

Wialon Message Hub — шлюз доставки уведомлений Wialon Local в Telegram и MAX

1. Проблема

Wialon Local — распространённая локальная система спутникового мониторинга транспорта. Она умеет формировать разнообразные уведомления: включение зажигания, движение, превышение скорости, потеря и восстановление связи с терминалом, разряд аккумулятора, перезагрузка терминала, срабатывания датчиков (температура, уровень топлива, геозоны) и другие события, настраиваемые администратором Wialon.

Штатные каналы доставки уведомлений Wialon не всегда покрывают то, что реально нужно конкретному заказчику. В частности, возникает необходимость доставлять уведомления через мессенджеры, которыми диспетчеры и ответственные лица пользуются постоянно — **Telegram** и российский мессенджер **MAX** — с гибким управлением тем, кто из сотрудников какие типы уведомлений получает, без необходимости открывать сам Wialon.

2. Задача

Построить промежуточный интеграционный сервис, который:

- подключается к произвольному числу компаний-клиентов Wialon Local независимо друг от друга;
- регулярно опрашивает сработавшие уведомления по каждой компании;
- определяет получателей внутри компании исходя из назначенных им прав на конкретные типы уведомлений;
- доставляет сообщения в Telegram и/или MAX, в зависимости от того, где пользователь зарегистрирован;
- даёт администратору единую панель управления компаниями, пользователями, правами и состоянием доставки;
- переживает временную недоступность Wialon, Telegram или MAX, не теряя события и не задерживая отправку.

Для решения этой задачи создан **Wialon Message Hub** (внутреннее техническое название — Wialon Message Gateway, WMG).

3. Архитектура решения

Wialon Message Hub — четыре взаимодействующих процесса на общей базе PostgreSQL:

- **Web UI** (FastAPI + Jinja2 + HTMX) — административная панель: компании, пользователи, права, журнал событий, диагностика подключений;
- **Collector** — фоновый процесс, по расписанию опрашивающий Wialon Local каждой активной компании и складывающий новые события в базу;
- **Dispatcher** — фоновый процесс, раскладывающий новые события по получателям согласно их правам и отправляющий сообщения в Telegram/MAX, с повторными попытками при временных сбоях;
- **PostgreSQL** — единственное общее хранилище состояния между всеми процессами.

Все четыре компонента развёрнуты как отдельные Docker-контейнеры за общим nginx-прокси. Подробная схема — в документе «Техническое описание архитектуры».

4. Как работает Wialon Message Hub

1. Администратор подключает компанию: указывает адрес сервера Wialon Local и учётные данные (токен или логин/пароль), при необходимости — токены Telegram- и MAX-ботов.
2. Collector этой компании авторизуется в Wialon, периодически запрашивает список объектов и сработавшие уведомления за прошедшее с последнего опроса время, пишет новые события в базу с защитой от повторной записи.
3. Dispatcher находит новые события, определяет их тип, ищет пользователей компании, которым назначено право на этот тип уведомления и которые зарегистрированы и активны в Telegram или MAX, формирует текст сообщения (с тематической иконкой и, если у события есть координаты, — картой местоположения) и ставит в очередь отправки.
4. Очередь обрабатывается с повторными попытками по расписанию при временных сбоях канала доставки; окончательные ошибки (например, пользователь заблокировал бота) фиксируются без бесконечных повторов.
5. Пользователь регистрируется в канале доставки самостоятельно, командой `/start` в соответствующем боте; администратор одобряет регистрацию и назначает нужные типы уведомлений через веб-панель.

5. Работа с несколькими компаниями

Одна инсталляция Wialon Message Hub одновременно обслуживает произвольное число независимых компаний. У каждой — собственные учётные данные Wialon, собственные Telegram- и MAX-боты, собственный справочник типов уведомлений, собственные пользователи и права, отдельно управляемые статусы сбора (Collector) и рассылки (Dispatcher). Данные компаний друг от друга изолированы на уровне запросов к базе.

Отдельно стоит случай, когда один и тот же MAX-бот обслуживает сразу несколько компаний (что в MAX технически возможно, в отличие от Telegram) — в этом случае регистрация пользователя привязывается к конкретной компании кодом, который администратор задаёт в свойствах компании.

6. Интеграция с Wialon Local

Подключение к Wialon Local выполняется по Remote API (HTTP/JSON): основной способ авторизации — постоянный API-токен (`token/login`), запасной — логин/пароль (`login/login`). Токен и пароль хранятся в базе в зашифрованном виде.

Для случаев, когда у пользователя Wialon нет отдельного доступа по API (частая ситуация на локальных инсталляциях), реализовано автоматическое получение токена по логину и паролю через управляемый headless-браузер, повторяющий обычный веб-вход — это единственный надёжный способ для тех версий Wialon Local, где страница входа формирует служебные поля через JavaScript и прямой HTTP-запрос к API входа недоступен пользователю.

7. Collector

Collector — процесс, отдельно для каждой активной компании авторизующийся в Wialon и с заданным интервалом запрашивающий сработавшие уведомления (`messages/load_interval`) по каждому объекту компании. Точкой отсчёта служит момент последней успешной синхронизации; при первом подключении компании она задаётся администратором явно. Между опросами окно намеренно немного перехлестывается, чтобы не потерять события на границе, а повторная запись исключается уникальным ключом события в базе.

Ошибка по одному объекту или временная недоступность Wialon не останавливают сбор по остальным объектам и компаниям — Collector восстанавливает сессию автоматически на следующем цикле.

8. Обработка событий

Каждое сработавшее уведомление сопоставляется со справочником типов уведомлений компании (синхронизируется отдельно из Wialon), формируется читаемый текст с тематической иконкой (по ключевым словам в названии уведомления — слив, заправка, скорость, геозона, тревога, потеря связи и т. д.), и, если у события есть географические координаты, — отдельно отправляется карта с меткой местоположения.

9. Dispatcher

Dispatcher раскладывает новые события по получателям (пользователям, у которых есть право на конкретный тип уведомления и активная регистрация в канале) и отправляет сообщения. Очереди Telegram и MAX полностью независимы: пользователь, зарегистрированный в обоих каналах, получит сообщение дважды, независимо друг от друга. При временных ошибках канала (ограничение частоты отправки, недоступность API) применяется расписание повторных попыток с нарастающей задержкой; окончательные ошибки (заблокированный бот, несуществующий чат) фиксируются без повторов.

10. Telegram

Каждая компания подключает собственного Telegram-бота (токен создаётся штатным способом через @BotFather и вводится в панели). Пользователь запускает бота командой /start, при необходимости указывает код регистрации компании, после чего администратор одобряет регистрацию и назначает права на типы уведомлений. Бот поддерживает управляющую клавиатуру (статус, справка, пауза/возобновление получения сообщений) и текстовые команды.

11. MAX

Интеграция с MAX реализована по протоколу MAX Bot API (собственный HTTP-клиент, long-polling получение сообщений). Логика регистрации пользователя и назначения прав зеркальна Telegram, но реализована отдельно — с учётом того, что один MAX-бот, в отличие от Telegram, может технически обслуживать сразу несколько компаний; в этом случае регистрация обязательно требует кода компании.

12. Управление пользователями

В веб-панели пользователи Telegram и MAX компании отображаются на отдельных вкладках. Администратор видит статус регистрации, может одобрить или заблокировать пользователя, включить/выключить получение сообщений независимо от статуса регистрации, переименовать (задать понятное отображаемое имя) и настроить его права.

13. Управление типами уведомлений

Справочник типов уведомлений формируется автоматически из настроек Wialon компании и отображается в панели переключателями «включено/выключено» — это управляет тем, учитывается ли тип уведомления при раскладке новых событий по получателям, независимо от прав конкретных

пользователей.

14. Контроль доставки

Для каждого события в журнале виден статус (новое/отправлено/ошибка), число успешно отправленных сообщений и число ошибок по всем получателям этого события. Администратор может в любой момент проверить подключение к Wialon, Telegram и MAX отдельными кнопками и выполнить внеплановую синхронизацию.

15. Журнал событий

Раздел «События» — таблица всех собранных из Wialon событий компании с фильтрами по датам, объекту, статусу и тексту. Отдельный общий «Журнал» на верхней панели фиксирует системные события (ошибки авторизации, сбои сети, результаты синхронизации) по всем компаниям.

16. Надёжность

Ошибка одной компании, одного объекта или одного получателя не блокирует работу остальных — изоляция ошибок реализована на всех уровнях (Collector — по объекту и по компании, Dispatcher — по одному сообщению в очереди). Повторная запись одного и того же события в базу исключена уникальным ограничением. Временные сбои каналов доставки обрабатываются повторными попытками по расписанию.

Полностью автоматической готовности к сбоям сервера в текущей версии нет: резервное копирование базы данных не реализовано (см. документ «Выявленные проблемы и рекомендации»), это открытый пункт для дальнейшего развития.

17. Безопасность

Учётные секреты (пароли и токены Wialon, Telegram, MAX) хранятся в базе в зашифрованном виде, ключ шифрования — только в конфигурации сервера. В интерфейсе секреты показываются только частично замаскированными. Пароль администратора хранится хэшированным, необратимо. Текущая эксплуатируемая версия работает по HTTP без TLS-сертификата — этот и ряд других пунктов безопасности разобраны отдельно в закрытом техническом разделе для владельца решения.

18. Возможности дальнейшего развития

- Резервное копирование базы данных и ключа шифрования по расписанию.
- Полноценная кнопка «Повторить отправку» в интерфейсе (серверная функция реализована, элемент управления в панели — нет).
- Множественные учётные записи администраторов с ролями.
- TLS/HTTPS для публичного доступа к панели.
- Подключение дополнительных каналов доставки по уже заложенной архитектуре (раздельные транспортные таблицы для каждого канала).

19. Итог

Wialon Message Hub закрывает разрыв между тем, что умеет формировать Wialon Local, и тем, каким каналом связи реально пользуются сотрудники заказчика. Решение уже эксплуатируется на

нескольких компаниях с реальной доставкой в Telegram и MAX, устойчиво к типовым сбоям интеграции (протухание сессии Wialon, временная недоступность каналов доставки, ошибки по отдельным объектам) и спроектировано так, чтобы подключение нового канала доставки в будущем не требовало переработки ядра системы.