

# Best Self AI - System Design Document

**Version:** 1.0 **Date:** February 2026 **Author:** Joseph Waine

---

## Table of Contents

- 1. [Overview](#)
  - 2. [Architecture Diagram](#)
  - 3. [Server Architecture](#)
  - 4. [Client Architecture](#)
  - 5. [External API Integrations](#)
  - 6. [Data Flows](#)
  - 7. [Database Schema](#)
  - 8. [Security Considerations](#)
  - 9. [API Reference](#)
  - 10. [Deployment](#)
- 

## 1. Overview

### 1.1 Purpose

Best Self AI is a full-stack voice coaching application that combines conversational AI, health data integration, and text-to-speech technology to provide personalized wellness guidance.

### 1.2 Key Features

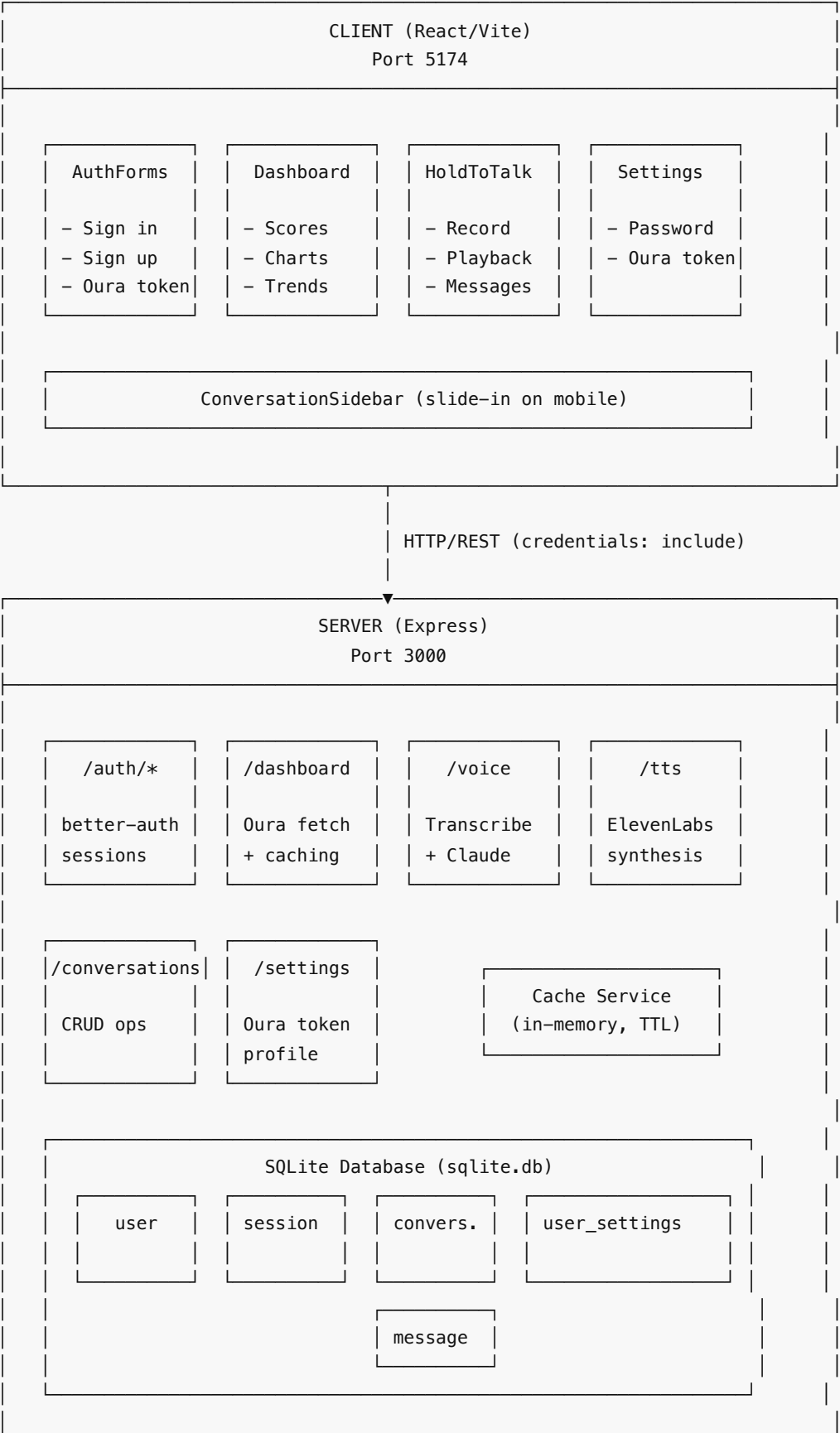
- Voice Interaction:** Hold spacebar to speak, receive spoken responses
- Health Data Integration:** Oura Ring data provides personalized context
- AI Coaching:** Claude AI generates contextual wellness advice
- Conversation History:** Persistent conversation storage and management
- Health Dashboard:** Visual display of sleep, readiness, activity, and vitals

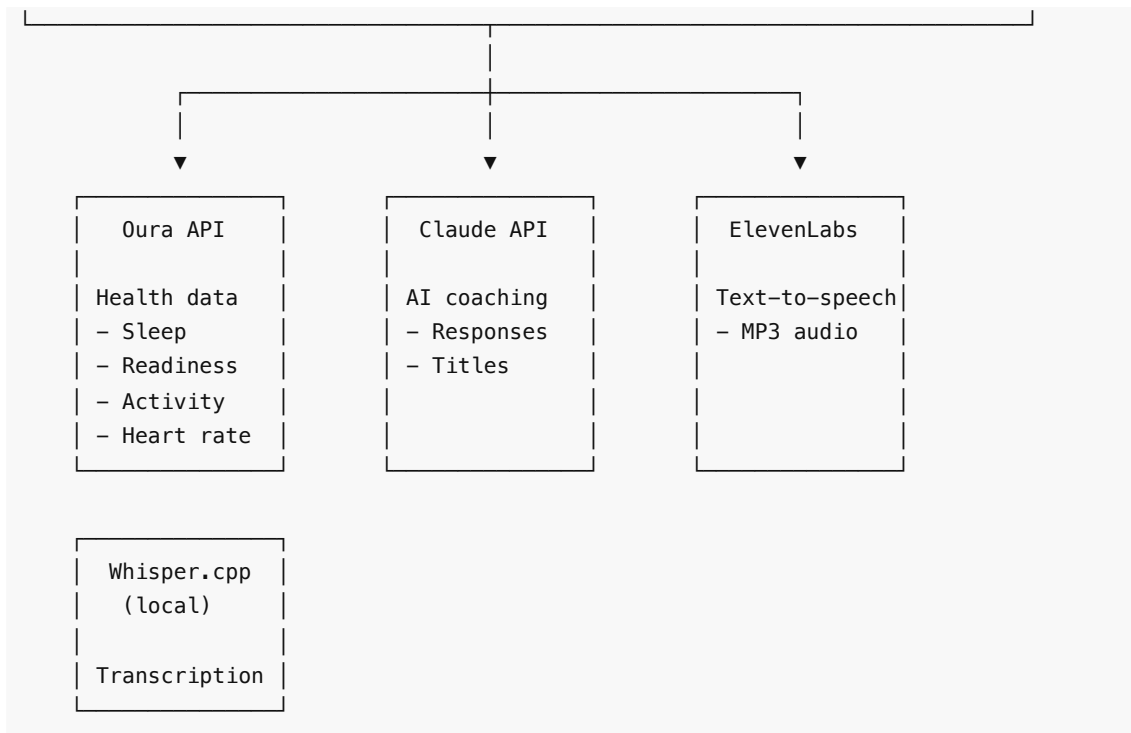
### 1.3 Technology Stack

Layer	Technologies
Frontend	React 19, TypeScript, Vite, Tailwind CSS, Recharts
Backend	Node.js, Express 5, TypeScript
Database	SQLite 3 (better-sqlite3)
Authentication	better-auth (cookie-based sessions)
AI/ML	Claude API (Anthropic), Whisper.cpp (local), ElevenLabs TTS
External APIs	Oura Ring API v2

---

## 2. Architecture Diagram





### 3. Server Architecture

#### 3.1 Directory Structure

```
server/
├── src/
│   ├── index.ts           # Express app setup, CORS, routes
│   ├── routes/
│   │   ├── auth.ts       # better-auth handler
│   │   ├── conversations.ts # CRUD operations
│   │   ├── dashboard.ts   # Oura data aggregation
│   │   ├── settings.ts    # User preferences
│   │   ├── tts.ts         # ElevenLabs integration
│   │   └── voice.ts       # Transcription + AI reply
│   ├── services/
│   │   ├── cache.ts       # In-memory TTL cache
│   │   ├── claude.ts      # Anthropic API client
│   │   ├── oura.ts        # Oura API client
│   │   ├── ouraSummary.ts  # Oura data aggregation
│   │   ├── storage.ts     # SQLite operations
│   │   ├── tts.ts         # ElevenLabs client
│   │   └── whisper.ts     # Local transcription
│   └── middleware/
│       └── auth.ts        # requireAuth middleware
├── sqlite.db             # Database file
└── package.json
```

#### 3.2 Core Server Configuration

```
// index.ts
const app = express();

// CORS: Allow client origins with credentials
app.use(cors({
  origin: ["http://localhost:5173", "http://localhost:5174"],
  credentials: true
}));

app.use(express.json());

// Routes
app.use("/api/auth", authRouter);
app.use("/api/conversations", requireAuth, conversationsRouter);
app.use("/api/dashboard", requireAuth, dashboardRouter);
app.use("/api/settings", requireAuth, settingsRouter);
app.use("/api/voice", requireAuth, voiceRouter);
app.use("/api/tts", ttsRouter);

app.listen(3000);
```

### 3.3 Authentication Middleware

```
interface AuthenticatedRequest extends Request {
  user?: { id: string; email: string; name?: string };
  session?: { id: string; userId: string };
}

async function requireAuth(req, res, next) {
  const session = await auth.api.getSession({ headers: req.headers });
  if (!session) return res.status(401).json({ error: "Unauthorized" });

  req.user = session.user;
  req.session = session.session;
  next();
}
```

### 3.4 Caching Strategy

Data Type	TTL	Cache Key Pattern
Today's dashboard	5 minutes	{userId}:dashboard:today:{date}
Weekly trends	30 minutes	{userId}:dashboard:week:{date}

```
// Cache service implementation
class Cache {
  private store = new Map<string, { data: any; expires: number }>();
```

```

get<T>(key: string): T | null {
  const entry = this.store.get(key);
  if (!entry || Date.now() > entry.expires) return null;
  return entry.data;
}

set<T>(key: string, data: T, ttlMs: number): void {
  this.store.set(key, { data, expires: Date.now() + ttlMs });
}
}

```

## 4. Client Architecture

### 4.1 Component Hierarchy

```

App.tsx
├─ AuthForms.tsx (when not authenticated)
├─ (when authenticated)
│   ├─ Header
│   │   ├─ Menu button (mobile)
│   │   ├─ Title
│   │   └─ User info + Settings link
│   ├─ ConversationSidebar
│   │   ├─ New conversation button
│   │   └─ Conversation list
│   ├─ Dashboard
│   │   ├─ Score chips (Readiness, Sleep, Activity, HR)
│   │   ├─ Score arcs (animated circular progress)
│   │   ├─ Health insight
│   │   ├─ Sleep details + Heart rate chart
│   │   ├─ Activity + Vitals metrics
│   │   ├─ Timeline events
│   │   └─ 7-day trend charts
│   └─ HoldToTalk (fixed FAB)
│       ├─ Microphone button
│       └─ Expanded message card

```

### 4.2 State Management

```

// App-level state (App.tsx)
const [user, setUser] = useState<User | null>(null);
const [loading, setLoading] = useState(true);
const [selectedConversation, setSelectedConversation] = useState<Conversation | null>(null);
const [conversationMessages, setConversationMessages] = useState<Message[]>([]);
const [biologicalSex, setBiologicalSex] = useState<string | null>(null);
const [sidebarOpen, setSidebarOpen] = useState(false);

// Dashboard state

```

```
const [todayData, setTodayData] = useState<DashboardToday | null>(null);
const [weekData, setWeekData] = useState<DashboardWeek | null>(null);

// HoldToTalk state
const [status, setStatus] = useState<"idle" | "sending" | "speaking" | "error">
("idle");
const [expanded, setExpanded] = useState(false);
```

4.3 Responsive Design

Breakpoint	Behavior
Mobile (< 1024px)	Sidebar hidden, hamburger menu, stacked layouts
Desktop (>= 1024px)	Fixed sidebar, multi-column grids

```
// Sidebar: hidden on mobile, fixed on desktop
className={`
  fixed top-0 left-0 h-full w-72 bg-card border-r
  transform transition-transform duration-300
  ${isOpen ? "translate-x-0" : "-translate-x-full"}
  lg:translate-x-0
`}
```

5. External API Integrations

5.1 Oura Ring API v2

**Base URL:** <https://api.ouraring.com/v2/usercollection> **Authentication:** Bearer token (Personal Access Token)

Endpoint	Data Retrieved
/daily_sleep	Sleep score, contributors
/daily_readiness	Readiness score, contributors
/daily_activity	Activity score, steps, calories
/heartrate	Hourly heart rate samples
/daily_stress	Stress/recovery minutes
/daily_spo2	Blood oxygen percentage
/sleep	Detailed sleep periods (stages, HRV, HR)
/personal_info	Biological sex, age, weight

5.2 Anthropic Claude API

**Base URL:** <https://api.anthropic.com/v1/messages> **Model:** claude-sonnet-4-5

## Use Cases:

### 1. Voice Coach Responses

- Max tokens: 300
- System prompt includes Oura context and conversation history
- Designed for audio-friendly output (short sentences)

### 2. Title Generation

- Max tokens: 30
- Generates 2-5 word conversation titles

## System Prompt Template:

```
You are {username}'s voice coach. Speak in short, audio-friendly sentences.  
Use the provided Oura context to give actionable guidance.
```

```
Oura context:  
{JSON serialized health data}
```

```
If the user says "morning briefing", respond with:
```

- 1 headline
- sleep score and readiness score
- training recommendation
- 2 recovery actions
- 1 nutrition focus

```
Keep it under 20 seconds.
```

## 5.3 ElevenLabs TTS

**Base URL:** `https://api.elevenlabs.io/v1` **Model:** `eleven_turbo_v2_5`

### Available Voices:

- rachel (default) - Calm, warm female
- drew - Confident male
- antoni - Friendly male
- And 10 more options

### Voice Settings:

- Stability: 0.5
- Similarity boost: 0.75

## 5.4 Whisper.cpp (Local)

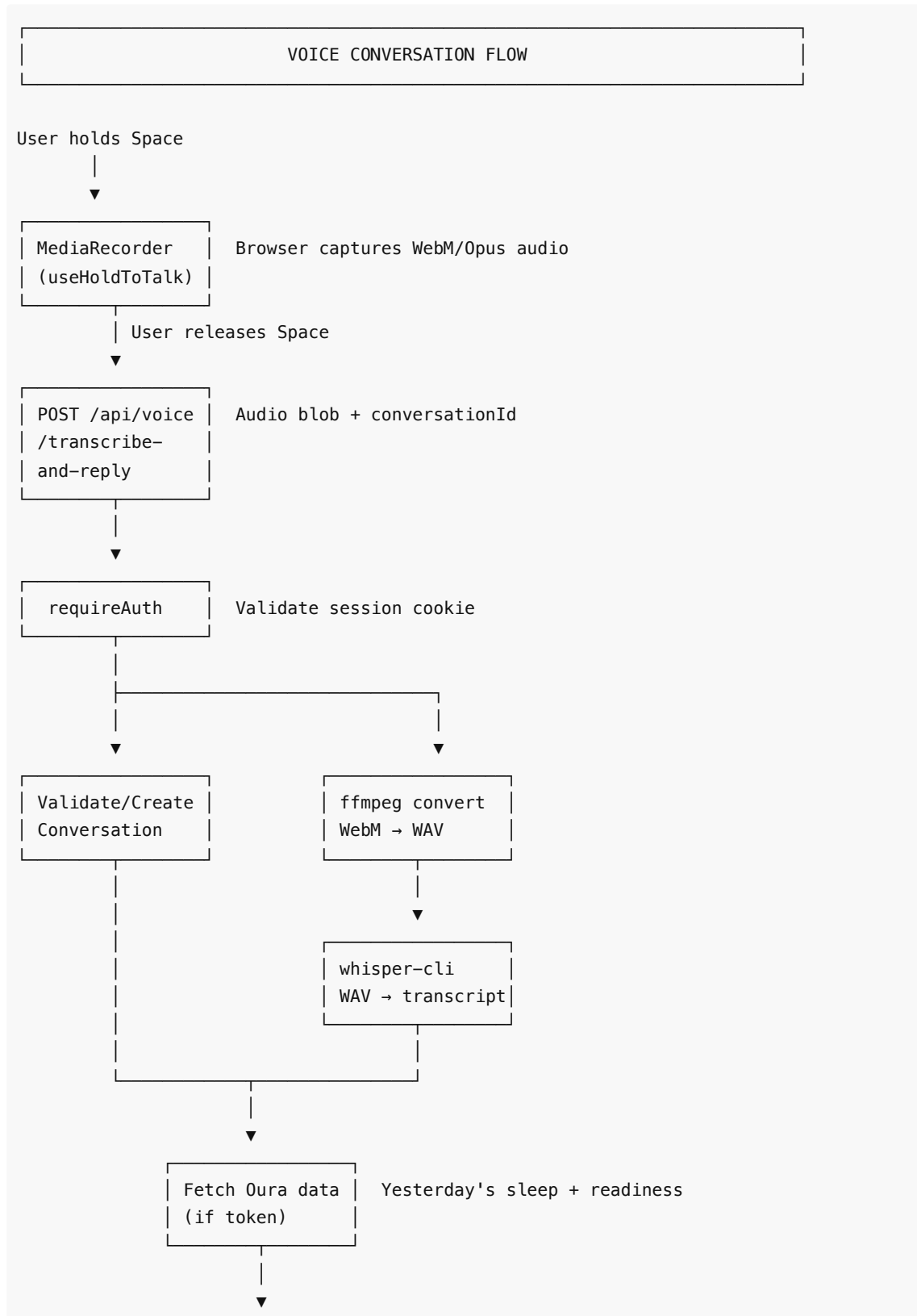
**Binary:** `whisper-cli` **Model:** `ggml-base.en.bin`

### Processing Pipeline:

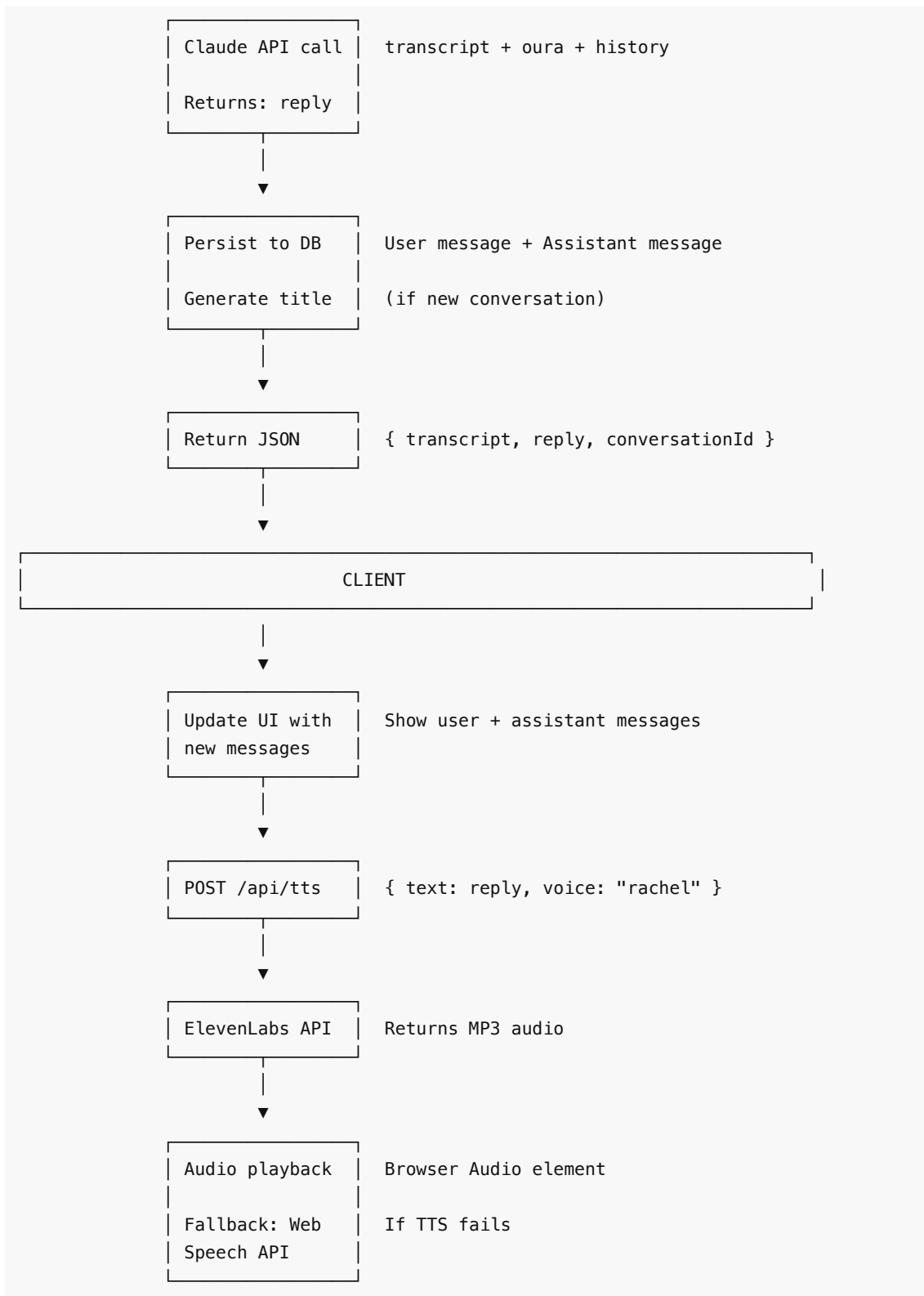
1. Receive WebM/Opus audio from browser
  2. Convert to 16kHz mono WAV via ffmpeg
  3. Run whisper-cli transcription
  4. Return text transcript
-

## 6. Data Flows

### 6.1 Voice Conversation Flow







## 6.2 Dashboard Data Flow

Dashboard mounts

|

▼  
Promise.all([  
 fetch /api/dashboard/today,  
 fetch /api/dashboard/week  
)])

|

▼

SERVER: /api/dashboard/today

1. Check cache: "{userId}:dashboard:today:{date}"
  - └ If hit: return cached data
2. If miss, fetch in parallel:
  - └ fetchDailySleep(today)
  - └ fetchDailyReadiness(today)
  - └ fetchDailyActivity(today)
  - └ fetchDailyStress(today)
  - └ fetchDailySpo2(today)
  - └ fetchHeartRate(today, today)
  - └ fetchSleepPeriods(yesterday, today)
3. Parse and shape response
4. Cache with 5-minute TTL
5. Return aggregated data

|

▼

CLIENT: Dashboard

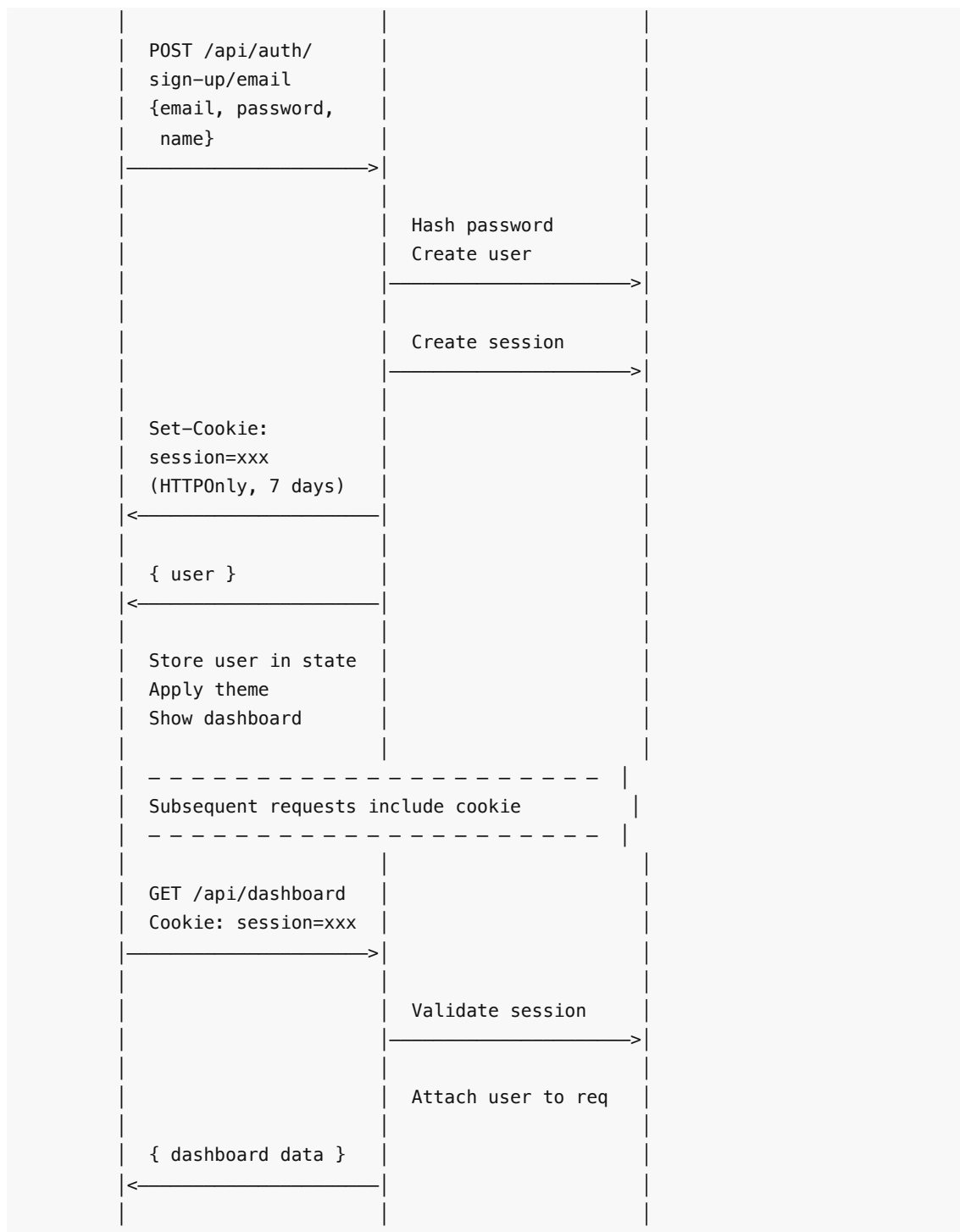
- Render:
- └ Score chips (Readiness, Sleep, Activity, HR)
  - └ Score arcs with status labels (OPTIMAL, GOOD, etc.)
  - └ Health insight message
  - └ Sleep details grid + stages bar
  - └ Heart rate line chart (Recharts)
  - └ Activity metrics (calories, SpO2, stress)
  - └ Timeline events
    - └ 7-day trend charts (sleep, readiness, activity, steps)

### 6.3 Authentication Flow

Client

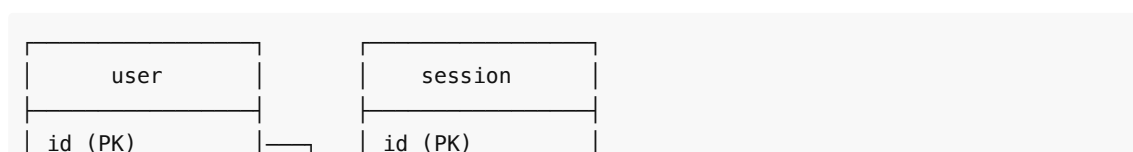
Server

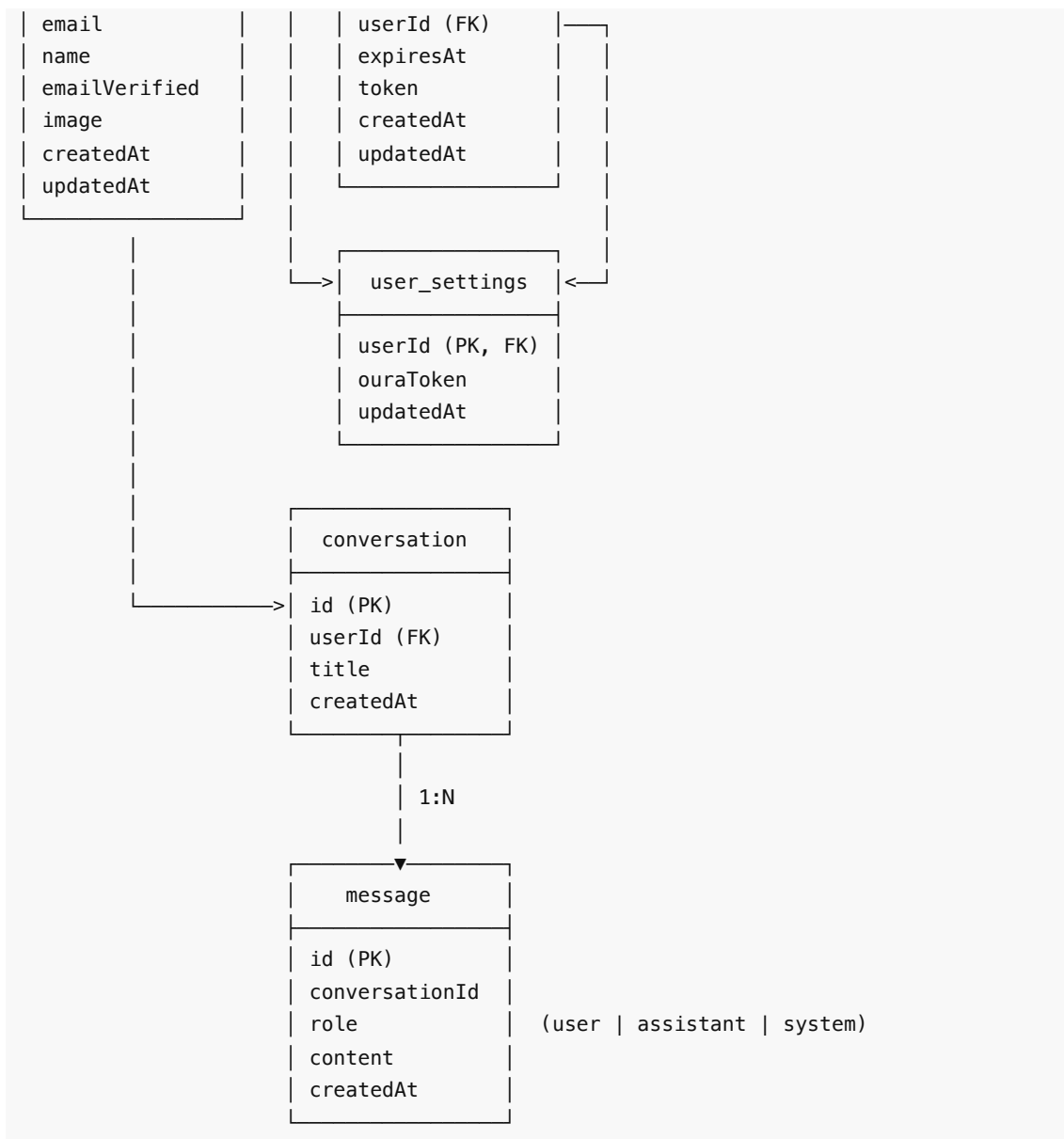
Database



## 7. Database Schema

### 7.1 Entity Relationship Diagram





## 7.2 Table Definitions

```

-- User table (managed by better-auth)
CREATE TABLE user (
  id TEXT PRIMARY KEY,
  email TEXT UNIQUE NOT NULL,
  name TEXT,
  emailVerified INTEGER DEFAULT 0,
  image TEXT,
  createdAt TEXT NOT NULL,
  updatedAt TEXT NOT NULL
);

-- Session table (managed by better-auth)
CREATE TABLE session (

```

```
id TEXT PRIMARY KEY,
userId TEXT NOT NULL REFERENCES user(id) ON DELETE CASCADE,
token TEXT UNIQUE NOT NULL,
expiresAt TEXT NOT NULL,
createdAt TEXT NOT NULL,
updatedAt TEXT NOT NULL
);

-- User settings (custom)
CREATE TABLE user_settings (
  userId TEXT PRIMARY KEY REFERENCES user(id) ON DELETE CASCADE,
  ouraToken TEXT,
  updatedAt TEXT NOT NULL
);

-- Conversations
CREATE TABLE conversation (
  id TEXT PRIMARY KEY,
  userId TEXT NOT NULL REFERENCES user(id) ON DELETE CASCADE,
  title TEXT,
  createdAt TEXT NOT NULL
);
CREATE INDEX conversation_userId_idx ON conversation(userId);

-- Messages
CREATE TABLE message (
  id TEXT PRIMARY KEY,
  conversationId TEXT NOT NULL REFERENCES conversation(id) ON DELETE CASCADE,
  role TEXT NOT NULL CHECK(role IN ('user', 'assistant', 'system')),
  content TEXT NOT NULL,
  createdAt TEXT NOT NULL
);
CREATE INDEX message_conversationId_idx ON message(conversationId);
```

## 8. Security Considerations

### 8.1 Authentication Security

Measure	Implementation
Session Storage	HTTPOnly cookies (not accessible via JavaScript)
Session Duration	7-day TTL with automatic renewal
Password Hashing	better-auth (bcrypt under the hood)
CORS	Restricted to specific origins
Credentials	Required for all authenticated requests

### 8.2 Data Protection

Measure	Implementation
SQL Injection	Parameterized queries via better-sqlite3
Authorization	User ownership checks on all resources
API Keys	Environment variables only, never exposed
Conversation Privacy	Filtered by userId on all queries

### 8.3 Input Validation

```
// Example: Conversation ownership check
const conversation = storage.getConversation(conversationId);
if (!conversation || conversation.userId !== req.user.id) {
  return res.status(404).json({ error: "Conversation not found" });
}
```

## 9. API Reference

### 9.1 Authentication

Method	Endpoint	Description	Auth
POST	/api/auth/sign-up/email	Create account	No
POST	/api/auth/sign-in/email	Sign in	No
POST	/api/auth/sign-out	Sign out	Yes
GET	/api/auth/get-session	Get current session	Yes

### 9.2 Conversations

Method	Endpoint	Description	Auth
GET	/api/conversations	List all	Yes
POST	/api/conversations	Create new	Yes
GET	/api/conversations/:id	Get with messages	Yes
PATCH	/api/conversations/:id	Update title	Yes
DELETE	/api/conversations/:id	Delete	Yes

### 9.3 Voice

Method	Endpoint	Description	Auth
POST	/api/voice/transcribe-and-reply	Process audio, get AI reply	Yes

**Request:** multipart/form-data

- `audio` : WebM audio file
- `conversationId` : (optional) existing conversation UUID

**Response:**

```
{
  "transcript": "User's spoken text",
  "reply": "AI coach response",
  "conversationId": "uuid",
  "conversationTitle": "Generated title",
  "isNewConversation": true
}
```

## 9.4 Dashboard

Method	Endpoint	Description	Auth
GET	<code>/api/dashboard/today</code>	Today's health data	Yes
GET	<code>/api/dashboard/week</code>	7-day trends	Yes

## 9.5 TTS

Method	Endpoint	Description	Auth
POST	<code>/api/tts</code>	Synthesize speech	No
GET	<code>/api/tts/voices</code>	List available voices	No

**Request:**

```
{
  "text": "Text to speak",
  "voice": "rachel"
}
```

**Response:** `audio/mpeg` (MP3 binary)

## 9.6 Settings

Method	Endpoint	Description	Auth
POST	<code>/api/settings/oura-token</code>	Save Oura token	Yes
GET	<code>/api/settings/oura-token</code>	Check if token exists	Yes
GET	<code>/api/settings/profile</code>	Get biological sex	Yes

# 10. Deployment

## 10.1 Environment Variables

```
# Required
ANTHROPIC_API_KEY=sk-ant-...
ELEVENLABS_API_KEY=...

# Optional
PORT=3000
DATABASE_PATH=./sqlite.db
AUTH_BASE_URL=http://localhost:3000
CLAUDE_MODEL=claude-sonnet-4-5

# Client (.env)
VITE_API_BASE=http://localhost:3000
```

## 10.2 Local Development

```
# Terminal 1: Server
cd server
npm install
npm run dev # Runs on port 3000

# Terminal 2: Client
cd client
npm install
npm run dev # Runs on port 5174

# Access: http://localhost:5174
```

## 10.3 External Dependencies

Dependency	Required	Purpose
ffmpeg	Yes	Audio format conversion
whisper.cpp	Yes	Local transcription
Oura Personal Access Token	Per-user	Health data access

## 10.4 Whisper.cpp Setup

```
# Clone and build whisper.cpp
git clone https://github.com/ggerganov/whisper.cpp.git
cd whisper.cpp
make

# Download model
./models/download-ggml-model.sh base.en

# Binary location: whisper.cpp/build/bin/whisper-cli
# Model location: whisper.cpp/models/ggml-base.en.bin
```



---

## Appendix A: Score Thresholds

Score Range	Status Label	Color
>= 85	OPTIMAL	Green
>= 70	GOOD	Green
>= 50	FAIR	Yellow
< 50	NEEDS ATTENTION	Red

---

## Appendix B: Voice Options

Name	ID	Description
rachel	21m00Tcm4TlvDq8ikWAM	Calm, warm female
drew	29vD33N1CtxCmqQRPOHJ	Confident male
clyde	2EiwWnXFvU5JabPnv8n	Warm male
paul	5Q0t7uMcjvnagumLfvZi	News anchor
domi	AZnzlk1XvdvUeBnXmIld	Energetic female
dave	CYw3kZ02Hs0563khs1Fj	British male
fin	D38z5RcWu1voky8WS1ja	Irish male
sarah	EXAVITQu4vr4xnSDxMaL	Soft female
antoni	ErXwobaYiN019PkySvjV	Friendly male
josh	TxGEqnHWrfWFTfGW9XjX	Deep male
arnold	VR6AewLTigWG4xSOukaG	Crisp male
adam	pNlnz6obpgDQGcFmaJgB	Narration male
sam	yoZ06aMxZJJ28mfd3POQ	Dynamic male

---