

СЕКЦИЯ 6

СОВРЕМЕННЫЕ МОБИЛЬНЫЕ И WEB-ТЕХНОЛОГИИ

И. П. МАРКОВСКИЙ, Г. Б. БАРСКАЯ

Тюменский государственный университет, г. Тюмень

УДК 004.773.3

ЕДИНАЯ СИСТЕМА МНОГОКАНАЛЬНОЙ РАССЫЛКИ СООБЩЕНИЙ

Аннотация. В статье рассматривается процесс создания единой системы многоканальной рассылки сообщений, обеспечивающей гибкость, масштабируемость и конфиденциальность доставки. Система разработана на основе преимуществ и недостатков имеющихся решений на рынке и позволяет организациям настраивать каналы рассылки под собственные нужды, разворачивать сервис на собственных серверах, а также интегрировать новые каналы рассылки сообщений без изменений архитектуры. В ходе работы разработана архитектура решения, MVP и представлены перспективы развития системы.

Ключевые слова: многоканальная рассылка сообщений, система уведомлений, self-host, event-driven архитектура, масштабируемость, расширяемость.

Введение. Современные организации нуждаются в эффективных и безопасных средствах коммуникации для взаимодействия с клиентами и сотрудниками. Однако существующие решения для рассылки сообщений зачастую узкоспециализированные и не позволяют включать и отключать каналы связи. Кроме технических ограничений, встает вопрос безопасности передаваемой информации. Использование сторонних сервисов третьей стороной в процессе рассылки сообщений, создает дополнительное звено между отправителем и получателем. Это повышает риск утечек данных и потери контроля над конфиденциальной рассылкой [1]. В ряде случаев данные пересылаются через облачные сервисы, расположенные за рубежом. Компании вынуждены интегрировать несколько отдельных сервисов, следить за их работоспособностью и платить за каждый из них отдельно. Это не только повышает издержки, но и увеличивает риск ошибок и утечек информации.

Проблема заключается в отсутствии единой универсальной системы [2], которая бы предоставляла:

- все основные каналы рассылки (e-mail, messenger, push, SMS [3], WebSocket и in-app);
- возможность развертывания на собственных серверах [4];
- возможность подключения новых каналов и логики без изменения архитектуры системы [5];
- простая масштабируемость под нагрузку;

В связи с этим была поставлена цель: разработать единую систему многоканальной рассылки сообщений, которая обеспечит централизованное управление всеми каналами доставки, конфиденциальность и расширяемость.

Для достижения поставленной цели сформулированы следующие задачи:

- проанализировать существующие решения и выявить их сильные и слабые стороны;

- спроектировать архитектуру системы на основе выявленных преимуществ;
- реализовать MVP — минимально жизнеспособную версию системы с поддержкой e-mail и telegram каналов.

Исследованы существующие сервисы рассылки по следующим критериям:

- количество и разнообразие поддерживаемых каналов;
- возможность установки на собственные серверы (self-host);
- гибкость настройки;
- расширяемость;
- стоимость и ограничения.

Результат анализа представлен в таблице 1.

Из результатов сравнительного анализа можно выявить достоинства и недостатки каждого из сервисов, для использования при проектировании архитектуры системы.

Таблица 1

Анализ решений на рынке

	<i>Twilio Send-Grid</i> [6]	<i>Unisender</i> [7]	<i>Mailgun</i> [8]	<i>Pusher</i> [9]	<i>OneSignal</i> [10]	<i>Centrifugo</i> [11]
Self-host	-	-	+	-	-	+
Каналы связи	e-mail	e-mail, sms, chat-bot	e-mail	push, WebSocket, in-app	e-mail, sms, chat-bot, in-app	WebSocket
Расширение каналов связи	-	-	-	-	-	-
Бесплатно	Бесплатно, но 14 дней	Бесплатно, но 1500 сообщений	Бесплатно	Бесплатно, но 200 к сообщений	Бесплатно, но 10 к сообщений	Бесплатно
Масштабируемость	Автоматическая	Автоматическая	Нет	Автоматическая	Автоматическая	Нет

Результаты анализа показали, что:

- большинство решений ориентированы только на один или два канала (например, только e-mail или SMS);
- редкие платформы поддерживают self-host, что ограничивает контроль над данными;
- поддержка расширения новыми каналами либо отсутствует, либо требует вмешательства в кодовую базу.

Материалы и методы. Исходя из результатов анализа была спроектирована следующая архитектура системы и представлена на рис. 1.

В данной архитектуре использованы следующие технологии:

- ядро системы — язык программирования Golang;
- очередь задач — Redis [12];
- база данных — PostgreSQL [13];
- веб-интерфейс и API — Next.js + ShadcnUI;
- инфраструктура — контейнеризация через Docker;
- мониторинг — OpenTelemetry.

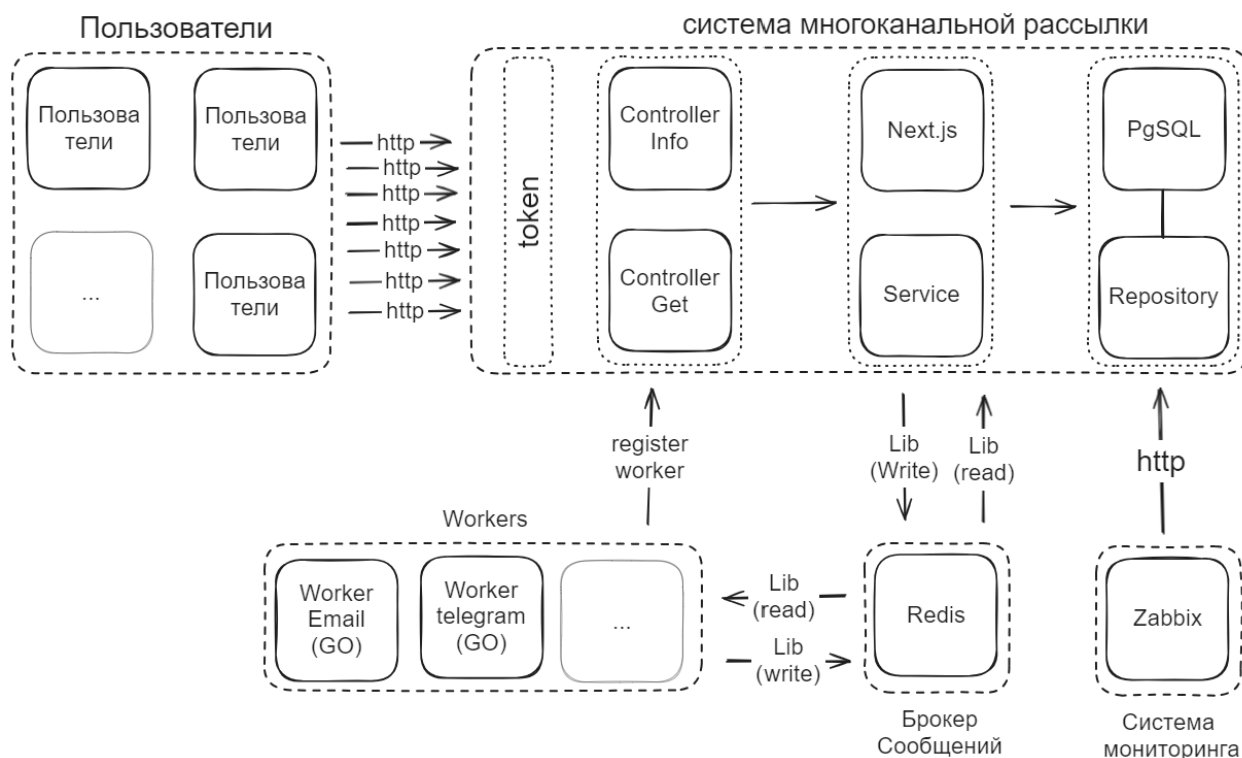


Рис. 1. Архитектура системы многоканальной рассылки

Представленная архитектура построена на основе event-drive паттерна распределенных система, позволяющая достичь высокую степень расширения и отказоустойчивости [4]. Каждый канал рассылки и функциональный блок системы работает в виде изолированного worker-сервиса, реагирующего на события, поступающие через централизованную очередь сообщений. Такой подход обеспечивает:

- асинхронную обработку сообщений;
- низкую связность компонентов, что упрощает добавление новых каналов и обновление логики;
- горизонтальное масштабирование, при котором нагрузка равномерно распределяется между воркерами [5];
- мониторинг и трассировку событий с помощью систем наблюдения, таких как OpenTelemetry.

Вся логика доставки сообщений реализована, как цепочка событий (workflow), каждая стадия которой независима и может масштабироваться отдельно. Это делает систему устойчивой к сбоям, легко адаптируемой под различные сценарии использования и готовой к внедрению новых каналов без модификации существующего кода.

Для описания базы данных выбрана нотация IDEF1X, которая позволяет наглядно представить связи между сущностями, их атрибуты и ограничения [14]. На рис. 2 показана логическая структура данных, лежащая в основе многоканальной системы рассылки сообщений.

Спроектирован также интерфейс для данной системы позволяющий управлять доступом, процессом рассылки и настройками каналов рассылки сообщений, представленный на рис. 3 и 4.

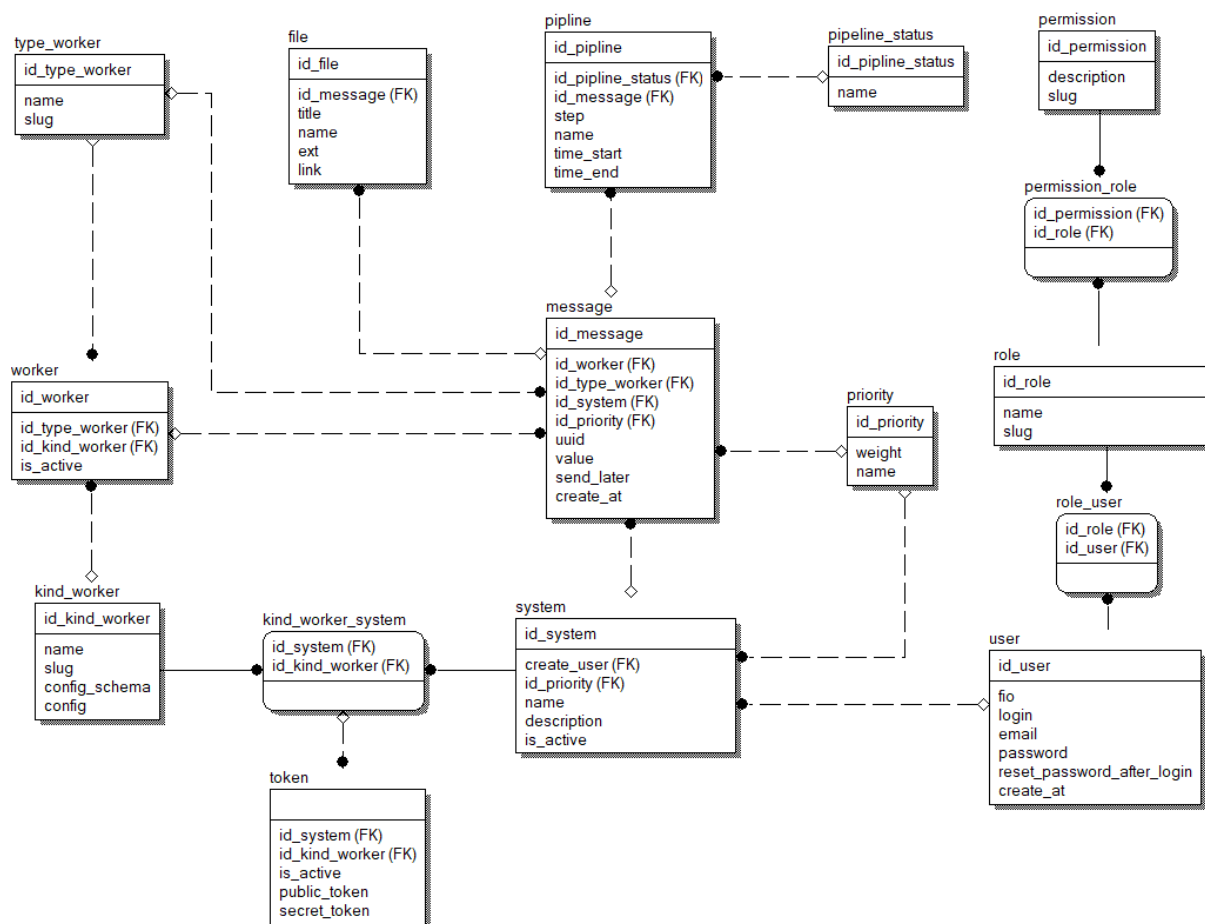


Рис. 2. Логическая модель данных в нотации IDEF1X

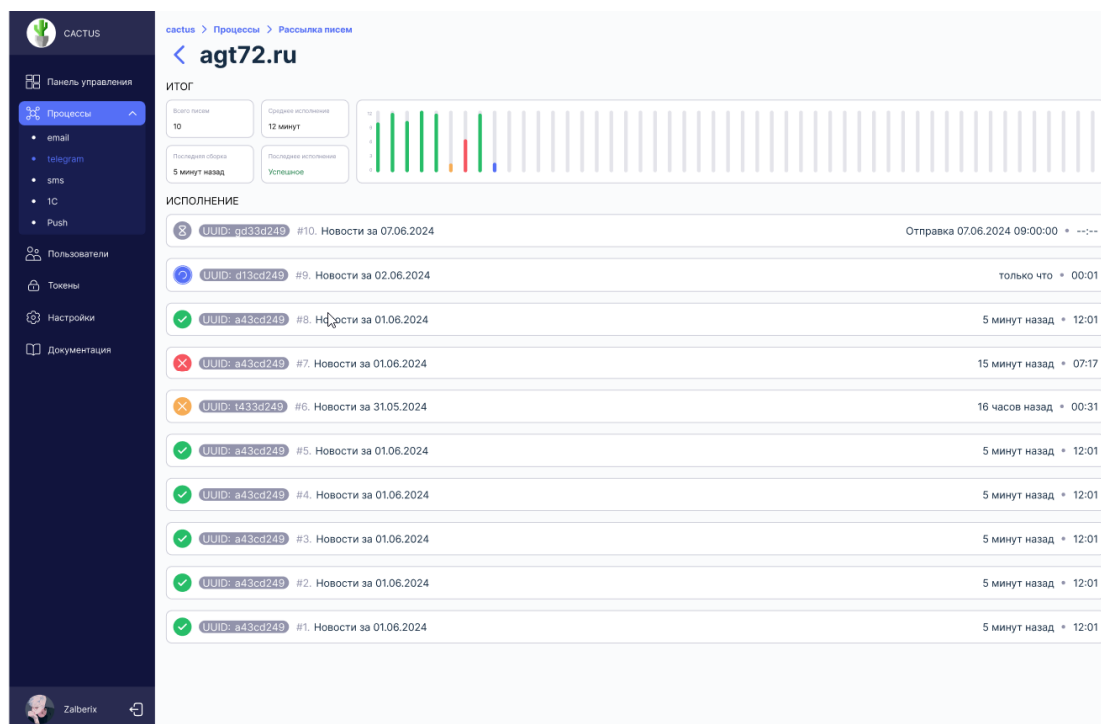


Рис. 3. Экран статусов рассылки сообщений

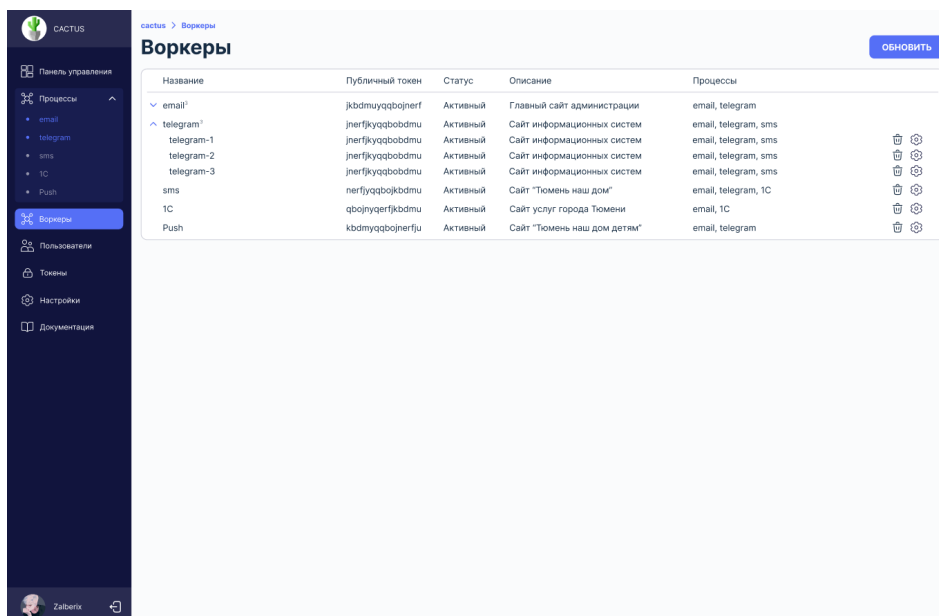


Рис. 4. Экран настройки каналов рассылки

Результаты. Для данной системы разработано MVP, позволяющее отправлять сообщения по e-mail и telegram каналам.

Для удобства использования настроен docker [15], позволяющий одной командой запускать проект, результат старта проекта представлен на рисунке 5.

```
[+] Running 5/5
✓ telegram-bot          Built
✓ Container cactus-redis-1 Running
✓ Container mailhog      Running
✓ Container cactus-pgsql-1 Running
✓ Container cactus-telegram-bot-1 Started
```

Рис. 5. Запуск docker образов системы

Созданы миграции структуры базы данных [10] и первичных данных для первого старта проекта, представленных на рисунке 6.

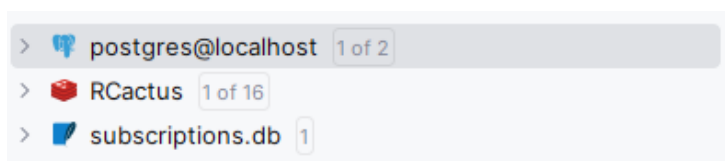


Рис. 6. Создание баз данных для первого старта

Реализована отправка сообщений через REST API по e-mail и telegram каналам рассылки. С возможностью отслеживать состояние сообщения в текущий момент времени, его расположение и возможные ошибки отправки.

Заключение. Разработка MVP единой системы многоканальной рассылки сообщений продемонстрировала возможность создания расширяемой и масштабируемой коммуникационной системы, ориентированной на нужды современных компаний. В сравнении с большинством

существующих решений, предложенная система позволяет отказаться от использования сторонних сервисов, которые часто выступают в роли третьих сторон в вопросе конфиденциальности данных рассылки.

В рамках разработанной MVP с поддержкой e-mail и telegram каналов показал минимальную стабильную работу и успешную расширяемость каналов. Кроме того, реализованы механизмы мониторинга, регистрации событий и управления приоритетами сообщений.

В перспективе планируется расширение функциональности:

- подключение дополнительных каналов (SMS, push, WebSocket и др.);
- реализация маркетплейса подключаемых каналов;
- создание облачной версии системы;
- внедрение модели монетизации через пакеты рассылок и подписку.

Таким образом, разрабатываемая система сможет решать задачи, которые подходят, как для внутреннего использования организациями, так и для взаимодействия с широкой аудиторией — включая автоматические уведомления, сервисные сообщения, маркетинговые рассылки и поддержку двустороннего общения с пользователями в режиме реального времени.

СПИСОК ЛИТЕРАТУРЫ

1. Унифицированные коммуникации для малого бизнеса. — URL: <https://cyberleninka.ru/article/n/unifitsirovannye-kommunikatsii-dlya-malogo-biznesa>. (дата обращения: 10.05.2025).
2. Программная архитектура решений для поддержки омниканальности. — URL: <https://cyberleninka.ru/article/n/programmная-arhitektura-resheniy-dlya-podderzhki-omnikanalnosti>. (дата обращения: 10.05.2025).
3. СМС-рассылки и административная ответственность. — URL: <https://cyberleninka.ru/article/n/sms-rassylki-i-administrativnaya-otvetstvennost>. (дата обращения: 10.05.2025).
4. Сэм Ньюмен. Создание микросервисов. — 2-е изд. — Санкт-Петербург: Питер, 2025. — 624 с.
5. Адам Беллемар. Создание событийно-управляемых микросервисов. — Санкт-Петербург: БХВ-Петербург, 2022 — 320 с.
6. Twilio Inc. Twilio SendGrid: документация для разработчиков. — URL: <https://www.twilio.com/docs/sendgrid> (дата обращения: 10.05.2025).
7. Unisender. API для массовых email-рассылок. — URL: <https://www.unisender.com/ru/support/api/common/bulk-email/> (дата обращения: 10.05.2025).
8. Sinch. Mailgun: документация для разработчиков. — URL: <https://documentation.mailgun.com/> (дата обращения: 10.05.2025).
9. Pusher Ltd. Pusher: документация. — URL: <https://pusher.com/docs/> (дата обращения: 10.05.2025).
10. OneSignal Inc. OneSignal: документация. — URL: <https://documentation.onesignal.com/> (дата обращения: 11.05.2025).
11. Centrifugal Labs. Centrifugo: документация. — URL: <https://centrifugal.dev/docs/getting-started/introduction> (дата обращения: 10.05.2025).
12. Redis. Документация по Redis. — URL: <https://redis.io/docs> (дата обращения: 10.05.2025).
13. PostgreSQL Professional. Документация по PostgreSQL. — URL: <https://postgrespro.ru/docs> (дата обращения: 13.05.2025).
14. Ивашко, А. Г. Проектирование информационных систем : учеб.-метод. пособие / А. Г. Ивашко. — Тюмень : Изд-во ТюмГУ, 2007. — 329 с. — С. 200-210.
15. Docker. Документация по контейнеризации. — URL: <https://docs.docker.com> (дата обращения: 10.05.2025).