Целью исследования является развертывание защищенной инфраструктуры удаленных рабочих мест на программном обеспечении с открытым исходным кодом. Для этого были использованы такие методы, как: виртуализация (Proxmox VE), настройка межсетевого экранирования (pfSense, OpenVPN) и контейнеризация системы мониторинга (Zabbix в Docker). Результатом являются готовые к внедрению виртуальные машины, сетевой оффлайн репозиторий для обновления пакетов на них, шифрованное VPN-туннелирование и централизованный сбор метрик со всех машин. Стресс-тесты и тесты на отказоустойчивость подтвердили стабильность, как отдельных виртуальных машин, так и системы целиком. Подводя итоги, можно сказать, что предложенная архитектура полностью соответствует поставленным задачам и предоставляет надежное решение для практического внедрения инфраструктуры в организацию.

Ключевые слова: защищенная инфраструктура, виртуальная машина, система мониторинга

M. Yu. Blagodarenko^{1✉}, T. Yu. Bugakova¹

Deployment of a secure remote workplace infrastructure with a monitoring system based on Proxmox VE and Docker containers

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: maxblag04@gmail.com

Abstract. The article is devoted to the problem of organizing remote access to workplaces for employees. Commercial solutions are often unnecessarily expensive and do not provide sufficient flexibility when configuring the system. The purpose of the study is to deploy a secure remote workplace infrastructure based on open source software. For this, methods such as virtualization (Proxmox VE), firewall configuration (pfSense, OpenVPN) and containerization of the monitoring system (Zabbix in Docker) were used. The result is ready-to-deploy virtual machines, an offline network repository for updating packages on them, encrypted VPN tunneling, and centralized collection of metrics from all machines. Stress tests and fault tolerance tests have confirmed the stability of both individual virtual machines and the entire system. Summing up, we can say that the proposed architecture is fully consistent with the tasks set and provides a reliable solution for the practical implementation of infrastructure in the organization.

Keywords: secure infrastructure, virtual machine, monitoring system

Введение

Современные организации все больше переходят на удаленный формат работы. Распределенная структура персонала диктует новые требования к ИТ-инфраструктуре, в частности, необходимость создания защищенного периметра для безопасного доступа сотрудников к внутренним ресурсам компании без ущерба для конфиденциальности данных.

Вместе с тем, анализ рынка показывает, что готовые коммерческие решения для организации удаленного доступа часто обладают избыточной стоимостью для малого и среднего бизнеса, либо не предоставляют необходимого уровня гибкости и контроля над инфраструктурой безопасности. В связи с этим, целесообразным представляется использование программного обеспечения с открытым исходным кодом. Применение таких инструментов, как гипервизор Proxmox VE, межсетевой экран pfSense и система мониторинга Zabbix, позволяет построить отказоустойчивую и экономически эффективную среду, адаптированную под конкретные нужды организации.

Целью исследования является построение полностью функционирующей защищенной инфраструктуры удаленных рабочих мест с использованием только программного обеспечения с открытым исходным кодом, а также подтверждение ее работоспособности и отказоустойчивости. Для достижения цели решены следующие задачи: спроектирована архитектура инфраструктуры, межсетевого экрана и системы мониторинга; развернуты виртуальные рабочие станции на базе Astra Linux; организован защищенный VPN-доступ к ним через OpenVPN; реализован централизованный сбор метрик и мониторинг состояния всех узлов.

Практическая значимость исследования заключается в возможности внедрения предложенного решения в организациях для обеспечения безопасной удаленной работы сотрудников. Разработанная инфраструктура была выполнена в рамках реального задания производственного отдела ООО «СИГНАТЕК» и принята в опытную эксплуатацию.

Методы и материалы

В рамках реальных производственных задач ООО «СИГНАТЕК» было разработано и утверждено техническое задание, которое определило совокупность обязательных требований к развертываемой инфраструктуре удаленных рабочих мест. Документ предписывал использование платформы виртуализации Proxmox VE версии 8.0, развертывание десяти виртуальных машин под управлением Astra Linux Common Edition 1.7.6 с заданной минимальной конфигурацией, создание межсетевого экрана pfSense с OpenVPN-сервером, организацию внутреннего АРТ-репозитория на базе Apache, а также выделение отдельного сервера для системы мониторинга Zabbix в среде Docker.

В части обеспечения безопасности техническое задание требовало настроить правила файрвола с открытием UDP-порта 1194 для VPN, запретить прямой доступ к серверу мониторинга из внешней сети, организовать выделенную VPN-сеть в диапазоне 10.0.8.0/24 с фиксированными адресами для узлов, обеспечить

невозможность доступа к любой виртуальной машине без активного VPN-подключения, а также провести тестирование всех защитных механизмов. Выполнению каждого из перечисленных пунктов посвящены соответствующие этапы практической реализации, описанные в разделе «Результаты».

В процессе выполнения исследования применялись следующие методы. Метод анализа предметной области – использовался для обоснования выбора компонентов инфраструктуры и изучения существующих решений [1]. Метод архитектурного моделирования, который позволил спроектировать взаимосвязь между гипервизором, межсетевым экраном, виртуальными машинами и системой мониторинга. Архитектура визуализирована при помощи аппаратно-программной диаграммы взаимосвязей компонентов инфраструктуры на рисунке А.1 (Приложение А). Одним из самых важных являлся метод экспериментального тестирования, этот метод включал в себя проведение стресс-тестов виртуальных машин при помощи специализированного скрипта, а также тестов отказоустойчивости путем имитации отказа работоспособности различных компонентов инфраструктуры. Последним методом является сравнительный анализ, который применялся для оценки эффективности предложенного решения по сравнению с коммерческими аналогами.

В качестве материалов использовалось серверное оборудование Dell PowerEdge с 64 ГБ оперативной памяти. Используемое программное обеспечение включало платформу виртуализации Proxmox VE, межсетевой экран pfSense с OpenVPN, систему мониторинга Zabbix, которая была развернута в Docker контейнерах, а также гостевую операционную систему Astra Linux 1.7.6.

Результаты

Была выполнена полная практическая реализация развертывания защищенной инфраструктуры удаленных рабочих мест в полном соответствии с техническими требованиями ООО «СИГНАТЕК». Первым этапом была выполнена настройка серверного оборудования, конфигурирование отказоустойчивого RAID-массива и установка системы виртуализации Proxmox VE, это обеспечило надежную аппаратную и программную подготовку для выполнения последующих задач. С использованием веб-интерфейса гипервизора созданы десять рабочих станций на базе Astra Linux, узел межсетевого экрана pfSense, и сервер сетевого оффлайн репозитория A-Repo с расширенным дисковым пространством (рис. 1).

PROXMOX Виртуальная среда

Server View

Центр обработки данных

Узел TDA3

Поиск

100 (pfSense)
101 (RM-1)
102 (RM-2)
103 (RM-3)
104 (RM-4)
105 (RM-5)
106 (RM-6)
107 (RM-7)
108 (RM-8)
109 (RM-9)
110 (RM-10)
111 (A-Repo)
локальная сеть (TDA3)
local-btrfs (TDA3)

Краткие сведения
Примечания
Оболочка
Система
Сеть
Сертификаты
DNS
Хозяева
Опции
Время
Системный журнал
Обновления
Репозитории
Брандмауэр
Диски
LVM
LVM-Тонкий
Каталог
ZFS
Циф

Тип	Описание	Использов...	Использов...	Использов...	Время безот...	Использов...	Использов...
qemu	100 (pfSense)	0.0 %	7.7 %	1.4 % от 2 ...	00:00:46	0.7 % от 4 ...	1.0 %
qemu	101 (RM-1)	0.0 %	10.5 %	2.7 % от м...	00:00:45	0.7 % от 4 ...	2.1 %
qemu	102 (RM-2)	0.0 %	10.6 %	2.4 % от м...	00:00:44	0.6 % из 4 ...	2.1 %
qemu	103 (RM-3)	0.0 %	10.9 %	2.4 % от м...	00:00:43	0.6 % из 4 ...	2.1 %
qemu	104 (RM-4)	0.0 %	11.0 %	2.4 % от м...	00:00:41	0.6 % из 4 ...	2.1 %
qemu	105 (RM-5)	0.0 %	10.4 %	2.4 % от м...	00:00:40	0.6 % из 4 ...	2.1 %
qemu	106 (RM-6)	0.0 %	10.8 %	2.4 % от м...	00:00:39	0.6 % из 4 ...	2.1 %
qemu	107 (RM-7)	0.0 %	10.4 %	2.4 % от м...	00:00:38	0.6 % из 4 ...	2.0 %
qemu	108 (RM-8)	0.0 %	10.2 %	2.5 % от м...	00:00:37	0.6 % из 4 ...	2.0 %
qemu	109 (RM-9)	0.0 %	10.2 %	2.4 % от м...	00:00:36	0.6 % из 4 ...	2.0 %
qemu	110 (RM-10)	0.0 %	10.9 %	2.4 % от м...	00:00:35	0.6 % из 4 ...	2.1 %
qemu	111 (A-Repo)	0.0 %	10.4 %	2.5 % от м...	00:00:33	0.6 % из 4 ...	2.1 %
sdn	локальная сеть (TDA3)	-	-	-	-	-	-
Хран...	local-btrfs (TDA3)	6.0 %	-	-	-	-	-

Рис. 1. Веб-интерфейс Proxmox VE

Особое внимание уделено настройке межсетевого экрана на базе pfSense, где организован защищенный сетевой периметр: настроены WAN и LAN интерфейсы, активирован DHCP-сервер, прописаны правила файрвола и реализован NAT для порта UDP 1194. Также одним из самых важных пунктов являлось создание OpenVPN-сервера с шифрованием трафика, для которого сгенерированы клиентские конфигурационные файлы для удаленных сотрудников.

Для обеспечения автономности и повышения безопасности инфраструктуры создан сетевой оффлайн репозиторий, позволяющий устанавливать и обновлять ПО без доступа к внешнему интернету. На отдельном физическом сервере реализована система мониторинга Zabbix в изолированной среде Docker-контейнеров. На всех узлах инфраструктуры установлен Zabbix Agent, и настроен сбор ключевых метрик, добавлены хосты в веб-интерфейс и сформирован дашборд для визуализации состояния ресурсов (рис. 2).

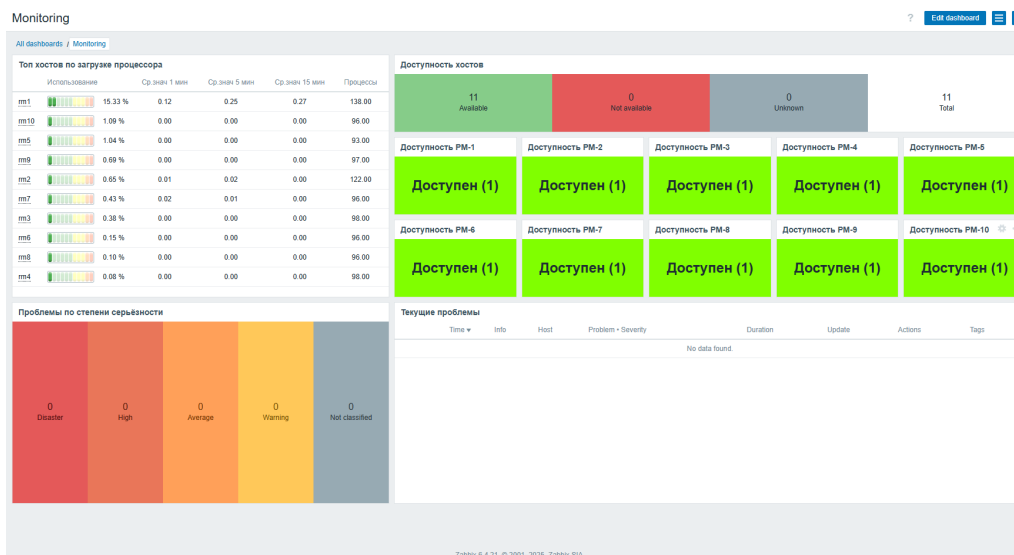


Рис. 2. Дашборд системы мониторинга

Обсуждение

Полученные в ходе исследования результаты подтверждают, что инфраструктура удаленных рабочих мест на базе открытого программного обеспечения способна обеспечить уровень доступности, отказоустойчивости, а главное безопасности, сопоставимый с коммерческими решениями. При этом система не включает в себя ограничений для масштабирования наподобие лицензионных правил.

Сравнение с результатами авторов [2, 3], изучавших контейнеризацию приложений, показывает, что выбранный подход с разделением Zabbix на три Docker контейнера обеспечивает изоляцию сервисов без потери производительности. Также стресс-тестирование позволило наглядно убедиться в том, что виртуальные рабочие станции сохраняют работоспособность при 100 % загрузке процессора в течение 60 секунд, что сопоставимо с результатами, представленными в исследованиях, где аналогичные показатели достигались на физическом оборудовании [4, 5].

Заключение

Основной идеей исследования являлось подтверждение возможности построения полностью функционирующей защищенной инфраструктуры удаленных рабочих мест с использованием только программного обеспечения с открытым исходным кодом. Ключевые результаты – стабильная работа десяти виртуальных рабочих станций с доступностью только через VPN туннелирование и централизованный сбор метрик для системы мониторинга.

Спроектированная и развернутая инфраструктура была передана в опытную эксплуатацию производственному отделу ООО «СИГНАТЕК», что подтверждает ее практическую пригодность и полное соответствие утвержденному техническому заданию. Направления дальнейших исследований включают оценку

максимального порога масштабирования инфраструктуры, а также поиск способов повышения отказоустойчивости сетевого шлюза.

Благодарности

Авторы выражают благодарность производственному отделу ООО «СИГ-НАТЕК» за содействие при выполнении работы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Корпоративная ИТ-инфраструктура: [сайт]. – URL: <https://cyberleninka.ru/article/n/korporativnaya-it-infrastruktura> (дата обращения: 16.04.2026). – Режим доступа: свободный. – Текст: электронный.
2. Системный подход к безопасности пользовательских файлов: от первичной валидации до изоляции в Docker: [сайт]. – URL: <https://cyberleninka.ru/article/n/sistemnyy-podhod-k-bezopasnosti-polzovatelskih-faylov-ot-pervichnoy-validatsii-do-izolyatsii-v-docker> (дата обращения: 16.04.2026). – Режим доступа: свободный. – Текст: электронный.
3. Контейнеризация и автоматизация развертывания веб-приложений: анализ и практическая реализация: [сайт]. – URL: <https://cyberleninka.ru/article/n/konteynerizatsiya-i-avtomatizatsiya-razvertyvaniya-veb-prilozheniy-analiz-i-prakticheskaya-realizatsiya> (дата обращения: 16.04.2026). – Режим доступа: свободный. – Текст: электронный.
4. Повышение надежности ИТ-инфраструктуры при использовании виртуализации: [сайт]. – URL: <https://cyberleninka.ru/article/n/povyshenie-nadezhnosti-it-infrastruktury-pri-ispolzovanii-virtualizatsii> (дата обращения: 16.04.2026). – Режим доступа: свободный. – Текст: электронный.
5. Виртуализация и виртуальные машины в подготовке современных ИТ-специалистов: [сайт]. – URL: <https://cyberleninka.ru/article/n/virtualizatsiya-i-virtualnye-mashiny-v-podgotovke-sovremennyh-it-spetsialistov> (дата обращения: 16.04.2026). – Режим доступа: свободный. – Текст: электронный.

© М. Ю. Благодаренко, Т. Ю. Бугакова, 2026

ПРИЛОЖЕНИЕ А АРХИТЕКТУРА ИНФРАСТРУКТУРЫ

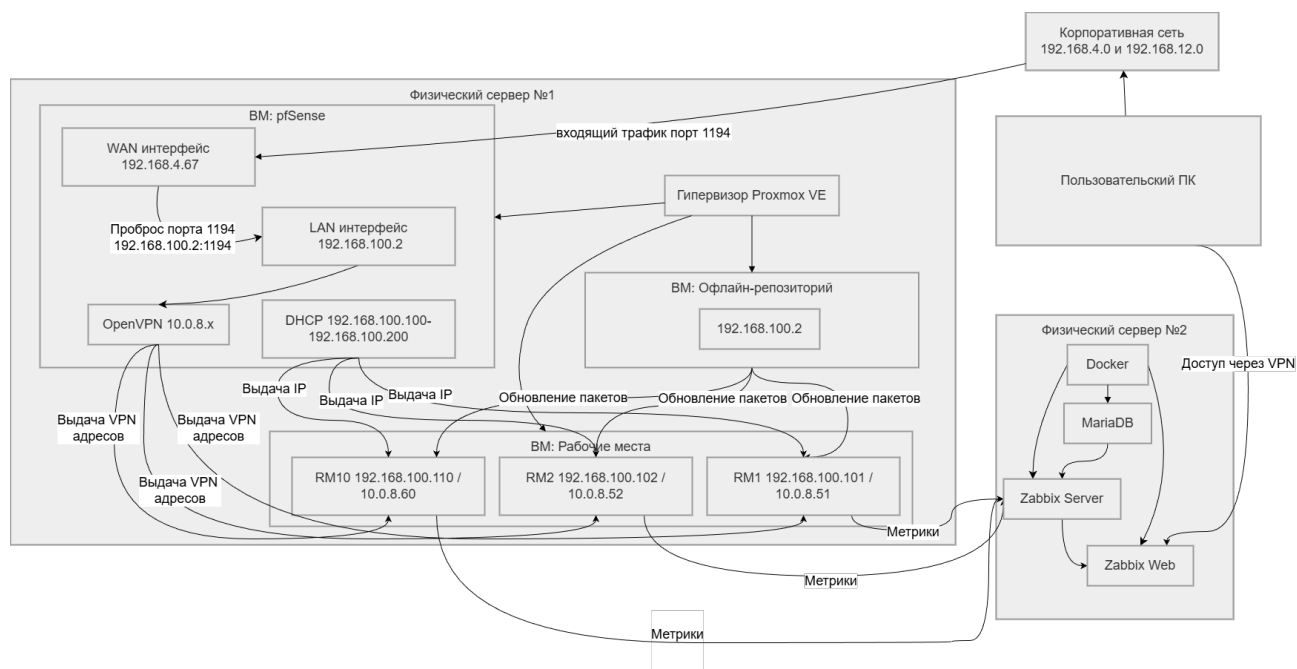


Рис. А.1. Аппаратно-программная архитектура инфраструктуры удаленных рабочих мест с системой мониторинга

А. А. Блохин^{1✉}, А. А. Шаранов¹

Программное обеспечение для анализа данных электронных таблиц

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: blohin4337@gmail.com

Аннотация. В статье представлены результаты разработки специализированного программного обеспечения для анализа сбалансированности планов закупок и продаж магазина тканей. Выполнен анализ предметной области, в ходе которого выявлены систематические отклонения закупок от плановых объемов, приводящие к дефициту ассортиментных позиций и образованию избыточных остатков по номенклатуре при всем годовом ассортименте товаров. Проведен обзор существующих решений, включая табличные процессы, учетные системы на платформе «1С:Предприятие» и BI-платформы, показавший их ограниченную применимость для специфических аналитических задач швейного ритейла в условиях автономной работы. Для реализации программного продукта использован стек технологий, объединяющий язык Python 3.10, библиотеки pandas и numpy, matplotlib и PyQt5 для построения графического интерфейса. Разработанное приложение обеспечивает загрузку двух Excel-файлов, расчет агрегированных ключевых показателей эффективности, построение тепловых карт сезонности закупок в трех аналитических разрезах, совмещенную ABC-XYZ классификацию ассортимента с вычислением коэффициента вариации помесечных данных, оценку рисков дефицита и затоваривания на основе балансовой модели, а также формирование отчетной документации в форматах HTML.

Ключевые слова: анализ, закупки, классификация, визуализация, Python

A. A. Blokhin^{1✉}, A. A. Sharapov¹

Software for analyzing spreadsheet data

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: blohin4337@gmail.com

Abstract. The article presents the results of developing specialized software for analyzing the balance of procurement and sales plans of a fabric store. An analysis of the subject area was carried out, revealing systematic deviations of purchases from planned volumes, leading to a shortage of assortment items and the formation of excess stock across the nomenclature for the entire annual product range. A review of existing solutions was conducted, including spreadsheet processors, accounting systems based on the 1C:Enterprise platform, and BI platforms, which showed their limited applicability to the specific analytical tasks of sewing retail under autonomous operating conditions. The software product was implemented using a technology stack combining the Python 3.10 language, the pandas and numpy libraries, matplotlib, and PyQt5 for building the graphical user interface. The developed application provides loading of two Excel files, calculation of aggregated key performance indicators, construction of seasonal procurement heat maps in three analytical dimensions, combined ABC-XYZ classification of the assortment with calculation of the coefficient of variation of monthly data, assessment of shortage and overstock risks based on a balance model, as well as generation of reporting documentation in HTML format.

Keywords: analysis, procurement, classification, visualization, Python

Введение

Особенность работы при ручном анализе закупок и продаж в работе магазина ЛОГОС показывает проблему, которая характеризуется отсутствием инструмента для аналитики данных в магазине, сотрудники вынуждены пользоваться стандартными средствами анализа, такими, как например - сводные таблицы в Excel, что является не удобным и для нынешнего времени устаревшей методикой анализа данных, так как требует обученного специалиста с навыками анализа такого типа данных. Разработка такого инструмента – позволяющего проводить анализ закупок и продаж без специальных навыков, необходима для возможной оптимизации работы производства.

Программа, которая доступна для абсолютно любого сотрудника ответственного за выбранную область не требующая особых навыков работы с таблицами и аналитики позволит сотруднику не тратить время на аналитику и позволит сконцентрироваться на другой не менее важной работе.

Исследование предметной области

Магазин тканей «Логос» функционирует на рынке розничной торговли с 1991 года, пройдя путь от оптовых поставок фурнитуры до многоформатного ритейлера, объединяющего реализацию тканей, текстиля, швейной техники и сопутствующих товаров. Ассортиментная матрица насчитывает несколько сотен позиций, сгруппированных по категориям: одежда, ткани, нитки и пряжа, фурнитура, аксессуары. Поставки осуществляются производителями нескольких стран, что формирует многомерную структуру закупок с выраженной сезонной вариацией – на пиковые весенние и осенние месяцы приходится значительная доля годового оборота.

Обследование бизнес-процессов выявило систематическое рассогласование планов продаж и закупок. Годовой план реализации разрабатывается коммерческим отделом, однако закупочный план утверждается обособленно, без должной координации. Вследствие этого заметная часть артикулов регулярно попадает в зону дефицита, когда начальный остаток и плановые поступления не покрывают запланированную реализацию, тогда как сопоставимая доля номенклатуры характеризуется избыточным прогнозным остатком, замораживающим оборотные средства.

Существующая аналитическая практика ограничивается эпизодическим использованием сводных таблиц Excel. При значительном числе позиций ручная обработка становится чрезмерно трудоемкой, а построение тепловых карт сезонности и выполнение ABC-XYZ классификации с расчетом коэффициента вариации требуют либо написания макросов, либо наличия специалиста с продвинутыми аналитическими навыками. Ни то, ни другое не реализовано на постоянной основе, что лишает менеджмент инструмента оперативного контроля.

Анализ альтернативных решений показал их ограниченную применимость. Системы на платформе «1С: Предприятие» ориентированы на сквозной учет и

требуют дорогостоящей доработки. VI-платформы предлагают развитую визуализацию, но их лицензионная стоимость и потребность в постоянном интернет-подключении делают их экономически нецелесообразными для малого предприятия. Таким образом, установлено, что ни одно готовое решение не удовлетворяет совокупности требований «Логоса»: автономность, работа с двумя Excel-файлами, автоматизация полного цикла анализа и формирование отчетности. Данный вывод послужил основанием для разработки специализированного программного продукта.

Методы и технологии

Выбор технологического стека для реализации программного продукта осуществлялся на основе критериев, сформулированных по результатам анализа предметной области. К числу таких критериев отнесены автономность функционирования без развертывания серверной инфраструктуры и без постоянного подключения к сети Интернет, возможность обработки файлов формата Microsoft Excel, наличие средств построения специализированных визуализаций, обеспечение удобного графического интерфейса, а также формирование отчетной документации без привлечения дополнительного платного программного обеспечения.

В качестве базового языка разработки принят Python версии 3.10. Данный выбор обусловлен синтаксической простотой и читаемостью кода, наличием мощных библиотек для обработки и визуализации данных, кроссплатформенностью, а также широкими возможностями интеграции с графическими библиотеками для создания пользовательского интерфейса.

Для работы с табличными данными и выполнения вычислительных операций задействована библиотека `pandas`, обеспечивающая импорт электронных таблиц форматов `xlsx` и `xls` через функцию `read_excel`, слияние датафреймов по ключевому полю «Товар», группировку записей по категориям, типам и странам-производителям, а также расчет нарастающих итогов, необходимых для ABC-классификации. Численные операции оптимизированы посредством интеграции с библиотекой `numpy`. Расчет коэффициента вариации для XYZ-анализа реализован как отношение стандартного отклонения помесечных закупок к их среднему арифметическому значению с добавлением малой константы в знаменатель, исключающей деление на ноль.

Визуализация ключевых показателей и тепловых карт выполнена средствами библиотеки `matplotlib`. Тепловые карты сезонности строятся методом `imshow` с цветовой палитрой `YlOrRd`, обеспечивающей наглядное различие интенсивности закупок по месяцам. На каждую ячейку наносится абсолютное числовое значение, причем цвет текста выбирается контрастным относительно фона. Интеграция `matplotlib`-фигур в графический интерфейс осуществлена через адаптер `FigureCanvasQTAgg`.

В качестве фреймворка для построения пользовательского интерфейса применен `PyQt5`. Интерфейс организован вокруг главного окна `QMainWindow` и включает вкладочный блок `QTabWidget` с четырьмя аналитическими панелями.

Для предотвращения блокировки графического интерфейса на время выполнения ресурсоемких операций задействован класс `QThread`, выносящий аналитическую обработку в отдельный поток. Обмен данными между потоками организован посредством сигналов `progress` и `finished`.

Генерация отчетной документации реализована с использованием стандартных средств языка Python. HTML-отчет формируется путем сборки строкового представления, включающего изображения тепловых карт в кодировке `base64`. CSV-отчет создается штатным методом `to_csv` библиотеки `pandas`. Модульная архитектура приложения разделяет слои представления, бизнес-логики и доступа к данным, что обеспечивает независимость компонентов и упрощает возможную модификацию системы.

Результаты

Разработано оконное приложение на базе Python (рис. 1), реализующее полный цикл анализа закупок и продаж магазина тканей. Программный продукт включает четыре функциональные вкладки. Первая вкладка содержит пять цветowych карточек агрегированных показателей: план продаж, объем закупок, начальный остаток, прогнозный остаток и коэффициент покрытия. Здесь же размещен фильтр по типам товаров с динамически обновляемой диаграммой десяти наиболее значимых позиций.

Вторая вкладка отображает тепловые карты сезонности закупок в трех разрезах: по категориям, типам товаров и странам-производителям. Интенсивность поставок кодируется цветовым градиентом с наложением числовых меток. Третья вкладка содержит ABC-XYZ матрицу размерностью три на три, полученную классификацией ассортимента по годовому плану продаж и коэффициенту вариации помесечных закупок, с дифференцированными рекомендациями для каждой группы.

Четвертая вкладка оценивает риски: прогноз конечного остатка вычисляется балансовым уравнением, позиции классифицируются как дефицитные, избыточные либо нормальные. Отчетный модуль формирует автономный HTML-документ с встроенными графиками и таблицами, а также CSV-файл с расчетными колонками. Полный цикл анализа выполняется менее чем за семь секунд, интерфейс осваивается сотрудником без технической подготовки в течение нескольких минут.

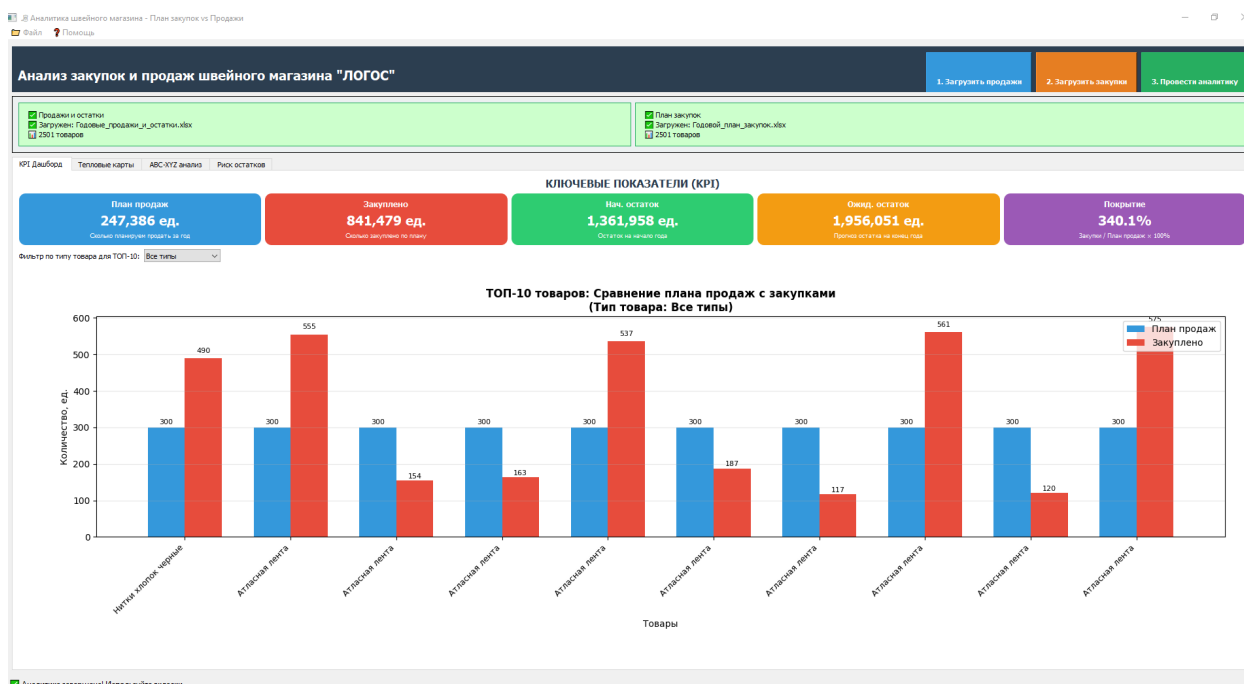


Рис. 1. Интерфейс оконного приложения

Заключение

Разработанное программное обеспечение решает задачу автоматизированного анализа сбалансированности планов закупок и продаж магазина тканей «Логос». Приложение устраняет выявленный в ходе исследования предметной области разрыв между потребностью в оперативной аналитике и возможностями ранее применявшихся табличных средств, обеспечивая выполнение полного цикла обработки данных – от загрузки исходных файлов до получения регламентированных отчетов – без привлечения дополнительного платного программного обеспечения и без необходимости постоянного интернет-подключения.

Практическая значимость полученных результатов заключается в снижении трудозатрат на аналитическую обработку данных и предоставлении сотрудникам, не обладающим специализированными навыками, инструмента для наглядного сопоставления плановых и фактических показателей, выявления сезонных закономерностей и количественной оценки рисков. Автоматическое формирование отчетной документации в форматах HTML и CSV обеспечивает возможность как визуального анализа, так и дальнейшей обработки данных внешними средствами. Дальнейшее развитие системы может быть связано с добавлением модуля прогнозирования спроса на основе временных рядов и реализацией механизма автоматической выработки рекомендаций по объемам и срокам закупок для каждой товарной позиции.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Маккинни, У. Python и анализ данных: Пер. с англ. / У. Маккинни. – 3-е изд. – Москва: ДМК Пресс, 2023. – 534 с. – ISBN 978-5-93700-174-0.

2. Прохоренок, Н. А. Python 3 и PyQt 6. Разработка приложений / Н. А. Прохоренок, В. А. Дронов. – Санкт-Петербург: БХВ-Петербург, 2023. – 832 с. – ISBN 978-5-9775-1706-5.
3. VanderPlas, J. Python Data Science Handbook: Essential Tools for Working with Data / J. VanderPlas. – 2nd ed. – Sebastopol: O'Reilly Media, 2023. – 588 p. – ISBN 978-1-098-12122-8.
4. Лутц, М. Изучаем Python: Т. 1: Пер. с англ. / М. Лутц. – 6-е изд. – Санкт-Петербург: Диалектика, 2023. – 832 с. – ISBN 978-5-907705-03-4.
5. Лутц, М. Изучаем Python: Т. 2: Пер. с англ. / М. Лутц. – 6-е изд. – Санкт-Петербург: Диалектика, 2023. – 720 с. – ISBN 978-5-907705-04-1.
6. Грас, Дж. Data Science с нуля: Пер. с англ. / Дж. Грас. – 2-е изд. – Санкт-Петербург: БХВ-Петербург, 2021. – 416 с. – ISBN 978-5-9775-6730-5.
7. Хантер, Д. Д. Matplotlib. Визуализация данных на Python / Д. Д. Хантер, Д. Дейл, Э. Фестинг. – Москва: ДМК Пресс, 2019. – 520 с. – ISBN 978-5-97060-761-6.

© А. А. Блохин, А. А. Шаранов, 2026

А. Р. Валов^{1✉}, Н. С. Головачев¹

Разработка прототипа симулятора эвакуации и поведения в чрезвычайных ситуациях на основе 3DGS-моделей реальных учебных помещений

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: Valovanryuha@yandex.kz

Аннотация. В статье представлены результаты начального этапа разработки симулятора, предназначенного для обучения действиям при чрезвычайных ситуациях. В основе симулятора лежат 3DGS-модели реальных учебных помещений, что обеспечивает высокую степень визуального реализма и пространственной достоверности. На данный момент реализован один уровень, моделирующий террористическое нападение, а также набор последовательных действий для обеспечения безопасности. Проект разрабатывается на платформе Unreal Engine 5. В статье описаны выбранные методы и технологии, логика работы симулятора, а также результаты пилотного тестирования.

Ключевые слова: симулятор ЧС, 3D Gaussian Splatting, Unreal Engine 5, эвакуация, террористическое нападение, виртуальный тренажер

A. R. Valov^{1✉}, N. S. Golovachev¹

Development of a prototype of an evacuation and emergency behavior simulator based on 3DGS models of real educational facilities

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: Valovanryuha@yandex.kz

Abstract. The article presents the results of the initial stage of the simulator development, designed for emergency training. The simulator is based on 3DGS models of real classrooms, which provides a high degree of visual realism and spatial reliability. At the moment, one level has been implemented that simulates a terrorist attack, as well as a set of sequential actions to ensure security. The project is being developed on the Unreal Engine 5 platform. The article describes the selected methods and technologies, the architecture of the simulator, as well as the results of the pilot testing.

Keywords: emergency simulator, 3D Gaussian Splatting, Unreal Engine 5, evacuation, terrorist attack, virtual simulator

Введение

Чрезвычайные ситуации природного, техногенного и социального характера, включая угрозы террористического характера, предъявляют повышенные требования к уровню подготовленности обучающихся, сотрудников и иных лиц, находящихся в зданиях образовательных организаций. Эффективность действий в подобных условиях определяется не только знанием нормативных инструкций, но и сформированностью практических навыков поведения в условиях дефицита времени, ограниченной видимости, стресса и неопределенности.

Проведение регулярных тренировок в реальной среде связано с организационными ограничениями, значительными временными затратами и невозможностью безопасного воспроизведения ряда опасных факторов. В связи с этим актуальным направлением является разработка виртуальных симуляторов, позволяющих отрабатывать алгоритмы действий в контролируемой цифровой среде.

Традиционные подходы к созданию виртуальных сцен ЧС часто используют полностью синтетические 3D-модели, что снижает чувство присутствия и достоверность. В данной работе предлагается использование технологии 3D Gaussian Splatting (3DGS) [1], позволяющей создавать фотореалистичные модели реальных учебных помещений на основе небольшого набора фотографий. Это может повысить иммерсивность обучения и позволит обучаемому узнавать реальное пространство.

Цель проекта состоит в разработке рабочего прототипа интерактивного симулятора эвакуации и защитного поведения при чрезвычайных ситуациях, интегрированного с 3DGS-моделями реальных учебных помещений и реализованного в среде Unreal Engine 5.

Исследование предметной области

ЧС природного и техногенного характера остаются одной из наиболее острых угроз для населения Российской Федерации. Как показывает официальная статистика МЧС России, в любой момент времени существует вероятность возникновения ЧС самого разного характера – от техногенных аварий и пожаров до опасных гидрологических явлений и природных катастроф. Так, согласно Государственному докладу о состоянии защиты населения и территорий РФ за 2024 год, на территории страны произошло 272 чрезвычайные ситуации, из которых 171 (63 %) носила техногенный характер, а 101 (37 %) – природный. Последствия этих событий оказались крайне тяжелыми: общее число погибших составило 494 человека, причем наибольшее количество жертв (474 человека, или 96 %) пришлось именно на техногенные ЧС. Количество пострадавших достигло 696 464 человек, а общий материальный ущерб превысил 70,6 млрд рублей.

Особую тревогу вызывает тот факт, что значительная часть ЧС происходит в местах массового пребывания людей, включая образовательные учреждения. При этом, как показано в докладе МЧС, традиционные формы подготовки населения – инструктажи, лекции и плановые учения – не всегда позволяют сформировать устойчивые навыки поведения в реальной кризисной ситуации, особенно в условиях стресса и ограниченного времени на принятие решений [1].

На российском рынке присутствует ряд программных решений для моделирования чрезвычайных ситуаций, однако в большинстве из них используются абстрактные 3D-сцены, не соответствующие реальной архитектуре и планировке конкретных зданий. Это снижает эффект переноса навыков: обучающийся, оказавшись в реальной чрезвычайной ситуации в родном вузе, может не узнать пространство и растеряться.

На фоне имеющихся разработок использование 3DGS модели в качестве основы для создания симулятора обладает рядом преимуществ:

- потенциальное снижение затрат времени на разработку;
- фотореалистичная и достоверная модель;
- соответствия реальным метрическим характеристикам помещения.

Обучение студентов и преподавателей на виртуальных копиях реальной архитектуры ВУЗа позволит им в безопасной среде отработать алгоритмы действий при пожаре, террористическом нападении или другой угрозе. В случае возникновения реальной ЧС это даст возможность быстрее сориентироваться в знакомом пространстве, правильно выбрать пути эвакуации и укрытия, что напрямую повышает шансы на сохранение жизни и здоровья.

Методы и технологии

Разработку симулятора поведения в чрезвычайных ситуациях было решено вести на базе игрового движка Unreal Engine 5, который зарекомендовал себя как одна из ведущих платформ для создания фотореалистичных интерактивных сред в реальном времени. Выбор данного движка обусловлен его ключевыми преимуществами для задач виртуальной подготовки к ЧС.

К преимуществам Unreal Engine 5 можно отнести следующие моменты:

- готовность Unreal Engine к интеграции и отображению 3DGS моделей;
- наличие технологии Nanite для дополнения 3DGS моделей высокодетализированными 3D моделями различных деталей реальных помещений без значительной потери производительности, автоматически управляя уровнем детализации в зависимости от расстояния до камеры [2];
- визуальная система программирования Blueprints дает возможность разработчикам быстрее реализовывать сложную интерактивную логику;
- библиотека Quixel Megascans предоставляет доступ к обширной коллекции фотореалистичных материалов и 3D-ассетов, позволяя быстро создать убедительное окружение (дым, огонь, разрушения, элементы интерьера), повышающее уровень иммерсивности обучения.

Для воссоздания реалистичной трехмерной модели реального учебного помещения была применена технология 3D Gaussian Splatting (3DGS) [3]. В отличие от традиционного полигонального моделирования, 3DGS позволяет получить фотореалистичную копию пространства на основе большого набора фотографий и облака точек, полученных при помощи SLAM сканера.

В рамках проекта исходные пространственные данные собираются с использованием SLAM-сканера Lixel L2 Pro, после чего выполняется подготовка 3DGS-модели в LCC Studio [4, 5]. Полученная модель сохраняется в формате LCC, предназначенном для дальнейшей работы с 3DGS-контентом (рис. 1).

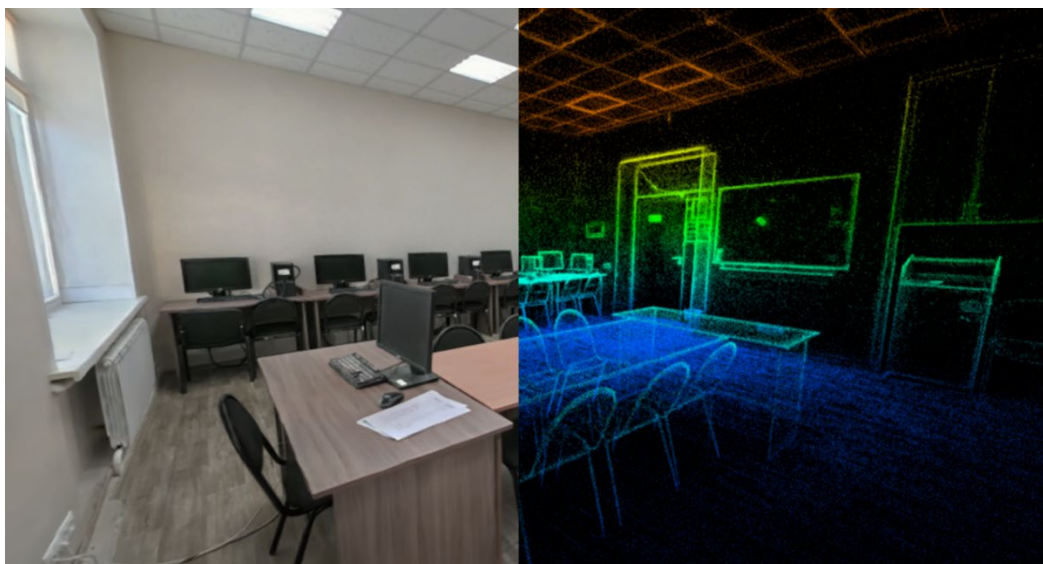


Рис. 1. Пример исходных 3DGS данных отображаемых в Unreal Engine 5

Результаты

В тестовом уровне реализован сценарий поведения пользователя при угрозе нападения на образовательное учреждение. После поступления сигнала тревоги пользователь должен выполнить последовательность защитных действий: закрыть помещение, забаррикадировать вход и занять безопасное положение внутри аудитории. Выполнение действий ограничено по времени, после чего система автоматически проверяет корректность прохождения сценария.

Логика уровня предусматривает анализ состояния интерактивных объектов и положения пользователя в пространстве помещения. В зависимости от выполненных действий система начисляет или снижает условное количество баллов, а также фиксирует ошибки пользователя. Дополнительно реализована проверка нахождения пользователя в безопасных или опасных зонах помещения, что позволяет оценивать не только факт выполнения сценария, но и правильность принятия решений в условиях чрезвычайной ситуации (рис. 2).



Рис. 2. Тестовый уровень учебной аудитории в Unreal Engine 5

Заключение

В результате выполнения работы был разработан действующий прототип симулятора чрезвычайных ситуаций, интегрированный с фотореалистичной моделью реальной учебной аудитории СГУГиТ. Реализованный пилотный уровень позволяет пользователю в безопасной виртуальной среде отработать алгоритм реагирования при угрозе, используя понятные инструменты взаимодействия с окружением. Разработанный сценарий включает последовательность защитных действий. Все вышеперечисленные функции направлены на формирование устойчивых навыков поведения в чрезвычайной ситуации, что критически повышает шансы на сохранение жизни и здоровья при реальном происшествии.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Государственный доклад о состоянии защиты населения и территорий Российской Федерации от чрезвычайных ситуаций природного и техногенного характера в 2024 году / МЧС России. – Москва : МЧС Медиа, 2025. – 272 с. – URL: <https://mchs.gov.ru/uploads/document/2025-05-30/de7fb2c0ea6561054f6b74e2db4cb150.pdf> (дата обращения: 14.05.2026). – Текст : электронный.
2. Epic Games. Unreal Engine 5 Documentation. – Режим доступа: <https://docs.unrealengine.com/5.3/> (дата обращения: 14.05.2026).
3. Kerbl, B. 3D Gaussian Splatting for Real-Time Radiance Field Rendering / B. Kerbl, G. Kopanas, T. Leimkühler, G. Drettakis // ACM Transactions on Graphics. – 2023. – Vol. 42, № 4. – P. 1–14. – URL: <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/> (дата обращения: 14.05.2026).
4. XGRIDS. LCC Studio. – URL: <https://www.xgrids.com/support/download?page=LCCStudio> (дата обращения: 15.05.2026).
5. GraphDECO-INRIA. Gaussian Splatting: official implementation. – URL: <https://github.com/graphdeco-inria/gaussian-splatting> (дата обращения: 19.05.2026).

© А. Р. Валов, Н. С. Головачев, 2026

Д. А. Ваулин^{1✉}, Р. В. Гришин¹

Разработка мультиагентной архитектуры универсальной платформы для автоматизированного обучения чат-ботов

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: aderigio@mail.ru

Аннотация. В статье приведена мультиагентная архитектура универсальной платформы, позволяющая осуществлять автоматизированное обучение чат-ботов на основе машинного обучения. Представленная архитектура распределяет функции между специализированными агентами: центральным, обучающим, агентом обработки запросов, агентом статистики и мониторинга и агентом пользовательского взаимодействия. Такое решение обеспечивает гибкость и масштабируемость системы, позволяя постоянно обновлять базу знаний, генерировать ответы и выдавать их в режиме реального времени. Научная новизна содержится в использовании мультиагентного подхода на универсальной платформе, что открывает новые возможности: алгоритмы предобработки, разметку и обучение на основе текстовых данных и адаптацию модели к условиям реальной эксплуатации. Итоги апробации мультиагентной архитектуры свидетельствуют о значительном повышении точности ответов, их согласованности и контекстной адекватности, и перспективности в автоматизации пользовательского взаимодействия.

Ключевые слова: мультиагентная система, универсальная платформа, автоматизированное обучение, чат-боты, машинное обучение, база знаний

D. A. Vaulin^{1✉}, R. V. Grishin¹

Development of a multi-agent architecture of a universal platform for automated training of chatbots

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: aderigio@mail.ru

Abstract. The article presents a multi-agent architecture of a universal platform that allows automated training of chatbots based on machine learning. The presented architecture distributes functions between specialized agents: central, training, query processing agent, statistics and monitoring agent, and user interaction agent. This solution provides flexibility and scalability of the system, allowing you to constantly update the knowledge base, generate answers and provide them in real time. The scientific novelty is contained in the use of a multi-agent approach on a universal platform, which opens up new possibilities: preprocessing algorithms, markup and training based on text data, and adaptation of the model to real-world operating conditions. The results of the testing of the multi-agent architecture indicate a significant increase in the accuracy of responses, their consistency and contextual adequacy and prospects in automating user interaction.

Keywords: multi-agent system, universal platform, automated learning, chatbots, machine learning, knowledge base

Введение

При использовании традиционных методов обучения чат-ботов и их обслуживания можно наблюдать существенные недостатки. Например, традиционные системы зачастую обладают медленной адаптацией к изменяющимся требованиям бизнеса, что делает их неэффективными. Централизованная архитектура становится причиной перегрузки системы, из-за чего она начинает работать с перебоями и теряет скорость. Все это увеличивает расходы на техническое сопровождение. Кроме того, существующие методы обучения не всегда позволяют продуктивно обновлять базу знаний. Точность ответов снижается, и появляется необходимость во вмешательстве специалистов. А отсутствие распределения задач между специализированными модулями снижает устойчивость системы при возникновении ошибок в ее отдельных частях [1–3].

Применение многоагентного подхода способствует устранению вышеперечисленных недостатков. Распределение заданий между самостоятельными агентами снижает зависимость от централизованной архитектуры, ускоряет обучение и повышает приспособленность системы. Данное решение обеспечит действенную координацию работы всей системы и позволит своевременно поправлять алгоритмы на основе данных статистики. В итоге, разработка мультиагентной архитектуры способствует решению проблем традиционных подходов: улучшению масштабируемости, гибкости и надежности систем для обслуживания клиентов [1–3].

Данное исследование посвящено разработке новой архитектуры, которая будет пригодна для повышения качества работы в постоянно меняющихся деловых процессах и автоматизирования обучения. Итоги исследования могут быть использованы в различных областях, требующих мгновенную и безошибочную обработку запросов от пользователей. Это делает описываемую тему востребованной для современных организаций.

Методы и исследования

Ключевым замыслом мультиагентной системы является объединение нескольких самостоятельных программных сущностей, именуемых агентами. Каждая сущность выполняет определенные задания и взаимодействует с другими, чтобы достичь общие или взаимодополняющие цели [4]. На универсальной платформе, где используются технологии машинного обучения, такой подход позволяет распределять задачи между агентами для обеспечения удобного обслуживания чат-ботов и их обучения.

Интеллектуальный агент – это самостоятельная программная единица, воспринимающая информацию из внешней среды, обрабатывающая ее с помощью встроенных алгоритмов и принимающая решения для достижения поставленных целей [5]. Агент может самообучаться, адаптироваться и взаимодействовать с другими агентами в системе. В многоагентной архитектуре каждая сущность выполняет определенные задачи: анализ информации, сбор данных, управление обучением или генерацию ответов.

Структура интеллектуального агента обладает следующими модулями:

1. Модуль восприятия. Он отвечает за сбор входной информации из внешней среды (например, текстовых сообщений пользователей) и обеспечивает предварительную обработку данных (например, очистку текста от нежелательных символов).

2. Модуль обработки и принятия решений. Он использует алгоритмы машинного обучения для анализа входных данных и формирует промежуточные представления информации.

3. Модуль взаимодействия и коммуникации. Он организует обмен информацией между агентами системы и пользователями. Также обеспечивает выполнение коллективных действий в многоагентной системе.

4. Модуль самообучения и адаптации. Он позволяет обновлять базы знаний агента на основе данных, поступающих из внешней среды. Также использует алгоритмы адаптивного обучения для улучшения качества ответов.

5. Модуль контроля и мониторинга. Он ведет статистику и мониторинг работы агентов. Также позволяет администраторам системы получать обратную связь, что способствует своевременному реагированию на изменения в работе агентов.

В рамках мультиагентной системы для автоматизированного обучения чат-ботов интеллектуальный агент должен достигать основные цели [6, 7]. Например, обеспечивать восприятие и обработку входящих запросов от пользователей. Самостоятельно генерировать ответы на основе заранее обученной модели, чтобы уменьшать вмешательство человека в систему. Адаптироваться к изменениям во входных данных и постоянно обновлять базы знаний. Взаимодействовать с другими агентами системы, чтобы распределять задания, обмениваться сведениями и руководить действиями. Собирать и исследовать статистические данные, чтобы выявлять области для улучшения.

Для успешной реализации поставленных целей интеллектуальный агент должен обладать следующими свойствами [8]:

- автономностью, то есть способностью работать независимо, принимать решения на основе личных алгоритмов;
- адаптивностью, а именно способностью изменять свои алгоритмы обучения в ответ на изменение входных данных или условий работы;
- интерактивностью, то есть взаимодействием с пользователями и другими сущностями посредством протоколов обмена сообщениями;
- масштабируемостью, а именно возможностью работы в рамках большой системы, состоящей из сотни или тысячи одновременно работающих агентов;
- устойчивостью, то есть стабильной работой при возникновении непредвиденных ситуаций;
- безопасностью, а именно наличием механизмов защиты от внешних атак и обеспечением конфиденциальности данных;

– самообучаемостью, то есть возможностью постоянного обновления и пополнения базы знаний на основе новых данных и обратной связи от пользователей.

Результаты

Во время апробации разработанной мультиагентной архитектуры были приобретены наглядные результаты, доказывающие действенность многоагентного подхода при обучении чат-ботов на универсальной платформе. Структура интеллектуального агента (рис. 1) представляет собой систему модулей, где содержатся компоненты обработки, восприятия, контроля, самообучения и коммуникации. Каждый агент направлен на автоматизацию обработки пользовательских запросов с применением машинного обучения, что обеспечивает снижение затрат и повышение эффективности обслуживания.

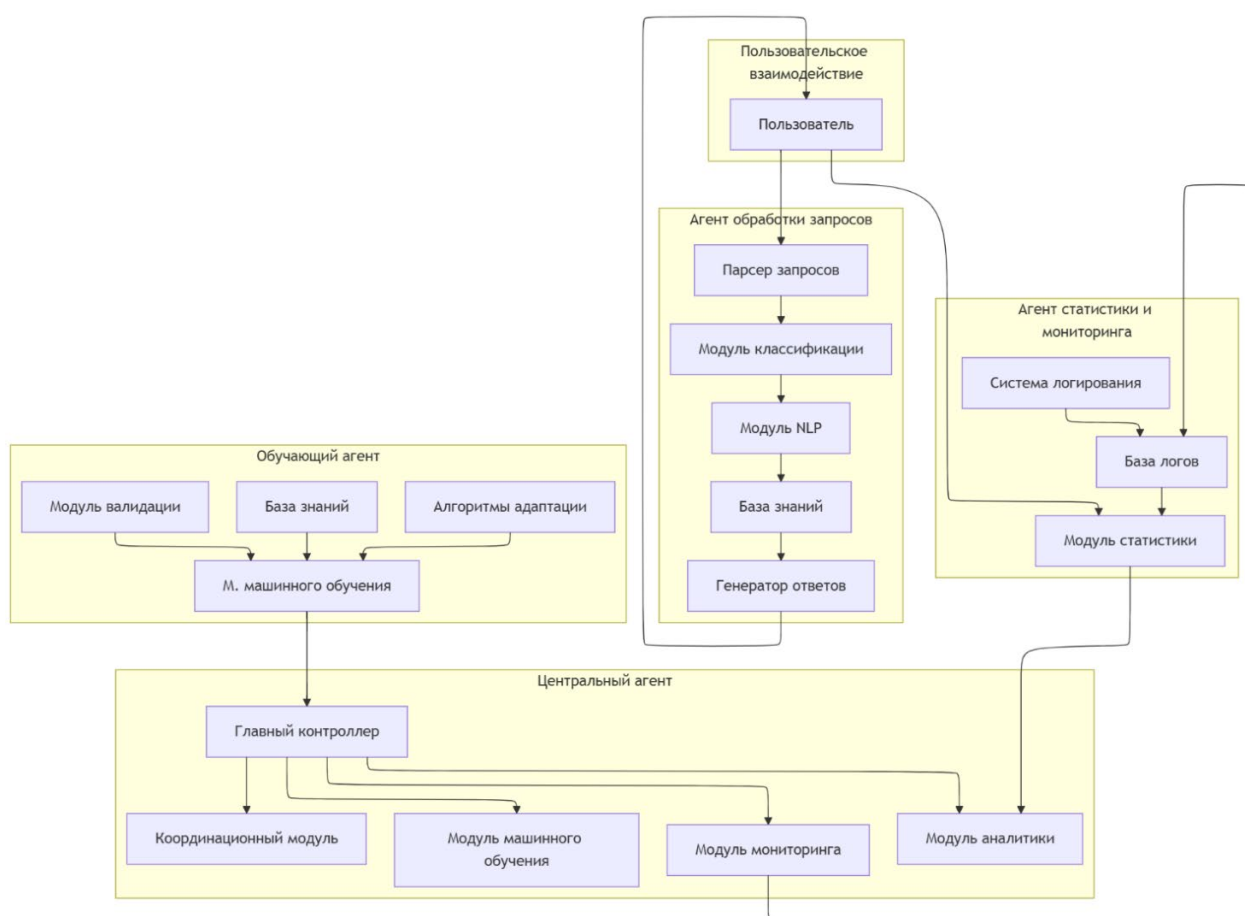


Рис. 1. Мультиагентная система универсальной платформы

Каждый агент в системе имеет свое назначение.

1. Центральный агент является главным компонентом системы, который координирует работу всех остальных агентов, управляет процессами обучения, обработки запросов и анализа данных. Этот агент получает данные от Агента статистики и мониторинга, где данные представлены в формате CSV. Передает

команды Агенту обучения и Агенту обработки запросов. После обработки информации отправляет текстовые ответы Агенту пользовательского взаимодействия, где пользователи получают готовые сообщения.

2. Обучающий агент обеспечивает автоматизированное обучение чат-ботов на основе текстовых данных, разметки информации и алгоритмов машинного обучения. Этот агент получает данные из базы знаний и отправляет обученные модели в Центральный агент. А потом анализирует статистику работы через Агент статистики и мониторинга.

3. Агент обработки запросов отвечает за разбор входящих пользовательских сообщений, анализирует их и формирует соответствующие ответы. Этот агент получает запросы от Агента пользовательского взаимодействия и обращается к базе знаний для поиска информации. Потом передает сформированные ответы Агенту пользовательского взаимодействия.

4. Агент статистики и мониторинга собирает, анализирует и хранит данные о работе чат-ботов и помогает администраторам контролировать их эффективность. Этот агент получает данные из Агента обработки запросов, а затем передает отчеты в Центральный агент и Административный интерфейс.

5. Агент пользовательского взаимодействия организует взаимодействие пользователей с системой, обрабатывает входящие и исходящие сообщения. Этот агент получает обработанные ответы от Агента обработки запросов. Потом взаимодействует с пользователем в режиме реального времени и логирует обращения пользователей в Агент статистики и мониторинга.

Вышеописанные модули используют API для обмена данными, находящимися в формате JSON [9]. Благодаря этому происходит унифицированное взаимодействие между сущностями, начиная с обработки запросов клиентов и заканчивая передачей обучающих данных.

Например, по окончании предобработки обучающий агент получает данные в виде CSV и конвертирует их в упорядоченные объекты JSON (такие объекты подходят для обучения нейросетевых моделей). Потом он передает результаты в центральный агент, где происходит последующее распределение. Данные аналитики (метрики точности, статистика ошибок и т.д.) собираются агентом статистики и хранятся в базе данных SQLite. Это позволяет проводить анализ и оптимизировать модели.

Пробная проверка системы показала ее успешное обрабатывание входящих данных, преобразование их в необходимый вид и применение для обучения моделей. Все это усовершенствовало качество генерируемых ответов (рис. 2).

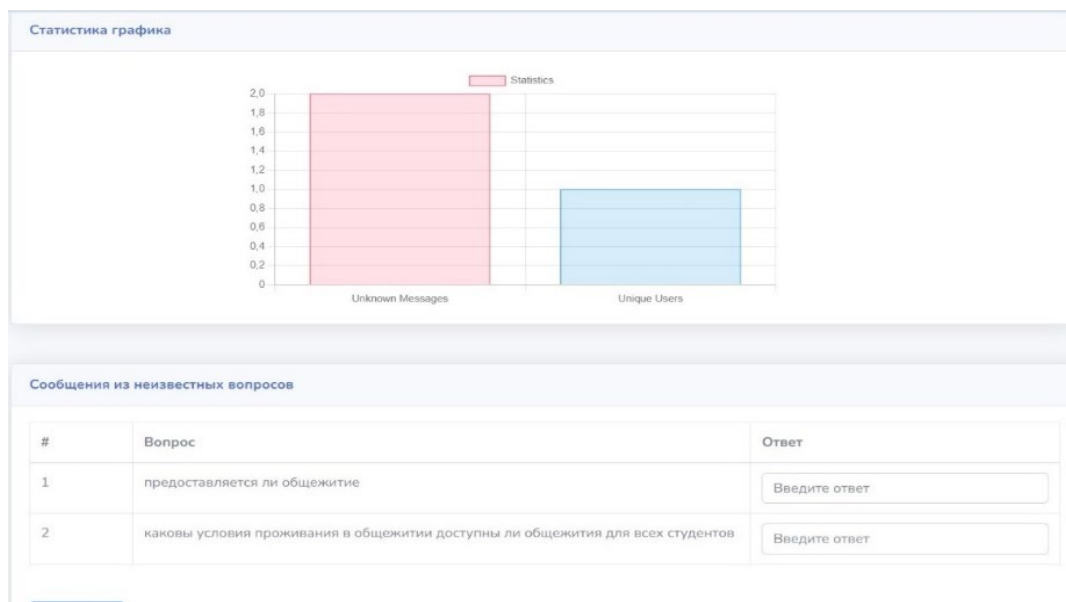


Рис. 2. Просмотр статистики обучения чат-ботов

Быстродействие и точность универсальной платформы при увеличении объема получаемой информации достигается благодаря обмену данными через API (рис. 3).

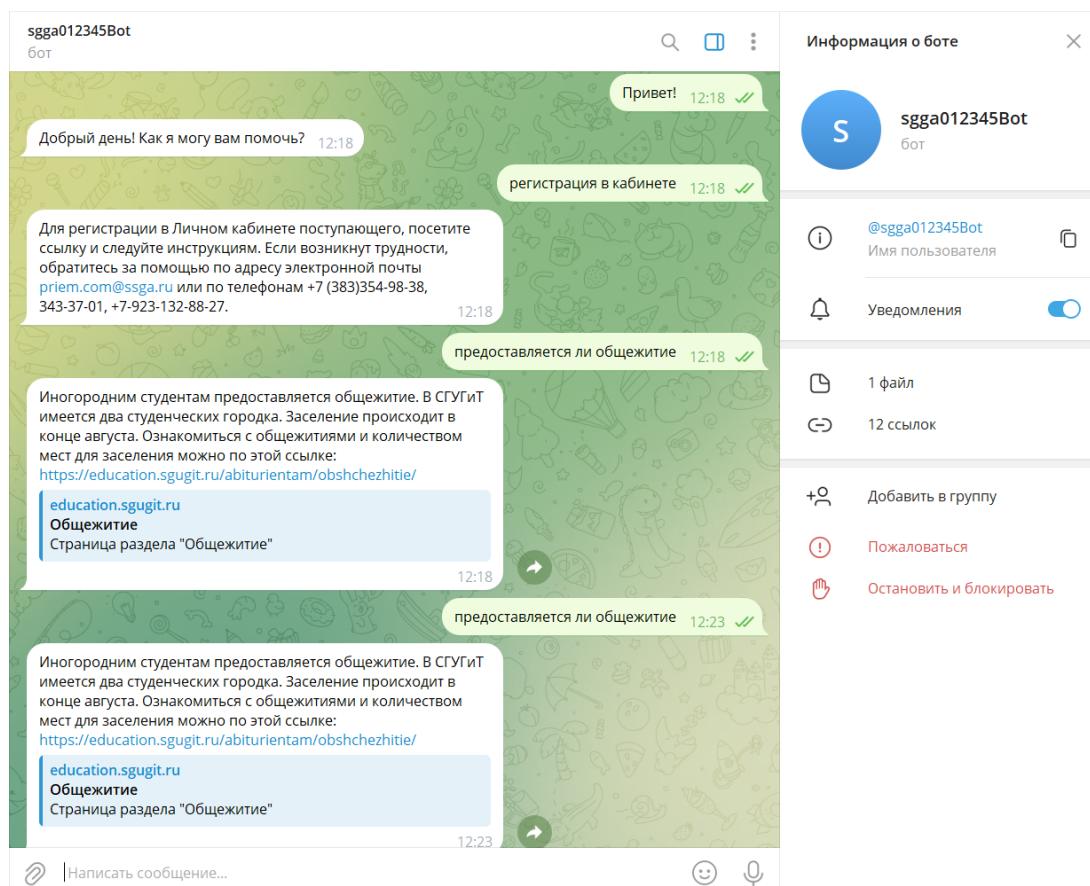


Рис. 3. Отражение работоспособности бота на обученных данных

Экспорт обучающих и аналитических данных в базу SQLite предоставляет возможность постоянно обновлять модели и следить за производительностью чат-ботов. Все это делает систему гибкой и приспособленной под различные условия эксплуатации [10].

Таким образом, разработанная мультиагентная архитектура универсальной платформы для автоматизированного обучения чат-ботов подтверждает свою пригодность посредством:

- унифицированного обмена данными между агентами, способствующего быстрой обработке запросов и высокой точности;
- действенного сбережения обучающих и аналитических данных в базе данных для увеличения качества обслуживания пользователей;
- автоматизированного конвертирования входных данных (из вида txt в вид csv) с последующим упорядочиванием в виде JSON.

Заключение

По итогу проведенного исследования разработана мультиагентная архитектура универсальной платформы, позволяющая осуществлять автоматизированное обучение чат-ботов на основе машинного обучения. Полученные результаты демонстрируют, что распределение заданий между специализированными агентами не только снижает нагрузку на платформу, но и повышает ее стабильность и масштабируемость.

Таким образом, разработанная многоагентная архитектура способствует решению проблемы высокой стоимости и неэффективности традиционных методов обслуживания, а также создает устойчивую систему, поддерживающую быструю и надежную обработку запросов в различных бизнес-средах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Басаргин, А.А. Методы искусственного интеллекта : учебное пособие / А. А. Басаргин. – Новосибирск: СГУГиТ, 2022. – 195 с. – Текст: непосредственный.
2. Бугакова, Т. Ю. Моделирование систем : учебное пособие / Т. Ю. Бугакова, П. Ю. Бугаков. – Новосибирск : СГУГиТ, 2020. – 95 с. – ISBN 978-5-907320-58-1. – Текст: непосредственный.
3. Гришин Р. В. Чат-боты как способ взаимодействия вуза со студентами / Р. В. Гришин. – Новосибирск : Актуальные вопросы образования. – 2022. – № 2. – С. 240-243. – EDN GIOEAW. – Текст : электронный. – Режим доступа: <https://sgugit.ru/upload/science-and-innovations/conference-ssga/international-scientific-methodical-conference-actual-problems-of-education/collections-of-materials-nmk-2022/part2/240-243.pdf>.
4. Петренко, А. Ю. Теория и практика создания обучающих экспертных систем : учебное пособие / А. Ю. Петренко, А. А. Чайковский. – Москва : Наука, 2018. – 312 с. – ISBN 978-5-02-037121-2. – Текст : непосредственный.
5. Петрунина, И. Н. Особенности экспертных систем и оправданность их применения в реальных проектах : учебное пособие / И. Н. Петрунина. – Москва : Интернаука, 2017. – 155 с. – Текст : непосредственный.
6. Гаскаров, Д. В. Интеллектуальные информационные системы : учебник для вузов / Д. В. Гаскаров. – Москва : Высшая школа, 2003. – 431 с. – Текст: непосредственный.

7. Зюзьков, В. М. Искусственный интеллект и экспертные системы : учебное пособие / В. М. Зюзьков. – Нижний Новгород : ННГАСУ, 2018. – 220 с. – ISBN 978-5-902482-86-7. – Текст : непосредственный.
8. Крылов, И. В. Основы искусственного интеллекта: экспертные системы и нейронные сети : учебное пособие / И. В. Крылов, В. Н. Смирнов. – Екатеринбург : УрФУ, 2020. – 300 с. – ISBN 978-5-7996-2757-8. – Текст : непосредственный.
9. Ефимов, А. А. Технологии разработки нейросетевых экспертных систем : учебное пособие / А. А. Ефимов, В. Н. Грачев. – Москва : Издательство МГТУ, 2021. – 185 с. – ISBN 978-5-94010-396-7. – Текст : непосредственный.
10. Ковалев А. В., Золотов В. С. Применение машинного обучения в экспертных системах: учебное пособие / А. В. Ковалев, В. С. Золотов. – Самара: СамГТУ, 2020. – 228 с. – ISBN 978-5-93424-851-3. – Текст: непосредственный.

© Д. А. Ваулин, Р. В. Гришин, 2026

Е. В. Ващенко^{1✉}, Р. В. Гришин¹

Разработка системы визуального контроля и классификации дефектов на основе технологий машинного обучения

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: Vaschenko-EV2022@sgugit.ru

Аннотация. В статье представлены результаты разработки мобильной системы визуального контроля и классификации производственных дефектов на основе технологий машинного обучения. Рассмотрены особенности автоматизации процессов контроля качества продукции на производстве, а также актуальность применения интеллектуальных методов анализа изображений для повышения точности обнаружения дефектов. Выполнен анализ существующих решений в области компьютерного зрения и мобильных систем мониторинга качества продукции. Разработано мобильное приложение Defect Control, обеспечивающее фиксацию производственных партий, визуальный анализ дефектов по изображениям, автоматическую классификацию выявленных отклонений и формирование статистических отчетов. В качестве технологической основы использованы современные средства веб- и мобильной разработки, а также инструменты интеграции моделей машинного обучения. Проведено тестирование разработанной системы на наборе производственных изображений с различными типами дефектов. Полученные результаты демонстрируют возможность повышения скорости первичного контроля качества и снижения нагрузки на операторов производственного контроля.

Ключевые слова: машинное обучение, компьютерное зрение, контроль качества, мобильное приложение, классификация дефектов, визуальный контроль

E. V. Vashchenko^{1✉}, R. V. Grishin¹

Development of a mobile system for visual inspection and defect classification based on machine learning technologies

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: Vaschenko-EV2022@sgugit.ru

Abstract. The article presents the results of developing a mobile system for visual inspection and classification of production defects based on machine learning technologies. The paper considers the automation of product quality control processes and the relevance of intelligent image analysis methods for improving defect detection accuracy. The Defect Control mobile application was developed to provide production batch tracking, visual defect analysis, automatic classification of detected defects and generation of statistical reports. Testing was carried out on a dataset of production images containing various defect types. The obtained results demonstrate the possibility of increasing the speed of quality inspection and reducing the workload on production quality control operators.

Keywords: machine learning, computer vision, quality control, mobile application, defect classification, visual inspection

Введение

Контроль качества в производстве редко сводится к одному осмотру. Специалист проверяет изделие, сравнивает результат с нормой и фиксирует решение. Затем эти сведения должны остаться в системе. Иначе спорную партию трудно восстановить по истории.

Визуальный контроль особенно чувствителен к человеческому фактору. Усталость, освещение и качество снимка меняют итоговую оценку. Для поверхностных дефектов это заметно сильнее. Царапина может быть тонкой, трещина – короткой, а пятно – слабоконтрастным. Методы цифровой обработки изображений и компьютерного зрения помогают формализовать такие признаки [1, 2].

Технологии машинного обучения дали этой задаче новый уровень. Модель учится на изображениях, а не только на заранее заданных правилах. Поэтому она лучше переносит разные формы дефектов. Подходы глубокого обучения активно применяются для распознавания образов и анализа сложных изображений [3–6].

Но одной модели мало. Производственный контроль требует учета партий, ролей и времени проверки. Нужна система, где снимок связан с партией, контролером и результатом. Без этой связи машинная оценка остается отдельным экспериментом, а не частью рабочего процесса. Общие принципы интеллектуальных систем подтверждают важность такой связки данных и алгоритмов [5].

Цель исследования – разработать мобильную систему визуального контроля и классификации дефектов. Система должна поддерживать производственный сценарий. Пользователь создает партию, фиксирует дефект, прикрепляет изображение и получает предварительный результат анализа. Что дает такая схема на практике? Она сокращает разрыв между осмотром изделия и цифровой записью результата.

Методы и материалы

В рамках исследования была разработана система Defect Control, предназначенная для сопровождения процессов производственного контроля. Архитектура системы включает клиентскую часть мобильного приложения, серверный модуль обработки данных и модуль анализа изображений.

При разработке приложения использовались современные технологии веб- и мобильной разработки. Мобильный интерфейс реализован на React и TypeScript [7, 8]. Серверная часть построена на Node.js и Express [9, 10]. Для хранения данных использована PostgreSQL, а для развертывания учитывалась контейнеризация Docker [11–13].

Для анализа изображений использовались инструменты машинного обучения и технологии компьютерного зрения [14].

Разработанная система обеспечивает выполнение следующих функций:

- учет и отслеживание производственных партий;
- автоматическое определение смен;
- загрузку и обработку изображений продукции;
- автоматическую классификацию дефектов;

- формирование статистики и аналитических отчетов;
- визуализацию результатов контроля качества.

Принцип работы системы основан на последовательной обработке изображения. Пользователь загружает фотографию изделия в мобильное приложение, после чего изображение передается в модуль анализа. Далее выполняется выделение признаков дефекта и классификация изображения по заданным категориям.

Для оценки эффективности системы было проведено тестирование на экспериментальном наборе изображений, содержащем поверхностные трещины, сколы, царапины, деформации поверхности и загрязнения.

Результаты

В результате исследования было разработано мобильное приложение Defect Control, обеспечивающее автоматизацию процессов визуального контроля продукции.

Главный экран приложения содержит информацию о текущих производственных партиях, статусе проверки и распределении данных по сменам. Пользователь может оперативно получить сведения о количестве проверенных изделий и выявленных дефектах. Интерфейс основных экранов разработанного мобильного приложения представлен на рис. 1.

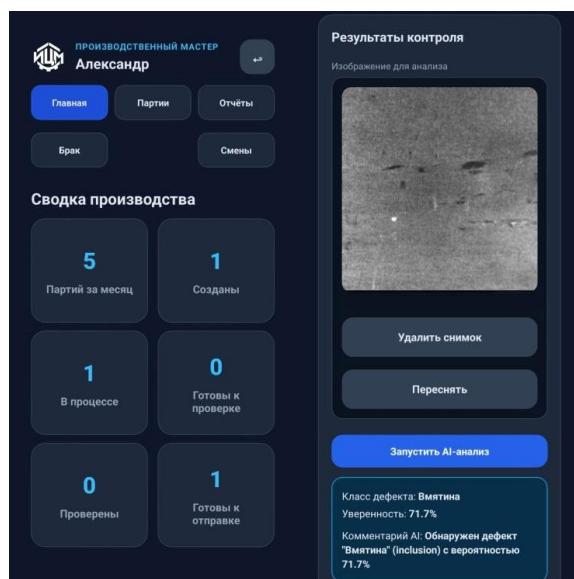


Рис. 1. Интерфейс мобильного приложения Defect Control
а) главный экран; б) анализ изображения

В приложении реализован механизм загрузки изображений продукции с мобильного устройства. После загрузки выполняется автоматический анализ изображения и определение предполагаемого типа дефекта. Результаты анализа отображаются пользователю в виде текстового описания и статистических показателей [15, 16, 17].

Отдельный результат связан с серверным хранением изображений. Файл загружается не в локальную память приложения, а на сервер. В отчет попадает ссылка. Поэтому фотографию видит не только контролер, который ее добавил, но и производственный мастер.

Проверка работоспособности проводилась на наборе из 500 изображений. В набор входили разные типы поверхностных дефектов. Для оценки использовались точность классификации и среднее время обработки. Сводные результаты приведены в табл. 1.

Таблица 1. Результаты тестирования системы визуального контроля

Показатель	Значение
Количество изображений	500
Средняя точность классификации	92,4 %
Среднее время обработки изображения	1,8 с
Доля корректного определения трещин	94,1 %
Доля корректного определения сколов	91,3 %
Доля корректного определения царапин	90,8 %

Полученные значения показывают, что система подходит для первичного отбора дефектных случаев. Средняя точность 92,4% не означает полной замены контроля. Она показывает, что модель может быть использована как подсказка при работе с изображением.

Среднее время обработки 1,8 с важно для мобильного сценария. Пользователь получает ответ почти сразу. Это снижает паузу между фотографированием изделия и занесением результата в отчет.

Обсуждение

Разработанная система отличается от простого классификатора изображений. Она включает производственный контекст. Для предприятия важен не только класс дефекта. Важны партия, смена, исполнитель, контролер и история изменения статуса.

Мобильный формат дает практическое преимущество. Контроль можно выполнить на участке, рядом с изделием. Оператору не нужно переносить сведения в отдельный файл. Меньше ручных операций – меньше случайных ошибок.

Серверная часть решает другую проблему. Она делает данные общими для пользователей. Если снимок сохранен только на телефоне, отчет теряет ценность. Если снимок хранится на сервере, его можно проверить позже. Это особенно важно для спорных дефектов [18, 19].

Есть и ограничения. Качество классификации зависит от освещения и резкости снимка. Сложные комбинированные дефекты могут распознаваться хуже. Кроме того, модель обучается на конечном наборе примеров. Чем беднее набор, тем осторожнее нужно относиться к результату.

Дальнейшее развитие системы связано с расширением датасета. В него стоит добавлять реальные фотографии с производственного участка. Тогда модель будет видеть больше вариантов поверхности, шума и масштаба. Еще одно направление – аналитика по партиям, сменам и видам брака [20].

Заключение

В ходе работы разработана мобильная система Defect Control. Она предназначена для визуального контроля и классификации производственных дефектов.

Система объединяет мобильное приложение, серверный модуль, базу PostgreSQL и анализ изображений. Пользователь может создать партию, провести контроль, загрузить фотографию и получить машинную оценку.

Проверка на наборе из 500 изображений показала среднюю точность 92,4 %. Среднее время обработки составило 1,8 с. Эти показатели подтверждают пригодность прототипа для первичного анализа дефектов.

Практическая ценность результата состоит в цифровой фиксации контроля. Данные о партии, дефекте и снимке хранятся вместе. Это повышает прослеживаемость и упрощает последующий анализ качества продукции.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – Москва : Техносфера, 2012. – 1104 с.
2. Szeliski R. Computer Vision: Algorithms and Applications. – London : Springer, 2022. – 979 p.
3. Бринк Х., Ричардс Дж. Машинное обучение. – Санкт-Петербург : Питер, 2020. – 336 с.
4. Кузнецов А. В. Методы компьютерного зрения в промышленности. – Москва : Инфра-М, 2022. – 290 с.
5. Goodfellow I., Bengio Y., Courville A. Deep Learning. – Cambridge : MIT Press, 2016. – 775 p.
6. LeCun Y., Bengio Y., Hinton G. Deep learning // Nature. – 2015. – Vol. 521. – P. 436–444.
7. React Native Documentation : сайт. – URL: <https://reactnative.dev/docs/getting-started> (дата обращения: 05.04.2026). – Текст : электронный.
8. TypeScript Documentation : сайт. – URL: <https://www.typescriptlang.org/docs/> (дата обращения: 05.04.2026). – Текст : электронный.
9. Node.js Documentation : сайт. – URL: <https://nodejs.org/docs/latest/api/> (дата обращения: 05.04.2026). – Текст : электронный.
10. Express Documentation : сайт. – URL: <https://expressjs.com/> (дата обращения: 05.04.2026). – Текст : электронный.
11. PostgreSQL Documentation : сайт. – URL: <https://www.postgresql.org/docs/> (дата обращения: 05.04.2026). – Текст : электронный.
12. Docker Documentation : сайт. – URL: <https://docs.docker.com/> (дата обращения: 05.04.2026). – Текст : электронный.
13. Axios Documentation : сайт. – URL: <https://axios-http.com/docs/intro> (дата обращения: 05.04.2026). – Текст : электронный.
14. Google Cloud Vision API Documentation : сайт. – URL: <https://cloud.google.com/vision/docs> (дата обращения: 05.04.2026). – Текст : электронный.
15. Советов Б. Я., Цехановский В. В. Информационные технологии. – Москва : Юрайт, 2023. – 327 с.

16. Грекул В. И., Денищенко Г. Н., Коровкина Н. Л. Проектирование информационных систем. – Москва : Юрайт, 2022. – 385 с.
17. Кузин А. В., Левонисова С. В. Базы данных. – Москва : Академия, 2021. – 320 с.
18. Фаулер М. Архитектура корпоративных программных приложений. – Москва : Вильямс, 2019. – 544 с.
19. Соммервилл И. Инженерия программного обеспечения. – Москва : Вильямс, 2020. – 864 с.
20. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. – Pearson, 2021. – 1136 p.

© Е. В. Ващенко, Р. В. Гришин, 2026

Е. Д. Востокова^{1✉}, Н. С. Головачев¹

Разработка подсистемы управления сценариями и процессами вовлечения пользователей в игровом приложении «BlushBound»

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: liza.vost@mail.ru

Аннотация. В статье представлены результаты разработки подсистемы управления сценариями и процессами вовлечения пользователей в игровом приложении «BlushBound». Выполнен анализ предметной области современных игровых систем с процедурной генерацией уровней, вероятностными механизмами получения наград и интеграцией интерактивного повествования. Разработка выполнена на платформе Unity с использованием языка программирования C#. Спроектирована модульная архитектура, обеспечивающая централизованное управление состояниями игры, генерацию уровней, поведение персонажей и противников, а также динамическую активацию сценарных событий. Проведено тестирование и выполнено экономическое обоснование проекта.

Ключевые слова: игровое приложение, Unity, C#, подсистема управления сценариями, процедурная генерация, вовлечение пользователей

E. D. Vostokova^{1✉}, N. S. Golovachev¹

Development of a scenario and user engagement management subsystem in the BlushBound game application

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: liza.vost@mail.ru

Abstract. The article presents the development results of a scenario and user engagement management subsystem for the BlushBound game application. The subject area of modern game systems with procedural level generation, probabilistic reward mechanics, and interactive storytelling integration is analyzed. The subsystem was implemented using the Unity platform and C#. A modular architecture was designed to provide centralized game state management, level generation, character and enemy control, and dynamic story activation. Testing and economic feasibility analysis were conducted.

Keywords: game application, Unity, C#, scenario management subsystem, procedural generation, user engagement

Введение

Современная игровая индустрия характеризуется высокой конкуренцией и быстрым развитием технологий. Пользователь ожидает не только качественную игровую механику, но и продуманную систему прогрессии, нарративное сопровождение и устойчивую мотивацию к продолжению игры.

Игровые проекты все чаще объединяют процедурную генерацию уровней, систему случайных наград и элементы интерактивного повествования. Однако в

ряде решений данные компоненты реализуются отдельно, что снижает уровень вовлеченности и удержания аудитории.

Актуальность работы обусловлена необходимостью создания подсистемы, объединяющей генерацию уровней, систему наград и сценарные события в единую архитектуру.

Цель работы – разработка подсистемы управления сценариями и процессами вовлечения пользователей в игровом приложении «BlushBound».

Исследование предметной области

Процедурная генерация уровней позволяет формировать уникальные игровые пространства при каждом прохождении. Алгоритмы генерации основаны на заданных правилах построения локаций и размещения объектов [1]. Данный подход повышает реиграбельность и снижает затраты на создание статического контента.

Вовлечение пользователей обеспечивается сочетанием прогрессии, системы наград и эмоционального контента [2]. Вероятностная модель получения персонажей стимулирует повторные игровые сессии, создавая эффект ожидания результата. При этом важен баланс вероятностей и постепенное усложнение игрового процесса.

Интеграция нарратива в игровой цикл усиливает эмоциональную связь пользователя с продуктом. Согласно подходу MDA (Mechanics–Dynamics–Aesthetics), игровые механики должны быть согласованы с динамикой и эмоциональным восприятием [3].

Анализ существующих решений показал, что редко реализуется централизованное управление всеми перечисленными механизмами. В проекте «BlushBound» предложена архитектура, объединяющая данные элементы.

Методы и технологии

Разработка выполнена в среде Unity с использованием языка C# [1, 4]. Архитектура построена по модульному принципу с применением принципов объектно-ориентированного проектирования и паттернов программирования [5, 6].

Основные модули системы:

- менеджер состояния игры;
- модуль генерации уровней;
- модуль управления персонажами;
- модуль искусственного интеллекта противников;
- модуль сценарного управления;
- система наград.

Хранение конфигурационных данных реализовано с помощью ScriptableObject [1], что позволяет отделить параметры игровых объектов от логики выполнения.

Поведение персонажа построено на основе конечного автомата состояний: ожидание, перемещение, атака, применение способности, получение урона.

Противники реализуют алгоритм обнаружения игрока по радиусу и переходят в режим преследования и атаки. Для боссов предусмотрена фазовая модель поведения: при снижении здоровья ниже порогового значения изменяются параметры атаки и скорости.

Сценарные события представлены структурированными объектами, содержащими текст, условия активации и ссылки на последующие события. Активация происходит через систему триггеров, связанных с прогрессом пользователя.

Архитектура и программные компоненты подсистемы

Для обеспечения масштабируемости, удобства тестирования и дальнейшего расширения функционала подсистема управления сценариями и процессами вовлечения реализована на основе модульной архитектуры. Взаимодействие между компонентами построено по событийно-управляемой модели, что минимизирует жесткие зависимости и упрощает отладку игровых процессов.

В состав подсистемы входят ключевые программные компоненты.

Менеджер состояния игры (GameManager), отвечающий за инициализацию сцен, загрузку уровней, сохранение и загрузку прогресса, а также синхронизацию работы всех модулей. Обеспечивает корректный переход между игровыми состояниями и обрабатывает глобальные события.

Модуль процедурной генерации уровней (LevelGenerator), формирующий структуру локации на основе заданных параметров (биом, текущая сложность, этап сюжета), размещает противников, интерактивные объекты и триггерные зоны. Генерация ограничивается правилами проходимости и сбалансированности, что исключает формирование непроходимых или избыточно сложных конфигураций.

Модуль управления персонажами и ИИ противников реализующий логику поведения на основе конечных автоматов состояний. Для игрового персонажа обрабатываются ввод пользователя, перемещение, расчет зон поражения и активация способностей. Для противников реализованы алгоритмы обнаружения цели, преследования, атаки и перезарядки. Боссовые сущности поддерживают фазовую модель поведения: при снижении здоровья ниже порогового значения изменяются параметры урона, скорости и набор доступных атак.

Сценарный движок (ScenarioEngine), хранящий структуру диалогов, событий и условий их активации. Интегрируется с игровым циклом через систему триггеров: сюжетные фрагменты запускаются при завершении уровня, победе над ключевыми противниками или получении персонажей определенной редкости. Обеспечивает приостановку игрового процесса на время отображения диалоговых окон и сохранение ветвления истории.

Система наград и прогрессии (RewardSystem), управляющий выдачей внутриигровых ресурсов, рассчитывает вероятности получения персонажей разной редкости, реализует механизмы гарантированного дропа (pity system) и отслеживает выполнение условий разблокировки новых локаций и контента.

Конфигурационные параметры игровых объектов (характеристики, редкость, условия активации сценариев, диапазоны генерации) отделены от программной

логики и хранятся в структуре ScriptableObject платформы Unity. Такой подход позволяет настраивать баланс, добавлять новый контент и корректировать сценарные ветвления без изменения исходного кода, что существенно ускоряет итерации разработки.

Взаимосвязь основных компонентов подсистемы и потоки данных между ними представлены на структурной схеме (рис. 1).

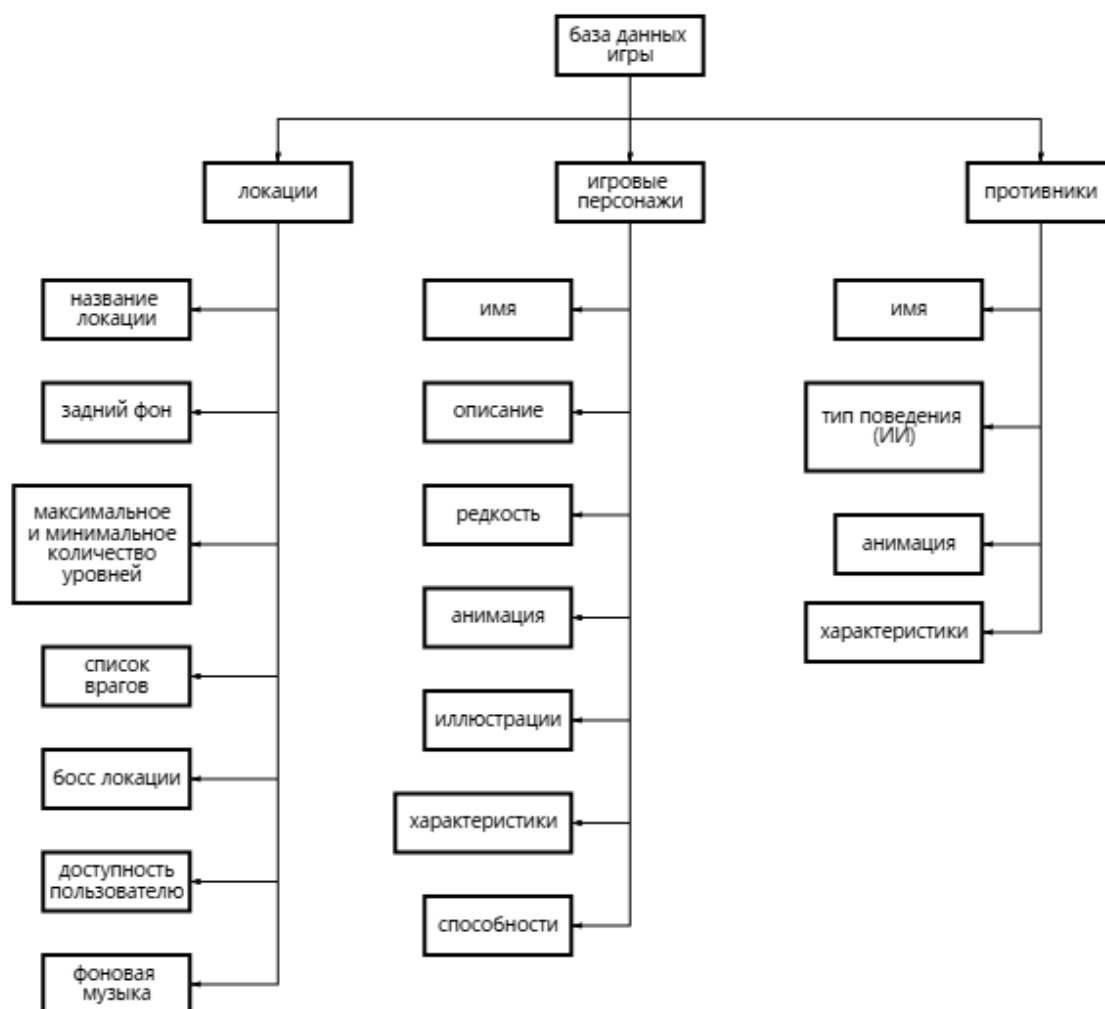


Рис. 1. Структура хранения компонентов подсистемы управления сценариями

Предложенная архитектура обеспечивает высокую степень связности игровых механик и нарративных элементов, позволяя динамически адаптировать игровой процесс под текущий прогресс пользователя и поддерживать устойчивый цикл вовлечения на протяжении всего жизненного цикла приложения.

Результаты

В ходе разработки реализованы следующие функциональные возможности:

- при загрузке сцены формируется структура комнат и размещаются противники в допустимых координатах;

- игрок выбирает персонажей различной редкости, влияющей на характеристики и стиль прохождения;
- боевая система с расчетом урона, осуществляющимся с учетом характеристик атакующего и защитных параметров цели;
- активные способности с механизмом перезарядки;
- поведение босса локации изменяется в зависимости от текущего уровня здоровья.
- сюжетные фрагменты активируются при прохождении уровней, победе над боссами и получении редких персонажей.

Проведено функциональное тестирование, включающее проверку корректности расчета урона, переходов между сценами и поведения противников. Критических ошибок не выявлено. Производительность соответствует требованиям платформы Unity [1].

Экономическое обоснование выполнено для базового сценария: 20 000 установок, конверсия 3,5 %, ARPPU 800 рублей и рекламный доход 25 рублей на пользователя. Совокупная выручка составляет 1 060 000 рублей при затратах 997 600 рублей. Точка окупаемости достигается при 18 823 установках.

Заключение

Разработана подсистема управления сценариями и процессами вовлечения пользователей для игрового приложения «BlushBound». Реализована модульная архитектура, обеспечивающая интеграцию процедурной генерации уровней, системы наград и сценарных механизмов.

Тестирование подтвердило стабильность функционирования системы. Экономические расчеты демонстрируют потенциальную самоокупаемость проекта в течение первого года эксплуатации.

Предложенные решения могут быть использованы при разработке аналогичных игровых приложений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Unity Technologies. Unity User Manual 2022.3 (LTS). – San Francisco : Unity Technologies, 2023.
2. Adams E. Fundamentals of Game Design. – 3rd ed. – New Riders, 2014.
3. Hunicke R., LeBlanc M., Zubek R. MDA: A Formal Approach to Game Design and Game Research // Proceedings of the AAAI Workshop on Challenges in Game AI. – 2004.
4. Рабинович С. Разработка 2D-игр на Unity. – Санкт-Петербург : Питер, 2022. – 448 с.
5. Ларошель Д. Паттерны проектирования игр на C#. – Москва : БХВ-Петербург, 2020. – 496 с.
6. Фримен Э., Робсон Э. Паттерны архитектуры программных систем. – Санкт-Петербург : Питер, 2019. – 656 с.

© Е. Д. Востокова, Н. С. Головачев, 2026

Е. И. Вяткина^{1✉}, Д. Ю. Смирнов¹

Разработка конфигурации и интерактивного дашборда для анализа проектов с использованием технологии искусственного интеллекта в 1С: Предприятие 8.3

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: vatkinakata67@gmail.com

Аннотация. В статье представлены результаты разработки конфигурации 1С: Предприятие и интерактивного веб-дашборда для анализа финансовых показателей проектов инжиниринговой компании. Типовые конфигурации зачастую не обеспечивают полноценный проектный учет и удобную интерактивную визуализацию данных. Предложен подход, сочетающий разработку прикладной конфигурации (справочники, документы, реквизиты), внешнего отчета на базе системы компоновки данных (СКД), загрузки агрегированных данных во внешний JavaScript-файл и автономного веб-дашборда на HTML5/CSS3/JavaScript с визуализацией на Chart.js. В рамках исследования детально раскрыт процесс разработки модуля предиктивной аналитики на базе рекуррентной нейронной сети LSTM с использованием стека Python и библиотек TensorFlow/Keras. Особое внимание уделено механизмам интеграции полученных прогнозов в интерфейс интерактивного дашборда, а также представлены итоги предварительного функционального тестирования основных сценариев эксплуатации. Ценность данной разработки заключается в практической применимости для инжиниринговых организаций, где внедрение решения способствует повышению прозрачности проектной экономики и обоснованности управленческих решений.

Ключевые слова: 1С: Предприятие, LSTM, прогнозирование, веб-дашборд, интерактивная визуализация

E. I. Vyatkina^{1✉}, D. Yu. Smirnov¹

Development of configuration and interactive dashboard for project analysis using artificial intelligence technology in 1C: Enterprise 8.3

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: vatkinakata67@gmail.com

Abstract. The article presents the outcomes of developing a 1C:Enterprise configuration and an interactive web dashboard for analyzing the financial performance of projects in an engineering company. Typical configurations often do not provide full-fledged project accounting and convenient interactive data visualization. The paper proposes an approach that combines the development of an applied configuration (catalogs, documents, attributes), an external report based on the Data Composition System (DCS), export of aggregated data to an external JavaScript file, and a standalone web dashboard built with HTML5/CSS3/JavaScript with visualization using Chart.js. As part of the study, the development of a predictive analytics module based on a Long Short-Term Memory (LSTM) recurrent neural network using Python and the TensorFlow/Keras libraries is described in detail. Particular attention is paid to the mechanisms for integrating the resulting forecasts into the interactive dashboard interface, and the results of preliminary functional testing of the main operational scenarios are presented. The value of this development lies in its practical applicability for engineering

organizations, where its implementation can improve transparency of project economics and strengthen the evidence base for management decisions.

Keywords: 1C:Enterprise, LSTM, forecasting, web dashboard, interactive visualization

Введение

Современные инжиниринговые компании формируют значительный объем учетных данных, связанных с реализацией услуг, закупками, трудозатратами и распределением затрат. При этом потребности в управленческом анализе смещаются от статичных регламентированных отчетов к динамическим агрегатам и срезам, позволяющим оперативно оценивать эффективность по проектам, временным отрезкам и ответственным сотрудникам. На практике такой анализ упрощается, если первичные операции изначально «привязаны» к проекту, а итоги могут быть быстро собраны, отфильтрованы и представлены в интерактивной форме.

Платформа «1С:Предприятие 8.3» является стандартом для автоматизации бухгалтерского и управленческого учета, однако типовые конфигурации часто не обеспечивают достаточного уровня проектной аналитики и интерактивной визуализации без значительных финансовых вложений в доработку или внедрение сложных продуктовых решений. В результате на предприятии возникает разрыв между учетом и аналитикой: данные есть, но их преобразование в удобные для руководства показатели занимает время и зависит от ручных процедур (экспорт, сводные таблицы, повторяющиеся операции).

Цель работы – разработка конфигурации «1С:Предприятие 8.3» и интерактивного веб-дашборда для анализа проектной деятельности инжиниринговой компании, включая элементы прогнозирования на основе методов искусственного интеллекта.

Задачи исследования:

1. Спроектировать структуру конфигурации и архитектуру дашборда;
2. Реализовать конфигурацию, внешний отчет, веб-дашборд и модуль ИИ-прогнозирования;
3. Обучить модель и интегрировать прогноз в дашборд;
4. Провести предварительное тестирование и устранить выявленные дефекты.

Объект исследования – процессы учета и анализа проектной деятельности на предприятии. Предмет – программные средства и методы интеграции данных «1С: Предприятие» с веб-визуализацией и прогнозированием временных рядов финансовых показателей.

Исследование предметной области

Проектный финансовый анализ в учетных системах обычно опирается на связку «документ → движения регистров → отчет». Для корректной аналитики в разрезе проектов необходимы, как минимум: единый классификатор проектов, обязательная (или управляемая) связь операций с проектом, согласованные

правила отнесения доходов и расходов, а также удобные механизмы агрегации за период.

На рынке существует несколько классов решений, которые частично закрывают потребность в аналитике, но имеют ограничения для малого и среднего бизнеса либо для «тонкой» кастомизации под конкретную модель проектов.

Ниже приведено обобщенное сопоставление подходов (по критериям: способ получения данных, настраиваемость, автоматизация, интерфейс, стоимость, замечания).

Таблица 1. Анализ предметной области

Аналог	Способ получения данных	Настраиваемость	Автоматизация	Отображение (UI)	Стоимость / доступность	Примечание
1С: Управление холдингом	Встроенные отчеты	Средняя	Частичная	Типовой интерфейс 1С	Платная (от 200 000 Р)	Тяжеловесное решение, избыточное для малого бизнеса
1С: ERP	Встроенные отчеты	Низкая	Частичная	Типовой интерфейс 1С	Платная (от 500 000 Р)	Требует внедрения под ключ, сложно настраивать
1С:Бухгалтерия (типовая конфигурация)	Стандартные отчеты	Низкая	Нет	Типовой интерфейс 1С	Входит в лицензию 1С	Нет проектной аналитики, нет привязки расходов к проектам
Microsoft Power BI + 1С	Выгрузка Excel/CSV	Высокая	Требуется настройка	Интерактивные дашборды	Бесплатный уровень ограничен	Требуются навыки DAX, сложная интеграция
Tableau + 1С	Выгрузка Excel/CSV	Высокая	Требуется настройка	Профессиональная визуализация	Платная (от \$70/мес)	Дорогое решение для малого бизнеса
Самодельный Excel-отчет	Ручной ввод/выгрузка из 1С	Высокая	Нет	Простое отображение	Бесплатно	Трудоемко, высокий риск ошибок
Реализуемое решение	JSON-выгрузка из 1С	Высокая	Полная	Веб-дашборд (Chart.js)	Бесплатно, open-source	Индивидуальное решение с ИИ-прогнозом

Из сравнения следует, что для компании, уже использующей «1С: Предприятие», рациональным компромиссом может стать узкоспециализированная конфигурация с минимально достаточной проектной моделью данных и легким веб-слоем, который не требует лицензирования BI-платформы, но дает интерактивность. Важным элементом является автоматизированная выгрузка данных из 1С

в формат, удобный для потребления браузером, что снижает долю ручного труда по сравнению с Excel-подходом.

Методы и технологии

Архитектурная схема выстраивается в последовательный поток: от учетных документов 1С через регистры и расчеты к внешнему отчету СКД, далее – в файл данных для дашборда, затем в веб-интерфейс и, при необходимости, в блок обновления ИИ-прогнозов с финальным отображением результатов. Подобная структура четко разграничивает зоны ответственности: 1С сохраняет статус первичного источника данных и места проведения операций, тогда как веб-панель берет на себя интерактивную визуализацию и пользовательский опыт, избегая усложнения встроенного интерфейса конфигурации.

В прикладной части платформы 1С реализуется проектная модель данных, опирающаяся на справочники, документы и реквизиты. Согласно техническому заданию, ключевые процессы привязаны к реквизиту «Проект» в таких документах, как «Реализация», «Поступление» и «Учет трудозатрат». Бизнес-логика написана на встроенном языке 1С и обеспечивает проведение документов, вычисление итоговых метрик и подготовку отчетности. Использование СКД позволяет гибко настраивать состав полей и правила компоновки срезов за нужный период, исключая жесткую привязку представления к конкретной форме отчета. Для группировки, агрегации и получения срезов за выбранный период используется СКД, что позволяет гибко настраивать состав полей и правила компоновки без «жесткой» привязки представления к конкретной форме отчета.

Связь учетной системы с веб-слоем осуществляется путем экспорта данных в JavaScript-файл (например, `dashboard_data.js`), который содержит структурированные показатели, доступные для загрузки веб-страницей дашборда. Данный подход характеризуется простотой развертывания и низким порогом внедрения для учебно-прикладного контура, поскольку для демонстрации функциональности достаточно статического веб-окружения и актуальной версии файла данных.

Веб-дашборд построен на HTML5 и CSS3 для разметки и оформления, а интерактивные функции реализованы на JavaScript в редакции ES6: фильтрация по руководителю и диапазону дат, обновление представлений, сортировка таблицы по заголовкам колонок, обработка отсутствия данных с выводом сообщения пользователю. Визуализация выполнена с применением Chart.js, в том числе для столбчатой диаграммы «Доходы по проектам» и сопутствующих графических представлений итоговой аналитики. Для выгрузки табличных данных в офисные форматы использован экспорт через CSV как универсальный способ подготовки файлов для последующего открытия в Excel.

Для прогнозирования динамики финансовых показателей использован подход глубокого обучения для временных рядов: LSTM (долгая краткосрочная память) в связке Python + TensorFlow/Keras. Данный метод оптимален для работы с временными рядами, так как позволяет учитывать инерционность данных и нелинейные взаимосвязи, что особенно важно при анализе ограниченных экономических выборок. Процесс обучения и валидации модели осуществляется на агрегированных

показателях, после чего полученные прогнозные значения интегрируются в интерфейс дашборда в качестве дополнительного аналитического слоя. При этом прогноз следует интерпретировать как вспомогательный инструмент сценарного анализа: их точность напрямую зависит от полноты исторических данных, стабильности бухгалтерских правил и отсутствия резких структурных изменений в бизнес-процессах компании, которые модель может не улавливать.

Результаты

В ходе реализации была разработана конфигурация «1С:Предприятие 8.3» и интерактивный веб-дашборд для анализа проектной деятельности, в которых реализованы следующие функции:

- проектная привязка операций через реквизит «Проект» в документах «Реализация», «Поступление», «Учет трудовых затрат»;
- проведение документов с формированием движений и расчетом итоговых показателей в разрезе проектов;
- внешний отчет на СКД за выбранный период с группировкой и агрегацией данных;
- выгрузка данных для дашборда в JavaScript-файл (dashboard_data.js);
- отображение на веб-странице карточек итогов (проекты, доход, расходы, прибыль), столбчатой диаграммы «Доходы по проектам», таблицы детализации с цветовой индикацией;
- фильтрация по руководителю и диапазону дат, сортировка таблицы по клику на заголовок колонки;
- экспорт табличных данных в CSV для дальнейшей работы в Excel;
- блок ИИ-прогнозирования (модель LSTM на Python/TensorFlow/Keras) и отображение прогноза на дашборде.

Предварительное тестирование показало работоспособность основного сценария: от учетных операций в 1С до обновления представления на дашборде, включая выгрузку данных, фильтрацию, сортировку, экспорт и отображение прогноза. Такой набор функций позволяет сократить ручную подготовку отчетности и повысить наглядность финансовых результатов по проектам для руководителей и специалистов, ответственных за проектную деятельность.

Заключение

В результате выполнения работы было создано программное решение для инжиниринговой компании, включающее конфигурацию «1С:Предприятие 8.3» и веб-дашборд для анализа проектов. Пользователь получает возможность фиксировать операции в учете с привязкой к проекту, получать сводные показатели через отчет СКД, автоматически передавать их в дашборд и работать с данными в интерактивном виде (фильтры, диаграммы, таблица). Дополнительно реализовано прогнозирование показателей с помощью нейросетевой модели и вывод прогноза в интерфейсе дашборда.

Все перечисленные функции направлены на решение практической задачи: упростить оперативный анализ доходов, расходов и прибыли по проектам и

снизить зависимость от «ручных» Excel-сводок при сохранении учета в привычной для организации среде 1С.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Радченко, М. Г. 1С:Предприятие 8.3. Практическое пособие разработчика / М. Г. Радченко. – М. : 1С-Паблишинг, 2020. – 964 с.
2. Чистов, Д. В. Система компоновки данных в 1С:Предприятие 8 / Д. В. Чистов. – М. : 1С-Паблишинг, 2019. – 420 с.
3. Глушаков, С. В. Разработка веб-интерфейсов на JavaScript и Chart.js / С. В. Глушаков. – М. : АСТ, 2021. – 350 с.
4. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – 2-е изд. – М. : Вильямс, 2016. – 1104 с.
5. Chollet, F. Deep Learning with Python / F. Chollet. – Shelter Island : Manning Publications, 2017. – 384 p.
6. Гудфеллоу, Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль ; пер. с англ. – М. : ДМК Пресс, 2018. – 652 с.
7. MDN Web Docs – JavaScript. – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата обращения: 15.05.2026).
8. TensorFlow Documentation. – Режим доступа: <https://www.tensorflow.org> (дата обращения: 10.05.2026).
9. Официальный сайт платформы 1С:Предприятие. – Режим доступа: <https://v8.1c.ru/> (дата обращения: 10.05.2026).
10. API справочник по встроенному языку 1С:Предприятие. – Режим доступа: <https://its.1c.ru/> (дата обращения: 10.05.2026).

© Е. И. Вяткина, Д. Ю. Смирнов, 2026

Р. Ж. Гафиятов^{1✉}, Е. Ю. Воронкин¹

Разработка веб-приложения для организации отрядной деятельности и методической поддержки вожатых в детском оздоровительном лагере

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: gafrus06@mail.ru

Аннотация. В статье рассматривается процесс разработки веб-приложения для организации отрядной деятельности и методической поддержки вожатых в детском оздоровительном лагере. Актуальность работы обусловлена необходимостью создания единой цифровой платформы, объединяющей управление организационной информацией, методическими материалами и информационным сопровождением. В качестве технологической основы выбраны Java, Spring Boot, PostgreSQL, Redis, Apache Kafka и React. Разработанная система направлена на повышение эффективности работы вожатых и администрации лагеря, а также на улучшение информационного сопровождения родителей во время смены.

Ключевые слова: веб-программирование, Java, Spring Boot, PostgreSQL, Redis, HTML, CSS, JavaScript, React JS, Apache Kafka, Api Gateway, разработка веб-приложений, база данных

R. J. Gafiyatov^{1✉}, E. Yu. Voronkin¹

Development of a web application for organizing detachment activities and methodological support for counselors in a children's health camp

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: gafrus06@mail.ru

Abstract. This article examines the development of a web application for organizing team activities and providing methodological support to camp counselors at a children's health camp. The relevance of this work stems from the need to create a unified digital platform that integrates the management of organizational information, methodological materials, and information support. Java, Spring Boot, PostgreSQL, Redis, Apache Kafka, and React were selected as the technological foundation. The developed system is aimed at increasing the efficiency of camp counselors and administration, as well as improving information support for parents during camp.

Keywords: web programming, Java, Spring Boot, PostgreSQL, Redis, HTML, CSS, JavaScript, React JS, Apache Kafka, API Gateway, web application development, database

Введение

Лагерная смена представляет собой сложный социально-педагогический процесс, в котором участвуют администрация, вожатые, дети и родители. При этом значительная часть организационной работы на практике выполняется с использованием разрозненных инструментов: мессенджеров, бумажных планов, таблиц, локальных файлов и устных распоряжений. Такой подход усложняет координацию действий, снижает прозрачность управления сменой и повышает риск потери важной информации.

Особую роль в лагере выполняет вожатый, который ежедневно организует взаимодействие с детьми и участвует в реализации общего плана смены. Родители нуждаются в удобном канале получения сведений о жизни лагеря. Поэтому возникает необходимость в создании единой цифровой платформы, ориентированной на потребности детского оздоровительного лагеря.

Целью работы – разработка веб-приложения для организации отрядной деятельности и методической поддержки вожатых в детском оздоровительном лагере.

Методы и материалы

На этапе проектирования веб-приложения были использованы методы объектно-ориентированного анализа и UML-моделирования. Применение UML позволило формализовать структуру системы, определить основные роли пользователей и описать взаимодействие ключевых компонентов приложения [1].

В системе выделены четыре базовые роли: пользователь, родитель, вожатый и администратор. Дополнительно для роли вожатого предусмотрена внутренняя иерархия подролей: вожатый, старший вожатый и медик. Поэтому сначала была построена диаграмма иерархии акторов, отражающая наследование прав и различия между категориями пользователей (рис. 1).

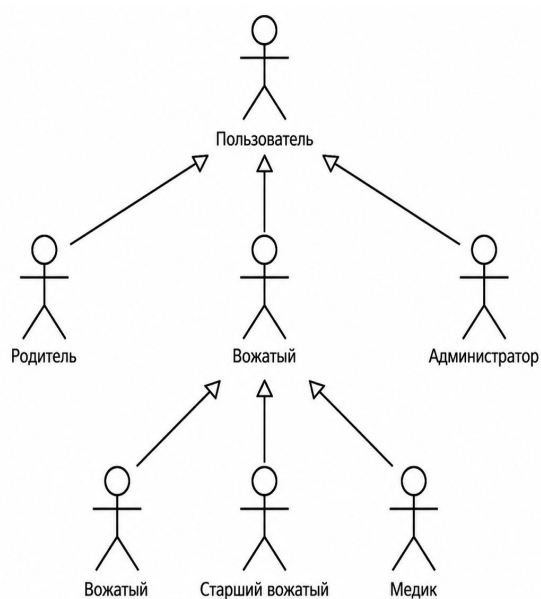


Рис. 1. Диаграмма иерархии

Архитектура веб-приложения построена на основе микросервисного подхода. Для отображения общей структуры системы была разработана схема функционирования веб-приложения, которая показывает взаимодействие клиентской части, API Gateway, микросервисов, баз данных и инфраструктурных компонентов. Такой подход позволяет обеспечить возможность дальнейшего масштабирования.

В качестве клиентской части используется React, который отвечает за отображение пользовательского интерфейса и взаимодействие пользователя с системой [3]. Все запросы от клиента поступают в API Gateway, реализованный на базе Spring Cloud Gateway. Шлюз является единой точкой входа в систему, выполняет маршрутизацию запросов к микросервисам, проверяет JWT-токены и передаёт пользовательские данные во внутренние сервисы [4].

Серверная часть приложения реализована в виде набора микросервисов. auth-service отвечает за регистрацию, аутентификацию и авторизацию пользователей. user-service хранит и обрабатывает профили пользователей. camp-service реализует основную бизнес-логику, связанную с лагерями, сменами, отрядами, детьми, заявками и назначениями сотрудников. news-feed-service отвечает за публикации, новости, медиафайлы и модерацию постов. file-storage-service обеспечивает загрузку и получение файлов через S3-совместимое хранилище. notification-service отвечает за уведомления пользователей.

В качестве основной системы хранения данных используется PostgreSQL. Каждый микросервис имеет собственную базу данных, что снижает связанность компонентов и позволяет независимо развивать отдельные части системы [5]. Для кэширования и проверки актуальности JWT-токенов используется Redis. Асинхронное взаимодействие между сервисами реализовано с помощью Apache Kafka, которая используется для передачи событий, например при регистрации пользователя или создании уведомлений [6].

Файлы и медиаобъекты хранятся в VK Cloud S3 хранилище, а в базе данных сохраняются только их метаданные: идентификатор, ссылка, тип, размер, владелец и связь с публикацией или другим объектом системы.

Разделение клиентской и серверной частей позволяет поддерживать независимость микросервисов, централизованную маршрутизацию запросов, событийное взаимодействие между сервисами и возможность дальнейшего масштабирования системы.

Результаты

В результате работы создано веб-приложение «Loom / FunKids» для организации отрядной деятельности и методической поддержки вожатых в детском оздоровительном лагере. Приложение реализует основные пользовательские сценарии для родителя, вожатого и администратора, а также обеспечивает работу с новостной лентой, карточками детей, заявками и отрядами.

В разработанном веб-приложении реализованы следующие функции:

- авторизация/регистрация пользователей с проверкой введённых данных и последующим переходом в систему;
- отображение новостной ленты лагеря с публикациями, медиафайлами и информацией для пользователей (рис. 2);
- управление отрядом, включая просмотр состава, работу с детьми и доступ к основным действиям вожатого (рис. 3);

- заполнение родителем формы заявки ребёнка на участие в смене с указанием персональных и контактных данных (рис. 4);
- просмотр карточки ребёнка с основной информацией, данными родителя и сведениями, необходимыми для работы сотрудников лагеря;
- разделение доступа к функциям приложения в зависимости от роли пользователя: родитель, вожатый или администратор;
- получение пользователями уведомлений о важных событиях в системе.

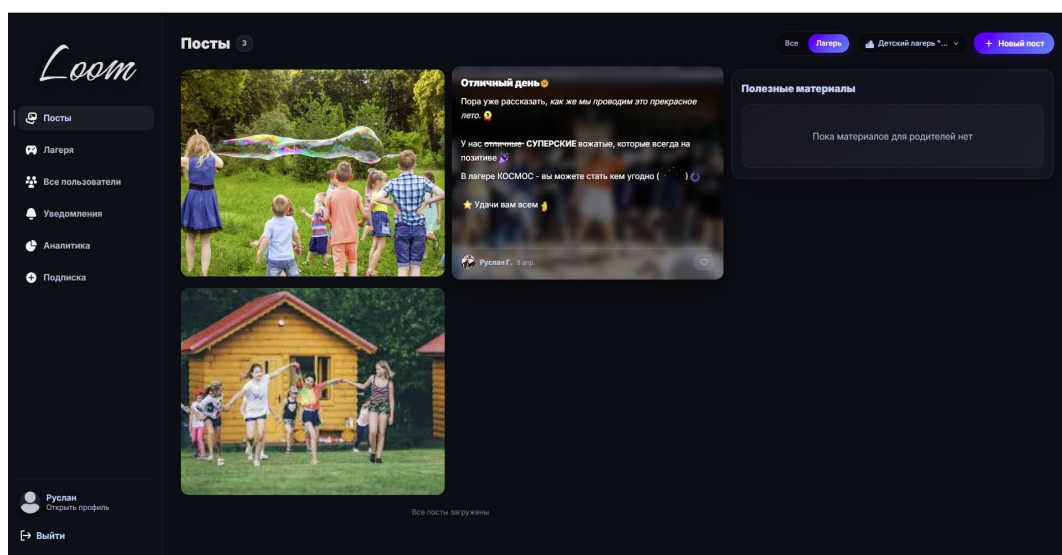


Рис. 2. Интерфейс страницы ленты новостей

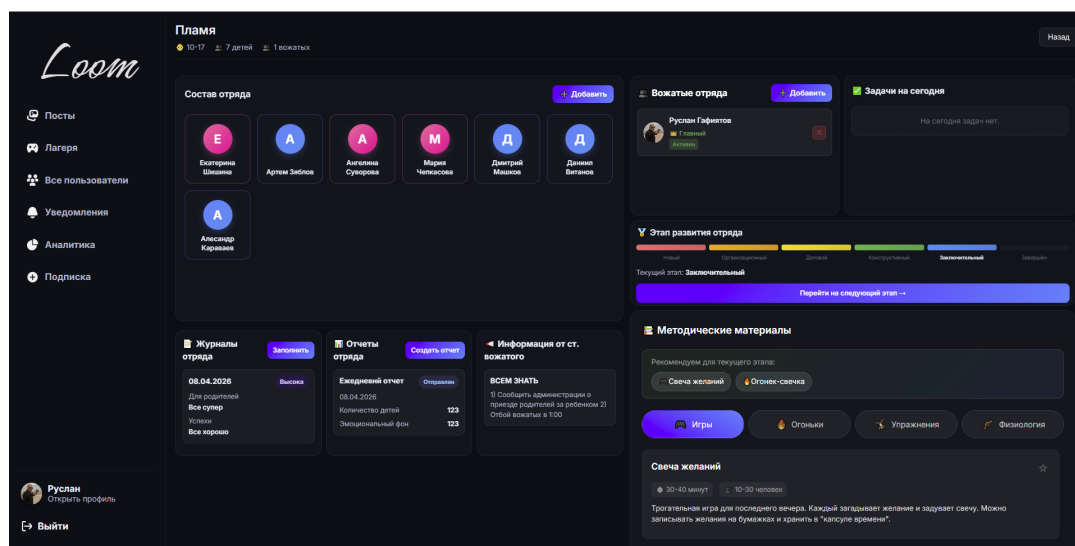


Рис. 3. Интерфейс страницы управления отрядом

Анкета ребёнка

Заполните информацию о ребёнке. Вожатый смены увидит анкету и примет решение.

Имя * Фамилия *

Руслан Гафиев

Дата рождения * Пол

10.12.2004 Мальчик

Город Кем приходиться

Новосибирск Мама

Аллергии

Нет

Медицинские особенности

Здоров

Особые потребности

Рис. 4. Интерфейс формы заявки

Нагрузочное тестирование k6 проводилось в 4 режимах: 50 пользователей в течение 6 минут, 50 пользователей в течение 31 минуты, плавный рост до 200 пользователей за 14,5 минуты и резкий скачок до 200 пользователей за 5 минут. В первых двух сценариях система работала стабильно: доля ошибок составила 2,3% и 1,9%, среднее время ответа – 399 мс и 342 мс, p95 – 1401 мс и 1223 мс. При росте до 200 пользователей ошибки увеличились до 6,8%, а при резком скачке – до 14,2%, при этом p95 остался в допустимых пределах: 2100 мс и 2400 мс. Основные ошибки возникали в news-feed-service, так как при загрузке ленты сервис обращается к другим компонентам системы. Для решения проблемы были добавлены кэширование, постраничная загрузка, сокращение межсервисных вызовов и увеличенные таймауты для операций с медиа.

Заключение

В статье рассмотрен процесс разработки веб-приложения для организации отрядной деятельности и методической поддержки вожатых в детском оздоровительном лагере. Целью проекта стало создание единой цифровой платформы, позволяющей объединить управление лагерем, сменами, отрядами, новостями, методическими материалами и взаимодействием с родителями. В отличие от разрозненного использования мессенджеров, таблиц, бумажных планов и локальных файлов, разработанная система предоставляет централизованный доступ к основным данным и инструментам лагерной смены.

Разработка велась с использованием Java, Spring Boot, PostgreSQL, Redis, Apache Kafka и React. В ходе работы были спроектированы и реализованы ключевые функциональные модули: авторизация пользователей, управление профилями, лагерями, сменами и отрядами, новостная лента, хранение файлов и уведомления. Особое внимание было уделено разграничению прав доступа между пользователями, родителями, вожатыми и администраторами, а также поддержке ролей внутри лагеря.

Итогом работы стало веб-приложение, которое позволяет повысить эффективность работы вожатых и администрации, упростить организацию отрядной деятельности и улучшить информационное сопровождение родителей во время лагерной смены. Разработанная система может быть использована как основа для дальнейшего развития цифровой инфраструктуры детского оздоровительного лагеря.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Использование диаграммы вариантов использования UML при проектировании программного обеспечения // Хабр URL: <https://habr.com/ru/articles/566218/> (дата обращения: 15.05.2026).
2. Диаграмма последовательности (sequence-диаграмма) // Хабр URL: <https://habr.com/ru/articles/814769/> (дата обращения: 15.05.2026).
3. ER-модель // Википедия URL: <https://ru.wikipedia.org/wiki/ER-модель> (дата обращения: 15.05.2026).
4. Диаграмма компонентов // Википедия URL: https://ru.wikipedia.org/wiki/Диаграмма_компонентов (дата обращения: 15.05.2026).
5. PostgreSQL Documentation // The PostgreSQL Global Development Group : official documentation. – URL: <https://www.postgresql.org/docs/> – Дата обращения: 13.05.2026.
6. React Documentation // React : official documentation. – URL: <https://react.dev/> – Дата обращения: 13.05.2026.
7. Apache Kafka Documentation // Apache Software Foundation : official documentation. – URL: <https://kafka.apache.org/documentation/> – Дата обращения: 13.05.2026.
8. Spring Cloud Gateway Reference Documentation // Spring : official documentation. – URL: <https://docs.spring.io/spring-cloud-gateway/reference/index.html> – Дата обращения: 13.05.2026.

© Р. Ж. Гафиятов, Е. Ю. Воронкин, 2026

И. П. Гилёв¹✉, Т. Ю. Бугакова¹

Разработка аппаратно-программного стенда для автоматизации процесса тестирования аудио- и видеорегистраторов

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: ivan.gilev.90@mail.ru

Аннотация. В работе рассматривается проблема автоматизации тестирования видеорегистраторов серии «Контроль». В условиях производства проверка одного устройства ручным способом занимает около 60 минут, при этом не обеспечивается документация процесса. Готовые коммерческие системы не могут быть адаптированы под специфику устройств. Целью работы является разработка аппаратно-программного стенда для автоматизации процесса функционального тестирования и обеспечения параллельности проверки сразу нескольких устройств. Для достижения цели определен стек технологий: язык программирования Python, библиотеки PySerial, Requests, Pydantic, Selenium, Pillow, Jinja2. Реализована автоматическая настройка сетевых параметров через COM-порт с генерацией пароля по серийному номеру изделия через стороннюю утилиту. Создана система скриншотов с выделением проверяемого элемента и генерация интерактивных HTML-отчетов. Обеспечено параллельное тестирование нескольких устройств. В процессе тестирования система показала существенное сокращение времени проверки, с 60 до 15 минут на одно устройство. Система готова к внедрению в производственный отдел компании заказчика ООО «СИГНАТЕК».

Ключевые слова: тестирование, параллельное тестирование, Astra Linux, COM-порт

I. P. Gilev¹✉, T. Yu. Bugakova¹

Development of a hardware-software test bench for automating the testing process of audio and video recorders

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: ivan.gilev.90@mail.ru

Abstract. The paper considers the problem of automating the testing of video recorders of the "Control" series. In production conditions, checking one device manually takes about 60 minutes, while no documentation of the process is provided. Ready-made commercial systems cannot be adapted to the specifics of the devices. The purpose of the work is to develop a hardware and software stand for automating the functional testing process and ensuring parallel testing of several devices at once. A technology stack has been defined to achieve this goal: the Python programming language, the PySerial, Requests, Pydantic, Selenium, Pillow, and Jinja2 libraries. Automatic configuration of network parameters via a COM port with password generation based on the serial number of the product via a third-party utility is implemented. A screenshot system has been created with the selection of the checked element and the generation of interactive HTML reports. Parallel testing of several devices is provided. During testing, the system showed a significant reduction in verification time, from 60 to 15 minutes per device. The system is ready for implementation in the production department of the customer's company, SIGNATEK LLC.

Keywords: testing, parallel testing, Astra Linux, COM port

Введение

Большинство предприятий сталкивается с необходимостью автоматизации процессов контроля качества выпускаемой продукции. Ручное тестирование изделий занимает достаточно продолжительное время и становится не эффективным при росте объемов производства.

Анализ существующих решений показывает, что готовые коммерческие решения не адаптированы под всю специфику изделий серии «Контроль», а различные скрипты, разработанные сотрудниками, не обеспечивают нужный уровень качества проверки.

Ключевыми требованиями к разрабатываемой системе стали: автоматическая настройка IP-адресов через COM-порт с генерацией пароля по серийному номеру, проверка работоспособности накопителей, LTE-модема, WiFi-адаптера и видеозаписи, создание скриншотов веб-интерфейса с выделением проверяемых элементов, формирование HTML-отчетов, а также возможность параллельного тестирования нескольких устройств.

Практическая значимость данной разработки заключается в возможности ее внедрения в производственный отдел компании ООО «СИГНАТЕК». Данная система разработана в рамках технического задания предприятия и принята в эксплуатацию.

Техническое задание

Согласно техническому заданию, полученному от ООО «СИГНАТЕК», был сформирован определенный перечень требований к разрабатываемой системе автоматизированного тестирования.

В качестве основного языка программирования был выбран Python версии 3.7. Для обеспечения взаимодействия с тестируемым оборудованием использованы библиотеки PySerial (организация обмена данными через COM-порты), Requests (выполнение HTTP-запросов), Pydantic (проверка корректности принимаемых данных), Selenium (управление браузером), Pillow (обработка графических файлов) и Jinja2 (построение HTML-страниц). Функционирование системы должно осуществляться под управлением операционной системы Astra Linux Common Edition 1.7.6. Согласно техническому заданию, разрабатываемое решение должно обеспечивать автоматическую настройку сетевых параметров устройств через последовательный интерфейс с генерацией пароля по серийному номеру. В обязательном порядке подлежали проверке следующие компоненты: система хранения (SD и SSD накопители), LTE модем, Wi-Fi адаптер, а также функционал записи видеофайла с несколькими вариантами параметров. Дополнительно требовалось внедрить механизм формирования снимков экрана веб-интерфейса с визуальным выделением проверяемых элементов, организовать генерацию HTML-отчетов, а также предусмотреть одновременное тестирование нескольких экземпляров устройств. Реализации перечисленных пунктов посвящены соответствующие этапы практической части, изложенные в разделе «Результаты».

Методы и материалы

В процессе исследования использовались следующие методы.

1. Метод анализа предметной области для изучения протоколов взаимодействия с тестируемыми регистраторами серии «Контроль» (UART, HTTP API, веб-интерфейс), что позволило обосновать выбор технологий, зафиксированных в техническом задании [1].

2. Метод архитектурного моделирования, позволил спроектировать физическую схему тестового стенда с двумя сетевыми картами, кастомным USB-хабом с внешним источником питания, тестовым коммутатором и Wi-Fi точкой доступа (рис. 1).

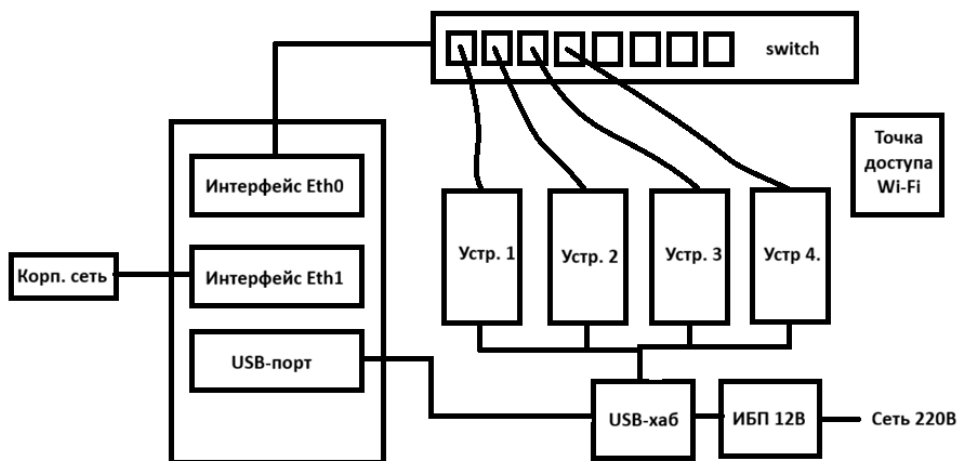


Рис. 1. Аппаратная схема стенда

3. Метод экспериментального тестирования включал проверку каждого функционального модуля (настройка IP через COM-порт, проверка дисков, LTE, WiFi, видеозапись, скриншоты, отчеты), а также тестирование параллельной работы сразу нескольких устройств с помощью многопоточности.

4. Метод сравнительного анализа был использован для оценки временных затрат на проверку по сравнению с ручным способом.

В качестве рабочих материалов использовался персональный компьютер с процессором Intel Core i5, 8 ГБ оперативной памяти DDR3, ОС Astra Linux Common Edition 1.7.5. и стек технологий - Python 3.7 с библиотеками PySerial, Requests, Pydantic, Selenium, Pillow, Jinja2. Оборудование включало USB-хаб с внешним питанием 12В/2А, коммутатор на 16 портов и корпоративную точку доступа Wi-Fi.

Результаты

В результате была полностью реализована система автоматизированного тестирования видеорегистраторов «Контроль» в соответствии с техническим заданием.

На первом этапе разработан модуль, отвечающий за настройку сетевых параметров устройств через COM-порт. Программа автоматически сканирует порты компьютера на наличие устройств USB или ACM, генерирует для них пароли и задает сетевые настройки (рис. 2). Так же решена проблема кодировки CP866, возникшая при чтении ответов регистраторов.

```
"wifi": {  
  "ssid": " ",  
  "bssid": " ",  
  "password": " "  
},  
"184087": {  
  "com_port": "/dev/ttyACM0",  
  "state": "OK",  
  "ip": "192.168.2.200",  
  "netmask": "255.255.255.0",  
  "gw": "192.168.2.254",  
  "password": " "  
}
```

Рис. 2. Сетевые параметры видеорегистратора

Вторым этапом реализованы функциональные тесты. Проверка накопителей подтверждает наличие SSD и SD-карты в статусе «Активен» или «Подключен» до 5 минут. Тест LTE-модема подразумевает, ожидание получения устройством IP, шлюза и DNS, после чего модем выключается. Тест WiFi-адаптера подключает устройство к точке доступа с фиксированными параметрами и сохраняет полученный IP для дальнейших тестов. Тест видеозаписи для каждого канала применяет 5 наборов настроек (разрешение, FPS, битрейт, аудио), записывает 30 секунд, скачивает AVI-файлы и проверяет их размер (рис. 3).

Хранилище данных	
Накопитель	MicroSD
Серийный номер	7190005c
Размер	237760.00 Мбайт
Доступно для записи	234564.50 Мбайт
Состояние	Активен
Температура	-
Форматировать Отключить	

Рис. 3. Пример проверки SD

Для отчетности организована система визуальной верификации, где с помощью Selenium делаются скриншоты страницы веб браузера с выделением проверяемого элемента [2]. Так же в процессе тестирования происходит генерация интерактивных HTML-отчетов с группировкой по типам проверок (рис. 4).

expand allcollapse all	
Общая информация	- Успешное подключение к устройству  (999999 / 192.168.9.225). - Функциональные проверки для устройства 999999 завершены.
Тест накопителей	- Начало проверки SSD-накопителя и SD-карты. - Текущие состояния SSD-накопителя и SD-карты: Запись, Подключен. - Окончание проверки SSD-накопителя и SD-карты.
Тест модема LTE	- Начало проверки LTE-модема. - Успешное подключение к сети оператора связи. Параметры сети: IP-адрес: 85.141.112.184, шлюз: 85.141.112.185, dns1: 217.8.224.10, dns2: 217.8.224.40. - Окончание проверки LTE-модема.
Тест адаптера Wi-Fi	- Начало проверки адаптера Wi-Fi и получение адреса в сети Wi-Fi. - Идентификатор адаптера Wi-Fi: Realtek RTL8192EU, IP-адрес в сети Wi-Fi: 172.16.1.133. - Окончание проверки адаптера Wi-Fi.
Тест каналов и записей (Eth)	- Начало проверки установки настроек каналов и записи файлов. - Записано файлов для канала 0: 5. - Записано файлов для канала 1: 5. - Окончание проверки установки настроек каналов и записи файлов.
Тест каналов и записей (Wi-Fi)	- Начало проверки установки настроек каналов и записи файлов. - Записано файлов для канала 0: 5. - Записано файлов для канала 1: 5. - Окончание проверки установки настроек каналов и записи файлов.

Рис. 4. Итоговый HTML-отчет

В соответствии с техническим заданием предусмотрена возможность параллельного тестирования сразу нескольких устройств с помощью пула потоков.

Полученные в результате исследования результаты доказывают, что данная система способна существенно сократить временные затраты на тестирование, обеспечить документированность процесса, а также гарантировать что все изделия будут проверены по одним и тем же критериям. При необходимости система может быстро масштабироваться путем увеличения оперативной мощности стенда.

Исходя из результатов других исследований можно сделать вывод что использование многопоточности отлично подходит для задач с явным ожиданием ответов браузера и самого устройства [3]. Компания YADRO в своих исследованиях утверждает, что применение готовых утилит не всегда закрывает специфические потребности, связанные с тестированием аппаратного оборудования, это

подтверждает правильность выбора стратегии разработки собственного продукта [4].

Экспериментальное тестирование показало, что данная система способна сократить время проверки одного устройства с 60 до 15 минут, при этом она показывает стабильную работу при одновременном тестировании сразу 10 устройств, что сопоставимо с результатами, представленными в исследованиях Wiren Board, где аналогичные показатели достигались на производстве [5].

Заключение

Результатом работы является готовая система автоматизированного функционального тестирования изделий собственного производства компании заказчика. Важными результатами разработки являются существенное сокращение времени для проверок, создание интерактивной системы отчетности и организация параллельности тестирования.

Разработка принята в опытную эксплуатацию производственным отделом компании ООО «СИГНАТЕК», что подтверждает ее практическую значимость. Основное направление для развития данной темы это адаптация системы под другие регистраторы той же серии.

Благодарности

Авторы выражают благодарность производственному отделу ООО «СИГНАТЕК» за содействие при выполнении работы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Петров А.А. Протокол передачи данных для UART [сайт] - URL: <https://cyberleninka.ru/article/n/protokol-peredachi-dannyh-dlya-uart> (дата обращения: 29.04.2026). - Режим доступа: свободный. - Текст: электронный.
2. Ткачев А.В., Иртегов Д.В. Методика автоматического тестирования развивающегося веб-приложения [сайт] - URL: <https://cyberleninka.ru/article/n/metodika-avtomaticheskogo-testirovaniya-razvivayuschegosya-veb-prilozheniya> (дата обращения: 29.04.2026). - Режим доступа: свободный. - Текст: электронный.
3. Хюппенен А. Тестирование CLI-приложений без костылей: единый фреймворк вместо десятка утилит [сайт] – URL: <https://engineer.yadro.com/article/testirovanie-cli-prilozhenij-bez-kostylej-edinyj-frejmwork-vmesto-desyatka-utilit/> (дата обращения: 29.04.2026). – Режим доступа: свободный. – Текст: электронный.
4. Мутилин В.С. Многопоточное тестирование программных интерфейсов [сайт] – URL: <https://cyberleninka.ru/article/n/mnogopotocnoe-testirovanie-programmnyh-interfejsov> (дата обращения: 29.04.2026). – Режим доступа: свободный. – Текст: электронный.
5. Wiren Board. Как мы тестируем устройства, которые производим: Python, Grafana и крафтовые стенды [сайт] - URL: <https://habr.com/ru/companies/wirenboard/articles/798325/> (дата обращения: 29.04.2026). - Режим доступа: свободный. - Текст: электронный.

© И. П. Гилёв, Т. Ю. Бугакова, 2026

В. А. Гилева^{1✉}, С. Ю. Кацко¹

Проектирование и реализация масштабируемого приложения для шаблонного создания структурированных отчетов

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: valeria.gileva.04@list.ru

Аннотация. Студенты зачастую затрачивают значительное время на самостоятельное форматирование учебных документов. Предоставление пользователю стандартного текстового редактора с избыточной функциональностью повышает риск возникновения ошибки оформления. В результате преподаватели возвращают подобные работы на доработку. В данной работе рассматривается опыт создания узкоспециализированного веб-инструмента. Логика работы системы построена на строгой регламентации: обучающийся осуществляет ввод содержательной части, а система обеспечивает автоматическое формирование итогового отчета. Техническая реализация проекта опирается на трехуровневую архитектуру на базе React и Node.js. Хранение разнородных фрагментов документа в классических реляционных таблицах представляется технически нецелесообразным. В качестве оптимального решения было выбрано применение формата JSONB в СУБД PostgreSQL, куда контент сохраняется в виде бинарного массива. Наиболее ресурсоемкие операции реализованы на серверной стороне. Взамен трудоемкой прямой реализации XML-разметки, разработанный алгоритм интегрирует данные веб-блоков в структуру базового эталонного шаблона DOCX. Изображения при этом масштабируются для обеспечения соответствия границам печатного листа. Таким образом, процессы форматирования автоматизируются. Снижается когнитивная нагрузка на студентов, а вероятность необходимости доработки из-за несоблюдения требований форматирования минимизируется.

Ключевые слова: генерация отчетов, веб-приложение, клиент-серверная архитектура, JSONB, автоматизация, шаблонизатор, React, Node.js

V. A. Gileva^{1✉}, S. Y. Katsko¹

Design and implementation of a scalable application for template-based creation of structured reports

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: valeria.gileva.04@list.ru

Abstract. Students often spend a considerable amount of time manually formatting academic documents. Providing users with a standard word processor featuring excessive functionality significantly increases the risk of formatting errors. As a result, instructors frequently return such assignments for revision. This paper discusses the practical experience of developing a highly specialized web-based tool. The system's operational logic is built on strict regulation: the student only inputs the core content, while the software ensures the automatic generation of the final report. The technical implementation of the project relies on a three-tier architecture based on React and Node.js. Storing heterogeneous document fragments within classic relational tables proved to be technically impractical. As an optimal workaround, the JSONB format was utilized within the PostgreSQL database, allowing all content to be saved as a single binary array. The most resource-intensive operations are strictly handled on the server side. Instead of laboriously generating XML markup from scratch, the developed algorithm seamlessly integrates web block data into the structure of a pre-configured DOCX reference

template. Uploaded images are programmatically scaled to ensure strict compliance with the borders of a standard printed page. Ultimately, the entire formatting process is fully automated. The cognitive load on students decreases significantly, and the probability of returning assignments due to formatting violations is reduced to a minimum.

Keywords: report generation, web application, client-server architecture, JSONB, automation, template engine, React, Node.js

Введение

Оформление документации по изучаемым дисциплинам требует существенных затрат учебного времени. Использование стандартных текстовых редакторов возлагает ответственность за нормоконтроль на самого студента. Избыточный функционал затрудняет концентрацию, что способствует увеличению количества неточностей. На поиск и исправление подобных технических недочетов затрачивается существенная часть времени проверки работ, что доказывает слабую эффективность ручного контроля [1].

Лидеры рынка вроде Microsoft Word визуально удобны. Однако слишком тесно связывают содержание документа с его представлением. Системы компьютерной верстки (LaTeX) исключают случайное изменение структуры. Данная степень надежности сопряжена с высокой сложностью освоения системы. Существуют также коммерческие блочные SaaS-платформы. PandaDoc удобен, но ориентирован исключительно на бизнес-контракты.

Выявляется потребность в разработке специализированного программного обеспечения. Цель исследования – проектирование и реализация масштабируемого приложения для шаблонного создания структурированных отчетов.

Разрабатываемое программное обеспечение призвано сочетать интерфейс блочного конструктора со строгим автоматическим контролем параметров форматирования. Это позволит студентам сфокусироваться на сути задания, а преподавателям – ускорить процесс проверки.

Методы и материалы

Проектирование системы базируется на классической трехуровневой клиент-серверной архитектуре. Разделение на визуальный интерфейс, вычислительный узел и подсистему хранения данных обеспечило необходимую гибкость.

Логика клиентской части написана на языке JavaScript с использованием библиотеки React. Выбор данной технологии обоснован ее компонентной архитектурой. За счет этой концепции массивную рабочую область удастся разделить на изолированные блоки (текст, изображение, список), пригодные для многократного встраивания без дублирования кода [2].

Схема взаимодействия компонентов интерфейса спроектирована с целью минимизации рисков совершения пользователем некорректных действий (рис. 1). Слева расположена панель инициализации команд, по центру – строго структурированная сетка редактора, а справа – изолированная колонка для ввода метаданных титульного листа.

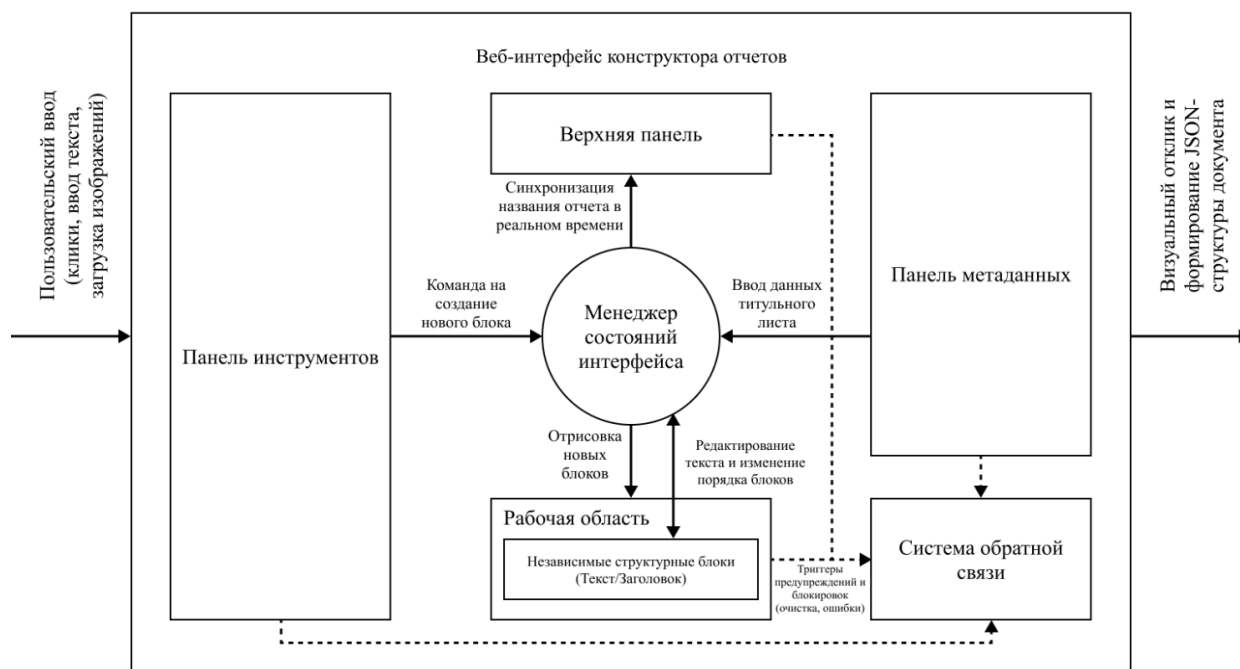


Рис. 1. Концептуальная модель взаимодействия компонентов интерфейса

Особое внимание при проектировании было уделено организации подсистемы хранения данных. Использование классической реляционной модели для хранения разноплановых элементов отчета (текста и графики) сопряжено с рядом технических ограничений. Решением стала разработка гибридной модели на базе объектно-реляционной СУБД PostgreSQL.

Метаданные организованы в типизированных столбцах. Слабоструктурированное содержимое документа хранится в отдельном поле типа JSONB (рис. 2). Динамический текст и изображения в кодировке Base64 предварительно преобразуются с помощью метода `JSON.stringify` для предотвращения ошибок сериализации глубоко вложенных структур, после чего сохраняются в базе данных в виде бинарных объектов.

reports

GeneralColumnsAdvancedConstraintsPartitionsParametersSecuritySQL

Inherited from table(s)Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	id	integer			☒	☒	nextval('r
	title	text			☒	☐	
	student_name	text			☐	☐	
	student_group	text			☐	☐	
	updated_at	timestamp without			☐	☐	CURREN
	meta_data	jsonb			☐	☐	
	content_data	jsonb			☐	☐	
	client_id	text			☐	☐	

Рис. 2. Структура хранения данных отчета в СУБД PostgreSQL

Серверный уровень реализован на платформе Node.js. Применение неблокирующей событийно-ориентированной модели ввода-вывода обеспечивает эффективную обработку параллельных асинхронных запросов [3]. Использование фреймворка Express позволило быстро настроить маршрутизацию. В целях предотвращения сбоев в работе сервера при загрузке объемных графических данных в кодировке Base64, базовый лимит пропускной способности был увеличен до 50 МБ.

Главная вычислительная задача сервера – сборка итогового файла. В качестве решения был реализован алгоритм заполнения эталонного шаблона.

В корневой директории сервера размещен файл-шаблон template.docx. Шаблон содержит предварительно заданные параметры форматирования, соответствующие требованиям ГОСТ. Для извлечения XML-структур из архива применяется утилита PizZip, а модуль docxtemplater используется для внедрения данных в шаблон, не затрагивая исходное форматирование полей и шрифтов [5]. За масштабирование графики отвечает библиотека image-size. Алгоритм считывает исходные размеры изображения. При превышении шириной изображения порогового значения в 600 пикселей, сервер вычисляет пропорциональный коэффициент и производит масштабирование, обеспечивающее соответствие графического объекта границам страницы формата А4 (рис. 3).



Рис. 3. Алгоритм серверной генерации документа

Сформированный бинарный буфер передается клиенту непосредственно. Реализованная архитектура не предусматривает запись промежуточных файлов формата DOCX на дисковый накопитель сервера.

Результаты

Базой для развертывания послужил арендованный виртуальный сервер под управлением Ubuntu. Функцию обратного проксирования взял на себя Nginx, тогда как фоновые процессы контролировались диспетчером PM2. По результатам проведенных испытаний заложенные в модули характеристики соответствуют заявленным требованиям.

Необходимо было подтвердить, что алгоритм фонового автосохранения не приведет к отказу базы данных вследствие избыточной частоты запросов. При эмуляции непрерывного ввода текста скорость поддерживалась на уровне,

превышающем 300 знаков в минуту. Избыточный сетевой трафик ограничивался на стороне клиента, а передача запросов к СУБД инициировалась только в тех случаях, когда в действиях пользователя фиксировалась пауза продолжительностью не менее двух секунд.

Техническое задание устанавливает предельно допустимое значение для одного пакета данных – 100 МБ. Попытка загрузить в редактор объемный графический архив размером 120 МБ привела к незамедлительному системному отказу. Веб-сервер Nginx заблокировал запрос на этапе приема, сформировав стандартную ошибку «413 Payload Too Large».

Оценка серверной компиляции файлов осуществлялась в условиях повышенной нагрузки. Эмулировалась одновременная активность пятидесяти пользователей. Размер входящих JSON-пакетов искусственно варьировался – изменялся объем графических данных, закодированных в Base64. Норматив генерации одного документа DOCX, согласно проектным требованиям, составляет 3-5 секунд. Анализ собранных статистических данных (табл. 1) свидетельствует о наличии значительного запаса производительности.

Таблица 1. Производительность серверной генерации отчетов

Сценарий наполнения	Прогнозируемый размер файла	Фактическое время обработки
Только текст	0,1 МБ	1,8 с
Текст и 3 изображения	~1,0 МБ	1,8 с
Текст и 30 изображений	8,0-9,0 МБ	2,0-3,0 с

График зависимости времени генерации от объема входных данных (рис. 4) имеет практически линейный характер. Отчет с большим объемом графических данных обрабатывается в установленные временные интервалы, а его итоговый размер значительно ниже критических значений.

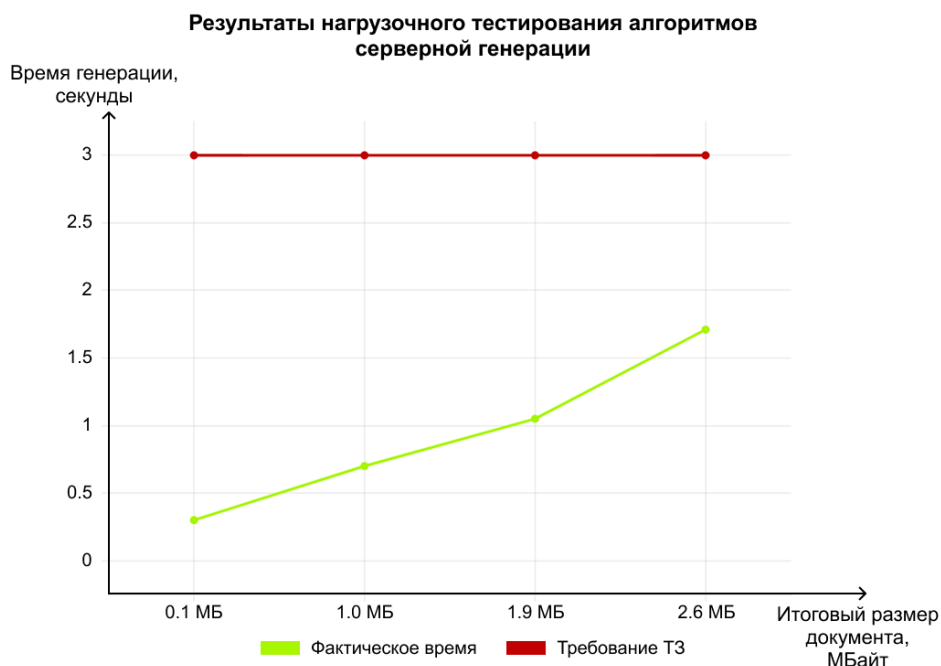


Рис. 4. График зависимости времени генерации документа от объема входных данных

Нарушений параметров форматирования в итоговых документах также не выявлено. Графические файлы большого объема масштабировались до ширины в 600 пикселей, вследствие чего выход за границы печатной области листа формата А4 во всех случаях был исключен.

В течение трех месяцев непрерывного функционирования системы средствами `pm2 monit` не было выявлено значимых утечек памяти (`Memory Leaks`). Оперативная память зафиксировалась на отметке 386 МБ. Полученные данные позволяют сделать вывод, что бинарные буферы `nodebuffer` удаляются из системы непосредственно в момент завершения скачивания готового архива клиентом.

Проверка кроссплатформенности проводилась в двух плоскостях. Во-первых, благодаря встроенному манифесту `Progressive Web App (PWA)`, редактор без проблем устанавливался прямо в среду `Windows`. Во-вторых, оценивалась сама `XML`-разметка готовых файлов. Проверка корректности отображения файлов осуществлялась посредством `Microsoft Word` и облачного сервиса `Google Документы`. Структура нигде не деформировалась, а визуально документы выглядели абсолютно идентично оригиналу.

Обсуждение

Апробация прототипа в условиях, приближенных к реальным, позволила выявить преимущества строгих архитектурных ограничений. Хранение данных с использованием формата `JSONB` в `PostgreSQL` позволило минимизировать количество избыточных связей между таблицами [4]. Базе данных не требуется

выполнение ресурсоемких транзакций при каждом добавлении пользователем нового блока в редактор.

Сравнение с классическим текстовым процессором Microsoft Word выявляет существенные различия в принципах работы. Нарушить верстку в данной системе не представляется возможным. В этом аспекте система близка к профессиональным средам, таким как LaTeX, которые также обеспечивают строгое соблюдение академических стандартов. Сложность освоения системы относительно невысока. Студентам не требуется изучать синтаксис разметки или устанавливать компиляторы. Удалось решить и типичную проблему коммерческих SaaS-конструкторов (таких как PandaDoc). При экспорте сохраняется исходная структура и форматирование документа. Строгая изоляция React-компонентов в браузере обеспечивает, что на сервер будет передана корректная иерархия блоков.

На стороне сервера алгоритм заполнения эталонного шаблона (модуль docxtemplater) продемонстрировал более высокую эффективность, чем внешние конвертеры, такие как Pandoc. Для архитектуры системы отсутствует необходимость инициализации нового процесса операционной системы для каждой генерации отчета. В отношении обработки графики конвертация изображений в Base64 стала обоснованным техническим компромиссом. Вследствие кодировки объем трафика возрастает приблизительно на треть, однако для формата MVP данное решение представляется оправданным.

Установленный уровень защиты является достаточным для автономного функционирования продукта. Доступ к сторонним отчетам ограничен посредством анонимных сессий, идентификаторы которых хранятся в локальном хранилище браузера. Однако данная модель обладает определенными техническими ограничениями, в частности, потенциальной уязвимостью к атакам межсайтового скриптинга (XSS). Поэтому перед возможным внедрением системы в информационную среду вуза необходимо будет провести дополнительное проектирование архитектуры безопасности.

Заключение

Требования технического задания реализованы в полном объеме. Результатом работы стал действующий масштабируемый веб-редактор, предназначенный для формирования структурированных отчетов на основе предварительно настроенного шаблона.

Выбранная трехуровневая архитектура продемонстрировала свою эффективность в ходе тестирования. Переход на формат данных JSONB внутри PostgreSQL обеспечил возможность структурирования динамического контента документа в виде единого массива.

На стороне клиента интерфейс (React) обеспечивает строгое соблюдение пользователем установленных ограничений. Для студентов минимизируется необходимость ручного контроля отступов и проверки соответствия иерархии шрифтов. Данные функции реализуются программным узлом на базе Node.js.

Алгоритм использует эталонный шаблон для адресного заполнения данными, при этом процесс формирования документа осуществляется в оперативной памяти.

Система продемонстрировала стабильность в ходе выполнения всех запланированных нагрузочных испытаний. Скорость работы соответствует установленным нормативам, поскольку время компиляции документов с большим объемом данных не превышает двух секунд. Обеспечение базового уровня информационной безопасности реализовано посредством механизма анонимных сессий с локальными ключами.

В перспективе одним из приоритетных направлений развития станет модернизация архитектуры информационной безопасности. Разработанный программный сервис может быть интегрирован в качестве функционального компонента в единую информационно-образовательную среду университета.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Стариченко, Б. Е. Программа автоматизации контроля оформления текстовых документов / Б. Е. Стариченко, М. А. Устинов. – Текст : электронный // Педагогическое образование в России. – 2018. – № 8. – С. 163–168. – URL: <https://cyberleninka.ru/article/n/programma-avtomatizatsii-kontrolya-oformleniya-tekstovyh-dokumentov> (дата обращения: 07.04.2026).
2. Мамбетов, Р. А. Построение пользовательских интерфейсов веб-приложений с помощью библиотеки React / Р. А. Мамбетов. – Текст : электронный // Экономика и социум. – 2019. – № 5 (60). – С. 906–909. – URL: <https://cyberleninka.ru/article/n/postroenie-polzovatelских-interfeysov-veb-prilozheniy-s-pomoschyu-biblioteki-react> (дата обращения: 09.04.2026).
3. Стрелецкий, А. Д. Асинхронное программирование в Node.js: преимущества и подходы / А. Д. Стрелецкий, Н. М. Нетребин, С. Н. Русак. – Текст : электронный // Инновационная наука. – 2023. – № 11-2. – С. 43–44. – URL: <https://cyberleninka.ru/article/n/asinhronnoe-programmirovaniye-v-node-js-preimuschestva-i-podhody> (дата обращения: 09.04.2026).
4. Широков, И. А. Исследование производительности и методов хранения JSON-объектов в СУБД PostgreSQL / И. А. Широков. – Текст : электронный // Вестник НГУЭУ. – 2023. – № 3. – С. 144–153. – DOI: 10.34020/2073-6495-2023-3-144-153.
5. Колесинский, М. В. Автоматизация создания документов: совмещение шаблонов и диалогового взаимодействия / М. В. Колесинский, Н. И. Белодед. – Текст : электронный // Управление информационными ресурсами : материалы XX Международной научно-практической конференции. – Минск : Академия управления при Президенте Республики Беларусь, 2024. – С. 453–454. – URL: <https://elibrary.ru/item.asp?id=67323265> (дата обращения: 09.04.2026).

© В. А. Гилева, С. Ю. Кацко, 2026

Я. И. Головки^{1✉}, Е. Ю. Воронкин¹

Разработка программного бота для ранжирования списков абитуриентов и информирования о вступительных испытаниях

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: golovko085@gmail.com

Аннотация. Разбирается процесс создания программного бота для автоматизации деятельности приемной комиссии СГУГиТ. Актуальность проекта следует из потребностей в автоматизированной информационной системе для уведомления абитуриентов о вступительных испытаниях, а также для просмотра абитуриентам их мест в конкурсных списках. Для разработки основной логической части использовался Golang, а для разработки интерфейса JavaScript, для хранения данных использовался SQLite, JWT использовался для механизма авторизации. Были предусмотрены возможности просмотра конкурсной позиции, подбора направлений по предметам ЕГЭ, подписка на уведомления об изменениях, также средства административной панели для создания, изменения, удаления специальностей, загрузки выгрузки конкурсных списков. Также был сделан дополнительный акцент на информационной безопасности системы. С помощью данной работы происходит демонстрация применения программных ботов в работе приемной комиссии.

Ключевые слова: веб-программирование, golang, javascript, sqlite, jwt

Y. I. Golovko^{1✉}, E. Yu. Voronkin¹

Development of a program bot to shuffle entrants lists and inform about entrance contests

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: golovko085@gmail.com

Abstract. The development of a software bot is described to automate the work of the admissions committee at Saratov State University of Geography and Technology. The project's relevance lies in the automated information system for receiving notifications to applicants about upcoming tests, as well as for viewing their places on the competition lists. Golang was used to develop the main logic, JavaScript was used for the interface, SQLite was used for data storage, and JWT was used for the authorization mechanism. There are included possibilities of browsing competition positions, choosing subjects based on Unified State Exam subjects, subscribing to change notifications, and administrative panel tools for working with specializations, as well as uploading and downloading competition lists. Additional emphasis was placed on information security. This project demonstrates the use of software bots in the work of the admissions committee.

Keywords: web programming, golang, javascript, sqlite, jwt

Введение

Современные образовательные учреждения нуждаются в эффективных цифровых инструментах для автоматизации процессов взаимодействия с абитуриентами [1, 2]. Приемные комиссии вузов ежедневно обрабатывают большое коли-

чество обращений, связанных с ранжированием конкурсных списков, информированием о вступительных испытаниях, а также предоставлением актуальных данных о поступлении. Использование традиционных способов информирования, таких как телефонные звонки, электронная почта и публикация информации на сайте, увеличивает нагрузку на сотрудников приемной комиссии и не всегда обеспечивает оперативное получение информации абитуриентами.

На данный момент значительная часть информации о ходе приемной кампании публикуется на официальном сайте СГУГиТ, однако данный способ не всегда является удобным для абитуриентов, поскольку требует постоянного самостоятельного отслеживания изменений. По этой причине было принято решение о разработке автоматизированного средства информирования, позволяющего упростить взаимодействие между абитуриентами и приемной комиссией.

Целью настоящей работы является разработка программного бота для ранжирования списка абитуриентов и информирования о вступительных испытаниях. Чтобы выполнить поставленную цель, было достигнуто выполнению определенных задачи:

- анализ существующих решений и сравнение с разрабатываемым проектом;
- проектирование архитектурной части основных модулей системы;
- разработка серверной программной части на языке программирования Golang;
- создание клиентской части на стеке технологий JavaScript, HTML, CSS;
- создание базы данных с использованием SQLite;
- авторизация через JWT;
- тестирование и включение проекта в рабочий процесс приемной комиссии.

Методы и материалы

Первым шагом необходимо спроектировать все необходимые диаграммы [3, 4].

Для правильного понимания взаимодействия различных пользователей, как админ панели, так и программного бота, была создана диаграмма прецедентов [5, 6] (рис. 1).

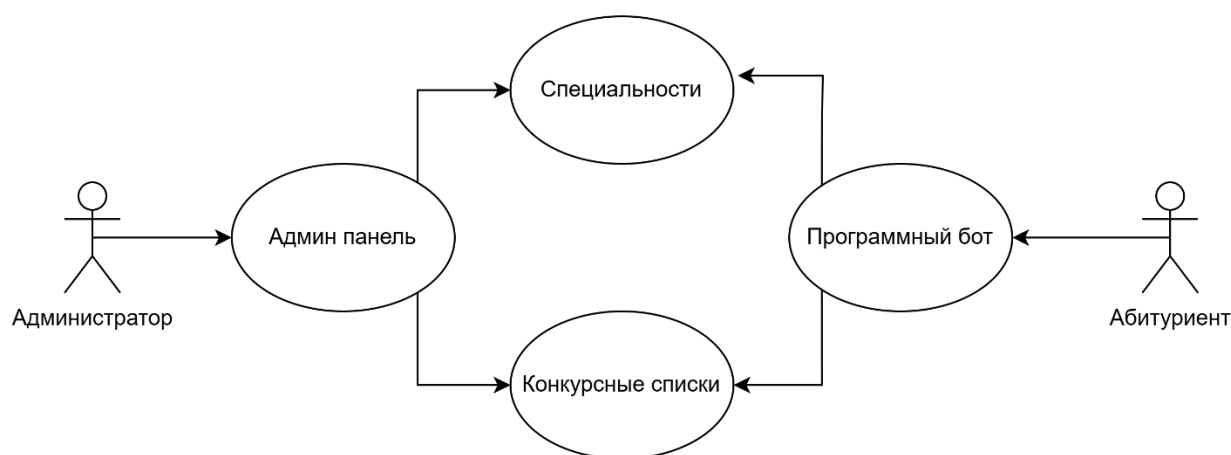


Рис. 1. Диаграмма прецедентов

Затем была сделана схема архитектуры проекта, она позволяет понять то, как части системы, в данном случае Telegram-бота и админ панели, будут взаимодействовать между собой [7, 8] (рис. 2).



Рис. 2. Архитектурная схема

Разработка

Далее был определен набор технологий проекта.

Для серверной части был выбран админ-панели Golang [9, 10]. Интерфейс был разработан с помощью HTML, CSS и JavaScript [11, 12].

Для разработки Telegram бота, также использовался Golang [13, 14].

Для хранения данных админ панели и бота используется SQLite и JSON, но для хранения конкурсных списков использовался XML [15, 16].

Далее был определен набор команд для бота, «Просмотр специальности для поступления» (рис. 3), «Посмотреть место в рейтинге» (рис. 4), «Подписаться на изменения» (рис. 5) [17, 18].

После был определен набор функциональных требований к админ-панели, Авторизация, выход, удаление, изменение, добавление специальности, выгрузка и загрузка конкурсных списков, смена языка и световой темы (рис. 6) [19, 20].

Далее был разработан определенный функционал.

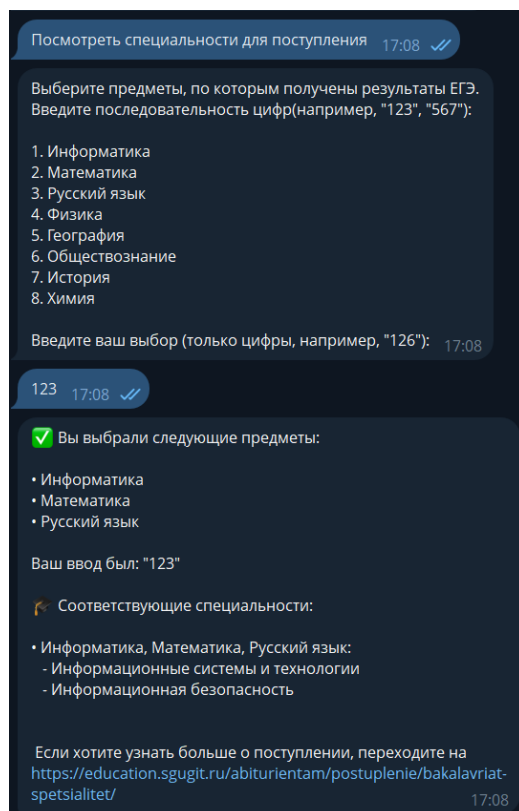


Рис. 3. Просмотр специальности для поступления

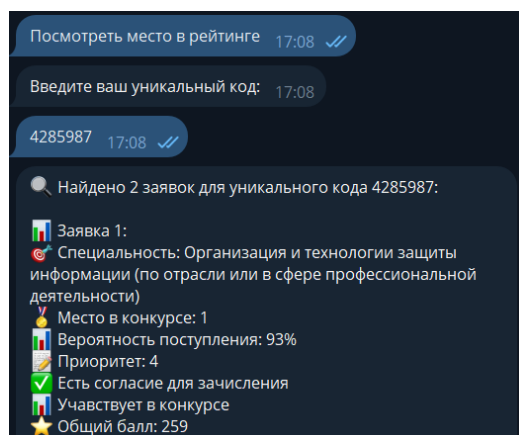


Рис. 4. Посмотреть место в рейтинге

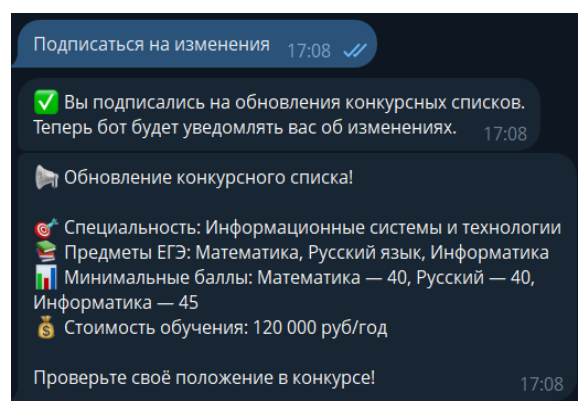


Рис. 5. Подписаться на изменения

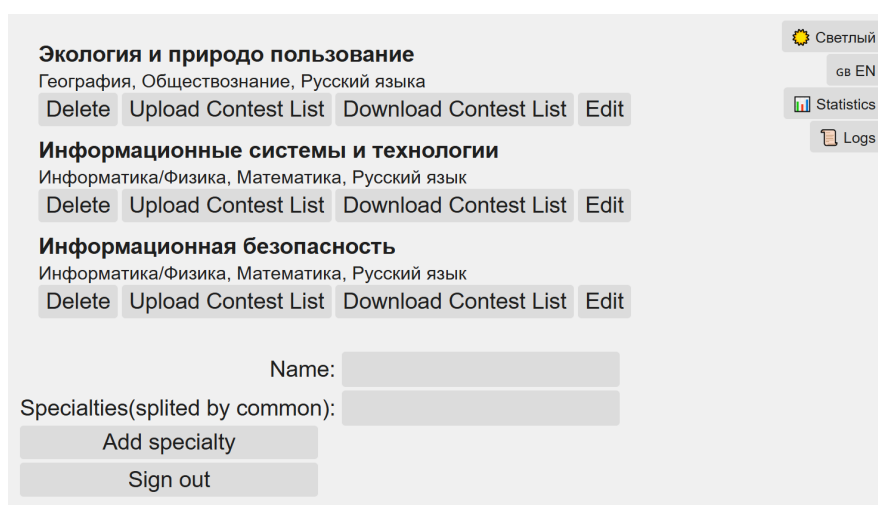


Рис. 6. Работа админ-панели

Заключение

В данной работе продемонстрированы проектирование и разработка системы для автоматизированной работы с абитуриентами. В отличие от традиционных способов информирования через сайт и телефонные обращения, разработанное решение позволяет получать необходимую информацию в автоматическом режиме через Telegram.

Разработка системы велась с использованием языка программирования Golang, базы данных SQLite, а также технологий JavaScript, HTML и CSS для реализации административной панели. Для обеспечения безопасной авторизации был реализован механизм JWT-аутентификации. В процессе разработки были спроектированы основные компоненты системы, структура базы данных и пользовательские сценарии взаимодействия.

Функциональные возможности Telegram-бота включают просмотр доступных специальностей с учетом выбранных предметов ЕГЭ, просмотр положения абитуриента в конкурсных списках, а также подписку на уведомления об изменениях конкурсной ситуации. Административная панель предоставляет сотрудникам приемной

комиссии возможности авторизации и выхода из системы, добавления, изменения и удаления специальностей, загрузки и выгрузки конкурсных списков, а также смены языка интерфейса и световой темы.

Итогом работы стала разработка полноценной системы автоматизации взаимодействия абитуриентов с приемной комиссией, позволяющей упростить получение информации о поступлении, снизить нагрузку на сотрудников и повысить удобство сопровождения приемной кампании.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Разработка автоматизированных систем информирования в высших учебных заведениях // Вестник высшей школы. – 2024. – № 1. – с. 5–11.
2. Автоматизация процессов в приемной комиссии вуза с использованием чат-ботов // Цифровизация образования. – 2024. – № 3. – с. 15–22.
3. Проектирование информационных систем с использованием UML // Открытые системы. – 2023. – № 2. – с. 40–45.
4. Фаулер, М. UML. Основы / М. Фаулер. – Санкт-Петербург : Символ-Плюс, 2020. – 192 с.
5. Применение диаграмм прецедентов для моделирования бизнес-процессов // Компьютерные науки и технологии. – 2025. – № 1. – с. 12–18.
6. Разработка архитектуры распределенных веб-приложений // Программная инженерия. – 2024. – № 7. – с. 34–41.
7. Проектирование клиент-серверных систем взаимодействия // Вестник компьютерных технологий. – 2023. – № 4. – с. 22–29.
8. Особенности построения архитектуры современных чат-ботов // Информационные технологии в управлении. – 2025. – № 2. – с. 50–57.
9. Донован, А. А. Язык программирования Go / А. А. Донован, Б. У. Керниган. – Санкт-Петербург : Питер, 2018. – 400 с.
10. Боднер, Дж. Go. Идиомы и паттерны проектирования / Дж. Боднер. – Санкт-Петербург : Питер, 2022. – 464 с.
11. Разработка пользовательских интерфейсов на JavaScript, HTML и CSS // Системный администратор. – 2024. – № 5. – с. 60–65.
12. Современные веб-технологии в создании административных панелей // Мир ПК. – 2023. – № 9. – с. 18–24.
13. Реализация серверной логики на языке Golang // Программирование и ИТ. – 2025. – № 3. – с. 27–33.
14. Опыт создания Telegram-ботов на языке Go // Технологии программирования. – 2024. – № 11. – с. 45–51.
15. SQLite Documentation // SQLite (дата обращения: 11.05.2026).
16. Использование форматов XML и JSON для обмена данными в информационных системах // Молодой ученый. – 2023. – № 42. – с. 89–94.
17. Разработка интерактивных интерфейсов для мессенджеров // Информационное общество. – 2024. – № 6. – с. 71–77.
18. Проектирование систем оповещения пользователей в реальном времени // Научно-технический вестник. – 2025. – № 4. – с. 103–110.
19. Безопасность веб-приложений и механизмы аутентификации // Защита информации. – 2024. – № 2. – с. 14–21.
20. JWT Introduction // JWT.io (дата обращения: 11.05.2026).

© Я. И. Головкин, Е. Ю. Воронкин, 2026

Д. А. Гура¹, Р. А. Дьяченко^{1,3}, Т. А. Тихонов¹✉

Подход к решению проблемы дисбаланса классов облаков точек на основе генерации синтетических данных

¹Кубанский государственный технологический университет, г. Краснодар,
Российская Федерация

²Кубанский государственный аграрный университет, г. Краснодар,
Российская Федерация

³Московский физико-технический институт (национальный исследовательский университет)
Москва, Российская Федерация
e-mail: timka2015@yandex.ru

Аннотация. Рассматривается подход к повышению качества семантической сегментации облаков точек городской территории в условиях выраженного дисбаланса классов. Актуальность исследования обусловлена тем, что реальные облака точек характеризуются неравномерным распределением объектов по семантическим классам: массовые классы представлены большим числом точек, тогда как отдельные классы встречаются существенно реже. В результате модель машинного обучения демонстрирует более устойчивое распознавание доминирующих классов и недостаточную чувствительность к малопредставленным классам. В качестве исходных данных использовано размеченное облако точек территории г. Краснодара, включающее более 2 миллионов точек, размеченное на восемь классов объектов городской среды. В работе выполнено базовое обучение архитектуры PointNext, затем применены методы компенсации дисбаланса классов на уровне процесса обучения. Дополнительно предложен подход к генерации синтетических облаков точек, позволяющий управляемо изменять распределение обучающей выборки и пополнять недостаточно представленные классы.

Ключевые слова: семантическая сегментация, облака точек, городская территория, дисбаланс классов, синтетические данные, генерация облаков точек, PointNext, машинное обучение, 3D-данные, городская застройка

D. A. Gura^{1,2}, R. A. Dyachenko^{1,3}, T. A. Tikhonov¹✉

An approach to solving the problem of point cloud class imbalance based on synthetic data generation

¹Kuban State Technological University, Krasnodar, Russian Federation

²Kuban State Agrarian University, Krasnodar, Russian Federation

³Moscow Institute of Physics and Technology (National Research University) Moscow, Russia
e-mail: timka2015@yandex.ru

Abstract. This paper examines an approach to improving the quality of semantic segmentation of urban point clouds under conditions of pronounced class imbalance. The relevance of the study lies in the fact that real-world point clouds are characterized by an uneven distribution of objects across semantic classes: common classes are represented by a large number of points, while individual classes are significantly less common. As a result, the machine learning model demonstrates more robust recognition of dominant classes and insufficient sensitivity to underrepresented classes. The initial data used was a labeled point cloud of Krasnodar, comprising over 2 million points, labeled into eight urban object classes. The study performed basic training of the PointNext architecture, followed by the application of class imbalance compensation methods at the training level. Additionally, an

approach to generating synthetic point clouds is proposed, allowing for controlled changes in the distribution of the training sample and the addition of underrepresented classes.

Keywords: semantic segmentation; point clouds; urban area; class imbalance; synthetic data; point cloud generation; PointNext; machine learning; 3D data; urban development

Введение

Методы искусственного интеллекта позволяют автоматизировать обработку облаков точек и извлекать из них сведения о земной поверхности, зданиях, сооружениях, растительности и других объектах городской среды [1]. Особое значение в этом процессе имеет семантическая сегментация, обеспечивающая переход от неструктурированного массива точек к классифицированному представлению пространственной сцены. Развитие таких методов может способствовать повышению качества геопространственного анализа, совершенствованию государственного кадастрового учета и формированию предпосылок для перехода к трехмерному кадастру недвижимости [2,3].

Развитие методов семантической сегментации во многом связано с появлением крупных открытых наборов данных, задавших стандарты экспериментальной проверки алгоритмов. К их числу относятся Stanford Large-Scale 3D Indoor Spaces (S3DIS), Semantic3D, SemanticKITTI, Toronto-3D и другие бенчмарки, содержащие размеченные облака точек различного масштаба и назначения [4-7].

Одновременно с ростом объема данных особенно заметной стала проблема дисбаланса классов. В облаках точек городской территории обычно доминируют поверхности земли, покрытия, здания и растительность, тогда как небольшие объекты, инженерные элементы, автомобили, артефакты и отдельные сооружения представлены существенно меньшим числом примеров. В результате модель при обучении получает значительно больше данных для одних классов и меньше данных для формирования устойчивых признаков других классов.

Проблема дисбаланса классов имеет принципиальное значение для задач семантической сегментации. Даже при приемлемых значениях общей точности модель может демонстрировать низкое качество распознавания мало представленных классов. Поэтому оценка качества сегментации должна учитывать не только общую точность, но и средние межклассовые метрики, прежде всего IoU, F1 и mIoU.

В существующих исследованиях для компенсации дисбаланса классов в облаках точек применяются различные подходы, направленные как на изменение структуры обучающей выборки, так и на модификацию процедуры обучения модели. В частности, используются методы гибридной выборки, сочетающие увеличение числа примеров редких классов и уменьшение числа примеров доминирующих категорий, а также специальные функции потерь, усиливающие вклад низко представленных классов в процессе обучения [9]. Применяются подходы, основанные на раздельной настройке этапов выделения признаков и классификации объектов, что позволяет уменьшить смещение модели в сторону доминирующих классов. Для дополнительной компенсации дисбаланса также используются модифицированные

функции потерь, повышающие влияние редких классов на процесс оптимизации модели [9]. Проанализированные методы позволяют снизить влияние дисбаланса на этапе обучения и перераспределить внимание модели к редким классам объектов, однако они преимущественно компенсируют последствия дисбаланса, а не устраняют его первопричину, связанную с недостатком репрезентативных примеров редких классов.

Одним из перспективных направлений является дополнение обучающей выборки синтетическими данными. В отличие от классической аугментации, сгенерированные данные можно использовать не только для случайного расширения набора данных, но и для управляемого изменения его структуры. Это позволяет целенаправленно пополнять те классы, которые недостаточно представлены в реальных облаках точек, сохраняя при этом пространственную логику городской сцены.

Цель настоящего исследования заключается в повышении качества семантической сегментации объектов городской территории в облаке точек за счет настройки параметров влияющих на дисбаланс классов при обучении модели и дополнения обучающей выборки синтетическими данными. Для достижения цели решались следующие задачи: формирование и подготовка набора данных облаков точек г. Краснодара, исследование влияния параметров обучения и методов компенсации дисбаланса классов, генерация синтетических данных и оценка влияния синтетических данных на качество сегментации по основным метрикам.

Методы и материалы

В качестве исходных данных использовалось размеченное облако точек территории г. Краснодара. Система семантических классов была сформирована по аналогии с бенчмарком Semantic3D [5]. В исследовании использовались восемь классов: класс 1 - искусственные поверхности, класс 2 - естественные поверхности, класс 3 - высокая растительность, класс 4 - низкая растительность, класс 5 - здания, класс 6 - сооружения, класс 7 - артефакты, класс 8 - автомобили. Выбранная классификация позволяет описывать основные компоненты городской сцены и сопоставлять результаты с общепринятыми подходами к классификации крупномасштабных облаков точек. Набор данных включал 25 участков с общим количеством точек более 2 миллионов. Использование данных реальной городской территории позволило оценить работу модели в условиях неоднородной пространственной структуры и выраженного дисбаланса классов (рис. 1).

В качестве базовой нейросетевой архитектуры использовалась PointNext. Качество обучения оценивалось по основным метрикам: IoU, F1, Acc и mIoU. Метрика IoU характеризует степень пересечения предсказанной и истинной области класса, F1 отражает баланс полноты и точности, Acc показывает точность распознавания класса, а mIoU использовалась как основная интегральная метрика качества сегментации по всем восьми классам.

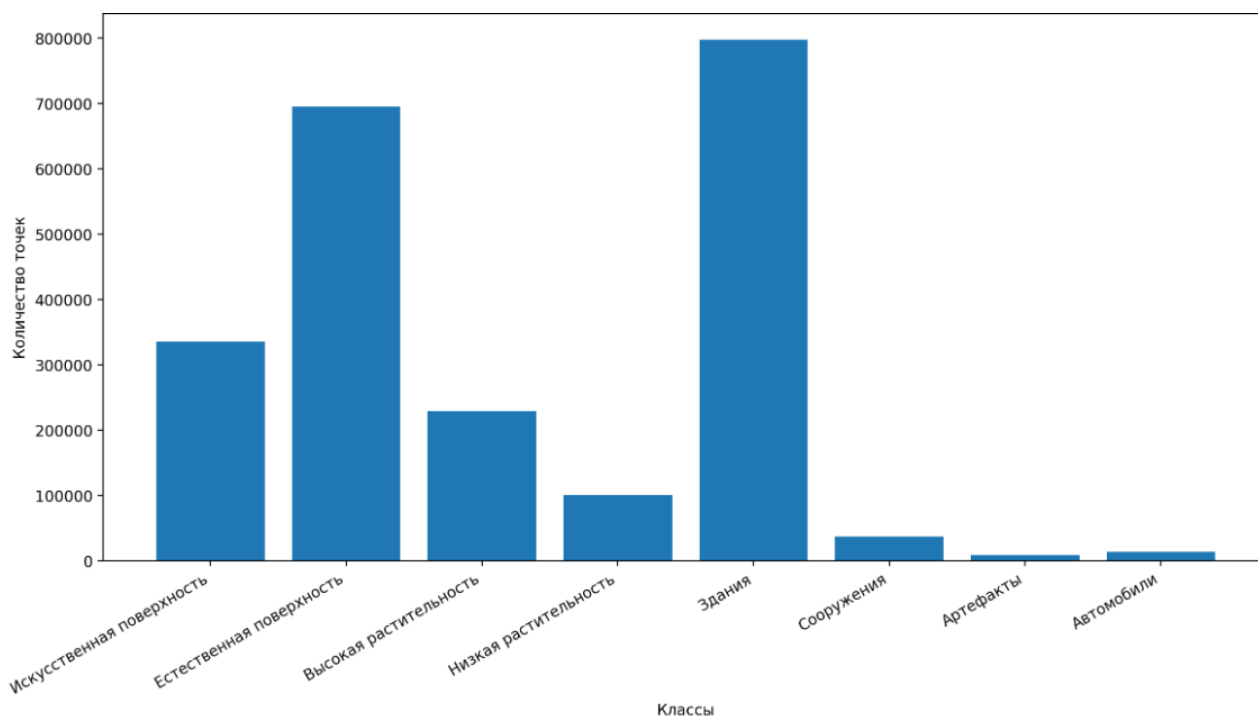


Рис. 1. Гистограмма распределения точек по классам в наборе данных

Вычислительный эксперимент включал три последовательных этапа. На первом этапе проводилось базовое обучение модели без специальных средств компенсации дисбаланса классов. На втором этапе изменялся процесс обучения: применялись стабилизация весовых коэффициентов редких классов, балансировка пространственных выборок, приоритетное обучение на фрагментах сцены с низко представленными классами и взвешивание функции потерь. На третьем этапе обучающая выборка дополнялась синтетически сгенерированными облаками точек.

Для компенсации дисбаланса на уровне обучения использовались параметры, представленные в табл. 1.

Таблица 1. Наилучшие параметры обучения для компенсации дисбаланса классов

Параметр	Значение	Назначение
class weight epsilon	0,010	Стабилизация весовых коэффициентов редких классов при вычислении функции потерь
class balanced crop	True	Балансировка пространственных фрагментов обучающих сцен
class balanced focus prob	0,7	Повышение вероятности выбора участков сцены с низко представленными классами
cls weighed loss	True	Взвешивание функции потерь для увеличения вклада редких классов

Синтетическая генерация рассматривалась как способ управляемого пополнения обучающей выборки редкими классами. Для этого на уровне генерации задавалось требуемое распределение точек по классам. Доля точек, приходящаяся на i -й класс, определялась по формуле:

$$p_i = \frac{w_i}{\sum_{j=1}^C w_j}, \quad (1)$$

где p_i - доля точек i -го класса; w_i - вес i -го класса; w_j - вес j -го класса, используемый при суммировании; C - общее число классов.

Количество точек, генерируемых для i -го класса, вычислялось как:

$$n_i = \text{round}(p_i N) + \delta_i, \quad (2)$$

где n_i - число точек i -го класса; N - заданное общее число точек синтетической сцены; δ_i - корректирующая добавка, компенсирующая ошибку округления.

Корректирующая добавка вводилась для выполнения условия:

$$\sum_{i=1}^C n_i = N, \quad (3)$$

Для учета масштаба моделируемой территории площадь сцены в гектарах определялась выражением:

$$A_{ca} = \frac{WL}{10^4}, \quad (4)$$

где A_{ca} - площадь синтетической сцены в гектарах; W - ширина моделируемой области, м; L - длина моделируемой области, м.

Ожидаемое количество объектов i -го класса задавалось формулой:

$$E_i = \max(m_i, \rho_i A_{ca}), \quad (5)$$

где E_i - ожидаемое количество объектов i -го класса; m_i - минимально допустимое число объектов i -го класса; ρ_i - плотность объектов i -го класса на один гектар.

Результаты

В результате последовательного применения базового обучения, обучения с учетом дисбаланса классов и дополнения обучающей выборки синтетическими данными были получены следующие значения IoU по классам и среднего mIoU (табл. 2).

Таблица 2

Результаты сегментации при различных подходах к компенсации дисбаланса классов

Класс	Базовое обучение, IoU, %	Обучение с учетом дисбаланса, IoU, %	Прирост после учета дисбаланса, п.п.	Обучение с синтетическими данными, IoU, %	Прирост после добавления синтетических данных, п.п.	Общий прирост, п.п.
1	65,49	70,77	+5,28	70,41	-0,36	+4,92
2	45,45	69,33	+23,88	78,31	+8,98	+32,86
3	41,77	46,61	+4,84	56,66	+10,05	+14,89
4	29,47	44,28	+14,81	63,40	+19,12	+33,93
5	47,81	72,64	+24,83	88,79	+16,15	+40,98
6	13,21	17,07	+3,86	26,12	+9,05	+12,91
7	2,51	3,60	+1,09	3,20	-0,40	+0,69
8	7,10	37,56	+30,46	52,11	+14,55	+45,01
mIoU	31.60	45.23	+13,63	54,88	+9,65	+23,28

Обсуждение

Анализ результатов показал, что учет дисбаланса классов на этапе обучения заметно улучшает качество сегментации по сравнению с базовым вариантом. Среднее значение mIoU увеличилось с 31,60 % до 45,23 %, что соответствует приросту на 13,63 п.п. Наиболее выраженное улучшение наблюдалось для классов 2, 5 и 8, что указывает на положительное влияние балансировки обучающих выборок и взвешивания функции потерь на распознавание менее устойчивых классов. Добавление искусственно сегментированных обучающих данных также улучшило показатели среднего значения mIoU до 54,88.

Заключение

Таким образом, результаты исследования показывают, что настройка параметров обучения и синтетическое пополнение обучающей выборки обладают взаимодополняющим эффектом. При этом сохранение низких значений качества по метрикам для отдельных классов указывает на необходимость дальнейшего совершенствования процедуры генерации синтетических данных и более точной настройки их распределения по классам.

Благодарности

Исследование выполнено при финансовой поддержке Кубанского научного фонда в рамках проекта № ЛАБ-24.1/2.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гура Д. А. Возможности использования технологий 3D-идентификации объектов недвижимости в автоматизированных системах для обнаружения самовольного строительства / Д. А. Гура, Т. А. Тихонов, К. В. Зеленская, К.О. Степаненко, Е.А. Шаркова // Московский экономический журнал. 2026. Т. 11, № 1. С. 116-139. DOI: 10.55186/2413-046X-2026-116-139.
2. Гура Д. А. Возможность внедрения модуля трехмерной идентификации объектов недвижимости в сервисы национальной системы пространственных данных / Д. А. Гура, М. М. Велибекова, Т. А. Тихонов // Региональные геосистемы. 2025. Т. 49, № 2. С. 282-293. DOI: 10.52575/2712-7443-2025-49-2-282-293.
3. Гура Д.А. Методика использования воздушного лазерного сканирования и нейросетевых технологий для обеспечения достоверными пространственными данными кадастра недвижимости / Д. А. Гура, Р. А. Дьяченко, С. В. Самарин, Д. А. Беспятчук // Геодезия и картография. 2025. Т. 86, № 6. С. 54-63. DOI: 10.22389/0016-7126-2025-1020-6-54-63.
4. Armeni I., Sener O., Zamir A. R., Jiang H., Brilakis I. K., Fischer M., Savarese S. 3D Semantic Parsing of Large-Scale Indoor Spaces // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 1534-1543. DOI: 10.1109/CVPR.2016.170.
5. Hackel T., Savinov N., Ladicky L., Wegner J. D., Schindler K., Pollefeys M. Semantic3D.net: A New Large-Scale Point Cloud Classification Benchmark // ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2017. Vol. IV-1/W1. P. 91-98. DOI: 10.5194/isprs-annals-IV-1-W1-91-2017.
6. Behley J., Garbade M., Milioto A., Quenzel J., Behnke S., Stachniss C., Gall J. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences // Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2019. P. 9296-9306. DOI: 10.1109/ICCV.2019.00939.
7. Tan W., Qin N., Ma L., Li Y., Du J., Cai G., Yang K., Li J. Toronto-3D: A Large-Scale Mobile LiDAR Dataset for Semantic Segmentation of Urban Roadways // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2020. DOI: 10.1109/CVPRW50498.2020.00109.
8. Lin H.-I., Nguyen M. C. Boosting Minority Class Prediction on Imbalanced Point Cloud Data // Applied Sciences. 2020. Vol. 10, No. 3. Art. 973. DOI: 10.3390/app10030973.
9. Li M., Lin S., Wang Z., Shen Y., Zhang B., Ma L. Class-Imbalanced Semi-Supervised Learning for Large-Scale Point Cloud Semantic Segmentation via Decoupling Optimization // Pattern Recognition. 2024. Vol. 156. Art. 110701. DOI: 10.1016/j.patcog.2024.110701.

© Д. А. Гура, Р. А. Дьяченко, Т. А. Тихонов, 2026

Р. А. Данилов^{1✉}, Е. Ю. Воронкин¹

Унификация полевой документации тахеометрической съёмки как основа автоматизированной 3D-визуализации археологических раскопок

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: romangg412@gmail.ru

Аннотация. В современной археологической среде актуальной проблемой выступает отсутствие единых стандартов ведения полевой документации. Это существенно затрудняет автоматизацию камеральной обработки данных, снижает скорость анализа и делает практически невозможным сопоставление материалов, полученных разными экспедициями. Для решения данной проблемы представлена программа, разработанная на языке Python. Она по результатам тахеометрической съёмки автоматически строит наглядные 2D- и 3D-модели археологического раскопа. В статье на основе практической апробации программного продукта предлагаются минимальные правила унификации документации. Они включают внедрение единой системы идентификаторов и стандартизированного формата файлов. Следование этим правилам позволяет полностью автоматизировать камеральную обработку и формировать сопоставимые базы данных для последующих археологических исследований.

Ключевые слова: тахеометрическая съёмка, автоматизация, 3D-моделирование, полевой стандарт, археологическая документация, Python

R. A. Danilov^{1✉}, E. Yu. Voronkin¹

Unification of field documentation of total station surveys as the Ba-sis for automated 3D visualization of archaeological excavation

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: romangg412@gmail.ru

Abstract. In the modern archaeological environment, a pressing issue is the lack of unified standards for maintaining field documentation. This significantly complicates the automation of laboratory data processing, reduces the speed of analysis, and makes it nearly impossible to compare materials obtained from different expeditions. To solve this problem, a program developed in Python is presented. Based on the results of tacheometric surveys, it automatically constructs clear 2D and 3D models of an archaeological excavation. Drawing on the practical testing of the software product, this article proposes minimal rules for documentation unification. These include the introduction of a unified identifier system and a standardized file format. Adhering to these rules makes it possible to fully automate laboratory processing and create comparable databases for subsequent archaeological research.

Keywords: tacheometric survey, automation, 3D modeling, field standard, archaeological documentation, Python

Введение

Современные археологические исследования невозможны без инструментальных съёмок: тахеометры, GNSS-приёмники, лазерные сканеры генерируют тысячи точек, которые ложатся в основу планов, разрезов и трёхмерных моделей [1]. Преимуществом тахеометра по сравнению с традиционными средствами инструментальной фиксации является то, что вся информация сохраняется и отображается в электронном виде, что позволяет автоматизировать и ускорить операции при последующих действиях по построению ЦМР и 3D реконструкций. Современные тахеометры сохраняют результаты съёмки в виде координат измеренной точки в текстовом формате txt [2].

Исследование предметной области

В настоящее время разработан и внедрен целый ряд эффективных геоинформационных систем (ГИС), позволяющих визуализировать результаты археологических исследований на основе тахеометрической съёмки. Положительный зарубежный и отечественный опыт получен как с обращением к материалам конкретных памятников, так при создании геоинформационных систем национального масштаба [3].

Однако анализ материалов, полученных разными экспедициями, показывает, что форматы данных, системы кодирования и даже разделители десятичных знаков различаются от полевого сезона к сезону, от одного исполнителя к другому. Это делает невозможной автоматическую обработку, вынуждая археологов вручную править файлы или заново оцифровывать планы. Отсутствие единого стандарта затрудняет создание сводных баз данных и сравнительных исследований.

Методы и технологии

В ответ для обозначенного круга проблем, с которыми сталкиваются археологи, была разработана и апробирована специализированная программа на языке Python, которая автоматически строит 2D/3D-модели раскопа по данным тахеометрической съёмки, и на её основе сформулировать минимальные, но достаточные правила для полевой документации, обеспечивающие автоматизацию камеральной обработки.

Данная программа выполняет следующие функции:

1. Классификация точек по семантическим группам (контуры, уровни зачистки, дно, находки) на основе идентификаторов (например, PER – контур, DNO – дно, SK – каменный артефакт и т. п.).
2. Построение замкнутых контуров объектов (выпуклая оболочка) для точек контура.
3. Создание вертикальных профилей – полупрозрачных стенок, соединяющих верхнюю и нижнюю границы контура, что наглядно показывает глубину залегания.
4. Визуализация в виде 2D-карт, 3D-scatter-графиков и 3D-моделей с закрашенными профилями.

5. Экспорт результатов в форматы GeoJSON/Shapefile для дальнейшего анализа в ГИС.

Апробация была проведена на материалах раскопа Р7 (уровни М1 и М2). Программа за несколько минут построила стратиграфическую модель, выявила закономерности залегания находок и перепады высот, что при ручной обработке заняло бы несколько часов.

Для того, чтобы любая подобная программа могла обрабатывать данные без ручной предобработки, рекомендуется при съёмке придерживаться следующих правил:

1. Система идентификаторов (кодов)

В поле PointID или отдельном поле Code указывать тип точки с помощью префиксов:

- CONT – контур объекта (ямы, жилища, раскопа);
- LEVn – уровень зачистки, где n – номер уровня;
- BOT – дно объекта;
- FIND_ – находки с указанием материала: FIND_CER (керамика), FIND_BON (кость), FIND_STO (камень), FIND_MET (металл);
- STR – точки стратиграфического разреза (стенки).

2. Порядок съёмки контуров

Точки одного контура снимать последовательно, по часовой стрелке или против, замыкая линию (последняя точка может не совпадать с первой, программа замкнёт автоматически, но последовательность важна для корректной геометрии).

3. Формат файла

Сохранять данные в текстовый файл с разделителем «;» и десятичной точкой (рекомендуется CSV). Обязательные колонки: идентификатор точки, X, Y, Z, код (если есть). Комментарии и лишние колонки допускаются, но не должны мешать чтению.

4. Документирование

В полевом дневнике фиксировать расшифровку всех использованных кодов и примечания (например, «точки с кодом LEV2 – второй уровень зачистки в яме 1»). Это позволит позже адаптировать правила под конкретный памятник.

На данных Р7, где идентификаторы частично соответствовали предложенной системе (PER – контур, DNO – дно, SK – каменные находки), программа корректно классифицировала более 95% точек и построила наглядные модели. Незначительные отклонения (например, наличие кода 111 во всех точках) не мешали работе, так как основная информация извлекалась из PointID.

Результаты

Разработанная программа может служить прототипом для создания единого цифрового инструментария археолога, а внедрение стандарта позволит создавать сопоставимые базы данных по разным памятникам, автоматически выявлять общие закономерности и ускорит подготовку отчётной документации а) вкладка

«Основное», добавление замера глюкозы; б) вкладка «Статистика», вывод таблиц; в) вкладка «Статистика», вывод графиков

Заключение

Таким образом, унификация полевой документации – первый шаг к полной автоматизации камеральной обработки, оперативное проведение которой, позволяет выявить ошибки еще в поле, а также представить целостную картину раскопа [1]. Предложенные правила минимальны, легко запоминаются и не требуют дополнительного оборудования. К тому же, разработанный подход к 3D-визуализации согласуется с современными тенденциями, описанными, к примеру, в методике полевой фиксации Институтом истории материальной культуры РАН [4].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Зайцева, О. В. Тахеометрическая съемка в археологических исследованиях: учеб.-метод. Пособие / О. В. Зайцева, А. А. Пушкарев. – 2010. – 48 с.
2. Золотухин, В. В. Цифровое геодезическое оборудование в археологических исследованиях с применением методов геоинформационных систем / В. В. Золотухин / Вестник Новосибирского государственного университета. – 2011. – №7. – С. 61–66.
3. Макаров, Н. А. Первые шаги по созданию Национальной географо-информационной системы «Археологические памятники России» / Н. А. Макаров, А. Н. Ворошилов, О. В. Зеленцова, Д. С. Коробов, А. П. Черников / Археология, этнография и антропология Евразии. – 2015. – Том 43–№ 4. – С. 86–93.
4. Основные методики полевой фиксации, применяемые ИИМК РАН [Электронный ресурс]. URL: <https://www.old.archeo.ru/struktura-1/otdel-ohrannoi-arheologii/nauchnye-proekty-otdela-ohrannoi-arheologii/foto-k-metodikam-iimk> (дата обращения: 02.03.2026).

© Р. А. Данилов, Е. Ю. Воронкин, 2026

Д. Р. Дарижапова¹✉, Д. Ю. Смирнов¹

Автоматизация оценки платежеспособности заемщиков на платформе «1С: Предприятие 8.3» с применением технологий искусственного интеллекта

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: darizhapova.darya@mail.ru

Аннотация. В статье представлены результаты проектирования и внедрения автоматизированной системы оценки кредитоспособности заемщиков. Разработка выполнена на платформе «1С:Предприятие 8.3» с интеграцией аналитического модуля на языке Python на примере сельскохозяйственного потребительского кредитного кооператива (СПКК) «Нива». Проведен подробный анализ предметной области, выявлены недостатки существующих экспертных подходов оценки рисков. Разработана специализированная конфигурация 1С, включающая учет договоров, графиков платежей и регистры взаиморасчетов. В рамках построения прогнозной модели проведено обучение и сравнительный анализ трех алгоритмов машинного обучения на основе исторического массива данных кооператива. В качестве оптимальной модели выбран метод градиентного бустинга XGBoost, показавший наибольшую точность. Интеграция систем выполнена по локальному файловому обмену в формате JSON. Разработанное решение позволяет в реальном времени рассчитывать индивидуальную вероятность дефолта заемщика и интерпретировать ключевые факторы риска.

Ключевые слова: 1С: Предприятие, машинное обучение, кредитный риск, автоматизация, JSON, кредитный скоринг

D. R. Darizhapova¹✉, D. Yu. Smirnov¹

Automation of borrower solvency assessment on the "1C: Enterprise 8.3" platform using artificial intelligence technologies

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: darizhapova.darya@mail.ru

Abstract. The article presents the results of designing and implementing an automated system for assessing the creditworthiness of borrowers. The development was carried out on the "1C: Enterprise 8.3" platform with the integration of an analytical module in Python, using the agricultural consumer credit cooperative (SECC) "Niva" as an example. A detailed analysis of the subject area was carried out, and the shortcomings of existing expert risk assessment approaches were identified. A specialized 1C configuration was developed, including accounting for contracts, payment schedules, and mutual settlement registers. As part of building the predictive model, training and comparative analysis of three machine learning algorithms were performed based on the historical data of the cooperative. The gradient boosting method XGBoost was selected as the optimal model, showing the highest accuracy. The systems integration is performed via local JSON file exchange. The developed solution allows real-time calculation of the individual default probability of the borrower and interpretation of key risk factors.

Keywords: 1C: Enterprise, machine learning, credit risk, automation, JSON, credit scoring

Введение

Оценка кредитоспособности физических лиц является одной из приоритетных задач для малых финансовых институтов, таких как сельскохозяйственные потребительские кредитные кооперативы (СПКК) [1]. В отличие от крупных коммерческих банков, региональные кооперативы часто не имеют доступа к дорогостоящим промышленным скоринговым системам. Принятие решений в таких организациях нередко базируется на экспертном методе или простейших балльных картах. Это приводит к росту влияния человеческого фактора, снижению скорости обработки заявок и повышению уровня просроченной задолженности.

Современные методы машинного обучения позволяют решать задачи кредитного скоринга с высоким уровнем точности за счет анализа многофакторных нелинейных зависимостей [2]. При этом ключевой проблемой остается внедрение этих моделей в повседневную практику работы кредитных специалистов. Полная замена существующего программного обеспечения является экономически нецелесообразной. Наиболее перспективным решением представляется создание гибридной архитектуры. В такой системе привычная для сотрудников учетная среда выступает интерфейсом ввода и хранения данных, а сложные аналитические вычисления выносятся во внешний специализированный модуль [3].

Целью данного исследования является разработка и практическая реализация автоматизированной информационной системы оценки платежеспособности заемщиков СПКК «Нива» посредством интеграции платформы «1С: Предприятие 8.3» и предиктивных алгоритмов машинного обучения на языке Python [4].

Исследование предметной области

Сельскохозяйственный потребительский кредитный кооператив «Нива» (СПКК «Нива») является типичным представителем малых финансовых учреждений, работающих в сельской местности. Основными видами деятельности кооператива являются выдача займов физическим лицам – пайщикам кооператива – для развития личных подсобных хозяйств, приобретения сельскохозяйственной техники, строительства и ремонта жилья, а также на неотложные нужды [5].

Бизнес-процесс выдачи займа включает сбор анкетных и финансовых данных заемщика, расчет параметров кредита, формирование графика платежей и принятие решения о выдаче [6]. На практике существующий процесс обладает рядом недостатков: высокой долей просроченной задолженности, длительным рассмотрением заявки, сильным влиянием человеческого фактора и субъективности при принятии решения, а также отсутствием единой формализованной методики оценки риска.

В ходе анализа предметной области были рассмотрены существующие подходы к оценке кредитного риска. Первый подход – экспертный метод, основанный на ручном анализе специалистом. Его сильными сторонами являются гибкость и возможность учета индивидуальных нюансов каждого заемщика. Однако

к слабым сторонам можно отнести субъективность, низкую скорость обработки заявок и высокую нагрузку на сотрудников.

Второй подход – кредитный скоринг, используемый в банковских системах. Он обеспечивает высокую точность оценки, формализованность процесса и высокую скорость обработки. Но при этом он обладает высокой стоимостью внедрения и сложностью адаптации под специфику конкретного кредитного кооператива.

Третий подход – использование стандартных учетных систем. Их сильными сторонами являются надежность хранения данных и привычный для сотрудников интерфейс. Однако они не имеют встроенных инструментов прогнозирования и искусственного интеллекта [7].

Таким образом, актуальной научно-практической задачей является интеграция предиктивной модели машинного обучения непосредственно в учетную систему без переноса персональных данных во внешние сети и без замены привычного для сотрудников программного обеспечения [8].

Методы и технологии

Для автоматизации процессов учета кредитных договоров в сельскохозяйственном кредитном потребительском кооперативе (СКПК) «Нива» была выбрана платформа «1С: Предприятие 8.3». Деятельность таких организаций строго регламентируется профильным законодательством, что требует высокой надежности информационных систем [9]. В рамках работы в среде «1С» спроектирована специализированная подсистема «Кредитный ПотребКооператив», содержащая справочники, перечисления, документы и аналитические отчеты. Справочник «Физические лица» используется для надежного хранения анкетных и финансовых данных заемщиков организации [10]. Центральным объектом разработанной конфигурации выступает справочник «Договоры», который содержит полные реквизиты договора и табличную часть для автоматического формирования помесечного графика платежей. В модуле этого документа реализована функция расчета суммы ежемесячного платежа, поддерживающая аннуитетный и дифференциальный типы начислений, причем при изменении параметров кредита программа автоматически подбирает необходимую процентную ставку [11]. За счет справочника «Договоры» обеспечивается оперативная фиксация всех первичных данных заемщика, а на его управляемой форме настроена индикация уровня дефолтного риска для повышения наглядности работы сотрудника.

Для построения предиктивной модели оценки рисков использован язык программирования Python [12]. Информационной базой для математического моделирования стал исторический деперсонализированный массив данных, содержащий 100 000 уникальных записей кредитных историй [13]. Вектор входных признаков заемщика включает в себя его возраст, годовой доход, официальный статус занятости, стаж работы, общую сумму и срок займа. Также алгоритмами учитываются тип платежа, процентная ставка, текущее семейное положение, количество иждивенцев, месячный доход, размер ежемесячного платежа, коэффициент соотношения платежа к доходу и месяц подачи заявки. При этом целевой

признак в обучающей выборке жестко фиксирует факт возникновения дефолта [14,15]. Обучение и сравнительный анализ моделей проводились с помощью библиотек Scikit-learn [16] и XGBoost [17]. На этапе предварительной обработки сырых данных выполнялось обязательное масштабирование числовых признаков и кодирование категориальных параметров. В ходе исследования тестирование прошли три популярных алгоритма классификации, а именно логистическая регрессия, случайный лес и XGBoost. Наилучшие результаты по всем целевым метрикам точности показал алгоритм градиентного бустинга XGBoost. Данная модель была выбрана в качестве основного предиктивного ядра для последующей интеграции с учетной системой кооператива. Специальный скрипт прогнозирования с именем predict.py загружает сохраненную модель, считывает переданные параметры и рассчитывает точную вероятность дефолта. Дополнительно с помощью алгоритма SHAP определяются ключевые факторы риска для каждого заявителя, а сам скрипт ведет подробный текстовый лог всех этапов работы для оперативного контроля корректности вычислений [18, 19].

Взаимодействие между информационной базой «1С» и внешним Python-сегментом реализовано локально через файлы обмена в формате JSON. При проведении документа в «1С» автоматически формируется файл input.json с полным набором признаков конкретного заемщика. Затем платформа «1С» с помощью встроенной функции ЗапускИИ() вызывает скрипт прогнозирования в фоновом режиме. Скрипт выполняет расчет и записывает итоговый результат в файл output.json. После этого система «1С» считывает данные из файла ответа и выводит расчетную вероятность дефолта и текстовые факторы риска прямо на форму справочника сотруднику. Такой подход позволяет эффективно внедрить методы машинного обучения в привычную рабочую среду сотрудников СКПК «Нива». При этом все вычисления выполняются исключительно на локальном компьютере, что гарантирует безопасность персональных данных и полностью исключает зависимость от сторонних интернет-сервисов [20].

Результаты

В результате работы создана автоматизированная информационная система прогнозирования платежеспособности заемщиков на платформе «1С: Предприятие 8.3» [21]. Разработанная система обеспечивает следующие возможности:

- ввод и хранение полных анкетных и финансовых данных о заемщике с контролем целостности (автоматический расчет годового дохода);
- создание договоров с автоматическим расчетом графика платежей как для аннуитетной, так и для дифференцированной схемы (рис. 1);
- автоматический подбор типа кредита и процентной ставки в зависимости от указанного заемщиком срока;
- расчет долговой нагрузки заемщика с выводом предупреждения при критическом значении;
- прогнозирование вероятности дефолта одним нажатием кнопки «Прогноз ИИ» (рис. 1);

- отображение ключевых факторов, повлиявших на прогноз;
- автоматическое сохранение результата прогноза в карточке договора;
- формирование наглядного отчета «Анализ платежеспособности» с цветовой индикацией уровня риска (рис. 2);
- полное документирование всех операций выдачи, начисления процентов и поступления платежей;
- ведение регистра остатков взаиморасчетов для контроля текущей задолженности;
- логирование всех вызовов ИИ.

Скриншот программного интерфейса. В центре — форма «Договор № 000000002 от 01.01.2026 (Фермеров Семен Сидорович)». В левом меню — «Договоры». В нижней части — таблица платежей.

Номер платежа	Дата платежа	Платеж осн. долга	Платеж проценты	Совокуп. платеж
4	01.05.2026	15 965,98	2 560,92	18 526,90
5	01.06.2026	16 232,08	2 294,82	18 526,90
6	01.07.2026	16 502,61	2 024,29	18 526,90
7	01.08.2026	16 777,65	1 749,25	18 526,90
8	01.09.2026	17 057,28	1 469,62	18 526,90
9	01.10.2026	17 341,57	1 185,33	18 526,90
Итого	по 01.11.2026	17 630,00	900,00	18 530,00
		200 000,00	22 322,81	222 322,81

Сообщения:
 ✓ Прогноз выполнен! Вероятность риска: 79,49%

Рис. 1. Договор и результат работы ИИ

Скриншот программного интерфейса. В центре — отчет «Анализ платежеспособности». В нижней части — таблица с данными о заемщиках и их риске.

№	Заемщик	Сумма	Платеж	Нагрузка	Риск	Статус
000000001	Иванова Анна Александровна	100 000	6 655,091111	15,28	1,42	НОРМА
000000002	Фермеров Семен Сидорович	200 000	18 526,900833	37,05	79,49	КРИТИЧЕСКИЙ
000000003	Петров Петр Петрович	120 000	7 986,110526	36,67	75,2	КРИТИЧЕСКИЙ
000000004	Иванов Сергей Иванович	280 000	18 634,257778	35,65	33,95	НОРМА
000000005	Сергеев Сергей Сергеевич	275 000	25 474,488333	36,39	38,79	НОРМА

Рис. 2. Отчет «Анализ платежеспособности» с цветовой индикацией цвета

Заключение

В результате работы разработана автоматизированная информационная система для прогнозирования платежеспособности заемщиков на платформе «1С:Предприятие 8.3» с интеграцией модели машинного обучения. Система позволяет вести учет договоров, рассчитывать графики платежей и автоматически получать прогноз вероятности дефолта. Реализованный функционал направлен на повышение объективности оценки заемщиков, снижение кредитного риска и ускорение обработки заявок в СПКК «Нива» [22].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Федерация. Законы. О кредитной кооперации : Федеральный закон № 190-ФЗ : [принят Государственной Думой 18 июля 2009 г.] – URL: http://www.consultant.ru/document/cons_doc_LAW_90058/ (дата обращения: 10.05.2026).
2. Стародубцева Е. Б., Маркова О. М. Использование искусственного интеллекта в кредитном скоринге // Финансовые рынки и банки. – Москва, 2024. – №10. – С. 162–166.
3. Российская Федерация. Законы. О персональных данных : Федеральный закон № 152-ФЗ : [принят Государственной Думой 27 июля 2006 г.] – URL: http://www.consultant.ru/document/cons_doc_LAW_61801/ (дата обращения: 10.05.2026).
4. Автоматизация деятельности кредитной организации на платформе «1С:Предприятие 8» : учебное пособие / Д. В. Чистов, Я. Л. Гобарева, А. В. Золотарюк [и др.] ; под ред. Д. В. Чистова. – М. : 1С-Паблишинг, 2012. – 436 с.
5. Российская Федерация. Законы. О потребительском кредите (займе) : Федеральный закон № 353 ФЗ : [принят Государственной Думой 23 декабря 2013 г.] – URL: http://www.consultant.ru/document/cons_doc_LAW_155986/ (дата обращения: 15.05.2026). – Текст : электронный.
6. Чистов Д. В. Банковские информационные системы : учебное пособие / Д. В. Чистов. – М.: Финансы и статистика, 2005. – 384 с.
7. Сервис 1С:Кредит. Официальная документация и описание возможностей // Фирма «1С» : официальный сайт – URL: <https://v8.1c.ru/tekhnologii/obmen-dannymi-i-integratsiya/realizovannye-resheniya/1s-kredit/> (дата обращения: 10.05.2026). – Текст : электронный.
8. Кредитный скоринг, реализованный с помощью машинного обучения С. М. Воронин [и др.] // Столыпинский вестник. 2022. №10. URL: <https://cyberleninka.ru/article/n/kreditnyy-skoring-realizovannyu-s-pomoschyu-mashinnogo-obucheniya> (дата обращения: 12.05.2026). – Текст : электронный.
9. ГОСТ Р 57580.1-2017. Безопасность финансовых (банковских) операций. Защита информации финансовых организаций. Базовый состав организационных и технических мер. – М. : Стандартинформ, 2017. – URL: <https://regulhub.kaspersky.ru/>. (дата обращения: 10.05.2026). – Текст : электронный.
10. Хрусталева Е. Ю. Разработка сложных отчетов в «1С:Предприятии 8». Система компоновки данных – М. : 1С-Паблишинг, 2019. – 300 с.
11. Радченко М. Г. 1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы – М. : 1С-Паблишинг, 2021. – 982 с.
12. Лутц М. Изучаем Python. Т. 1. – СПб.: Диалектика, 2019. – 832 с.
13. McKinney, W. Data Structures for Statistical Computing in Python // Proceedings of the 9th Python in Science Conference (SciPy 2010). – Austin, 2010. – P. 56–61.
14. Программная модель оценки кредитоспособности клиентов с применением алгоритмов искусственного интеллекта / А. Г. Шмелева [и др.]// Труды НГТУ им. Р. Е. Алексеева. 2020. №3 (130). URL: <https://cyberleninka.ru/article/n/programmaya-model-otsenki-kreditosposobnosti-klientov-s>

primeneniem-algoritmov-iskusstvennogo-intellekta (дата обращения: 15.05.2026). – Текст : электронный.

15. Сидоренко М. Г. Машинное обучение в задачах кредитного скоринга: обзор методов и моделей // Искусственный интеллект и принятие решений. – 2022. – № 3. – С. 56–68.

16. Scikit learn: Machine Learning in Python / F. Pedregosa, G. [et al.] // Journal of Machine Learning Research. – 2011. – Vol. 12. – P. 2825–2830.

17. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. – San Francisco, 2016. – P. 785–794.

18. Smith, J. R. Credit Risk Modeling Using Machine Learning Techniques: A Comparative Study // Journal of Financial Data Science. – 2022. – Vol. 8, No. 3. – P. 210–225.

19. Маккинни, У. Python и анализ данных / пер. с англ. А. А. Слинкина. – М. : ДМК Пресс, 2020. – 482 с.

20. Документация по платформе 1С:Предприятие 8.3. Взаимодействие с внешними приложениями // Фирма «1С» : официальный сайт – URL: <https://its.1c.ru/> (дата обращения: 12.05.2026). – Текст : электронный.

21. Документация по платформе 1С:Предприятие 8.3 [Электронный ресурс]. – URL: <https://its.1c.ru/db/v8321doc#browse:13:-1:5> (дата обращения: 10.05.2026). – Текст : электронный.

22. Абдуллаев, Н. А. Перспективы внедрения современных технологий искусственного интеллекта в скоринговые системы // Экономика и финансы (Узбекистан). – 2023. – №1 (161).

– URL: <https://cyberleninka.ru/article/n/perspektivy-vnedreniya-sovremennyh-tehnologiy-iskusstvennogo-intellekta-v-skoringovye-sistemy> (дата обращения: 10.05.2026). – Текст : электронный.

© Д. Р. Дарижапова, Д. Ю. Смирнов, 2026

Д. Д. Догадин^{1✉}, Д. Ю. Смирнов¹

Разработка защищенного интеграционного шлюза между облачной CRM-системой и локальной учетной платформой

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: dogadin.danil@list.ru

Аннотация. В статье рассматривается проблема интеграции облачных систем управления взаимоотношениями с клиентами (CRM) и локальных учетных платформ на базе «1С: Предприятие 8.3» в условиях отсутствия выделенного статического IP-адреса и строгих требований информационной безопасности. Предложен архитектурный подход, основанный на использовании промежуточного проксирующего сервера и криптографического VPN-туннеля WireGuard для безопасной маршрутизации входящих вебхуков. Разработан кастомный HTTP-сервис на встроенном языке 1С, реализующий асинхронную модель обработки данных. Описаны алгоритмы автоматического создания нормативно-справочной информации, формирования бухгалтерских проводок по счетам реализации и зачета авансов, а также программный обход лицензионных ограничений учебной версии платформы путем динамической генерации транзакционных HTML-уведомлений. Результаты исследования подтверждают высокую эффективность предложенного решения: время полного цикла обработки клиентского заказа сокращено более чем в 400 раз, при этом локальная вычислительная сеть предприятия остается полностью изолированной от глобальной сети Интернет. Предложенные методы могут быть масштабированы для интеграции гетерогенных корпоративных информационных систем малого и среднего бизнеса.

Ключевые слова: системная интеграция, 1С: Предприятие, Bitrix24, VPN WireGuard, автоматизация бизнес-процессов, REST API

D. D. Dogadin^{1✉}, D. Yu. Smirnov¹

Development of a secure integration gateway between a cloud CRM system and a local accounting platform

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: dogadin.danil@list.ru

Abstract. The article deals with the problem of integration of cloud-based customer relationship management (CRM) systems and local accounting platforms based on «1C: Enterprise 8.3» in the absence of a dedicated static IP address and strict information security requirements. An architectural approach based on the use of an intermediate proxy server and a cryptographic WireGuard VPN tunnel for secure routing of incoming webhooks is proposed. A custom HTTP service in the built-in 1C language implementing an asynchronous data processing model was developed. The algorithms of automatic creation of master data, formation of accounting entries for sales and advance offset accounts, as well as programmatic circumvention of licensing restrictions of the educational version of the platform by dynamic generation of transactional HTML notifications are described. The results of the research confirm the high efficiency of the proposed solution: the time of the full cycle of processing a client order is reduced by more than 400 times, while the local area network of the enterprise remains completely isolated from the global Internet. The proposed methods can be scaled up for integration of heterogeneous corporate information systems of small and medium-sized businesses.

Keywords: system integration, 1C: Enterprise, Bitrix24, VPN WireGuard, business process automation, REST API

Введение

Современный этап цифровизации малого и среднего бизнеса характеризуется массовым переходом к использованию облачных систем управления взаимоотношениями с клиентами (CRM). Однако финансовый и складской учет на территории Российской Федерации традиционно ведется в локальных версиях программных продуктов семейства «1С: Предприятие» [1]. Возникающий «цифровой разрыв» между фронт-офисом и бэк-офисом приводит к необходимости ручного дублирования информации, что снижает скорость обслуживания клиентов и повышает риск возникновения ошибок в бухгалтерских регистрах [2].

Существующие типовые модули синхронизации предъявляют жесткие требования к сетевой инфраструктуре предприятия: они требуют наличия «белого» статического IP-адреса и публикации базы данных в глобальной сети путем открытия портов на маршрутизаторе [3]. В условиях роста киберугроз и атак программ-вымогателей, публикация финансовой базы данных в открытом интернете является критической уязвимостью. Кроме того, типовые решения часто несовместимы с базовыми конфигурациями (например, «Бухгалтерия предприятия 3.0») без приобретения дополнительных платных расширений [4].

Целью данного исследования является разработка и практическая реализация отказоустойчивой и защищенной инфраструктуры автоматического обмена данными между облачной CRM-системой Bitrix24 и локальной учетной базой «1С: Бухгалтерия 8.3», исключающей необходимость компрометации локальной корпоративной сети.

Методы и материалы

Для реализации поставленной цели применялись методы системного анализа, сетевой маршрутизации и объектно-ориентированного программирования. Разработанный интеграционный модуль базируется на гибридной событийно-ориентированной архитектуре (Event-driven architecture).

В качестве источника данных выступает облачная CRM Bitrix24, инициирующая отправку сигнальных вебхуков (Webhooks) при изменении стадий сделки [5]. Для обеспечения безопасности транспортного уровня была спроектирована распределенная сеть. Внешний трафик перехватывается облачным сервисом Cloudflare, выполняющим функции межсетевого экрана веб-приложений (WAF) и обеспечивающим SSL/TLS-шифрование [6]. Отфильтрованные запросы поступают на арендованный виртуальный выделенный сервер (VPS) под управлением ОС Ubuntu Linux.

В качестве Reverse-прокси выступает веб-сервер Nginx, который перенаправляет HTTP-запросы в защищенный L3-туннель. Туннелирование реализовано с помощью криптографического протокола WireGuard [7]. Использование данного протокола позволило организовать прямую сетевую видимость между облачным сервером и локальным ПК с базой 1С, полностью скрыв последний за

NAT-шлюзом провайдера. Графическая интерпретация спроектированной маршрутизации трафика представлена на схеме (рис. 1).

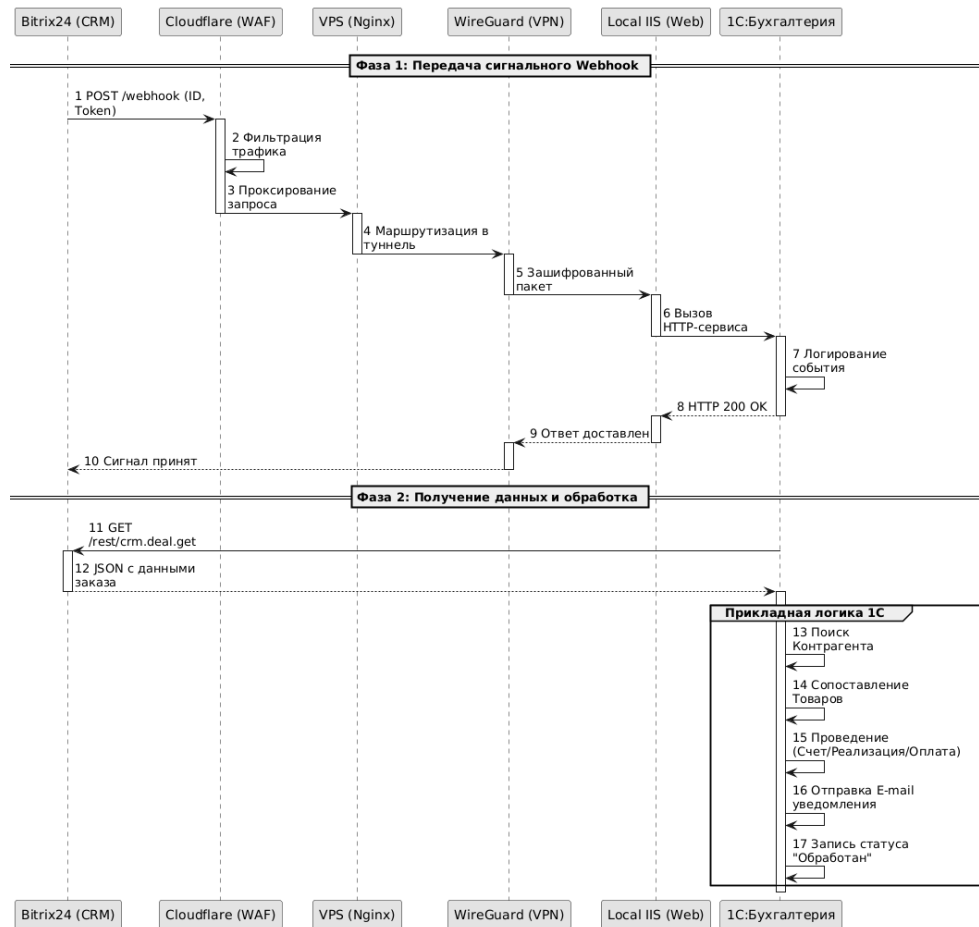


Рис. 1. Диаграмма последовательности обработки транзакции

На стороне платформы 1С реализован HTTP-сервис, принимающий POST-запросы. Обмен данными разделен на две асинхронные фазы во избежание сетевых таймаутов на стороне CRM [8]. Первая фаза включает валидацию секретного токена безопасности, буферизацию запроса в служебный регистр информационной базы и мгновенный возврат ответа со статусом «200 OK». Вторая фаза инициируется учетной системой самостоятельно: 1С выполняет исходящий защищенный запрос к REST API Bitrix24 для получения агрегированного JSON-пакета [9].

Для оценки эффективности разработанного интеграционного шлюза применяется математическая модель расчета сокращения операционных издержек. Вычисление коэффициента оптимизации времени обработки (K_{opt}) рассчитывается по формуле:

$$K_{opt} = \frac{\sum_{i=1}^n (T_{man} + T_{err})}{\sum_{i=1}^n T_{auto}}, \quad (1)$$

где T_{man} – время ручного переноса данных о заказе в учетную систему;

T_{err} – среднее время, затрачиваемое на поиск и устранение опечаток в первичных документах;

T_{auto} – процессорное время выполнения фонового задания по парсингу JSON и генерации объектов;

n – количество обрабатываемых заказов.

Результаты

Практическая реализация разработанных алгоритмов позволила полностью автоматизировать контур Order-to-Cash. При поступлении JSON-пакета программный модуль 1С выполняет сопоставление нормативно-справочной информации. В случае отсутствия контрагента, система программно инициализирует создание новой карточки и привязанного к ней договора. При этом для обеспечения ссылочной целостности бухгалтерских механизмов принудительно задаются валюта расчетов и ставка НДС [10].

Маршрутизация бизнес-событий реализована следующим образом:

Сигнал «Выставление счета» генерирует в 1С документ «Счет на оплату».

Сигнал «Отгрузка» формирует документ «Реализация товаров и услуг». Модуль автоматически устанавливает счета учета (41.01, 90.01.1, 90.02.1) и заполняет аналитику, что позволяет системе провести документ без участия бухгалтера, списав себестоимость товара.

Сигнал «Успешная сделка» инициирует создание документа «Поступление на расчетный счет». Внедренный алгоритм принудительно заполняет технические реквизиты курсов валют во избежание ошибки ядра СУБД (деление на ноль) и устанавливает параметр «Автоматическое погашение задолженности».

Сравнение показателей эффективности традиционного и автоматизированного подходов приведено в табл. 1.

Дополнительным результатом работы стала программная реализация генератора адаптивной HTML-разметки. В связи с ограничениями учебной версии платформы 1С на конвертацию табличных документов в формат PDF, разработан скрипт, использующий объект «ИнтернетПочтовоеСообщение» для отправки детализированных чеков и уведомлений непосредственно в теле электронного письма [11].

Таблица 1. Сравнительный анализ методов обработки заказов

Параметр оценки	Ручной перенос данных	Интеграционный шлюз (1С + Webhook)
Время обработки	От 10 до 15 минут	1–2 секунды
Вероятность ошибки ввода	Высокая	Исключена
Доступность базы данных	Высокий риск	Изолирована (защищенный туннель VPN WireGuard)
Информирование клиента	Ручная отправка E-mail	Автоматическая рассылка
Трудозатраты бухгалтера	Высокие	Нулевые

Обсуждение

Полученные в ходе нагрузочного тестирования данные подтверждают состоятельность выбранной архитектуры. Сравнение предложенного метода с типовыми решениями для обмена данными между сайтом и 1С выявило существенное преимущество в части безопасности инфраструктуры. Отказ от тяжеловесного протокола CommerceML в пользу точечных REST API запросов снизил нагрузку на вычислительные мощности локального сервера.

Проблема обработки «битых» данных от CRM была решена внедрением блоков Попытка...Исключение и строгим типизированием переменных, что соотносится с лучшими практиками разработки высоконагруженных систем [12]. Основным ограничением предложенного метода является необходимость администрирования промежуточного облачного VPS-сервера, однако затраты на его содержание кратно ниже стоимости устранения последствий потенциальной кибератаки на открытую базу предприятия.

Заключение

В рамках проведенного исследования разработана, протестирована и внедрена система двусторонней интеграции облачной CRM и учетной системы предприятия. Успешно решена проблема безопасного проксирования трафика в локальную сеть с помощью протокола WireGuard. Разработанный HTTP-сервис 1С обеспечил сквозную автоматизацию формирования бухгалтерских проводок, полностью исключив ручной труд. Коэффициент оптимизации времени обработки транзакции подтверждает многократное повышение эффективности операционной деятельности предприятия. Дальнейшие исследования будут направлены на реализацию двусторонней синхронизации складских остатков и внедрение механизмов мультиканальных уведомлений посредством API мессенджеров.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бизнес-процессы в CRM-системе [Электронный ресурс] // Журнал Битрикс24: сайт. – URL: <https://www.bitrix24.ru/journal/biznes-processy-v-crm-sisteme/> (дата обращения: 22.04.2024). – Текст: электронный.
2. Руководство разработчика. 1С: Предприятие 8.3 [Электронный ресурс] // Информационно-технологическое сопровождение (ИТС): электронно-библиотечная система. – URL: <https://its.1c.ru/db/v83doc> (дата обращения: 22.04.2024). – Режим доступа: для зарегистрир. пользователей. – Текст: электронный.
3. Радченко, М. Г. 1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы / М. Г. Радченко, Е. Ю. Хрусталева. – Москва: 1С-Паблишинг, 2020. – 926 с. – ISBN 978-5-9677-2900-5. – Текст: непосредственный.
4. Документация REST API Битрикс24 [Электронный ресурс] // Официальный портал разработчиков Bitrix24: сайт. – URL: https://dev.1c-bitrix.ru/rest_help/ (дата обращения: 22.04.2024). – Текст: электронный.
5. Локальные интеграции и исходящие вебхуки [Электронный ресурс] // Учебный курс Битрикс24: сайт. – URL: https://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=99 (дата обращения: 22.04.2024). – Текст: электронный.

6. Брандмауэр веб-приложений (WAF) Cloudflare [Электронный ресурс] // Документация разработчика Cloudflare: сайт. – URL: <https://developers.cloudflare.com/waf/> (дата обращения: 22.04.2024). – Текст: электронный.
7. Быстрый старт и настройка WireGuard [Электронный ресурс] // Официальный сайт проекта WireGuard: сайт. – URL: <https://www.wireguard.com/quickstart/> (дата обращения: 22.04.2024). – Текст: электронный.
8. Филдинг, Р. Архитектурные стили и дизайн сетевых программных архитектур [Электронный ресурс] / Р. Филдинг // Information and Computer Science: сайт. – URL: https://ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm (дата обращения: 22.04.2024). – Текст: электронный.
9. Брей, Т. Формат обмена данными JavaScript Object Notation (JSON). RFC 8259 [Электронный ресурс] / Т. Брей // RFC Editor: сайт. – URL: <https://www.rfc-editor.org/rfc/rfc8259> (дата обращения: 22.04.2024). – Текст: электронный.
10. Бухгалтерский учет в 1С:Предприятии 8 [Электронный ресурс] // Руководство по ведению учета. – URL: <https://its.1c.ru/db/acceng8> (дата обращения: 22.04.2024). – Режим доступа: для зарегистрир. пользователей. – Текст: электронный.
11. Габец, А. П. 1С: Предприятие 8.3. Простые примеры разработки / А. П. Габец, Д. И. Гончаров. – Москва: 1С-Публишинг, 2019. – 410 с. – Текст: непосредственный.
12. Клеппман, М. Высоконагруженные приложения. Программирование, масштабирование, поддержка / М. Клеппман; пер. с англ. – Санкт-Петербург: Питер, 2018. – 640 с. – ISBN 978-5-4461-0576-1. – Текст: непосредственный.

© Д. Д. Догадин, Д. Ю. Смирнов, 2026

Р. В. Енчу¹✉, А. А. Басаргин¹

Разработка веб-сервиса для мониторинга успеваемости с применением технологий искусственного интеллекта

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: yenchuroman@gmail.com

Аннотация. В статье представлена разработка веб-сервиса «КампусПлюс» для мониторинга успеваемости с применением технологий искусственного интеллекта. Актуальность работы обусловлена необходимостью автоматизации контроля знаний в образовательном процессе и устранения разрозненности учебных инструментов. Описан сравнительный анализ существующих платформ – Moodle, Google Forms и Stepik, выявивший отсутствие единого прикладного маршрута преподавателя и встроенных средств AI-генерации тестов. Представлены функциональная и информационно-логическая модели системы, архитектура прототипа на базе FastAPI, Jinja2 и OpenAI API, а также механизм нормализации AI-сгенерированных вопросов. Описаны результаты ручного, автоматизированного и нагрузочного тестирования сервиса. Практическая значимость работы заключается в формировании единого цифрового контура «дисциплина – лекция – тест – попытка – аналитика» и снижении трудоемкости сопровождения дисциплины для преподавателя.

Ключевые слова: мониторинг успеваемости, веб-сервис, искусственный интеллект, генерация тестов, образовательная аналитика, FastAPI, OpenAI API, контроль знаний

R. V. Enchu¹✉, A. A. Basargin¹

Development of a web service for academic performance monitoring using artificial intelligence technologies

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: yenchuroman@gmail.com

Abstract. The article presents the development of the CampusPlus web service for academic performance monitoring using artificial intelligence technologies. The relevance of the work stems from the need to automate knowledge assessment in the educational process and overcome the fragmentation of teaching tools. A comparative analysis of existing platforms – Moodle, Google Forms, and Stepik – is conducted, revealing the lack of an integrated instructor workflow and built-in AI-driven test generation. The paper presents the functional and information-logical models of the system, the prototype architecture based on FastAPI, Jinja2 and the OpenAI API, as well as the normalization mechanism for AI-generated questions. The results of manual, automated and load testing of the service are described. The practical significance of the work lies in establishing a unified digital pipeline «discipline – lecture – test – attempt – analytics» and reducing the workload of teachers in supporting a course.

Keywords: academic performance monitoring, web service, artificial intelligence, test generation, educational analytics, FastAPI, OpenAI API, knowledge assessment

Введение

Современный образовательный процесс в высших учебных заведениях по-прежнему требует от преподавателя значительных временных затрат на подготовку контрольно-оценочных материалов, проверку корректности формулировок тестовых заданий и интерпретацию накопленных результатов. При этом учебный материал, тестирование и аналитика, как правило, разнесены по разнородным инструментам – текстовому редактору, Google Forms или модулям LMS, – вследствие чего утрачивается единый предметный контекст сопровождения дисциплины [1].

Появление больших языковых моделей (LLM) открывает возможность автоматической генерации тестовых вопросов по тексту лекции, однако прямое использование результатов генерации без последующей нормализации и содержательной валидации приводит к появлению дублирующих, шаблонных или семантически некорректных формулировок [2]. Это определяет необходимость разработки системы, в которой AI-модуль выступает помощником преподавателя, а не подменяет его в принятии методических решений.

Целью настоящей работы является разработка специализированного веб-сервиса «КампусПлюс», формирующего единый прикладной маршрут преподавателя: от загрузки лекционного материала до интерпретации результатов тестирования по конкретной учебной группе. В состав задач исследования входят анализ предметной области и существующих решений, проектирование функциональной и информационно-логической моделей системы, разработка прототипа с интеграцией AI-модуля генерации тестов и его апробация на реальных данных.

Исследование предметной области

Контроль знаний обучающихся в высшем учебном заведении представляет собой систематический процесс проверки степени усвоения учебного материала, включающий формирование контрольно-оценочных заданий, проведение тестирования, обработку результатов и их интерпретацию. В практике кафедр преподаватель параллельно ведет несколько дисциплин, закрепленных за различными учебными группами, и регулярно повторяет полный цикл операций – подготовку лекционного материала, составление тестовых вопросов, публикацию доступа и формирование сводных ведомостей. Совокупная трудоемкость этих операций растет линейно с числом дисциплин и групп, тогда как методически значимым этапом, не поддающимся полной автоматизации, остается содержательная интерпретация результатов.

Современные информационные системы поддержки образовательного процесса делятся на три класса. Универсальные платформы управления обучением (LMS), такие как Moodle, предоставляют широкий набор инструментов – курсы, тесты, форумы, журналы оценок, – однако требуют значительной предварительной настройки и не содержат встроенного модуля AI-генерации тестовых заданий по тексту лекции. Упрощенные инструменты создания онлайн-форм (Google Forms, Яндекс.Формы) отличаются простотой интерфейса, но не поддерживают связь тестов с учебным материалом и не предоставляют дисциплинарной

аналитики. Специализированные образовательные платформы (Stepik и аналоги) ориентированы на формат массовых открытых курсов и не учитывают специфику текущего контроля знаний на занятии, в том числе оперативный QR-доступ к тесту [3].

Сравнительный анализ показал, что ни одно из рассмотренных решений не обеспечивает сквозного прикладного маршрута преподавателя, объединяющего подготовку контрольно-оценочных материалов, AI-генерацию вопросов, публикацию доступа на занятии и предметную аналитику в едином цифровом контуре. Данное обстоятельство обосновывает целесообразность разработки специализированного веб-сервиса, ориентированного на типовой сценарий работы преподавателя кафедры прикладной информатики и информационных систем.

Методы и методики

Был проведен сравнительный анализ существующих образовательных платформ контроля знаний – Moodle, Google Forms и Stepik. Moodle представляет собой универсальную систему управления обучением (LMS) с многоуровневой аналитикой, однако ее внедрение требует значительной предварительной настройки и отсутствует встроенный механизм AI-генерации вопросов по тексту лекции. Google Forms ограничена форматом онлайн-формы и не обеспечивает ни связи тестов с лекционным материалом, ни дисциплинарной аналитики. Stepik ориентирован на формат массовых открытых курсов и не учитывает специфику текущего контроля знаний на занятии [3].

Анализ показал, что ни одно из рассмотренных решений не обеспечивает сквозного прикладного маршрута преподавателя, объединяющего подготовку контрольно-оценочных материалов, AI-генерацию вопросов, публикацию доступа к тесту и предметную аналитику в едином цифровом контуре. Данное обстоятельство обосновывает целесообразность разработки специализированного веб-сервиса, ориентированного именно на типовой сценарий работы преподавателя кафедры.

В качестве языка реализации выбран Python. Выбор обоснован прямой интеграцией с библиотеками обработки естественного языка и публичными API больших языковых моделей. Среди веб-фреймворков рассматривались Django, Express, ASP.NET Core и FastAPI; по совокупности критериев – скорость старта, нативная асинхронная обработка запросов, простота интеграции с AI – выбран FastAPI в связке с шаблонизатором Jinja2 [4]. Для хранения данных применяются SQLite на этапе разработки и PostgreSQL в продуктивной среде, что обеспечивает единый прикладной код для обеих СУБД. AI-модуль реализован через OpenAI API с моделью GPT-4.1 [5].

Архитектура решения включает четыре уровня: клиентский уровень (адаптивные HTML-страницы), серверный уровень (маршруты FastAPI, авторизация, бизнес-логика), AI-модуль (генерация тестовых вопросов через OpenAI API) и слой данных (модуль db.py с поддержкой SQLite/PostgreSQL). Для обеспечения отказоустойчивости функции генерации предусмотрено автоматическое переключение на резервный AI-провайдер при недоступности основного, а при

превышении лимитов запросов система предлагает преподавателю воспользоваться ручным конструктором тестов.

Результаты

В рамках проектирования веб-сервиса разработан комплекс UML-моделей, описывающих функциональные, поведенческие и информационно-логические аспекты системы. Диаграмма прецедентов выделяет четырех акторов: преподавателя, студента, администратора и AI-модуль; преподаватель является центральным актором, поскольку именно его прикладной маршрут определяет основную ценность сервиса. Диаграмма классов отражает учебные сущности и связи между ними, в основе модели лежит цепочка «дисциплина → лекция → тест → попытка → аналитика». Диаграмма последовательностей описывает жизненный цикл теста – от создания лекции и AI-генерации вопросов до получения аналитики после прохождения студентом, а диаграмма активности моделирует процесс работы преподавателя с развилкой выбора между автоматической AI-генерацией и ручным конструктором тестов.

Физическая модель базы данных построена по фактической структуре проекта и включает таблицы `users`, `groups`, `disciplines`, `lectures`, `tests`, `questions`, `attempts` и `answers`, а также служебные таблицы `group_teachers`, `teacher_disciplines`, `teaching_assignments` и `teaching_assignment_blocks`. Ключевая таблица `teaching_assignments` реализует тройки «преподаватель + дисциплина + группа» и определяет, к каким тестам имеют доступ обучающиеся и какие результаты попадают в преподавательскую аналитику [6, 7]. Доступ преподавателя к студентам реализован не напрямую, а через указанную тройку, что точнее отражает реальный учебный процесс.

Прототип системы реализован как полнофункциональное веб-приложение с серверным рендерингом на базе FastAPI и Jinja2. Интерфейс охватывает три ролевых контура – преподавателя, студента и администратора – и адаптирован для мобильных устройств, что позволяет студентам проходить тестирование непосредственно на занятии через QR-код. Стартовый экран веб-сервиса (рис. 1) обеспечивает быстрый вход в систему и обзор ключевых разделов платформы: личного кабинета, результатов тестирования и точек роста обучающегося.

Особое внимание уделено контролю качества AI-генерации тестовых вопросов. Серверный модуль нормализации выполняет дедупликацию вопросов по порогу текстового сходства ($\geq 82\%$), фильтрацию шаблонных и обрезанных формулировок, проверку соотношения правильного и неправильных ответов (при сходстве $\geq 90\%$ вопрос отбрасывается), а также рандомизацию порядка вариантов ответа. При этом преподаватель остается в контуре принятия решения – он видит все вопросы до публикации и может удалить, отредактировать или дополнить набор [8].

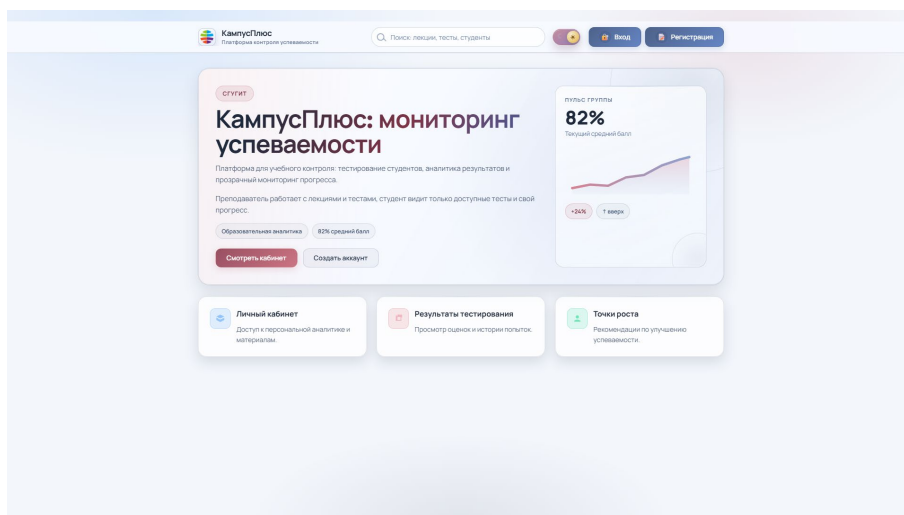


Рис. 1. Стартовый экран веб-сервиса «КампусПлюс»

Тестирование системы выполнено в трех форматах: ручное сценарное тестирование (20 тест-кейсов по пяти направлениям), автоматизированное модульное и интеграционное тестирование на базе pytest (свыше 140 тестов, покрывающих авторизацию, разграничение ролей, бизнес-логику, SQL-хелперы и smoke-проверки), а также нагрузочное тестирование. Нагрузочное тестирование подтвердило соответствие сервиса заявленным нефункциональным требованиям: среднее время отклика составило 48 мс, пропускная способность – 142 запроса в секунду, доля успешных запросов – 99,8 % при P95-latency 174 мс.

Заключение

В результате выполнения работы разработан и апробирован веб-сервис «КампусПлюс» для мониторинга успеваемости с применением технологий искусственного интеллекта. Сервис формирует единый прикладной маршрут преподавателя «дисциплина – лекция – тест – попытка – аналитика», обеспечивает автоматизированную AI-генерацию тестовых вопросов с последующей нормализацией и валидацией преподавателем, а также предметную аналитику по группам и отдельным студентам. Прототип развернут на облачной платформе Render и доступен по адресу campusplus.onrender.com. Результаты работы могут быть использованы для сопровождения текущего контроля знаний обучающихся на кафедре прикладной информатики и информационных систем СГУГиТ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Басаргин, А. А. Методы и средства проектирования информационных систем / А. А. Басаргин. – Новосибирск: СГУГиТ, 2015. – 236 с.
2. Brown, T. Language Models are Few-Shot Learners / T. Brown [et al.] // Advances in Neural Information Processing Systems. – 2020. – Vol. 33. – P. 1877–1901.
3. Coates, H. Student engagement in campus-based and online education / H. Coates. – London: Routledge, 2006. – 200 p.
4. Tiangolo, S. FastAPI: modern, fast web framework for building APIs with Python. – URL: <https://fastapi.tiangolo.com> (дата обращения: 10.03.2026).

5. OpenAI. Models – OpenAI API documentation. – URL: <https://platform.openai.com/docs/models> (дата обращения: 15.03.2026).
6. Буч, Г. UML. 2-е издание / Г. Буч, А. Якобсон, Д. Рамбо. – Санкт-Петербург: Питер, 2005. – 736 с.
7. Райордан, Р. Основы реляционных баз данных / Р. Райордан. – Москва: Русская Редакция, 2010. – 384 с.
8. Anil, R. Gemini: A Family of Highly Capable Multimodal Models / R. Anil [et al.] // arXiv preprint arXiv:2312.11805. – 2023.

© *Р. В. Енчу, А. А. Басаргин, 2026*

Т. О. Ермакова¹✉, Н. С. Головачев¹

Проектирование и программная реализация интерфейсов для системы взаимодействия пользователей с проектом «BlushBound»

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: ermakova06tanya@mail.ru

Аннотация. В статье представлены результаты проектирования и программной реализации адаптивной подсистемы пользовательских интерфейсов для игрового проекта «BlushBound». Выполнен анализ проблемы аппаратной фрагментации дисплеев и необходимости интеграции систем динамической локализации в инди-проектах. Для создания клиентского модуля использованы среда разработки Unity 6 и язык C#. Разработана трехуровневая архитектура, базирующаяся на паттерне Singleton и событийно-ориентированной модели взаимодействия компонентов. Реализованы алгоритмы динамического перестроения макетов под сверхширокие экраны, расчет безопасных зон и подсистема инклюзивных настроек с динамическим масштабированием шрифтов. Программное решение позволяет бесшовно интегрировать игровые механики с экранами сюжетной сцены, обеспечивая высокий уровень пользовательского опыта.

Ключевые слова: пользовательский интерфейс, адаптивная верстка, локализация, Unity, паттерны проектирования

T. O. Ermakova¹✉, N. S. Golovachev¹

Design and software implementation of user interfaces for the «BlushBound» project

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: ermakova06tanya@mail.ru

Abstract. The article presents the results of the design and software implementation of an adaptive user interface subsystem for the «BlushBound» game project. An analysis of the hardware displays fragmentation problem and the need for dynamic localization systems in indie projects is performed. The Unity 6 development environment and the C# language were used to create the client module. A three-tier architecture based on the Singleton pattern and an event-driven component interaction model was developed. Algorithms for dynamic layout rebuilding for ultrawide screens, safe area calculation, and an inclusive settings subsystem with dynamic font scaling were implemented. The software solution seamlessly integrates game mechanics with story scene screens providing a high level of user experience.

Keywords: user interface, adaptive layout, localization, Unity, design patterns

Введение

Развитие индустрии интерактивных развлечений диктует новые стандарты качества для разрабатываемых программных продуктов. Графический пользовательский интерфейс и пользовательский опыт перешли в разряд ключевых факторов, напрямую определяющих коммерческую успешность приложения. Современный

графический интерфейс выполняет сложную комплексную задачу: он не только предоставляет интуитивный доступ к базовому функционалу программы, но и обеспечивает глубокое погружение пользователя в виртуальную среду, снижая когнитивную нагрузку при взаимодействии с контентом. Особенностью проектирования интерфейсов для гибридных жанров, к которым относится проект «BlushBound», является необходимость органичного совмещения различных визуальных пространств без перегрузки экрана [1, 2].

Главной трудностью при конструировании визуальных элементов для современного программного обеспечения выступает непрерывная аппаратная фрагментация и физическая разнородность экранов. Матрицы конечных устройств варьируются от традиционных пропорций шестнадцать к девяти до панорамных форматов сверхшироких мониторов. Кроме того, мобильный сегмент отягощен наличием аппаратных вырезов под оптические сенсоры и системные панели. Применение статической верстки и жестко заданных координат в подобной среде неизбежно влечет за собой искажение графики, обрезку текста или полное перекрытие интерактивных элементов. Дополнительным вызовом является обязательная потребность в интернационализации продукта. Проекту требуется централизованный механизм языковой адаптации, позволяющий транслировать тексты без модификации исходного кода. Целью данного исследования является проектирование и программная реализация отказоустойчивой, эластичной и масштабируемой системы пользовательских интерфейсов клиентского модуля для игрового проекта «BlushBound».

Методы и материалы

В качестве основного технологического фундамента для создания визуальной логики интерфейсов применялась программная платформа шестого поколения Unity, а кодовая база и математические алгоритмы формировались посредством объектно-ориентированного языка программирования C# [3]. Использование среды Unity продиктовано широкими встроенными возможностями подсистемы визуализации Canvas, а также наличием библиотеки TextMeshPro. Данная библиотека осуществляет векторное построение текстовых символов на основе вычисления полей расстояний. Внедрение данной технологии имеет критическое значение для интерфейса, так как оно полностью исключает размытие пикселей и потерю качества графики при масштабировании текстов на дисплеях сверхвысокого разрешения [4, 5].

Особое внимание при проектировании клиентского модуля было уделено разработке структуры самих пользовательских интерфейсов на основе принципов гештальт-психологии и модульных сеток. Макет главного меню спроектирован по модульному принципу: графическое пространство логически разделено на функциональные навигационные блоки. Центральную и фоновую часть занимает художественная иллюстрация, создающая эмоциональную привязку. Левая панель интерфейса инкапсулирует клавиши доступа к коллекциям героев и системным настройкам, тогда как правая панель выделена под акцентные элементы запуска бесконечного или сюжетного игровых режимов. Подобное зонирование

снижает время визуального поиска нужного раздела пользователем и формирует четкие паттерны взаимодействия.

Для обеспечения интернационализации графических оболочек была спроектирована подсистема централизованной локализации. Базы данных словарей и переводов опираются на легковесный табличный формат CSV. Структура файла локализации предусматривает наличие заголовочной строки с языковыми кодами, где каждой последующей строке строго сопоставлен уникальный графический ключ-идентификатор и соответствующие текстовые переводы, разделенные символом точки с запятой.

Для обеспечения стабильной работы интерфейсов, алгоритмов адаптивности и подсистемы локализации общая архитектура приложения была разграничена на изолированные слои. Глобальные управляющие классы, такие как контроллер навигации `SceneController`, менеджер параметров `SettingsManager` и менеджер локализации `LocalizationManager`, выстроены с применением порождающего паттерна `Singleton`. Данный управляющий узел инициализируется единожды внутри отдельной стартовой сцены `BootScene` и функционирует полностью автономно на протяжении всей пользовательской сессии. Корректность отображения на экранах различных форматов дополнительно обеспечивалась разработкой вспомогательных скриптов `AspectRatioLayout` и `SafeAreaFitter`. Указанные алгоритмы вычисляют доступные физические габариты монитора и программно сжимают графический интерфейс в пределы гарантированной видимости, защищая интерактивные элементы от перекрытия аппаратными вырезами мобильных устройств.

Результаты

Главным практическим результатом проведенных исследовательских и проектных работ стала комплексная визуальная и программная реализация экранов взаимодействия клиентской среды. Итоговый программный продукт органично объединяет масштабируемое стартовое меню, инклюзивные панели пользовательских конфигураций и производительную систему локализации.

Визуальное представление главного меню и системы навигации демонстрирует интуитивно понятное расположение элементов управления. Для обеспечения высокого уровня доступности интерфейса дополнительно реализован механизм динамического масштабирования типографики. Специализированный модуль `UIFontScaler`, перехватывая системный сигнал об изменении конфигурации из меню настроек, математически умножает исходный размер базового шрифта на заданный пользователем числовой коэффициент. Кроме того, для безопасного завершения работы приложения было реализовано модальное окно `ExitConfirmDialog`. Данный графический узел перекрывает основное меню затемненным фоном, полностью блокируя случайные нажатия на элементы заднего плана, и ожидает явного подтверждения действия от пользователя (рис.1).

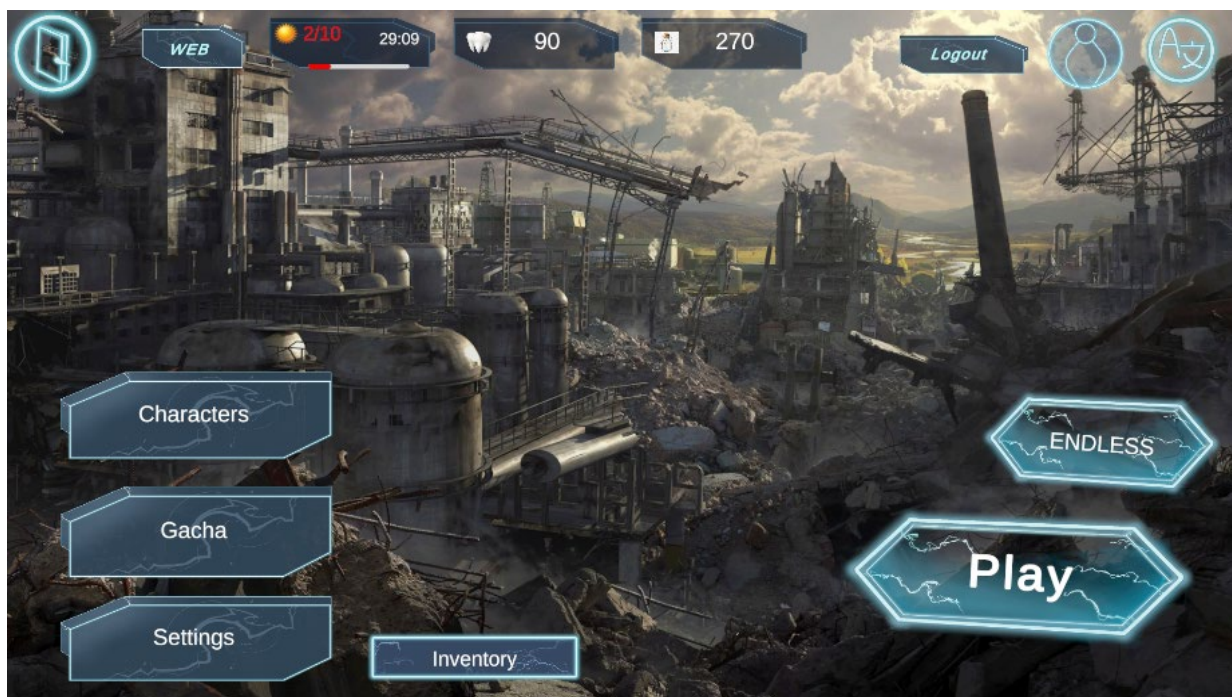


Рис. 1. Визуальная реализация адаптивного пользовательского интерфейса клиентского модуля

Программный механизм локализации графических экранов опирается на ресурсосберегающую событийно-ориентированную архитектуру. Логика работает по следующему алгоритму: элемент-инициатор передает команду в управляющий класс локализации. Класс обрабатывает запрос, сохраняет выбранный язык в локальную энергонезависимую память устройства через интерфейс `PlayerPrefs` и генерирует глобальное системное событие `LanguageChanged`.

Все активные текстовые узлы интерфейса, содержащие скрипт `LocalizedTextTMP`, выступая в роли подписчиков, перехватывают данный сигнал. Получив уведомление, они автономно извлекают корректный перевод из загруженного в оперативную память CSV-массива и мгновенно обновляют свое визуальное содержимое. Указанная программная методология полностью устраняет потребность в ресурсоемком циклическом опросе или программном поиске графических компонентов на сцене (рис. 2).

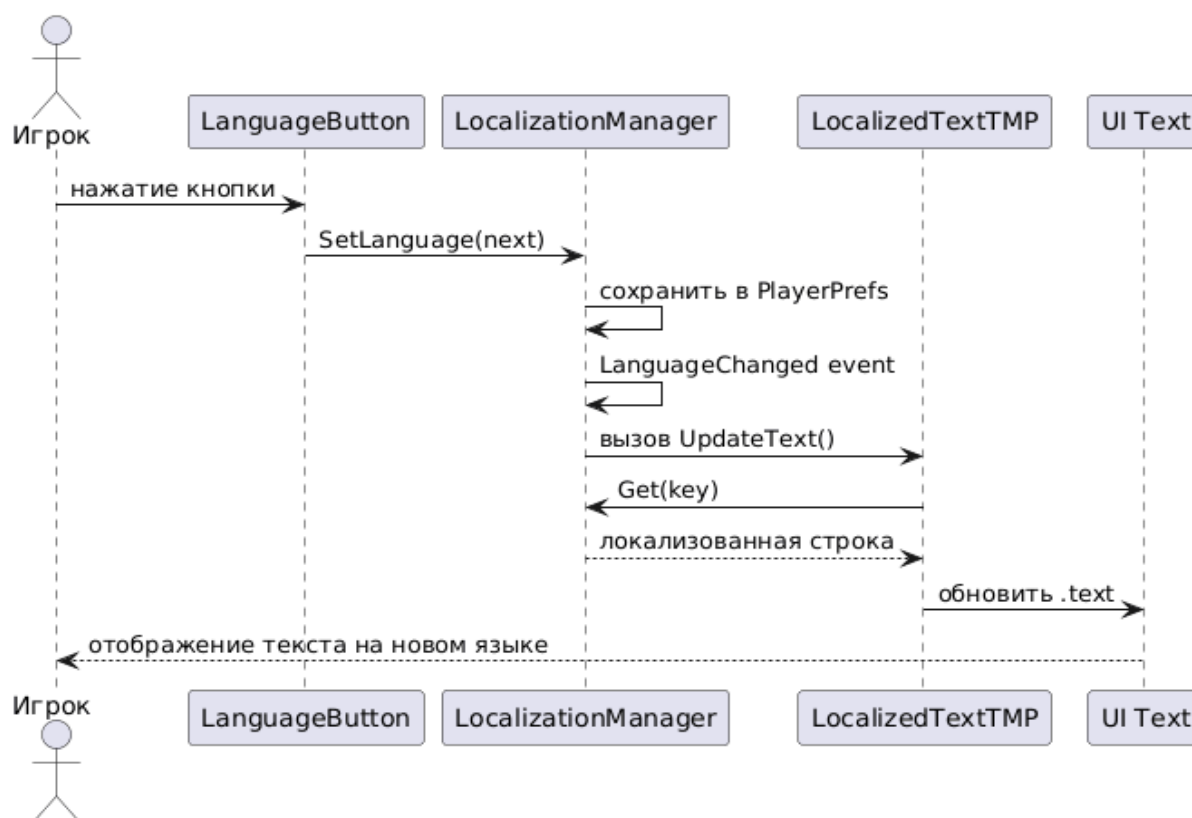


Рис. 2. Алгоритм событийного обновления локализованного текста в интерфейсе

Дополнительно в рамках работы сюжетный нарративный кластер был спроектирован как изолированная интерактивная среда «StoryScene_FirstMeeting». В текстовый интерфейс визуальной новеллы внедрен алгоритм графической фокусировки для управления вниманием игрока: художественный портрет активного спикера отображается на экране с максимальной непрозрачностью, тогда как графическая модель слушающего персонажа программно затемняется путем снижения альфа-канала до значения ноль целых пять десятых.

Заключение

В результате проведенного исследования и программной разработки была создана отказоустойчивая подсистема пользовательских интерфейсов для проекта «BlushBound». Использование алгоритмов расчета безопасных зон и скриптов динамического переключения макетов позволило обеспечить полную кроссплатформенную адаптивность интерфейса. Разработанная архитектура на базе паттерна Singleton и системы событий успешно решает задачи централизованного управления настройками и мгновенной локализации. Бесшовная интеграция модуля визуальной новеллы подтверждает надежность спроектированной навигационной системы. Предложенные архитектурные решения могут быть использованы при разработке масштабируемых интерфейсов для коммерческих кроссплатформенных приложений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сондерс, К. Проектирование игровых интерфейсов / К. Сондерс, П. Новак. – Москва : ДМК Пресс, 2020. – 312 с.
2. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – Санкт-Петербург : Питер, 2021. – 368 с.
3. Троелсен, Э. Язык программирования C# 9 и платформа .NET 5: основные принципы и практики программирования / Э. Троелсен, Ф. Джепикс. – Москва : Вильямс, 2022. – 1328 с.
4. Хокинг, Д. Unity в действии. Мультиплатформенная разработка на C# / Д. Хокинг. – Санкт-Петербург : Питер, 2023. – 400 с.
5. Unity User Manual 2023. User Interfaces (UI) [Электронный ресурс] // Unity Technologies. – URL: <https://docs.unity3d.com/Manual/UIToolkits.html> (дата обращения: 12.05.2026).

© Т. О. Ермакова, Н. С. Головачев, 2026

Н. А. Ермилов^{1✉}, Е. Ю. Воронкин¹

Разработка серверной части SaaS-платформы по управлению e-commerce бизнесом

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: nikitaermilovvv@gmail.com

Аннотация. В статье рассматривается процесс разработки серверной части SaaS-платформы для управления e-commerce бизнесом и автоматизации взаимодействия между самозанятыми продавцами и партнерскими магазинами, размещающими товары на маркетплейсах. Актуальность работы обусловлена ростом числа независимых продавцов, реализующих товары через социальные сети и интернет-магазины, но не имеющих собственной инфраструктуры для работы с маркетплейсами. Разработанная система предоставляет возможность публикации товаров, управления заказами, формирования документов для отгрузки, а также интеграции с внешними сервисами и сайтами. Серверная часть реализована с использованием современных веб-технологий и ориентирована на масштабируемость, безопасность и централизованное управление бизнес-процессами. В результате создана программная платформа, обеспечивающая автоматизацию логистики, обработки заказов и администрирования e-commerce операций.

Ключевые слова: SAAS, e-commerce, backend, API, автоматизация бизнеса, маркетплейсы, управление заказами, интеграция сервисов, логистика, веб-разработка

N. A. Ermilov^{1✉}, E. Yu. Voronkin¹

Development of the server part of a SaaS-platform for e-commerce business management

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: nikitaermilovvv@gmail.com

Abstract. The article discusses the development process of the backend part of a SaaS platform for e-commerce business management and automation of interactions between self-employed sellers and partner stores operating on marketplaces. The relevance of the work is conditioned by the growing number of independent sellers who sell products through social networks and websites but do not have direct access to marketplace infrastructure. The developed system provides product management, order processing, shipment documentation generation, and integration with external services and websites. The backend architecture is designed with a focus on scalability, security, and centralized business process management. As a result, a software platform was created to automate logistics, order handling, and administrative e-commerce operations.

Keywords: SAAS, e-commerce, backend, API, business automation, marketplaces, order management, logistics, web development

Введение

Современный рынок электронной коммерции активно развивается за счет малого бизнеса, самозанятых и независимых производителей. Значительная

часть продавцов реализует товары через социальные сети, мессенджеры и собственные интернет-магазины, однако сталкивается с ограничениями при выходе на крупные маркетплейсы.

Основными проблемами являются сложность интеграции с инфраструктурой маркетплейсов, высокая стоимость логистики при самостоятельной отправке заказов и отсутствие централизованного инструмента управления продажами.

Для решения данной задачи была разработана SaaS-платформа, позволяющая самозанятым размещать товары через партнерские магазины, работающие на маркетплейсах. Платформа обеспечивает управление каталогом товаров, обработку заказов, формирование документов для отгрузки и контроль логистических процессов.

Целью работы является разработка серверной части SaaS-платформы для автоматизации e-commerce бизнес-процессов. Для достижения цели были выполнены следующие задачи:

- анализ требований к платформе;
- проектирование серверной архитектуры;
- разработка API для взаимодействия клиентской части и внешних сервисов;
- реализация системы управления товарами, заказами и пользователями;
- интеграция модулей логистики и документооборота;
- тестирование и развертывание серверной части.

Методы и материалы

Перед началом разработки серверной части SaaS-платформы был проведен анализ предметной области и сформулированы основные требования к системе. Разрабатываемая платформа ориентирована на автоматизацию процессов управления e-commerce бизнесом и взаимодействия между самозанятыми продавцами, партнерскими магазинами и администрацией сервиса. На этапе проектирования были определены ключевые бизнес-процессы платформы, включающие управление товарами, обработку заказов, сопровождение отправок, работу с документацией и интеграцию с внешними сервисами. Также была спроектирована логика разграничения ролей пользователей, предусматривающая различные уровни доступа для продавцов, администраторов и партнерских аккаунтов.

Для реализации серверной части использовался язык программирования Python 3.11 и веб-фреймворк Flask 3, обеспечивающий обработку HTTP-запросов и построение серверного приложения. Архитектура приложения построена по модульному принципу с разделением конфигурации, маршрутов и бизнес-логики, что упрощает поддержку и развитие проекта. Взаимодействие между клиентской и серверной частями реализовано посредством REST API, позволяющего передавать данные между личным кабинетом пользователя, административной панелью и серверной инфраструктурой. Для обеспечения работы приложения в production-среде используется WSGI-сервер Gunicorn, а корректная

обработка запросов при работе за reverse проху реализована с использованием ProxyFix из библиотеки Werkzeug.

Для хранения данных используется реляционная модель базы данных. В процессе локальной разработки и тестирования применялась СУБД SQLite, обеспечивающая простоту настройки и автономность окружения. Для production-развертывания используется PostgreSQL, подключаемая через переменные окружения, что позволяет обеспечить масштабируемость и более надежную работу с данными. Взаимодействие с базой данных реализовано с использованием ORM SQLAlchemy 2, позволяющей работать с сущностями приложения на уровне объектной модели. Для управления версиями структуры базы данных и безопасного внесения изменений применяется система миграций Alembic.

Серверная часть платформы интегрирована с рядом внешних сервисов, необходимых для автоматизации бизнес-процессов. Для выполнения HTTP-запросов к внешним API используется библиотека requests. Реализована интеграция с API маркетплейса Wildberries, позволяющая взаимодействовать с сервисами управления товарами и контентом. Дополнительно реализована интеграция с банковской системой Точка, включающая OAuth-авторизацию и обработку входящих вебхуков. Для проверки JWT-подписей используются библиотеки PyJWT и cryptography. Генерация вспомогательных документов, PDF-файлов и обработка изображений выполняется с использованием библиотек ReportLab и Pillow. Для обеспечения воспроизводимости окружения и упрощения развертывания серверной части используется контейнеризация на основе Docker.

Результаты

В результате проведенного анализа и проектирования была реализована серверная часть SaaS-платформы для управления e-commerce бизнесом. На основе спроектированной архитектуры были реализованы основные серверные модули системы, отвечающие за управление пользователями, товарами, заказами и бизнес-процессами платформы. В серверной части реализован модуль управления товарами, позволяющий сохранять, обновлять и обрабатывать данные о продукции пользователей. Также реализована система обработки заказов, включающая получение информации о заказах, обновление их статусов и сопровождение логистических операций.

Для хранения данных были реализованы модели сущностей и связи между ними с использованием SQLAlchemy. Структура базы данных включает сущности пользователей, товаров, заказов, магазинов, отправок и интеграционных данных. При этом с помощью Alembic реализован механизм миграций, позволяющий управлять изменениями структуры базы данных и поддерживать целостность данных при развитии платформы.

Одним из ключевых результатов разработки стала реализация интеграции с API Wildberries и банковской системой Точка. Модуль Wildberries обеспечивает обмен данными с маркетплейсом и автоматизацию операций, связанных с товарами и заказами, в то время как интеграция с банком позволяет обрабатывать финансовые события, OAuth-авторизацию и входящие вебхуки. Для обеспечения

безопасности реализована система аутентификации и разграничения прав доступа пользователей. Серверная часть поддерживает контроль ролей и ограничение доступа к административным функциям платформы.

Дополнительно реализован функционал генерации документов и вспомогательных материалов для логистики с использованием ReportLab и Pillow. В целом разработанная серверная архитектура обладает модульной структурой, что обеспечивает возможность дальнейшего масштабирования и расширения функциональности платформы.

Заключение

В статье рассмотрен процесс разработки серверной части SaaS-платформы для управления e-commerce бизнесом, ориентированной на автоматизацию взаимодействия между самозанятыми продавцами, партнерскими магазинами и административной системой платформы. Основной целью проекта являлось создание масштабируемой и безопасной серверной инфраструктуры, обеспечивающей управление товарами, обработку заказов, сопровождение логистических процессов и централизованное администрирование основных бизнес-операций. В ходе работы были проанализированы требования к системе, определены ключевые бизнес-процессы и спроектирована архитектура серверного приложения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Docker Inc. Docker Documentation. – Режим доступа: <https://docs.docker.com/> (дата обращения: 27.03.2026). – (Контейнеризация и оркестрация серверного приложения.)
2. PostgreSQL Global Development Group. PostgreSQL 16 Documentation. – Режим доступа: <https://www.postgresql.org/docs/16/> (дата обращения: 25.03.2026). – (Реляционная СУБД для production-среды.)
3. SQLAlchemy Authors. SQLAlchemy 2.0 Documentation. – Режим доступа: <https://docs.sqlalchemy.org/> (дата обращения: 26.03.2026). – (ORM и ядро для работы с базой данных.)
4. Alembic Project. Alembic Migration Documentation. – Режим доступа: <https://alembic.sqlalchemy.org/> (дата обращения: 26.03.2026). – (Управление версиями схемы БД.)
5. Gunicorn Project. Gunicorn WSGI Server Documentation. – Режим доступа: <https://docs.gunicorn.org/> (дата обращения: 24.03.2026). – (Production-сервер для WSGI-приложений.)
6. Wildberries. API Marketplace (v2). – Режим доступа: <https://openapi.wildberries.ru/> (дата обращения: 20.03.2026). – (Интеграция с маркетплейсом: товары, заказы, логистика.)
7. Банк «Точка». API для партнеров. – Режим доступа: <https://enter.tochka.com/api> (дата обращения: 22.03.2026). – (Официальная документация на русском языке, OAuth 2.0 и вебхуки.)
8. Python Software Foundation. The Python Standard Library: requests, PyJWT, cryptography, hashlib. – Режим доступа: <https://docs.python.org/3/library/> (дата обращения: 27.03.2026). – (Стандартные и дополнительные библиотеки для HTTP, JWT, шифрования.)

© Н. А. Ермилов, Е. Ю. Воронкин, 2026

И. И. Захаренко^{1✉}, Д. С. Худяков¹

Интеллектуальная платформа разработки инженерных конструкций

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: ilyazakharenko1994@mail.ru

Аннотация. Исследование посвящено разработке интеллектуальной платформы для генеративного проектирования и оптимизации мультифизических инженерных систем. Особое внимание уделяется применению платформы для оптимизации систем охлаждения литий-ионных аккумуляторных батарей и электролизёров зелёного водорода с целью повышения их энергоэффективности, безопасности и ресурса.

В результате анализа были выявлены фундаментальные проблемы существующих подходов, среди которых – отсутствие единого замкнутого цикла от текстового описания требований до физически корректной оптимизированной конструкции, слабая интеграция строгих физических моделей с адаптивными алгоритмами оптимизации, а также недостаточная ориентированность инструментов на задачи зелёных технологий. Предложена архитектура интеллектуальной платформы, объединяющая языковой интерфейс, параметрическую генерацию геометрии, физически-информированные нейронные сети и гибридные методы оптимизации. Предварительные вычислительные эксперименты демонстрируют потенциал снижения вычислительных затрат на этапе раннего проектирования на 30–40 % и улучшения целевых характеристик конструкции (масса, тепловые потери, эффективность) на 10–15 % по сравнению с традиционными методами.

Ключевые слова: генеративное проектирование, физически-информированные нейронные сети, обучение с подкреплением, мультифизическое моделирование, интеллектуальная платформа

I. I. Zakharenko^{1✉}, D. S. Khudyakov¹

Intelligent system for engineering design development

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: ilyazakharenko1994@mail.ru

Abstract. The study is devoted to the development of an intelligent platform for generative design and optimization of multiphysics engineering systems. Particular attention is paid to the application of the platform for optimizing cooling systems of lithium-ion battery packs and green hydrogen electrolyzers in order to improve their energy efficiency, safety, and operational lifetime.

The analysis revealed several fundamental limitations of existing approaches, including the absence of an integrated closed-loop workflow from textual design requirements to a physically consistent optimized construction, weak integration of rigorous physical models with adaptive optimization algorithms, and insufficient focus on green technology applications.

An architecture of an intelligent platform combining a natural language interface, parametric geometry generation, physics-informed neural networks, and hybrid optimization methods is proposed. Preliminary computational experiments demonstrate the potential to reduce computational costs at the early design stage by 30–40 % and improve key design characteristics, including mass, thermal losses, and overall efficiency, by 10–15 % compared to traditional approaches.

Keywords: generative design, physics-informed neural networks, reinforcement learning, multiphysics modeling, intelligent design platform

Введение

Разработка современных мультифизических инженерных систем требует значительных вычислительных затрат, особенно при использовании методов полного численного моделирования (CFD и FEM) [1, 2]. Это ограничивает количество рассматриваемых проектных решений и замедляет ранние этапы проектирования, особенно в задачах зелёных технологий, где критически важны энергоэффективность, безопасность и надёжность [3].

Особую актуальность данная проблема приобретает при проектировании систем охлаждения литий-ионных аккумуляторных батарей и теплового управления электролизёров зелёного водорода [3, 4]. Эффективность таких систем во многом определяется качеством инженерных решений, принимаемых на ранних стадиях разработки.

Существующие подходы в области генеративного проектирования и интеллектуальной оптимизации имеют ряд ограничений: отсутствие единого замкнутого цикла проектирования, слабая интеграция строгих физических моделей с адаптивными алгоритмами оптимизации, высокая вычислительная стоимость и недостаточная ориентированность на задачи устойчивой энергетики [5–7].

Настоящая работа посвящена разработке интеллектуальной платформы генеративного проектирования мультифизических систем на основе физически-информированных нейронных сетей и обучения с подкреплением [8–11]. Предлагаемый подход ориентирован на автоматизацию ранних этапов проектирования и оптимизацию систем охлаждения литий-ионных батарей и электролизёров зелёного водорода при сохранении физической достоверности результатов.

Постановка задачи

Целью исследования является разработка и экспериментальная апробация интеллектуальной платформы для генеративного проектирования и оптимизации мультифизических инженерных систем на ранних этапах их разработки.

Для достижения поставленной цели была проведена декомпозиция на следующие задачи:

- 1) анализ современного состояния и ограничений существующих методов генеративного проектирования и оптимизации;
- 2) разработка архитектуры интеллектуальной платформы и математической постановки задачи;
- 3) реализация прототипа платформы и проведение вычислительных экспериментов на примере систем охлаждения литий-ионных аккумуляторных батарей и электролизёров зелёного водорода.

Теоретическая значимость исследования заключается в развитии подходов к интеграции физически-информированных нейронных сетей и методов обучения с подкреплением в едином контуре генеративного проектирования [8, 9]. Практическая значимость определяется возможностью применения разработанных методов и

прототипа платформы для ускорения создания высокоэффективных систем устойчивой энергетики [3, 4].

Формализация задачи интеллектуального генеративного проектирования инженерных систем

Задача интеллектуального генеративного проектирования мультифизических инженерных систем формулируется как многокритериальная задача условной оптимизации в многомерном пространстве проектных параметров.

Пусть $p \in R^n$ – вектор проектных параметров, включающий геометрические характеристики конструкции (внутренний радиус канала охлаждения r , толщину стенки δ , длину канала L и другие переменные).

Требуется найти оптимальный вектор параметров p^* , минимизирующий целевую функцию:

$$J(p) = w_1 \cdot \underline{T}(p) + w_2 \cdot \underline{P}(p) + w_3 \cdot \underline{\sigma}(p),$$

где $\underline{T}(p)$, $\underline{P}(p)$, $\underline{\sigma}(p)$ – нормированные значения максимальной температуры стенки, перепада давления и максимального эквивалентного напряжения соответственно, полученные из PINN-суррогатной модели; w_1, w_2, w_3 – весовые коэффициенты, отражающие относительную важность каждого критерия.

Ограничения задачи имеют вид:

$$T(p) \leq T_{max}, \quad \Delta P(p) \leq \Delta P_{max}, \quad \sigma(p) \leq \sigma_{доп}$$

где T_{max} , ΔP_{max} и $\sigma_{доп}$ – предельно допустимые значения, определяемые эксплуатационными требованиями и нормами безопасности.

Для решения данной задачи в работе разработан гибридный алгоритм оптимизации, сочетающий обучение с подкреплением (Proximal Policy Optimization) и эволюционные стратегии [12, 13]. Обучение с подкреплением обеспечивает адаптивное исследование пространства решений, а эволюционные методы повышают глобальную устойчивость поиска и снижают вероятность застревания в локальных оптимумах физически-информированных нейронных сетей. Такой подход позволяет учитывать сложные мультифизические взаимодействия (теплоперенос, гидродинамика, механика деформируемого твёрдого тела) без необходимости выполнения полного численного моделирования на каждой итерации оптимизационного цикла.

Исследование предметной области

Анализ современных решений позволяет выделить несколько основных направлений развития интеллектуальных систем проектирования.

Первое направление связано с развитием генеративного дизайна и топологической оптимизации в традиционных CAD/CAE-системах (Autodesk Fusion 360, Siemens NX, Altair Inspire) [5, 6]. Эти инструменты хорошо справляются с геометрической оптимизацией и позволяют учитывать некоторые производственные огра-

ничения, однако практически не способны работать со сложными мультифизическими процессами и не обеспечивают строгой физической корректности получаемых решений.

Альтернативное направление представлено работами по созданию суррогатных моделей и применению геометрического глубокого обучения (Neural Concept Shape, Altair PhysicsAI, Ansys Discovery) [7, 14]. Такие подходы обеспечивают значительное ускорение расчётов, но страдают от фундаментального недостатка – отсутствия гарантии соблюдения физических законов за пределами обучающей выборки. Это особенно критично для задач зелёных технологий, где ошибки в тепловом или электрохимическом моделировании могут привести к перегреву батарей или снижению эффективности электролизёров.

Так же существует ещё одно направление, которое связано с физически-информированными нейронными сетями (PINN). Работы [8–10] демонстрируют высокий потенциал PINN для решения сопряжённых задач теплопереноса, гидродинамики и механики. Тем не менее, большинство существующих решений остаются на уровне отдельных библиотек и научных экспериментов и практически не интегрированы в полный цикл генеративного проектирования – от текстового описания требования до готовой оптимизированной конструкции.

Отдельную группу составляют исследования по применению обучения с подкреплением (RL) в инженерной оптимизации [12, 13]. Несмотря на успехи в адаптивном поиске оптимальных решений, такие подходы редко сочетаются с жёсткими физическими моделями и почти не используются для задач зелёных технологий, где требуется одновременный учёт тепловых, электрохимических и механических процессов.

Анализ существующих подходов показывает наличие существенного пробела [1, 5, 15]. Отсутствует интегрированная интеллектуальная система, которая одновременно обеспечивала бы естественный языковой интерфейс для постановки задачи, автоматическую генерацию параметрической геометрии, физически корректную оценку на основе PINN и адаптивную многокритериальную оптимизацию с помощью обучения с подкреплением [1, 16].

Особенно остро этот пробел проявляется при проектировании систем для устойчивой энергетики – систем охлаждения литий-ионных аккумуляторных батарей и электролизёров зелёного водорода, где требуется быстрый и надёжный учёт сложных многофизических взаимодействий на ранних стадиях разработки [3, 4].

Моделирование предметной области

В настоящей работе предложена интеллектуальная платформа генеративного проектирования многофизических систем, ориентированная на задачи зелёных технологий. Платформа построена по модульному принципу и представляет собой замкнутый итеративный цикл с обратной связью, позволяющий автоматически переходить от текстового описания требований инженера к оптимизированной параметрической модели конструкции.

Архитектура платформы состоит из четырёх основных взаимосвязанных модулей:

- модуль языкового интерфейса и агентной оркестрации – преобразует естественный язык в структурированные требования и координирует работу всех модулей с помощью фреймворка LangGraph;

- модуль генерации параметрической геометрии – создаёт параметрические и перспективно неявные (Neural Fields) модели конструкции на основе библиотеки CadQuery;

- модуль физической оценки – выполняет быстрый и физически корректный расчёт многофизических процессов с использованием physics-informed neural networks (PINN) на базе NVIDIA PhysicsNeMo;

- модуль оптимизации – гибридный оптимизатор, сочетающий обучение с подкреплением (PPO) и эволюционные стратегии для эффективного поиска компромиссных решений.

Архитектура платформы представлена на рис. 1.

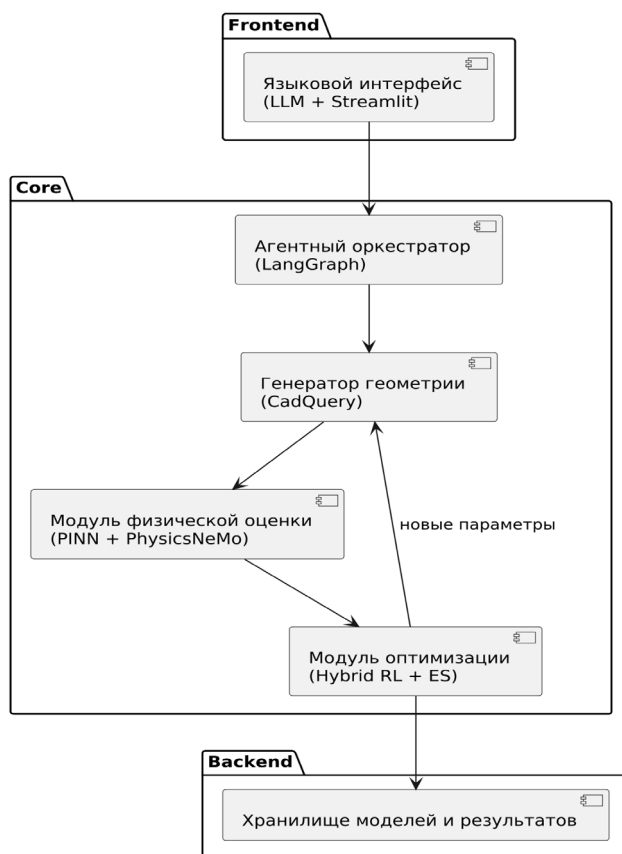


Рис. 1. Диаграмма компонентов интеллектуальной платформы

Взаимодействие модулей организовано следующим образом: модуль оркестрации принимает требования, передаёт их в генератор геометрии, получает параметры конструкции, направляет их в PINN-модуль для физической оценки, а затем в оптимизатор, который генерирует новые параметры. Цикл повторяется

до достижения критериев сходимости. Такой подход позволяет значительно сократить количество итераций, традиционно выполняемых инженером вручную.

Предлагаемая архитектура отличается от существующих решений интеграцией строгой физической корректности (PINN) с адаптивной оптимизацией (RL) и естественным языковым интерфейсом в едином контуре. Это позволяет применять платформу для сложных многофизических задач зелёных технологий, где традиционные методы сталкиваются с высокой вычислительной сложностью.

Аспекты информационной безопасности платформы

Разработка интеллектуальной платформы AetherForge осуществляется в рамках концепции DevSecOps, что подразумевает интеграцию требований информационной безопасности на всех этапах жизненного цикла системы – от проектирования архитектуры до эксплуатации.

В рамках реализации платформы обеспечивается:

- полная трассируемость всех проектных решений и действий агентов, что позволяет проводить аудит и верификацию корректности сгенерированных конструкций;
- защита от генерации небезопасных решений путём встраивания дополнительных ограничений безопасности в reward-функцию и систему правил оркестратора;
- защита интеллектуальной собственности и конфиденциальности проектных данных за счёт локального выполнения всех вычислений и возможности использования приватных моделей;
- устойчивость к атакам на ИИ-компоненты (инъекции промпта, адверсарные примеры) путём применения строгих промпт-инженерных практик и валидации входных данных;
- контроль целостности и воспроизводимости результатов за счёт фиксированных начальных состояний и логирования всех этапов оптимизационного цикла.

Такой подход позволяет не только решать инженерные задачи, но и гарантировать безопасность и надёжность самой платформы при её применении в критически важных отраслях, включая энергетику и аэрокосмическую промышленность.

Заключение

Предложенный в работе подход переводит задачу генеративного проектирования мультфизических систем из области традиционной параметрической оптимизации в постановку интеллектуального адаптивного моделирования. Научная новизна исследования заключается в интеграции следующего стека технологий: языкового интерфейса, параметрической генерации геометрии, физически информированных нейронных сетей (PINN) и гибридных алгоритмов оптимизации (RL + эволюционные стратегии) в едином замкнутом цикле [8–10, 12, 13].

В отличие от существующих решений, ориентированных либо на чисто геометрическую оптимизацию, либо на отдельное применение суррогатных моделей,

разрабатываемая платформа обеспечивает строгую физическую корректность решений на уровне уравнений в частных производных при сохранении высокой адаптивности поиска оптимальных конфигураций.

В дальнейшем разработанный метод и прототип платформы могут служить основой для полноценных экспериментальных исследований, масштабирования на более сложные объекты, а также интеграции в реальные инженерные процессы предприятий энергетического и транспортного секторов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Matusik W., Liu J., Zhang Y. Large Language Models for Computer-Aided Design: A Survey and Outlook // *Computer-Aided Design*. 2024. Vol. 178. 103456.
2. Sharma A., Gupta R., Singh P. Physics-Informed Neural Networks in Aerospace Engineering: Applications and Challenges // *Aerospace Science and Technology*. 2025. Vol. 156. 109812.
3. Karniadakis G.E., Kevrekidis I.G., Lu L. et al. Physics-informed machine learning // *Nature Reviews Physics*. 2021. Vol. 3. P. 422–440.
4. Cuomo S., Di Cola V.S., Giampaolo F. et al. Scientific machine learning through physics-informed neural networks: Where we are and what's next // *Journal of Scientific Computing*. 2022. Vol. 92. 88.
5. Raissi M., Perdikaris P., Karniadakis G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations // *Journal of Computational Physics*. 2019. Vol. 378. P. 686–707.
6. Brunton S.L., Noack B.R., Koumoutsakos P. Machine learning for fluid mechanics // *Annual Review of Fluid Mechanics*. 2020. Vol. 52. P. 477–508.
7. Liu J., Matusik W. Learning generative models for parametric CAD // *ACM Transactions on Graphics*. 2020. Vol. 39. No. 6. 185.
8. Banga S., Geers M., van der Meer F. 3D topology optimization using deep learning // *Nature Machine Intelligence*. 2022. Vol. 4. P. 112–121.
- Willard J., Jia X., Xu S. et al. Integrating physics-based modeling with machine learning: A survey // *Nature Reviews Physics*. 2022. Vol. 4. P. 157–175.
9. Goswami S., Kontolati K., Shields M.D. et al. Physics-informed neural networks for surrogate modeling // *Computer Methods in Applied Mechanics and Engineering*. 2022. Vol. 390. 114464.
10. Chen Z., Zhang H. RL-PINNs: Reinforcement Learning-Driven Adaptive Sampling for Physics-Informed Neural Networks // *Computer Methods in Applied Mechanics and Engineering*. 2024. Vol. 418. 116512.
11. Tang Y., Daoutidis P. Physics-informed reinforcement learning for process control // *AIChE Journal*. 2023. Vol. 69. No. 4. e17958.
12. Degraeve J., Felici F., Buchli J. et al. Magnetic control of tokamak plasmas through deep reinforcement learning // *Nature*. 2022. Vol. 602. P. 414–419.
13. Pfaff T., Fortunato M., Sanchez-Gonzalez A. et al. Learning mesh-based simulation with graph networks // *International Conference on Machine Learning (ICML)*. 2021. P. 8459–8468.
14. Rackauckas C., Ma Y., Martensen J. et al. Universal differential equations for scientific machine learning // *arXiv preprint arXiv:2001.04385*. 2020.
15. Kochkov D., Smith J.A., Alieva A. et al. Machine learning–accelerated computational fluid dynamics // *Proceedings of the National Academy of Sciences*. 2021. Vol. 118. No. 21. e2101784118.
16. Thuerey N., Weissenow K., Prantl L. et al. Deep learning methods for Reynolds-averaged Navier–Stokes simulations // *Journal of Fluid Mechanics*. 2020. Vol. 886. A17.

© И. И. Захаренко, Д. С. Худяков, 2026

В. А. Зинков^{1✉}, Е. Ю. Воронкин¹

Разработка информационной системы автоматизированного учета средств индивидуальной защиты сотрудников предприятия с применением методов искусственного интеллекта

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: vmtalya.zinkov@bk.ru

Аннотация. В статье представлены основные результаты разработки интеллектуальной информационной системы для автоматизации учета, выдачи и прогнозирования потребности в средствах индивидуальной защиты сотрудников предприятия. Рассмотрены нормативные цифровизации процессов обеспечения работников СИЗ, выделены проблемы разрозненного хранения данных, ручного формирования личных карточек и недостаточной точности планирования запасов. Предложена модульная архитектура системы, включающая загрузку данных, интеграцию с 1С, подготовку аналитических наборов, формирование документов, контроль сроков повторной выдачи, уведомления и прогнозирования потребности на основе исторических данных. В качестве технологической основы использованы Python, Streamlit, pandas и библиотека Prophet. Разработанное решение позволяет повысить прозрачность учета, снизить трудоемкость документооборота и обеспечить поддержку управленческих решений в области охраны труда.

Ключевые слова: средства индивидуальной защиты, учет СИЗ, охрана труда, прогнозирование потребности, машинное обучение, Prophet

V. A. Zinkov^{1✉}, E. Yu. Voronkin¹

Development of an information system for automated accounting of enterprise employees' personal protective equipment using artificial intelligence methods

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: vmtalya.zinkov@bk.ru

Abstract. The article presents the main results of the development of an intelligent information system for automating the accounting, issuing and forecasting of demand for personal protective equipment at an enterprise. The regulatory and organizational prerequisites for the digitalization of PPE supply processes are considered. The problems of fragmented data storage, manual preparation of personal issue cards and insufficient accuracy of inventory planning are identified. A modular system architecture is proposed, including data loading, integration with 1C, preparation of analytical datasets, document generation, control of reissue deadlines, notifications and demand forecasting based on historical data. Python, Streamlit, pandas and the Prophet library are used as the technological basis. The developed solution increases the transparency of accounting, reduces the labor intensity of document flow and supports management decisions in occupational safety.

Keywords: personal protective equipment, PPE accounting, occupational safety, demand forecasting, machine learning, Prophet

Введение

Обеспечение работников средствами индивидуальной защиты – обязательное требование охраны труда, установленное Трудовым кодексом РФ и Приказом Минтруда № 766н от 29 октября 2021 года. Касается это любой организации, где есть вредные или опасные условия: производство, строительство, transport и смежные отрасли. Учет при этом обязателен – работодатель фиксирует каждую операцию выдачи, контролирует сроки замены и ведет личные карточки установленной формы. Нарушения влекут административную ответственность; если произошел несчастный случай – становятся аргументом против работодателя при расследовании. На практике этот процесс автоматизирован частично. Данные о сотрудниках – в кадровой системе, нормы выдачи – в Excel, складские операции – в 1С, личные карточки заполняет специалист по охране труда вручную, сверяя несколько источников [1, 2].

Когда сотрудника переводят в другое подразделение, кадровая система отражает изменение, но нормы выдачи и незакрытые сроки остаются в таблице, которую никто не обновил. В итоге специалист либо выявляет расхождения вручную при плановой проверке, либо не выявляет вообще. Отдельная задача – планирование потребности. Стандартный подход: умножить норму на численность сотрудников по каждой профессии. Это дает расчетную цифру, но не фактическую. На реальный объем выдачи влияют износ конкретных партий, интенсивность эксплуатации, досрочные замены и сезонность. Одни позиции расходуются быстрее нормы, другие – медленнее. Итог – периодический дефицит одних СИЗ и затоваривание других: и риск для безопасности, и лишние расходы на хранение. Точность планирования повышается, если опираться на фактическую историю выдач, а не только на нормативный расчет. Методы прогнозирования временных рядов учитывают реальные паттерны потребления и сезонность. Но для этого нужна достаточно длинная история данных и инструмент, доступный специалисту по охране труда без навыков программирования. Цель работы – разработать систему, которая объединяет учет СИЗ, формирование документов, аналитику и прогнозирование в одном приложении. Система должна работать с данными из 1С и закрывать практические задачи специалиста: от оформления личной карточки до расчета заявки на закупку [3, 4].

Исследование предметной области

Учет СИЗ охватывает несколько связанных процессов, каждый из которых требует согласованных данных. Нужно вести актуальные сведения о сотрудниках: табельные номера, подразделения, профессии, даты приема на работу, а для спецодежды и обуви – также размерные характеристики. Отдельно хранятся нормы выдачи: какие средства защиты положены работнику конкретной профессии, в каком количестве, в каких единицах измерения и на какой срок. На основе этих данных фиксируются фактические выдачи и рассчитываются контрольные даты следующей замены. Наконец, по каждому сотруднику формируется личная карточка учета выдачи СИЗ – документ установленной формы, который подтверждает выполнение

работодателем нормативных требований. Проблема в том, что перечисленные данные редко хранятся в одном месте [5]. Кадровая система содержит актуальный состав сотрудников и их профессии. Складская учетная программа фиксирует движение СИЗ как товарных позиций, но не всегда связывает выдачу с конкретным сотрудником и нормой. Нормы выдачи чаще всего живут в Excel-файлах, которые не обновляются автоматически при изменении законодательства или внутренних норм предприятия. Личные карточки ведутся либо в бумажном виде, либо в отдельной таблице, которую специалист заполняет вручную на основе данных из всех перечисленных источников. При такой организации нельзя автоматически ответить на простые вопросы: какие сроки замены истекают в следующем месяце, сколько единиц конкретной позиции потребуется на квартал, у кого из сотрудников карточка заполнена некорректно. Получение этих данных требует ручной сверки нескольких источников, что занимает значительное время и сопряжено с ошибками.

Существующие варианты автоматизации можно разделить на несколько групп. Первая – корпоративные системы, прежде всего решения на платформе 1С. Конфигурации «Зарплата и управление персоналом» и «Бухгалтерия предприятия» содержат кадровые и складские данные, необходимые для учета СИЗ, но специализированного модуля для ведения норм, личных карточек и контроля сроков в стандартных конфигурациях нет. Существуют отраслевые расширения, однако их внедрение требует существенных затрат на настройку и сопровождение. Вторая группа – специализированные системы по охране труда. Они лучше отражают нормативные требования этой области, но нередко плохо интегрируются с действующими кадровыми и складскими системами, что порождает дублирование данных. Третья группа – локальные решения на основе Excel и небольших баз данных. Они быстро внедряются и не требуют IT-поддержки, но при увеличении числа сотрудников и позиций СИЗ становятся неудобными: файлы растут, связи между таблицами нарушаются, а ручное обновление данных отнимает все больше времени. Таким образом, в большинстве случаев на предприятии нет единого инструмента, который бы связывал кадровые данные, нормы, факты выдачи и планирование закупок. Логичный подход – отдельное прикладное приложение, которое использует данные из корпоративных систем через стандартные форматы обмена и при этом специализируется на задачах учета СИЗ: хранение связанных данных, контроль сроков, формирование документов, аналитика и расчет потребности.

При разработке модуля прогнозирования важно понимать природу спроса на СИЗ. Большинство средств защиты не расходуется ежедневно равномерно. Выдача происходит партиями: когда истекает срок носки, при досрочном износе, при найме новых сотрудников или изменении состава рабочих мест. Временные ряды по отдельным позициям получаются прерывистыми – скользящее среднее или линейный тренд на таких данных работают плохо: они не видят периодичность замен и сезонные пики. Prophet строит аддитивную модель с несколькими уровнями сезонности и нормально справляется с пропусками и нерегулярными выбросами – именно такая структура типична для истории выдачи СИЗ.

Методы и технологии

Система – модульный монолит с веб-интерфейсом и файловым хранением в CSV и Parquet. Для учебно-практического проекта это оправданный выбор: приложение разворачивается локально без серверной инфраструктуры, каждый модуль можно дорабатывать независимо, а данные в любой момент открываются и проверяются в обычном табличном редакторе. При переходе к промышленной эксплуатации файловое хранение заменяется на реляционную базу данных без изменения логики модулей – именно для этого разграничение ответственности между модулями выдержано с самого начала. Структура приложения включает следующие модули: загрузки данных, подготовки данных, формирования документов, прогнозирования, контроля сроков, уведомлений и интеграции с 1С. Каждый модуль имеет собственный раздел в Streamlit и работает с подготовленными наборами из рабочего каталога. Порядок работы линейный: сначала загрузка и подготовка данных, затем аналитика и документы [6, 7]. Основной внешний источник – 1С. Большинство организаций ведут в ней кадровый и складской учет, поэтому без интеграции с 1С система останется изолированной от актуальных данных. Получить их можно двумя способами: CSV-файлы, экспортированные из 1С вручную, или автоматический обмен через REST/OData API, доступный в современных конфигурациях 1С:Предприятие 8. Перед сохранением система проверяет обязательные поля, форматы дат и соответствие структуры ожидаемой схеме.

Модель данных строится вокруг трех основных сущностей. Сущность «Сотрудник» хранит табельный номер, подразделение, должность, профессию, дату приема на работу и, при необходимости, размерные характеристики – рост, размер одежды, размер обуви, размер головного убора. Сущность «Норма выдачи» связывает профессию с конкретными позициями СИЗ: наименование, количество, единица измерения, срок носки в месяцах. Сущность «Фактическая выдача» хранит дату операции, идентификатор сотрудника, наименование позиции СИЗ, количество и основание – плановая замена, досрочная или первичная выдача. Из этих трех сущностей модуль подготовки строит рабочие наборы: расширенную историю с рассчитанными контрольными датами, помесечные агрегированные ряды по позициям и обучающий набор для Prophet.

Стек реализации – Python 3.11. Pandas обрабатывает табличные данные: загружает, чистит, объединяет, агрегирует. Streamlit дает веб-интерфейс с фильтрами, таблицами и графиками – без отдельного frontend. Личные карточки формирует python-docx: заполняет DOCX-шаблон данными выбранного сотрудника и сохраняет файл. Уведомления уходят по двум каналам – электронная почта через SMTP и Telegram через Bot API. Прогнозирование выполняет библиотека Prophet, разработанная в Meta. Prophet строит аддитивную модель временного ряда, раскладывая его на тренд, сезонность нескольких уровней и компоненту праздников. Для учета СИЗ это удобно: модель хорошо работает с рядами, в которых есть пропуски и нерегулярные выбросы, что типично для истории выдачи спецодежды и средств защиты. Пользователь выбирает позицию СИЗ, задает

горизонт прогноза в месяцах и получает прогнозные значения с доверительным интервалом, которые отображаются в виде таблицы и графика. Для позиций с коротким рядом – менее шести точек агрегации – система предупреждает о недостаточности данных и не запускает прогноз, чтобы не вводить пользователя в заблуждение ненадежным результатом. Подготовка данных перед прогнозированием – обязательный этап [8, 9]. Система проверяет обязательные поля в каждой загруженной таблице, приводит имена колонок к единому виду, обрабатывает даты в нескольких форматах, объединяет выдачи с данными о сотрудниках и нормах через табельный номер и профессию, рассчитывает контрольные даты как сумму даты выдачи и срока носки в месяцах, затем агрегирует выдачи по позициям и периодам для построения временного ряда. Если этот этап пропустить, исходные учетные файлы содержат слишком много расхождений и пропусков.

Результаты

Система включает восемь функциональных разделов: загрузка данных, просмотр истории выдачи, анализ потребления, формирование личных карточек, контроль сроков, уведомления, интеграция с 1С и прогнозирование потребности. При загрузке пользователь поочередно предоставляет три файла: список сотрудников, нормы выдачи по профессиям и историю фактических выдач. Система проверяет каждый файл: наличие обязательных колонок, типы данных, корректность дат и отсутствие дублирующих записей. Если при загрузке обнаружена проблема – система показывает строку и характер ошибки прямо в интерфейсе. Пользователь исправляет исходный файл, а не разбирается потом, почему аналитика выдает странные цифры. Прошедшие проверку таблицы сохраняются в рабочем каталоге в формате Parquet [10].

Модуль подготовки объединяет три таблицы в единый набор: связывает выдачи с сотрудниками по табельному номеру и с нормами по профессии, рассчитывает для каждой записи дату следующей выдачи и строит помесечный агрегированный ряд по каждой позиции СИЗ. Этот ряд используют и аналитический раздел, и модуль прогнозирования. Итоговую таблицу выдач можно фильтровать по сотруднику, подразделению, позиции и периоду. Раздел анализа потребления показывает динамику выдачи по выбранной позиции в виде столбчатой диаграммы за произвольный период. Рядом – сводка по подразделениям: сколько единиц выдано каждому и какова средняя интенсивность потребления. Это помогает выявлять подразделения с повышенным расходом и заблаговременно увеличивать для них плановый объем закупки.

Формирование личной карточки выполняется в несколько кликов: пользователь выбирает сотрудника из списка, система подставляет его данные и связанные нормы в DOCX-шаблон, заполняет табличную часть записями о выдачах из расширенной истории и сохраняет готовый документ. Шаблон соответствует форме личной карточки учета выдачи СИЗ, рекомендованной нормативными документами. Автоматическое заполнение убирает ручной ввод – а значит, и опечатки, и пропущенные строки в юридически значимом документе.

Раздел контроля сроков показывает записи, у которых замена наступает в течение заданного числа дней. По умолчанию горизонт – 30 дней, пользователь может изменить его. Таблица отсортирована по дате и показывает сотрудника, подразделение, позицию СИЗ и сколько дней осталось до замены. Рядом – счетчик: сколько позиций требует внимания в этом периоде. Это переводит контроль из реактивного режима – «карточка уже просрочена» – в предупредительный. Модуль уведомлений работает с теми же контрольными датами. Пользователь задает горизонт и выбирает каналы: электронная почта или Telegram. Система собирает список позиций, замена которых наступает в указанный период, и рассылает его по настроенным адресам и чатам. Ответственному за закупки не нужно каждый день открывать приложение – информация приходит сама.

Прогнозный модуль строит временной ряд по выбранной позиции СИЗ из агрегированных данных и передает его в Prophet. Пользователь задает горизонт прогноза – от одного до двенадцати месяцев. Модель обучается на имеющейся истории и возвращает прогнозные значения с нижней и верхней границами доверительного интервала. Результат выводится как таблица с прогнозными значениями по периодам и график, на котором видны фактическая история, прогнозный тренд и доверительный интервал. Суммарный прогноз по горизонту можно использовать как ориентир при формировании заявки – с поправкой на текущий складской остаток.

Систему проверяли на демонстрационном наборе: 50 сотрудников в трех условных подразделениях, 12 профессий, 20 позиций СИЗ, история выдачи за 24 месяца. Проверили загрузку и валидацию файлов, формирование расширенной истории, генерацию карточек, расчет контрольных сроков и прогноз по пяти позициям с разными паттернами потребления. Все отработало корректно. Из очевидных ограничений: данные хранятся в файлах, а не в базе данных – при параллельной работе нескольких пользователей это создаст проблемы. Разграничение прав не реализовано. Набор автоматизированных тестов охватывает модуль подготовки данных и прогнозирования, но не весь функционал. Точность прогноза существенно зависит от длины и качества исторических данных: для позиций с историей менее года результаты ненадежны.

Заключение

Учет СИЗ – задача, у которой уже есть решения: модули 1С, специализированные системы охраны труда, Excel. Ни одно из них не закрывает задачу целиком: данные хранятся раздельно, документы заполняются вручную, потребность считается по нормам без учета фактического потребления. В этой работе разработана система, которая объединяет учет, документооборот и прогнозирование в одном приложении. Нормативные требования к учету СИЗ определили структуру данных – сотрудники, профессии, нормы, выдачи. Prophet добавил возможность строить прогноз потребности на основе фактической истории. Streamlit сделал все это доступным без навыков программирования. Прототип запускается локально и не требует серверной инфраструктуры. Чтобы перейти к промышленному применению, нужна

база данных вместо файлов, ролевая модель и регламентная синхронизация с 1С – задачи понятные, и архитектура системы рассчитана на этот переход.

Разработанная система объединяет в одном приложении данные о сотрудниках, профессиях, нормах выдачи, фактических операциях, контрольных сроках, юридически значимых документах и прогнозной потребности. Это позволяет специалисту по охране труда работать с единым источником данных вместо нескольких несвязанных таблиц и систем. Главный практический результат – не сама система, а то, что специалист по охране труда перестает тратить время на сборку данных из разных источников. Личные карточки формируются за секунды вместо получаса ручного заполнения. Сроки повторной выдачи уже рассчитаны и видны в интерфейсе – их не нужно искать по таблицам. Прогноз Prophet дает не точный ответ, а обоснованную отправную точку: специалист видит, сколько единиц конкретной позиции потребуется в следующем квартале, и корректирует эту цифру исходя из собственного опыта. Текущая реализация – прототип. Чтобы система работала в корпоративной среде, потребуется перейти от файлового хранения к базе данных, добавить разграничение прав между специалистом по охране труда, кладовщиком и руководителем, настроить регулярную синхронизацию с 1С и расширить тестовое покрытие [11, 12]. Это решаемые задачи – архитектура системы изначально рассчитана на такой переход.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Трудовой кодекс Российской Федерации от 30.12.2001 № 197-ФЗ (ред. от 01.04.2024).
2. Приказ Министерства труда и социальной защиты Российской Федерации от 29 октября 2021 г. № 766н «Об утверждении Правил по охране труда при использовании отдельных видов средств индивидуальной защиты и смывающих средств».
3. ГОСТ Р 12.4.280-2019. Система стандартов безопасности труда. Одежда специальная для защиты от общих производственных загрязнений и механических воздействий. Общие технические требования.
4. Методические рекомендации по организации учета и контроля за обеспечением работников средствами индивидуальной защиты. – М.: Минтруд России, 2022.
5. Письмо Федеральной службы по труду и занятости от 05.02.2022 № 14-2/10/В-1123 «О порядке ведения личных карточек учета выдачи средств индивидуальной защиты».
6. Рамм В.М., Котляр А.Е. Автоматизация процессов охраны труда на предприятиях. – Москва: Альфа-Пресс, 2023. – 284 с.
7. Streamlit Documentation. URL: <https://docs.streamlit.io/> (дата обращения: 28.05.2026).
8. Prophet Documentation. Forecasting at scale. Meta. URL: <https://facebook.github.io/prophet/> (дата обращения: 28.05.2026).
9. Taylor S.J., Letham B. Forecasting at scale // PeerJ Preprints. 2017.
10. Pandas Development Team. pandas-dev/pandas: Pandas. Zenodo, 2024.
11. 1С: Предприятие 8.3. Руководство по разработке и администрированию. – Москва: Фирма «1С», 2024.
12. Системы управления охраной труда: обзор современных решений и подходов к цифровизации / А.В. Иванов, С.П. Сидоров // Безопасность и охрана труда. – 2023. – № 4. – С. 45–52.

© В. А. Зинков, Е. Ю. Воронкин, 2026

В. Д. Злочевский¹✉

Метод сетевого зондирования объектов корпоративной вычислительной сети

¹Национальный Исследовательский Ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: vladzlotchevskiy@mail.ru

Аннотация. В статье рассматривается задача сетевого зондирования объектов корпоративной вычислительной сети для обнаружения уязвимостей и некорректных настроек сервисов, служб и средств защиты информации. Актуальность исследования обусловлена наличием требований к зондированию и поиску уязвимостей со стороны государства и индустрии информационной безопасности, а также недостатками существующих решений: низкая точность определения версий операционных систем в режиме «черного ящика» и отсутствие проверок на возможный спуфинг портов. Цель исследования заключается в разработке системы зондирования, обеспечивающей сбор информации о сетевых узлах, открытых портах, версиях служб и операционных систем, а также наличии средств защиты информации и возможных ошибок конфигураций в них. В результате работы был предложен метод сетевого зондирования и представлен в виде блок-схемы и создана база данных системы сетевого зондирования. Предлагаемый метод позволит снизить число ошибок первого рода при возможном использовании систем сокрытия сервисов.

Ключевые слова: сетевое зондирование, обнаружение уязвимостей, активный анализ сети, определение версий служб и сервисов

V. D. Zlochevskii¹✉

Method for network security probing of corporate network assets

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: vladzlotchevskiy@mail.ru

Abstract. The article addresses the task of network probing of corporate computing network assets for detecting vulnerabilities and misconfigurations of services, daemons, and network security systems. The relevance of the study is due to the presence of requirements for information gathering and vulnerability detection from the state and the information security industry, as well as the shortcomings of existing solutions: low accuracy of operating system version identification in black-box mode and the lack of checks for possible port spoofing. The aim of the study is to develop a probing system that provides collection of information about network nodes, open ports, versions of services and operating systems, as well as the presence of information security tools and possible configuration errors within them. Because of the work, a network probing method was proposed and presented in the form of a block diagram, and a database for the network probing system was created. The proposed method will reduce the number of type I errors when using service concealment systems.

Keywords: network probing, vulnerability detection, network active analysis, service version identification

Введение

Информационная безопасность является состоянием защищенности информации от несанкционированного доступа, изменения или уничтожения, соответственно процесс ее обеспечения обязан носить непрерывный и системный характер. Для оценки уровня защищенности информационных систем применяют различные методы анализа сетевого трафика, среди которых два основных: активный и пассивный. Активный анализ предполагает взаимодействие с целевой системой. На этом принципе работают инструменты тестирования на проникновение и сетевого зондирования. Под сетевым зондированием понимается процесс, при котором имитируются различные атаки и отслеживается реакция объекта с идентификацией эшелонов его защиты [1].

Актуальность темы исследования со стороны индустрии информационной безопасности обусловлена наличием пунктов, связанных с зондированием и поиском уязвимостей в общепризнанных мировых методологиях тестирования на проникновения, таких как: OWASP [2], OSSTMM [3], NIST 800-115 [4] и PTES [5].

В научной среде существует несколько групп, изучающих методы активного анализа трафика и зондирования устройств на наличие уязвимостей и некорректных настроек служб, сервисов и средств защиты информации. Одна из них во главе с создателем утилиты nmap Лионом [6] в своих работах предлагают варианты сканирования устройств, определения версий служб, сервисов и операционных систем с помощью специально сформированных пакетов данных. Основные проблемы существующих решений в данной области связаны с низкой точностью определения семейства операционных систем в режиме «черного ящика» и их непосредственной версии из-за схожести получаемых данных и регулярных обновлений. Также, авторы целенаправленно не используют методы пассивного анализа трафика и не имеют модулей для проверки безопасности настройки средств защиты информации и определения их вендора, что осложняет автоматизированный подбор возможных полезных нагрузок специалистами по тестированию на проникновение. Соответственно, в работе предлагается совместить методы активного и пассивного анализа трафика, что может повысить метрики обнаружения операционных систем как минимум при нахождении в локальной сети.

Одной из проблем активного анализа трафика является спуфинг портов – процесс, при котором автоматизированными системами [7], либо с помощью правил iptables и xtables-addons имитируются сервисы на закрытых портах. Существующие сканеры уязвимостей и системы сетевого зондирования не реализуют дополнительные проверки на данный тип противодействия, что понижает точность определения версий сервисов.

Как было отмечено ранее, одной из задач зондирования является идентификация эшелонов защиты. В работах [8, 9] исследуются межсетевые экраны и их реакция на определенные виды запросов. Из основных недостатков можно отметить искажение результатов по причине сетевых задержек, сложности их идентификации при изучении методом «черного ящика» и ограниченности данных

алгоритмов, так как уязвимыми является лишь определенный тип межсетевых экранов.

В связи с расширением процесса сетевого зондирования и комбинации алгоритмов пассивного и активного анализа, были рассмотрены основные работы в сфере обнаружения версий операционных систем методами пассивного анализа трафика. Так, в работе научной группы Ластовички [10] разбираются основные методы пассивного анализа трафика для определения возможной операционной системы устройства и выделяются основные характеристики протоколов канального, транспортного и прикладного уровня модели OSI. Также, были исследованы работы [11, 12] и анализ TLS-шифров до версии 1.2 включительно был добавлен в список значений для сбора статистики об операционной системе.

Итоговая цель работы заключается в разработке метода и системы сетевого зондирования объектов корпоративной вычислительной сети для обнаружения уязвимостей и некорректных настроек сервисов и служб. Основными задачами исследования являлись: разработка собственного метода сетевого зондирования, сочетающего в себе алгоритмы пассивного и активного анализа трафика и создание собственной базы данных, содержащей в себе основные характеристики используемых сервисов, служб, операционных систем и средств защиты информации.

Предлагаемый метод сетевого зондирования

Итоговый метод сетевого зондирования представлен на рис. 1 в виде блок-схемы. Параллельно ведется пассивный сбор метрик, таких как:

- 1) TTL, TOS, DF;
- 2) MAC-адреса;
- 3) порядок TCP-опций и размер окна TCP;
- 4) HTTP User-Agent, набор шифров TLS (до версии 1.2 включительно);
- 5) действующие в сети протоколы.

В качестве алгоритма поиска активных устройств используются классические алгоритмы ICMP Echo Request и Timestamp Request, в локальных сетях отправляется ARP-запрос, а для внешних сетей используется отправка TCP-SYN на 22, 80 и 443 порты и ожидания реакции на него. Открытые порты ищутся классическими методами – полуоткрытое TCP-сканирование, суть которого заключается в отправке SYN-запроса на интересующий порт. В случае, если сервер отвечает SYN-ACK, то считаем, что порт открыт и посылаем RST, тем самым завершая соединение без полноценного установления. Для определения открытого UDP-порта используется некоторый набор заготовленных дейтаграмм и ожидание ответа от целевых портов.

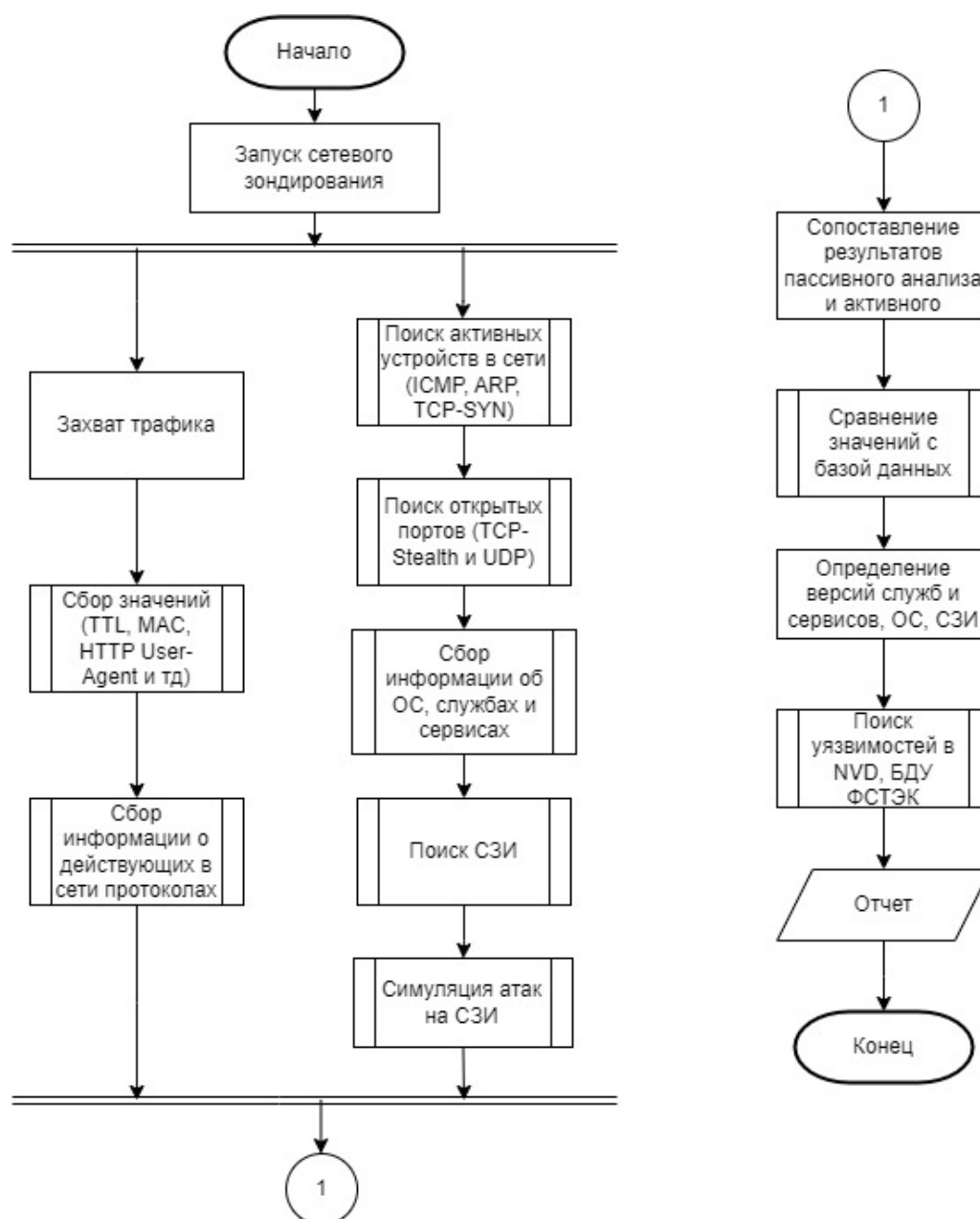


Рис. 1. Блок-схема метода сетевого зондирования объектов корпоративной вычислительной сети

Под сбором информации об ОС, службах и сервисах подразумевается алгоритм отправки специально сформированных запросов для разных сервисов, где также учитывается возможный спуфинг портов (рис. 2). Повторный запрос баннера сервиса через случайное время и с измененными параметрами подключения позволит обмануть системы автоматической маскировки портов, тем самым повысив точность их определения за счет понижения числа ложноположительных срабатываний сканера.

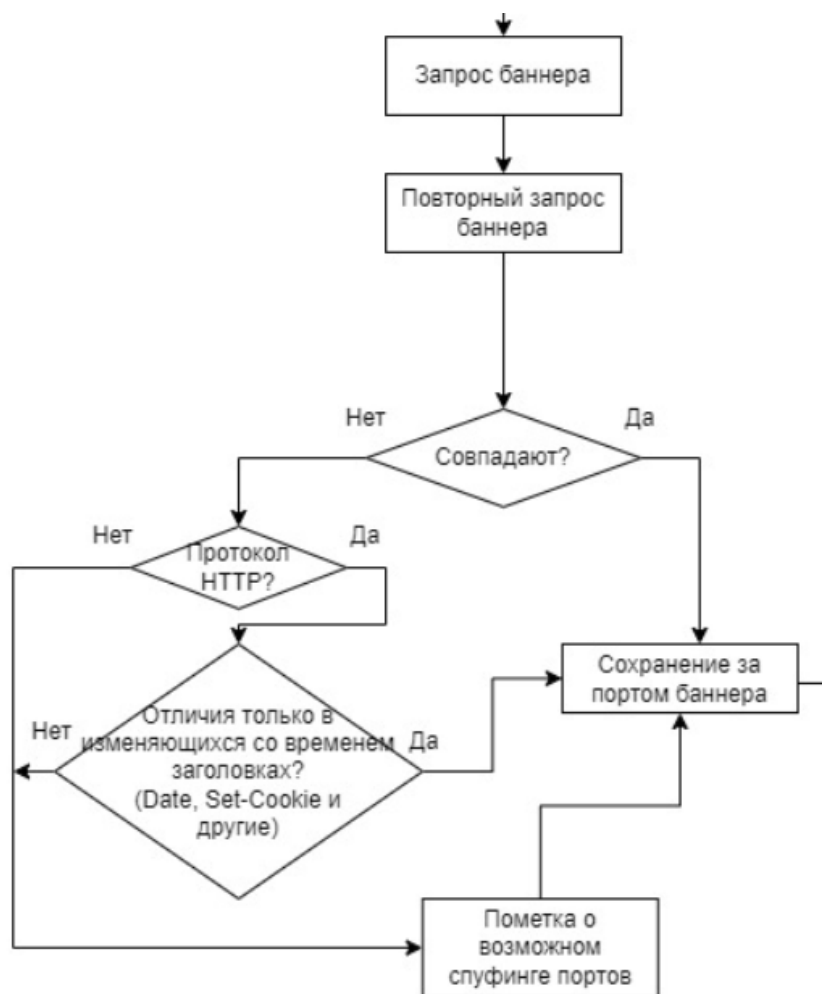


Рис. 2. Часть блок-схемы алгоритма сборки информации о сервисах, связанная с проверкой на спуфинг портов

В метод дополнительно заложены алгоритмы поиска уязвимостей и некорректных настроек часто используемых сервисов. В их число входит DNS, IKE, SNMP, SMTP и ряд других (всего на данный момент 20). Как пример, представлена блок-схема алгоритма проверки DNS на основные ошибки конфигурации (рис. 3). Поочередно производятся проверки на уязвимость к распределенной атаке в обслуживании, раскрытия информации через трансфер зон, отравление кэша и раскрытия информации через этот кэш. Также, их можно реализовать параллельно, тем самым повысив скорость проверки DNS-сервера, но повышает нагрузку на него.

Стоит отметить, что поиск CVE или уязвимостей из БДУ ФСТЭК происходит уже на этапе после сопоставления результатов пассивного и активного анализа, то есть в данном случае даже если в базе данных системы не будет какой-то записи с версией целевого сервера, либо если система не сможет определить ее, то основные проверки будут протестированы.

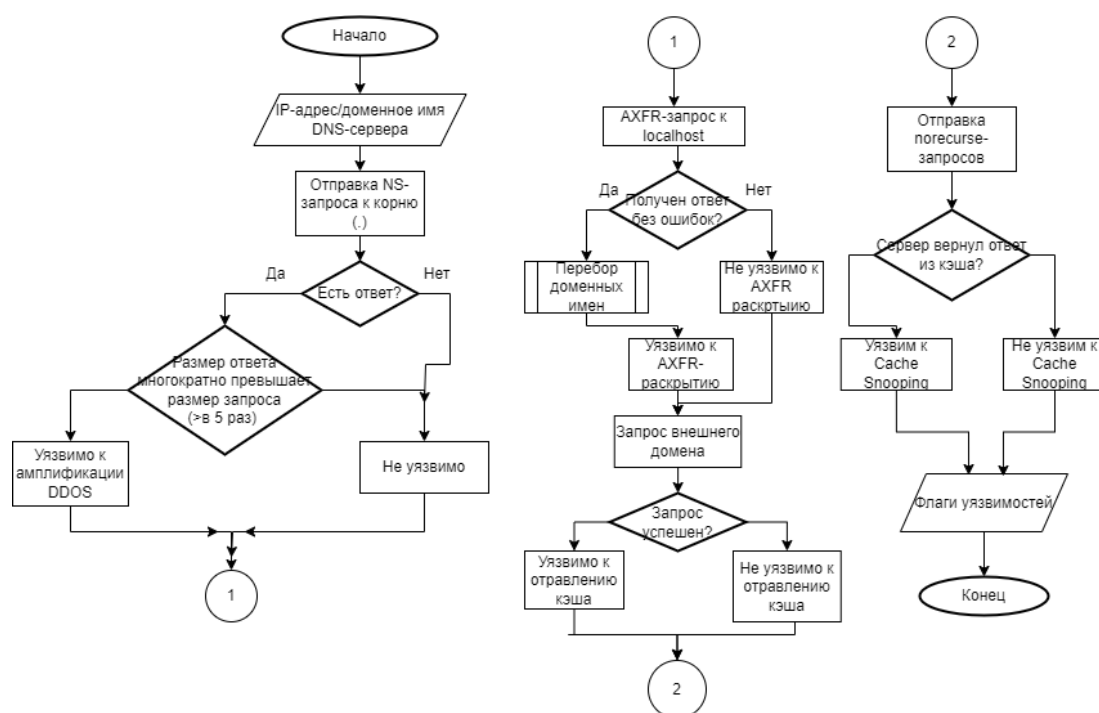


Рис. 3. Алгоритм проверки DNS-сервера на основные ошибки конфигурации

На текущий момент предложено два алгоритма поиска СЗИ: это межсетевые экраны и WAF. Поиск WAF заключается в отправке набора вредоносных полезных нагрузок (межсайтовый скриптинг, SQL-инъекция и другие) и анализ HTTP-ответов. Алгоритм же поиска и сбора метаданных межсетевого экрана представлен на рис. 4. Одна из симулируемых атак помечена отдельным процессом – BlackNurse. По ее успешности возможно определить stateful или stateless межсетевого экран, а также есть возможность узнать его вендора вплоть до версии ОС внутри.

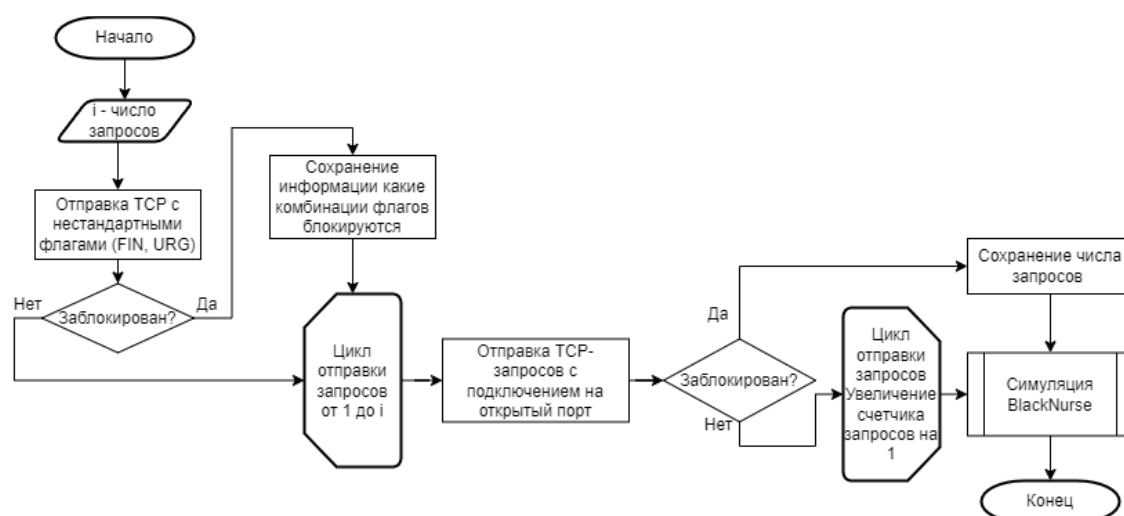


Рис. 4. Блок-схема алгоритма поиска межсетевых экранов

Стоит отметить, что в случае отсутствия четких показателей определенной операционной системы, в системе заложен поиск операционной версии по информации, полученной из открытых портов. Например, у Ubuntu отличаются пакеты (ssh, openssl) в зависимости от ее минорной версии, что было использовано в качестве дополнительного фактора при определении ОС.

Обсуждение

В отличие от лежащих в основе ряда существующих инструментов сетевого зондирования и сканеров уязвимостей, предложенный метод реализует собственные алгоритмы анализа ответов сетевых сервисов и проверки их поведения. Особое внимание в текущей работе уделяется механизму обнаружению признаков спуфинга портов и маскировки сетевых сервисов, что повышает точность определения версий за счет понижения числа ошибок первого рода. Также, созданная база данных содержит записи с информацией об отечественных средствах защиты информации, что отсутствуют во всех аналогичных инструментах в открытом доступе. Пассивный анализ предлагается использовать при возможности внедрения сниффера в исследуемую сеть – это возможно сделать при работах аудита и тестировании на проникновения при дополнительном согласовании. В случае невозможности внедрения, система может работать как классический сетевой зондер, используя исключительно методы активного анализа.

Заключение

В ходе исследования была рассмотрена задача сетевого зондирования объектов корпоративной вычислительной сети, необходимой для поиска уязвимостей и мисконфигураций сервисов и служб.

В результате работы был предложен метод сетевого зондирования объектов корпоративной вычислительной сети для поиска уязвимостей сервисов и служб, сочетающий в себе активный и пассивный анализ трафика. Также, была создана база данных системы, содержащей в себе более 1000 записей с известными веб-серверами, почтовыми сервисами, операционными системами и средствами защиты информации.

На текущем этапе разработан программный прототип системы зондирования, работающей на основе предложенного метода и проведено ее первичное тестирование в тестовой среде, подтвердившее ее работоспособность и корректность взаимодействия компонентов. Однако, подробные результаты исследования будут представлены в дальнейшем после завершения исследования ее работы по сравнению с другими решениями. В дальнейшем, возможна интеграция новых алгоритмов, связанных с поиском скрытых сервисов и межсетевых экраном, а также обогащение базы данных системы сетевого зондирования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Басыня Е.А. Сетевая информационная безопасность: Учебник. – М.: НИЯУ МИФИ, 2023. – 224 с.
2. OWASP Testing Guide. Version 4.2. – 2020. – URL: https://www.owasp.org/index.php/OWASP_Testing_Project (дата обращения 14.11.2025)

3. Herzog P. OSSTMM – The Open Source Security Testing Methodology Manual. – New York: 2006. – 129 с. – URL: <https://www.isecom.org/OSSTMM.3.pdf> (дата обращения: 13.11.2024)
4. NIST 800-115. Technical Guide to Information Security Testing and Assessment - NIST, 2008
5. PTES Technical Guidelines: Penetration Testing Execution Standard [Электронный ресурс]. — URL: <http://www.pentest-standard.org/>. — PTES — (Дата обращения: 21.03.2026).
6. Lyon G. F. Nmap network scanning: The official Nmap project guide to network discovery and security scanning. – Insecure, 2009.
7. Piotr Duszyński. Portspooof. - URL - <https://github.com/drklwi/portspooof> (дата обращения 20.03.2026)
8. Krishna T. V., Karthik P. Dominance of hardware firewalls and denial of firewall attacks (case study blacknurse attack) //International Journal of Science and Research (IJSR). – 2022. – Т. 11. – С. 28-33.
9. Liu A. X. et al. Firewall fingerprinting and denial of firewalling attacks //IEEE Transactions on information forensics and security. – 2017. – Т. 12. – №. 7. – С. 1699-1712.
10. Laštovička M. et al. Passive operating system fingerprinting revisited: Evaluation and current challenges // Computer Networks. – 2023. – Т. 229. – С. 109782
11. Hagos D. H. et al. Advanced passive operating system fingerprinting using machine learning and deep learning //2020 29th International Conference on Computer Communications and Networks (ICCCN). – IEEE, 2020. – С. 1-11
12. Muehlstein J. et al. Analyzing HTTPS encrypted traffic to identify user's operating system, browser and application // 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC). – IEEE, 2017. – С. 1-6

© В. Д. Злочевский, 2026

И. А. Зыков¹✉, Е. Ю. Воронкин¹

Разработка веб-приложения для автоматизации клиентских коммуникаций с поддержкой интеграции во внешние сервисы

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: heler.2004@gmail.com

Аннотация. В статье представлены основные результаты разработки веб-приложения для автоматизации клиентских коммуникаций с поддержкой интеграции во внешние сервисы. Рассмотрена проблема фрагментарного хранения клиентских контактов, задач и договорённостей в мессенджерах, таблицах и заметках. Для решения данной проблемы была разработана CRM-платформа, объединяющая веб-интерфейс, базу данных, Telegram-бота, VK-бота и модуль автоматических напоминаний. В качестве основного стека технологий использованы Python, FastAPI, Jinja2, SQLAlchemy, SQLite, Aiogram, VKBottle и APScheduler. Разработанное приложение позволяет создавать и редактировать лидов, изменять статусы обработки, добавлять задачи с дедлайнами, получать напоминания о просроченных задачах, а также добавлять лидов через внешние сервисы на примере Telegram и VK.

Ключевые слова: CRM-система, веб-приложение, клиентские коммуникации, FastAPI, Telegram-бот, VK-бот, SQLite, автоматизация

I. A. Zykov¹✉, E. Yu. Voronkin¹

Development of a web application for automating customer communications with support for integration into external services

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: heler.2004@gmail.com

Abstract. The article presents the main results of the development of a web application for automating customer communications with support for integration into external services. The problem of fragmented storage of customer contacts, tasks and agreements in messengers, spreadsheets and notes is considered. To solve this problem, a CRM platform was developed, combining a web interface, a database, a Telegram bot, a VK bot and an automatic reminder module. Python, FastAPI, Jinja2, SQLAlchemy, SQLite, Aiogram, VKBottle and APScheduler were used as the main technology stack. The developed application allows users to create and edit leads, change processing statuses, add tasks with deadlines, receive reminders about overdue tasks, and add leads through external services using Telegram and VK as examples.

Keywords: CRM system, web application, customer communications, FastAPI, Telegram bot, VK bot, SQLite, automation

Введение

В современных условиях значительная часть взаимодействия с потенциальными клиентами осуществляется через цифровые каналы связи: мессенджеры, социальные сети, электронную почту и веб-формы. Такой формат коммуникации удобен для быстрого контакта, однако при отсутствии централизованной информационной

системы приводит к фрагментации данных. Контакты клиентов, договорённости, заметки, задачи и сроки выполнения могут храниться в разных чатах, личных сообщениях, таблицах или текстовых документах. В результате повышается риск потери информации, дублирования записей и пропуска важных действий по клиенту.

Для организаций и специалистов, работающих с входящими заявками, особенно важны оперативная фиксация лида, контроль его текущего статуса и своевременное выполнение последующих задач. Использование только таблиц или заметок не обеспечивает достаточного уровня связности данных, а внедрение крупных CRM-систем может быть избыточным для небольшой компании. В связи с этим актуальной становится разработка прикладной CRM-платформы, ориентированной на базовые сценарии работы с клиентскими коммуникациями и поддерживающей интеграцию с внешними сервисами.

Целью работы является разработка веб-приложения для автоматизации клиентских коммуникаций с поддержкой интеграции во внешние сервисы. В рамках проекта была реализована CRM-платформа, позволяющая вести список лидов, работать с карточками клиентов, назначать задачи, контролировать дедлайны и получать напоминания. В качестве внешних сервисов были рассмотрены мессенджеры Telegram и VK, через которые пользователь может быстро добавить нового лида в систему.

Исследование предметной области

CRM-системы используются для организации работы с клиентами, фиксации обращений, ведения истории взаимодействий, контроля статусов и повышения управляемости процессов продаж и сопровождения [1]. Основная задача таких систем заключается в том, чтобы объединить разрозненные сведения о клиентах в едином информационном пространстве. Это особенно важно в тех случаях, когда коммуникация ведётся через несколько каналов и менеджеру необходимо быстро понимать, на каком этапе находится каждый клиент.

На практике можно выделить несколько распространённых подходов к ведению клиентских коммуникаций. Первый подход заключается в использовании таблиц, заметок и отдельных текстовых файлов. Его преимущество состоит в простоте начала работы и отсутствии необходимости внедрения специализированного программного обеспечения. Однако такой способ плохо масштабируется, не обеспечивает полноценного контроля задач и не позволяет удобно отслеживать статус каждого лида.

Второй подход связан с ведением коммуникации непосредственно в мессенджерах. Telegram, VK и другие сервисы позволяют быстро получать сообщения от клиентов и оперативно отвечать на них. Однако мессенджер сам по себе не является CRM-системой: в нём сложно структурировать лидов, назначать статусы, контролировать дедлайны и формировать общую картину работы с клиентской базой.

Третий подход заключается во внедрении готовых CRM-систем. Такие решения обладают широким функционалом, но могут быть избыточными для небольших проектов, учебных задач или MVP. Кроме того, их настройка часто

требует времени, обучения пользователей и адаптации бизнес-процессов под конкретную платформу.

С учётом рассмотренных ограничений было принято решение разработать собственную CRM-платформу в формате MVP. Такой подход позволяет реализовать только необходимые функции: создание лида, хранение контактных данных, изменение статуса, добавление задач, контроль дедлайнов и интеграцию с внешними сервисами. При этом система остаётся достаточно простой для пользователя и может быть расширена в дальнейшем.

Методы и технологии

Разработка CRM-платформы была выполнена с использованием языка программирования Python. Данный язык был выбран благодаря широкому набору библиотек для веб-разработки, работы с базами данных, создания ботов и планирования фоновых задач. Такой подход позволил реализовать основные компоненты системы в едином технологическом стеке.

В качестве веб-фреймворка был выбран FastAPI. Он используется для создания серверной части приложения, обработки HTTP-запросов, маршрутизации и взаимодействия с шаблонами интерфейса. FastAPI хорошо подходит для разработки прикладных веб-сервисов, так как поддерживает современный подход к описанию маршрутов и позволяет быстро создавать структуру серверного приложения [3].

Для формирования пользовательского интерфейса применялись Jinja2, HTML, CSS и JavaScript. Веб-интерфейс CRM предоставляет пользователю доступ к списку лидов, карточкам клиентов, формам создания и редактирования данных, а также к управлению задачами. Такой вариант реализации был выбран из-за простоты и достаточности для MVP-версии системы.

Для хранения данных использовалась база данных SQLite. Она подходит для локального прототипа, так как не требует отдельной серверной установки и позволяет быстро развернуть приложение. В базе данных хранятся сведения о пользователях, лидах и задачах. При дальнейшем развитии проекта возможен переход на PostgreSQL, что позволит повысить устойчивость и масштабируемость системы [4].

Работа с базой данных реализована через SQLAlchemy. Данная библиотека используется как ORM-инструмент, позволяющий описывать модели данных в виде классов Python и выполнять операции создания, чтения, обновления и удаления записей [5]. В проекте были выделены основные сущности: пользователь, лид и задача. Пользователь может иметь несколько лидов, а каждый лид может быть связан с несколькими задачами.

Для интеграции с Telegram был использован фреймворк Aiogram. Он позволяет создавать асинхронных Telegram-ботов и обрабатывать пользовательские команды, текстовые сообщения и контакты [6]. В рамках проекта Telegram-бот реализует сценарий быстрого добавления лида: пользователь вводит имя, username и заметку, после чего данные сохраняются в CRM.

Для интеграции с VK был использован VKBottle. Данный инструмент предназначен для создания ботов и работы с VK API [7]. VK-бот реализует похожий сценарий: пользователь запускает диалог, вводит данные лида, после чего запись сохраняется в общей базе данных и становится доступной в веб-интерфейсе CRM.

Для автоматической проверки просроченных задач применялся APScheduler. Планировщик периодически запускает фоновую задачу, которая проверяет дедлайны и отправляет пользователю напоминания в Telegram при наличии просроченных задач [8]. Это позволяет автоматизировать контроль сроков и снизить риск пропуска важных действий по клиенту.

Таким образом, для разработки CRM-платформы был использован следующий стек технологий:

- язык программирования Python;
- веб-фреймворк FastAPI;
- шаблонизатор Jinja2;
- HTML, CSS и JavaScript для пользовательского интерфейса;
- SQLite для хранения данных MVP-версии;
- SQLAlchemy для работы с базой данных;
- Aiogram для интеграции с Telegram Bot API;
- VKBottle для интеграции с VK API;
- APScheduler для планирования автоматических напоминаний.

Результаты

В результате работы было разработано веб-приложение для автоматизации клиентских коммуникаций с поддержкой интеграции во внешние сервисы. Система реализована как CRM-платформа, объединяющая веб-интерфейс, базу данных, Telegram-бота, VK-бота и механизм автоматических напоминаний.

В базе данных реализованы три основные сущности: users, leads и tasks. Сущность users хранит информацию о пользователях системы, включая идентификаторы Telegram и VK. Сущность leads содержит карточки лидов: имя, username, статус, заметку, дату создания и дату обновления. Сущность tasks используется для хранения задач, связанных с лидами: название задачи, срок выполнения, состояние выполнения и время последнего напоминания.

В веб-интерфейсе реализованы следующие функции:

- авторизация пользователя через внешние сервисы;
- просмотр списка лидов;
- создание нового лида;
- просмотр карточки лида;
- редактирование данных лида;
- изменение статуса обработки лида;
- создание задачи по конкретному лиду;
- отметка задачи как выполненной;
- удаление задач и лидов;
- просмотр информации о просроченных задачах.

Карточка лида является одним из основных элементов CRM. Она объединяет сведения о клиенте, текущий статус обработки, заметку и список связанных задач. Такой подход позволяет менеджеру быстро оценить текущее состояние работы с конкретным клиентом и понять, какое действие необходимо выполнить следующим.

Отдельное внимание было уделено интеграции с мессенджерами. Telegram-бот позволяет пользователю добавить лида без входа в веб-интерфейс. Для этого реализован пошаговый сценарий: пользователь вводит имя лида, username и заметку. После завершения сценария бот сохраняет запись в базу данных и отправляет пользователю сообщение с подтверждением и ссылкой на карточку лида в CRM.

VK-бот реализует аналогичный механизм добавления лида через сообщения сообщества. Пользователь может выбрать действие с помощью кнопок, ввести имя, ссылку или username, а также заметку. После этого информация сохраняется в общей базе данных. Это показывает возможность расширения CRM-платформы на разные внешние каналы коммуникации.

Модуль напоминаний предназначен для контроля просроченных задач. Система периодически проверяет задачи, срок выполнения которых уже прошёл, но которые ещё не были отмечены как выполненные. Если такие задачи обнаружены, пользователь получает уведомление в Telegram. В сообщении отображается список просроченных задач, связанный лид и информация о времени просрочки. Это помогает менеджеру своевременно возвращаться к клиентским действиям и снижает риск потери лида.

Для проверки работоспособности системы был разработан план тестирования. В качестве основного метода использовалось функциональное тестирование методом «чёрного ящика». Данный метод был выбран потому, что проверка выполнялась с позиции пользователя: осуществлялся ввод данных через веб-интерфейс и ботов, после чего ожидаемый результат сравнивался с фактическим. Были проверены сценарии авторизации, создания лида, редактирования карточки, изменения статуса, создания задачи, выполнения задачи и получения напоминания.

Результаты тестирования показали, что основные функции MVP-версии работают корректно. Создание лидов через веб-интерфейс, Telegram-бота и VK-бота приводит к сохранению данных в базе. Изменение статусов и редактирование карточек выполняются корректно. Задачи привязываются к соответствующим лидам, а просроченные задачи отображаются и используются модулем напоминаний.

Разработанная система является MVP-версией CRM-платформы. Она демонстрирует основной принцип автоматизации клиентских коммуникаций: данные о клиенте можно быстро зафиксировать через внешний сервис, а дальнейшая обработка выполняется в веб-интерфейсе CRM. Такой подход объединяет удобство мессенджеров и структурированность CRM-системы.

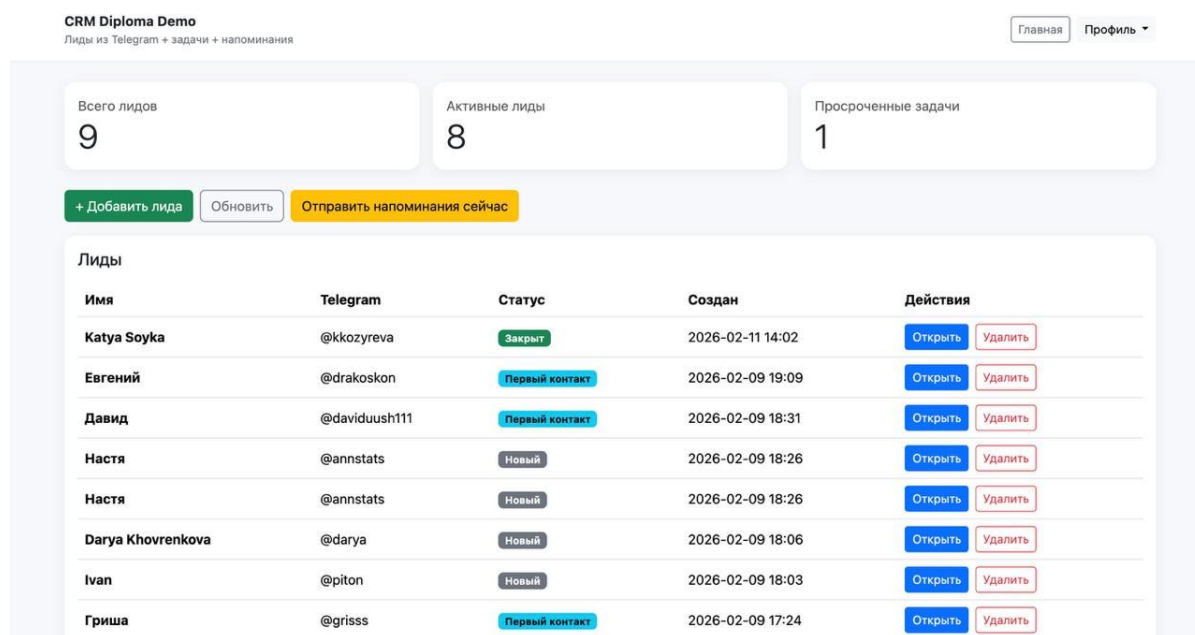


Рис. 1. Интерфейс приложения

Заключение

В результате выполнения работы было разработано веб-приложение для автоматизации клиентских коммуникаций с поддержкой интеграции во внешние сервисы. Реализованная CRM-платформа позволяет создавать и редактировать лидов, изменять статусы обработки, назначать задачи, контролировать дедлайны и получать напоминания о просроченных действиях.

В рамках проекта были решены основные задачи: проведён анализ предметной области, определены требования к системе, спроектирована структура данных, реализован веб-интерфейс CRM, разработаны Telegram-бот и VK-бот, создан модуль автоматических напоминаний и проведено тестирование основных пользовательских сценариев.

Практическая значимость работы заключается в том, что разработанная система может использоваться как основа для дальнейшего развития CRM-платформы. В перспективе возможно расширение функциональности за счёт перехода с SQLite на PostgreSQL, добавления ролей пользователей, реализации истории коммуникаций, аналитики по лидам, резервного копирования и повышения безопасности. Также одним из направлений дальнейшего развития является интеграция с мессенджером MAX, что позволит расширить количество внешних каналов взаимодействия с клиентами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Buttle, F. Customer Relationship Management: Concepts and Technologies / F. Buttle, S. Maklan. – 4th ed. – London : Routledge, 2019. – 444 p.
2. Sommerville, I. Software Engineering / I. Sommerville. – 10th ed. – Boston : Pearson, 2015. – 816 p.

3. FastAPI : official documentation. – URL: <https://fastapi.tiangolo.com/> (дата обращения: 13.05.2026). – Текст : электронный.
4. SQLite Documentation. – URL: <https://sqlite.org/docs.html> (дата обращения: 13.05.2026). – Текст : электронный.
5. SQLAlchemy Documentation. – URL: <https://docs.sqlalchemy.org/> (дата обращения: 13.05.2026). – Текст : электронный.
6. Aiogram documentation. – URL: <https://docs.aiogram.dev/> (дата обращения: 13.05.2026). – Текст : электронный.
7. VKBottle Documentation. – URL: <https://vkbottle.readthedocs.io/ru/latest/> (дата обращения: 13.05.2026). – Текст : электронный.
8. Advanced Python Scheduler : documentation. – URL: <https://apscheduler.readthedocs.io/> (дата обращения: 13.05.2026). – Текст : электронный.

© И. А. Зыков, Е. Ю. Воронкин, 2026

В. Р. Ильясов¹✉, Д. Ю. Смирнов¹

Разработка трехмерного интерактивного пособия по изучению космических исследовательских аппаратов в среде Unreal Engine 5

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: ilasov895@gmail.com

Аннотация. В статье рассматривается процесс разработки трехмерного интерактивного пособия по изучению космических исследовательских аппаратов. В качестве технологической основы выбран Unreal Engine 5, позволяющий реализовать высокодетализированную трехмерную визуализацию, интерактивное взаимодействие с объектами и поддержку современных графических технологий. В рамках работы были разработаны трехмерные модели космических аппаратов, реализована система взаимодействия пользователя с объектами сцены, а также создан интерфейс для изучения технических характеристик исследовательских аппаратов. Особое внимание уделено вопросам оптимизации производительности, удобству пользовательского интерфейса и образовательной ценности программного продукта. Результатом работы стало создание прототипа интерактивного пособия, предназначенного для использования в учебном процессе.

Ключевые слова: unreal engine, интерактивное пособие, трехмерное моделирование, космические аппараты, образовательные технологии, виртуальная среда, 3d-графика, blueprint, интерактивные системы

V. R. Ilyasov¹✉, D. Yu. Smirnov¹

Development of a three-dimensional interactive guide for studying space research vehicles in Unreal Engine 5

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: ilasov895@gmail.com

Abstract. This article examines the development process of a three-dimensional interactive educational application for the study of space research vehicles. Unreal Engine 5 was selected as the technological foundation due to its capability to provide highly detailed three-dimensional visualization, interactive object manipulation, and support for modern graphics technologies. Within the framework of the study, three-dimensional models of spacecraft were developed, a user interaction system for scene objects was implemented, and an interface for studying the technical characteristics of research vehicles was created. Particular attention was paid to performance optimization, user interface usability, and the educational value of the software product. The result of the work is a prototype of an interactive educational application intended for use in the educational process.

Keywords: unreal engine, interactive guide, three-dimensional modeling, spacecraft, educational technologies, virtual environment, 3d graphics, blueprint, interactive systems

Введение

Современные образовательные технологии активно используют интерактивные и визуальные средства представления информации. Особенно актуально применение трехмерной графики и виртуальных сред в техническом образовании, где

требуется демонстрация сложных конструкций и принципов работы различных устройств.

Изучение космических исследовательских аппаратов требует высокой степени наглядности, поскольку реальные образцы техники являются дорогостоящими, труднодоступными или физически недоступными для демонстрации обучающимся. Традиционные методы обучения, основанные на текстовых материалах и статических изображениях, не всегда позволяют в полной мере сформировать представление о конструкции и принципах функционирования космических аппаратов.

В связи с этим возникает необходимость разработки интерактивного трехмерного пособия, позволяющего пользователю взаимодействовать с моделями космических исследовательских аппаратов в виртуальной среде. Использование игрового движка Unreal Engine 5 обеспечивает реалистичную визуализацию, поддержку интерактивных элементов и возможность дальнейшего расширения функционала программного продукта.

Целью настоящей работы является разработка трехмерного интерактивного пособия по изучению космических исследовательских аппаратов в среде Unreal Engine 5.

Для достижения поставленной цели были выполнены следующие задачи:

- анализ существующих образовательных решений и аналогов;
- определение требований к интерактивному пособию;
- разработка структуры и интерфейса приложения;
- создание трехмерных моделей космических аппаратов;
- реализация интерактивного взаимодействия пользователя с объектами;
- тестирование и оптимизация программного продукта.

Методы и материалы

Перед началом разработки трехмерного интерактивного пособия были определены основные функциональные и пользовательские требования к программному продукту. Особое внимание уделялось обеспечению удобного взаимодействия пользователя с виртуальной средой, наглядности представления информации и стабильности работы приложения.

На этапе проектирования была определена структура взаимодействия пользователя с системой. Пользователь получает возможность запускать приложение, выбирать сценарий изучения, управлять камерой, взаимодействовать с объектами виртуальной сцены и просматривать информационные материалы о космических исследовательских аппаратах. Также была предусмотрена возможность возврата в главное меню и переключения между сценами приложения.

Работа системы организована поэтапно. После запуска приложения происходит отображение главного меню, из которого пользователь может перейти к изучению выбранной сцены. При загрузке виртуального пространства выполняется инициализация объектов космической системы, загрузка трехмерных моделей и настройка параметров отображения сцены.

В процессе взаимодействия пользователь может управлять камерой, изменять угол обзора, приближать или отдалять объекты, а также выбирать отдельные элементы сцены для получения дополнительной информации. При выборе объекта система выполняет обработку запроса, получает данные о выбранном космическом аппарате и отображает их в пользовательском интерфейсе.

Для обеспечения стабильной работы приложения была предусмотрена система управления сценами, позволяющая корректно выполнять переходы между разделами пособия и завершать текущие сценарии без потери работоспособности программы.

Для определения структуры приложения и взаимосвязи его компонентов была разработана компонентная диаграмма [3] (рис. 1).

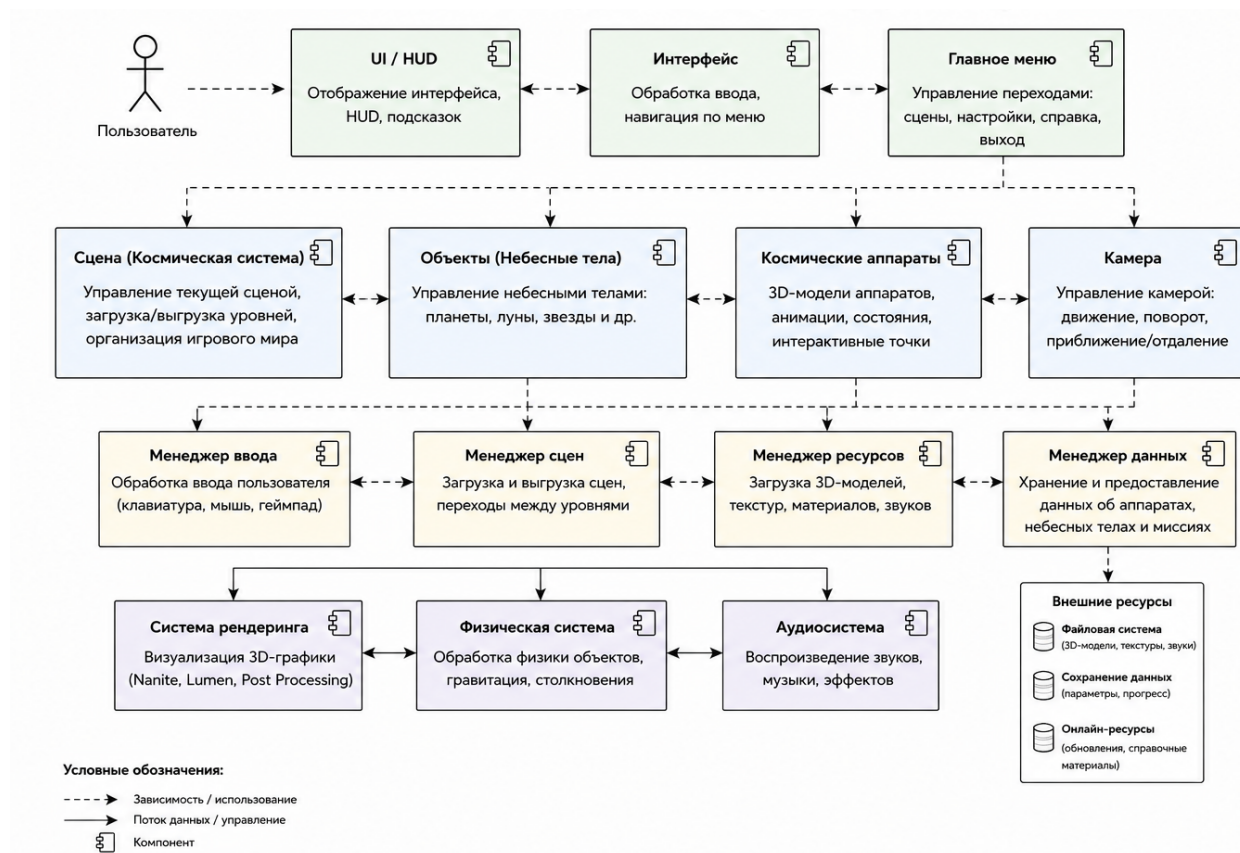


Рис. 1. Компонентная диаграмма интерактивного пособия

На этапе проектирования интерфейса был создан дизайн пользовательского взаимодействия с использованием инструмента Adobe Photoshop (рис. 2).



Рис. 2. Проектирование пользовательского интерфейса

В качестве среды разработки выбран Unreal Engine 5 – современный программный комплекс для создания интерактивных трехмерных приложений, обеспечивающий широкие возможности визуализации. Эта среда позволит реализовать высокодетализированную визуализацию космических аппаратов, обеспечить интерактивное взаимодействие пользователя с объектами сцены и организовать отображение технической информации в режиме реального времени.

Применение Unreal Engine 5 также позволит использовать встроенные средства разработки пользовательского интерфейса, систему визуального программирования Blueprint и инструменты оптимизации, необходимые для стабильной работы приложения в учебном процессе.

Для создания трехмерных моделей космических аппаратов использовались инструменты Blender и встроенные средства импорта Unreal Engine 5. Логика взаимодействия пользователя с объектами реализована с помощью системы визуального программирования Blueprint.

В интерактивном пособии реализованы следующие функции:

- свободное перемещение пользователя по виртуальной сцене;
- взаимодействие с трехмерными моделями;
- отображение технической информации о космических аппаратах;
- масштабирование и вращение моделей;
- система навигации между разделами пособия.

Результаты

В результате выполненной работы было разработано интерактивное трехмерное пособие для изучения космических исследовательских аппаратов.

На начальном этапе были созданы трехмерные модели космических аппаратов, оптимизированные для работы в реальном времени. Модели были импортированы в Unreal Engine 5 и размещены в виртуальной сцене (рис. 3).

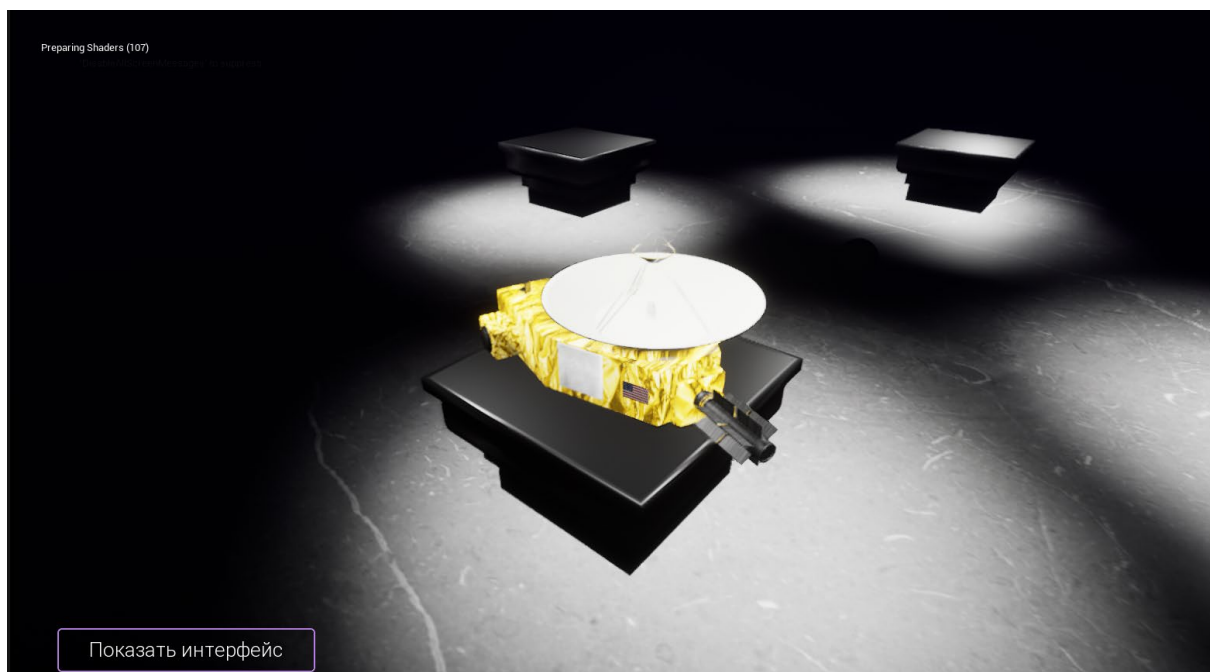


Рис. 3. Трехмерная модель космического исследовательского аппарата

Для взаимодействия пользователя с объектами была реализована система выбора моделей с помощью трассировки луча (Line Trace), позволяющая получать информацию об объекте при наведении или выборе.

С использованием Blueprint была реализована логика отображения информационных панелей с техническими характеристиками аппаратов (рис. 4).

Для повышения реалистичности сцены были использованы технологии динамического освещения Lumen и система виртуализированной геометрии Nanite.

Также была реализована система управления камерой, обеспечивающая возможность осмотра модели с различных ракурсов (рис. 5).

В ходе тестирования было установлено, что интерактивное пособие обеспечивает стабильную работу и позволяет эффективно демонстрировать особенности конструкции космических исследовательских аппаратов.

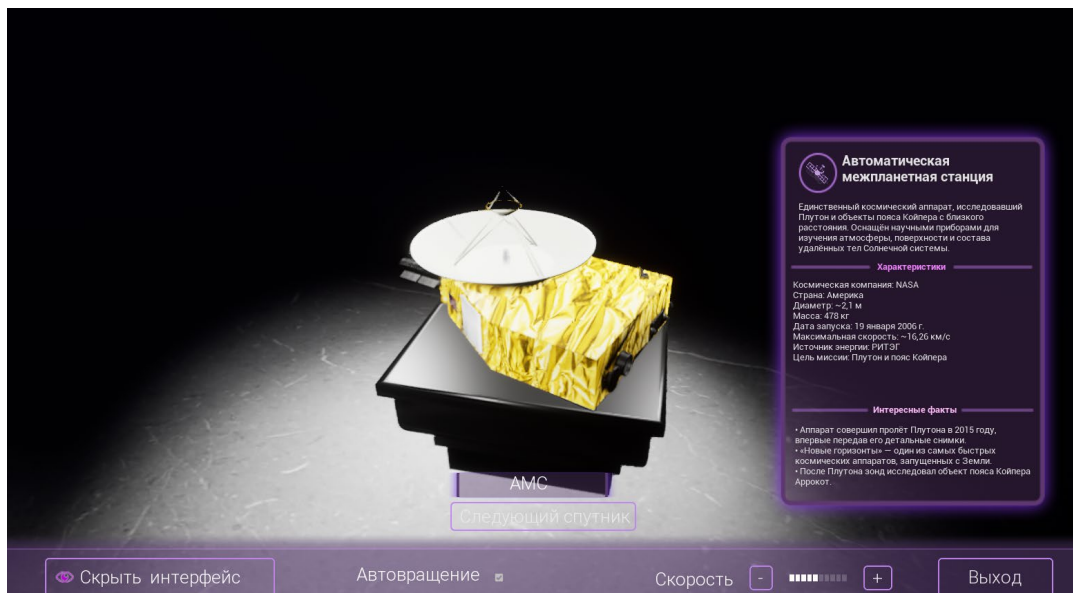


Рис. 4. Интерфейс отображения информации о космическом аппарате

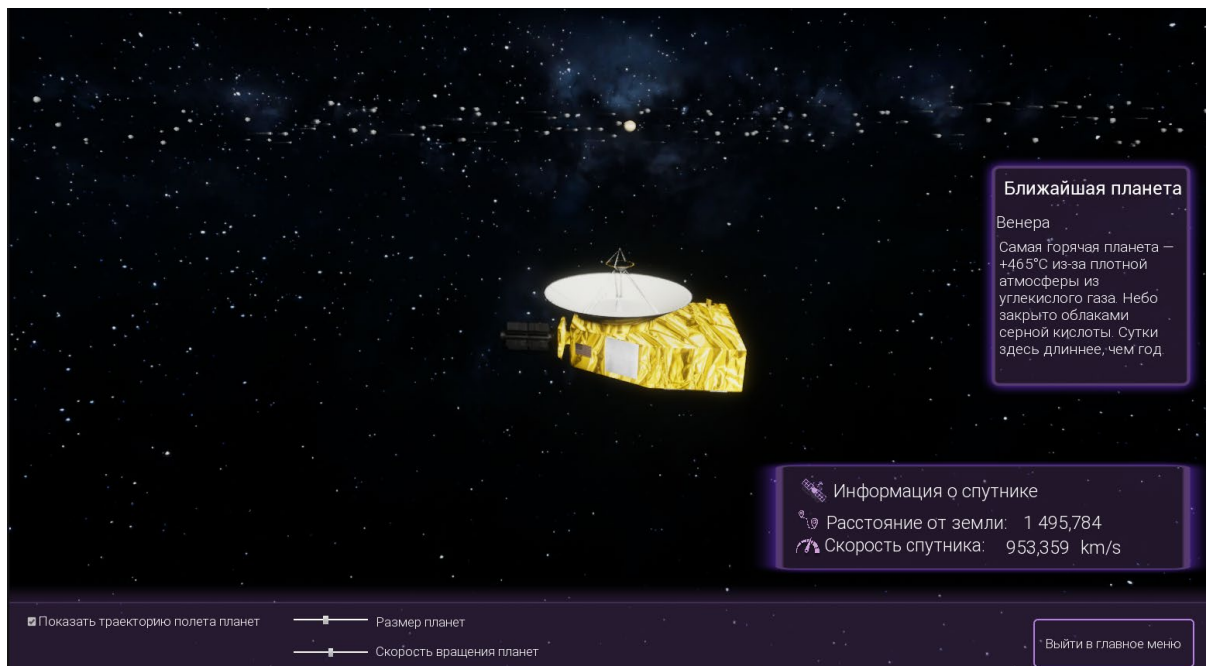


Рис. 5. Пример взаимодействия пользователя с трехмерной сценой

Заключение

В статье рассмотрен процесс разработки трехмерного интерактивного пособия по изучению космических исследовательских аппаратов в среде Unreal Engine 5.

В ходе работы были проанализированы существующие подходы к созданию образовательных интерактивных систем, определены требования к программному продукту и разработана архитектура приложения. С использованием Unreal

Engine 5 реализована виртуальная среда с трехмерными моделями космических аппаратов и интерактивным взаимодействием пользователя с объектами.

Разработанное пособие позволяет повысить наглядность образовательного процесса и способствует более эффективному изучению конструкции и характеристик космических исследовательских аппаратов.

В дальнейшем планируется расширение функциональности системы, включая добавление новых моделей космической техники, реализацию VR-поддержки и интеграцию дополнительных образовательных модулей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language / M. Fowler. – 3rd ed. – Boston : Addison-Wesley, 2004. – 208 p.
2. OMG Unified Modeling Language, Version 2.5.1. – Текст : электронный // Object Management Group. – 2017. – URL: <https://www.omg.org/spec/UML/2.5.1/About-UML/> (дата обращения: 14.05.2026).
3. Bell D. Explore the UML sequence diagram / D. Bell. – Текст : электронный // IBM Developer. – 2023. – URL: <https://developer.ibm.com/articles/the-sequence-diagram/> (дата обращения: 14.05.2026).
4. Unreal Engine Documentation. – Текст : электронный // Epic Games Developer Documentation. – URL: <https://dev.epicgames.com/documentation/unreal-engine/> (дата обращения: 14.05.2026).
5. Blender Manual. – Текст : электронный // Blender Documentation. – URL: <https://docs.blender.org/manual/en/latest/> (дата обращения: 14.05.2026).
6. Blueprints Visual Scripting in Unreal Engine. – Текст : электронный // Epic Games Developer Documentation. – URL: <https://dev.epicgames.com/documentation/unreal-engine/blueprints-visual-scripting-in-unreal-engine> (дата обращения: 14.05.2026).
7. Nanite Virtualized Geometry in Unreal Engine. – Текст : электронный // Epic Games Developer Documentation. – URL: <https://dev.epicgames.com/documentation/unreal-engine/nanite-virtualized-geometry-in-unreal-engine> (дата обращения: 14.05.2026).
8. Lumen Global Illumination and Reflections in Unreal Engine. – Текст : электронный // Epic Games Developer Documentation. – URL: <https://dev.epicgames.com/documentation/unreal-engine/lumen-global-illumination-and-reflections-in-unreal-engine> (дата обращения: 14.05.2026).

© В. Р. Ильясов, Д. Ю. Смирнов, 2026

К. Д. Кайгородов¹✉, Н. С. Головачев¹

Разработка VR-приложения виртуального музея всемирной истории «ArtHistory» с реализацией интерактивных пользовательских сценариев

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: MuseumArtHistory@gmail.com

Аннотация. В статье рассматривается процесс проектирования и программной реализации иммерсивной образовательной среды «ArtHistory» на базе игрового движка Unreal Engine 5. Актуальность исследования продиктована необходимостью поиска новых форм взаимодействия с культурным наследием в условиях цифровизации общества. Описан технологический стек разработки, включая применение систем Nanite и Lumen для достижения фотореалистичной визуализации, а также реализацию пространственных интерфейсов и механик взаимодействия в виртуальной реальности. Практическая значимость работы заключается в создании масштабируемого образовательного продукта, способного повысить вовлеченность учащихся через геймификацию и эффект полного погружения.

Ключевые слова: виртуальная реальность, Unreal Engine 5, виртуальный музей, иммерсивные технологии, Nanite, Lumen, образовательный процесс, цифровизация культуры

K. D. Kaigorodov¹✉, N. S. Golovachev¹

Development of the VR application of the virtual museum of world history «ArtHistory» with the implementation of Interactive user scenarios

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: MuseumArtHistory@gmail.com

Abstract. The article examines the design and software implementation of the "ArtHistory" immersive educational environment based on the Unreal Engine 5 game engine. The relevance of the study is dictated by the need to find new forms of interaction with cultural heritage in the context of society's digitalization. The development technology stack is described, including the use of systems Nanite and Lumen systems to achieve photorealistic visualization, as well as the implementation of spatial interfaces and interaction mechanics in virtual reality. The practical significance of the work lies in the creation of a scalable educational product that can increase student engagement through gamification and the effect of full immersion.

Keywords: virtual reality, Unreal Engine 5, virtual museum, immersive technologies, Nanite, Lumen, educational process, digitalization of culture

Введение

Технологии виртуальной реальности (VR) служат эффективным инструментом популяризации культурного наследия, нивелирующим пространственные ограничения традиционных музеев. Актуальность исследования обусловлена развитием функционала среды разработки Unreal Engine 5, потенциал которой

позволяет осуществлять фотореалистичную визуализацию и моделировать сложные интерактивные сценарии [1]. Цель работы заключается в проектировании и программной реализации приложения «ArtHistory», ориентированного на создание эффекта полного погружения в исторические эпохи. Разработка данного продукта отвечает современным тенденциям информатизации исторической науки и запросу общества на внедрение иммерсивных образовательных методик.

Исследование предметной области

Анализ современных тенденций в музейном деле свидетельствует о трансформации пассивного изучения экспонатов в концепцию «живого» пространства, предполагающую прямое интерактивное взаимодействие посетителя с историческим контекстом. Ведущим вектором развития выступает иммерсивность – интеграция пользователя в виртуальную среду, нивелирующая границу между реальным и виртуальным мирами. Дополнительным инструментом обучения служит геймификация: внедрение квестовых механик и сюжетных задач способствует повышению качества удержания информации.

С позиции когнитивной психологии, технологии VR активизируют пространственную память и эмоциональный интеллект, что детерминирует более эффективное усвоение материала по сравнению с традиционными методами [2]. Так, полномасштабная визуализация представителей разных эпох в секции «Доисторический мир» показывает выраженный эмоциональный отклик, стимулируя формирование заинтересованности пользователя [3].

Методы и технологии

В качестве среды разработки была выбрана платформа Unreal Engine 5 [4]. В отличие от альтернативных решений (в частности, Unity), данный инструмент содержит комплекс средств для достижения высокого качества графики без необходимости интеграции сторонних плагинов.

Nanite – система микрополигональной геометрии. Ее применение позволило импортировать высокодетализированные трехмерные модели музейных экспонатов с сохранением микротекстурных особенностей при стабильно высокой производительности рендеринга [5].

Lumen – технология динамического глобального освещения в реальном времени, обеспечивающая реалистичные отражения и мягкие тени [6].

Blueprints – система визуального программирования, на которой реализована логика взаимодействия с объектами и триггерные сцены [7].

MetaSounds – система для создания пространственного 3D-звука, привязанного к источникам в виртуальном пространстве.

Особое внимание уделено UI-дизайну. В приложении реализован пространственный интерфейс: элементы управления интегрированы непосредственно в игровое окружение в виде подсказок от гида. Для борьбы с укачиванием использован метод телепортации и система динамического виньетирования. Совместимость с различными VR-гарнитурами обеспечена поддержкой стандарта OpenXR.

Для тестирования проекта использовался шлем Pico 4, подключаемый к персональному компьютеру через приложение Pico Connect и среду SteamVR. При этом основной рендеринг виртуальной сцены выполняется на ПК средствами Unreal Engine 5, после чего сформированное изображение и аудиоданные передаются в шлем по сети Wi-Fi 5 или 6 либо через интерфейс USB-C. В обратном направлении передаются данные трекинга шлема и контроллеров, обеспечивающие интерактивное взаимодействие пользователя с виртуальной средой (рис. 1).

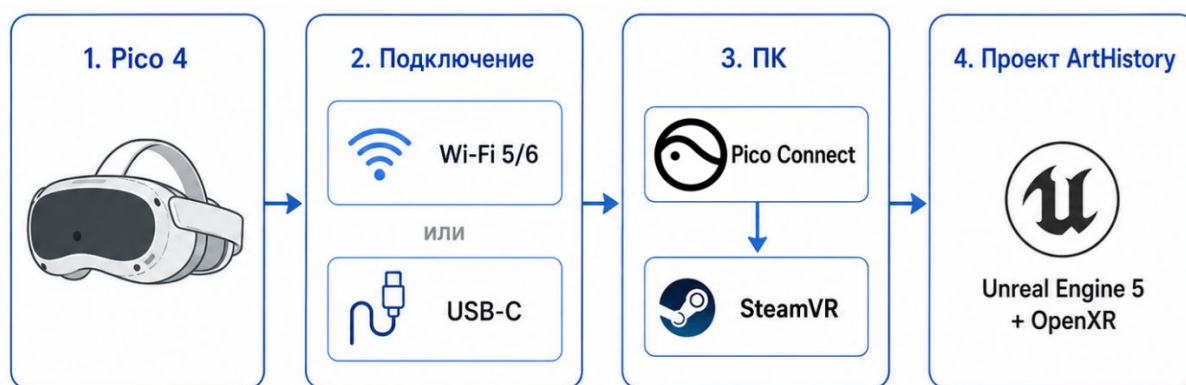


Рис. 1. Алгоритм подключения Pico 4 к ПК через Pico Connect и SteamVR

Результаты

Результатом разработки стала реализация VR-приложения, структурированного по трем тематическим зонам, а именно: доисторический мир, история, а также галерея искусств. Уникальным аспектом «Arthistory», в отличие от традиционных виртуальных галерей, является интеграция совокупности интерактивных механик, которые обеспечивают возможность пользователя находиться в естественной среде того или иного экспоната, в формате VR (рис. 2).

На этапе технического тестирования система подтвердила жизнеспособность, а также стабильное функционирование. В ходе оптимизации был достигнут баланс между высокой степенью детализации фотореалистичного контента и производительностью системы.



Рис. 2. Демонстрационный уровень времен строительства Санкт-Петербург

Заключение

Виртуальное пространство «ArtHistory» представляет собой вклад в развитие цифровизации культурного наследия, которое объединяет историческую достоверность с передовыми технологиями.

Практическая значимость проекта заключается в возможности его внедрения в качестве интерактивного учебного пособия в образовательных учреждениях.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Киллик, М. Гейм-дизайн: как создаются игры / М. Киллик; перевод с английского А. Н. Киселева. – Москва: ДМК Пресс, 2019. – 186 с.
2. Маккеффри, М. Unreal Engine VR для разработчиков / М. Маккеффри; перевод с английского Р. В. Терханова. – Москва: ДМК Пресс, 2019. – 266 с. – ISBN 978-5-97060-642-1. – URL: <https://dmkpress.com/catalog/computer/games/978-5-97060-642-1/>.
3. Quixel Megascans Library. – Текст: электронный // Quixel: [сайт]. – URL: <https://quixel.com/megascans/>.
4. Unreal Engine 5 Documentation. – Текст: электронный // Unreal Engine: [сайт]. – URL: <https://docs.unrealengine.com/5.0/en-US/>.
5. Nanite Virtualized Geometry. – Текст: электронный // Unreal Engine Documentation: [сайт]. – URL: <https://docs.unrealengine.com/5.0/en-US/nanite-virtualized-geometry-in-unreal-engine/>.
6. Lumen Global Illumination and Reflections. – Текст: электронный // Unreal Engine Documentation: [сайт]. – URL: <https://docs.unrealengine.com/5.0/en-US/lumen-global-illumination-and-reflections-in-unreal-engine/>.
7. Веселко, Н. И. Программирование в Unreal Engine 5 для начинающего игродела. Основы визуального языка Blueprint / Н. И. Веселко, О. В. Максименкова. – Москва: Эксмо, 2023. – 320 с.

© К. Д. Кайгородов, Н. С. Головачев, 2026

*Л. Д. Какабаева*¹✉

Метод централизованного управления сетевым оборудованием и средствами обеспечения сетевой безопасности

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: leyli.kakabayeva@bk.ru

Аннотация. В статье рассматривается задача централизованного управления сетевым оборудованием и средствами обеспечения сетевой безопасности в корпоративных вычислительных сетях. Актуальность исследования обусловлена ростом числа атак через эксплуатацию уязвимостей сетевого оборудования, требованиями нормативно-правовых актов Российской Федерации, а также возникновением большого количества ошибок при ручном децентрализованном конфигурировании. Цель исследования заключается в обеспечении аутентичности объектов сетевой инфраструктуры предприятия и подотчетности действий ее пользователей посредством централизованного управления. В работе предложен метод, основанный на криптографической аутентификации операций с использованием алгоритма Ed25519, управлении доступом на основе capability-токенов, программной аттестации устройств и обеспечении целостности журналов аудита на основе цепочки хеш-значений. Разработан четырехфазный протокол развертывания конфигураций, обеспечивающий атомарность изменений и возможность отката. Программная реализация системы (ZT-NMS) прошла комплексное тестирование, продемонстрировавшее достижение целевых показателей производительности и устойчивость к основным типам атак. Сравнительный анализ с существующими решениями подтвердил преимущества предложенного метода в части обеспечения механизмов безопасности.

Ключевые слова: централизованное управление, сетевое оборудование, средства защиты информации, capability-токены, аттестация устройств, журналы аудита

*L. D. Kakabaeva*¹✉

Method for centralized management of network equipment and network security tools

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: leyli.kakabayeva@bk.ru

Abstract. The article addresses the task of centralized management of network equipment and network security tools in corporate computing networks. The relevance of the research is due to the growing number of attacks exploiting vulnerabilities in network equipment, the requirements of regulatory acts of the Russian Federation, as well as a large number of errors in manual decentralized configuration. The aim of the study is to ensure the authenticity of network infrastructure assets of an enterprise and the accountability of user actions through centralized management. The paper proposes a method based on cryptographic authentication of operations using the Ed25519 algorithm, capability-based access control, software attestation of devices, and integrity protection of audit logs based on a hash chain. A four-phase configuration deployment protocol is designed to ensure atomicity of changes and rollback capability. The software implementation of the system (ZT-NMS) underwent comprehensive testing, demonstrating the achievement of target performance metrics and resilience to main types of attacks. A comparative analysis with existing solutions confirmed the advantages of the proposed method in terms of security mechanisms.

Keywords: centralized management, network equipment, network security tools, capability tokens, device attestation, audit logs

Введение

Информационная безопасность сетевой инфраструктуры предприятия определяется не только корректностью настройки отдельных средств защиты информации (СЗИ), но и согласованностью конфигураций всего парка сетевого оборудования. Согласно отчету Verizon Data Breach Investigations Report 2025, эксплуатация уязвимостей сетевого оборудования стала одним из основных векторов атак на корпоративные сети [1]. При этом значительная часть инцидентов связана не с уязвимостями нулевого дня, а с ошибками конфигурации, возникающими при ручном децентрализованном администрировании.

Актуальность темы исследования подтверждается требованиями нормативно-правовых актов Российской Федерации. Указ Президента РФ от 1 мая 2022 г. № 250 устанавливает дополнительные меры по обеспечению информационной безопасности, включая обязательный переход на отечественные СЗИ [2]. Постановление Правительства РФ от 12 февраля 2020 г. № 127 утверждает «Правила централизованного управления сетью связи общего пользования» [3]. Приказ ФСТЭК России от 25 декабря 2017 г. № 239 рекомендует применять централизованное управление подсистемой безопасности значимого объекта.

В научной среде задача централизованного управления сетевым оборудованием рассматривается в работах нескольких исследовательских групп. Архитектурная методология SDN с разделением плоскостей управления и данных через стандартизированный интерфейс OpenFlow развивается в работах N. McKeown, S. Shenker и J. Rexford [4]. Метод автоматического распределения и конфигурирования пакетных фильтров в сервисном графе предложен D. Brighenti, R. Sisto и F. Valenza [5]. Архитектурно-ориентированная методология проектирования SOAR-платформы рассматривается в работах С. Islam [6]. Методология построения типового центра интеллектуального управления сетевой безопасностью разработана Н. Г. Милославской [7]. Комплексная методология интеллектуально-адаптивного управления информационной инфраструктурой предложена Е. А. Басыней [8–10].

Несмотря на значительный объем проведенных исследований, в существующих работах не сформирован единый теоретико-методологический аппарат, охватывающий централизованное управление гетерогенным сетевым оборудованием и СЗИ в рамках одного жизненного цикла. Не разработана формальная модель взаимодействия контроллера с гетерогенными СЗИ, отсутствует теория абстракции функций безопасности для Northbound API. Анализ существующих коммерческих и открытых решений (NOC Project, rConfig, HPE IMC, Eltex.EMS, QNAP QRM+, TP-Link tpNMS, HOTA Купол) показал, что ни одно из них не реализует одновременно все ключевые механизмы обеспечения безопасности: криптографическую аутентификацию операций, использование токенов доступа, аттестацию устройств и контроль целостности журналов аудита [11].

Цель исследования заключается в обеспечении аутентичности объектов сетевой инфраструктуры предприятия и подотчетности действий ее пользователей посредством централизованного управления СЗИ и сетевым оборудованием. Для достижения цели поставлены задачи: исследование предметной области; моделирование объектов и процессов; разработка и программная реализация системы централизованного управления; тестирование и сравнительный анализ предложенного решения с существующими.

Методы и материалы

Предлагаемый метод централизованного управления сетевым оборудованием и средствами обеспечения сетевой безопасности базируется на четырех основных механизмах: криптографической аутентификации операций с использованием алгоритма Ed25519 [12], управлении доступом на основе capability-токенов, программной аттестации управляемых устройств и обеспечении целостности журналов аудита на основе цепочки хеш-значений.

Архитектура системы построена по принципу разделения на слои и реализована в виде набора слабосвязанных компонентов. В состав плоскости управления входят: сервис идентификации (Identity Service), отвечающий за управление учетными записями и аутентификацию субъектов; модуль выпуска capability-токенов (Capability Issuer), обеспечивающий генерацию и подписание токенов доступа; менеджер конфигураций (Config Manager); верификатор аттестации (Attestation Verifier); сервис аудита (Audit Service); прокси-компонент устройств (Device Proxy). Внешний доступ к системе осуществляется через API-шлюз с применением транспортного протокола TLS версии 1.3. Архитектура системы централизованного управления приведена на рис. 1.

Аутентификация операторов реализована по схеме challenge-response на основе эллиптической криптографии. Клиентское приложение генерирует ключевую пару Ed25519, после чего публичный ключ регистрируется в сервисе идентификации. При аутентификации сервер выдает случайное значение (challenge), которое подписывается приватным ключом оператора и проверяется сервером с использованием зарегистрированного публичного ключа. При успешной верификации формируется токен доступа.

Управление доступом реализовано на основе модели capability-based access control. Capability-токен содержит идентификатор субъекта, список грантов (ресурс, действия, ограничения), правила делегирования, период действия и подпись издателя. Перед выпуском токена движок политик (Policy Engine) оценивает запрос на основе атрибутов субъекта, ресурса и контекста. Защита от атак воспроизведения обеспечивается хранилищем одноразовых случайных значений (NonceStore).

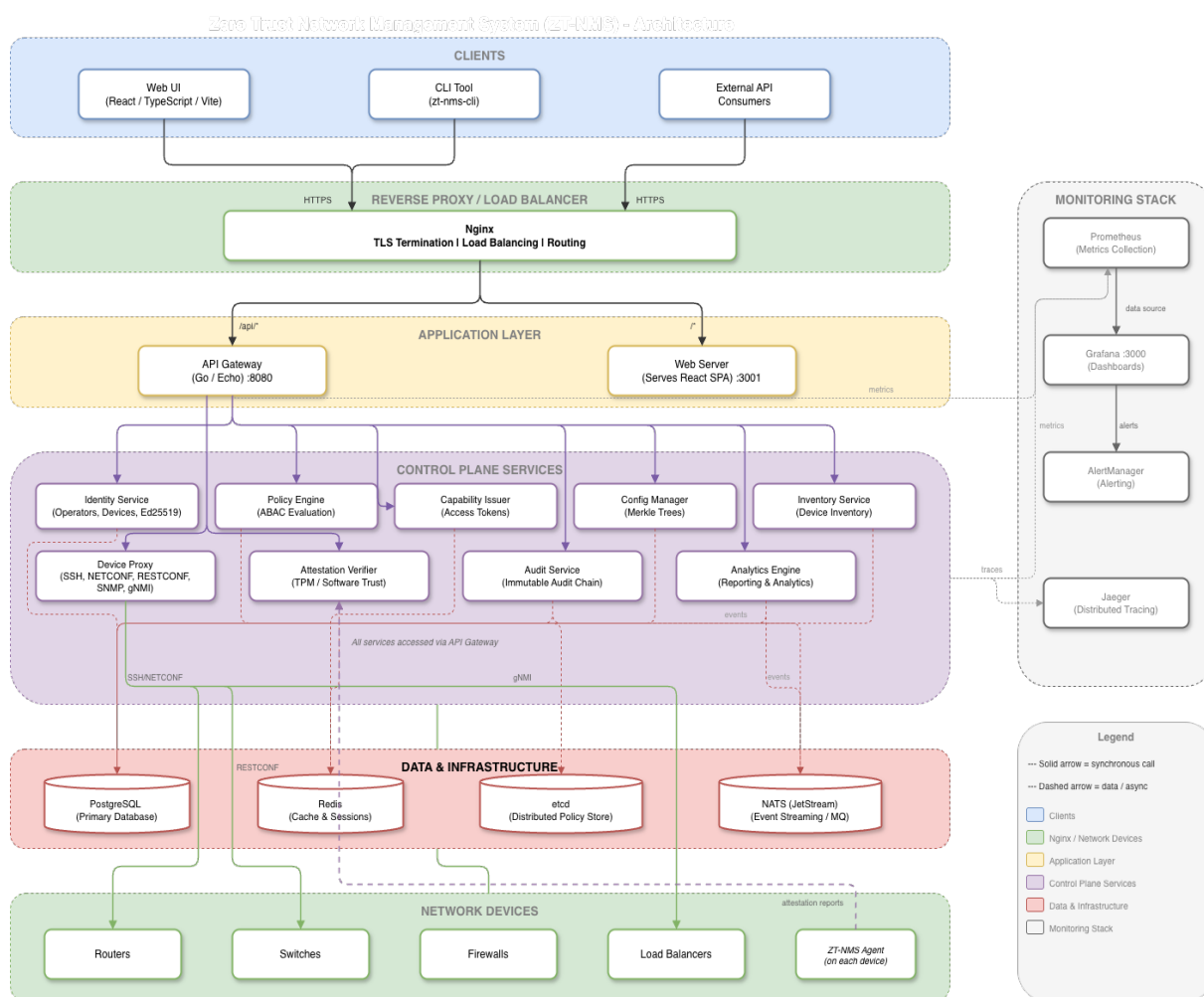


Рис. 1. Архитектура системы централизованного управления

Логика функционирования системы централизованного управления, реализующая описанные механизмы, представлена на рис. 2 в виде блок-схемы. Развертывание конфигураций на управляемые устройства осуществляется по разработанному четырехфазному протоколу, обеспечивающему атомарность изменений и возможность отката. Фаза *validating* включает проверку синтаксиса конфигурации, валидацию относительно действующих политик безопасности и проверку отсутствия потенциально опасных конструкций. На фазе *preparing* система формирует структуру *ConfigBlock*, содержащую предлагаемую конфигурацию, ее хеш-значение и хеш-значение предыдущего блока, образуя непрерывную цепочку. На фазе *committing* команда фиксации передается на устройство, где агент применяет подготовленную конфигурацию. Фаза *verifying* включает запрос текущей конфигурации с устройства, вычисление ее хеш-значения и сравнение с ожидаемым значением; при несовпадении инициируется автоматический откат.

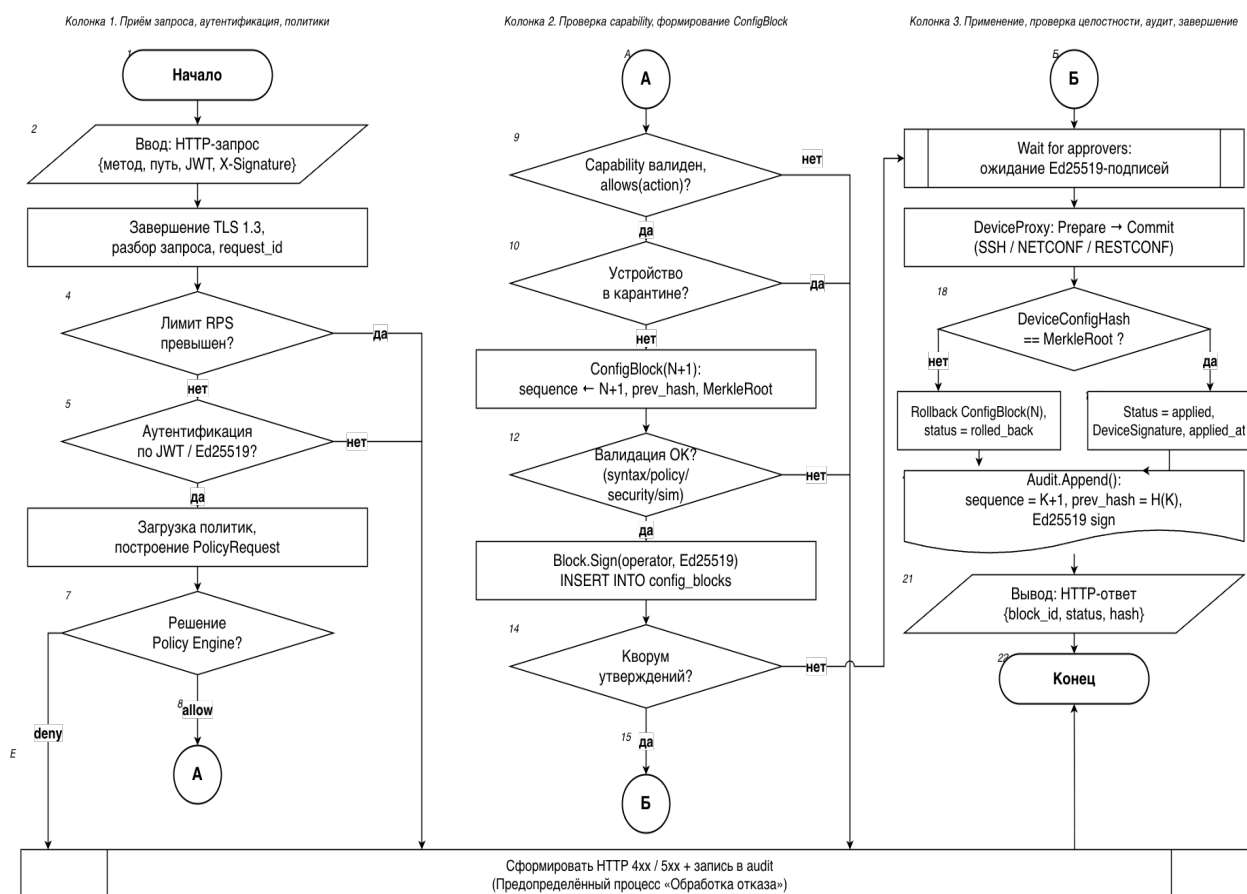


Рис. 2. Блок-схема работы системы централизованного управления

Программная аттестация управляемых устройств обеспечивает контроль целостности их программного обеспечения. Агент на устройстве собирает набор измерений (хеш-значения встроенного ПО, операционной системы, конфигурации и исполняемого файла агента), объединяет их с одноразовым значением, полученным от сервера, и подписывает результат своим приватным ключом. Сервис верификации сравнивает полученные измерения с эталонными значениями и обновляет статус доверия устройства.

Журналирование событий безопасности реализовано на основе цепочки хеш-значений. Каждое событие аудита содержит хеш-значение предыдущего события, обеспечивая возможность последующей верификации целостности журнала. Любая модификация или удаление событий приводит к нарушению цепочки и обнаруживается при верификации.

Результаты

На основе предложенного метода реализована программная система централизованного управления ZT-NMS. Для разработки выбран язык Go версии 1.22 в сочетании с фреймворком Echo v4, обеспечивающим высокую пропускную способность HTTP-обработчиков. В качестве хранилища используется PostgreSQL 16 с расширением pgvector, кеширование решений политик и хранение nonce

реализовано на Redis 7, шинная интеграция событий обеспечивается NATS JetStream. Контейнеризация и оркестрация выполняются на базе Kubernetes 1.29.

Для проверки корректности и производительности системы разработана технологическая дорожная карта тестирования, включающая шесть фаз: модульное, интеграционное, системное тестирование, тестирование безопасности, нагрузочное и приемочное тестирование. Проведено 348 модульных тестов с покрытием исходного кода 88 %, что превышает целевое значение в 80 %. Все 20 интеграционных тестов и 20 тестов безопасности успешно пройдены. Нагрузочное тестирование подтвердило достижение целевых показателей производительности: время верификации подписи Ed25519 – 0,09 мс при целевом значении менее 0,1 мс; время выпуска sarability-токена – 12 мс при целевом значении менее 50 мс; пропускная способность API-шлюза – 2340 запросов в секунду при целевом значении более 1000; время записи события в журнал аудита – 3 мс при целевом значении менее 10 мс.

Сравнение функциональных возможностей предложенной системы ZT-NMS с распространенными решениями (NOC Project, rConfig) показало, что разработанная система обладает наиболее полным набором механизмов безопасности: реализованы криптографическая аутентификация операций, использование токенов доступа, аттестация устройств и контроль целостности журналов, отсутствующие в обеих сравниваемых системах. По функциональным возможностям все три системы обеспечивают базовый набор функций управления гетерогенными системами, резервного копирования и развертывания конфигураций.

По показателю времени первичной аутентификации система ZT-NMS демонстрирует наилучший результат среди сравниваемых решений (150 мс против 185 мс у NOC Project и 320 мс у rConfig), что объясняется применением алгоритма Ed25519 и кешированием результатов проверки политик. Время применения изменений и отката увеличено относительно NOC Project, что обусловлено выполнением четырехфазного протокола развертывания и формированием криптографических подписей; данное увеличение является приемлемой платой за обеспечение целостности и подотчетности изменений.

Обсуждение

В отличие от существующих решений централизованного управления, ориентированных преимущественно на удобство администрирования, предложенный метод изначально проектировался с учетом требований обеспечения аутентичности и подотчетности. Криптографическая привязка операций к идентичности оператора через подпись Ed25519 исключает возможность отказа от авторства совершенного действия. Использование sarability-токенов с явным перечнем грантов обеспечивает выполнение принципа минимальных привилегий: токен может предоставлять доступ только к конкретному набору устройств и действий и имеет ограниченный срок действия [13].

Четырехфазный протокол развертывания конфигураций обеспечивает гарантии атомарности и возможность автоматического отката, что особенно важно при массовом обновлении гетерогенного парка сетевого оборудования. Цепочка

хеш-значений в журнале аудита обеспечивает обнаружение модификаций журнала, что исключает возможность скрытия следов атаки путем подделки записей [14].

Ограничениями предложенного метода являются увеличение времени отдельных операций по сравнению с системами, не использующими криптографические механизмы, и необходимость наличия агента на управляемых устройствах для выполнения процедуры аттестации. Для устройств без возможности установки агента предусмотрен режим работы с ограниченной функциональностью без проведения программной аттестации.

Заключение

В ходе исследования рассмотрена задача централизованного управления сетевым оборудованием и средствами обеспечения сетевой безопасности в корпоративных вычислительных сетях. Предложен метод централизованного управления, основанный на криптографической аутентификации операций, capability-токенах, программной аттестации устройств и обеспечении целостности журналов аудита.

Разработана и программно реализована система централизованного управления ZT-NMS. Проведено комплексное тестирование, подтвердившее корректность функционирования и достижение целевых показателей производительности. Сравнительный анализ с существующими решениями (NOC Project, rConfig) подтвердил преимущества предложенного метода в части обеспечения механизмов безопасности [15].

В дальнейшем планируется расширение поддержки сетевого оборудования и СЗИ различных вендоров, интеграция с системами анализа инцидентов и реагирования на угрозы, а также развитие математического аппарата унифицированного управления политиками для гетерогенных СЗИ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Verizon Business. 2025 Data Breach Investigations Report. – Verizon, 2025. – URL: <https://www.verizon.com/business/resources/reports/dbir/> (дата обращения: 04.05.2026).
2. Указ Президента РФ от 1 мая 2022 г. № 250 «О дополнительных мерах по обеспечению информационной безопасности Российской Федерации». – URL: <http://publication.pravo.gov.ru/Document/View/0001202205010023> (дата обращения: 27.12.2025).
3. Постановление Правительства РФ от 12 февраля 2020 г. № 127 «Об утверждении Правил централизованного управления сетью связи общего пользования». – URL: <https://base.garant.ru/73567870/> (дата обращения: 27.12.2025).
4. McKeown N., Anderson T., Balakrishnan H. et al. OpenFlow: Enabling Innovation in Campus Networks // ACM SIGCOMM Computer Communication Review. – 2008. – Т. 38, № 2. – С. 69–74.
5. Bringhenti D., Sisto R., Valenza F., Marchetto G. Automation for Network Security Configuration: State of the Art and Research Trends // ACM Computing Surveys. – 2025. – Т. 56, № 3.
6. Islam C., Babar M.A., Nepal S. Architecture-Centric Support for Integrating Security Tools in a Security Orchestration Platform // Proc. of European Conference on Software Architecture (ECSA 2020). – 2020.

7. Милославская Н.Г., Толстой А.И. Управление информационной безопасностью: учебное пособие. – М.: НИЯУ МИФИ, 2020. – 256 с.
8. Басыня Е.А. Сетевая информационная безопасность: учебник. – М.: НИЯУ МИФИ, 2023. – 224 с.
9. Басыня Е.А. Комплексная методология интеллектуально-адаптивного управления информационной инфраструктурой предприятия // Защита информации. Инсайд. – 2021. – № 5 (101). – С. 16–25.
10. Басыня Е.А., Худяков Д.С. Комплексный мониторинг информационной инфраструктуры предприятия // Защита информации. Инсайд. – 2020. – № 6 (96). – С. 28–33.
11. NOC Project: официальный сайт. – URL: <https://getnoc.com/> (дата обращения: 09.01.2026).
12. Bernstein D.J., Duif N., Lange T., Schwabe P., Yang B.-Y. High-speed high-security signatures // Journal of Cryptographic Engineering. – 2012. – Т. 2, № 2. – С. 77–89.
13. Зегжда Д.П., Калинин М.О., Полтавцева М.А. Методы защиты сетевой инфраструктуры в программно-определяемых сетях // Труды СПбГЭТУ «ЛЭТИ». – 2021.
14. Kreutz D., Ramos F.M.V., Verissimo P.E. et al. Software-Defined Networking: A Comprehensive Survey // Proceedings of the IEEE. – 2015. – Т. 103, № 1. – С. 14–76.
15. Siddiqui A., Rimal B.P. Unified Threat Management Framework for Home Networks // Dakota State University Scholar. – 2024.

© Л. Д. Какабаева, 2026

Е. А. Калинина^{1✉}, А. А. Шаранов¹

Разработка серверной части автоматизированной системы формирования цифрового портфолио обучающегося

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: kalininaevgeniaqq@gmail.com

Аннотация. В статье представлены основные результаты разработки серверной части автоматизированной системы формирования цифрового портфолио обучающегося для центра дополнительного образования. Проведён анализ предметной области и существующих решений, выявивший функциональный разрыв в части сквозного сопровождения конкурсной деятельности. Обоснован выбор технологического стека на основе Node.js, Express, PostgreSQL и JWT-аутентификации. Спроектирована модульная архитектура серверного приложения, реляционная база данных, включающая двенадцать таблиц с поддержкой мягкого удаления, ролевая модель из четырёх ролей и REST API из двенадцати групп маршрутов. Описана реализация ключевых модулей: аутентификации и разграничения прав, регистрации пользователей, подачи заявок на конкурсы, публикации результатов с выгрузкой в Excel и формированием ZIP-архивов дипломов. Проведено функциональное, нагрузочное тестирование и тестирование безопасности. Нагрузочные испытания подтвердили стабильную работу системы при одновременной работе ста пользователей при среднем времени отклика 2,44 мс и нулевом количестве ошибок. Разработанная система готова к внедрению в образовательных организациях.

Ключевые слова: цифровое портфолио, REST API, Node.js, PostgreSQL, аутентификация, дополнительное образование

Е. А. Kalinina^{1✉}, A. A. Sharapov¹

Development of the server part of an automated system for forming a student digital portfolio

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: kalininaevgeniaqq@gmail.com

Abstract. The paper presents the main results of developing the server-side of an automated system for forming a student's digital portfolio for a supplementary education center. An analysis of the subject area and existing solutions revealed a functional gap in end-to-end support of competition activities. The technology stack based on Node.js, Express, PostgreSQL and JWT authentication is justified. A modular server architecture, a relational database comprising twelve tables with soft-delete support, a four-role access model, and a REST API with twelve route groups were designed. The implementation of key modules is described: authentication and access control, user registration, competition application submission, results publication with Excel export and ZIP archive generation. Functional, load, and security testing were conducted. Load tests confirmed stable system operation with one hundred concurrent users, an average response time of 2.44 ms and zero errors. The developed system is ready for deployment in educational organizations.

Keywords: digital portfolio, REST API, Node.js, PostgreSQL, authentication, supplementary education

Введение

Учёт учебных и творческих достижений обучающихся на протяжении десятилетий осуществлялся преимущественно в бумажном формате: дипломы хранились в папках, итоги соревнований заносятся в журналы вручную. Цифровизация образования облегчила процесс хранения таких данных, однако одновременно привела к их рассредоточению по разрозненным инструментам – электронной почте, табличным редакторам, ситуативным платформам – вследствие чего единая картина достижений учащегося оказалась труднодостижимой [1].

В организациях дополнительного образования указанная проблема проявляется наиболее выражено. Проведение каждого конкурсного мероприятия сопряжено с отдельным сбором заявок, автономным ведением протоколов и ручным оформлением наград для участников. В результате педагог вынужден многократно вносить одни и те же сведения в разные системы, тогда как обучающийся лишён возможности получить консолидированный доступ к собственным достижениям [2, 3].

Настоящая работа посвящена созданию серверной части программного комплекса, способного объединить перечисленные процессы в единый контур: от подачи заявки до публикации итогов и пополнения портфолио. Для достижения цели решались задачи обследования предметной области, сравнения существующих аналогов, проектирования структуры хранения данных и программного интерфейса, а также реализации и верификации прикладных модулей.

Анализ предметной области и существующих решений

Несмотря на признанную важность портфолио как инструмента фиксации индивидуального прогресса, процесс его формирования в большинстве учреждений остаётся децентрализованным и слабо автоматизированным. Анализ рынка показал, что организаторы конкурсов в сфере дополнительного образования чаще всего прибегают к универсальным конструкторам форм: Google Формы и Яндекс Формы позволяют оперативно развернуть анкету и собрать данные в таблицу [4]. Вместе с тем подобные решения не поддерживают разграничение ролей участников процесса, а портфолио на их основе не формируется в принципе.

Государственный портал «Навигатор дополнительного образования» решает иную задачу – каталогизацию программ – и не рассчитан на управление конкурсным процессом [5]. Платформа «ЯКласс» ориентирована на проверочные задания академического типа и не предусматривает работы с творческими конкурсными материалами [6]. Площадки «Парад талантов России» и «Конкурсы.рф» обслуживают открытые всероссийские мероприятия и не предоставляют возможности развернуть собственный экземпляр системы [7, 8].

Анализ имеющихся решений показывает, что каждое из них, как правило, направлено на автоматизацию какого-либо участка работы, тогда как сквозное сопровождение процесса остаётся нереализованным. Приём заявок, публикация итогов и ведение портфолио зачастую обеспечивается разными инструментами, между которыми отсутствует прямая передача данных. Это вынуждает либо

переносить сведения вручную, либо мириться с их рассредоточенностью. Платформы, претендующие на более полный охват задач, в большинстве случаев ориентированы на широкий круг применений и не предусматривают гибкой адаптации под условия работы отдельной организации. Ни один из рассмотренных сервисов не перекрывает весь перечень необходимых функций одновременно, что подтверждает актуальность разработки специализированного решения.

Выбор технологического стека

При выборе технологического стека принималась во внимание специфика предполагаемой нагрузки: сервер должен параллельно принимать загружаемые пользователями файлы, выполнять запросы к базе данных и генерировать документы на выгрузку. Из доступных серверных платформ предпочтение было отдано Node.js, прежде всего благодаря её однопоточной событийно-ориентированной модели с неблокирующим вводом-выводом, позволяющей обрабатывать большое число параллельных соединений без пропорционального роста расхода памяти. В качестве надстройки применён фреймворк Express –минималистичная библиотека маршрутизации, предоставляющая гибкий механизм цепочек промежуточных обработчиков (middleware), в которые естественным образом встраиваются слои аутентификации и валидации запросов [9].

Реляционная СУБД PostgreSQL выбрана с учётом двух ключевых критериев: декларативного контроля целостности данных средствами самой СУБД и наличия системы в реестре отечественного программного обеспечения Минцифры России [10]. Для управления сессиями применён беззнаковый подход: сразу после входа сервер формирует компактный JSON Web Token, самостоятельно несущий в полезной нагрузке сведения о пользователе и подписанный секретным ключом, что исключает необходимость обращаться к хранилищу сессий при каждом запросе [11]. Хранение паролей организовано с использованием адаптивной функции bcrypt на основе алгоритма Blowfish: соль, встраиваемая в хеш при каждом вычислении, и настраиваемый коэффициент сложности делают атаки по предвычисленным таблицам практически нецелесообразными [12]. Вспомогательные задачи – загрузка файлов, экспорт таблиц, архивирование, почтовые уведомления – реализованы средствами библиотек Multer [13], ExcelJS [14], Archiver [15] и Nodemailer [16] соответственно. Единообразное использование JavaScript на всех уровнях приложения снижает когнитивную нагрузку при сопровождении кода и упрощает переход разработчика между клиентской и серверной частями.

Проектирование системы

Архитектурно сервер представляет собой монолит с чёткой внутренней модульностью: каждая функциональная область изолирована в отдельный набор файлов, тогда как точка входа server.js играет роль компоновщика и не содержит бизнес-логики. Взаимодействие с клиентом осуществляется через REST API [19], что делает интерфейс независимым от реализации фронтенда. Внутри сервер поделён

на слои маршрутизации, промежуточной обработки, прикладной логики и файловой подсистемы.

Разграничение доступа строится на четырёх ролях. Администратор контролирует жизненный цикл учётных записей и рассматривает заявки на регистрацию. Модератор ведёт конкурсную программу: открывает приём работ, утверждает призёров, публикует итоги и управляет выставкой. Учитель прикреплен к конкретным ученикам и подаёт заявки исключительно от их имени. Ученик просматривает доступные конкурсы, самостоятельно подаёт работы и знакомится с накопленными наградами.

Схема базы данных включает двенадцать таблиц, распределённых по функциональным блокам: управление учётными записями (Users, Profiles, Registration_requests, Roles, Teacher_student), конкурсный контур (Competitions, Themes, Competition_applications, Application_attachments), блок итогов (Results, Degrees) и витрина выставки (Exhibition). Вместо физического удаления строк применено мягкое удаление: поле `deleted_at` фиксирует момент деактивации и позволяет сохранять историческую запись без нарушения ссылочной целостности. Ролевая модель разграничения доступа типа RBAC охватывает четыре роли; каждый маршрут API защищён двухэтапным конвейером промежуточных обработчиков – верификацией токена с последующей проверкой допустимой роли [17, 18].

Практическая реализация

Кодовая база организована по принципу разделения ответственности: директория `routes` хранит объявления маршрутов, `middleware` – повторно используемые обработчики, `utils` – вспомогательные утилиты, `uploads` – загружаемые файлы, разбитые на три поддиректории по типу контента. Подключение к PostgreSQL реализовано через пул соединений библиотеки `pg`: входящий запрос получает свободное соединение из пула и возвращает его после завершения транзакции, что устраняет накладные расходы на повторное установление TCP-соединения при каждом вызове базы данных.

При входе в систему сервер формирует JWT, вкладывая в полезную нагрузку идентификатор пользователя, идентификатор профиля, назначенную роль и логин; токен подписывается ключом из переменных окружения. При каждом запросе `middleware auth.js` извлекает токен из заголовка `Authorization` и верифицирует подпись; `middleware roleCheck.js` сопоставляет роль из токена с множеством ролей, допущенных к данному маршруту, и при несоответствии прерывает обработку с кодом 403.

Логика регистрации учитывает три ситуации: если в базе уже есть деактивированная запись с тем же номером телефона, аккаунт восстанавливается без создания дубликата; если записи нет, создаются новые строки во всех связанных таблицах; если активная запись существует, операция отклоняется с поясняющим сообщением. Экспортный модуль формирует многостраничную Excel-книгу; изображения конкурсных работ вставляются непосредственно в ячейки: сервер считывает заголовки JPEG и PNG для определения исходных размеров,

масштабирует их до 450 пикселей по большей стороне и освобождает временные файлы сразу после отправки архива клиенту.

Тестирование

Корректность реализованных сценариев проверялась вручную через Postman: для каждого маршрута формировались наборы корректных и заведомо ошибочных запросов, охватывающие граничные значения входных данных, нестандартные форматы файлов и попытки вызова маршрута с недопустимой ролью. Отдельно верифицировалась генерация Excel-выгрузок и ZIP-архивов дипломов. Нагрузочное испытание выполнено инструментом k6 [20]: виртуальные пользователи циклически выполняли четыре операции – запрос каталога конкурсов, вход в систему, чтение профиля и просмотр выставки. Измеренные показатели сведены в табл. 1.

Таблица 1. Результат нагрузочного тестирования

Показатель	Значение
Общее число запросов	15 644
Максимальное количество одновременных пользователей	100
Среднее время ответа	2,44 мс
Медианное время ответа	1,90 мс
95-й перцентиль времени ответа	3,31 мс
Количество ошибок	0
Процент успешных запросов	100 %

Нагрузочный сценарий со ста одновременными виртуальными пользователями завершился без единой ошибки: среднее время отклика составило 2,44 мс, медиана – 1,90 мс, 95-й перцентиль не превысил 3,31 мс. Тестирование защищённости охватывало три вектора. Инъекции SQL: в текстовые поля подставлялись выражения «UNION SELECT» и операторы «DROP»; параметризованные запросы к PostgreSQL полностью исключили интерпретацию введённых данных как кода. Несанкционированный доступ: запросы к защищённым маршрутам без токена возвращали статус 401, запросы с токеном неподходящей роли – статус 403. Перебор учётных данных: после пяти неверных попыток входа с одного IP-адреса система вводит временную блокировку длительностью пятнадцать минут.

Заключение

В результате выполненной работы спроектирован и реализован серверный компонент автоматизированной системы, обеспечивающей сквозное сопровождение конкурсной деятельности в центре дополнительного образования. Архитектура системы охватывает реляционную базу данных из двенадцати таблиц, ролевую модель управления доступом и REST API из двенадцати групп

маршрутов. Нагрузочные испытания при одновременной работе ста пользователей зафиксировали среднее время отклика 2,44 мс при полном отсутствии отказов; тестирование безопасности не выявило критических уязвимостей. Программный продукт готов к опытной эксплуатации в образовательных организациях и может быть адаптирован для смежных предметных областей, где требуется централизованный учёт достижений участников.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Архипенко, А. В. Автоматизированное электронное портфолио как метод оценки учебно-профессиональных достижений учащегося / А. В. Архипенко // Гуманизация образования. – 2024. – №2. URL: <https://cyberleninka.ru/article/n/avtomatizirovannoe-elektronnoe-portfolio-kak-metod-otsenki-uchebno-professionalnyh-dostizheniy-uchaschegosya-i-sredstvo-rovysheniya> (дата обращения: 07.05.2026). – Текст: электронный.
2. Пираков, Ф. Д. Разработка и применение системы повышения портфолио обучающегося в вузе / Ф. Д. Пираков, А. П. Клишин, Н. Л. Еремина, Е. Н. Клыжко // Вестник НГУ. Серия: Информационные технологии. – 2019. – № 4. – С. 87–100.
3. Курмангалиева, А. М. Разработка структуры информационной системы формирования портфолио обучающегося / А. М. Курмангалиева, М. М. Жаманкарин, М. К. Айткенова // Наука и реальность. – 2021. – № 2(6). – С. 228–231.
4. Google Forms : официальный сайт. – URL: <https://workspace.google.com/intl/ru/products/forms/> (дата обращения: 07.05.2026). – Текст: электронный.
5. Навигатор дополнительного образования детей : федеральный портал. – URL: <https://navigator.edu54.ru/> (дата обращения: 07.05.2026). – Текст: электронный.
6. ЯКласс : официальный сайт. – URL: <https://www.yaklass.ru/> (дата обращения: 07.05.2026). – Текст: электронный.
7. Парад талантов России : информационно-образовательный портал. – URL: <https://paradtalant.ru/> (дата обращения: 07.05.2026). – Текст: электронный.
8. Конкурсы.рф : официальный сайт. – URL: <https://конкурсы.рф> (дата обращения: 07.05.2026). – Текст: электронный.
9. Браун, И. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript / И. Браун. – Санкт-Петербург : БХВ-Петербург, 2021. – ISBN 978-5-4461-0590-8.
10. Дейт, К. Дж. Введение в системы баз данных / К. Дж. Дейт. – Москва : Вильямс, 2005. – ISBN 978-5-8459-0788-4.
11. Jones, M. JSON Web Token (JWT) / M. Jones, J. Bradley, N. Sakimura // IETF. – 2015. – URL: <https://www.rfc-editor.org/info/rfc7519> (дата обращения: 09.05.2026). – Текст: электронный.
12. Шнейер, Б. На пути к Skein: просто и понятно про Blowfish / Б. Шнейер // Habr. – 2012. – URL: <https://habr.com/ru/articles/140394/> (дата обращения: 09.05.2026). – Текст: электронный.
13. Multer npm package documentation. – URL: <https://www.npmjs.com/package/multer> (дата обращения: 09.05.2026). – Текст: электронный.
14. ExcelJS documentation. – URL: <https://github.com/exceljs/exceljs> (дата обращения: 09.05.2026). – Текст: электронный.
15. Archiver documentation. – URL: <https://www.archiverjs.com/> (дата обращения: 09.05.2026). – Текст: электронный.
16. Nodemailer documentation. – URL: <https://nodemailer.com/> (дата обращения: 09.05.2026). – Текст: электронный.
17. Хрущёв, Д. Г. Разработка ролевой модели доступа к объектам и сервисам веб-приложения на примере системы коллективного наполнения контентом / Д. Г. Хрущёв // Молодой

учёный. – 2016. – URL: <https://cyberleninka.ru/article/n/razrabotka-rolevoy-modeli-dostupa-k-obektam-i-servisam-veb-prilozheniya-na-primere-sistemy-kollektivnogo-napolneniya-kontentom> (дата обращения: 13.05.2026). – Текст: электронный.

18. Ляшов, Е. И. Виды аутентификации в современных веб-приложениях / Е. И. Ляшов // *Universum: технические науки*. – 2024. – № 12(129). – URL: <https://cyberleninka.ru/article/n/vidy-autentifikatsii-v-sovremennyh-veb-prilozheniyah> (дата обращения: 13.05.2026). – Текст: электронный.

19. Введение в REST API — RESTful веб-сервисы // *Habr.com*. – URL: <https://habr.com/ru/articles/483202/> (дата обращения: 09.05.2026). – Текст: электронный.

20. Grafana K6 documentation. – URL: <https://grafana.com/docs/k6/latest/> (дата обращения: 13.05.2026). – Текст: электронный.

© *Е. А. Калинина, А. А. Шарапов, 2026*

М. С. Карапетьянц¹

Метод оценки доступности информационных ресурсов в оверлейной сети на базе луковой маршрутизации

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: mkarapetyants@mephi.ru

Аннотация. Оверлейные сети на базе луковой маршрутизации формируют распределенную инфраструктуру сетевых сервисов, не индексируемых поисковыми системами глобальной сети, мониторинг которой представляет практический интерес для специалистов в области информационной безопасности. Существующие инструменты применяют разрозненные методы проверки без формального разграничения уровней взаимодействия, что существенно снижает точность мониторинга и воспроизводимость результатов. Целью настоящей работы является систематизация и сравнительная оценка методов определения доступности информационных ресурсов оверлейной инфраструктуры для повышения эффективности инструментов разведки на основе открытых источников. В работе предложена пятиуровневая иерархическая модель, формализующая уровни проверки от публикации дескриптора в распределенной хэш-таблице до семантической верификации контента, и проведена экспериментальная оценка трех методов на выборке из 1000 узлов оверлейной инфраструктуры третьей версии протокола. Результаты показали, что методы существенно различаются по профилю ошибок и области применимости. Метод HTTP-зондирования обеспечил наилучшее общее качество классификации: точность 89,56%, F1-мера 0,910 при умеренной средней задержке 2213 мс и незначительной доле неопределенных ответов (1,3%). Метод проверки публикации дескриптора продемонстрировал почти идеальную полноту ($\text{recall}=0,989$), однако низкую прецизионность (0,689) вследствие сохранения устаревших дескрипторов в распределенной хеш-таблице (DHT, англ. Distributed Hash Table) после перехода ресурса в офлайн, что ограничивает его применимость задачами предварительной фильтрации. Метод проверки TCP-соединения показал наихудшие результаты (точность 58,80%, $F1=0,560$) в силу систематической ошибки, обусловленной нестандартным распределением прослушиваемых портов целевых ресурсов. Практическая значимость работы состоит в обоснованной иерархической стратегии комбинированного применения методов, позволяющей совместить скорость метода проверки публикации дескриптора с точностью HTTP-зондирования при минимизации числа запросов к целевой инфраструктуре. Направления дальнейших исследований включают разработку и верификацию метода проверки доступности точек введения, а также анализ временной динамики доступности ресурсов при многократном тестировании.

Ключевые слова: оверлейные сети, луковая маршрутизация, доступность ресурсов, сетевой мониторинг, инструменты OSINT

M. S. Karapetyants¹

A method for assessing the availability of information resources in an onion routing overlay network

¹National Research Nuclear University "MEPhI", Moscow, Russian Federation
e-mail: mkarapetyants@mephi.ru

Abstract. Onion-routing-based overlay networks form a distributed infrastructure of network services that are not indexed by search engines on the global Internet, and monitoring this infrastructure

is of practical interest to information security specialists. Existing tools rely on fragmented verification methods without a formal separation of interaction levels, which significantly reduces monitoring accuracy and the reproducibility of results. The aim of this study is to systematize and comparatively evaluate methods for determining the availability of information resources within overlay infrastructures in order to improve the effectiveness of open-source intelligence tools. The paper proposes a five-level hierarchical model formalizing verification stages ranging from descriptor publication in a distributed hash table to semantic content verification, and presents an experimental evaluation of three methods on a sample of 1,000 nodes of a version 3 overlay infrastructure protocol. The results show that the methods differ substantially in both error profile and scope of applicability. The HTTP probing method achieved the best overall classification performance, with an accuracy of 89.56%, an F1 score of 0.910, a moderate average latency of 2,213 ms, and a negligible proportion of indeterminate responses (1.3%). The descriptor publication verification method demonstrated near-perfect recall (0.989), but low precision (0.689) due to the persistence of stale descriptors in the distributed hash table (DHT, Distributed Hash Table) after a resource transitions offline, which limits its applicability to preliminary filtering tasks. The TCP connectivity verification method showed the weakest performance (accuracy 58.80%, $F1 = 0.560$) because of a systematic error caused by the non-standard distribution of listening ports across the target resources. The practical significance of the study lies in a justified hierarchical strategy for combining methods, making it possible to couple the speed of descriptor publication verification with the accuracy of HTTP probing while minimizing the number of requests sent to the target infrastructure. Future research directions include the development and validation of a method for checking the availability of introduction points, as well as the analysis of temporal availability dynamics through repeated testing.

Keywords: overlay networks, onion routing, resource availability, network monitoring, OSINT tools

Введение

Оверлейные сети на базе луковой маршрутизации реализуют децентрализованную модель сетевых сервисов, при которой адресация узлов инфраструктуры осуществляется через криптографически верифицированные псевдослучайные идентификаторы, а не через публичные IP-адреса [1]. Данная архитектура обеспечивает анонимность как для операторов сервисов, так и для их пользователей, что обуславливает широкий диапазон сценариев применения – от легитимных коммуникационных платформ до инфраструктуры противоправной деятельности [2–4].

Задача оценки доступности информационных ресурсов в подобных сетях принципиально отличается от аналогичной задачи в открытом сегменте интернета. Стандартные инструменты сетевого мониторинга такие как ICMP-зондирование, DNS-резолуция, публичное IP-сканирование неприменимы в силу архитектурных особенностей луковой маршрутизации [5]. Вместе с тем практическая потребность в систематическом мониторинге узлов оверлейной инфраструктуры велика. По данным измерительных исследований Owen и Savage [6], от 30% до 50% известных ресурсов недоступны в любой момент времени, а в работе Sanatinia, Park и Blass [7] показано, что медианное время существования ресурса составляет порядка 12 суток. При этом Boshmaf et al. [8] зафиксировали, что 47,8% узлов оверлейной инфраструктуры находятся в офлайн в любой произвольный момент наблюдения.

Анализ существующих работ выявляет принципиальный методологический пробел. Инструменты мониторинга OnionScan, Onionprobe, TorBot и другие – применяют разрозненные методы проверки доступности без формального разграничения проверяемых уровней стека взаимодействия. Loesing et al. [9] первыми декомпозировали задержки протокола по отдельным шагам, однако не формализовали понятие уровня доступности. Это приводит к несопоставимости результатов между исследованиями и к систематическим ошибкам классификации, механизм которых остается необъясненным при отсутствии единой иерархической модели.

Целью настоящей работы является систематизация и сравнительная оценка методов определения доступности информационных ресурсов оверлейной инфраструктуры для повышения эффективности инструментов разведки на основе открытых источников. Для достижения поставленной цели решались следующие задачи: формализация пятиуровневой иерархической модели оценки доступности; разработка и реализация трех независимых методов, соответствующих отдельным уровням модели; проведение сравнительного эксперимента на репрезентативной выборке; количественная оценка сравнительной эффективности методов.

Теоретическая значимость работы заключается в формализации иерархической модели уровней доступности применительно к оверлейным сетям - данная классификация в явном виде отсутствует в существующей литературе, несмотря на то что отдельные ее элементы имплицитно используются в ряде инструментальных реализаций. Практическая значимость определяется возможностью непосредственного применения предложенной иерархической стратегии в системах автоматизированного мониторинга и OSINT-инструментах, ориентированных на работу с ресурсами оверлейной инфраструктуры.

Методы и материалы

Предложенная иерархическая модель оценки доступности включает пять уровней, каждый из которых соответствует более глубокому слою протокольного стека взаимодействия с ресурсом.

Первый уровень. Публикация дескриптора проверяет факт наличия актуального дескриптора сервиса в распределенной хэш-таблице (DHT) сети. Наличие дескриптора свидетельствует о том, что оператор сервиса активно анонсирует свой ресурс в инфраструктуре. Ключевое ограничение данного уровня состоит в том, что дескриптор сохраняет валидность в течение нескольких часов после остановки сервиса, что порождает ложноположительные срабатывания [10–12].

Второй уровень. Доступность точек введения извлекает из дескриптора список relay-узлов, выполняющих функцию точек введения, и проверяет их актуальный статус в консенсусе сети. Дескриптор может существовать, однако все точки введения при этом могут быть уже исключены из активной инфраструктуры, что делает сервис фактически недостижимым.

Третий уровень. Транспортная доступность – попытка установить TCP-соединение с ресурсом через SOCKS5-прокси на указанных портах. Данный уровень

проверяет достижимость прикладного процесса, но принципиально зависит от корректного указания прослушиваемых портов.

Четвертый уровень. Прикладное зондирование направляет полноценный протокольный запрос (HTTP/HTTPS) и оценивает валидность полученного ответа. Это наиболее информативный и практически значимый уровень для задач мониторинга.

Пятый уровень. Семантическая верификация анализирует содержимое ответа на соответствие ожидаемому контенту. Данный уровень выходит за рамки задачи мониторинга доступности в узком смысле и относится к задачам краулинга и идентификации ресурсов.

Для экспериментальной проверки сформирована выборка из 1000 узлов оверлейной инфраструктуры третьей версии протокола. Выборка включает три категории: 121 эталонный ресурс с постоянно верифицированным онлайн-статусом, 535 ресурсов, обнаруженных посредством автоматизированного краулинга и находившихся в онлайн на момент сбора, и 344 ресурса с верифицированным офлайн-статусом. Фактическое соотношение статусов на момент прогона составило приблизительно 533 онлайн и 465 офлайн в силу естественной нестабильности ресурсов между периодами краулинга и тестирования – явление, хорошо известное из литературы [13].

Для оценки качества применялись стандартные метрики бинарной классификации: точность (accuracy), прецизионность (precision), полнота (recall) и F1-мера. Дополнительно фиксировались средняя, медианная задержка и 95-й перцентиль времени проверки, а также доля результатов с неопределенным статусом.

Результаты

В рамках экспериментального исследования проведена сравнительная оценка трех методов определения доступности информационных ресурсов оверлейной инфраструктуры. Результаты фиксировались с временными метками и детализированными кодами завершения, что обеспечило возможность сравнения как метрик качества классификации, так и показателей производительности. Сводные результаты представлены в табл. 1 и 2.

Метод HTTP-зондирования демонстрирует наилучшее соотношение показателей качества: точность 89,56%, F1-мера 0,910 при сравнительно умеренных задержках (среднее 2213 мс, медиана 1966 мс). Незначительное число ложноотрицательных срабатываний обусловлено таймаутами у части онлайн-ресурсов, а 95 ложноположительных результатов связаны преимущественно с ресурсами, доступными на уровне HTTP-ответа, но отдающими коды ошибок.

Таблица 1. Оценка методов на основе метрик качества классификации

Метод	Acc, %	Prec	Rec	F1	TP	TN	FP	FN	Unk, %	Ср. t, мс
Метод публикации дескриптора	75,85	0,689	1,000	0,816	533	224	241	0	0,20	1024
Метод ТСП-соединения	58,80	0,653	0,490	0,560	262	326	139	273	0,00	4286
Метод НТТР-зондирования	89,56	0,846	0,985	0,910	523	361	95	8	1,30	2213

Таблица 2. Оценка методов на основе метрик производительности и покрытия

Метод	count_total	count_known	median_t, мс	p95_t, мс	Интерпретация
Публикации дескриптора	1000	998	935	2486	Стабильный, быстрый
ТСП-соединения	1000	1000	3701	10012	Нет unknown, медленный
НТТР-зондирования	1000	987	1966	3918	Незначит. unknown, средний

Метод проверки публикации дескриптора обнаруживает парадоксальную картину: при почти идеальной полноте он демонстрирует низкую прецизионность. Это означает, что метод не пропускает ни одного онлайн-ресурса, однако признает доступными 241 ресурс, реально находящийся в офлайн. Механизм данного эффекта объясняется архитектурой DHT: дескриптор продолжает существовать в распределенном хранилище в течение нескольких часов после остановки сервиса, формируя устаревшие записи. Высокая скорость метода делает его привлекательным для предварительной фильтрации, но недостаточным для финальной верификации.

Метод проверки ТСП-соединения показал наихудшие результаты по всем ключевым метрикам: точность 58,80%, F1-мера 0,560, 273 ложноположительных срабатывания. Анализ деталей показывает, что значительная часть онлайн-ресурсов прослушивает нестандартные порты, что влечет систематическую ошибку при использовании портов по умолчанию. Дополнительно метод является самым медленным: среднее время 4286 мс, 95-й перцентиль 10012 мс – это объясняется накоплением задержек при ожидании таймаутов соединения.

Обсуждение

Установленный профиль ошибок метода проверки публикации дескриптора – высокая полнота при низкой прецизионности – находит прямое объяснение в архитектурных свойствах DHT. Buitrago et al. [10] показали, что дескриптор сохраняет валидность в хранилище в течение нескольких часов после остановки ресурса, а

доля запросов к DHT, приходящихся на несуществующие записи, достигает 80%. Зафиксированные $FP=241$ при $FN=0$ полностью согласуются с данной закономерностью, подтверждая, что метод первого уровня не способен различить активный ресурс и устаревший дескриптор, что ограничивает его применимость задачами предварительной фильтрации.

Результаты метода НТТР-зондирования соответствуют показателям, зафиксированным Loesing et al. [9] при тестировании аналогичных зондов - 85–92% в зависимости от конфигурации таймаутов. Незначительное число ложноотрицательных срабатываний объясняется таймаутами у перегруженных узлов, что Owen и Savage [6] также отмечают, как характерную особенность высоконагруженных ресурсов оверлейной инфраструктуры.

Низкие показатели метода ТСР-соединения согласуются с наблюдениями Buitrago et al. [10]. Значительная доля ресурсов прослушивает нестандартные порты, вследствие чего методы, ориентированные на порты 80 и 443, систематически занижают оценку доступности. Данная проблема устранима введением предварительного портового зондирования с расширенным набором типичных портов.

Зафиксированное изменение статуса ~18% ресурсов за период менее 24 часов согласуется с данными Owenson et al. [11] и указывает на необходимость многократного тестирования с временным интервалом как самостоятельного исследовательского направления.

На основании полученных результатов предлагается иерархическая стратегия комбинированного применения методов: метод публикации дескриптора используется как быстрый предварительный фильтр, а метод НТТР-зондирования как финальная верификация для ресурсов, прошедших первый шаг, и как резервный при статусе «неопределено» [14, 15]. Аналогичный каскадный подход предложен в работе Bergman [16], однако без количественного обоснования на реальной выборке настоящее исследование восполняет данный пробел.

Заключение

В настоящей работе предложена пятиуровневая иерархическая модель оценки доступности информационных ресурсов оверлейных сетей на базе луковой маршрутизации и проведена экспериментальная оценка трех методов, соответствующих первому, третьему и четвертому уровням модели, на выборке из 1000 узлов инфраструктуры третьей версии протокола.

Установлено, что методы принципиально различаются по профилю ошибок и области применимости: метод НТТР-зондирования обеспечивает наилучшее качество классификации и рекомендуется для финальной верификации; метод проверки публикации дескриптора применим исключительно как предварительный фильтр вследствие систематических ложноположительных срабатываний; метод ТСР-соединения требует доработки в части портового зондирования. На основании результатов обоснована иерархическая стратегия комбинированного применения методов, обеспечивающая баланс точности и скорости при минимизации нагрузки на целевую инфраструктуру.

Направления дальнейших исследований включают реализацию метода второго уровня иерархии - проверки доступности точек введения, - введение механизма портового зондирования для транспортного уровня, а также анализ временной динамики доступности ресурсов при многократном тестировании.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Appelbaum, J., & Muffett, A. (2015). RFC 7686: The “.onion” Special-Use Domain Name. Internet Engineering Task Force. URL:<https://datatracker.ietf.org/doc/html/rfc7686>
2. Pastor-Galindo J. et al. The not yet exploited goldmine of OSINT: Opportunities, open challenges and future trends //IEEE access. – 2020. – Т. 8. – С. 10282-10304.
3. Barredo-Valenzuela A. R., Pastrana S., Suarez-Tangil G. Snorkeling in dark waters: A longitudinal surface exploration of unique Tor Hidden Services //IEEE Transactions on Information Forensics and Security. – 2025.
4. Winter P., Edmundson A., Roberts L. M., Dutkowska-Żuk A., Chetty M., & Feamster N. How do tor users interact with onion services? //27th USENIX Security Symposium (USENIX Security 18). – 2018. – С. 411-428.
5. Басыня Е. А., Хиценко В. Е., Рудковский А. А. Метод идентификации киберпреступников, использующих инструменты сетевого анализа информационных систем с применением технологий анонимизации //Доклады Томского государственного университета систем управления и радиоэлектроники. – 2019. – Т. 22. – №. 2. – С. 45-51.
6. Owen G., Savage N. Empirical analysis of tor hidden services //IET Information Security. – 2016. – Т. 10. – №. 3. – С. 113-118.
7. Sanatinia A., Park J., Blass E.-O., Mohaisen A., & Noubir G. A Privacy-preserving longevity study of Tor's hidden services //arXiv preprint arXiv:1909.03576. – 2019.
8. Boshmaf Y., Perera I., Kumarasinghe U., Liyanage S., Al Jawaheri H. Dizzy: large-scale crawling and analysis of onion services //Proceedings of the 18th International Conference on Availability, Reliability and Security. – 2023. – С. 1-11.
9. Learmonth I. R., Loesing K. Towards modernising data collection and archive for the Tor network //arXiv preprint arXiv:1812.08429. – 2018.
10. Biryukov A., Pustogarov I., Weinmann R. P. Trawling for tor hidden services: Detection, measurement, deanonymization //2013 IEEE Symposium on Security and Privacy. – IEEE, 2013. – С. 80-94.
11. Owenson G., Cortes S., Lewman A. The darknet's smaller than we thought: The life cycle of Tor Hidden Services //Digital Investigation. – 2018. – Т. 27. – С. 17-22.
12. Höller, T., & Mayrhofer, R. (2025). Evaluating Onion Address Collection Methods. In PETS 2025 / Free and Open Communications on the Internet (pp. 28–36). URL: <https://www.petsymposium.org/foci/2025/foci-2025-0009.pdf>
13. Buitrago López A., Pastor-Galindo J., Gómez Mármol F. Updated exploration of the Tor network: advertising, availability and protocols of onion services //Wireless Networks. – 2024. – Т. 30. – №. 9. – С. 7527-7541.
14. Huete Trujillo D. L., Ruiz-Martínez A. Tor hidden services: A systematic literature review //Journal of Cybersecurity and Privacy. – 2021. – Т. 1. – №. 3. – С. 496-518.
15. Оценка применимости инструментов анализа скрытых сервисов Tor // Интерэкспо Гео-Сибирь, 2025г. Т. 7, Вып. 2 Стр. 118-123
16. Bergman, J., & Popov, O. B. (2023). Exploring Dark Web Crawlers: A Systematic Literature Review of Dark Web Crawlers and Their Implementation. IEEE Access, 11, 35914–35933. URL: <https://su.diva-portal.org/smash/person.jsf?pid=authority-person:99686>

© М. С. Каранетьянц, 2026

Н. Карпетянц¹✉

Методы обработки и хранения данных транзакций сети Bitcoin

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: nkarapetyants@mephi.ru

Аннотация. Сеть Bitcoin и лежащая в ее основе технология распределенного реестра представляют собой значимый элемент глобальной финансово-технической инфраструктуры, мониторинг которой представляет практический интерес для специалистов в области информационной безопасности в связи с устойчивым ростом объема незаконных криптовалютных операций. Существующие реализации систем работы с транзакционными данными зачастую опираются на разрозненные инструменты без системного разграничения этапов конвейера обработки и моделей организации хранения, что снижает отказоустойчивость сбора гетерогенной информации и эффективность последующих аналитических запросов. Целью настоящей работы является систематизация и обоснование выбора методов агрегации, потоковой обработки и многоуровневого хранения транзакционных данных сети Bitcoin для повышения эффективности их аналитического исследования в задачах информационной безопасности. В работе предложена архитектура отказоустойчивого конвейера обработки данных, формализующая этапы от первичной агрегации информации из узла Bitcoin Core и внешних OSINT-источников до финальной стандартизации, а также обоснован мультимодельный подход к хранению, сочетающий реляционную, колоночную, графовую, резидентную и объектную модели организации данных. Результаты показывают, что выбор инструментов определяется типом данных и характером аналитических запросов. Использование Apache Kafka, Flink, NiFi и Airflow позволяют реализовать буферизацию, потоковую нормализацию, маршрутизацию и оркестрацию конвейера, а комбинация СУБД PostgreSQL, ClickHouse, Neo4j, Redis и хранилище MinIO в совокупности покрывают весь спектр транзакционных и аналитических запросов. Практическая значимость работы состоит в обосновании архитектуры конвейера на базе комбинации инструментов вендора Apache и мультимодельной стратегии хранения, минимизирующей задержки потоковой нормализации и обеспечивающей горизонтальную масштабируемость. Направления дальнейших исследований включают разработку алгоритмов машинного обучения для автоматизированной кластеризации адресов и выявления сложных паттернов транзакционной активности.

Ключевые слова: Bitcoin, Blockchain, KYT, Базы данных, Обработка данных

N. Karapetyants¹✉

Methods for processing and storing bitcoin network transaction data

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: nkarapetyants@mephi.ru

Abstract. Abstract. The Bitcoin network and its underlying distributed ledger technology constitute a significant component of the global financial and technical infrastructure, the monitoring of which is of practical interest to information security specialists in light of the steady growth in the volume of illicit cryptocurrency operations. Existing implementations of transaction data processing systems often rely on fragmented tools without a systematic separation of pipeline processing stages and storage organization models, which reduces the fault tolerance of heterogeneous data collection and the

efficiency of subsequent analytical queries. The aim of this study is to systematize and substantiate the selection of methods for the aggregation, stream processing, and multi-tier storage of Bitcoin network transaction data in order to improve the effectiveness of their analytical investigation in information security tasks. The paper proposes the architecture of a fault-tolerant data processing pipeline that formalizes the stages from the primary aggregation of information from a Bitcoin Core node and external OSINT sources to final standardization, and substantiates a multi-model storage approach combining relational, columnar, graph, in-memory, and object data organization models. The results show that the choice of tools is determined by the data type and the nature of analytical queries. The use of Apache Kafka, Flink, NiFi, and Airflow enables the implementation of buffering, stream normalization, routing, and pipeline orchestration, while the combination of the PostgreSQL, ClickHouse, Neo4j, and Redis DBMS together with the MinIO storage collectively covers the full range of transactional and analytical queries. The practical significance of the work lies in the substantiation of a pipeline architecture based on a combination of Apache vendor tools and a multi-model storage strategy that minimizes stream normalization latencies and ensures horizontal scalability. Future research directions include the development of machine learning algorithms for the automated clustering of addresses and the detection of complex patterns of transactional activity.

Keywords: Bitcoin, Blockchain, KYT, ETL, Databases, Data processing

Введение

В современных условиях сеть Bitcoin и лежащая в ее основе технология распределенного реестра трансформировались в значимый элемент глобальной финансово-технической инфраструктуры. Популярность цифровых активов обусловлена возможностью проведения круглосуточных трансграничных платежей с относительно низкими издержками, где ключевую роль играют стейблкоины как инструменты с высокой ликвидностью. Широкая доступность и неизменяемость данных в публичных блокчейнах обеспечивают высокий уровень прозрачности, позволяя любому его участнику проводить транзакционный анализ.

Развитие экосистем криптовалют сопровождаются появлением новых угроз информационной безопасности. Согласно отчетным данным за 2025 год, совокупный объем незаконных криптовалютных операций достиг 158 миллиардов долларов США, что в абсолютном выражении является историческим максимумом. Тем не менее, относительная доля криминальной активности в общем обороте рынка демонстрирует тенденцию к снижению и составляет 1,2% (против 1,3% годом ранее). Особую проблему в сфере кибербезопасности представляют хакерские атаки и программные эксплойты: в 2025 году в результате 150 инцидентов было похищено активов на сумму 2,87 миллиарда долларов [1, 2]. Подобная статистика подчеркивает необходимость совершенствования средств защиты, использующих методы сбора, обработки, хранения и анализа транзакционных данных.

Нормативно-правовая база в области регулирования криптовалют развивается в направлении ужесточения надзора за финансовыми потоками и внедрения международных стандартов противодействия отмыванию доходов. Современные регуляторные подходы ориентированы на использование аналитических инструментов блокчейна для идентификации подозрительных паттернов и пресечения незаконной деятельности на ранних стадиях. Адаптация законодательства

к архитектурным особенностям блокчейна направлена на минимизацию рисков киберпреступности путем установления требований к операторам по обеспечению сквозной прозрачности всех операций с цифровыми активами.

Исследования в области данной проблематики осуществляется ведущими учеными такими как Басыня Е.А., Стрельцов А.С., Французова Г.А., Мелихова Ю. А., Анаденков Д. В., Милосердова Л. П., Гапанович А. М., Талай Е. В., Вилаков Н. В., Бочаров М. И. [3 – 8].

В работе [9] был разработан виртуальный испытательный стенд для автоматизации процессов сбора, обработки, хранения и анализа транзакционных данных сети Bitcoin. В текущем исследовании рассматриваются подходы расширяющие возможности обработки и хранения данных транзакций сети Bitcoin.

Методы и материалы

Процесс агрегации первичных данных из сети Bitcoin (через ZMQ и RPC-интерфейсы узла Bitcoin Core) [9] и внешних интернет-источников (форумов, OSINT-отчетов, баз вредоносного ПО) требует создания отказоустойчивого слоя промежуточного хранения. В качестве основного инструмента агрегации и первичной буферизации обосновано использование Apache Kafka [10]. Данная платформа с открытым исходным кодом позволяет объединять разнородные потоки данных в единый поток событий, обеспечивая высокую пропускную способность и гарантированную доставку сообщений. Это необходимо для предотвращения потери данных в условиях пиковых нагрузок в блокчейн-сети и при нестабильной скорости получения ответов от веб-ресурсов.

Поточная обработка данных в режиме реального времени делегируется Apache Flink [11]. Этот инструмент позволяет проводить мгновенную нормализацию и очистку поступающих транзакций и текстовых отчетов. В отличие от аналогов, Flink обеспечивает развитую поддержку состояний, что необходимо для выявления паттернов подозрительной активности в режиме мягкого реального времени – например, для сопоставления адреса из текущего блока с упоминанием в базе вредоносного ПО или на форумах. Это позволяет структурировать данные еще до их попадания в постоянное хранилище, сокращая задержки при последующем анализе.

Для специфической задачи структуризации и предварительной очистки неструктурированной информации из открытых источников (тексты форумов, отчеты, логи вредоносного ПО) целесообразно использование Apache NiFi [12]. Данный инструмент предоставляет расширенные возможности по управлению потоками данных и автоматизации маршрутизации между различными парсерами и нормализаторами. Преимущество NiFi заключается в его гибкости при работе с изменчивыми форматами данных из сети Интернет и возможности визуального контроля происхождения данных.

Координация и автоматизация всех этапов обработки – от первичной агрегации до финальной трансформации может осуществляться оркестратором Apache Airflow [13]. Он позволяет формализовать зависимости между процессами в виде направленных ациклических графов. Применение Airflow обосновано необходимостью строгого контроля последовательности операций очистки и структуризации, а также

обеспечения отказоустойчивости всей системы за счет механизмов автоматического перезапуска задач и мониторинга состояния конвейеров данных.

Завершающим этапом обработки является приведение данных к стандартизированным структурам (например, в форматах Apache Avro или Apache Parquet) [14] перед их последующей передачей в системы хранения. Этот процесс включает окончательную трансформацию атрибутов и проверку схем данных на соответствие заданным правилам. Использование бинарных форматов с поддержкой схем обосновано необходимостью минимизации объема подготовленных данных и обеспечения их высокой совместимости с любыми аналитическими инструментами, которые будут задействованы на следующих этапах.

Исследование методов хранения данных для последующего анализа сети Bitcoin обусловлено необходимостью обработки разнородной информации: структурированных данных из Bitcoin Core (блоки, транзакции) и неструктурированных данных из внешних источников (форумы, отчеты о мошенничестве, сигнатуры вредоносного ПО). Использование различных моделей организации данных позволяет оптимизировать выполнение как транзакционных, так и аналитических запросов в рамках единой экосистемы.

Реализация строковой организации данных (реляционной модели) наиболее эффективно осуществляется посредством СУБД PostgreSQL [15]. Данный инструмент с открытым исходным кодом оптимален для хранения метаданных, информации о пользователях форумов и структурированных отчетов о мошенничестве. PostgreSQL поддерживает строгую согласованность (ACID), систему индексации и возможность построения сложных связей между сущностями внешних источников и конкретными идентификаторами в блокчейне. PostgreSQL выступает в роли основного реестра для нормализованных данных, требующих высокой точности и частого обновления отдельных записей.

Анализ массивов исторических данных требует колоночной организации, которую предоставляет СУБД ClickHouse [16]. Блокчейн-данные, такие как значения входов/выходов и временные метки транзакций, характеризуются высокой избыточностью, что позволяет ClickHouse достигать значительной степени сжатия и обеспечивать высокую скорость выполнения агрегирующих запросов. Использование этого инструмента позволяет производить быстрое сканирование миллиардов записей для выявления аномалий в объемах передаваемых средств без критической нагрузки на основные транзакционные базы.

Исследование топологии сети и сложных взаимосвязей между адресами опирается на графовую организацию данных, реализуемую в Neo4j [17]. В контексте Bitcoin транзакции и адреса естественным образом формируют направленный граф. Применение Neo4j обосновано эффективностью алгоритмов обхода графа для выявления цепочек перемещения активов, кластеризации адресов, принадлежащих одному субъекту, и отслеживания пути криптовалют от источника (например, адреса, связанного с вредоносным ПО) до конечного получателя. Графовая модель позволяет выполнять запросы на поиск кратчайших путей, которые крайне ресурсозатратны в традиционных реляционных СУБД.

Высокая скорость доступа к критически важным и часто запрашиваемым объектам может быть обеспечена за счет резидентного хранения (in-memory) на

базе Redis [18]. Данный инструмент лучше подходит для кэширования актуального набора неизрасходованных выходов (UTXO), оперативного мониторинга сети в реальном времени и хранения промежуточных результатов обработки данных из блокчейн-обозревателей. Redis обеспечивает минимальные задержки доступа к данным, что является критическим фактором для систем оперативного обнаружения подозрительной активности, где время реакции на инцидент строго ограничено.

Систематизация больших объемов неструктурированной информации, таких как бинарные файлы блоков, дампы веб-страниц или файлы вредоносного ПО, возлагается на объектно-ориентированное хранилище MinIO [19]. Это высокопроизводительное S3-совместимое решение с открытым кодом позволяет масштабировать хранилище горизонтально и эффективно управлять объектами любого размера без жесткой схемы данных.

Результаты и обсуждения

В результате проведенных исследований методов обработки и хранения данных транзакций сети Bitcoin была сформирована сводная таблица 1, наглядно демонстрирующая распределение технических средств в соответствии с их функциональным назначением, типом решаемых задач и характеристиками обрабатываемой информации. Данная систематизация позволяет сопоставить возможности современных Big Data инструментов с архитектурными требованиями системы, обеспечивая комплексный подход к анализу как строго структурированных блоков транзакций, так и нелинейных данных из внешних источников.

Внедрение конвейера на базе Kafka и Flink позволяет обеспечить минимальные задержки при потоковой нормализации данных, а связка NiFi и Airflow дает возможность автоматизировать отказоустойчивый сбор неструктурированной информации из внешних источников. Применение бинарных форматов Avro и Parquet способствует оптимизации последующего анализа за счет механизмов строгой типизации и высокого сжатия данных, что минимизирует нагрузку на вычислительные ресурсы.

Таблица 1. Средства обработки и хранения данных транзакций сети Bitcoin

Инструмент	Решаемая задача	Тип обрабатываемых данных
Apache Kafka	Промежуточное хранение, буферизация и агрегация потоков	Разнородные события из узла Bitcoin Core и внешних API
Apache Flink	Поточная обработка, нормализация и фильтрация	Транзакции в реальном времени, текстовые оповещения
Apache NiFi	Управление маршрутизацией и первичная очистка данных	Неструктурированные данные (форумы, OSINT-отчеты)
Apache Airflow	Оркестрация рабочих процессов и контроль последовательности задач	Метаданные конвейеров, зависимости между этапами

Окончание табл. 1

Инструмент	Решаемая задача	Тип обрабатываемых данных
PostgreSQL	Хранение структурированных реестров и метаданных	Информация о пользователях, отчеты о мошенничестве
ClickHouse	Аналитическая обработка больших исторических массивов	Временные метки, значения входов/выходов транзакций
Neo4j	Исследование топологии сети и графовых взаимосвязей	Адреса, связи между транзакциями, кластеры субъектов
Redis	Резидентное хранение для обеспечения минимальных задержек	Актуальный набор UTXO, оперативные результаты мониторинга
MinIO	Объектное хранение неструктурированных массивов	Бинарные дампы блоков, файлы вредоносного ПО, PDF-отчеты
Avro / Parquet	Финальная структуризация и оптимизация хранения	Подготовленные аналитические датасеты со строгой схемой

В части хранения применяется мультимодельный подход, где выбор модели данных продиктован спецификой аналитических запросов. PostgreSQL обеспечивает необходимую согласованность метаданных, ClickHouse позволяет достичь кратного ускорения агрегирующих запросов, а Neo4j способствует эффективному выявлению графовых связей и кластеризации адресов. Использование Redis и MinIO позволяет сбалансировать скорость доступа к оперативным данным и общую горизонтальную масштабируемость системы.

Заключение

В ходе проведенного исследования были рассмотрены методы агрегации, обработки и многоуровневого хранения данных сети Bitcoin и сопутствующих внешних источников. Практическим результатом работы является обоснование архитектуры высокопроизводительного конвейера на базе Apache Kafka, Flink и Airflow, а также применение мультимодального подхода хранения данных, объединяющей реляционный, колоночный и графовый методы хранения данных. Сформированный перечень инструментов и методов позволяет обеспечить отказоустойчивую нормализацию гетерогенных данных, преобразуя сырые транзакции и неструктурированные данные в стандартизированные форматы Avro и Parquet для последующего анализа.

В рамках дальнейших исследований планируется реализация алгоритмов машинного обучения на базе подготовленных массивов данных для автоматизированной кластеризации адресов и выявления сложных паттернов транзакционной активности, связанных с угрозами информационной безопасности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. The Chainalysis 2026 Crypto Crime Report [Электронный ресурс] // Chainalysis. URL: <https://www.chainalysis.com/reports/crypto-crime-2026/> (дата обращения: 28.04.2026).

2. 2026 Crypto Crime Report [Электронный ресурс] // TRM. URL: <https://www.trmlabs.com/reports-and-whitepapers/2026-crypto-crime-report> (дата обращения: 28.04.2026).
3. Basynya E. A., Karapetyants N., Karapetyants M. Bitcoin Transaction Analysis System //Programming and Computer Software. – 2024. – Т. 50. – №. Suppl 2. – С. S104-S112.
4. Стрельцов А. С., Французова Г. А., Басыня Е. А. Разработка системы сбора, обработки, анализа, идентификации корреляции событий информационной инфраструктуры предприятия //Системы анализа и обработки данных. – 2023. – №. 1 (89). – С. 101-113.
5. Мелихова Ю. А., Анаденков Д. В. Блокчейн bitcoin: принципы работы и особенности //Проблемы развития современного общества. – 2023. – С. 287-289.
6. Милосердова Л. П. Системные аспекты обработки транзакций в блокчейн-узлах: конкурентность и управление памятью //Актуальные вопросы современной науки 3. – 2026. – С. 30.
7. Гапанович А. М., Талай Е. В. О криптокошельках, транзакциях и аресте криптовалюты //ББК 67.408 я43 О75. – 2024. – С. 324.
8. Вилаков Н. В., Бочаров М. И. Способы защиты криптовалютных блокчейн систем и систематизация угроз //Системная инженерия и информационные технологии. – 2025. – Т. 7. – №. 2 (21). – С. 143-154.
9. Mun H., Lee Y. Bitsql: A sql-based bitcoin analysis system //2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). – IEEE, 2022. – С. 1-8.
10. Raptis T. P., Passarella A. A survey on networked data streaming with apache kafka //IEEE access. – 2023. – Т. 11. – С. 85333-85350.
11. Mezati M., Aouria I. Flink-ML: machine learning in Apache Flink Flink-ML: aprendizado de máquina no Apache Flink Flink-ML: aprendizagem automático en Apache Flink //Brazilian Journal of Technology. – 2024. – Т. 7. – №. 4. – С. 01-18.
12. Carthen C. et al. Orchestrating apache nifi/minifi within a spatial data pipeline //2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA). – IEEE, 2023. – С. 366-371.
13. Haines S. Workflow orchestration with apache airflow //Modern Data Engineering with Apache Spark: A Hands-On Guide for Building Mission-Critical Streaming Applications. – Berkeley, CA : Apress, 2022. – С. 255-295.
14. Nelluri S. R., Saldanha F. A. A. Mastering Big Data Formats: ORC, Parquet, Avro, Iceberg, and the Strategy of Selection //International Journal of Computer Trends and Technology. – 2025. – Т. 73. – №. 1. – С. 44-50.
15. Salunke S. V., Ouda A. A performance benchmark for the PostgreSQL and MySQL databases //Future Internet. – 2024. – Т. 16. – №. 10. – С. 382.
16. Schulze R. et al. Clickhouse-lightning fast analytics for everyone //Proceedings of the VLDB Endowment. – 2024. – Т. 17. – №. 12. – С. 3731-3744.
17. Naha R. T., Zhang K. Cryptocurrencies forensics with real-time intelligence and graph database: A comprehensive review //2024 6th International Conference on Blockchain Computing and Applications (BCCA). – IEEE, 2024. – С. 1-12.
18. Abu Kausar M., Nasar M., Soosaimanickam A. A study of performance and comparison of nosql databases: MongoDB, cassandra, and redis using ycsb //Indian Journal of Science and Technology. – 2022. – Т. 15. – №. 31. – С. 1532-1540.
19. Malhotra S. et al. Evaluating fault tolerance and scalability in distributed file systems: A case study of gfs, hdfs, and minio //arXiv preprint arXiv:2502.01981. – 2025.

© Н. Карпетьянц, 2026

К. Ю. Копыл¹✉

Анализ проблематики автоматизированного аудита Kubernetes-кластеров

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: kikopyl3@gmail.com

Аннотация. В работе рассматривается проблематика автоматизированного аудита информационной безопасности Kubernetes-кластеров в условиях массового внедрения микросервисной архитектуры. Актуальность темы обусловлена тем, что существующие инструменты сканирования уязвимостей и проверки конфигураций генерируют избыточное количество оповещений, большинство из которых не представляют реальной угрозы в конкретной эксплуатационной среде. Данная проблема определяется как «контекстная слепота» изолированных сканеров, опирающихся исключительно на статические метрики вроде CVSS. Цель исследования – анализ ограничений современных подходов к аудиту и обоснование необходимости перехода к контекстно-ориентированной модели оценки рисков, учитывающей сетевую экспозицию, ролевую модель доступа, чувствительность данных и параметры среды исполнения. В ходе работы выполнена классификация существующих решений по четырем классам (анализаторы конфигураций, сканеры образов, системы runtime, агрегаторы) и проведен их сравнительный анализ по расширенным техническим критериям. Выявлены принципиальные ограничения существующего инструментария, включая ближайший инженерный аналог — утилиту Reticulum. Основным результатом работы является формулирование концептуальных требований к мета-анализатору, агрегирующему данные узкоспециализированных сканеров и выполняющему перерасчет риска на основе метаданных Kubernetes API. Предложенная концепция создает методическую основу для перехода к риск-ориентированной модели управления уязвимостями и определяет направления дальнейших исследований, включая валидацию контекстных коэффициентов методами многокритериального анализа.

Ключевые слова: Kubernetes, контейнеризация, сканирование уязвимостей, контекстно-ориентированный анализ, оценка рисков, DevSecOps

К. У. Копыл¹✉

Analysis of the problems of automated auditing of kubernetes clusters

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: kikopyl3@gmail.com

Abstract. This paper examines the problem of automated information security auditing of Kubernetes clusters in the context of widespread adoption of microservice architecture. The relevance of the topic stems from the fact that existing vulnerability scanning and configuration checking tools generate an excessive number of alerts, most of which do not represent a real threat in a specific operational environment. This problem is defined as "context blindness" of isolated scanners that rely solely on static metrics such as CVSS. The aim of the research is to analyze the limitations of modern auditing approaches and to justify the need for a transition to a context-aware risk assessment model that takes into account network exposure, role-based access control, data sensitivity, and execution environment parameters. In the course of the work, a classification of existing solutions into four classes (configuration analyzers, image scanners, runtime systems, and aggregators) was carried out, and their

comparative analysis according to extended technical criteria was performed. Fundamental limitations of the existing toolset were identified, including the closest engineering counterpart – the Reticulum utility. The main result of the work is the formulation of conceptual requirements for a meta-analyzer that aggregates data from specialized scanners and performs risk recalculation based on Kubernetes API metadata. The proposed concept provides a methodological foundation for transitioning to a risk-based vulnerability management model and defines directions for further research, including the validation of context coefficients using multi-criteria analysis methods.

Keywords: Kubernetes, containerization, vulnerability scanning, context-aware analysis, risk assessment, DevSecOps

Введение

В условиях массового перехода корпоративных ИТ-инфраструктур на микросервисную архитектуру платформа оркестрации Kubernetes закрепила статус основного инструмента управления контейнерами. По данным Cloud Native Computing Foundation (2023), его используют 71% компаний Fortune 100 [1]. Однако сложность настройки распределенных компонентов расширяет поверхность атаки и среди наиболее распространенных проблем безопасности Kubernetes выделяются уязвимости приложений (37% организаций), ошибки управления доступом (32%), небезопасные образы (32%) [2]. При этом 45% успешных атак нарушают основную деятельность организаций [3].

Нормативная база (NIST SP 800-190 [4], руководство NSA/CISA, Приказ ФСТЭК №118 [5]) требует регулярного контроля кластеров. Однако внедрение средств автоматизированного сканирования порождает новую эксплуатационную проблему – это избыточность оповещений (alert fatigue). Причина заключается в том, что большинство практических детекторов и сканеров оценивают риск изолированно, опираясь исключительно на статическую базовую метрику CVSS, без учета динамического контекста оркестрируемой среды.

Однако сканирование вызывает проблему избыточности оповещений (alert fatigue) – ИБ-подразделения получают массу отчетов о критических уязвимостях, не несущих реальной угрозы в текущей архитектуре. Причина: сканеры оценивают риск изолированно по статичной метрике CVSS, игнорируя динамический контекст среды.

Целью данного исследования ставится анализ проблематики автоматизированного аудита Kubernetes-кластеров, выявление ограничений существующих подходов и обоснование необходимости перехода к контекстно-ориентированной модели оценки рисков, учитывающей сетевую экспозицию, ролевую модель доступа и параметры среды исполнения.

Сравнительный анализ подходов к аудиту

Для выявления причин неэффективности классического аудита существующие программные решения целесообразно классифицировать на четыре основных группы, каждый из которых имеет специфические ограничения.

К первому классу решений относятся статические анализаторы конфигураций, такие как Kube-bench, Checkov и KubeLinter. Их задача сводится к проверке

YAML-манифестов и параметров запуска узлов на соответствие стандартам, таким как CIS Kubernetes Benchmark. Ограничение подхода заключается в бинарной логике проверок: система фиксирует нарушение (например, отсутствие сетевой политики), но не способна оценить, какой именно код выполняется внутри изолируемого сегмента [6].

Следующую группу образуют сканеры уязвимостей образов, представленные инструментами Trivy, Grype и Clair. Инструменты анализируют слои контейнеров, извлекают спецификации программного обеспечения (SBOM) и ищут совпадения с базами CVE. Главный недостаток метода состоит в отрыве от среды выполнения. Сканер маркирует уязвимую библиотеку высоким приоритетом, не анализируя сетевую связность пода и его полномочия в кластере [7].

Отдельную категорию составляют системы анализа времени выполнения. Решения на базе eBPF (Falco, Tetragon) перехватывают системные вызовы на уровне ядра ОС хоста. Несмотря на высокую точность выявления аномалий (например, попыток запуска интерактивной оболочки), подход носит реактивный характер и неприменим для задач превентивного патч-менеджмента [8].

Наконец, четвертую группу формируют системы агрегации, к которым относят DefectDojo и vimp. Данные платформы консолидируют отчеты сторонних утилит в единой базе данных. Однако на практике агрегация сводится к централизации хранения: уязвимости в коде и параметры доступа пода фиксируются независимо друг от друга, без выстраивания корреляционных связей.

Результаты сравнительного анализа обобщены в табл. 1.

Таблица 1. Сравнение подходов к автоматизированному аудиту

Класс решений	Проверяемые объекты	Базовая метрика риска	Эксплуатационные ограничения
Анализаторы конфигураций	Манифесты, узлы K8s, RBAC	Бинарная (соответствует / не соответствует)	Изоляция от данных об уязвимостях прикладного ПО
Сканеры образов	Пакеты ОС, зависимости кода	Базовый балл CVSS	Отсутствие учета сетевой связности и привилегий
Системы Runtime	Системные вызовы, трафик	Приоритет сигнатуры	Реактивность, невозможность оценки до инцидента
Системы агрегации	Отчеты сторонних утилит	Сводная статистика	Отсутствие перерасчета рисков на основе топологии

Анализ приведенной выше таблицы показывает, что узкая специализация инструментов формирует системную проблему, при которой применение средств по отдельности снижает общую достоверность аудита.

Детальное сравнение инструментов по техническим критериям

Для более глубокого анализа целесообразно перейти от обобщенных классов к сравнению конкретных программных продуктов, отобранных по одному

представителю из каждого функционального класса, а также инструмента, заявляющего о наличии элементов контекстной оценки. В перечень включены: Kube-bench (анализатор конфигураций), Trivy (сканер образов), Falco (Runtime Security), Kubescape (гибридная платформа, сочетающая функции агрегации и оценки соответствия) и Reticulum (утилита контекстной приоритизации уязвимостей).

Таблица 2. Сравнение инструментов аудита по техническим критериям

Инструмент	Модель сбора данных	Учет контекста исполнения	Гибкость настройки приоритетов	Корреляция CVE и событий	Формат и качество отчетов
Kube-bench	Безагентный анализ манифестов и конфигурации узлов через API или локальные файлы	Нет. Бинарная логика проверок (Pass/Fail/Warning), без учета прикладных данных	Фиксированный набор правил CIS	Нет	Консольный вывод и структурированный JSON/текстовый отчет
Trivy	Безагентное сканирование реестров и локальных образов	Минимальный учет контекста: игнорирование CVE по ID или через файл .trivyignore	Наличие .trivyignore	Нет	Поддержка JSON, SARIF и HTML-форматов отчетов
Falco	Агентский анализ через eBPF и модуль ядра	Высокий учет контекста: правила с условиями по namespace, labels, образу и другим параметрам	Пользовательские правила (YAML)	С SIEM/UI через внешние системы	Логи в форматах syslog и JSON, пригодных для интеграции с SIEM
Kubescape	Безагентный анализ через API Kubernetes	Частичный учет контекста: исключение пространств имен, но без динамической топологии и экспозиции	Фреймворки (NSA, MITRE), без гибкой настройки весов	Базовая корреляция реализована в open-source версии. Визуальная корреляция реализована в коммерческой версии	Поддержка JSON, SARIF, HTML и JUnit XML и PDF-форматов отчетов
Reticulum	Безагентный анализ манифестов и данных API Kubernetes	Сетевая доступность (Ingress, тип сервиса) как фактор приоритета	Жестко заданные весовые коэффициенты	Нет	Консольный вывод и структурированный JSON-отчет

В качестве критериев сравнения выбраны параметры, отражающие способность инструмента учитывать эксплуатационный контекст и адаптироваться под нужды конкретной инфраструктуры (табл. 2).

Анализ приведенной выше таблицы показывает, что ни один из рассмотренных инструментов не обеспечивает полноценной корреляции между статическими уязвимостями и динамическим окружением их эксплуатации. Наиболее близким к решению данной задачи является проект Reticulum, который реализует контекстную корректировку приоритета CVE в зависимости от сетевой доступности

сервиса, однако используемые весовые коэффициенты задаются в рамках контекстного анализа уязвимостей без явного формального обоснования. В то же время, в доступных описаниях не найдено свидетельств того, что инструмент учитывает ряд важных факторов безопасности: наличие привилегированного режима у контейнера (`securityContext.privileged`), тип обрабатываемых данных в виде `StatefulSet` с подключенными постоянными томами (PVC), целевое назначение пространства имен (`production / development`) и детализированные ролевые права `ServiceAccount`.

Таким образом, существующий инструментарий, включая *Reticulum*, либо игнорирует контекст, либо реализует его учет в ограниченном и эмпирическом виде, что создает предпосылки для разработки формализованного метода контекстно-ориентированной оценки рисков.

Проблема контекстной слепоты

Неспособность системы аудита учитывать архитектурные и конфигурационные условия функционирования уязвимого компонента в рамках данной работы определяется как «контекстная слепота».

На практике статические метрики риска теряют релевантность в контейнерной среде. В качестве примера можно рассмотреть обнаружение критической уязвимости (оценка CVSS 9.8) в библиотеке `curl` внутри рабочего пода, развернутого в изолированном пространстве имен, не имеющего маршрутизации извне (отсутствие объектов `Ingress`) и ограниченного запрещающими сетевыми политиками. В таких условиях риск практической эксплуатации угрозы является минимальным.

С другой стороны, запуск контейнера с привилегированным доступом в продуктивном контуре представляет собой критическую архитектурную уязвимость, позволяющую злоумышленнику преодолеть границы изоляции и получить доступ к операционной системе сервера, на котором размещен контейнер, либо к другим контейнерам, выполняющимся в той же среде [9, 10].

Концепция контекстно-зависимого аудита

Проведенный анализ показывает, что устранение «контекстной слепоты» невозможно в рамках изолированного применения существующих сканеров. Требуется переход от статической инвентаризации угроз к управляемому корреляционному анализу. В связи с этим предлагается архитектура мета-анализатора, который агрегирует результаты работы узкоспециализированных утилит (например, *Kube-bench*, *Trivy* и других) и выполняет перерасчет уровня риска на основе метаданных, получаемых непосредственно из *Kubernetes-API*. Простейшим интуитивным подходом является введение мультипликативной модели:

$$R = V_{base} \times K_{context} ,$$

где R – единый показатель риска, V_{base} – исходная оценка уязвимости (например, балл CVSS), а $K_{context}$ – композитный коэффициент контекста.

Однако прямое применение этой модели без формального обоснования весовых коэффициентов приводит к ряду нежелательных эффектов и проблем эксплуатации.

Во-первых, числовые значения весов контекста зачастую задаются разработчиками эмпирически, без опоры на статистику инцидентов или формализованные экспертные процедуры.

Во-вторых, введение «презумпции критичности» и присвоение максимальных весов компонентам с неопределенным контекстом (что часто предлагается для предотвращения ложноотрицательных срабатываний) на практике приводит к лавинообразному росту ложноположительных срабатываний.

В качестве третьего вектора проблем можно обозначить отсутствие компенсационных механизмов. Простое перемножение коэффициентов предполагает, что все факторы равнозначны и линейно влияют на итоговый риск. В реальности наличие привилегированного режима является блокирующим фактором, который не может быть компенсирован низкой сетевой экспозицией или отнесением контейнера к тестовой среде.

Таблица 3. Группы контекстных параметров для оценки риска

Группа параметров	Примеры анализируемых данных из Kubernetes API	Характер влияния на риск
Сетевая экспозиция	Ingress, Service (тип LoadBalancer/NodePort), NetworkPolicy	Определяет доступность уязвимого компонента для внешнего атакующего
Привилегии и доступ (IAM)	SecurityContext (privileged, capabilities), ServiceAccount (роли и ClusterRole)	Определяет потенциальный ущерб от компрометации пода (возможность выхода на узел или доступ к секретам)
Чувствительность данных	Наличие PersistentVolumeClaim, монтирование Secrets/ConfigMaps	Влияет на критичность последствий утечки или повреждения данных
Характеристика среды	Namespace (метки prod/dev), пользовательские аннотации	Отражает бизнес-критичность сервиса и допустимый уровень риска

Для преодоления указанных ограничений разработка формализованного метода оценки риска должна опираться на следующую концептуальную схему, включающую четыре группы анализируемых параметров (табл. 3).

Научная задача на следующем этапе исследования заключается не в эвристическом подборе весов, а в выборе и обосновании математического аппарата, адекватного природе угроз в оркестрируемой среде. В качестве перспективных направлений могут быть рассмотрены методы многокритериального принятия решений, в частности метод анализа иерархий (МАИ) [11], а также аппарат нечеткой логики.

Таким образом, предложенная концепция задает вектор перехода к риск-ориентированной модели. Практическая реализация такой модели позволит алгоритмически снижать приоритет уязвимостей, характеризующихся низким уровнем реальной эксплуатационной угрозы в данном архитектурном контексте.

Заключение

Предложенный в работе анализ задает схему сравнительной оценки инструментов аудита Kubernetes-кластеров с точки зрения применимости для контекстной приоритизации угроз. Научная новизна подхода состоит в переходе от изолированной оценки уязвимостей по метрикам CVSS к контекстно-ориентированной модели, учитывающей сетевую экспозицию, ролевую модель доступа, чувствительность данных и параметры среды исполнения. В отличие от существующих инженерных решений, реализующих эвристическую корректировку приоритетов, в работе обоснована необходимость формализованного метода расчета риска на основе метаданных кластера.

Ключевой результат исследования заключается в выявлении проблемы «контекстной слепоты» изолированных сканеров и формулировании концептуальных требований к мета-анализатору, агрегирующему результаты узкоспециализированных утилит. Практическая значимость работы состоит в создании методической основы для перехода к риск-ориентированной модели управления уязвимостями, позволяющей снизить избыточность оповещений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Kubernetes Project Journey Report [Электронный ресурс] // CNCF. – 2023. – URL: <https://www.cncf.io/reports/kubernetes-project-journey-report/> (дата обращения: 22.10.2025).
2. Red Hat State of Kubernetes Security Report. – 2024.
3. Актуальные киберугрозы: I–II кварталы 2025 года [Электронный ресурс] // Positive Technologies. – 2025. – URL: <https://ptsecurity.com/research/analytics/aktual-nye-kiberugrozy-i-ii-kvartaly-2025-goda/> (дата обращения: 22.10.2026).
4. Souppaya M. Application Container Security Guide / M. Souppaya, J. Morello, K. Scarfone DOI: 10.6028/NIST.SP.800-190. – National Institute of Standards and Technology, 2017.
5. Требования по безопасности информации. Утверждены приказом ФСТЭК России от 4 июля 2022 г. N 118 // ФСТЭК России. – 2022.
6. Денисенко Д.И. Методика тестирования на проникновение контейнерной технологии Kubernetes / Д.И. Денисенко // Информационная безопасность, искусственный интеллект и большие данные : сб. ст. студентов и аспирантов ИИКС. – 2024. – С. 87.
7. Ramasankar Molleti. Container runtime security detection and prevention techniques // World Journal of Advanced Research and Reviews. – 2024. – Vol. 24. – № 3. – P. 2626–2639.
8. Лавров Б.О. Интеллектуальное обнаружение аномалий в контейнеризованных приложениях: методы, архитектура и инструменты / Б.О. Лавров, М.А. Иванов // Безопасность информационных технологий. – 2025. – Т. 32, № 4. – С. 149-164.
9. Agrawal M. Security Audit of Kubernetes based Container Deployments: A Comprehensive Review / M. Agrawal, K. Abhijeet. – 2020. – Vol. 07. – № 06.
10. Липатова С. Е., Белов Ю. С. Практики обеспечения кибербезопасности в Kubernetes // E-Scio. – 2022. – №. 1 (64). – С. 489-497.
11. Корчагин П.В. Модель контроля целостности кластера Linux-серверов под управлением Kubernetes / П.В. Корчагин, А.Д. Шульженко, А.Б. Кириенко // Вопросы оборонной техники. Серия 16: Технические средства противодействия терроризму. – 2023. – № 11-12 (185-186). – С. 16-26.

© К. Ю. Копыл, 2026

А. Е. Коровянский¹✉, А. А. Шарапов¹

Разработка и исследование серверной части веб-платформы управления научными конференциями с использованием REST API

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: lexparker925@gmail.com

Аннотация. В статье рассматривается проблема недостаточной производительности и отказоустойчивости традиционных монолитных веб-систем для управления научными мероприятиями в условиях пиковых нагрузок. Предложен и исследован подход к разработке серверной части на основе микросервисной архитектуры. Описан процесс декомпозиции системы на независимые сервисы, обоснован выбор технологического стека, включающего фреймворк FastAPI, брокер сообщений RabbitMQ и технологию контейнеризации Docker. Ключевой частью исследования является проведение и анализ результатов нагрузочного тестирования с использованием инструмента Locust. Экспериментально доказано, что предложенная архитектура после оптимизации способна стабильно обрабатывать нагрузку до 250 запросов в секунду при времени отклика, не превышающем 450 мс для 95% запросов. Результаты подтверждают высокую эффективность микросервисного подхода для создания масштабируемых и отказоустойчивых систем в образовательной сфере.

Ключевые слова: микросервисная архитектура, REST API, FastAPI, высокая нагрузка, нагрузочное тестирование, RabbitMQ, Docker, асинхронное программирование, веб-платформа

A. E. Korovyanskiy¹✉, A. A. Sharapov¹

Development and research of the server part of a web platform for managing scientific conferences using REST API

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: lexparker925@gmail.com

Abstract. This article addresses the issue of insufficient performance and fault tolerance of traditional monolithic web systems for managing scientific events under peak loads. An approach to developing a server-side based on a microservice architecture is proposed and investigated. The process of decomposing the system into independent services is described, and the choice of a technology stack, including the FastAPI framework, RabbitMQ message broker, and Docker containerization technology, is justified. A key part of the research is the execution and analysis of load testing results using the Locust tool. It has been experimentally proven that the proposed architecture, after optimization, can stably handle loads of up to 250 requests per second with a response time not exceeding 450 ms for 95% of requests. The results confirm the high efficiency of the microservice approach for creating scalable and fault-tolerant systems in the educational sphere.

Keywords: microservice architecture, REST API, FastAPI, high load, load testing, RabbitMQ, Docker, asynchronous programming, web platform

Введение

Специфика администрирования научных конференций сопряжена с рядом критических инфраструктурных и технологических проблем. Основная проблема проведение таких мероприятий заключается в том, что периоды низкой активности сменяются резкими всплесками в дни окончания приема заявок. Резкое повышение нагрузки становится настоящим стресс-тестом для серверной инфраструктуры. Системы, основанные на монолитной архитектуре, могут не справляться с этой задачей.

Анализ существующих решений, показывает, что они, как правило, базируются на универсальных CMS или устаревших монолитных фреймворках. Это приводит к низкой производительности, нестабильной работе и к потере данных и негативному опыту пользователей. Такие системы дороги в поддержке и практически не поддаются гибкому масштабированию. Для обработки пиковой нагрузки приходится увеличивать мощность всего сервера, что экономически неэффективно.

Целью данной работы является разработка системы, а также исследование и практическое доказательство эффективности применения микросервисной архитектуры для создания производительной, отказоустойчивой и масштабируемой серверной платформы для управления научными конференциями.

Методы и материалы

Для достижения поставленной цели был применен комплексный подход, включающий проектирование распределенной системы, выбор современного технологического стека и проведение проверки производительности.

В основе системы лежит микросервисная архитектура [1]. Система разделена на четыре сервиса с разграниченными зонами ответственности. Процессы регистрации, проверки прав доступа и ведения профилей изолированы внутри модуля Auth service. Бизнес-логика, связанная с координацией научных мероприятий, секций и обработкой поступающих заявок, реализована в рамках Core service. Для снижения нагрузки на основной сервис отправка транзакционных писем была вынесена в асинхронный Notification service. Analytics service отвечает за сбор логов и системных метрик.

Взаимодействие между сервисами осуществляется как синхронно, так и асинхронно через брокер сообщений RabbitMQ, что позволяет разгрузить основной API от тяжелых операций [2–4]. Все внешние запросы проходят шлюз API, реализованный на базе Nginx [5]. Он является единой точкой входа для всей системы [6].

Выбор технологий для реализации проекта был продиктован требованиями к высокой производительности, отказоустойчивости и гибкости архитектуры. В качестве основного фреймворка был выбран FastAPI [7–9]. В отличие от Django, данный фреймворк обладает поддержкой асинхронности через библиотеку `asyncio` [10]. Поддержка асинхронности позволяет системе эффективно обрабатывать большой поток одновременных I/O-операций, включая запросы к базе

данных и внешним сервисам. Это является важным фактором для стабильной работы API.

Для реализации асинхронного взаимодействия между компонентами системы был внедрен брокер сообщений RabbitMQ [11, 12]. Данное решение используется для организации очередей задач с гарантированной доставкой сообщений. Использование брокера сообщений позволяет добиться слабой связности сервисов системы. В качестве системы управления базами данных была выбрана реляционная СУБД PostgreSQL [13]. PostgreSQL обладает высокой надежностью, имеет расширенную поддержку сложных типов данных.

Развертывание и масштабирование было обеспечено с помощью технологий контейнеризации Docker и Docker Compose [14]. Docker позволяет упаковать каждый сервис со всеми зависимостями в изолированный контейнер. В свою очередь, инструмент Docker Compose предоставляет возможность управлять сложной инфраструктурой и достаточно быстро запускать всю систему как единое целое.

Для проверки производительности системы был использован фреймворк Locust [15, 16]. Для проверки был разработан сценарий, имитирующий поведение реальных пользователей. По сценарию после аутентификации виртуальные пользователи в случайном порядке выполняли запросы к API, преимущественно запрашивая списки своих заявок [17].

Нагрузочное тестирование проводилось в два этапа. На первом этапе было проведено моделирование 100 одновременных пользователей для проверки соответствия базовым требованиям. На втором этапе было количество пользователей было увеличено до 300 для выявления узких мест и исследования пределов масштабируемости системы.

Результаты

В результате разработки была создана полнофункциональная серверная часть, состоящая из шести Docker-контейнеров, управляемых через Docker Compose. Реализованные программные модули обеспечивают корректное выполнение всех базовых операций, начиная от авторизации пользователей и заканчивая асинхронной отправкой сервисных уведомлений. Данная система была развернута на сервере университета.

Замеры производительности при умеренной нагрузке с помощью инструмента Postman [18] показали низкое время отклика для ключевых эндпоинтов, в среднем 13.8 мс, что превосходит заявленное в требованиях технического задания (менее 300 мс). Результаты первого этапа нагрузочного тестирования показаны в табл. 1.

Таблица 1. Результаты теста при нагрузке в 100 пользователей

Метрика	Значение
Пиковая производительность (RPS)	85 запросов/сек
Время отклика (95-й перцентиль)	240
Количество ошибок	0

На втором этапе нагрузочного тестирования была выявлена деградация производительности, связанная с исходной конфигурацией запуска FastAPI-сервисов в одном рабочем процессе. Данный дефект был устранен путем перехода на продакшен-конфигурацию с использованием Gunicorn [19] в качестве менеджера процессов. В итоге конфигурация была усилена путем добавления четырех асинхронных Uvicorn-воркеров для каждого сервиса [20]. Результаты тестирования показаны в табл. 2. Графики продемонстрированы на рис. 1.

Таблица 2. Результаты теста при нагрузке в 300 пользователей

Метрика	Значение
Пиковая производительность (RPS)	250 запросов/сек
Время отклика (95-й перцентиль)	450
Количество ошибок	0

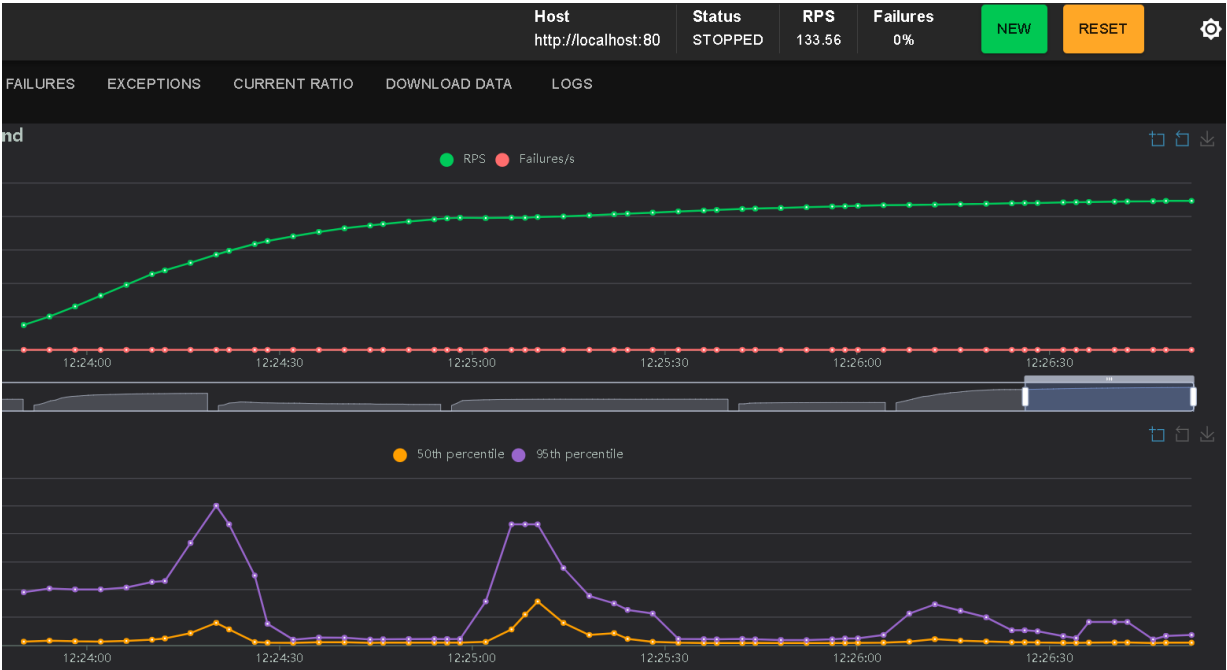


Рис. 1. Результаты нагрузочного тестирования

Обсуждение

Полученные результаты наглядно демонстрируют преимущества выбранного архитектурного подхода. Низкое время отклика при единичных запросах подтверждает высокую эффективность фреймворка FastAPI и асинхронного драйвера базы данных.

Наиболее важным результатом исследования является реакция системы на стресс-тест. Первоначальная деградация производительности и ее последующее успешное устранение путем горизонтального масштабирования на уровне процессов доказывают два ключевых положения. Во-первых, система ведет себя предсказуемо под нагрузкой. Во-вторых, микросервисная архитектура обеспечивает необходимую гибкость для точечной оптимизации производительности, что

было бы крайне затруднительно в монолитной системе, где пришлось бы масштабировать все приложение целиком.

Стабильная работа на уровне 250 запросов в секунду без ошибок подтверждает, что система обладает значительным запасом прочности и способна выдерживать пиковые нагрузки, многократно превышающие типовые для подобных проектов.

Заключение

В результате была успешно решена задача разработки и исследования серверной части веб-платформы для управления научными конференциями. По результатам тестирования можно сделать вывод, что применение микросервисной архитектуры в сочетании с современным асинхронным стеком технологий является эффективным решением для создания высокопроизводительных, отказоустойчивых и легко масштабируемых систем.

Данные, полученные в ходе нагрузочного тестирования, подтверждают преимущества микросервисного подхода. Разработанный программный продукт является готовым к внедрению, обладает значительным потенциалом для дальнейшего развития и масштабирования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ньюмен С. Создание микросервисов. – Санкт-Петербург: Питер, 2016. – 304 с.
2. Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга. – Санкт-Петербург: Питер, 2019. – 544 с.
3. Филдинг, Р. Т. Архитектурные стили и проектирование сетевых программных архитектур : диссертация на соискание степени доктора философии / Рой Томас Филдинг ; перевод с английского. – Ирвайн : Калифорнийский университет, 2000.
4. Фаулер М. Микросервисная архитектура //martinfowler.com. – 2014. – URL: <https://martinfowler.com/articles/microservices.html> (дата обращения: 15.05.2026).
5. Nginx: официальная документация / Nginx, Inc. – 2026. – Текст : электронный. – URL: <https://nginx.org/ru/docs> (дата обращения: 25.05.2026).
6. Ричардсон, Л. RESTful веб-API : [перевод с английского] / Леонард Ричардсон, Майк Амундсен. – Москва : О'Рейли, 2014. – 402 с. – Текст : непосредственный.
7. FastAPI: официальная документация. – URL: <https://fastapi.tiangolo.com/> (дата обращения: 20.05.2026).
8. Pydantic: официальная документация / S. Colvin. – 2024. – Текст : электронный. – URL: <https://docs.pydantic.dev/latest/> (дата обращения: 09.05.2026).
9. Рамальо, Л. Python. К вершинам мастерства : [перевод с английского] / Лучано Рамальо. – Москва : ДМК Пресс, 2021. – 768 с. – Текст : непосредственный.
10. asyncio – Asynchronous I/O : официальная документация Python 3.10 / Python Software Foundation. – 2024. – Текст : электронный. – URL: <https://docs.python.org/3.10/library/asyncio.html> (дата обращения: 11.05.2024).
11. Видник, А. RabbitMQ в действии. Распределенные приложения на основе очередей сообщений : [перевод с английского] / Альваро Видник, Джейсон Уильямс. – Москва : ДМК Пресс, 2014. – 354 с. – Текст : непосредственный.
12. RabbitMQ: официальная документация / Broadcom. – 2026. – Текст : электронный. – URL: rabbitmq.com (дата обращения: 25.05.2026). Nginx: официальная документация / F5, Inc. – 2024. – Текст : электронный. – URL: <http://nginx.org/ru/docs/> (дата обращения: 10.05.2024).

13. PostgreSQL 15: официальная документация / The PostgreSQL Global Development Group. – 2024. – Текст : электронный. – URL: <https://www.postgresql.org/docs/15/index.html> (дата обращения: 10.05.2024).
14. Тернбулл, Д. Docker: разработка и эксплуатация : [перевод с английского] / Джеймс Тернбулл, Карл Мэттиас. – Санкт-Петербург : Питер, 2017. – 336 с. – Текст : непосредственный.
15. Locust: официальная документация. – URL: <https://docs.locust.io/> (дата обращения: 25.05.2024).
16. Locust. Running Locust distributed: официальная документация / Locust Project. – 2026. – Текст : электронный. – URL: locust.io (дата обращения: 25.05.2026)
17. Бек, К. Экстремальное программирование. Разработка через тестирование : [перевод с английского] / Кент Бек. – Санкт-Петербург : Питер, 2003. – 224 с. – Текст : непосредственный.
18. Postman Learning Center: официальная документация / Postman, Inc. – 2026. – Текст : электронный. – URL: <https://learning.postman.com/docs> (дата обращения: 25.05.2026).
19. Gunicorn - Green Unicorn: официальная документация / В. Т. Jones. – 2024. – Текст : электронный. – URL: <https://gunicorn.org/> (дата обращения: 11.05.2024).
20. Uvicorn: официальная документация / Encode OSS Ltd. – 2026. – Текст : электронный. – URL: <https://uvicorn.dev> (дата обращения: 25.05.2026).

© А. Е. Коровянский, А. А. Шаранов, 2026

Н. А. Кравец¹✉, А. Ф. Лобода², Е. В. Шевчук¹

Мультиагентный мобильный помощник криминалиста «Фиксатор»

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация

²Высший строительно-экономический колледж, г. Петропавловск, Республика Казахстан
e-mail: fozien@ya.ru

Аннотация. В статье представлено Android-приложение «Мобильный помощник криминалиста «Фиксатор». «Фиксатор» автоматизирует ключевые этапы осмотра места происшествия. Архитектура приложения построена на мультиагентном подходе. Три специализированных программных агента последовательно обеспечивают контроль параметров фотофиксации, преобразование описания происшествия в официально-деловой стиль и формирование итогового PDF-отчета. Агент фотофиксации использует Google ML Kit для анализа углов Эйлера, центрирования и размера лица в кадре. Агент формализации взаимодействует с языковой моделью YandexGPT через API. Агент отчетности генерирует PDF-документ с сохранением через MediaStore API. Представлена архитектура системы, результаты функционального тестирования прототипа, рассмотрены варианты дальнейшего расширения функционала системы.

Ключевые слова: криминалистика, искусственный интеллект, мультиагентные системы, мобильные приложения

N. A. Kravets¹✉, A. F. Loboda², E. V. Shevchuk¹

Multi-agent mobile forensic assistant "Fixator"

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation

²Higher Construction and Economics College, Petropavlovsk, Republic of Kazakhstan
e-mail: fozien@ya.ru

Abstract. The article presents "Fixator" – a mobile forensic assistant implemented as an Android application that automates the key stages of crime scene examination. The application architecture is based on a multi-agent approach. Three specialized software agents sequentially handle photo quality control, formalization of incident descriptions into official documentary style, and generation of a final PDF report. The photo agent uses Google ML Kit to analyze Euler angles, face centering, and face size in the frame in real time. The formalization agent interacts with the YandexGPT language model via API. The reporting agent generates a PDF document saved via the MediaStore API. The article presents the system architecture, the results of functional prototype testing, and discusses directions for further expansion of the system.

Keywords: forensics, artificial intelligence, multi-agent systems, mobile applications

Введение

В последнее десятилетие криминалистика претерпела качественную трансформацию, перейдя от традиционных методов сбора улик к высокотехнологичному анализу больших данных. Внедрение систем искусственного интеллекта стало не просто «вспомогательным инструментом», а фундаментальным базисом, позволяющим рас-

крывать преступления. Наиболее очевидная сфера применения – системы распознавания лиц и биометрии.

Однако современные алгоритмы ушли далеко вперед от простого сличения фото с базой данных. Сегодня искусственный интеллект способен идентифицировать человека по походке, особенностям сосудистого рисунка кисти или даже по манере набора текста (клавиатурный почерк).

В условиях работы с низкокачественными записями с камер видеонаблюдения применяются нейросети-реконструкторы. Они «достраивают» недостающие пиксели, восстанавливая черты лица на основе обученных моделей человеческой анатомии, что критически важно для оперативного розыска.

В расследовании киберпреступлений и дел о терроризме ключевую роль играет семантический анализ. Искусственный интеллект способен анализировать тысячи сообщений в мессенджерах и интернете, выявляя скрытый жаргон, психологическое состояние автора и его социально-демографический портрет. Автоматизированное лингвистическое профилирование помогает сузить круг подозреваемых, анализируя уникальные речевые ошибки и идиомы, характерные для конкретного региона или социальной группы.

Протокол осмотра места происшествия – центральный процессуальный документ следственного действия. К формированию протокола осмотра необходимо подходить с учетом того, что в результате он будет играть важную роль доказательства в раскрытии и расследовании преступления. По этой причине в нем должна быть предельно наглядно и целостно раскрыта обстановка, чтобы по одному лишь приведенному в нем описанию можно было сформировать представление о произошедшем событии, восстановить факты, имеющие значение для дела.

На практике, однако, протоколы нередко оказываются неполными или неточными: сказывается недостаточный опыт молодых следователей, у большинства из которых стаж работы не превышает одного-двух лет [1]. Традиционное ручное документирование повышает риск ошибок и увеличивает время обработки данных непосредственно на месте событий.

Применение технологий искусственного интеллекта в криминалистической практике становится все более распространенным направлением исследований [2, 3]. В 2025 году авторами была опубликована первая версия мобильного помощника криминалиста, реализованная на базе библиотек ML Kit и CameraX с генерацией отчетов в формате DOCX [4]. Накопленный опыт эксплуатации прототипа выявил ряд архитектурных ограничений монолитного подхода и обозначил направление для дальнейшего развития системы.

Настоящая статья посвящена переработке системы на основе мультиагентного подхода.

Мультиагентные системы (МАС) – программные комплексы, в которых несколько автономных агентов с собственными базами знаний и правилами принятия решений совместно достигают общей цели [5, 6]. Применительно к задачам криминалистики такой подход позволяет декомпозировать сквозной рабочий цикл на независимые специализированные модули, повысить гибкость и расширяемость сис-

темы, а также реализовать адаптивный режим работы, учитывающий уровень подготовки пользователя [7].

Методы и материалы

Приложение реализовано на языке Kotlin [8] для платформы Android. В основу архитектуры положен мультиагентный подход: три программных агента взаимодействуют последовательно, каждый принимает на вход результат предыдущего и передает свой результат следующему.

Агент фотофиксации (CameraAgent) построен на библиотеках ML Kit [9] и CameraX [10]. В режиме реального времени агент анализирует видеопоток с камеры устройства: вычисляет углы Эйлера для определения ориентации головы в пространстве, оценивает размер лица относительно ширины кадра и его центрирование по горизонтали и вертикали. При отклонении любого параметра от допустимого диапазона на экране появляется конкретная текстовая подсказка. Захват снимка выполняется автоматически только при условии, что все параметры одновременно находятся в норме не менее двух секунд. Это исключает фиксацию кадров, не отвечающих требованиям опознавательной съемки [11].

Агент формализации (FormalizationAgent) взаимодействует с языковой моделью YandexGPT [12] через REST API. Пользователь вводит описание обстановки в произвольной форме; агент направляет текст модели с системной инструкцией сохранить все факты без добавления новых и привести изложение к нормам официально-делового стиля следственной документации. Результат выводится рядом с исходным текстом и недоступен для редактирования, что исключает непреднамеренные искажения.

Агент отчетности (ReportAgent) завершает рабочий цикл: консолидирует фотоматериал, формализованный текст и метаданные осмотра в единый PDF-документ. Для сохранения файла используется MediaStore API [13], что обеспечивает доступность документа в стандартной папке «Загрузки» без необходимости разрешений на запись во внешнее хранилище (актуально для Android 10 и выше).

Результаты

Разработка велась поэтапно в соответствии со структурой мультиагентной архитектуры. На первом этапе реализован пользовательский интерфейс. По сравнению с версией 2025 года интерфейс переработан: главное меню теперь выполнено в виде карточек с описанием каждого модуля, что снижает порог входа для новых пользователей (рис. 1).

На втором этапе реализован CameraAgent. Система оценивает три группы параметров кадра: ориентацию головы по осям Y (рыскание) и Z (крен) через углы Эйлера, размер лица и его центрирование. Каждое отклонение транслируется в конкретную подсказку на экране. Захват снимка блокируется до тех пор, пока все параметры не окажутся в норме одновременно (рис. 2). Опознавательная съемка выполняется в соответствии с установленными правилами криминалистики: анфас и в правый профиль [13].

На третьем этапе реализован FormalizationAgent. Пользователь вводит описание происшествия; система направляет текст в YandexGPT и отображает формализованный результат рядом с исходным (рис. 2). На четвертом этапе реализован ReportAgent: нажатие одной кнопки генерирует PDF-документ, который сохраняется в «Загрузки/Fixator» и немедленно открывается (рис. 3).

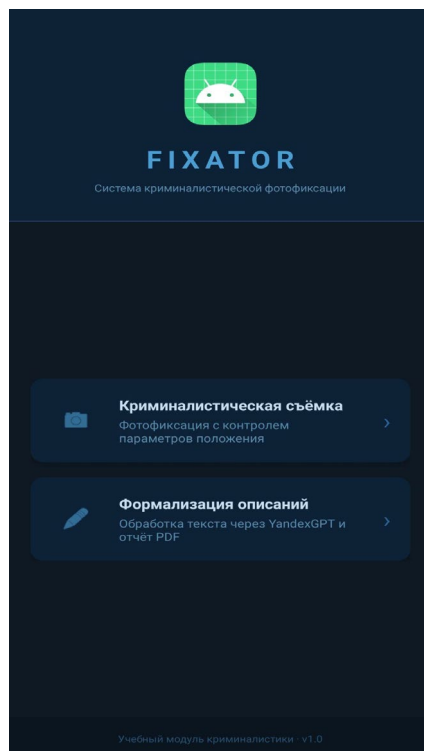


Рис. 1. Обновленное главное меню приложения «Фиксатор»

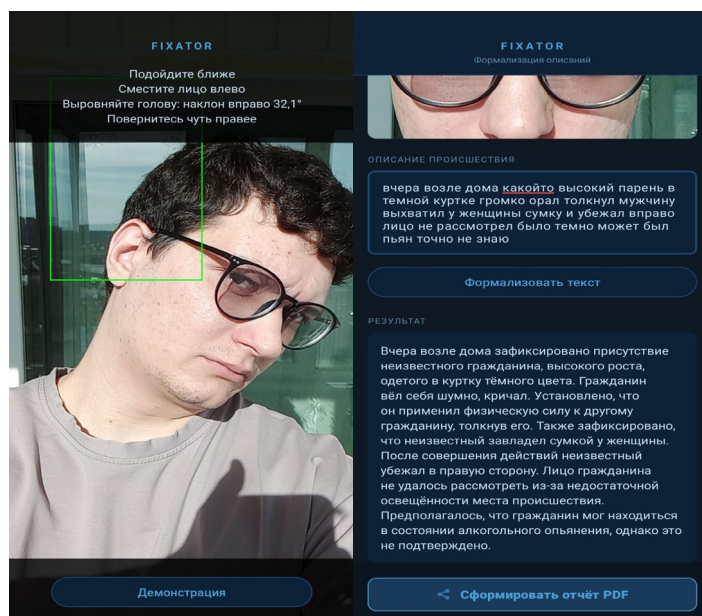


Рис. 2. Примеры работы «умной» камеры и формализации текста

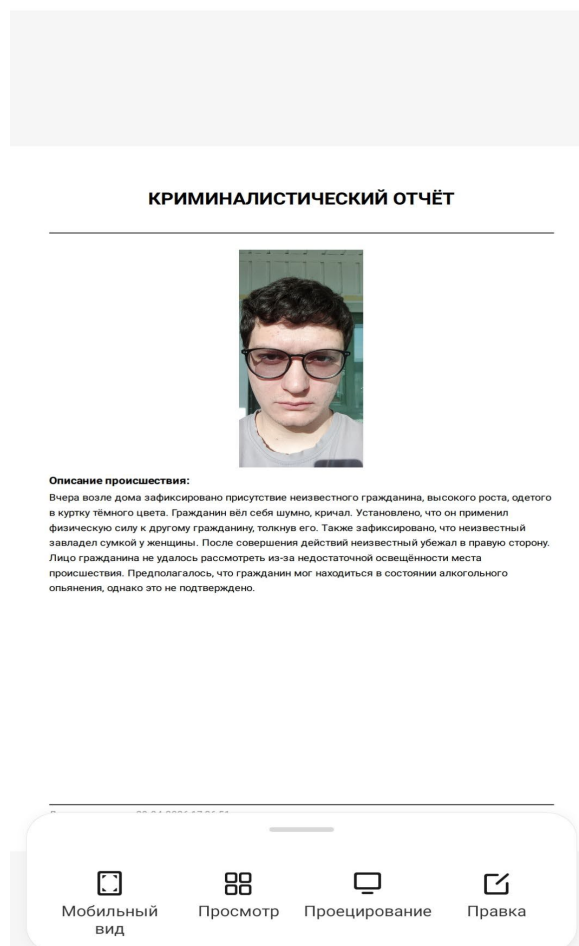


Рис. 3. Сгенерированный PDF-отчет

Обсуждение

Переход к мультиагентной архитектуре устранил основные ограничения предыдущей версии. Модульность позволяет обновлять и тестировать каждый агент независимо. Расширяемость обеспечивается самой природой MAC: подключение нового агента не требует переработки существующих [5, 6]. В перспективе возможно добавление агента контроля освещенности, агента распознавания вещественных объектов и агента проверки соответствия текста описания и фотографии.

Переход с DOCX на PDF и замена Apache POI на нативный Android PDF API с сохранением через MediaStore [12] устранили проблему недоступности файлов отчетов в системных папках устройства (ограничение Scoped Storage, введенное в Android 10). Теперь документ сразу доступен в стандартном файловом менеджере без дополнительных действий.

Заключение

Разработанная система «Фиксатор» демонстрирует практическую применимость мультиагентного подхода при автоматизации ключевых задач криминалистической деятельности. Три специализированных агента обеспечивают сквозной рабочий цикл от захвата снимка до формирования готового документа. По

сравнению с версией 2025 года система получила переработанный интерфейс, новую архитектуру на основе МАС и улучшенный механизм сохранения отчетов.

Система применима в двух сценариях: как инструмент поддержки следователя в полевых условиях и как учебный тренажер для студентов криминалистических специальностей. Дальнейшие исследования планируется направить на количественную оценку педагогической эффективности системы и расширение набора агентов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лобода А.Ф., Шевчук Е.В. Проблемы составления протокола осмотра места происшествия как источника доказательств и перспективы их решения посредством цифровизации // Матер. Междунар. науч.-практ. конф. «Актуальные проблемы современной юридической науки» (24 мая 2024 года). Петропавл: СКУ им. М. Козыбаева, 2024. С. 16–20.
2. Бахтеев Д.В. О связи криминалистики и технологии искусственного интеллекта // Сибирские уголовно-процессуальные и криминалистические чтения. 2022. № 1(35). URL: <https://cyberleninka.ru/article/n/o-svyazi-kriminalistiki-i-tehnologii-iskusstvennogo-intellekta> (дата обращения: 24.04.2025).
3. Ащук Д.Д., Соснин А.С. Применение искусственного интеллекта в криминалистике // Вестник науки. 2024. № 12(81). URL: <https://cyberleninka.ru/article/n/primenenie-iskusstvennogo-intellekta-v-kriminalistike> (дата обращения: 24.04.2025).
4. Кравец Н.А., Лобода А.Ф., Шевчук Е.В. Создание мобильного помощника криминалиста с использованием технологий искусственного интеллекта // Интерэкспо Гео-Сибирь. 2025. Т. 7, № 2. С. 155–161. EDN UCHMTZ.
5. Wooldridge M. An Introduction to MultiAgent Systems. 3rd ed. Wiley, 2019. – 484 p.
6. Кокаев А.Э., Сущеня Р.В. Области применения и платформы для разработки мультиагентных систем. Свойства мультиагентных систем // Вестник науки. 2023. № 7(64). URL: <https://cyberleninka.ru/article/n/oblasti-primeneniya-i-platformy-dlya-razrabotki-multiagentnyh-sistem-svoystva-multiagentnyh-sistem> (дата обращения: 14.04.2026).
7. Расулова Н.Ю. Мультиагентный подход в создании адаптивных интеллектуальных обучающих систем // Экономика и социум. 2021. № 2-2(81). URL: <https://cyberleninka.ru/article/n/multiagentnyy-podhod-v-sozdanii-adaptivnyh-intellektualnyh-obuchayushih-sistem> (дата обращения: 14.04.2026).
8. Kotlin Docs [Электронный ресурс] // Kotlin. URL: <https://kotlinlang.org/docs/home.html> (дата обращения: 21.04.2025).
9. Справочник по Android-комплекту ML [Электронный ресурс] // Google. URL: <https://developers.google.com/ml-kit/reference/android?hl=ru> (дата обращения: 21.04.2025).
10. CameraX Overview [Электронный ресурс] // Android Developers. URL: <https://developer.android.com/media/camera/cameras?hl=ru> (дата обращения: 22.04.2025).
11. Судебная запечатлевающая и исследовательская фотография: учеб.-практ. пособие / сост. Воронин В.В., Камелов А.В., Павличенко Г.В., Петров П.В., Раева А.С. Н. Новгород: Нижегородский государственный университет, 2018. – 100 с.
12. Особенности реализации API в сервисе Yandex Foundation Models [Электронный ресурс] // Yandex Cloud. URL: <https://yandex.cloud/ru/docs/foundation-models/concepts/api> (дата обращения: 23.04.2025).
13. Save media files with MediaStore [Электронный ресурс] // Android Developers. URL: <https://developer.android.com/training/data-storage/shared/media> (дата обращения: 25.04.2026).

© Н. А. Кравец, А. Ф. Лобода, Е. В. Шевчук, 2026

С. Е. Латыш¹✉, Д. Д. Лидер¹, А. А. Шаранов¹

Модернизация гексапода: увеличение степеней свободы и применение Jetson Nano

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: latysh@example.ru

Аннотация. В статье представлены результаты активной разработки модернизированной версии шестиногого робота-гексапода. Рассмотрены конструктивные особенности базовой модели с тремя степенями свободы на каждую ногу. Предложены пути модернизации, включающие увеличение числа степеней свободы до четырёх на ногу, замену сервоприводов на модели с вращением на 360°, интеграцию одноплатного компьютера Jetson Nano для задач компьютерного зрения и автономной навигации. Описана модульная конструкция верхней платформы для установки сменного целевого оборудования (видеокамер, датчиков, манипуляторов). Приведены результаты 3D-печати деталей и сборки текущего прототипа. Обсуждаются перспективы применения робота в спасательных операциях, геологоразведке и экологическом мониторинге.

Ключевые слова: гексапод, шестиногий робот, степени свободы, Jetson Nano, компьютерное зрение, модульная конструкция, автономная навигация

S. E. Latysh¹✉, D. D. Lider¹, A. A. Sharapov¹

Modernization of a hexapod: increasing degrees of freedom and using Jetson Nano

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: latysh@example.ru

Abstract. The article presents the results of the active development of a modernized version of a six-legged hexapod robot. The design features of the basic model with three degrees of freedom per leg are considered. Modernization paths are proposed, including increasing the number of degrees of freedom to four per leg, replacing servos with 360° rotation models, integrating a Jetson Nano single-board computer for computer vision and autonomous navigation tasks. The modular design of the upper platform for mounting interchangeable mission equipment (cameras, sensors, manipulators) is described. The results of 3D printing of parts and assembly of the current prototype are presented. The prospects for using the robot in rescue operations, geological exploration and environmental monitoring are discussed.

Keywords: hexapod, six-legged robot, degrees of freedom, Jetson Nano, computer vision, modular design, autonomous navigation

Введение

Шестиногие роботы (гексаподы) занимают особое место в современной робототехнике благодаря уникальному сочетанию статической устойчивости и высокой проходимости. В отличие от колесных и гусеничных платформ, гексапод способен преодолевать препятствия за счёт выборочной постановки ног на устойчивые точки,

что делает его незаменимым для работы в условиях завалов, горной местности, техногенных аварий [1]. Однако большинство существующих решений представляют собой либо дорогостоящие исследовательские платформы, либо любительские конструкторы с ограниченным функционалом. Актуальной задачей является создание доступной, ремонтпригодной и технологичной платформы, способной нести полезную нагрузку и выполнять автономные миссии.

Цель данной работы – модернизация классической конструкции гексапода с трёхстепенными ногами до версии с четырьмя степенями свободы (СО) на каждую конечность, интеграция высокопроизводительного вычислителя Jetson Nano для задач компьютерного зрения и реализация модульной системы крепления сменного оборудования. Работа находится на активной стадии разработки: выполнена 3D-печать основных элементов ног, ведётся сборка корпуса, разрабатывается программное обеспечение.

Исследование предметной области

В мире существует множество конструкций шестиногих роботов, которые можно разделить на три основные категории. Первая – высокобюджетные исследовательские платформы (например, Spot от Boston Dynamics, Laika от ANYbotics), обладающие совершенной кинематикой и системами управления, но имеющие стоимость от нескольких миллионов рублей и закрытую архитектуру. Вторая – любительские конструкторы на основе Arduino и стандартных сервоприводов (PhantomX, Lynxmotion), доступные по цене, но с низкой грузоподъемностью, примитивными походками и отсутствием автономной навигации. Третья – единичные open-source проекты энтузиастов, как правило, незавершённые или плохо документированные [1, 2].

При анализе российского рынка и научных публикаций выявлено, что большинство разработок ориентировано либо на лабораторные исследования походок, либо на демонстрацию базовых навыков ходьбы. Системы компьютерного зрения и автономной навигации в них либо отсутствуют, либо реализованы на маломощных контроллерах (Raspberry Pi 3/4), что приводит к низкой частоте обработки кадров и задержкам при принятии решений [3]. Также недостаточно проработана модульность: смена целевой нагрузки (например, замена видеодомофона на газоанализатор) требует переделки корпуса.

В многочисленных исследованиях отмечается, что трёхступенная кинематика ноги является минимально необходимой для перемещения по пересечённой местности, но четырёхступенная конструкция обеспечивает существенно большую манёвренность, позволяя вращать конечность в горизонтальной плоскости без разворота всего робота [4]. Поэтому, в рамках данной работы сначала создается рабочий прототип с тремя степенями свободы для отладки базовых алгоритмов, затем переход на четырёхступенную версию.

Кроме того, анализ потребностей потенциальных заказчиков, легших в основу тех. задания для последующей доработки конструкции показал, что основными требованиями выступили такие немаловажные параметры, как совмещение возможности дистанционного управления и автономного движения по заданному

маршруту, включая распознавание людей, также дешевизна и простота ремонта в полевых условиях.

Методы и материалы

На данный момент базовая конструкция гексапода, взятая за основу, имеет шесть ног, каждая из которых обладает тремя степенями свободы вращения суставов в пределах 180° , позволяющая использовать систему передвижения, при которой робот будет в любом положении опираться на три точки опоры, образуя устойчивый треножник, в то время как остальные три ноги переносятся вперед, делая шаг. Однако данная конструкция накладывает некоторые ограничения, связанные с преодолением сложных траекторий движения конечностей и маневренностью на пересеченной местности.

Для модернизации конструкции были предусмотрены следующие технические решения:

1. Увеличение количества степеней свободы, при которых на каждую ногу добавляется четвёртый привод, обеспечивающий вращение в горизонтальной плоскости, перпендикулярно корпусу в месте крепления, делая из ноги четырехзвенный манипулятор, что позволяет реализовать более сложные походки (например, волнообразную или метахрональную), а также поможет при передвижении на равнине за счет достижения конструкции колесной формы для увеличения скорости перемещения. Сами сервоприводы также не останутся без внимания, вместо стандартных с углом поворота до 180° планируется использовать сервоприводы с непрерывным вращением на все 360° , что даст возможность каждой ноге вращаться без ограничений, расширяя возможности расчета обратной кинематики, а следовательно, и диапазон достижимых положений стопы относительно корпуса.

2. Интеграция одноплатного компьютера Jetson Nano. Выбор обусловлен необходимостью выполнения ресурсоемких задач компьютерного зрения в реальном времени, одноплатный компьютер содержит встроенный графический процессор с 128 ядрами CUDA, что позволит запускать сверточные и предикативные нейронные сети для распознавания объектов, сегментации препятствий и последующего реагирования на них, также построения карты методом SLAM [2] Планируется использование камеры глубины и инерциального измерительного модуля для точной оценки положения робота в пространстве.

3. Модульная конструкция верхней платформы. для модификаций спектра возможностей робота для специфических задач в различных отраслях. Будет разработана универсальная система крепления на верхней части корпуса, а также сами модули в зависимости от пожеланий заказчика. Она выполнена в виде сетки отверстий с шагом 20 мм и позволяет устанавливать сменные модули: тепловизионные камеры, газоанализаторы, дозиметры, лёгкие манипуляторы для взятия проб, а также транспортные контейнеры для доставки медикаментов или воды. Замена модуля занимает не более одной минуты.

Материалы изготовления: корпус и основные элементы ног выполнены методом 3D-печати из пластика PETG, обладающего хорошей ударной вязкостью

и теплостойкостью. В местах наибольших механических нагрузок предусмотрены алюминиевые вставки. Габаритный диаметр робота в сборе – 35 см, масса без сменного модуля – около 4 кг. Питание осуществляется от литий-ионной аккумуляторной сборки, обеспечивающей до 2 часов активного движения. Коммуникация с оператором – по радиоканалу с дальностью до 2 км на открытой местности.

Программное обеспечение базируется на операционной системе ROS (Robot Operating System) с использованием библиотек OpenCV, TensorFlow Lite и GTSAM для оптимизации графов поз. Управление приводами реализовано через микроконтроллер ESP32, принимающий команды от Jetson Nano по UART.

Результаты

На текущий момент (активная стадия разработки) достигнуты следующие результаты:

- Полностью спроектирована 3D-модель модернизированной конструкции с четырьмя степенями свободы на ногу. Выполнен расчёт кинематики для четырёхзвенной конечности.
- Изготовлены и проходят сборку все элементы ног для базовой трёхступенной версии (рис. 1). Это позволит отладить навигационное ПО до завершения печати деталей для четырёхступенного варианта.
- Начата печать усиленного корпуса с посадочными местами под Jetson Nano и модульную платформу.
- Подготовлен стенд для тестирования алгоритмов компьютерного зрения на Jetson Nano: реализовано детектирование человека по видео с камеры в реальном времени с частотой 15 кадров/с.
- Разработана схема электропитания с отдельными шинами для сервоприводов (высокий ток) и вычислителя (5 В).



Рис. 1. Напечатанные на 3D-принтере детали ног в процессе сборки

В табл. 1 приведены сравнительные характеристики базовой и модернизированной версий.

Таблица 1.

Параметр	Базовая версия	Модернизированная версия
Количество степеней свободы	3	4
Тип сервоприводов	180°	360°
Вычислитель	Arduino Mega	Jetson Nano
Наличие компьютерного зрения	Нет	Да
Сменные модули	Нет	Да
Ориентировочная максимальная скорость, км/ч	3	12

Планируемая максимальная скорость модернизированной версии возрастёт за счёт более эффективного управления шагом благодаря четвёртой степени свободы.

Обсуждение

Увеличение числа степеней свободы с трёх до четырёх принципиально меняет кинематические возможности гексапода. Трёхступенная нога позволяет перемещать стопу только в пределах конуса, вершина которого находится в точке крепления бедра. Четырёхступенная нога добавляет вращение вокруг горизонтальной оси, что позволяет достичь точек, расположенных сверху корпуса робота и даже зеркальной смены плоскости передвижения без потери маневренности, при повороте робота, к примеру, также при движении в узких проходах или между препятствиями разной высоты, как и вышеупомянутого движения, используя конструкцию колеса.

Сравнение с аналогичными разработками [3, 4] показывает, что большинство любительских гексаподов используют именно три степени свободы из-за простоты управления и реализации, хотя для прикладных задач, требующих высокой маневренности, например, инспекция трубопроводов или разведка под завалами, четырёхступенная кинематика является более предпочтительной. Интеграция Jetson Nano, а не Arduino Mega, как во многих проектах, обусловлена необходимостью обработки видеопотока в реальном времени без чрезмерного количества задержек.

Модульная конструкция верхней платформы создает возможность быстрой смены целевой нагрузки, дабы потенциальные заказчики, включая спасателей, геологов, экологов, могли использовать весь потенциал гексапода: сегодня робот должен нести тепловизор, завтра – газоанализатор. Также, разработанная система крепления позволяет оператору в полевых условиях заменить модуль без специального инструмента.

В ходе обсуждения с потенциальными пользователями были выявлены требования к простоте управления и надёжности. В ответ на это планируется реализовать полуавтономный режим, при котором оператор задаёт только конечную точку маршрута, а робот самостоятельно строит путь с обходом препятствий на основе данных с камер.

Заключение

В результате работы спроектирована модернизированная версия шестиногого робота-гексапода, отличающаяся от классических конструкций увеличением числа степеней свободы до четырёх на каждую ногу, применением сервоприводов с вращением на 360° , интеграцией вычислителя Jetson Nano и модульной платформой для сменного оборудования. Разработка находится на активной стадии: изготовлены и собраны элементы ног, ведётся печать корпуса, отлаживаются алгоритмы компьютерного зрения. Ожидаемое время завершения полнофункционального прототипа – два месяца.

Дальнейшие перспективы включают:

- установку лёгкого манипулятора для взятия проб грунта;
- реализацию голосового управления при добавлении микрофона и системы распознавания команд;
- переход на отечественные микроэлектронные компоненты там, где это экономически и технически оправдано;
- проведение полевых испытаний совместно с региональным управлением МЧС.

Разработанная платформа может найти применение в спасательных работах (разведка в зонах завалов и задымления), геологоразведке на сложном рельефе, экологическом мониторинге труднодоступных территорий. Благодаря модульности и открытой архитектуре возможна дальнейшая адаптация под конкретные задачи заказчика.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Карпенко, А. П. Шагающие роботы: кинематика, динамика, управление / А. П. Карпенко, В. Б. Лапшин. — Москва: Наука, 2017. — 320 с.
2. Jetson Nano: встраиваемые вычисления для искусственного интеллекта / Д. С. Кузнецов, А. В. Петров // Робототехника и техническая кибернетика. — 2022. — № 3. — С. 45-52.
3. Белоногов, О. Б. Разработка шестиногого робота на базе сервоприводов и микроконтроллера Arduino / О. Б. Белоногов, И. А. Смирнов // Информационные технологии в науке и образовании. — 2020. — № 4. — С. 112-118.
4. Филимонов, Н. Б. Алгоритмы стабилизации походки гексапода при движении по неровной поверхности / Н. Б. Филимонов, Е. В. Козлов // Мехатроника, автоматизация, управление. — 2019. — Т. 20, № 7. — С. 415-421.

© С. Е. Латыш, Д. Д. Лидер, А. А. Шаранов, 2026

П. А. Лашин¹✉, Н. С. Головачев¹

Разработка интерактивного 3D-тура по исторической усадьбе Строгоновых с элементами реконструкции исторических объектов

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: pavel2003.lashin@gmail.com

Аннотация. В статье представлены основные результаты разработки интерактивного 3D-тура по исторической усадьбе Строгоновых с элементами реконструкции исторических объектов. Выполнен анализ предметной области, в котором рассмотрено различные среды разработки и моделирования, а также история усадьбы Строгоновых. Для создания интерактивного 3D-тура были использованы среда моделирования ArchiCad, среда разработки Unreal Engine 5 и визуальная система программирования Blueprints. Интерактивный 3D-тур предоставляет пользователю возможность изучить историю усадьбы Строгоновых, а также визуально познакомиться с усадьбой.

Ключевые слова: интерактивный 3D-тур, ArchiCad, Unreal Engine 5, Blueprints, усадьба Строгоновых

P. A. Lashin¹✉, N. S. Golovachev¹

Development of an interactive 3D tour of the historic Strogonov estate with elements of reconstruction of historical sites

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: pavel2003.lashin@gmail.com

Abstract. The article presents the main results of the development of an interactive 3D tour of the historic Strogonov estate with elements of reconstruction of historical sites. An analysis of the subject area has been performed, which examines various development and modeling environments, as well as the history of the Strogonov estate. The ArchiCAD modeling environment, the Unreal Engine 5 development environment, and the Blueprints visual programming system were used to create an interactive 3D tour. The interactive 3D tour provides the user with the opportunity to explore the history of the Strogonov estate, as well as visually explore the estate.

Keywords: interactive 3D tour, ArchiCAD, Unreal Engine 5, Blueprints, Strogonov manor

Введение

Усадьба Строгоновых – уникальная достопримечательность, которая отражает эстетику и образ жизни русского провинциального дворянства. Сегодня сооружение находится в полуразрушенном состоянии, и здание 3D-модели позволит сохранить память о нем и передать в будущее чтобы, у следующих поколений было представление о культуре прошлого [1].

Реконструкция исторического здания в виртуальном пространстве позволит подкрепить интерес туристов и исследователей, что откроет возможность виртуальных путешествий и познавательных мероприятий. Таким образом, проект

может помочь привлечь внимание общества к данному региону, стимулируя развитие туризма.

Так же проект может помочь в процессе изучения и сохранения культурно-исторического наследия связаны с развитием образования. Цифровое воссозданием объекта позволит глубже изучить архитектуру, художественные приемы и технику строительства того времени, что послужит хорошим примером для дальнейших научно-исторических изысканий.

Историко-культурное наследие – неотъемлемый источник знаний о прошлом страны. Современное общество заинтересованно в важности сохранения и восстановления утраченных объектов истории и культуры. Особенно это касается архитектурных ансамблей, таких как усадьба Строгоновых, которые показывают дух той эпохи и несут значительный культурный смысл.

Главной задачей этого проекта является создание высокоточной 3D-модели усадьбы Строгоновых, а также создание виртуальной среды. Модель должна передавать все характеристики и внешние черты оригинального строения, включая архитектуру.

Среда моделирования

В качестве среды моделирования использовался программный продукт Archicad, который предоставляет инструменты для создания стен, перекрытий, оконных и дверных проемов, кровель и других архитектурных элементов. Это делает программу удобной для разработки моделей исторических зданий, поскольку основные формы сооружений могут быть воспроизведены с достаточно высокой геометрической точностью [2, 3].

К преимуществам Archicad относятся поддержка работы в операционной системе Windows, широкий набор инструментов для архитектурного моделирования, удобная система навигации и понятный интерфейс. Кроме того, программа позволяет точно задавать параметры геометрических объектов, что важно при создании моделей зданий усадебного комплекса.

К ограничениям программы можно отнести высокую стоимость лицензии, а также снижение скорости рендеринга при работе со сложными и детализированными сценами. Поэтому в рамках данной работы Archicad использовался преимущественно для построения архитектурных моделей, а дальнейшая визуализация и организация интерактивного взаимодействия выполнялись в среде Unreal Engine 5.

Среда разработки

В качестве программной среды для реализации интерактивного 3D-тура была выбрана среда разработки Unreal Engine 5, разработанный компанией Epic Games. Данная платформа ориентирована на создание интерактивных приложений реального времени и широко применяется не только в игровой индустрии, но и в архитектурной визуализации, виртуальной реконструкции, образовательных проектах и системах виртуальной реальности [4].

Для реализации интерактивной логики в проекте использовалась система Blueprints – визуальная среда программирования, встроенная в Unreal Engine. Она позволяет создавать сценарии взаимодействия пользователя с виртуальной средой без необходимости написания программного кода на C++. С помощью Blueprints могут быть реализованы элементы навигации, главное меню, информационные панели, интерактивные объекты, а также различные события, возникающие при перемещении пользователя по сцене [5].

Отдельное значение имеет поддержка мультиплатформенной разработки. Unreal Engine 5 позволяет адаптировать создаваемое приложение для запуска на персональных компьютерах, мобильных устройствах, VR-оборудовании и других целевых платформах.

Результаты

После выбора среды моделирования и среды разработки начался основной этап создания интерактивного 3D-тура. Сначала были подготовлены трехмерные модели зданий усадебного комплекса, включая главный дворец. При моделировании учитывались общие пропорции сооружений, их расположение и внешний архитектурный облик (рис. 1).

Далее модели были перенесены в среду Unreal Engine 5, где была сформирована единая виртуальная сцена (рис. 2). На этом этапе были выполнены следующие действия:

- создана среда с рельефом местности;
- размещены модели зданий на территории усадьбы;
- добавлены элементы растительности и ландшафтного окружения;
- настроена возможность перемещения пользователя по сцене;
- разработан предварительный вариант главного меню;
- добавлены информационные таблички с историческими сведениями об объектах усадьбы.

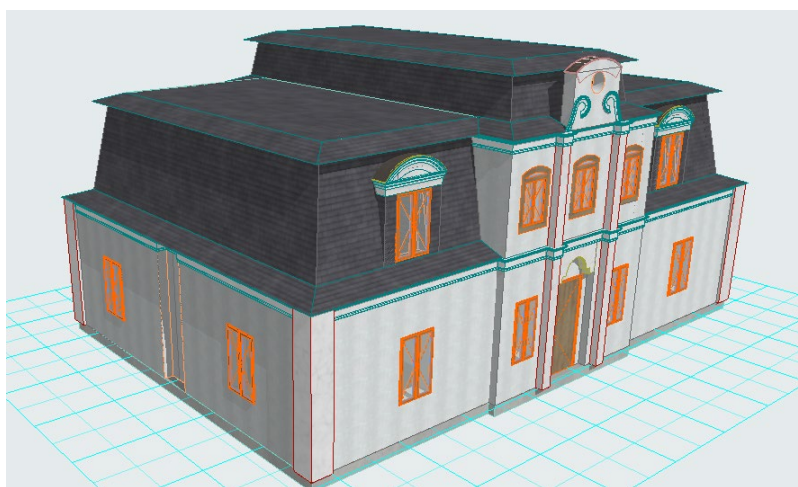


Рис. 1. Пример созданной в ArchiCAD 3D модели одного из зданий усадьбы

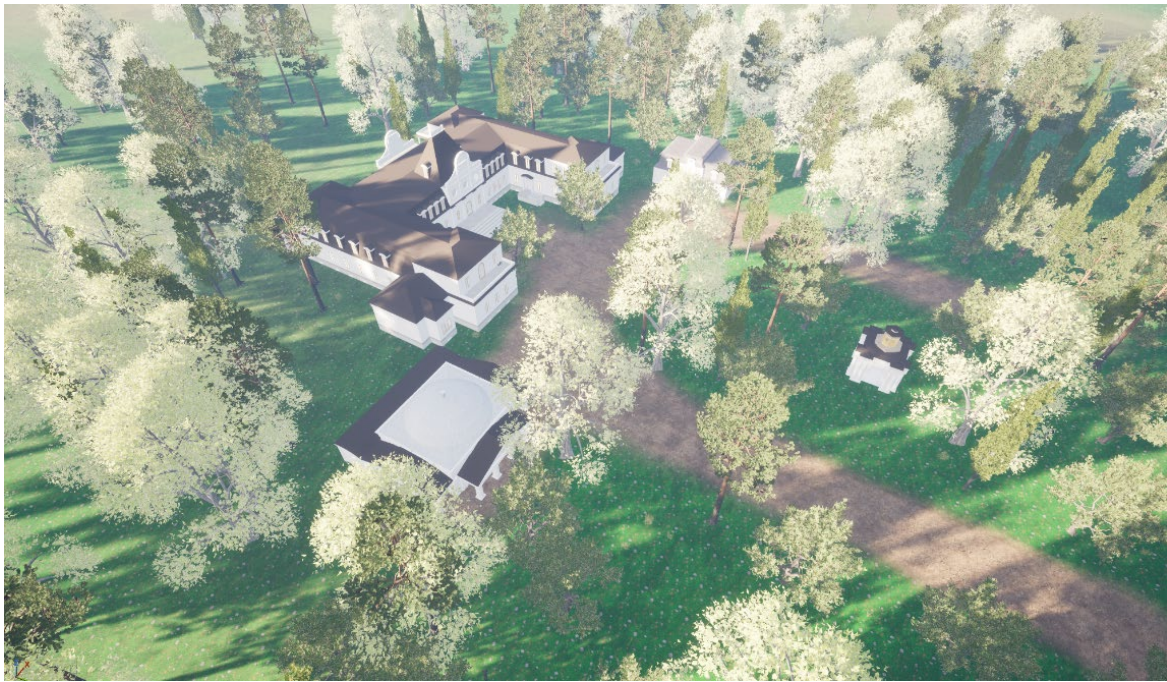


Рис. 2. Рельеф местности с моделями зданий и растительности

После размещения основных объектов была выполнена первичная настройка пользовательского взаимодействия. Пользователь получил возможность перемещаться по виртуальной территории, осматривать здания и знакомиться с краткой исторической информацией, размещенной в виде текстовых табличек (рис. 3).

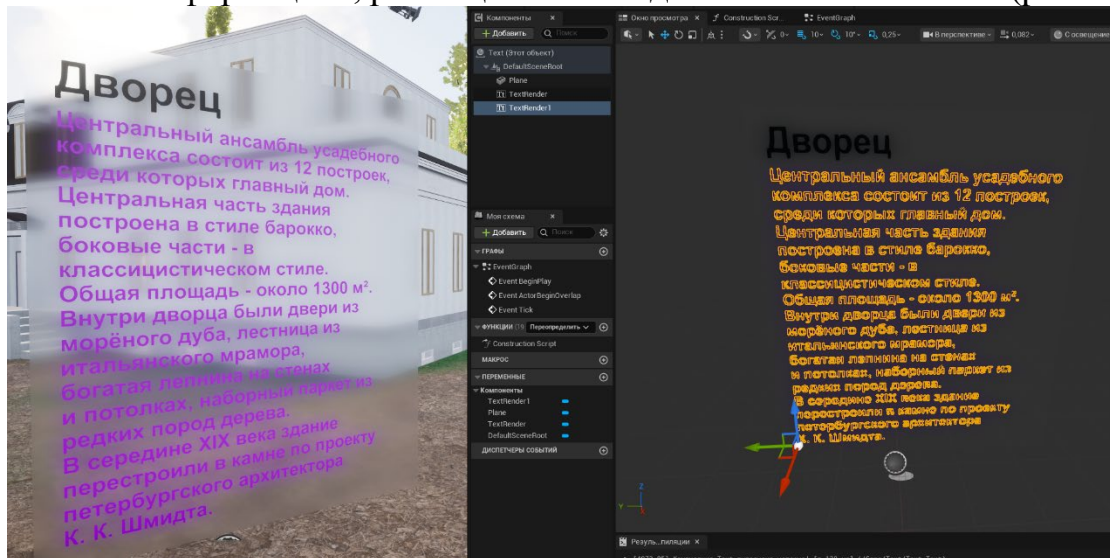


Рис. 3. Пример создания информационной таблички

На данном этапе проект представляет собой рабочую сцену (рис. 4), в которой объединены архитектурные модели, элементы окружения и базовые интерактивные функции.



Рис. 4. Фрагмент виртуальной сцены

Дальнейшая работа в рамках разработки связана с уточнением отдельных элементов сцены: корректировкой расположения объектов, настройкой материалов и освещения, улучшением интерфейса и расширением сценариев взаимодействия пользователя с виртуальной средой [6].

Заключение

В результате выполнения работы был создан прототип интерактивного 3D-тура по усадьбе Строгоновых. В состав проекта вошли трехмерные модели основных зданий усадебного комплекса, виртуальная среда с рельефом местности, элементы растительности, главное меню и информационные таблички с историческими сведениями.

Разработанный прототип позволяет пользователю перемещаться по территории усадьбы в виртуальной среде, осматривать архитектурные объекты и получать краткую информацию об истории усадебного комплекса. Это дает возможность использовать проект как средство визуального представления историко-архитектурного объекта.

Дальнейшее развитие проекта связано с повышением детализации моделей, улучшением качества материалов и освещения, расширением интерактивных функций и адаптацией тура для использования с технологиями виртуальной реальности. После доработки проект может применяться в образовательных, туристических и культурно-просветительских целях.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ягофарова Надежда Олеговна История усадьбы Строгановых в Волышово как уникального памятника культурного наследия Российской провинции // Культурное наследие России. 2024. № 4 (47). С. 99–103.
2. Archicad // Graphisoft : официальный сайт. URL: <https://www.graphisoft.com/en-us/plans-and-products/archicad/> (дата обращения: 23.05.2026).
3. System Requirements for Graphisoft Solutions // Graphisoft: официальный сайт. URL: <https://www.graphisoft.com/en-us/plans-and-products/system-requirements/> (дата обращения: 13.05.2026).

4. Unreal Engine 5.6 Documentation // Epic Developer Community. URL: <https://dev.epicgames.com/documentation/unreal-engine/unreal-engine-5-6-documentation> (дата обращения: 13.05.2026).

5. Blueprints Visual Scripting in Unreal Engine // Epic Developer Community. URL: <https://dev.epicgames.com/documentation/unreal-engine/blueprints-visual-scripting-in-unreal-engine> (дата обращения: 13.05.2026).

6. Rodriguez-Garcia B. A systematic review of virtual 3D reconstructions of Cultural Heritage in immersive Virtual Reality / B. Rodriguez-Garcia, H. Guillen-Sanz, D. Checa [et al.] // *Multimedia Tools and Applications*. 2024. Vol. 83. P. 89743–89793. DOI: 10.1007/s11042-024-18700-3.

© П. А. Лашин, Н. С. Головачев, 2026

Р. С. Лонский¹✉, А. А. Шаронов¹

Разработка мобильного приложения для информационной поддержки машиниста электровоза на платформе Android

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: bah4ezzmzk@yandex.ru

Аннотация. В статье описана разработка мобильного приложения «Ассистент машиниста» для операционной системы Android. Во время работы изучались особенности деятельности локомотивной бригады и причины, влияющие на безопасность движения поездов. Также был проведен анализ существующих средств информирования машиниста. Рассматривались их сильные и слабые стороны. В результате было создано мобильное приложение для отображения профиля пути, определения положения поезда и предупреждения о приближении к ограничениям скорости и объектам инфраструктуры. Разработка выполнялась в среде Android Studio с использованием языка Kotlin. В приложении реализована обработка GPS/ГЛОНАСС-координат. Добавлены система предупреждений, километровая шкала и анимация движения профиля пути. Испытания показали стабильную работу основных функций. Результаты подтвердили возможность применения приложения как вспомогательного средства информирования машиниста.

Ключевые слова: мобильное приложение, Android, Kotlin, GPS, ГЛОНАСС, железнодорожный транспорт

R. S. Lonskiy¹✉, A. A. Sharapov¹

Development of a mobile application for information support for electric locomotive drivers on the Android platform

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: bah4ezzmzk@yandex.ru

Abstract. The article describes the development of the “Driver Assistant” mobile application for the Android operating system. The study examined the specifics of locomotive crew work and the factors affecting railway traffic safety. Existing driver information systems were also analyzed. Their advantages and disadvantages were reviewed. As a result, a mobile application was created for displaying the railway track profile, determining the train position and warning about approaching speed restrictions and infrastructure objects. The application was developed in Android Studio using the Kotlin programming language. GPS/GLONASS coordinate processing, a warning system, kilometer scale visualization and animated movement of the track profile were implemented. Testing confirmed stable operation of the main functions. The results showed that the application can be used as an auxiliary driver information tool.

Keywords: mobile application, Android, Kotlin, GPS, GLONASS, railway transport

Введение

Работа машиниста электровоза считается одной из самых ответственных на железной дороге. Во время движения локомотивная бригада постоянно контролирует

сигналы светофоров, скорость и положение поезда. Также необходимо учитывать профиль пути и особенности маршрута. Из-за высокой нагрузки возрастает вероятность ошибок. Особенно при длительных поездках и интенсивном движении.

На железной дороге используются стационарные системы безопасности и бортовые комплексы контроля. Но большинство таких решений направлено именно на обеспечение безопасности движения. Возможности отображения профиля пути у них ограничены. Режимные карты тоже требуют постоянного сопоставления информации с фактическим местоположением поезда. Это увеличивает нагрузку на локомотивную бригаду.

Сейчас мобильные устройства все чаще используются как вспомогательное средство информирования. GPS и ГЛОНАСС позволяют определять положение поезда. А современные смартфоны обладают достаточной производительностью для отображения маршрута и предупреждений.

Для Android было разработано приложение «Ассистент машиниста». Оно показывает профиль пути, положение поезда и заранее сообщает о приближении к ограничениям скорости.

Анализ предметной области

На движение поезда влияет много факторов. Один из основных – состояние пути. Также важны исправность локомотива и корректная работа сигнализации [1, 2]. Не меньшее значение имеет профиль пути. Машинисту приходится учитывать подъемы, спуски, кривые и другие особенности участка.

Во время движения нужно постоянно контролировать ограничения скорости. Они могут появляться из-за ремонта пути или состояния инфраструктуры. Иногда ограничения вводятся на потенциально опасных участках.

При высокой нагрузке возрастает вероятность ошибки. Особенно если машинисту одновременно приходится следить сразу за несколькими источниками информации. Даже опытный машинист может ошибиться при высокой нагрузке или усталости. Особенно когда приходится одновременно контролировать несколько источников информации. Работа с режимными картами тоже требует постоянного сравнения километража и фактического положения поезда [3].

Существующие решения можно разделить на несколько групп:

- стационарные системы безопасности;
- бортовые комплексы контроля;
- традиционные режимные карты;
- мобильные программные решения.

Стационарные и бортовые системы отличаются высокой надежностью. Но полноценного отображения профиля пути они обычно не обеспечивают. Традиционные средства информирования требуют ручной обработки информации. Мобильные приложения более гибкие. Они позволяют удобно отображать маршрут и предупреждения.

Проведенный анализ показал, что мобильное приложение может упростить навигацию и сделать восприятие информации удобнее во время движения поезда.

Методы и технологии

Для разработки приложения была выбрана операционная система Android. Это связано с широким распространением Android-устройств и встроенной поддержкой GPS и ГЛОНАСС. Еще одним преимуществом стала поддержка фоновых сервисов. Они позволяют обрабатывать координаты в реальном времени [4, 5].

Разработка выполнялась в среде Android Studio. Среда содержит инструменты для проектирования интерфейса, написания кода и тестирования приложения [6].

В качестве языка программирования использовался Kotlin. Язык поддерживается компанией Google и рекомендуется для современной Android-разработки [7, 8]. Kotlin позволяет писать более компактный код, чем Java. Также язык поддерживает механизм null-безопасности. Это снижает вероятность ошибок во время работы приложения [9].

Для обработки координат применялись стандартные Android API. Работа с навигационными данными включает получение GPS/ГЛОНАСС-координат, фильтрацию помех и определение текущего километра пути [10].

Архитектура приложения включает несколько основных компонентов:

- пользовательский интерфейс;
- модуль обработки координат;
- система предупреждений;
- модуль хранения данных.

Пользовательский интерфейс отвечает за отображение профиля пути, километровой шкалы и объектов инфраструктуры. Модуль обработки координат вычисляет текущее положение поезда. Система предупреждений формирует звуковые и визуальные сигналы при приближении к ограничениям скорости.

Для хранения пользовательских настроек используется SharedPreferences. Воспроизведение звуков реализовано через MediaPlayer. Для плавного перемещения элементов интерфейса применяется ValueAnimator.

Результаты

Во время работы было создано мобильное приложение «Ассистент машиниста» для Android.

В приложении реализованы следующие функции:

- учет направления движения поезда (рис. 1, а);
- отображение профиля пути;
- отображение текущего положения поезда (рис. 1, б).;
- обработка GPS/ГЛОНАСС-координат;
- отображение километровой шкалы;
- ввод ограничений скорости (рис. 1, в);
- формирование звуковых и визуальных предупреждений;

Поезд в приложении всегда отображается в центральной части экрана. Во время движения перемещается профиль пути и километровая шкала. Такой под-

ход упрощает восприятие информации. Пользователь может сосредоточить внимание на центральной части интерфейса.

Программа получает координаты устройства и определяет ближайший участок маршрута. После этого вычисляется текущее положение поезда. При приближении к ограничениям скорости или объектам инфраструктуры пользователь получает предупреждение.

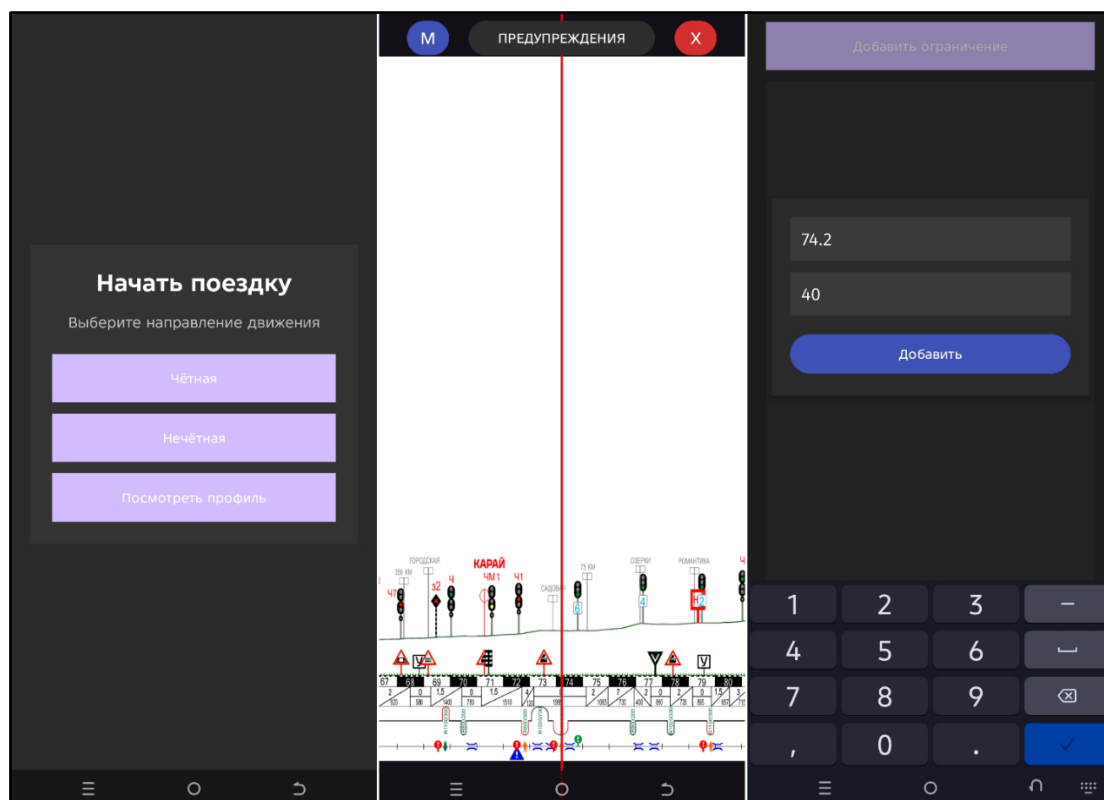


Рис. 1. Интерфейс мобильного приложения

- а) экран выбора направления движения; б) основной экран;
в) экран ввода ограничений

Во время разработки отдельно проверялась точность определения координат. Также тестировалась корректность отображения положения поезда и работа системы предупреждений. Дополнительно оценивалась устойчивость интерфейса при длительной обработке навигационных данных.

Во время тестирования серьезных проблем в работе приложения не возникало. Проверялись разные варианты использования. Основные функции при этом работали без сбоев.

Заключение

Итогом работы стало приложение «Ассистент машиниста» для устройств на Android. Программа создавалась как дополнительный инструмент для локомотивной бригады.

Во время разработки пришлось изучить особенности работы машиниста, профиль пути и существующие способы информирования. После анализа стало понятно, что мобильное приложение может сделать работу с маршрутной информацией удобнее.

Приложение обрабатывает GPS/ГЛОНАСС-координаты, показывает маршрут и предупреждает о приближении к ограничениям скорости. Для использования нужен только смартфон на Android.

Во время проверки приложение работало стабильно. Ошибок, которые мешали бы работе основных функций, обнаружено не было.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Басыров Р. М. Системы обеспечения безопасности движения поездов. – Москва : Транспорт, 2019. – 318 с.
2. Ковалев В. И. Железнодорожная автоматика и телемеханика. – Санкт-Петербург : Лань, 2021. – 412 с.
3. Гундорин Н. П. Организация работы локомотивных бригад. – Москва : Маршрут, 2018. – 276 с.
4. Роджерс Р. Android. Разработка приложений / Р. Роджерс. – Москва : Эком, 2010. – 400 с.
5. Android Developers. – URL: <https://developer.android.com> (дата обращения: 12.05.2026).
6. Android Studio. – URL: <https://developer.android.com/studio> (дата обращения: 12.05.2026).
7. Жемеров Д. Kotlin в действии / Д. Жемеров, С. Исакова. – Москва : ДМК Пресс, 2018. – 402 с.
8. Kotlin Documentation. – URL: <https://kotlinlang.org/docs/home.html> (дата обращения: 12.05.2026).
9. Пьер-Ив С. Волшебство Kotlin / С. Пьер-Ив. – Москва : ДМК Пресс, 2020. – 536 с.
10. Соловьев А. В. Навигационные системы GPS и ГЛОНАСС. – Новосибирск : СГУГиТ, 2020. – 244 с.

© Р. С. Лонский, А. А. Шарапов, 2026

Е. К. Макаров¹✉, С. Ю. Кацко¹

Кроссплатформенное приложение для тренировки скорочтения

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: makaroff.12@yandex.ru

Аннотация. В статье рассмотрена разработка кроссплатформенного приложения для тренировки скорочтения. Актуальность темы связана с ростом объема текстовой информации и потребностью в инструменте, который помогает развивать скорость чтения без потери понимания. В качестве проблемы выделены ограниченная персонализация существующих тренажеров, слабая детализация статистики и отсутствие единой истории прогресса на разных устройствах. Цель работы состоит в создании приложения на C# с использованием .NET MAUI, которое поддерживает регистрацию пользователя, выполнение упражнений, сохранение результатов и отображение аналитики. В ходе разработки были применены анализ предметной области, сравнение аналогов, проектирование клиент-серверной архитектуры, моделирование базы данных и функциональное тестирование. Получено программное решение для Android и Windows с упражнениями на скорость восприятия слов, внимание, память и расширение поля зрения. Результаты тестирования показали корректную работу основных сценариев: регистрации, входа, выбора упражнения, сохранения результата, просмотра статистики и профиля пользователя.

Ключевые слова: скорочтение, кроссплатформенное приложение, .NET MAUI, C#, статистика, адаптивность

Е. К. Makarov¹✉, S. Yu. Katsko¹

Cross-platform application for speed reading training

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: makaroff.12@yandex.ru

Abstract. The article describes the development of a cross-platform application for speed reading training. The topic is relevant because modern users work with a growing amount of textual information and need tools that improve reading speed while preserving comprehension. The main problem is that many existing trainers provide a limited level of personalization, weak progress analytics and no unified training history across devices. The purpose of the work is to create an application in C# using .NET MAUI that supports user registration, training exercises, result storage and progress analytics. The work included domain analysis, comparison of similar applications, client-server architecture design, database modeling and functional testing. The resulting application supports Android and Windows, including exercises for word perception speed, attention, memory and field of view expansion. Testing confirmed correct operation of the main scenarios: registration, authorization, exercise selection, result saving, statistics viewing and user profile display.

Keywords: speed reading, cross-platform application, .NET MAUI, C#, statistics, adaptability

Введение

С увеличением объема информации все более заметной становится потребность в быстром и осмысленном чтении. Скорочтение нельзя сводить только к

росту количества прочитанных слов в минуту, потому что главным условием остается понимание текста. На качество чтения влияют сложность материала, уровень подготовки, концентрация внимания, объем зрительного восприятия, субвокализация, частота фиксаций взгляда и возвраты глаз к уже прочитанным фрагментам [1]. Поэтому тренировка скорочтения должна быть направлена не только на темп, но и на развитие внимания, памяти, поля зрения и способности быстро выделять смысловые блоки.

В настоящее время существуют мобильные приложения для тренировки скорочтения, однако многие из них ограничиваются набором типовых упражнений. В таких решениях пользователь часто получает одинаковые задания независимо от текущего уровня подготовки, количества ошибок и динамики результатов. Отдельной проблемой остается недостаточная детализация статистики: приложение может показывать лучший результат или общий балл, но не объяснять, какие навыки улучшаются, а какие требуют дополнительной работы. Для персонализированного обучения важны скорость выполнения заданий, количество ошибок, стабильность результатов и история занятий [2].

В качестве основы для разработки было выбрано создание кроссплатформенного приложения для тренировки скорочтения на C# [3] с использованием .NET MAUI [4]. Такой подход позволяет поддерживать Android и Windows в рамках единой кодовой базы. Актуальность разработки связана с тем, что пользователь может тренироваться на мобильном устройстве и на персональном компьютере, а результаты должны сохраняться в единой системе. В работе было проанализировано скорочтение как навык, рассмотрены существующие цифровые тренажеры и выделены требования к приложению, которое объединяет упражнения, профиль, статистику и адаптацию сложности.

Цель разработки состоит в создании приложения, обеспечивающего выполнение тренировочных упражнений, сохранение результатов пользователя, отображение аналитики прогресса и поддержку дальнейшей персонализации. Для достижения цели было выполнено сравнение аналогов, сформулированы функциональные и нефункциональные требования, спроектирована клиент-серверная архитектура, выбраны технологии реализации и проведена проверка основных сценариев работы приложения. В предметной области уже отмечается интерес к цифровым средствам развития навыков чтения [5, 6], а адаптивное обучение рассматривается как один из способов повышения вовлеченности и практической пользы занятий [7].

Методы и материалы

На первом этапе были выделены функциональные требования к приложению. Система должна обеспечивать создание учетной записи, вход пользователя, выбор упражнения, выполнение разных форматов заданий, сохранение результата, просмотр статистики и отображение профиля. Такой набор функций связан с основной логикой тренажера: пользователь проходит упражнения, получает результат, а затем видит динамику выполнения. Формализация требований важна

для разработки информационных систем, потому что позволяет связать ожидания пользователя с будущими компонентами приложения [8].

Также были определены нефункциональные требования. Приложение должно работать на Android и Windows, иметь понятный интерфейс, сохранять пользовательские данные, ограничивать доступ к личной статистике только авторизованным пользователям, стабильно выполнять упражнения и обрабатывать основные действия без заметных задержек. Структура приложения должна допускать расширение: добавление новых упражнений, показателей статистики и элементов аналитики. Учет таких требований особенно важен для ИТ-архитектуры, где качество системы определяется не только набором функций, но и удобством, надежностью и возможностью развития [9].

Архитектура приложения построена по клиент-серверной многоуровневой модели. Клиентская часть отвечает за интерфейс, запуск упражнений, обработку действий пользователя, отображение результатов и вывод статистики. Серверная часть реализована на основе ASP.NET Core Web API [10] и включает слой API, слой прикладной логики и слой доступа к данным. Для связи с базой данных применяется Entity Framework Core [11]. Такое разделение снижает связанность компонентов и позволяет независимо дорабатывать интерфейс, логику обработки результатов и хранение данных.

Для хранения информации выбрана PostgreSQL [12]. В базе данных предусмотрены таблицы пользователей, общих результатов тренировок, прогресса по упражнениям и отдельных результатов для разных типов заданий. Такая структура нужна потому, что упражнения имеют разные показатели: время выполнения, количество ошибок, скорость отображения слов, успешность прохождения и достигнутый уровень. Централизованное хранение данных позволяет сохранять историю тренировок и использовать ее для отображения аналитики.

Выбор целевых платформ связан с распространенностью Android на мобильных устройствах и Windows на персональных компьютерах [13]. В качестве среды разработки использована Microsoft Visual Studio [14], платформа .NET [15], язык C#, .NET MAUI и XAML [16]. Этот набор технологий позволяет описывать интерфейс в XAML, реализовывать логику на C#, создавать клиентскую часть для нескольких платформ и разрабатывать серверную часть в той же технологической среде.

Результаты

В ходе разработки было создано приложение для тренировки скорочтения, включающее регистрацию, авторизацию, главный экран, раздел упражнений, статистику и профиль пользователя. Регистрация проверяет корректность введенных данных и доступность учетной записи. После входа пользователь попадает на главный экран, где отображаются текущий статус, количество очков за день, число завершенных сессий и краткая подсказка по тренировкам. Основные экраны приложения представлены на рис. 1.

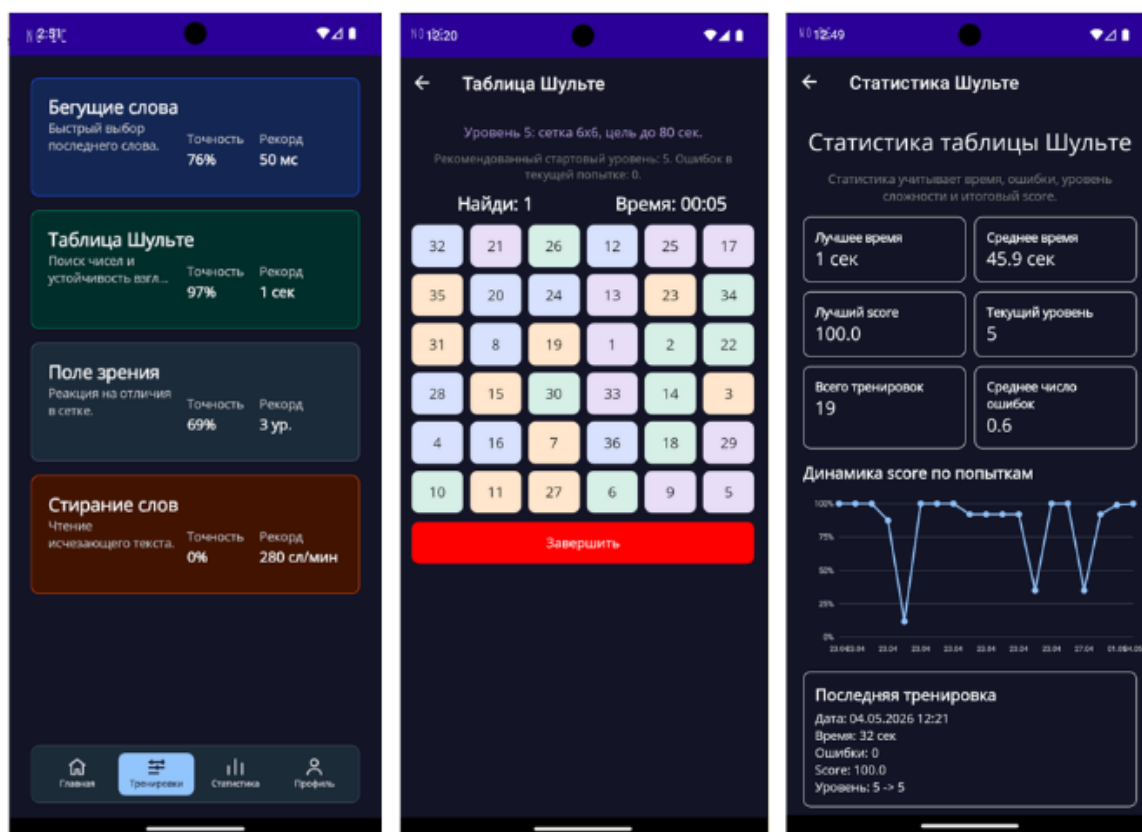


Рис. 1. Интерфейс приложения для тренировки скорочтения

Раздел упражнений содержит несколько форматов заданий. В приложении предусмотрены упражнения «Бегущие слова», «Таблица Шульте», «Поле зрения» и «Затирание слов». В карточке каждого упражнения отображаются название, краткое описание и ключевой показатель результата. Для «Бегущих слов» фиксируется скорость отображения, для «Таблицы Шульте» – лучшее время прохождения, для «Поля зрения» – достигнутый уровень, для «Затирания слов» – скорость чтения в словах в минуту. Такой подход делает тренировочный процесс разнообразным и связывает каждое упражнение с конкретным навыком.

Страница статистики показывает результаты по выбранному тренировочному режиму. Она не смешивает все показатели в один общий балл, а дает возможность рассмотреть отдельное упражнение. Профиль пользователя объединяет общий балл, очки за текущий день, число сессий, количество выполненных упражнений, состояние обучения и краткую аналитику. В профиле отображаются сильная сторона, проблемная зона и подсказка по следующей тренировке. Это позволяет сделать приложение не просто набором экранов, а инструментом для отслеживания личного прогресса.

Работоспособность приложения была проверена с помощью функционального тестирования. Проверялись сценарии, напрямую связанные с требованиями: создание учетной записи, вход, выбор упражнения, выполнение заданий, сохранение результата, просмотр статистики и отображение профиля. Результаты представлены в табл. 1.

Таблица 1. Результаты функционального тестирования приложения

№	Проверяемая функция	Ожидаемый результат	Итог
1	Создание учетной записи	Учетная запись создается, выводится сообщение об успешной регистрации	Да
2	Вход пользователя	Пользователь получает доступ к основным функциям приложения	Да
3	Выбор упражнения	Открывается выбранный тренировочный режим	Да
4	Выполнение упражнений	Задания разных типов запускаются и завершаются корректно	Да
5	Сохранение результата	Результат сохраняется после завершения упражнения	Да
6	Просмотр статистики	Отображаются результаты ранее выполненных упражнений	Да
7	Просмотр профиля	Отображаются профиль, общий прогресс, история и подсказки	Да

По итогам проверки все основные сценарии были выполнены успешно. Это подтверждает, что реализованные функции соответствуют заявленным требованиям и могут использоваться как основа для дальнейшего развития приложения. Особенно важным результатом является связка между выполнением упражнения и сохранением статистики: пользователь получает не только факт завершения задания, но и данные для анализа прогресса.

Полученное приложение поддерживает разные форматы тренировок и дает возможность рассматривать развитие навыка по отдельным показателям. За счет этого приложение отличается от простого набора упражнений: данные пользователя сохраняются, группируются по типам заданий и отображаются в статистике и профиле.

Обсуждение

Сравнение с существующими приложениями показывает, что основное отличие разработки связано с более внимательным отношением к пользовательским данным. В рассмотренных аналогах статистика часто сводится к лучшему результату, графику или суммарному количеству очков. Такой способ удобен для быстрого просмотра, но он не всегда показывает стабильность навыка и причины изменения результата. В созданном приложении прогресс рассматривается через отдельные упражнения, историю тренировок, успешность выполнения и состояние профиля. При оценке таких приложений важно учитывать, что резкий рост скорости чтения без потери понимания имеет естественные ограничения: в работах по скорочтению описан компромисс между темпом чтения и качеством понимания текста [17]. Отдельно подчеркивается, что практическая польза тренировки связана не с обещанием мгновенного увеличения скорости, а с осознанным выбором темпа и сохранением нужного уровня понимания [18].

Кроссплатформенность также имеет практическое значение. Пользователь может выполнять короткие тренировки на мобильном устройстве, а более длительную работу проводить на компьютере. При централизованном хранении данных такая схема не разбивает прогресс на отдельные устройства. Это особенно важно для регулярных занятий, потому что навык скорочтения формируется постепенно и

требует накопления результатов во времени. Для мобильного обучения значимыми остаются ожидания пользователя, доступность функций и удобство регулярного применения приложения [19]. Систематические обзоры также показывают, что мобильные образовательные приложения дают лучший эффект, когда они не ограничиваются подачей материала, а поддерживают самостоятельную работу и обратную связь [20].

Разработка не исчерпывает все возможности персонализации. На текущем этапе реализована основа для адаптации: сохранение результатов, учет ошибок, отображение слабых и сильных сторон, вывод подсказки по тренировке. Следующим логичным направлением может стать более точная настройка сложности упражнений по статистике пользователя: изменению скорости, количеству ошибок, стабильности прохождения и регулярности занятий. Такая доработка позволит сделать тренировочный процесс еще более индивидуальным. В этом смысле статистика приложения может рассматриваться как часть аналитики обучения: данные о действиях пользователя помогают точнее связывать структуру заданий, обратную связь и ожидаемые результаты [21].

Заключение

В работе было рассмотрено создание кроссплатформенного приложения для тренировки скорочтения. На основе анализа предметной области и существующих программных решений были выделены ограничения аналогов: слабая персонализация, неполная статистика, ограниченная роль профиля и недостаточная поддержка разных платформ. Эти ограничения были учтены при проектировании приложения.

Была спроектирована клиент-серверная архитектура, в которой клиентская часть на .NET MAUI отвечает за интерфейс и выполнение упражнений, а серверная часть на ASP.NET Core Web API обеспечивает авторизацию, обработку данных и сохранение результатов в PostgreSQL. В приложении реализованы регистрация, вход, выбор упражнений, выполнение тренировочных заданий, сохранение результатов, просмотр статистики и профиль пользователя.

Функциональное тестирование подтвердило корректную работу основных сценариев. В итоге была получена программная основа тренажера скорочтения, которая может применяться для самостоятельных занятий и дальнейшего расширения. Главная практическая ценность разработки состоит в объединении упражнений, статистики и профиля пользователя в одной системе, где прогресс можно отслеживать не по одному итоговому числу, а по нескольким понятным показателям.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Eye movements, the perceptual span, and reading speed // National Library of Medicine. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC3075059/> (дата обращения: 02.04.2026).
2. Персонализация в образовании: от программируемого к адаптивному обучению // PsyJournals. URL: https://psyjournals.ru/journals/jmfp/archive/2020_n3/Kravchenko_et_al (дата обращения: 19.04.2026).
3. C# language documentation // Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата обращения: 07.04.2026).

4. What is .NET MAUI? // Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-10.0> (дата обращения: 01.04.2026).
5. Тачаева А. Е., Санжарова И. Э., Тухтарова Н. А. Медленное чтение в быстрый век // Гуманитарные исследования. История и филология. 2024. № 16. URL: <https://cyberleninka.ru/article/n/medlennoe-chtenie-v-bystryy-vek> (дата обращения: 13.05.2026).
6. Минин В. А. Обзор существующего программного обеспечения для развития когнитивных способностей на примере скорочтения // Colloquium-journal. 2024. № 6 (199). URL: <https://cyberleninka.ru/article/n/obzor-suschestvuyushchego-programmnogo-obespecheniya-dlya-razvitiya-kognitivnyh-sposobnostey-na-primere-skorochteniya> (дата обращения: 13.05.2026).
7. Personalized adaptive learning in higher education: A scoping review of key characteristics and impact on academic performance and engagement // National Library of Medicine. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11544060/> (дата обращения: 02.04.2026).
8. Гутгарц Р. Д., Провилков Е. И. О формализации функциональных требований в проектах по созданию информационных систем // Программные продукты и системы. 2019. № 3. URL: <https://cyberleninka.ru/article/n/o-formalizatsii-funktsionalnyh-trebovaniy-v-proektah-posozdaniyu-informatsionnyh-sistem> (дата обращения: 13.05.2026).
9. Чеглаков А. Л., Нехотина В. С. Оценка нефункциональных требований ИТ-архитектуры с использованием онтологий // НАУ. 2015. № 4-3 (9). URL: <https://cyberleninka.ru/article/n/otsenka-nefunktsionalnyh-trebovaniy-it-arhitektury-s-ispolzovaniem-ontologiy> (дата обращения: 13.05.2026).
10. Create web APIs with ASP.NET Core // Microsoft Learn. URL: <https://learn.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-10.0> (дата обращения: 01.04.2026).
11. Entity Framework Core // Microsoft Learn. URL: <https://learn.microsoft.com/en-us/ef/core/> (дата обращения: 05.04.2026).
12. Documentation // PostgreSQL. URL: <https://www.postgresql.org/docs/> (дата обращения: 10.04.2026).
13. Mobile Operating System Market Share Worldwide // StatCounter. URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202005-202604-bar> (дата обращения: 30.04.2026).
14. Visual Studio 2026 // Microsoft. URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 08.04.2026).
15. Сборка. Тестирование. Развертывание // Microsoft .NET. URL: <https://dotnet.microsoft.com/ru-ru/> (дата обращения: 08.04.2026).
16. XAML // Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/maui/xaml/?view=net-maui-10.0> (дата обращения: 05.04.2026).
17. Rayner K., Schotter E. R., Masson M. E. J., Potter M. C., Treiman R. So much to read, so little time: How do we read, and can speed reading help? // Psychological Science in the Public Interest. - 2016. - Vol. 17, № 1. - P. 4-34. - DOI: 10.1177/1529100615623267.
18. Balota D. A. Speed reading: You cannot always get what you want, but can you sometimes get what you need? // Psychological Science in the Public Interest. - 2016. - Vol. 17, № 1. - P. 1-3. - DOI: 10.1177/1529100615623268.
19. Alrasheedi M., Capretz L. F., Raza A. A systematic review of the critical factors for success of mobile learning in higher education (university students perspective) // Journal of Educational Computing Research. - 2015. - Vol. 52, № 2. - P. 257-276. - DOI: 10.1177/0735633115571928.
20. Naveed Q. N., Choudhary H., Ahmad N., Alqahtani J., Qahmash A. I. Mobile learning in higher education: A systematic literature review // Sustainability. - 2023. - Vol. 15, № 18. - Article 13566. - DOI: 10.3390/su151813566.
21. Blumenstein M. Synergies of learning analytics and learning design: A systematic review of student outcomes // Journal of Learning Analytics. - 2020. - Vol. 7, № 3. - P. 13-32. - DOI: 10.18608/jla.2020.73.3.

Д. Д. Малыхин^{1✉}, А. А. Шаранов¹

Разработка веб-ресурса по предоставлению сантехнических услуг для индивидуального предпринимателя

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: malihin55555@gmail.com

Аннотация. В статье представлены основные результаты разработки веб-ресурса для индивидуального предпринимателя (ИП), специализирующегося на сантехнических услугах. В первую очередь, был проведен тщательный анализ предметной области. В рамках этого анализа исследовались уже существующие сайты, предлагающие аналогичные услуги. Это позволило выявить как сильные, так и слабые стороны конкурентов, а также определить, какие элементы дизайна и функционала наиболее эффективны для привлечения клиентов. Разработка велась на базе CMS WordPress с кастомизацией клиентской части посредством HTML/CSS/PHP и созданием уникального функционального модуля калькуляции, который предлагает пользователю выбрать тип услуги, ввести свой округ и получить мгновенный расчет. Этот интерактивный элемент не только повышает вовлеченность пользователя, но и снижает часть нагрузки с исполнителя, предоставляя клиентам предварительную информацию. Веб-ресурс предоставляет пользователю возможность заказа услуг, ознакомление с уже выполненными заказами, возможность приблизительно рассчитать стоимость.

Ключевые слова: веб-ресурс, разработка, услуга, анализ, пользователь

D. D. Malykhin^{1✉}, A. A. Sharapov¹

Development of a web resource for the provision of plumbing services for individual entrepreneurs

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: malihin55555@gmail.com

Abstract. This article presents the key results of developing a web resource for a sole proprietor specializing in plumbing services. First, a thorough analysis of the subject area was conducted. This analysis included a review of existing websites offering similar services. This allowed us to identify both the strengths and weaknesses of competitors and determine which design and functional elements are most effective in attracting customers. Development was conducted using the WordPress CMS, with customization of the client side using HTML/CSS/PHP and the creation of a unique, functional calculation module that allows users to select a service type, enter their district, and receive an instant quote. This interactive element not only increases user engagement but also alleviates some of the burden on the service provider by providing clients with preliminary information. The web resource allows users to order services, view completed orders, and calculate approximate costs.

Keywords: web resource, development, service, analysis, user

Введение

В условиях цифровой трансформации рынка бытовых услуг традиционные методы поиска клиентов (бумажные объявления, «сарафанное радио») теряют эффективность. Современный потребитель предпочитает заказывать услуги через интер-

нет, ожидая оперативной связи и прозрачного ценообразования. Для индивидуальных предпринимателей в сфере сантехники наличие собственного веб-ресурса становится обязательным условием конкурентоспособности.

Существующие решения, такие как агрегаторы (например, Profi.ru), предполагают высокие комиссии и ограничивают прямую связь с клиентом. Конструкторы сайтов не всегда позволяют гибко настраивать специфические функции, а индивидуальная разработка «с нуля» требует значительных финансовых вложений. Целью данной работы является создание доступного, легко масштабируемого и функционального веб-ресурса, решающего задачи автоматизации приема заявок и расчета стоимости работ.

Исследование предметной области

Современный рынок услуг характеризуется высокой конкуренцией, особенно в сфере бытового обслуживания, к которому относятся и сантехнические работы. Индивидуальные предприниматели (ИП) представляют собой большую группу участников этого рынка и решают определенные задачи:

Привлечение клиентов: для поиска заказчиков ИП используют цифровые инструменты, так как печатные издания и объявления становятся менее эффективными [1, 2].

Формирование доверия: в сфере услуг, клиенты не могут проверить качество работы заранее. По этой причине ИП важно открыто показывать информацию о себе и иметь хорошие отзывы от других людей [3].

Управление заявками: когда предприниматель принимает звонки и сообщения, он тратит на это время и отвлекается от выполнения заказов [4]. Если использовать автоматические системы для обработки заявок, человек работает быстрее и распределяет свое время более эффективно.

При изучении цифровых платформ становится понятно, что они предлагают разные функции для ИП:

Агрегаторы услуг (например, Profi.ru, YouDo) дают возможность находить многих заказчиков. Но эти сервисы забирают часть денег в виде комиссии, из-за чего доход предпринимателя становится меньше. Также эти системы устанавливают свои правила и часто запрещают общаться с клиентами напрямую [5, 6].

Конструкторы сайтов (Tilda, Wix): позволяют быстро создать простой сайт, но часто обладают ограниченным функционалом для специфических бизнес-задач (например, сложность интеграции динамических калькуляторов или специализированных форм). Стоимость подписки также может расти с увеличением функционала [7, 8].

Индивидуальная разработка: обеспечивает максимальную гибкость, но требует существенных финансовых вложений и длительного времени на разработку, что зачастую недоступно для малого бизнеса [9, 10].

Методы и технологии

Разработку веб-ресурса было решено вести на платформе WordPress. Данная среда разработки бесплатно предоставляет полный набор инструментов и функций,

упрощающий процесс разработки и повышающий производительность разработчиков.

К преимуществам WordPress можно отнести следующие моменты [11–14]:

- для этой системы характерна высокая популярность, также функционируют форумы поддержки, разработчики объединяются в сообщества;
- Эта система работает гибко, с помощью плагинов специалисты создают сайты разной сложности, темы оформления расширяют возможности платформы, эти инструменты подходят для выполнения задач из первого раздела;
- для конечного пользователя управление сайтом является простым процессом, сотрудники без технических знаний обновляют информацию на страницах, они добавляют новости через административный интерфейс, там же они размещают фотографии, интерфейс понятен на интуитивном уровне.

Для разработки были выбраны HTML/CSS/PHP – типизированные языки программирования. Эти три технологии являются фундаментом современной веб-разработки, работая в тесном взаимодействии для создания полноценных веб-сайтов и приложений [15, 16].

К преимуществам языка программирования PHP можно отнести следующие аспекты [17–21]:

- PHP имеет относительно низкий порог вхождения. Из-за того, что синтаксис имеет сходство с C, Java и Perl, его осваивают быстро;
- На различных операционных системах, таких как Windows, Linux и macOS, PHP функционирует стабильно, его можно запускать на веб-серверах Apache, Nginx или IIS. Среда для разработки и развертывания выбирается свободно благодаря таким характеристикам;
- В новых версиях скорость работы выше, чем в старых. По этой причине язык пригоден для проектов, которые имеют большое количество запросов;
- С помощью PHP код вставляется прямо в HTML. И этот язык обрабатывает данные в форматах XML, JSON и CSV.

Таким образом, для разработки веб-ресурса был использован следующий стек технологий:

- язык программирования PHP 8.2.30 на базе WordPress версии 6.9.4;
- HTML;
- CSS.

Результаты

В результате работы был создан веб-ресурс, в котором реализованы следующие части:

- модуль заказа услуг содержит поля, где указываются имя, телефон, адрес, вид услуги и текст комментария. При помощи плагина WP Mail SMTP система отправляет заявки на электронную почту исполнителя без сбоев;
- в модуле «Отзывы» пользователи оставляют свои мнения, которые видны на сайте, система публикует его автоматически;

– для расчета стоимости работ написан отдельный PHP-модуль. Данные из массивов PHP поступают в JavaScript через механизм `wp_localize_script`. Таким образом расчеты происходят мгновенно, и пользователю не нужно обновлять страницу. Последующее после ввода данных добавление их в статистику, которая содержит отдельные таблицы для каждого типа вводимых данных, обновляемые автоматически после каждого ввода (рис. 1).

Навигационная структура сайта спроектирована таким образом, чтобы минимизировать количество кликов до целевого действия (заказа). Меню дублируется в шапке и подвале («Услуги», «Калькулятор», «Портфолио», «Контакты», «Отзывы»).

Калькулятор расчёта стоимости работы

Округ:
Свердловский округ

Услуга:
Установка и ремонт сантехнических приборов

Посчитать

Результат расчета:

Вы выбрали: Свердловский округ Установка и ремонт сантехнических приборов.

Минимальная цена : 5100 рублей.

Максимальная цена : 20400 рублей.

Страницы
Заказать услугу
Калькулятор примерной стоимости работ
Контакты

Рис. 1. Интерфейс модуля калькуляции

Заключение

В ходе выполнения работы был создан полнофункциональный веб-ресурс для индивидуального предпринимателя. Разработанный программный продукт позволяет: формировать положительный имидж исполнителя за счет качественного портфолио, автоматизировать процесс получения заявок, исключая ручную обработку сообщений, предоставлять клиентам инструмент предварительной оценки стоимости (калькулятор), что повышает доверие и снижает количество уточняющих звонков. Использование открытой CMS WordPress позволило предоставить заказчику удобный инструмент для самостоятельного управления контентом.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Эволюция маркетинга: от газетных объявлений до искусственного интеллекта. – URL: <https://sky.pro/wiki/profession/evolyuciya-marketinga-ot-tradicionnogo-k-cifrovomu/> (дата обращения: 14.05.2026).
2. Что такое реклама простыми словами: виды, задачи, цели. – URL: <https://kokoc.com/blog/reklama/> (дата обращения: 14.05.2026).

3. Доверие: принципы и практики*. – URL: https://sberuniversity.ru/upload/uf/36f/g0wa7cpbi21q9epxzinn8otqzpaub4y/SK_plus-Doverie-_13_-s-razvorotom.pdf (дата обращения: 15.05.2026).
4. Коммуникация с клиентами: методы, принципы, инструменты. – URL: <https://sales-generator.ru/blog/kommunikatsiya-s-klientami/> (дата обращения: 15.05.2026).
5. Агрегаторы услуг Profi.ru и YouDo: стоит ли там продвигаться или лучше вложиться в свой сайт. – URL: <https://xn--80adbv1ag.xn--80aswg/blog/agregatory-uslug-profi-youdo> (дата обращения: 17.05.2026).
6. Profi.ru, YouDo и другие: обзор площадок для заказчиков и исполнителей. – URL: <https://bankstoday.net/last-articles/profi-ru-youdo-i-drugie-obzor-ploshhadok-dlya-zakazchikov-i-ispolnitelej> (дата обращения: 17.05.2026).
7. Преимущества и недостатки сайтов-конструкторов Tilda и Wix: стоит ли выбирать готовые решения для бизнеса?. – URL: <https://www.sostav.ru/blogs/266689/57275> (дата обращения: 18.05.2026).
8. No-code платформы Tilda, Wix и ко - плюсы и минусы. – URL: <https://bp-oblako.ru/blog/no-code-platformy-tilda-wix-i-ko-plyusy-i-minusy/> (дата обращения: 18.05.2026).
9. Разработка сайта с нуля: 7 преимуществ против 5 недостатков. – URL: <https://sky.pro/wiki/html/razrabotka-sajta-s-nulya-preimushhestva-i-nedostatki/> (дата обращения: 18.05.2026).
10. Разработка индивидуальных программ для бизнеса: плюсы, минусы и альтернативы. – URL: <https://falconspace.ru/blog/individualnaya-razrabotka-programm-dlya-biznesa--plyusy--minusy-i-alternativy> (дата обращения: 20.05.2026).
11. Почему многие выбирают WordPress - несколько причин выбора. – URL: https://ok2web.ru/plaginy-wordpress/plugin-forminator-sozдание-kontaktnyh-platezhnyh-i-kastomizirovannyh-form-dlya-vashego-sayta/?utm_referrer=https%3A%2F%2Fwww.google.com%2F (дата обращения: 20.05.2026).
12. WordPress: характеристика, преимущества и недостатки. – URL: <https://otus.ru/journal/wordpress-harakteristika-preimushhestva-i-nedostatki/> (дата обращения: 20.05.2026).
13. Преимущества Wordpress. – URL: <https://habr.com/ru/articles/251315/> (дата обращения: 22.05.2026).
14. CMS WordPress: плюсы и минусы. – URL: <https://eurobyte.ru/articles/plyusy-i-minusy-wordpress/> (дата обращения: 22.05.2026).
15. Языки для web-разработки сайтов. – URL: <https://brandlif.ru/blog/languages-web-development/> (дата обращения: 24.05.2026).
16. HTML и CSS в веб-разработке. – URL: <https://younglinux.info/htmlcss/intro> (дата обращения: 24.05.2026).
17. Что такое реклама простыми словами: виды, задачи, цели. – URL: <https://kokos.com/blog/tekstovaya-reklama/> (дата обращения: 24.05.2026).
18. PHP: описание, особенности, области применения. – URL: <https://otus.ru/journal/php-opisanie-osobennosti-oblasti-primeneniya/> (дата обращения: 24.05.2026).
19. Язык программирования PHP. – URL: <https://optimalgroup.ru/blog/yazyk-programirovaniya-php/> (дата обращения: 15.05.2026).
20. Почему PHP идеально подходит для веб-разработки: преимущество языка и запуск первого сайта. – URL: <https://ru.hexlet.io/blog/posts/pochemu-php-idealno-podhodit-dlya-veb-razrabotki> (дата обращения: 15.05.2026).
21. Язык программирования PHP – обзор. – URL: <https://edu.sravni.ru/kursy/info/yazyk-programirovaniya-php/> (дата обращения: 15.05.2026).

© Д. Д. Малыхин, А. А. Шарапов, 2026

СОДЕРЖАНИЕ

1. В. Д. Аверин. Метод мониторинга работы локальных пользователей электронных вычислительных машин в условиях использования средств анонимизации	3
2. К. Д. Антонников, Н. С. Головачев. Проектирование и разработка иммерсивной виртуальной среды для изучения всемирной истории «ArtHistory»	11
3. Д. С. Антропов. Метод классификации киберинцидентов с использованием контекстных признаков и анализа временных закономерностей	16
4. М. С. Аристов, А. Денисенко, С. А. Мартынов, А. А. Шарапов. Разработка модульного корпуса мобильного робота в Autodesk Inventor	24
5. В. К. Базанов, А. А. Шарапов. Проектирование и исследование архитектуры высоконагруженного веб-приложения для организации научных конференций на основе микросервисного подхода	29
6. В. З. Бальжинимаева, Д. Ю. Смирнов. Разработка симулятора для получения навыков работы с лабораторным оборудованием, выполняющим анализ ДНК и РНК	36
7. Е. А. Басыня, В. Ю. Сапегин. Метод состязательного обучения моделей обнаружения виртуальных защищенных каналов связи на основе адаптивной генерации конфигураций Xgray	42
8. Е. А. Басыня, В. Ю. Сапегин. Методика сравнительного анализа методов обнаружения VPN-трафика с учетом сдвига домена и дрейфа данных	49
9. М. Ю. Благодаренко, Т. Ю. Бугакова. Развертывание защищенной инфраструктуры удаленных рабочих мест с системой мониторинга на базе Proxmox VE и Docker контейнеров	57
10. А. А. Блохин, А. А. Шарапов. Программное обеспечение для анализа данных электронных таблиц	64
11. А. Р. Валов, Н. С. Головачев. Разработка прототипа симулятора эвакуации и поведения в чрезвычайных ситуациях на основе 3DGS-моделей реальных учебных помещений	70
12. Д. А. Ваулин, Р. В. Гришин. Разработка мультиагентной архитектуры универсальной платформы для автоматизированного обучения чат-ботов	75
13. Е. В. Ващенко, Р. В. Гришин. Разработка системы визуального контроля и классификации дефектов на основе технологий машинного обучения	83
14. Е. Д. Востокова, Н. С. Головачев. Разработка подсистемы управления сценариями и процессами вовлечения пользователей в игровом приложении «BlushBound»	89

15. Е. И. Вяткина, Д. Ю. Смирнов. Разработка конфигурации и интерактивного дашборда для анализа проектов с использованием технологии искусственного интеллекта в 1С: Предприятие 8.3	94
16. Р. Ж. Гафиятов, Е. Ю. Воронкин. Разработка веб-приложения для организации отрядной деятельности и методической поддержки вожатых в детском оздоровительном лагере.....	100
17. И. П. Гилёв, Т. Ю. Бугакова. Разработка аппаратно-программного стенда для автоматизации процесса тестирования аудио- и видеорегистраторов.....	106
18. В. А. Гилева, С. Ю. Кацко. Проектирование и реализация масштабируемого приложения для шаблонного создания структурированных отчетов.....	112
19. Я. И. Головкин, Е. Ю. Воронкин. Разработка программного бота для ранжирования списков абитуриентов и информирования о вступительных испытаниях.....	121
20. Д. А. Гура, Р. А. Дьяченко, Т. А. Тихонов. Подход к решению проблемы дисбаланса классов облаков точек на основе генерации синтетических данных	127
21. Р. А. Данилов, Е. Ю. Воронкин. Унификация полевой документации тахеометрической съемки как основа автоматизированной 3D-визуализации археологических раскопок	134
22. Д. Р. Дарижапова, Д. Ю. Смирнов. Автоматизация оценки платежеспособности заемщиков на платформе «1С: Предприятие 8.3» с применением технологий искусственного интеллекта	138
23. Д. Д. Догадин, Д. Ю. Смирнов. Разработка защищенного интеграционного шлюза между облачной CRM-системой и локальной учетной платформой	145
24. Р. В. Енчу, А. А. Басаргин. Разработка веб-сервиса для мониторинга успеваемости с применением технологий искусственного интеллекта	151
25. Т. О. Ермакова, Н. С. Головачев. Проектирование и программная реализация интерфейсов для системы взаимодействия пользователей с проектом «BlushBound»	157
26. Н. А. Ермилов, Е. Ю. Воронкин. Разработка серверной части SaaS-платформы по управлению e-commerce бизнесом	163
27. И. И. Захаренко, Д. С. Худяков. Интеллектуальная платформа разработки инженерных конструкций	167
28. В. А. Зинков, Е. Ю. Воронкин. Разработка информационной системы автоматизированного учета средств индивидуальной защиты сотрудников предприятия с применением методов искусственного интеллекта	174
29. В. Д. Злочевский. Метод сетевого зондирования объектов корпоративной вычислительной сети	181
30. И. А. Зыков, Е. Ю. Воронкин. Разработка веб-приложения для автоматизации клиентских коммуникаций с поддержкой интеграции во внешние сервисы	189

31. В. Р. Ильясов, Д. Ю. Смирнов. Разработка трехмерного интерактивного пособия по изучению космических исследовательских аппаратов в среде Unreal Engine 5	196
32. К. Д. Кайгородов, Н. С. Головачев. Разработка VR-приложения виртуального музея всемирной истории «ArtHistory» с реализацией интерактивных пользовательских сценариев	203
33. Л. Д. Какабаева. Метод централизованного управления сетевым оборудованием и средствами обеспечения сетевой безопасности	207
34. Е. А. Калинина, А. А. Шарапов. Разработка серверной части автоматизированной системы формирования цифрового портфолио обучающегося	215
35. М. С. Карапетьянц. Метод оценки доступности информационных ресурсов в оверлейной сети на базе луковой маршрутизации	222
36. Н. Карапетьянц. Методы обработки и хранения данных транзакций сети Bitcoin	229
37. К. Ю. Копыл. Анализ проблематики автоматизированного аудита Kubernetes-кластеров	236
38. А. Е. Коровянский, А. А. Шарапов. Разработка и исследование серверной части веб-платформы управления научными конференциями с использованием REST API	243
39. Н. А. Кравец, А. Ф. Лобода, Е. В. Шевчук. Мультиагентный мобильный помощник криминалиста «Фиксатор»	249
40. С. Е. Латыш, Д. Д. Лидер, А. А. Шарапов. Модернизация гексапода: увеличение степеней свободы и применение Jetson Nano	255
41. П. А. Лашин, Н. С. Головачев. Разработка интерактивного 3D-тура по исторической усадьбе Строгоновых с элементами реконструкции исторических объектов	261
42. Р. С. Лонский, А. А. Шарапов. Разработка мобильного приложения для информационной поддержки машиниста электровоза на платформе Android	267
43. Е. К. Макаров, С. Ю. Кацко. Кроссплатформенное приложение для тренировки скорочтения	272
44. Д. Д. Малыхин, А. А. Шарапов. Разработка веб-ресурса по предоставлению сантехнических услуг для индивидуального предпринимателя	279

CONTENTS

1. V. D. Averin. A method for monitoring the activity of local users of electronic computing machines under conditions involving the use of anonymization tools.....	3
2. K. D. Antonnikov, N. S. Golovachev. Design and development of an immersive virtual environment for studying world history "ArtHistory".....	11
3. D. S. Antropov. Cyber incident classification method based on contextual features and temporal pattern analysis	16
4. M. S. Aristov, A. Denisenko, S. A. Martynov, A. A. Sharapov. Development of a modular mobile robot housing using Autodesk Inventor.....	24
5. V. K. Bazanov, A. A. Sharapov. Design and research of the architecture of a high-load web application for organizing scientific conferences based on a microservice approach.....	29
6. V. Z. Balzhinimaeva, D. Yu. Smirnov. Development of a simulator for acquiring skills in operating laboratory equipment for DNA and RNA analysis	36
7. E. A. Basinya, V. Y. Sapegin. Method for adversarial training of virtual secure communication channel detection models based on adaptive Xray configuration generation.....	42
8. E. A. Basinya, V. Y. Sapegin. A methodology for comparative analysis of VPN traffic detection methods under domain shift and data drift	49
9. M. Yu. Blagodarenko, T. Yu. Bugakova. Deployment of a secure remote workplace infrastructure with a monitoring system based on Proxmox VE and Docker containers.....	57
10. A. A. Blokhin, A. A. Sharapov. Software for analyzing spreadsheet data...	64
11. A. R. Valov, N. S. Golovachev. Development of a prototype of an evacuation and emergency behavior simulator based on 3DGS models of real educational facilities	70
12. D. A. Vaulin, R. V. Grishin. Development of a multi-agent architecture of a universal platform for automated training of chatbots.....	75
13. E. V. Vashchenko, R. V. Grishin. Development of a mobile system for visual inspection and defect classification based on machine learning technologies.....	83
14. E. D. Vostokova, N. S. Golovachev. Development of a scenario and user engagement management subsystem in the BlushBound game application.....	89
15. E. I. Vyatkina, D. Yu. Smirnov. Development of configuration and interactive dashboard for project analysis using artificial intelligence technology in 1C: Enterprise 8.3.....	94
16. R. J. Gafiyatov, E. Yu. Voronkin. Development of a web application for organizing detachment activities and methodological support for counselors in a children's health camp.....	100

17. I. P. Gilev, T. Yu. Bugakova. Development of a hardware-software test bench for automating the testing process of audio and video recorders	106
18. V. A. Gileva, S. Y. Katsko. Design and implementation of a scalable application for template-based creation of structured reports.....	112
19. Y. I. Golovko, E. Yu. Voronkin. Development of a program bot to shuffle entrants lists and inform about entrance contests.....	121
20. D. A. Gura, R. A. Dyachenko, T. A. Tikhonov. An approach to solving the problem of point cloud class imbalance based on synthetic data generation	127
21. R. A. Danilov, E. Yu. Voronkin. Unification of field documentation of total station surveys as the Ba-sis for automated 3D visualization of archaeological excavation.....	134
22. D. R. Darizhapova, D. Yu. Smirnov. Automation of borrower solvency assessment on the "1C: Enterprise 8.3" platform using artificial intelligence technologies.....	138
23. D. D. Dogadin, D. Yu. Smirnov. Development of a secure integration gateway between a cloud CRM system and a local accounting platform.....	145
24. R. V. Enchu, A. A. Basargin. Development of a web service for academic performance monitoring using artificial intelligence technologies.....	151
25. T. O. Ermakova, N. S. Golovachev. Design and software implementation of user interfaces for the «BlushBound» project.....	157
26. N. A. Ermilov, E. Yu. Voronkin. Development of the server part of a SaaS-platform for e-commerce business management.....	163
27. I. I. Zakharenko, D. S. Khudyakov. Intelligent system for engineering design development	167
28. V. A. Zinkov, E. Yu. Voronkin. Development of an information system for automated accounting of enterprise employees' personal protective equipment using artificial intelligence methods	174
29. V. D. Zlochevskii. Method for network security probing of corporate network assets.....	181
30. I. A. Zykov, E. Yu. Voronkin. Development of a web application for automating customer communications with support for integration into external services	189
31. V. R. Ilyasov, D. Yu. Smirnov. Development of a three-dimensional interactive guide for studying space research vehicles in Unreal Engine 5.....	196
32. K. D. Kaigorodov, N. S. Golovachev. Development of the VR application of the virtual museum of world history «ArtHistory» with the implementation of Interactive user scenarios	203
33. L. D. Kakabaeva. Method for centralized management of network equipment and network security tools.....	207
34. E. A. Kalinina, A. A. Sharapov. Development of the server part of an automated system for forming a student digital portfolio.....	215
35. M. S. Karapetyants. A method for assessing the availability of information resources in an onion routing overlay network.....	222

36. N. Karapetyants. Methods for processing and storing bitcoin network transaction data.....	229
37. K. Y. Kopyl. Analysis of the problems of automated auditing of kubernetes clusters.....	236
38. A. E. Korovyanskiy, A. A. Sharapov. Development and research of the server part of a web platform for managing scientific conferences using REST API.....	243
39. N. A. Kravets, A. F. Loboda, E. V. Shevchuk. Multi-agent mobile forensic assistant "Fixator".....	249
40. S. E. Latysh, D. D. Lider, A. A. Sharapov. Modernization of a hexapod: increasing degrees of freedom and using Jetson Nano	255
41. P. A. Lashin, N. S. Golovachev. Development of an interactive 3D tour of the historic Strogonov estate with elements of reconstruction of historical sites.....	261
42. R. S. Lonskiy, A. A. Sharapov. Development of a mobile application for information support for electric locomotive drivers on the Android platform	267
43. E. K. Makarov, S. Yu. Katsko. Cross-platform application for speed reading training.....	272
44. D. D. Malykhin, A. A. Sharapov. Development of a web resource for the provision of plumbing services for individual entrepreneurs	279

Научное издание

ИНТЕРЭКСПО ГЕО-СИБИРЬ

XXII Международный научный конгресс

Сборник материалов в 9 т.

Т. 9

Международная научная конференция

«ПЕРСПЕКТИВНЫЕ РЕШЕНИЯ В СФЕРЕ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ И КИБЕРБЕЗОПАСНОСТИ»

№ 1

Материалы публикуются в авторской редакции

Компьютерная верстка *А. Бочарникова*

Изд. лиц. ЛР № 020461 от 04.03.1997.

Подписано в печать 19.08.2026. Формат 60 × 84 1/16.

Усл. печ. л. 16,85. Тираж 25 экз. Заказ 146.

Гигиеническое заключение

№ 54.НК.05.953.П.000147.12.02. от 10.12.2002.

Издательско-полиграфический центр СГУГиТ

630108, Новосибирск, ул. Плахотного, 10.

Отпечатано в издательско-полиграфическом центре СГУГиТ

630108, Новосибирск, ул. Плахотного, 8