ПОДХОД К РЕАЛИЗАЦИИ БЕЗОПАСНОГО УДАЛЕННОГО АДМИНИСТРИРОВАНИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ ИНФРАСТРУКТУР

Морозов Ф.А.[1]

Ключевые слова: *брокеры сообщений, асинхронное взаимодействие, отказоустойчивость, обратный опрос, информационная безопасность, микросервисная архитектура, персистентные очереди, гарантии доставки, централизованный аудит.*

Аннотация

Цель работы. *Анализ архитектурных ограничений существующих подходов к удалённому администрированию и разработка архитектурной модели защищённого механизма управления распределённой серверной инфраструктурой с гарантированной доставкой управляющих воздействий.*

Метод исследования. *В работе применены методы сравнительного анализа существующих систем удалённого администрирования, архитектурного моделирования и обобщения эксплуатационных характеристик программных решений. Рассмотрены синхронные и асинхронные модели взаимодействия, оценены их свойства в условиях сетевой сегментации и динамически изменяющегося состава узлов. Проведена систематизация требований к механизму администрирования с учётом критериев безопасности, масштабируемости и отказоустойчивости.*

Результаты исследования. *В ходе исследования обоснована целесообразность применения асинхронной архитектуры на основе брокера сообщений и модели обратного взаимодействия, при которой управляемые узлы инициируют соединение самостоятельно. Данный подход позволяет отказаться от организации входящих соединений к серверам и существенно сократить потенциальную атакующую поверхность распределённой инфраструктуры. Сформулирован формализованный подход к гарантированной доставке управляющих воздействий, учитывающий подтверждение получения, фиксацию исполнения и аудит операций. Предложенная модель описывает жизненный цикл команды через формально определённые состояния и переходы между ними, что обеспечивает сквозное отслеживание выполнения каждой операции. Полученные результаты позволяют снизить атакующую поверхность и обеспечить устойчивость механизма управления в условиях временной недоступности отдельных компонентов, что подтверждено экспериментальной апробацией разработанного прототипа.*

DOI: 10.24412/1994-1404-2026-3-60-69

Введение

Современные распределённые серверные инфраструктуры характеризуются высокой степенью территориальной и логической децентрализованности, использованием виртуализированных сред и сегментацией сетевого пространства. Такие условия предъявляют повышенные требования к механизмам удалённого администрирования, поскольку управление должно обеспечивать масштабируемость, устойчивость к сетевым ограничениям и соответствие требованиям информационной безопасности.

Традиционные синхронные модели управления предполагают установление соединения со стороны управляющего компонента к каждому узлу инфраструктуры. В распределённых средах с многоуровневой сегментацией и динамически изменяющимся составом серверов подобный подход может сопровождаться усложнением контроля доступа и ростом числа входящих соединений. Это формирует необходимость поиска архитектурных решений, ориентированных на снижение атакующей поверхности и повышение устойчивости к временной недоступности узлов.

Дополнительной задачей является формализация критерия гарантированной доставки управляющего воздействия. В практических системах выполнение команды нередко трактуется как факт её отправки или получения подтверждения на уровне протокола, что не всегда отражает реальное завершение операции. Отсутствие формализованного описания жизненного

[1] ***Морозов Фёдор Андреевич****, специалист отдела мониторинга и реагирования на киберугрозы ПАО «Ростелеком».*
E-mail: fedor-555m@yandex.ru

цикла команды затрудняет оценку корректности функционирования механизма управления.

В связи с этим в работе ставится задача разработки архитектурной модели защищённого механизма администрирования распределённой серверной инфраструктуры, обеспечивающего асинхронное взаимодействие, минимизацию входящих соединений и формализованную гарантированную доставку управляющих воздействий.

АНАЛИЗ ПОДХОДОВ К УПРАВЛЕНИЮ РАСПРЕДЕЛЁННЫМИ СЕРВЕРНЫМИ ИНФРАСТРУКТУРАМИ

Особенности управления распределёнными серверными средами

Высокая территориальная и логическая распределённость современных серверных инфраструктур предъявляет дополнительные требования к управлению. Организации используют комбинацию локальных центров обработки данных, удалённых филиалов, виртуализированных платформ и облачных сервисов. Подобная архитектура позволяет повысить гибкость вычислительных ресурсов, обеспечить отказоустойчивость и адаптивность к изменяющимся нагрузкам, однако одновременно усложняет процессы администрирования.

Управление распределённой инфраструктурой предполагает координацию большого числа узлов, находящихся в различных сетевых сегментах и функционирующих в условиях неоднородных политик безопасности. В отличие от централизованных систем, где взаимодействие осуществляется в пределах единого доверенного контура, распределённая среда включает сегменты с различными уровнями доступа, сетевыми ограничениями и механизмами фильтрации трафика. Это накладывает ограничения на способы установления соединений и передачи управляющих воздействий.

Рост числа узлов прямо влияет на увеличение объёма административных операций: конфигурация программного обеспечения, обновление компонентов, контроль состояния сервисов, аудит действий и реагирование на инциденты. При использовании традиционных методов удалённого доступа увеличение масштаба приводит к росту числа активных сетевых соединений и усложнению контроля доступа, что повышает требования к управляемости и безопасности.

Дополнительным фактором является динамичность инфраструктуры. Виртуальные машины и контейнеры могут создаваться и удаляться автоматически, изменяться в процессе масштабирования или миграции между площадками. В таких условиях управление должно учитывать изменяемость состава узлов и обеспечивать устойчивость к временной недоступности отдельных компонентов.

Таким образом, распределённость, сегментация и динамичность формируют совокупность ограничений, при которых традиционные синхронные модели администрирования могут демонстрировать снижение эффективности. Это обусловливает необходимость анализа архитектурных подходов, ориентированных на масштабируемость, асинхронность взаимодействия и снижение атакующей поверхности.

Архитектурные факторы, влияющие на процессы администрирования

Архитектура распределённой инфраструктуры, как правило, включает несколько уровней: центральные вычислительные ресурсы, периферийные узлы, облачные сегменты и изолированные подсети, предназначенные для размещения специализированных сервисов. Между сегментами применяются механизмы фильтрации трафика, разграничения доступа и контроля соединений.

Наличие сетевой сегментации является объективным требованием информационной безопасности. Однако с точки зрения администрирования это означает, что прямое установление входящих соединений к управляемым узлам может быть ограничено политиками межсетевых экранов или требованиями изоляции. В результате классическая модель, при которой управляющий сервер инициирует соединение к каждому узлу, становится менее универсальной.

Кроме того, различие в уровнях доверия между сегментами требует строгого контроля аутентификации и авторизации. Административные учётные записи должны использоваться в минимально необходимом объёме, а сами управляющие воздействия — протоколироваться и контролироваться. При увеличении числа узлов централизованный контроль за множеством прямых подключений усложняется.

Виртуализация и контейнеризация усиливают динамический характер инфраструктуры. Узлы могут временно отсутствовать в сети, находиться за трансляцией адресов или быть доступными только через промежуточные компоненты. В подобных условиях синхронное выполнение команд, предполагающее немедленный отклик, не всегда отражает фактическое состояние системы. В связи с этим, как отмечено в [1], асинхронное взаимодействие с использованием брокеров сообщений является ключевым средством для решения недостатков синхронных вызовов, таких как каскадные сбои и высокие задержки.

Таким образом, архитектурные особенности распределённой среды ограничивают применение моделей, основанных исключительно на постоянной доступности управляемых узлов и прямом сетевом взаимодействии. Это требует рассмотрения альтернативных схем обмена управляющими сообщениями, способных функционировать в условиях сегментации и временной недоступности отдельных компонентов.

Сравнительный анализ существующих подходов к удалённому администрированию

На практике применяются несколько классов решений для управления серверной инфраструктурой.

Первый класс — прямые методы удалённого доступа, основанные на использовании протоколов интерактивного подключения. Их достоинством является простота реализации и оперативность выполнения задач. Однако при увеличении числа узлов возрастает количество открытых сервисов и входящих соединений, что увеличивает требования к контролю доступа и мониторингу безопасности. В условиях распределённой сегментированной сети такие методы могут требовать дополнительной настройки маршрутизации и политик фильтрации.

Второй класс — системы управления конфигурациями, ориентированные на автоматизацию развёртывания и поддержания заданного состояния узлов. Они обеспечивают централизованное описание конфигурации и позволяют выполнять массовые операции. При этом эффективность их применения зависит от сетевой доступности управляемых компонентов и корректности механизма доставки задач. В ряде сценариев выполнение операций предполагает наличие постоянного соединения между управляющим компонентом и узлами.

Третий класс — решения, использующие брокеры сообщений и асинхронную модель взаимодействия. В данной схеме узлы самостоятельно инициируют соединение к централизованному компоненту обмена сообщениями и получают задания через очереди или каналы подписки. Такой подход позволяет сократить количество входящих соединений к управляемым узлам и адаптироваться к временной недоступности отдельных сегментов. Асинхронный характер взаимодействия упрощает масштабирование и балансировку нагрузки. Как подтверждают исследования [2, 3], использование брокеров сообщений (например, Kafka, RabbitMQ) способствует повышению отказоустойчивости и масштабируемости систем за счёт внедрения событийно-ориентированной архитектуры (EDA) и асинхронной обработки данных.

Сравнительный анализ показывает, что каждый подход имеет преимущества и ограничения в условиях высокой распределённости. Прямые методы оперативны, но требуют множества входящих соединений. Системы управления конфигурациями эффективны для поддержания состояния, однако зависят от сетевой топологии. Решения на основе брокеров сообщений гибки и асинхронны, но требуют формализации подтверждения выполнения задач.

Таким образом, ни один из подходов в изолированном виде не гарантирует одновременное выполнение требований масштабируемости, безопасности и подтверждённой доставки управляющих воздействий. Это указывает на необходимость формализации критериев корректности управления и уточнения требований к механизму доставки команд.

Для более наглядного сопоставления рассмотренных подходов проведём их сравнительный анализ по ряду обобщённых критериев, отражающих требования распределённой серверной инфраструктуры.

Результаты сравнительного анализа представлены в таблице 1.

Обобщённые требования к механизму администрирования

На основании проведённого анализа можно сформулировать совокупность требований к механизму управления распределённой инфраструктурой.

В контексте безопасности механизм должен минимизировать количество входящих соединений к управляемым узлам и обеспечивать строгую аутентификацию участников взаимодействия. Необходимо предусмотреть разграничение прав доступа и фиксацию административных действий в журнале аудита.

Таблица 1

Сравнительный анализ подходов к администрированию

Критерий	Прямые методы удалённого доступа	Системы управления конфигурациями	Решения на основе брокеров сообщений
Инициирование соединения	Со стороны управляющего узла	Чаще со стороны управляющего узла	Со стороны управляемого узла
Требования к входящим подключениям	Наличие открытых сервисов на каждом узле	Зависит от архитектуры решения	Минимальные, используется исходящее соединение
Масштабируемость	Ограничена ростом числа соединений	Поддерживается средствами автоматизации	Обеспечивается асинхронной моделью
Устойчивость к временной недоступности узлов	Низкая, требуется повторное подключение	Частично обеспечивается	Поддерживается механизмами очередей
Подтверждение выполнения операции	Обычно на уровне сессии	На уровне выполнения задания	Может быть формализовано на уровне сообщений
Централизованный аудит	Требует дополнительной интеграции	Реализуется средствами системы	Может быть встроен в механизм обработки сообщений

В отношении доставки управляющих воздействий требуется определить формальный критерий завершённости операции. Отправка команды сама по себе не может рассматриваться как достаточное условие её выполнения. Механизм должен обеспечивать подтверждение получения, фиксацию результата исполнения и сохранение информации о состоянии операции.

С позиции масштабируемости система должна поддерживать увеличение числа узлов без пропорционального роста сложности управления. Это предполагает использование архитектурных решений, допускающих горизонтальное масштабирование и асинхронный обмен сообщениями.

Обеспечение отказоустойчивости требует сохранения корректности функционирования при временной недоступности отдельных компонентов. Управляющие воздействия не должны теряться при кратковременных сбоях связи, что может быть обеспечено за счёт применения паттернов отказоустойчивости и гарантий доставки, описанных в [2, 3].

Наконец, важным требованием является централизованный аудит, позволяющий отслеживать последовательность административных операций и обеспечивать воспроизводимость состояния системы.

Формирование указанных требований задаёт направление дальнейшего проектирования и обосновывает необходимость разработки модели, в которой процесс управления будет описан через формально определённые состояния и переходы.

ОБОСНОВАНИЕ АРХИТЕКТУРНЫХ РЕШЕНИЙ И ВЫБОРА ТЕХНОЛОГИЙ

Анализ существующих решений и технологических подходов

Для обоснования архитектуры разработанного механизма проведён анализ распространённых инструментов удалённого администрирования и средств межсервисного взаимодействия с точки зрения их применимости в распределённых средах.

Прямые методы удалённого доступа обеспечивают оперативность выполнения задач, однако при масштабировании инфраструктуры увеличивается количество активных соединений и усложняется централизованный контроль доступа и аудит. Их эффективность снижается в условиях сегментированных сетей и динамически изменяющегося состава узлов.

Системы управления конфигурациями ориентированы на автоматизацию поддержания заданного состояния серверов. Они повышают управляемость инфраструктуры, однако архитектурные особенности отдельных решений предполагают зависимость от сетевой доступности узлов или периодического опроса агентов. В динамичных средах это может приводить к задержкам выполнения операций.

Отдельный класс решений представляют брокеры сообщений, обеспечивающие асинхронную доставку данных между компонентами системы [4]. Такие инструменты не являются средствами администрирования в готовом виде, однако предоставляют механизмы подтверждения получения сообщений, буферизации и устойчивости к временной недоступности участников взаимодействия.

Для более детального анализа возможностей различных архитектурных подходов рассмотрим их функциональные характеристики в контексте задач удалённого администрирования, как показано в таблице 2.

Проведённый анализ показывает, что решения на основе асинхронного обмена сообщениями в наибольшей степени соответствуют требованиям распределённой среды, однако требуют архитектурной надстройки, ориентированной на задачи администрирования. Как показано в [5], использование имитационного моделирования процессов обработки сообщений с применением брокеров позволяет оценить производительность распределённых систем ещё на этапе проектирования.

Таблица 2

Сопоставление архитектурных подходов

Критерий	Прямые методы доступа (SSH)	Системы управления конфигурациями	Подход на базе брокера сообщений
Гибкость выполнения команд	Произвольные команды в интерактивном режиме	Выполнение заранее описанных сценариев	Передача произвольных команд через очередь сообщений
Поддержка отложенного выполнения	Отсутствует	Ограниченно поддерживается средствами планирования	Поддерживается за счёт хранения сообщений в очередях
Контроль корректности команд	Отсутствует	Проверка структуры сценариев	Может быть реализован на уровне обработки сообщений
Управление на основе метаданных узлов	Отсутствует	Используются группы и инвентари	Требует дополнительного механизма тегирования
Централизованный аудит операций	Ограничен журналами соединений	Поддерживается средствами системы	Требует реализации на уровне управляющего сервиса

Обоснование архитектуры на основе асинхронного обмена сообщениями

Предлагаемая архитектура основана на сочетании микросервисного подхода и механизма асинхронного взаимодействия через брокера сообщений.

Ключевым принципом является модель обратного взаимодействия, при которой управляемые узлы самостоятельно инициируют соединение с компонентом обмена сообщениями и получают управляющие воздействия через выделенные каналы. Это позволяет отказаться от организации входящих соединений к серверам и сократить атакующую поверхность.

Архитектура включает три логически разделённых компонента: сервер управления, реализующий прикладную логику и интерфейс взаимодействия; компонент маршрутизации, обеспечивающий распределение команд; агенты выполнения, осуществляющие обработку задач на узлах инфраструктуры.

Асинхронный обмен через брокера сообщений обеспечивает буферизацию команд и подтверждение их обработки. При временной недоступности узла управляющее воздействие сохраняется до момента восстановления соединения. Как показано в [6], такой подход позволяет формализовать жизненный цикл команды и определить критерий её завершённости.

Альтернативные архитектурные решения, основанные на монолитной структуре или прямом межсервисном взаимодействии, обладают меньшей гибкостью масштабирования и требуют дополнительных механизмов повторной передачи данных. Использование брокера сообщений обеспечивает баланс между структурной простотой и устойчивостью к отказам.

Обоснование выбора технологического стека

Выбор технологий осуществлялся исходя из требований к масштабируемости, безопасности и сопровождаемости системы.

В качестве архитектурной основы использован микросервисный подход, обеспечивающий разделение ответственности и независимое масштабирование компонентов.

Для реализации серверной части выбран язык Python, позволяющий использовать асинхронную модель обработки запросов и развитую экосистему библиотек для сетевого взаимодействия. Применение современного веб-фреймворка обеспечивает реализацию REST-интерфейса и унифицированного механизма интеграции с внешними системами.

В качестве брокера сообщений использован инструмент, поддерживающий подтверждение получения сообщений и механизмы их сохранения. Это позволяет реализовать гарантированную доставку управляющих воздействий и обеспечить устойчивость к кратковременным сбоям связи.

Для хранения данных применена реляционная система управления базами данных с поддержкой транзакционности и репликации. Как показано в [7], применение архитектурного паттерна отказоустойчивого взаимодействия с брокерами сообщений (ограничение интенсивности запросов, обработка ошибок) обеспечивает гарантированную доставку управляющих воздействий и устойчивость к сбоям связи.

Выбранный стек ориентирован на использование зрелых и поддерживаемых технологий, что снижает риски внедрения и обеспечивает возможность дальнейшего развития системы.

Формирование требований к реализации

На основании проведённого анализа сформулированы требования к реализации механизма администрирования.

Система должна обеспечивать централизованное управление узлами распределённой инфраструктуры без необходимости открытия входящих портов на управляемых серверах. Необходимо реализовать механизм разграничения доступа, журналирование действий и формализованный контроль выполнения команд.

Функционально система должна поддерживать регистрацию агентов, адресное и групповое выполнение команд, хранение истории операций и интеграцию с внешними сервисами через программный интерфейс.

Архитектурно требуется поддержка горизонтального масштабирования, асинхронной обработки сообщений и устойчивости к временной недоступности отдельных компонентов.

Эксплуатационные требования включают совместимость с гетерогенными средами, минимальные зависимости на стороне агента и возможность интеграции с существующими средствами мониторинга и управления.

Сформулированные требования определяют рамки дальнейшего проектирования и обеспечивают согласованность между архитектурными решениями и целями разработки.

ПРОЕКТИРОВАНИЕ, РЕАЛИЗАЦИЯ И АПРОБАЦИЯ МЕХАНИЗМА АДМИНИСТРИРОВАНИЯ

Архитектурная модель и функциональные возможности системы

Для реализации защищённого механизма администрирования распределённой серверной инфраструктуры была разработана архитектура программного комплекса, основанная на микросервисном подходе и асинхронной модели взаимодействия компонентов. Основной задачей проектирования являлось создание системы, обеспечивающей безопасное централизованное управление серверами, гарантированную доставку управляющих воздействий и устойчивость к сетевым ограничениям распределённой инфраструктуры.

Архитектура системы включает три основных компонента: центральный сервер управления, сервис

маршрутизации команд и программный агент, устанавливаемый на управляемых узлах. Взаимодействие компонентов осуществляется через брокера сообщений, обеспечивающий асинхронную передачу управляющих команд и результатов их выполнения. Хранение состояния системы и истории операций осуществляется в реляционной базе данных.

Ключевой особенностью архитектуры является использование механизма **обратного опроса**. В отличие от традиционных систем администрирования, где управляющий сервер инициирует соединение с управляемыми узлами, в предложенном решении инициатором взаимодействия выступает агент. Управляемые серверы самостоятельно устанавливают соединение с брокером сообщений и получают команды из персональных очередей. Такой подход позволяет исключить необходимость открытия входящих сетевых портов на управляемых серверах и снижает потенциальную атакующую поверхность системы.

На рис. 1 представлена схема взаимодействия компонентов системы.

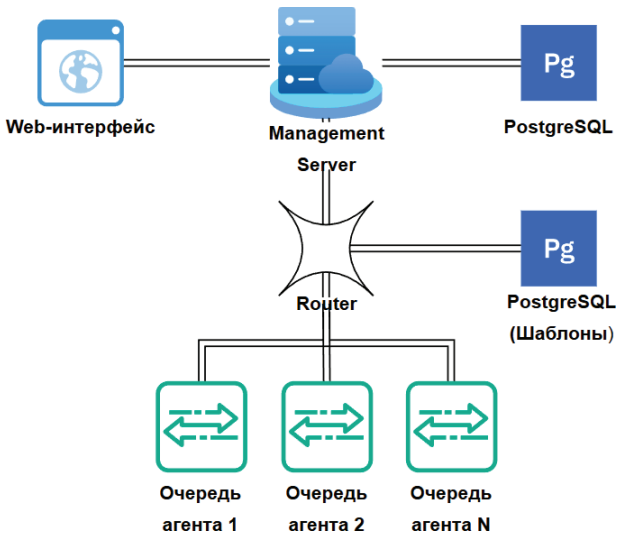


Рис. 1. Схема взаимодействия компонентов системы

Предложенная архитектура обеспечивает реализацию следующих функциональных возможностей:

- централизованное управление инфраструктурой, позволяющее выполнять административные операции на множестве серверов из единой точки управления;
- выполнение команд на группах серверов, формируемых динамически на основе метаданных управляемых узлов;
- система шаблонов команд, обеспечивающая создание параметризованных сценариев администрирования и повторное использование типовых операций;
- динамическое таргетирование серверов, позволяющее автоматически определять целевые узлы на основе заданных критериев;
- журналирование и аудит операций, фиксирующие историю всех выполненных административных действий;
- мониторинг состояния серверов, включая сбор базовых метрик и контроль активности агентов.

Одним из ключевых требований к системе являлось обеспечение гарантированной доставки управляющих воздействий. Для решения данной задачи используется брокер сообщений с поддержкой персистентных очередей и механизмов подтверждения обработки сообщений. Сравнительный анализ производительности RabbitMQ и Kafka [8] демонстрирует, что выбор конкретного брокера зависит от требований к пропускной способности и задержкам, что было учтено при проектировании архитектуры. Такой подход позволяет сохранять команды в очереди до момента их успешного выполнения агентом.

Асинхронная модель взаимодействия обеспечивает устойчивость системы к временной недоступности отдельных компонентов. При разрыве соединения команды сохраняются в очереди сообщений и выполняются после восстановления связи с агентом.

Предложенная архитектура обеспечивает модульность системы, независимость компонентов и возможность масштабирования при увеличении количества управляемых серверов.

Реализация ключевых механизмов администрирования

В ходе реализации программного комплекса были разработаны основные механизмы управления распределённой инфраструктурой.

Ключевым элементом системы является механизм обратного опроса, обеспечивающий безопасное подключение управляемых серверов. Каждый агент при запуске автоматически регистрируется в системе и создаёт персональную очередь команд в брокере сообщений. Далее агент периодически проверяет свою очередь и получает новые задачи.

Использование брокера сообщений позволяет реализовать гарантированную доставку команд [9]. Для этого применяются персистентные очереди и подтверждение обработки сообщений. В случае временной недоступности узла команда сохраняется в очереди и будет выполнена после восстановления соединения.

В системе реализована возможность динамического таргетирования серверов. Каждый агент может иметь набор метаданных (тегов), характеризующих его назначение, роль или расположение. При отправке команды пользователь может указать фильтр по тегам, после чего система автоматически определяет набор целевых серверов.

Дополнительной функцией является система шаблонов команд, позволяющая создавать параметризованные сценарии администрирования. Шаблон представляет собой описание команды с переменными параметрами, которые подставляются при выполнении

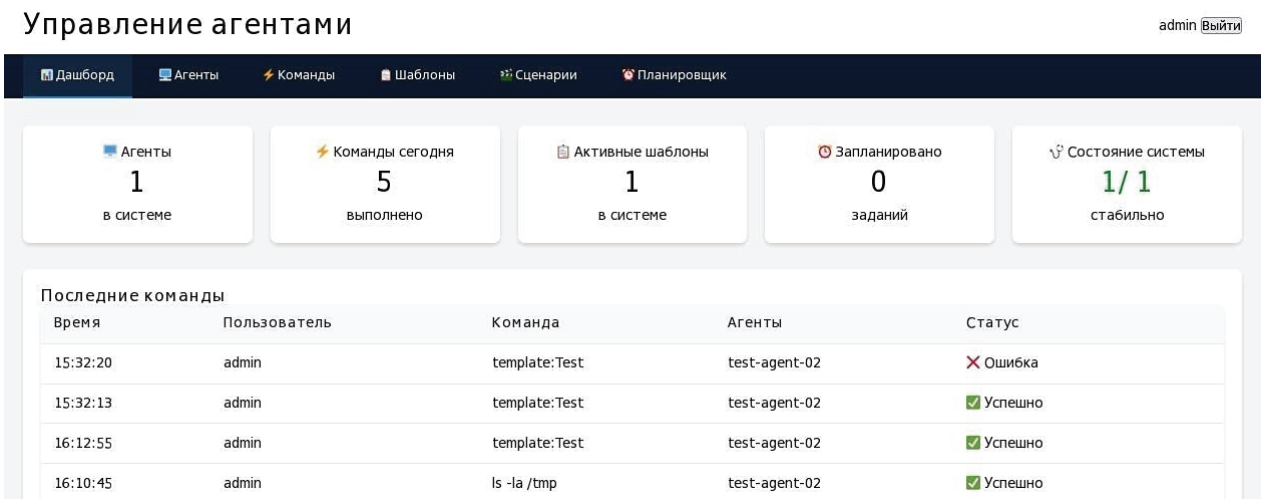


Рис. 2. Главный дашборд системы

операции. Такой подход позволяет стандартизировать типовые операции и снизить вероятность ошибок администратора.

Для предотвращения ошибочных действий реализован механизм валидации команд. Перед отправкой команда проверяется на наличие потенциально опасных операций и опечаток. Это позволяет уменьшить вероятность случайного выполнения критических команд.

Также в системе реализована подсистема аудита и мониторинга, включающая:

– журнал всех выполненных операций;
– историю выполнения команд;
– мониторинг состояния агентов;
– сбор метрик ресурсов серверов.

Интерфейс системы обеспечивает централизованное управление инфраструктурой и визуализацию состояния серверов. На рис. 2 показан главный дашборд системы, предоставляющий сводную информацию о состоянии инфраструктуры.

Использование централизованного интерфейса позволяет администратору получать сводную информацию о состоянии инфраструктуры и контролировать выполнение операций управления.

Апробация прототипа и оценка эффективности решения

Для проверки работоспособности разработанного механизма был развернут тестовый стенд, включающий:

– одну виртуальную машину для центральных компонентов системы;
– две виртуальные машины, выступающие в роли агентов.

Использование компактного тестового стенда является распространённой практикой тестирования распределённых систем, поскольку позволяет проверить функциональные возможности системы на минимальной конфигурации с последующей экстраполяцией результатов на более крупные инфраструктуры.

В ходе тестирования проверялись следующие сценарии:

– регистрация и управление агентами;
– выполнение команд на отдельных серверах;
– выполнение команд на группах серверов;
– работа системы шаблонов;
– механизмы безопасности и аутентификации;
– отказоустойчивость системы.

Сравнительный анализ брокеров сообщений (RabbitMQ, Kafka, NATS) по задержке доставки и пропускной способности [10] обосновывает выбор архитектуры для высоконагруженных систем.

Основные временные характеристики выполнения операций представлены в таблице 3.

Таблица 3

Временные характеристики выполнения операций

Операция	Минимальное время	Среднее время	Максимальное время
Проверка нагрузки (uptime)	10 мс	24 мс	35 мс
Проверка состояния служб (ps aux | grep -c «nginx»)	12 мс	21 мс	40 мс
Обновление состояния	8 с	12 с	15 с

Дополнительно была проведена проверка механизмов безопасности системы. Результаты тестирования представлены в таблице 4.

Проведённый анализ показал, что разработанная система обеспечивает стабильную работу при типовых нагрузках и демонстрирует устойчивость к одиночным сбоям компонентов.

Сравнительный анализ с традиционными подходами администрирования представлен в таблице 5.

Таблица 4

Эффективность механизмов безопасности

Механизм безопасности	Тестовых сценариев	Успешно обработано	Эффективность
Валидация опасных команд	45	44	97.8%
Обнаружение опечаток	35	33	94.3%
Доставка команд после разрыва	25	24	96.0%

Таблица 5

Сравнение подходов к администрированию

Критерий	Разработанная система	Ansible	Прямой SSH
Безопасность	Обратный опрос	Прямые соединения	Открытые порты
Доставка команд	Персистентные очереди	Требует доступности узлов	Нет механизма
Централизованное управление	Единый Web-интерфейс Динамическое таргетирование	CLI & Статические инвентари	Разрозненные подключения
Мониторинг	Встроенная система здоровья	Требует доп. средств	Отсутствует

Полученные результаты демонстрируют преимущества предложенной архитектуры в аспектах безопасности, централизованного управления и устойчивости к сбоям.

Выводы

В ходе исследования разработан механизм безопасного удалённого администрирования распределённых серверных инфраструктур, основанный на асинхронной модели взаимодействия с использованием брокера сообщений и принципа обратного опроса управляемых узлов. Предложенный подход позволяет минимизировать количество входящих сетевых соединений, что снижает потенциальную атакующую поверхность инфраструктуры, и обеспечивает гарантированную доставку управляющих воздействий за счёт применения персистентных очередей и механизмов подтверждения обработки сообщений. Научная новизна работы заключается в формализации жизненного цикла команды и критерия её завершённости, что позволяет однозначно интерпретировать состояние выполнения административных операций в условиях временной недоступности отдельных компонентов распределённой системы.

Достоверность предложенных решений подтверждена экспериментальной апробацией разработанного прототипа на тестовом стенде. В ходе тестирования подтверждена работоспособность ключевых механизмов: валидации опасных команд, обнаружения опечаток и доставки команд после разрыва связи, что свидетельствует о практической применимости разработанного подхода. Полученные результаты позволяют использовать предложенную архитектуру в качестве основы для построения систем централизованного управления распределённой инфраструктурой, обеспечивающих безопасность, масштабируемость и отказоустойчивость.

Литература

1. Баженов А.Э., Райков А.В., Нехаев М.В., Ореховский Н.В. Оптимизация взаимодействия микросервисов с использованием асинхронных вызовов и брокеров сообщений (Kafka, RabbitMQ) // Программные системы и вычислительные методы. 2025. № 4. С. 77—92.
2. Барабанов П.П. Исследование паттернов отказоустойчивости и гарантий доставки для высоконагруженного обмена пользовательскими метриками // Актуальные исследования. 2026. №9 (295). Ч. I. С. 26—30.
3. Сухов Д.Д., Еленев Д.В., Муравьева Е.В. Исследование методов обмена данными с помощью брокеров сообщений // Минобрнауки России, Самар. нац. исслед. ун-т им. С.П. Королева. Самара, 2025.
4. Агеев А.Е. Разработка архитектурного паттерна для отказоустойчивого взаимодействия высоконагруженных приложений с брокерами сообщений // Сборник тезисов докладов конгресса молодых ученых. СПб. : Университет ИТМО. 2025.

5. Паринов П.А., Сирота А.А. Моделирование процессов обработки сообщений в системах с микросервисной архитектурой с использованием брокера сообщений // Системный анализ и информационные технологии. 2025. № 3. С. 15—28.
6. Макаров П.С. Исследование эффективности брокеров сообщений в высоконагруженных распределённых системах // Студенческий электронный научный журнал. 2025. № 18 (314).
7. Карпович М.Н. Особенности проектирования микросервисно-событийных архитектур для высоконагруженных распределенных систем обработки информации // Труды БГТУ. Серия 3: Физико-математические науки и информатика. 2023. № 1 (266). С. 89—95.
8. Kamiński F., Kłonica R., Pańczyk B. Comparative Performance Analysis of RabbitMQ and Kafka Message Queue Systems in Spring Boot and ASP.NET Environments // Journal of Computer Sciences Institute. 2025. Vol. 37. P. 457—462. DOI: 10.35784/jcsi.7977.
9. Селезнёв А.И., Селезнёв И.Л. Брокеры сообщений в системах обработки данных // Молодой ученый. 2025. № 46 (597).
10. Зарубин В.А. Построение событийно-ориентированной архитектуры в системе кредитного скоринга // Молодежь и современные информационные технологии : сб. тр. XXII Межд. науч.-практ. конф., 15—17 апреля 2025 г., Томск. Томск : Изд-во ТПУ, 2025. С. 477—481.

SECTION:
SYSTEM ANALYSIS, MANAGEMENT, ARTIFICIAL INTELLIGENCE, AND INFORMATION PROTECTION

A SECURE REMOTE ADMINISTRATION APPROACH FOR DISTRIBUTED COMPUTING INFRASTUCTURES

Morozov F.A.[2]

Keywords: *message brokers, asynchronous communication, fault tolerance, reverse polling, information security, microservice architecture, persistent queues, delivery guarantees, centralized audit.*

Abstract

Objective. *The aim of the research is to analyze architectural limitations of existing approaches to remote administration and to develop an architectural model of a secure management mechanism for distributed server infrastructure with guaranteed delivery of control actions.*

Methods. *The research employs methods of comparative analysis of existing remote administration systems, architectural modeling, and generalization of operational characteristics of software solutions. Synchronous and asynchronous interaction models are considered; their properties are evaluated under conditions of network segmentation and dynamically changing node composition. The requirements for the administration mechanism are systematized considering security, scalability, and fault tolerance criteria.*

Results. *This study substantiates the feasibility of using an asynchronous architecture based on a message broker and a reverse communication model, in which controlled nodes initiate connections independently. This approach eliminates the need for incoming connections to servers and significantly reduces the potential attack surface of a distributed infrastructure. A formalized approach to guaranteed delivery of control actions is formulated, taking into account confirmation of receipt, execution logging, and operation auditing. The proposed model describes the command lifecycle through formally defined states and transitions between them, ensuring end-to-end tracking of each operation. The obtained results reduce the attack surface and ensure the resilience of the control mechanism under conditions of temporary unavailability of individual components, as confirmed by experimental validation of the developed prototype.*

[2] ***Fedor A. Morozov****, Specialist, Cyber Threat Monitoring and Response Department, PJSC Rostelecom.*
E-mail: fedor-555m@yandex.ru

References

1. Bazhenov A.E., Raikov A.V., Nekhaev M.V., Orekhovskii N.V. Optimizatsiia vzaimodeistviia mikroservisov s ispol'zovaniem asinkhronnykh vyzovov i brokerov soobshchenii (Kafka, RabbitMQ) // Programmnye sistemy i vychislitel'nye metody. 2025. No. 4. S. 77—92.
2. Barabanov P.P. Issledovanie patternov otkazoustoichivosti i garantii dostavki dlia vysokonagruzhennogo obmena pol'zovatel'skimi metrikami // Aktual'nye issledovaniia. 2026. No.9 (295). Ch. I. S. 26—30.
3. Sukhov D.D., Elenev D.V., Murav'eva E.V. Issledovanie metodov obmena dannymi s pomoshch'iu brokerov soobshchenii // Minobrnauki Rossii, Samar. nats. issled. un-t im. S.P. Koroleva. Samara, 2025.
4. Ageev A.E. Razrabotka arkhitekturnogo patterna dlia otkazoustoichivogo vzaimodeistviia vysokonagruzhennykh prilozhenii s brokerami soobshchenii // Sbornik tezisov dokladov kongressa molodykh uchenykh. SPb. : Universitet ITMO. 2025.
5. Parinov P.A., Sirota A.A. Modelirovanie protsessov obrabotki soobshchenii v sistemakh s mikroservisnoi arkhitekturoi s ispol'zovaniem brokera soobshchenii // Sistemnyi analiz i informatsionnye tekhnologii. 2025. No. 3. S. 15—28.
6. Makarov P.S. Issledovanie effektivnosti brokerov soobshchenii v vysokonagruzhennykh raspredelennykh sistemakh // Studencheskii elektronnyi nauchnyi zhurnal. 2025. No. 18 (314).
7. Karpovich M.N. Osobennosti proektirovaniia mikroservisno-sobytiinykh arkhitektur dlia vysokonagruzhennykh raspredelennykh sistem obrabotki informatsii // Trudy BGTU. Seriia 3: Fiziko-matematicheskie nauki i informatika. 2023. No. 1 (266). S. 89—95.
8. Kamiński F., Kłonica R., Pańczyk B. Comparative Performance Analysis of RabbitMQ and Kafka Message Queue Systems in Spring Boot and ASP.NET Environments // Journal of Computer Sciences Institute. 2025. Vol. 37. P. 457—462. DOI: 10.35784/jcsi.7977.
9. Seleznev A.I., Seleznev I.L. Brokery soobshchenii v sistemakh obrabotki dannykh // Molodoi uchenyi. 2025. No. 46 (597).
10. Zarubin V.A. Postroenie sobytiino-orientirovannoi arkhitektury v sisteme kreditnogo skoringa // Molodezh' i sovremennye informatsionnye tekhnologii : sb. tr. XXII Mezhd. nauch.-prakt. konf., 15—17 aprelia 2025 g., Tomsk. Tomsk : Izd-vo TPU, 2025. S. 477—481.