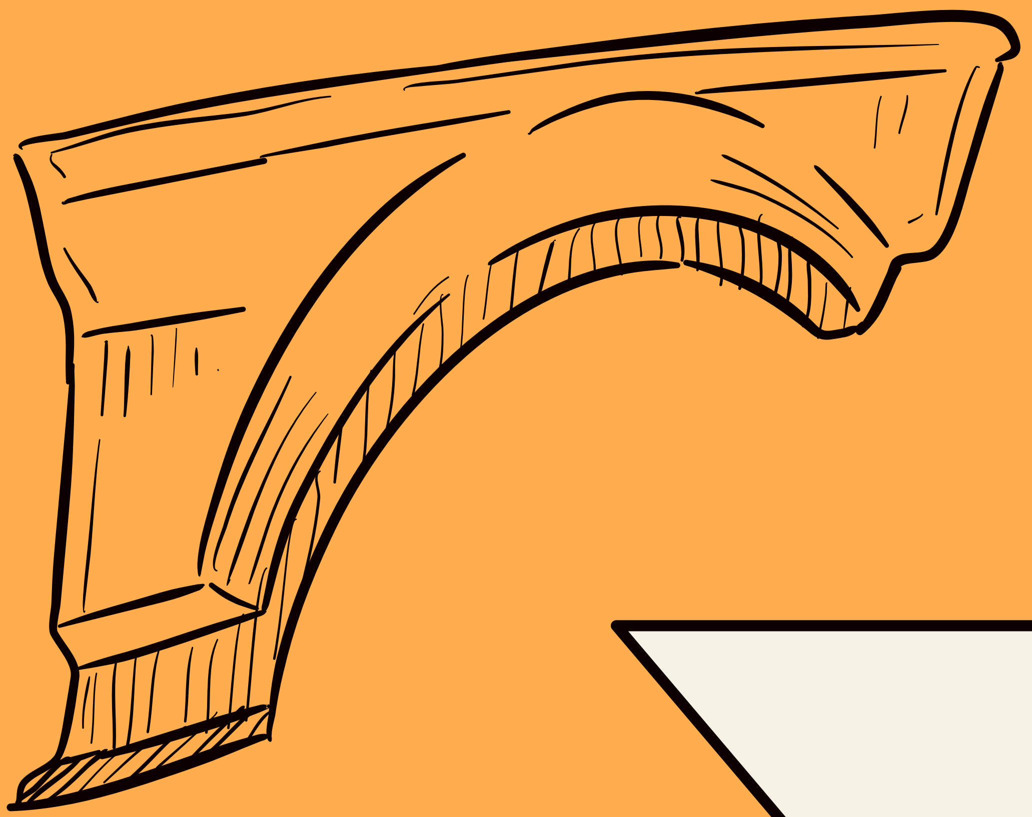
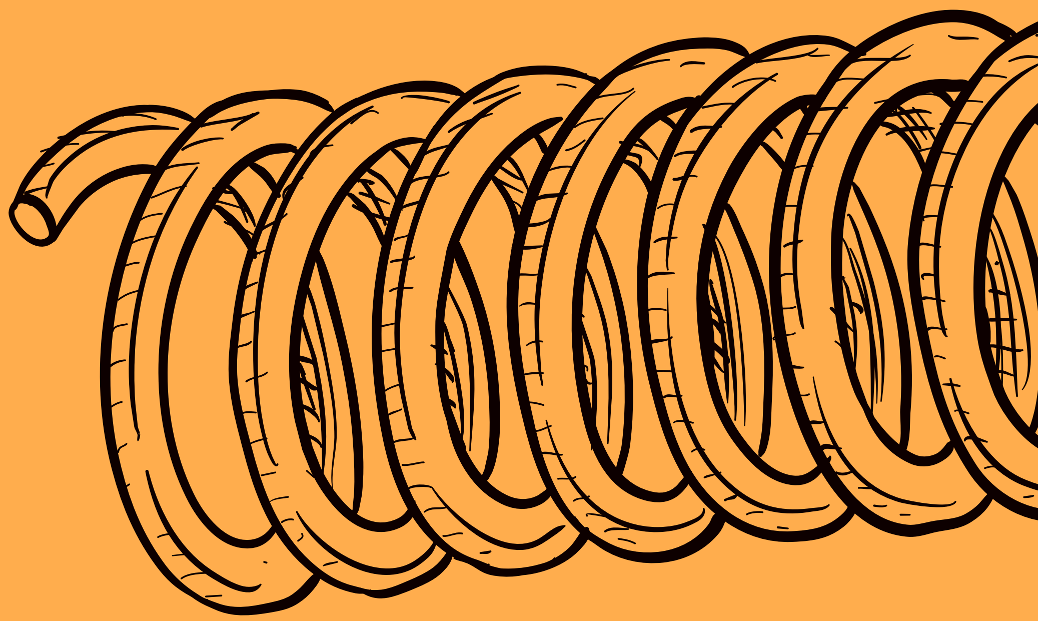
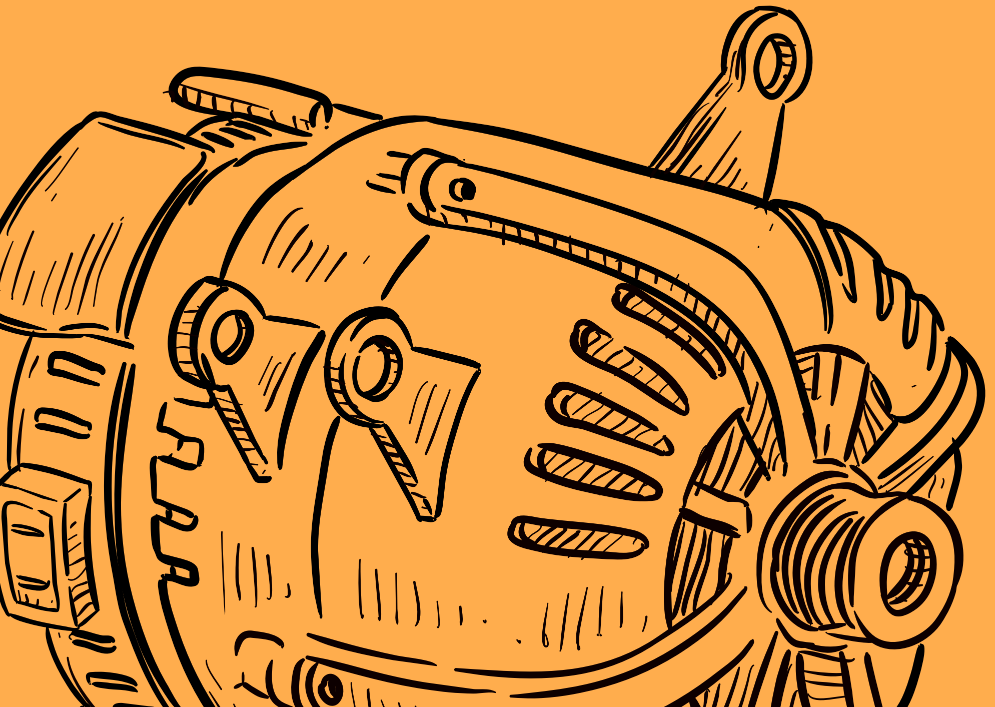
Ali Khani @ hiya

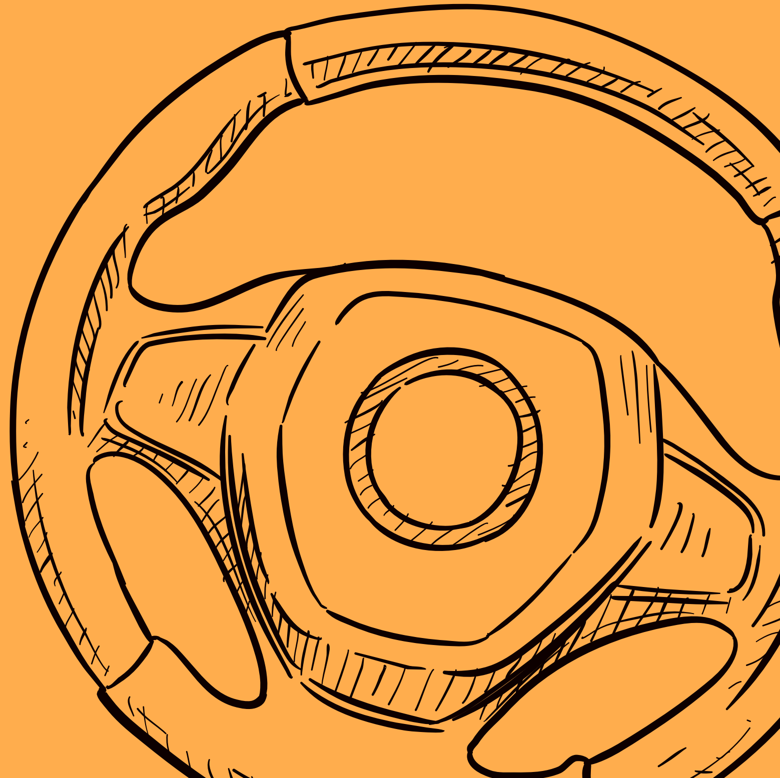




Otto's Auto

Real-Time Voice AI for Car Dealerships





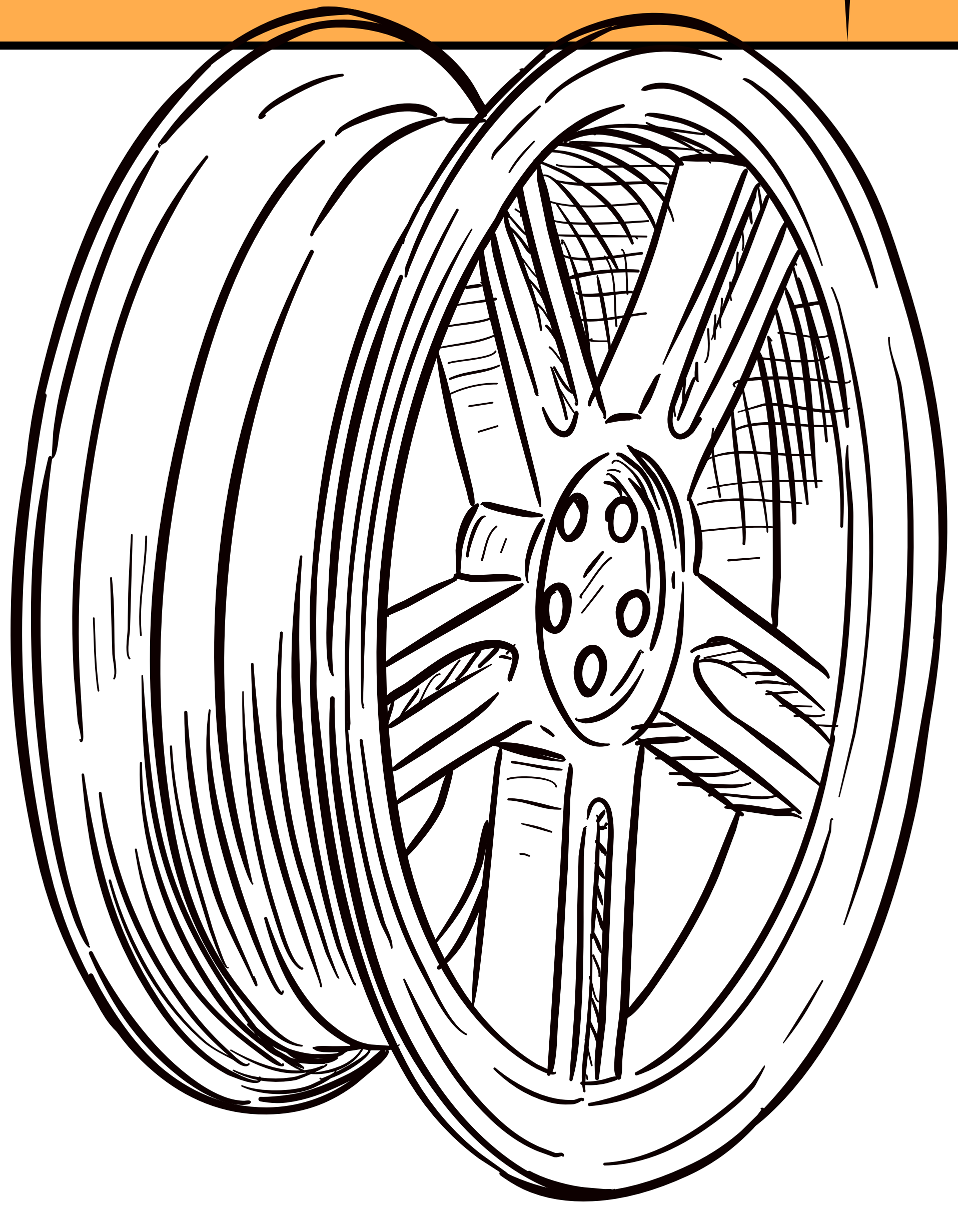
Customer Scenario

Problem:
- Otto's Auto gets 50+ daily scheduling calls, 40% of them after-hours
- Manual scheduling creates booking friction
- Staff spends 5-7 min/call on routine bookings

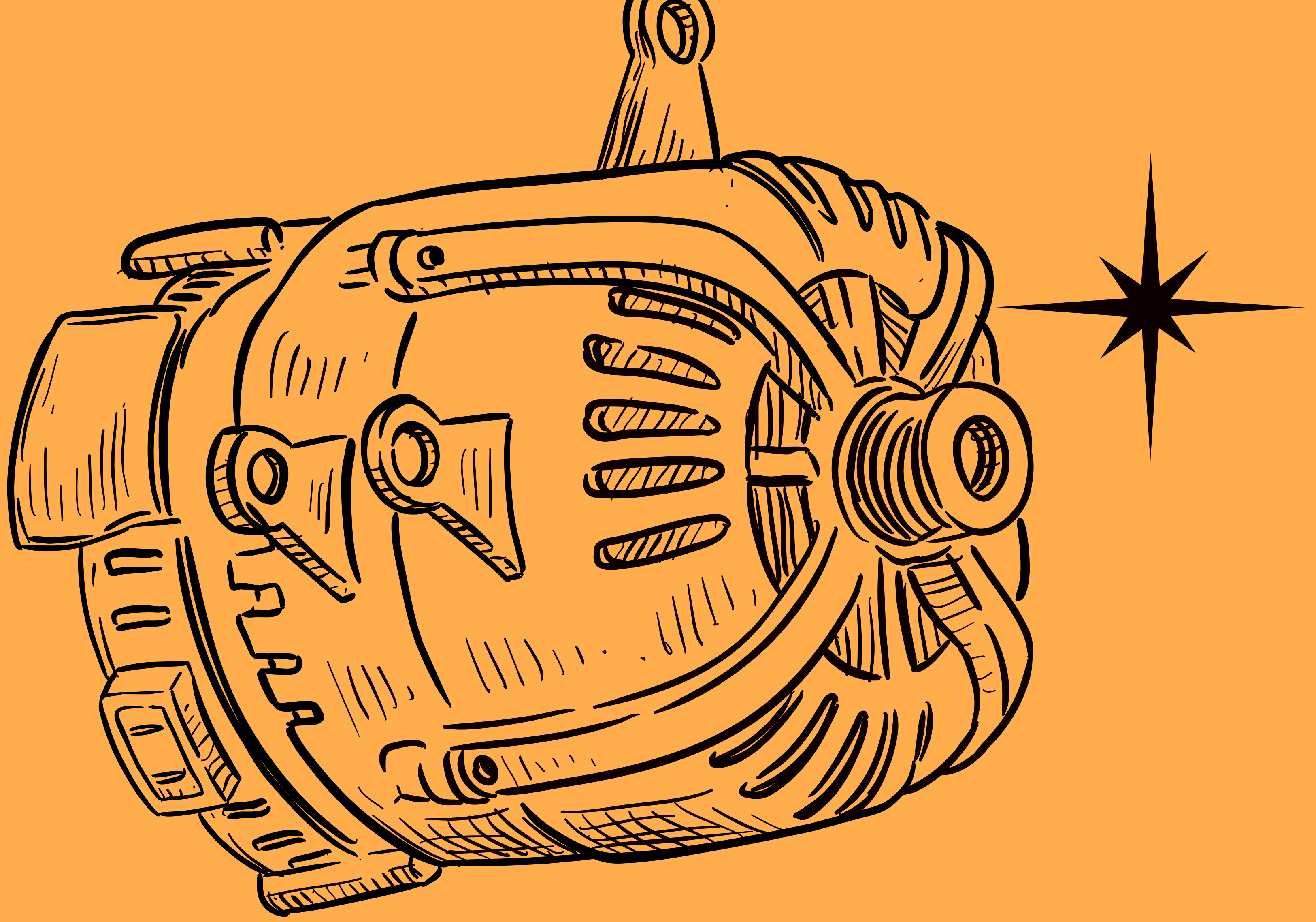
Solution:
- AI voice agent Sophie handles inbound bookings + CRM lookups + outbound reminders 24/7 with <2s latency

Target User:
- Existing customers calling to book oil changes, inspections, repairs

Value:
- 60% cost reduction, instant customer lookup in CRM
- Reduction in no-shows via automated reminders
- 24/7 availability to capture after-hours leads

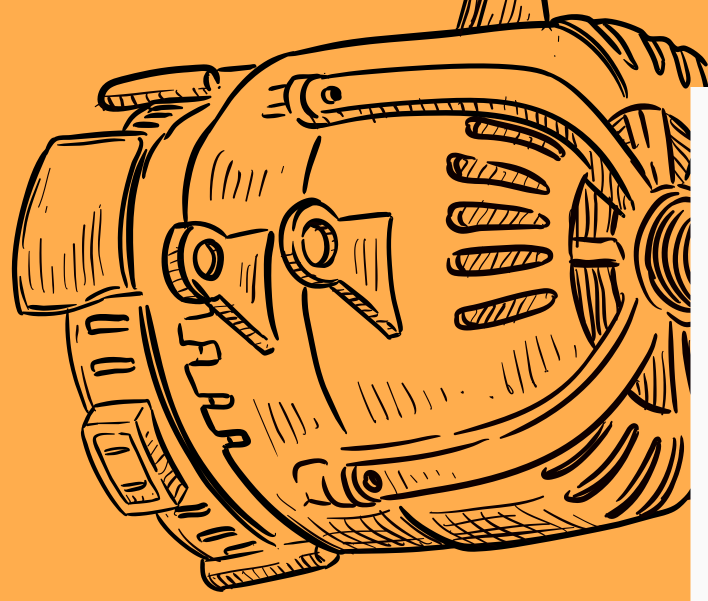


Technical Architecture

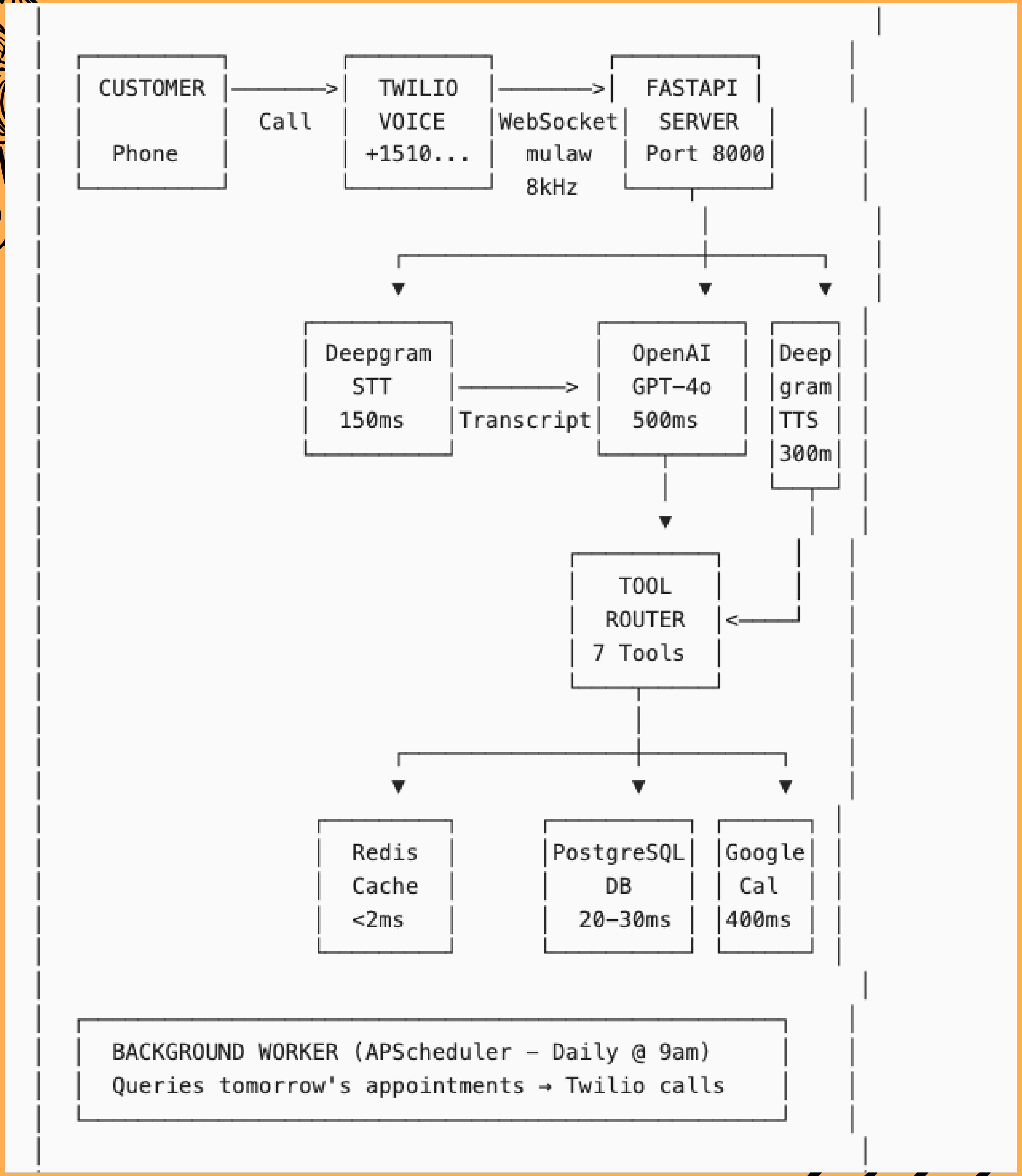


Key Components:

- **Voice Pipeline:** Twilio Media Streams + Deepgram STT & TTS
- **AI Engine:** OpenAI GPT-4o with 7 CRM tools, streaming responses
- **Data Layer:** Postgres (customers, vehicles, appointments) + Redis (2-tier cache)
- **Integrations:** Google Calendar (OAuth2, freebusy API), NHTSA VIN decoder
- **Background Worker:** APScheduler for 24hr outbound reminder calls



Technical Architecture



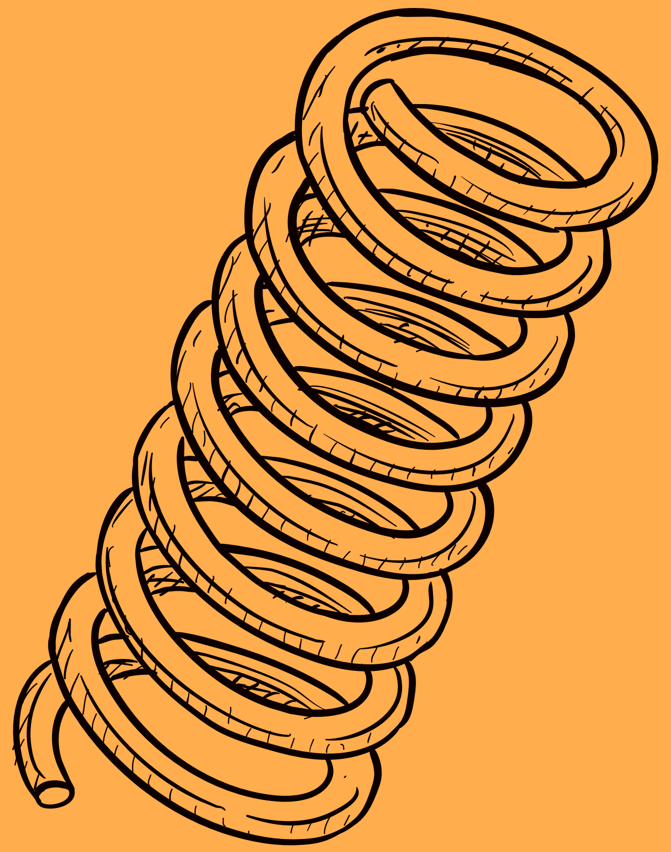
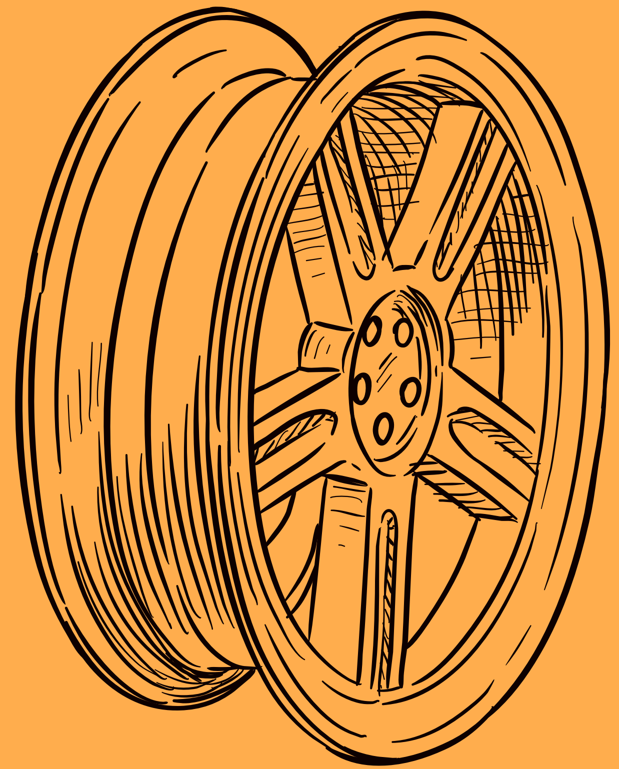
```
┌──────────┐          ┌──────────┐          ┌──────────┐
│ CUSTOMER │ ───────> │  TWILIO  │ ───────> │ FASTAPI  │
│          │   Call   │  VOICE   │WebSocket │  SERVER  │
│  Phone   │          │ +1510... │  mulaw   │Port 8000 │
└──────────┘          └──────────┘   8kHz   └────┬─────┘
                                                 │
                    ┌────────────────────────────┼────────┐
                    ▼                            ▼        ▼
             ┌──────────┐               ┌──────────┐ ┌────┐
             │ Deepgram │               │  OpenAI  │ │Deep│
             │   STT    │ ────────────> │  GPT-4o  │ │gram│
             │  150ms   │  Transcript   │  500ms   │ │TTS │
             └──────────┘               └────┬─────┘ │300m│
                                             │       └──┬─┘
                                             ▼          │
                                       ┌──────────┐     │
                                       │   TOOL   │     │
                                       │  ROUTER  │ <───┘
                                       │ 7 Tools  │
                                       └────┬─────┘
                                            │
                    ┌───────────────────────┼──────────┐
                    ▼                       ▼          ▼
             ┌──────────┐           ┌──────────┐ ┌──────┐
             │  Redis   │           │PostgreSQL│ │Google│
             │  Cache   │           │    DB    │ │ Cal  │
             │  <2ms    │           │ 20-30ms  │ │400ms │
             └──────────┘           └──────────┘ └──────┘

┌─────────────────────────────────────────────────────────┐
│ BACKGROUND WORKER (APScheduler - Daily @ 9am)           │
│ Queries tomorrow's appointments → Twilio calls          │
└─────────────────────────────────────────────────────────┘
```

Technical Architecture

Orchestration:

1. **`lookup_customer(phone)`** - 2-tier cache (Redis → PostgreSQL)
2. **`search_customers_by_name(first, last)`** - Partial match search
3. **`get_available_slots(date)`** - Real Google Calendar API (not mocked)
4. **`book_appointment(...)`** - DB + Calendar event creation with retry logic
5. **`get_upcoming_appointments(customer_id)`** - History with vehicles
6. **`cancel_appointment(id, reason)`** - DB + Calendar deletion
7. **`reschedule_appointment(id, new_time)`** - DB + Calendar update

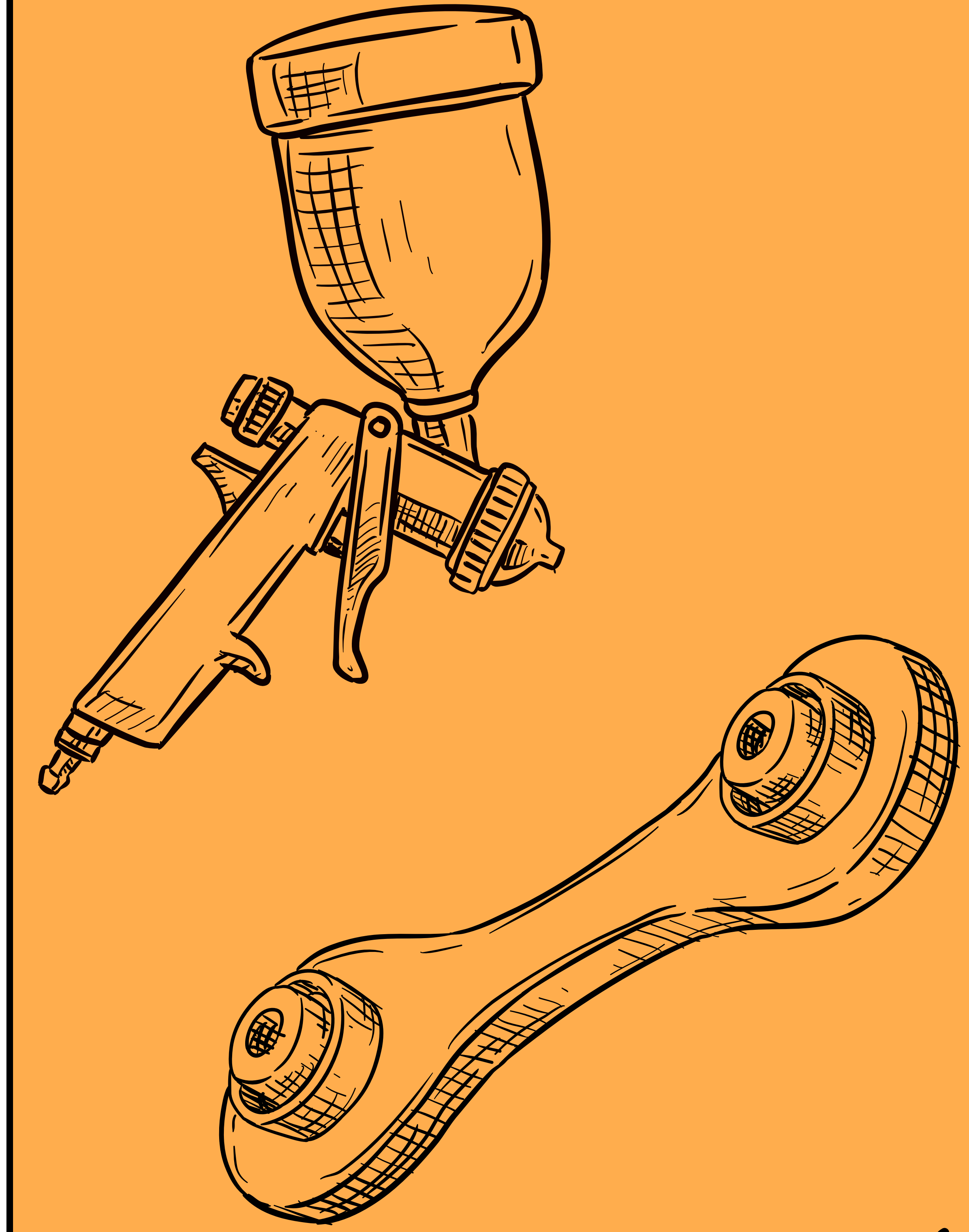
Technical Architecture

Error Handling

- **Retry logic for Google Calendar:** 3 attempts to auth, backoff afterwards.
- **Graceful degradation:** Redis failure → continue without cache; Calendar failure → use mock calendar fallback.
- **Barge-in detection:** Interim STT results → send “clear” to Twilio → process new incoming voice inputs.

State Management

- **Redis sessions** (1hr TTL) preserve context despite barge-ins.
- **GPT-4o conversation history** with auto-trim at 4k tokens.
- **State machine**: Hello → verification → intent → action → confirmation → close with automatic hang-up.



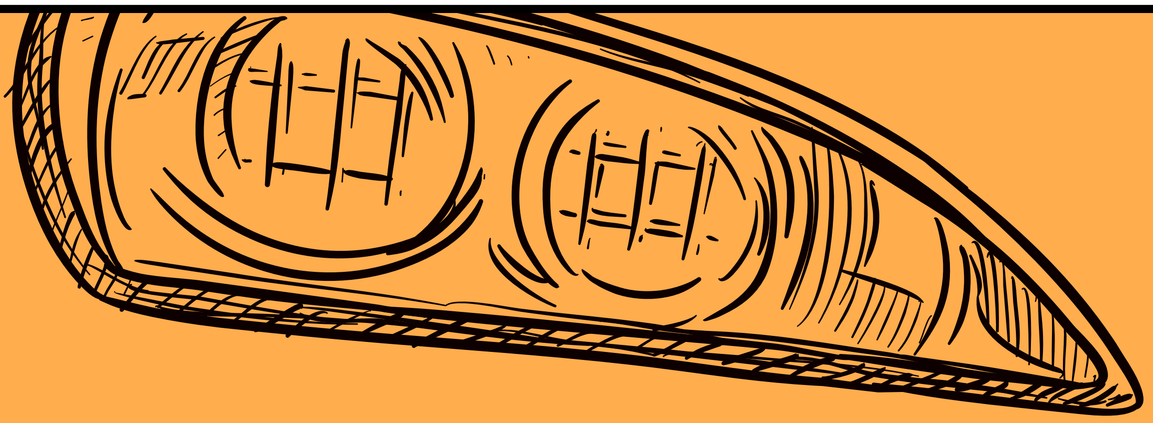
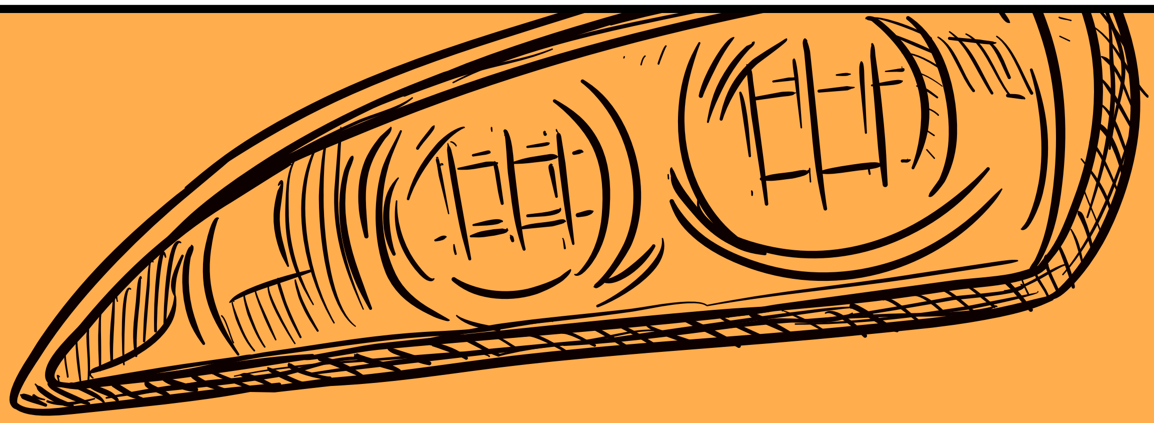
Technology Choices & Rationale

Orchestration Framework: Custom Python/FastAPI
- **Why:** Full control over WS streams, async/await for concurrency, production-ready
- **Alternatives:** LangChain (unnecessary abstraction overhead)

Voice Stack: Deepgram (both STT and TTS)
- **Why:** Optimized for telephony, <500ms latency, best handling of barge-ins/interruptions
- **Alternatives:** Google STT (higher latency), ElevenLabs TTS (too expensive for POC)

LLM: OpenAI GPT-4o with tool calling
- **Why:** Native tool support, streaming, 128k context, established reputation for voice AI
- **How:** 7 registered CRM tools (*lookup_customer*, *book_appointment*, *get_available_slots*, etc.)

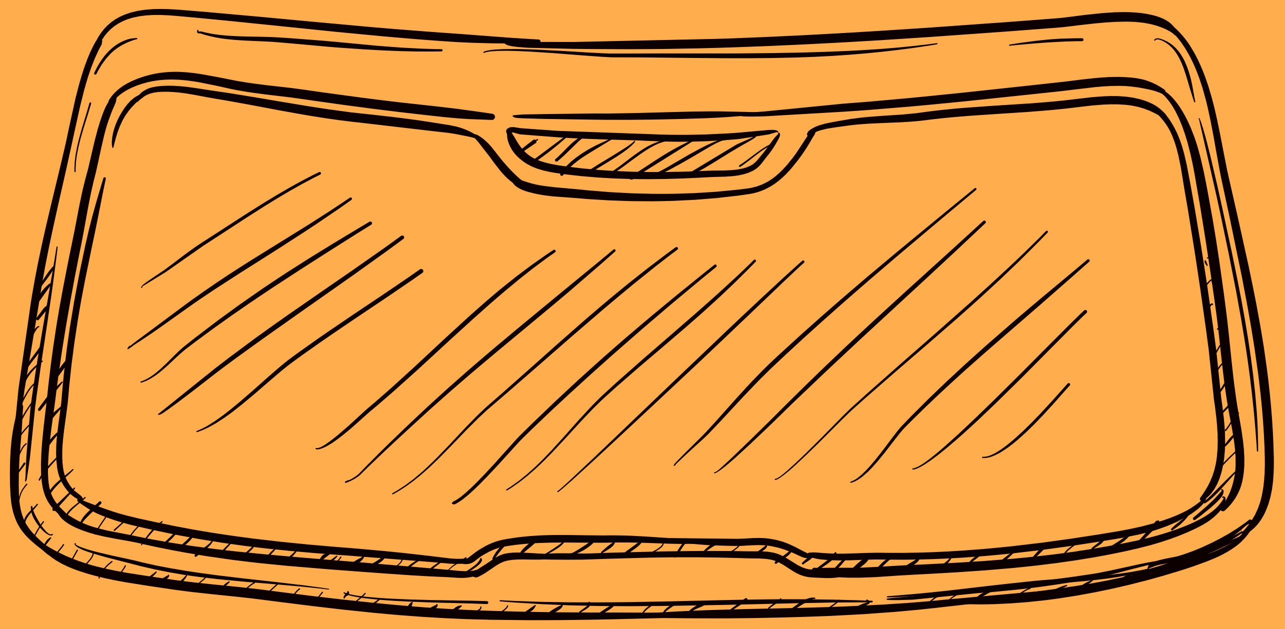
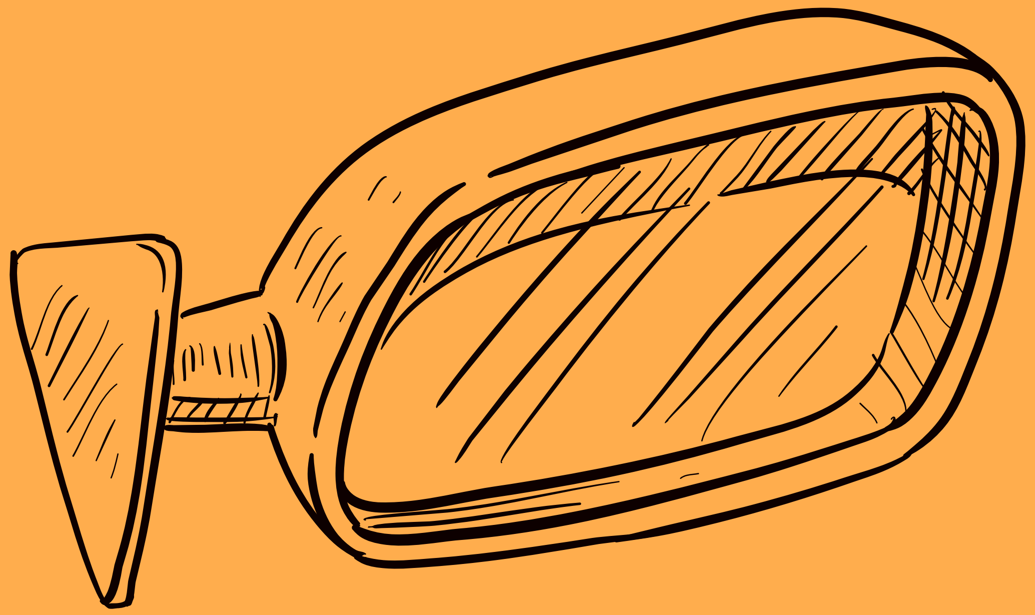
Database: Postgres (Neon) + Redis (Upstash)
- **Why:** Relational DB for customers/vehicles, and Redis for <2ms session lookup
- **Benefits:** ACID compliance, async SQLAlchemy ORM, serverless scaling down the line

POC Demos

What works:

1. **Inbound call simulation:** Customer lookup by phone, appointment booking flow
2. **7 CRM Tools:** real database integration
3. **Error handling**: Invalid inputs, missing customers, scheduling conflicts
4. **Conversation:** Multi-turn dialogues, barge-in detection, context retention over calls
5. **Security:** Prompt injection defense, session timeouts

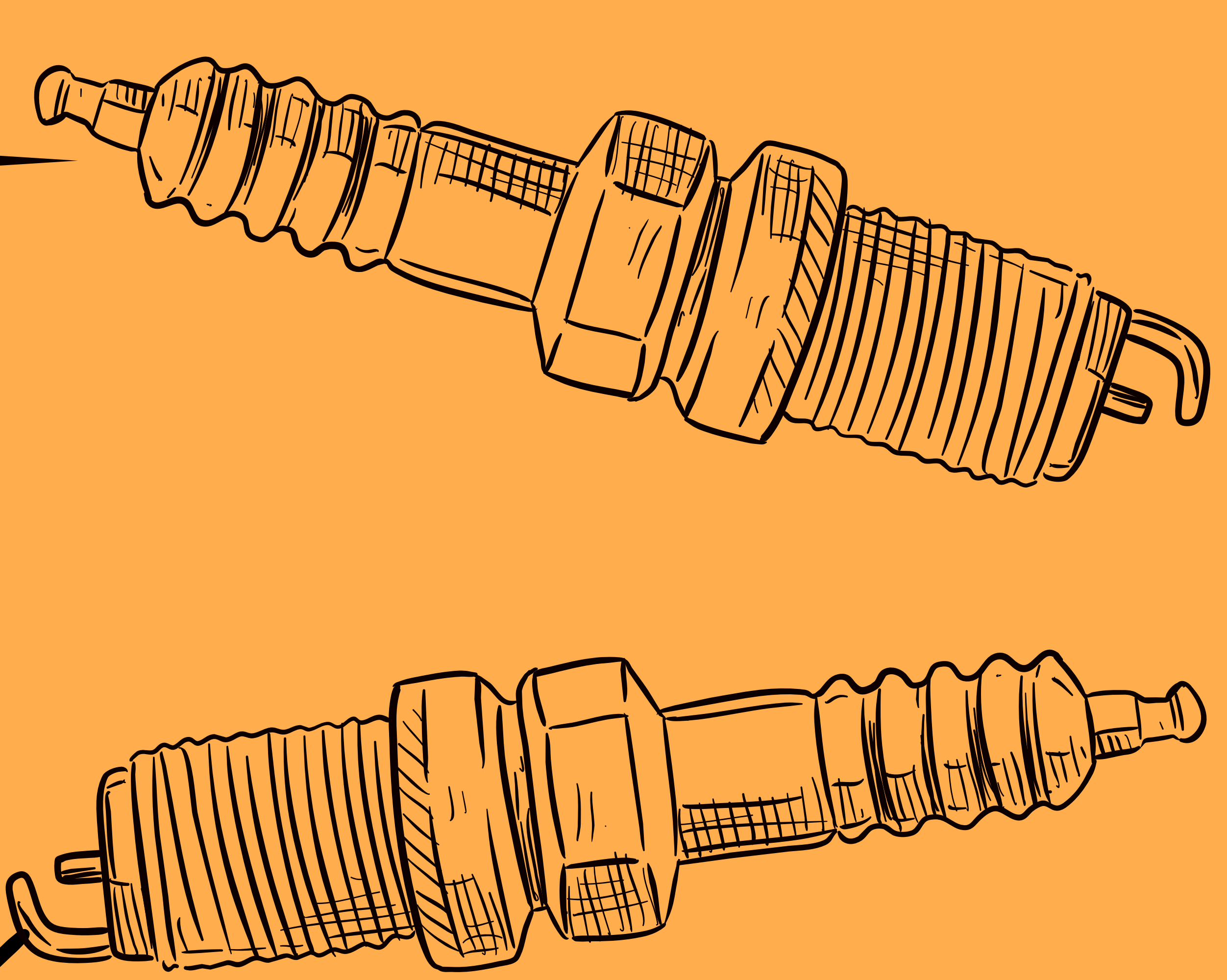



POC Demos

Demo: one-command local testing via run_test_environment.sh:

1. Auto-start Postgres, Redis, ngrok tunnel
2. Generate fresh ngrok URL & auto-config Twilio webhooks via CLI
3. Launch FastAPI server with seeded mock CRM data
4. Interactive menu to test inbound/outbound calls instantly

Limitations & Next Steps



Incomplete: Google Calendar OAuth

- Refresh token flow works locally but OAuth 2.0 redirect URI challenges
- Currently using mock calendar fallback for testing

Future enhancements:

1. **Integrations:** Plug into CRMS (Salesforce, Twenty, Attio) for live customer data. Email automations (AWS SES) for scheduled appointments.
2. **Audio quality:** ElevenLabs TTS for more natural conversations.
3. **DevEx:** Migrate Python → TypeScript for better SDK support (Twilio/Deepgram/OAI). Robust webhook validation & retry logic.
4. **Production:** Deploy to Railway/Render with managed Postgres & Redis. o11y for latency monitoring.
5. **Multi-language support:** Spanish for automotive market, voicemail transcription, callback scheduling.