

# Архитектура

- исходная архитектура потоков воркфлоу
- список функций для Автоюрист/Аварком Чат-Ассистент Voronka
- автоюристы бизнес процесс
- mvp автоюрист

# исходная архитектура потоков воркфлоу

Ниже — архитектура “минимально жизнеспособного” агент-бота в n8n для MAX → Voronka CRM → (OCR/документы) → Google Calendar → уведомления из CRM. Я опишу **какие workflow нужны, какие ноды в каждом, где хранить состояние, и как сделать подключение Google Calendar на пользователя** так, чтобы это масштабировалось (один бот, много пользователей/чатов).

---

## 0) Базовые принципы, чтобы система не развалилась

### 1. LLM = только “интент + сущности + вопросы”

Все реальные действия (создать клиента, создать встречу, прикрепить паспорт) выполняются **детерминированными нодами** (HTTP Request/DB/CRM API). Это стандартный мировой паттерн “LLM as router / extractor”.

### 2. Состояние диалога хранится вне n8n

n8n хорошо оркестрирует, но плохо хранит долгие “слоты” и контекст. Поэтому:

- быстрые ожидания/TTL: **Redis**
- долговременное: **Postgres** (у тебя уже есть)

### 3. Один бот — много пользователей/чатов ⇒ нужна таблица идентификации

Минимально: `max_user_id`, `chat_id`, `tenant_id / vclient`, `node_vpn_ip`, роли/права, токены Google.

### 4. Файлы и паспорта: всегда “сырьё” в MinIO + ссылкой дальше

Не гоняем большие байты через десяток нод. Сохранил → получил `s3: //bucket/key` → передал дальше.

---

## 1) “Минимальный набор workflow” в n8n (на перспективу)

# WF-1. MAX Inbound Router (единая точка входа)

**Цель:** принять любое сообщение (текст/фото/файл), определить пользователя/чат, понять намерение и запустить нужную ветку.

Минимальные ноды:

1. **Trigger:** MAX Trigger (если используешь community node) или Webhook node (MAX подписка на вебхуки: HTTPS, допустимые порты — у MAX это требование) ([MAX для разработчиков](#))
2. **IF:** проверка `x-max-bot-api-secret` (как ты уже видишь в заголовках)
3. **Code/Set:** нормализация входа в единый формат:
  - `chat_id`, `max_user_id`, `message_id/mid`, `text`, `attachments[]`
4. **Postgres:** `tenant_resolve` (по `max_user_id/chat_id` найти vclient/node/права)
5. **Switch:** по типу входа:
  - `text` → в LLM-парсер
  - `attachments` → в файловый конвейер
6. **LLM (DeepSeek):** твой системный промпт → вернуть JSON-команду
7. **Parse JSON (Code):** распарсить строку → объект
8. **IF:** `missing_fields`? → ветка “уточнить”
9. **Switch** по `action` → дергаем “WF-х исполнители”
10. **Отправка ответа в MAX** (Send Message)

Шаблон/пример для MAX-бота в n8n уже есть: “AI chatbot for Max Messenger ...” (там ещё и voice) — можно импортировать и выкинуть лишнее. ([n8n](#))

---

## WF-2. File Ingest (фото/паспорт/PDF/DOCX/XLSX/txt)

**Цель:** скачать вложение, положить в MinIO, получить текст/данные (OCR/парсинг), вернуть в WF-1 “готовую сущность”.

Ноды:

1. **HTTP Request:** скачать файл по `payload.url`
2. **MinIO/S3 node или HTTP:** положить в bucket (например `crm-inbox-raw`)
3. **Router по типу:**
  - image → OCR microservice

- pdf/docx/xlsx/txt → document parser microservice
4. **HTTP Request:** OCR/парсер → получить structured text
  5. **Return:** упаковать результат как “input.text + input.file\_ref” и передать обратно в LLM или сразу в действие `recognize_passport/create_client_from_passport/...`
- 

## WF-3. CRM Action Executor (создать/найти/обновить клиента, прикрепить документы)

**Цель:** единый исполнитель действий с Voronka.pro (через API/вебхук), чтобы WF-1 не разрастался.

Ноды:

1. **HTTP Request:** `api.voronka.pro` (resolver + маршрутизация на node1 по VPN)
  2. **HTTP Request:** вызов CRM API на конкретной ноде (внутренний VPN адрес)
  3. **Postgres:** запись связок (например `client_id`, `passport_doc_id`, `source_mid`)
  4. **Return:** результат (например номер клиента)
- 

## WF-4. Google Calendar Connect (OAuth-привязка календаря к пользователю)

**Цель:** дать пользователю ссылку, он авторизуется Google, и ты получаешь refresh token.

Почему отдельный WF: multi-user OAuth в n8n не “из коробки” удобен — это частая тема, люди делают “dynamic credentials / broker” подход. ([n8n Community](#))

Ноды:

1. **Trigger:** из WF-1 (команда “подключить календарь”)
2. **HTTP Request:** к твоему Laravel `api.voronka.pro` → “создать OAuth-линк” (state = max\_user\_id + nonce)
3. **Send Message (MAX):** отправить ссылку пользователю
4. **Webhook (callback):** (лучше пусть callback принимает Laravel, а не n8n)  
Laravel сохранит refresh token и дернет **внутренний webhook** в n8n “calendar\_connected”
5. **Send Message (MAX):** “календарь подключен ”

(Можно и напрямую через n8n-credentials, но тогда ты либо вручную плодишь креды на каждого пользователя, либо строишь сложный обход — есть даже шаблон, который автоматизирует создание Google OAuth кредов в n8n, но это всё равно про n8n-credential storage. (n8n))

---

## WF-5. Create Meeting (Google Calendar + запись в CRM по клиенту)

**Цель:** команда “создай встречу ...” → событие в календаре пользователя → запись/комментарий/активность в CRM в карточке клиента.

Ноды:

1. **Input:** из WF-1 (action = schedule\_meeting)
  2. **Postgres:** взять “Google refresh token” пользователя
  3. **HTTP Request:** к твоему “OAuth broker” (Laravel) → получить access token (или прокси-вызов календаря)
  4. **Google Calendar:** Create Event (или HTTP Request в Calendar API)
  5. **HTTP Request:** в CRM (создать Activity/Comment/Task с ссылкой на event/meet link)
  6. **Send Message (MAX):** подтверждение + дата/время
- 

## WF-6. CRM → Bot Notifications (напоминания/чек-листы)

**Цель:** CRM по событию шлет webhook → n8n отправляет человеку/в группу задачи.

Ноды:

1. **Webhook:** входящий от CRM
  2. **Postgres:** найти `chat_id`/пользователей, кому слать
  3. **Send Message (MAX):** текст + кнопки (если используешь callbacks, MAX это поддерживает) (MAX для разработчиков)
- 

## WF-7. Error & Audit (чтобы не забивать логи, но видеть ошибки)

**Цель:** ошибки в любом workflow → в один канал (лог-таблица + уведомление админам).

Ноды:

1. **Error Trigger** (глобальный)
  2. **Postgres:** записать кратко (workflow, node, error, mid, tenant)
  3. **Send Message (MAX) / Email:** только критичные (по фильтру)
- 

## 2) Что лучше “не писать с нуля”: ГОТОВЫЕ ОСНОВЫ/ШАБЛОНЫ

### 1. MAX + AI-бот шаблон n8n

Есть готовый workflow именно под MAX (включая voice). Его можно взять как “скелет”, оставив только MAX Trigger/Send + твою бизнес-логику. ([n8n](#))

### 2. Шаблоны “бот + календарь/бронь”

У n8n много готовых booking-workflow под Google Calendar (проверка слотов, создание события, подтверждения). Их можно адаптировать под твою команду “создай встречу”. ([n8n](#))

### 3. Каталоги шаблонов (можно импортировать JSON и менять)

- Категория “AI Chatbot” в n8n (сотни шаблонов) ([n8n](#))
- `awesome-n8n-templates` (огромный набор JSON) ([GitHub](#))
- архив официальных n8n workflows (удобно версионировать/таскать оффлайн) ([GitHub](#))

### 4. Community node для MAX

Чтобы не собирать руками HTTP-вызовы: `n8n-nodes-max`. ([GitHub](#))

(И вообще n8n официально описывает установку community nodes и их режимы) ([docs.n8n.io](https://docs.n8n.io))

---

## 3) Первая “боевая” настройка: подключение Google Calendar пользователем

# Почему я предлагаю OAuth через твой Laravel (api.voronka.pro), а не “чисто n8n”

Потому что у тебя **один бот и много пользователей**, и каждому нужен **свой** Google доступ. В n8n это упирается в “dynamic credentials / multi-tenant OAuth” (у сообщества это отдельная боль, есть обсуждения и обходы). ([n8n Community](#))

Laravel-broker решает это красиво: **хранит refresh token на пользователя**, а n8n всегда просто запрашивает “дай access token для max\_user\_id”.

## Минимальный user-flow

1. Пользователь в MAX пишет: **“Подключить календарь”**
2. WF-1 распознаёт intent → вызывает WF-4
3. WF-4 запрашивает у `api.voronka.pro` ссылку Google OAuth (с state = max\_user\_id + одноразовый nonce)
4. Бот отправляет ссылку в чат
5. Пользователь проходит Google consent
6. Laravel сохраняет refresh token (привязка к max\_user\_id/tenant\_id)
7. Laravel вызывает внутренний webhook n8n “calendar\_connected”
8. n8n отправляет пользователю: **“Google Calendar подключен ”**

## Что это даёт дальше

Когда пользователь попросит “создай встречу...”, WF-5:

- берет refresh token по max\_user\_id
- создаёт событие в календаре именно этого пользователя
- фиксирует связку в CRM/у клиента

---

## 4) Как это ложится на твою инфраструктуру (центральный сервер + node1)

- **n8n на центральном**: оркестрирует, хранит минимум, очереди/локи в Redis, аудиты в Postgres.
  - **api.voronka.pro (Laravel/MySQL)**: идеальное место для:
    - tenant resolver (max\_user\_id → vclient/node)
    - OAuth broker (Google tokens per user)
    - политика доступа (кто что может)
  - **node1 (CRM + базы клиентов)**: принимает только “чистые команды” от n8n (создать/найти/прикрепить документ/создать activity), по VPN адресу.
  - **MinIO**: raw-файлы/паспортные сканы/промежуточные результаты (с TTL/корзиной).
- 

## 5) Что делаем прямо сейчас, чтобы “встреча в календаре” стала реальной

Без кода, только шагами:

1. В WF-1 добавляешь команду **“подключить календарь”** (можно через LLM или просто if по тексту).
  2. Делаешь WF-4 “Google connect”:
    - нода HTTP Request → `api.voronka.pro` получить auth\_link
    - Send Message → отдать ссылку в MAX
  3. В Laravel добавляешь страницу OAuth (ты и так можешь, у тебя billing на Laravel) — она хранит refresh token и дергает n8n webhook.
  4. После “подключено” — запускаем WF-5: команда “создай встречу ...” → Create Event → ответ в MAX.
- 

Если ты скажешь, как ты хочешь решать **идентификацию пользователя с твоим tenant/vclient** (например: “бот доступен только авторизованным в billing, привязка по email/коду”), я предложу самый короткий onboarding-флоу (2–3 сообщения в MAX) и какие таблицы минимально нужны, чтобы дальше спокойно прикручивать CRM и паспорта.

# список функций для Автоюрист/Аварком Чат- Ассистент Voronka

## Список функций для автоюрист/аварком

управление через мессенджер

---

## Управление клиентами и карточкой в CRM

- **Создание клиента по сообщению** — напишите “создай клиента ...”, бот заведёт карточку и пришлёт номер клиента.
  - **Поиск клиента за секунды** — по телефону, имени, email или ИНН.
  - **Короткая карточка клиента в чате** — контакты, статус, последние действия, ближайшие задачи, ответственный.
  - **Фиксация результата общения одной фразой** — бот добавит комментарий в CRM и при необходимости предложит следующий шаг.
-

# Кейс/дело и работа “по ситуации” (особенно для выездных сотрудников)

- **Создание “кейса” (дела) по клиенту** — отдельная папка для конкретного события (например ДТП/объект/заказ), со всеми документами и сроками.
  - **Статус кейса в одном сообщении** — на каком этапе, что собрано, что дальше.
  - **Контроль полноты пакета** — бот подскажет, каких документов/данных ещё не хватает.
- 

# Документы, фото и распознавание

- **Приём фото/сканов/файлов прямо из чата** (PDF/DOCX/XLSX/TXT/изображения).
  - **Авто-определение типа документа** — паспорт, водительское, СТС/ПТС, акт, чек, договор и т.д.
  - **Распознавание паспорта и автозаполнение данных** — создание нового клиента или дополнение существующего.
  - **Прикрепление документов к клиенту и кейсу** — всё сохраняется в CRM, ничего не теряется.
  - **Фото “с места” → в CRM** — объект, товар, акт, чек, повреждения, и т.п.
- 

# Генерация документов и выдача “на месте”

- **Сформировать документ по шаблону** — заявление, опись, доверенность, договор, акт, бланк и т.п. (настраивается под вашу нишу).
- **Получить готовый PDF/DOCX в чат** — удобно открыть на телефоне и распечатать/переслать.

- **Отправить документ клиенту или в хранилище** — в чат, ссылкой, или в корпоративный архив (если подключён).
- 

## Задачи, календарь и контроль сроков

- **Быстрое создание задачи/напоминания** “на себя” или коллеге — “напомни завтра...”, “поставь задачу Ивану...”.
  - **Планирование встреч в Google Календаре** — бот создаёт событие в календаре пользователя и фиксирует запись в CRM у клиента.
  - **Автоматические цепочки сроков** — напоминания “через 20 дней”, “плюс 7 дней”, “проверить ответ” и т.п. (правила задаются в CRM).
  - **Отчёты в один клик** — “что у меня сегодня”, “задачи на неделю”, “кто без ответа 2 дня”.
- 

## Воронка, продажи и командная работа

- **Перевод сделки/заявки по стадиям** текстом (и в перспективе голосом).
  - **Подсказки “что делать дальше” по стадиям** — бот предлагает следующий шаг.
  - **Шаблоны быстрых действий кнопками** — позвонить, написать, добавить комментарий, поставить задачу, запланировать встречу.
  - **Работа в личных чатах и группах** — один на один или командой.
  - **Упоминания и назначение ответственных** — “@Иван, задача по клиенту...”.
- 

## Уведомления от CRM и контроль дисциплины

- **CRM сама присылает чек-листы и напоминания** по событиям и срокам — чтобы сотрудники не забывали действия.

- **Подтверждение выполнения** — отметка “сделано” прямо в чате (и фиксация в CRM).
- 

## Платежи и документы продаж (если подключено в CRM)

- **Счёт/ссылка на оплату по команде** — “выставь счёт на ...”, бот создаёт документ и даёт ссылку.
  - **КП/договор/прайс из чата** — отправили файл, бот прикрепил к клиенту/сделке и уведомил ответственного.
- 

## Безопасность и права доступа

- **Доступ по ролям** — сотрудник видит только то, что ему разрешено.
  - **Подтверждение опасных действий** — защита от случайных изменений в чате.
- 

## Перспектива: голосовой режим (необязательно для старта)

- **Приём голосовых сообщений** → перевод в текст → выполнение действий → ответ голосом.
  - **Озвучивание чек-листов и статуса кейса** для работы “за рулём”.
- 

Если хочешь, я могу сделать ещё одну версию — **короткую на 8-10 пунктов** (идеально для первого экрана лендинга) и **длинную** (для презентации/коммерческого предложения).

# автоюристы бизнес процесс

## Этап 0. Первичный контакт и квалификация

- Принять обращение (звонок/мессенджер), провести консультацию.
- Зафиксировать потребность клиента.
- Проверить, есть ли документы по ДТП, что уже оформлено (европротокол/справка/постановление и т.п.).
- Определить категорию дела: идти **в страховую, на виновника**, или **в страховую и на виновника**.
- Проверить уровень доверия/готовности клиента, уточнить «что надумал, что с документами, когда дальше действуем».

## Этап 1. Сбор документов (база)

- Паспорт собственника.
- СТС/ПТС.
- Водительское удостоверение (кто был за рулём).
- Страховки (если есть).
- Документы по аварии: европротокол / справка о ДТП / постановление-определение-протокол.
- Реквизиты собственника (чтобы не было истории “выплата почтой, потому что реквизитов нет”).

## Этап 2. Европротокол и подготовка пакета

- Если европротокол не заполнен — заполнить (в схеме отдельно: “если нет — заполнить”; “заполняем только нашу сторону”).
- Подготовить доверенность/варианты представительства (в схеме ветвления):

- собственник/водитель, доли/варианты, когда нужна **нотариальная доверенность** на получение денег.
- Понять, кто и как сканирует документы:
  - иногда страховой представитель “сам сканирует”,
  - иногда нужно **нотариально заверить** паспорт + СТС.

## Этап 3. Подача в страховую

- Заполнить заявление в страховую.
- Заполнить опись (перечень документов).
- Договориться о времени приёма документов / времени осмотра.
- Организовать осмотр ТС: согласовать дату и место, при необходимости присутствовать.
- Получить/сфотографировать акт осмотра (иногда “сразу не получили — получить позже”).
- Собрать “расходники”: эвакуатор, стоянка, разбор/подъём, стапель и т.д.
- Передать документы (в схеме есть вариант через курьера), получить квитанцию и копию описи.
- Сделать сканы всех документов и сохранить в диск (в вашей схеме прямо так и написано).

## Этап 4. Ожидание решения страховой (ключевой срок)

- Поставить ожидание **20 календарных дней** (“запланировать и ничего не делаем”).
- В это время страховая может назначать осмотр и уведомлять (звонок/телеграмма).
- Клиент должен сразу сообщать о контакте со страховой и “ничего не подписывать”.

## Этап 5. Решение страховой: деньги или ремонт, либо тишина

- Получить решение страховой.
- Варианты:
  - деньги (в т.ч. “могут отправить почтой на адрес собственника”),
  - направление на ремонт,
  - решения нет → подождать ещё **7 календарных дней**, дальше разбираться/эскалировать.
- Если ремонт: возможны сценарии “отказ СТО, чтобы страховая платила деньгами”, либо “если отремонтировали — отмена сделки”.

## Этап 6. Независимая экспертиза (если нужна)

- Уведомить клиента, что экспертиза платная, получить согласие.
- Сформировать стоимость услуг по экспертизе, выставить счёт (до/после осмотра — у вас оба варианта).
- Сформировать договор на экспертизу, собрать подписи.
- Эксперт делает расчёт ущерба и формирует экспертное заключение (в схеме есть файл списка запчастей/цен/кол-ва/стоимости).

## Этап 7. Суд на виновника (если клиент решил идти в суд)

- Принять решение с клиентом “будет ли суд на виновника”.
- Сформировать исковое заявление (шаблон → сформированный документ под редактирование).
- Организовать доверенность на представление в суде (шаблон отправить клиенту; клиент делает нотариальную доверенность).
- Госпошлина: клиент оплачивает, присылает скан чека; подготовить/распечатать.
- Собрать подтверждения выплат/получения страхового возмещения (выписка банка, платёжка/поручение).
- Подписать договор юр. сопровождения, выставить счёт, отследить оплату/рассрочку.
- Подготовить “3 конверта” и отправки Почтой России: два — сторонам, третий — в суд; сохранить чеки.
- Контроль регистрации и дальнейшие ветки:
  - отказ в регистрации → добрать документы,
  - получение даты заседания (в схеме “примерно через 6 месяцев”) → поставить в календарь,
  - довести запрошенные судом документы заранее,
  - участие в суде / перенос суда,
  - после суда ждать решение, получить решение.
- По итогам решения:
  - удовлетворили 100% / частично / отказ,
  - если клиента устраивает → передать решение судебным приставам для взыскания.

# mvp автоюрист

Ок, минимальный MVP под автоюриста/аваркома поэтапно так, чтобы уже после первых шагов человек мог **работать из машины** и реально экономить время.

## Этап 1. “Вход и привязка” (1 пользователь → 1 CRM)

**Цель:** бот понимает, кто пишет, и куда (в какую CRM/воронку) сохранять данные.

Что делаем:

- Команда «**Старт**» / «**Подключить CRM**» → бот привязывает пользователя (MAX user\_id / чат) к его рабочему аккаунту.
- Бот подтверждает: “Вы подключены, можно создавать кейсы ДТП”.
- Минимальные права: только свои клиенты/кейсы.

Результат: один бот обслуживает многих, но каждый управляет **только своими** данными.

---

## Этап 2. “Кейс ДТП за 30 секунд” (главное для аваркома)

**Цель:** фиксировать не просто “клиента”, а **дело/кейс ДТП**, потому что именно по нему идут документы, сроки, страховая, суд.

Что делаем:

- Команда «**Новое ДТП**» (или “клиент в ДТП”) → бот задаёт 3–5 коротких вопросов:
  - дата/время, место, авто/госномер (если надо), кратко что случилось
  - телефон клиента (если новый)
- Бот:
  - создаёт (или находит) клиента
  - создаёт **кейс ДТП** в CRM
  - возвращает **номер кейса** и “чек-лист что прислать”.

Результат: у автоюриста появляется “папка дела” и номер — дальше всё цепляется к кейсу.

---

## Этап 3. “Документы из чата → в кейс” (фото/сканы)

**Цель:** человек фоткает паспорт/права/СТС/европротокол и просто отправляет в чат.

Что делаем:

- Бот принимает **изображения и файлы** и автоматически понимает тип:
  - паспорт, водительское, СТС/ПТС, европротокол, постановление/справка, чеки эвакуатора/стоянки
- Бот:
  - складывает файл в хранилище
  - запускает распознавание
  - прикрепляет **и фото, и распознанные данные** к кейсу и клиенту
  - если клиент уже есть — добавляет в существующую карточку, если нет — создаёт нового.

Результат: **никаких “приеду — внесу в CRM”**. Всё попадает в дело сразу.

---

## Этап 4. “Сформировать документы и отдать на печать”

**Цель:** ваш главный “вау-сценарий”: прислал фото → получил готовый документ → распечатал на мобильном принтере.

Минимальный набор документов для MVP (самые частые и полезные):

1. **Европротокол (заполнение полей из распознанных документов)**
2. **Заявление в страховую**
3. **Опись/перечень документов**

Как это выглядит в работе:

- Команда «**Сформируй европротокол по кейсу №...**»

- Бот собирает данные, если чего-то не хватает — задаёт 1 короткий вопрос.
- Возвращает **PDF** (или DOCX) в чат и/или ссылку на диск.

Результат: автоюрист получает документ “здесь и сейчас” без ноутбука.

---

## Этап 5. “Сроки, календарь, напоминания” (чтобы ничего не забыть)

**Цель:** автоматом вести сроки и напоминать — в дороге это критично.

Что делаем:

- Подключение Google Calendar (один раз).
- Команда «**Запланируй осмотр/встречу по кейсу**» → бот создаёт событие в календаре и делает запись в CRM.
- Автозадачи по кейсу (минимально):
  - контроль решения страховой (20 дней)
  - “+7 дней” если ответа нет
  - напоминания о недостающих документах
- CRM сама шлёт в чат напоминания/чек-листы по наступлению событий.

Результат: бот становится “дежурным диспетчером” и держит процессы.

---

## Что получится после MVP

Автоюрист сможет **только с телефоном**:

- создать клиента и кейс ДТП,
  - собрать документы фотками,
  - автоматически распознать и прикрепить их в CRM,
  - получить готовые PDF документов для печати,
  - вести сроки через календарь и напоминания из CRM.
-

# Что оставляем на следующий этап (не в MVP)

- “Пакет в суд” (иск, конверты, треки, приставы) — это второй релиз.
- Интеграция с Яндекс.Диском как обязательный архив (в MVP можно, но лучше сначала стабильный MinIO/CRM).
- Голос: speech-to-text и text-to-speech.

Если хочешь, я сужу MVP ещё сильнее до **3 функций**, которые дают максимум пользы “здесь и сейчас” (обычно это: кейс ДТП → документы → генерация европротокола/заявления).