УДК 004.773.5

Архитектура локально развёртываемой платформы голосовых комнат с ролевой модерацией и событийной синхронизацией состояний участников

Жук И. А. (ИТМО)
Научный руководитель – Цопа Е.А. (ИТМО)

Введение. В условиях ограниченной доступности части зарубежных коммуникационных платформ и неоднородного качества альтернативных решений актуальной становится задача создания независимого сервиса голосового взаимодействия, пригодного для локального развёртывания и адаптации под реальные пользовательские сценарии. Для малых сообществ (студенческие группы, проектные команды, дружеские сообщества) ключевыми требованиями являются не только стабильная голосовая связь, но и наличие социального контура: профили, контакты, гибкая модерация комнат и управляемый доступ. [4, 5]

Анализ существующего положения показывает, что массовые зарубежные продукты обеспечивают высокий уровень технологической зрелости, однако ориентированы на внешнюю облачную инфраструктуру и ограниченно контролируются локальным оператором. Ряд отечественных решений обеспечивает базовую функциональность, но часто не предоставляет достаточной гибкости в управлении ролями участников, в согласовании бизнес-правил с real-time логикой и в наблюдаемости системы на уровне событий и метрик. Таким образом, возникает прикладная техническая проблема: разработка архитектуры, которая объединяет транзакционный контур информационной системы и real-time контур голосового взаимодействия без потери согласованности состояний. [3, 4, 5]

Основная часть. В работе предложена многокомпонентная архитектура, разделяющая ответственность между подсистемами: frontend-клиент, backend-сервис управления доменной моделью, выделенный voice-сервис real-time взаимодействия, реляционное хранилище и событийная шина. Backend на Kotlin/Spring Boot реализует доменные сущности и политики доступа (пользователь, профиль, контакт, комната, участник комнаты, уведомление, тег), а voice-сервис на Go реализует сигналинг и обработку подключений в реальном времени. PostgreSQL используется как источник истины для бизнес-данных, Redis — как канал событийной синхронизации и присутствия. [1, 2, 3]

Ключевой сценарий (вход в комнату) построен как двухконтурный процесс: пользователь аутентифицируется в backend-контуре, получает короткоживущий токен подключения к комнате с ролевым снимком состояния, после чего подключается к voice-контру. Дальнейшие изменения ролей и mute-состояний распространяются через post-commit события, что уменьшает риск рассинхронизации между транзакционным и real-time слоями. [1, 2]

Для обеспечения целостности доменной модели применяются не только проверки на уровне API, но и инварианты на уровне базы данных (ограничения и триггеры): предотвращение self-contact сценариев, блокировка некорректных переходов состояния участников, недопущение критических нарушений модерации. Такой подход формирует многоуровневую защиту корректности данных: policy-уровень приложения + транзакционные гарантии хранилища. [4, 5]

Выводы. Разработан архитектурный подход и реализация к построению локально развёртываемой платформы голосовых комнат, сочетающий социальный функционал и real-time коммуникацию в единой информационной системе. Предложенное решение устраняет ключевые прикладные ограничения существующих альтернатив за счёт контролируемой инфраструктуры, ролевой модели доступа и событийной синхронизации состояний. [1, 2, 4, 5]

Результаты работы могут быть использованы для внедрения в учебных, проектных и корпоративных сообществах, где требуются технологическая независимость, расширяемость и управляемая модерация коммуникации. В качестве следующего этапа внедрения предлагается пилотная эксплуатация с экспериментальной оценкой качества сервиса по показателям устойчивости подключения, времени восстановления сессий, корректности ролевых переходов и пользовательской удовлетворённости. [3, 6]

Список использованных источников

1. Гарипов, Э. И. Разработка и внедрение драйвера протокола WebSocket в асинхронный фреймворк userver [Электронный ресурс] // Конгресс молодых ученых Университета ИТМО. — 2023. — URL: https://kmu.itmo.ru/digests/article/11464 (дата обращения: 28.02.2026).
2. Фролов, М. А. Анализ взаимодействия распределенных сервисов с использованием телеметрии [Электронный ресурс] // Конгресс молодых ученых Университета ИТМО. — 2023. — URL: https://kmu.itmo.ru/digests/article/9736 (дата обращения: 28.02.2026).
3. Чупров, С. С. Влияние показателя качества обслуживания (QoS) коммуникационной сети на точность классификации данных системами на основе машинного обучения [Электронный ресурс] // Конгресс молодых ученых Университета ИТМО. — 2022. — URL: https://kmu.itmo.ru/digests/article/8823 (дата обращения: 28.02.2026).
4. Мошников, А. С. Сравнение архитектур систем управления по метрикам структурной и надежностной значимости [Электронный ресурс] // Конгресс молодых ученых Университета ИТМО. — 2022. — URL: https://kmu.itmo.ru/digests/article/8979 (дата обращения: 28.02.2026).
5. Мошников, А. С., Сивов, В. В. Имитационное моделирование как инструмент проектирования отказоустойчивых систем с заданными характеристиками надежности [Электронный ресурс] // Конгресс молодых ученых Университета ИТМО. — 2023. — URL: https://kmu.itmo.ru/digests/article/10475 (дата обращения: 28.02.2026).
6. Инкин, М. Е. Исследование оценки качества речевых сигналов для автоматической системы идентификации дикторов [Электронный ресурс] // Конгресс молодых ученых Университета ИТМО. — 2023. — URL: https://kmu.itmo.ru/digests/article/11433 (дата обращения: 28.02.2026).

Автор_____ Жук И.А.

Научный руководитель _____ Цопа Е.А.