

СТАТЬИ

УДК 004.78

**АРХИТЕКТУРНЫЕ ПОДХОДЫ К МОДЕРНИЗАЦИИ
ИНТЕГРАЦИОННЫХ МОДУЛЕЙ ДЛЯ ВЗАИМОДЕЙСТВИЯ
С СИСТЕМАМИ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ**^{1,2}Буряков А.А.

¹Федеральное государственное автономное учреждение
«Научно-исследовательский институт “Восход”», Москва;
²ФГБОУ ВО «Российский технологический университет МИРЭА»,
Москва, e-mail: alex_buryakov@mail.ru

Статья посвящена исследованию современных архитектурных подходов к модернизации интеграционного модуля, обеспечивающего взаимодействие государственных информационных систем с техническими службами поддержки. Актуальность исследования обусловлена существенным ростом нагрузки на системы, усложнением масштабирования и увеличением требований к надежности и гибкости функционирования информационной инфраструктуры государственных учреждений и ведомств. Цель исследования заключалась в определении оптимальных архитектурных решений для повышения эффективности работы интеграционного модуля. В качестве материалов использовались нормативные и технические документы, научные публикации, монографические труды и результаты экспертных оценок. Применялись методы сравнительного анализа различных архитектурных моделей, таких как микросервисная архитектура, корпоративная шина данных и единая точка доступа к программным интерфейсам, а также анализ протоколов обмена информацией с учетом особенностей передачи данных. Результаты исследования показывают, что наиболее эффективным подходом является сочетание оптимальной архитектуры, современных протоколов взаимодействия и механизмов отказоустойчивости. Это позволяет существенно повысить надежность, масштабируемость и скорость работы системы. Практическая значимость исследования заключается в возможности использования предложенной методики при модернизации аналогичных государственных информационных систем, что способствует их устойчивости, оперативности и соответствию современным требованиям информационной безопасности.

Ключевые слова: микросервисная архитектура, ESB, API Gateway, SOAP, REST, gRPC, отказоустойчивость, OpenTelemetry

**ARCHITECTURAL APPROACHES TO THE MODERNIZATION
OF INTEGRATION MODULES FOR INTERACTION
WITH TECHNICAL SUPPORT SYSTEMS**^{1,2}Buryakov A.A.

¹Federal State Autonomous Institution «Scientific Research Institute “Voskhod”», Moscow;
²Russian Technological University MIREA, Moscow, e-mail: alex_buryakov@mail.ru

Article is devoted to the study of modern architectural approaches to the modernization of the integration module, which ensures the interaction of the government information systems with technical support services. The relevance of the study is determined by the significant increase in systems load, the complications in scaling, and the rising demands for the reliability and flexibility of the information infrastructure in government agencies and departments. The aim of the study was to determine the optimal architectural solutions to improve the efficiency of the integration module. Regulatory and technical documents, scientific publications, monographs, and the results of expert evaluations were used as materials. Methods of comparative analysis were applied to various architectural models, such as microservice architecture, enterprise service bus, and a unified access point to application programming interfaces, as well as to an analysis of data exchange protocols that takes into account the specific features of data transmission. The findings indicate that the most effective approach involves combining optimal architecture, modern interaction protocols, and fault tolerance mechanisms. This significantly improves system reliability, scalability, and operational efficiency. The practical significance of this study lies in the potential application of the proposed methodology to modernize similar government information systems, ensuring their stability, responsiveness, and compliance with contemporary information security standards.

Keywords: microservices, ESB, API Gateway, SOAP, REST, gRPC, fault tolerance, OpenTelemetry

Введение

Описываемая государственная информационная система (ГИС) предназначена для повышения качества взаимодействия информационных систем, входящих в инфраструктуру, обеспечивающую информационно-технологическое взаимодействие информационных систем, используемых для предоставления государственных и муниципальных услуг и исполнения госу-

дарственных и муниципальных функций в электронной форме (далее – инфраструктура взаимодействия), и информационных систем, использующих инфраструктуру взаимодействия, а также для обеспечения управления качеством обслуживания пользователей инфраструктуры взаимодействия, непрерывностью и доступностью услуг и сервисов инфраструктуры взаимодействия, формирования отчетности о ее

работе, управления информационной безопасностью и управления инцидентами в работе инфраструктуры взаимодействия. Управление непрерывностью и доступностью услуг и сервисов реализуется за счет комплексного мониторинга и оперативного управления информационными ресурсами в рамках государственных услуг и внутренних процессов. Система обрабатывает большой поток данных от различных источников, обеспечивает взаимодействие между подразделениями и внешними организациями, включая службы технической поддержки (СТП).

Современные государственные информационные системы становятся все более распределенными и сложноорганизованными структурами, где интеграционные модули являются одним из ключевых элементов. От их эффективности и производительности напрямую зависят оперативность реагирования на возникающие инциденты и скорость межведомственного взаимодействия. Внедрение новых сервисов, рост числа обращений и необходимость оперативной обработки больших массивов информации требуют постоянного совершенствования интеграционных механизмов, а также адаптации системы под новые технологические и регламентные требования.

В условиях постоянно растущей сложности государственных информационных систем, жестких требований к их надежности и масштабируемости, а также стремительного увеличения объема обращений возникает необходимость в эффективной и гибкой интеграции с СТП. Термин «модернизация» используется в общепринятом смысле: «комплекс мероприятий по обновлению и доработке существующих компонентов ГИС (программного обеспечения (ПО), технических средств, организационной документации) с целью повышения эффективности их применения» [1].

В случае ГИС модернизация может включать обновление интеграционного модуля, усиление серверных мощностей или оптимизацию нормативно-правовой и методической базы – в зависимости от целей и приоритетов проекта.

Актуальность работы обусловлена тем, что без своевременного обновления интеграционного модуля повышаются риски сбоев, затрудняется масштабирование и усложняется обслуживание системы в целом.

Цель исследования – выявить оптимальные архитектурные решения и технологические подходы к модернизации интеграционного модуля, позволяющие повысить эффективность взаимодействия ГИС с системами техподдержки.

Материалы и методы исследования

Материалами исследования послужили компоненты распределенной интеграционной системы технической поддержки и управления интеграционными и аналитическими сервисами ГИС. В работе были использованы нормативные и технические документы, научные статьи и монографии по вопросам архитектуры информационных систем и протоколов взаимодействия. В работе применялись методы сравнительного анализа для оценки архитектурных решений и протоколов (SOAP, REST, gRPC), обобщения и системного подхода при формулировке выводов.

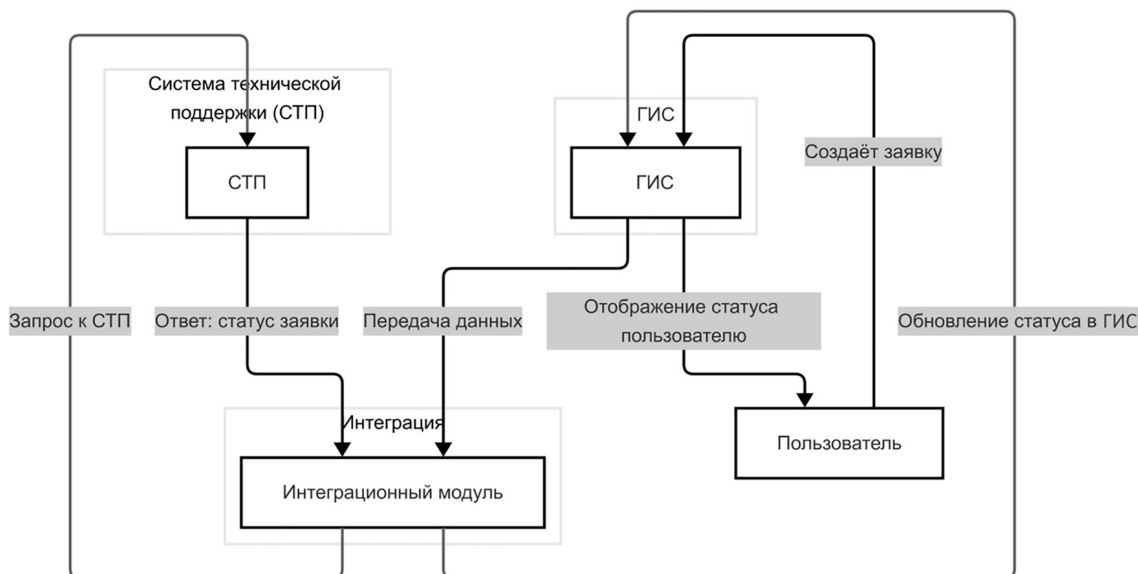
Также изучался опыт аналогичных государственных информационных систем, что позволило выявить практические проблемы и типовые ошибки, которые возникают при модернизации интеграционных решений в государственных учреждениях и ведомствах. Важной частью методологии стало также моделирование работы системы при различных уровнях нагрузки с использованием специализированного программного обеспечения, что позволило дать объективную оценку преимуществ и недостатков каждого подхода, представленного в статье.

Результаты исследования и их обсуждение

Предназначение интеграционного модуля и обоснование необходимости его модернизации

Обработка обращений пользователей к ГИС проходит несколько этапов. Пользователи (внутренние или внешние) формируют обращения или инциденты. ГИС принимает запросы и передает их в интеграционный модуль. Интеграционный модуль обеспечивает связь с системой техподдержки: создает, обновляет и отслеживает заявки, а также возвращает результаты обработки назад в ГИС. Обобщенная схема взаимодействия участников в рамках ГИС представлена на рисунке.

В рамках ГИС взаимодействие с СТП предполагает: автоматизированный обмен данными (инцидентами, проблемами, запросами на обслуживание) в режиме реального времени; синхронизацию статусов и эскалацию заявок (контроль критичности, дедлайны, назначенные исполнители); интеграцию с базами знаний для оперативного решения частых проблем; единую точку доступа для операторов ГИС и других ответственных лиц. В конечном итоге такая интеграция снижает время отклика на инциденты, улучшает согласованность работы разных подразделений и повышает прозрачность контроля исполнения заявок [2].



Обобщенная схема взаимодействия участников в рамках использования ГИС

Достоинства и недостатки архитектур и протоколов

Решение / Протокол	Достоинства	Недостатки
Микросервисная архитектура	– Гибкое масштабирование отдельных сервисов [5] – Независимый цикл разработки – Устойчивость к сбоям отдельных компонентов	– Повышенная сложность инфраструктуры (Kubernetes, сервис-меш) – Сложности с мониторингом и логированием – Необходимость согласованного управления версиями API
ESB (Enterprise Service Bus)	– Централизованный контроль и наблюдение за потоками – Маршрутизация и трансформация «из коробки» – Поддержка различных протоколов (SOAP, REST, JMS и др.)	– Может стать «бутылочным горлышком» при высоких нагрузках – Сложности с внесением изменений в единую шину – Требуются существенные аппаратные ресурсы для масштабирования
API Gateway	– Единая точка входа и управления безопасностью – Удобные кеширование и балансировка – Управление версиями API	– Не решает задачу бизнес-логики интеграции – Может усложнить трассировку в больших системах (нужен сервис-меш) – Требует корректной настройки маршрутизации для разных сервисов
SOAP	– Строгая типизация и формальный контракт (WSDL) – Поддержка широкого спектра стандартов безопасности (WS-*) – Хорошая совместимость с «корпоративными» платформами (.NET, Java EE)	– «Тяжелый» XML-формат, избыточность сетевого трафика – Более сложная настройка и отладка – Уступает REST/gRPC в гибкости
REST	– Простота реализации (JSON) – Высокая популярность и поддержка веб-клиентами – Гибкость и масштабируемость	– Нет жесткого контракта без OpenAPI/Swagger – Необходимость дополнительной документации – Могут возникать разночтения при реализации разных REST-эндпоинтов
gRPC	– Высокая производительность и низкие задержки – Двоичный формат Protocol Buffers – Встроенная поддержка потоков (streaming)	– Сложнее в освоении, чем REST. Не всегда прозрачно работает с существующими L7-балансировщиками – Требует специальных инструментов для генерации кода и описания контрактов

Рост количества пользователей, обусловленный развитием сервисов по предоставлению государственных услуг в электронном виде, вызывает потребность в модернизации интеграционного модуля. Помимо этого, в ходе исследований были выявлены следующие причины необходимости модернизации.

- *Рост объемов данных и нагрузки.* ГИС обрабатывает все больше заявок и инцидентов, поэтому для поддержания скорости реакции и качества обслуживания важно оптимизировать производительность и масштабируемость.

- *Устаревание существующих решений.* Традиционные (монолитные) способы взаимодействия не поддерживают современные протоколы и форматы, усложняя интеграцию с новыми СТП [3].

- *Возрастающие требования к надежности и безопасности.* Государственные регламенты и стандарты (например, ГО-СТы и требования ФСТЭК) требуют надежных методов аутентификации, шифрования и протоколов обмена данными [4].

Для устранения выявленных проблем в первую очередь необходимо модернизировать архитектуру интеграционного модуля, что обеспечит устойчивость системы к нагрузкам, быстрое внедрение новых сервисов, а также упрощенное обслуживание. Кроме того, модернизация архитектуры создаст гибкость при расширении функционала ГИС, т.е. позволит добавлять в ГИС новые сервисы (например, модули аналитики), не затрагивая при этом основные процессы и не увеличивая время на внесение изменений.

Сравнительный анализ архитектурных решений и протоколов

Для выбора оптимального подхода к модернизации важно учитывать преимущества и ограничения различных архитектур (микросервисов, ESB, API Gateway) и протоколов (SOAP, REST, gRPC), каждая из которых обладает своими плюсами и минусами (таблица).

Опираясь на данные таблицы, в конкретном проекте выбирают сочетание наиболее подходящих решений [6, 7]. Например, при необходимости легкой интеграции с веб-приложениями может быть сделан упор на REST, а для высоконагруженного обмена пакетными данными – gRPC. Если требуется жесткий контроль маршрутизации и трансформации сообщений, следует рассмотреть ESB. При создании масштабируемых систем с множеством микросервисов целесообразно внедрять API Gateway и сервис-меш.

Обеспечение отказоустойчивости интеграционного модуля и внедрение средств мониторинга

Одной из главных причин модернизации интеграционного модуля является все возрастающая нагрузка на ГИС, поэтому необходимо обратить внимание на обеспечение отказоустойчивости интеграционного модуля. Предлагается использовать следующие подходы и инструменты.

- *Горизонтальное масштабирование.* Масштабирование отдельных сервисов с помощью Docker и Kubernetes позволяет реагировать на резкий рост числа обращений.

- *Circuit Breaker.* Защищает систему от каскадных сбоев, позволяя вовремя «размыкать цепочку» при ошибках со стороны одного из сервисов.

- *Балансировщики нагрузки.* Позволяют распределять входящие запросы по нескольким экземплярам сервиса, повышая общую доступность и живучесть системы.

Для прогнозирования и своевременного предотвращения проблемных ситуаций при использовании ГИС предлагается использовать следующие инструменты телеметрии, мониторинга и трассировки.

- *OpenTelemetry.* Сбор метрик, логов и распределенных трасс из различных компонентов системы в единую точку. Это критично для оперативного выявления узких мест и анализа инцидентов.

- *Системы визуализации.* Grafana, Kibana (Elastic Stack) помогают в реальном времени отслеживать ключевые показатели (время отклика, процент ошибок и т.д.).

- *Распределенная трассировка.* Инструменты вроде Jaeger и Zipkin позволяют проследить «путь» запроса через весь комплекс сервисов, ESB или API Gateway, ускоряя диагностику проблем.

Заключение

Таким образом, в результате проведенного исследования были определены оптимальные архитектурные подходы к модернизации интеграционного модуля ГИС. Реализация предложенных решений позволяет значительно повысить надежность, производительность и гибкость всей структуры ГИС. Комплексное сочетание современных архитектурных моделей, эффективных протоколов взаимодействия и средств обеспечения отказоустойчивости обеспечивает повышение устойчивости системы к нагрузкам, упрощает процессы интеграции новых сервисов и способствует быстрому внедрению инноваций. Практическая значимость работы заключается в возможности применения разработанных рекомен-

даций для модернизации государственных информационных систем, что обеспечит их стабильность, соответствие современным требованиям информационной безопасности и, как следствие, повысит общую эффективность работы государственных учреждений и ведомств.

Список литературы

1. Бусыгина Е.В., Жученко М.А., Якунова М.С., Куракина Ю.В. Модернизация системы управления информационной безопасностью в образовательной организации // Исследования молодых ученых. 2024. № 8(64). [Электронный ресурс]. URL: <https://mgpu-media.ru/issues/issue-64/psikhologo-pedagogicheskie-nauki/modernizatsiya-sistemy-upravleniya-informatsionnoj-bezopasnostyu-v-obrazovatelnoj-organizatsii.html> (дата обращения: 01.03.2025).
2. OpenTelemetry Documentation. [Электронный ресурс]. URL: <https://opentelemetry.io/> (дата обращения: 01.03.2025).
3. Малыгин Д.С. Микросервисная архитектура в облачных системах: риски и возможности применения в 2024–2030 гг. // Моделирование, оптимизация и информационные технологии. 2024. № 12(2). URL: <https://moitvvt.ru/ru/journal/pdf?id=1561>. DOI: 10.26102/2310-6018/2024.45.2.029 (дата обращения: 01.03.2025).
4. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). W3C Recommendation, 27 Apr. 2007. [Электронный ресурс]. URL: <https://www.w3.org/TR/soap12-part1/> (дата обращения: 01.03.2025).
5. Обухов А.Д., Волков А.А., Назарова А.О. «Микросервисная архитектура виртуальных тренажерных комплексов» // Информатика и автоматизация. 2022. Т. 6. Вып. 21. С. 112–118. DOI: 10.15622/ia.21.6.7.
6. Афанасьев М.Я., Федосов Ю.В., Крылова А.А., Порохов С.А. Применение микросервисной архитектуры при проектировании промышленного оборудования с числовым программным управлением // Научно-технический вестник информационных технологий, механики и оптики. 2018. Т. 18, № 1. С. 87–97. DOI: 10.17586/2226-1494-2018-18-1-87-97.
7. Никитин И.В., Гриценко Т.Ю. Сравнение подходов монолитной архитектуры и микросервисной архитектуры при реализации серверной части веб-приложения // Дневник науки. 2020. № 3 (39). URL: https://dnevniknauki.ru/images/publications/2020/3/technics/Nikitin_Gritsenko.pdf (дата обращения: 01.03.2025).