Caresma Platform Architecture

This document describes the complete system architecture for the Caresma cognitive health assessment platform.

System Overview

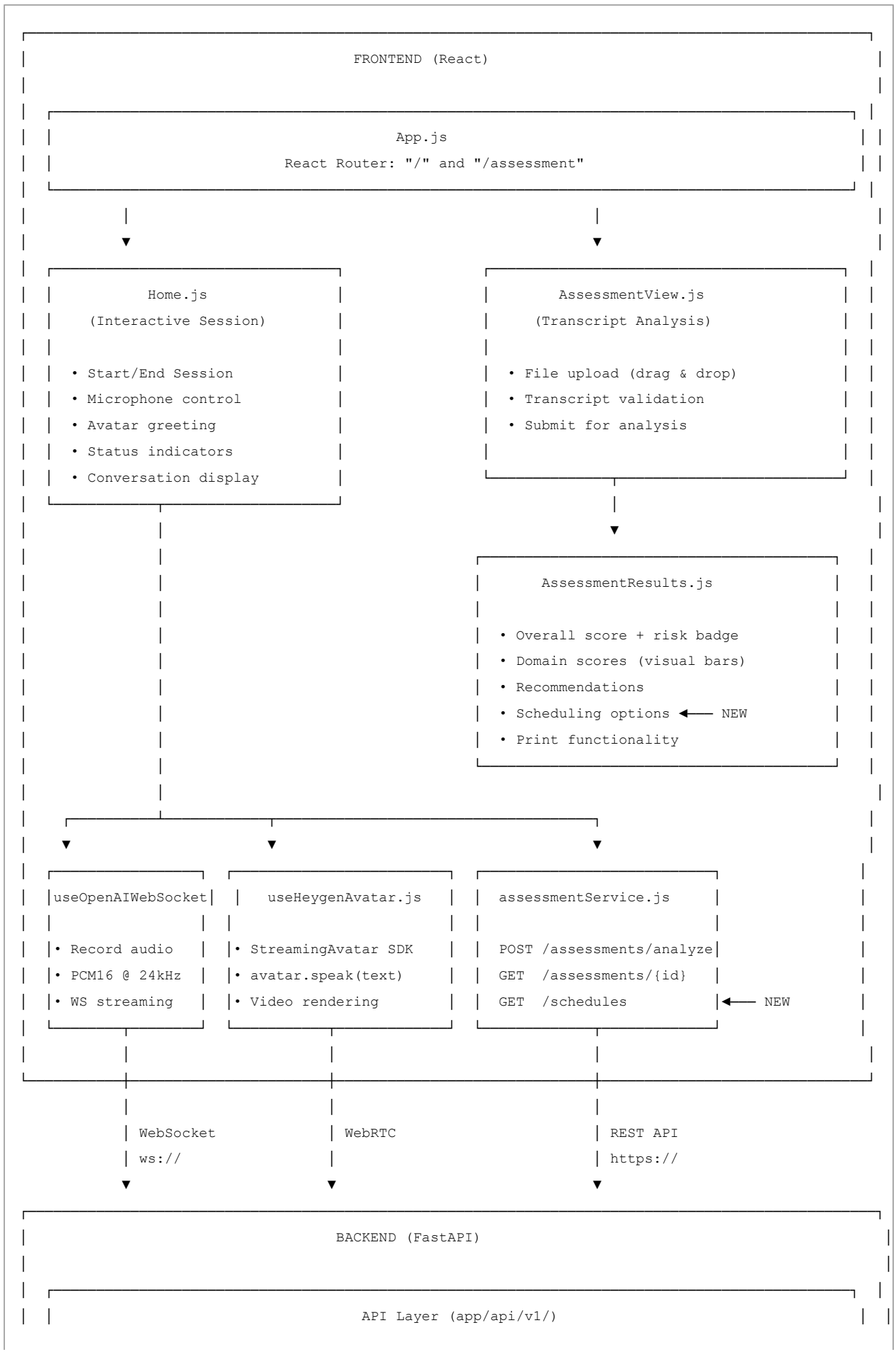
Caresma is a cognitive health assessment platform that enables:

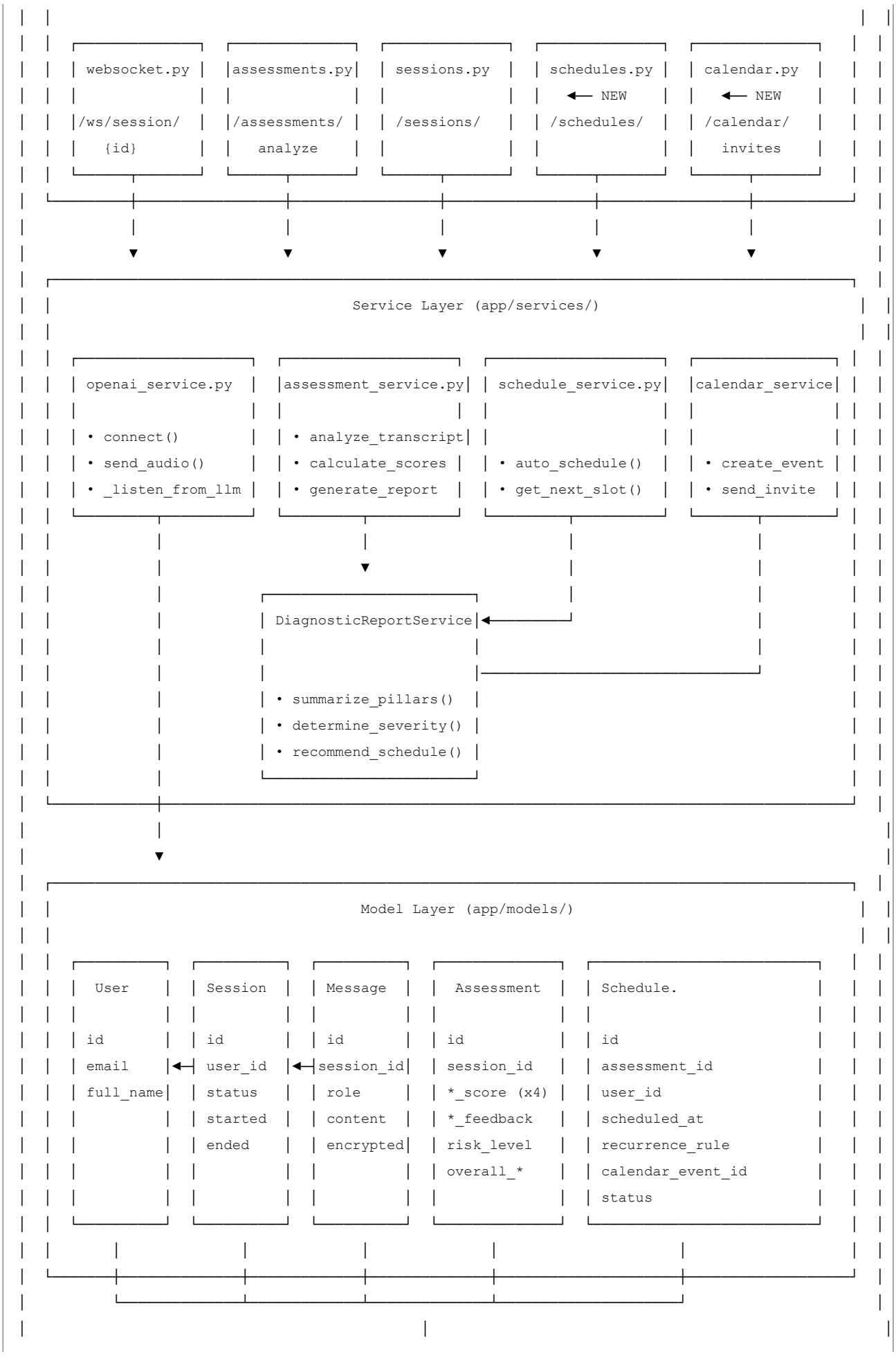
- **Real-time voice conversations** with an AI-powered avatar
- **Cognitive assessment** across key domains (Memory, Language, Executive Function, Orientation)
- **Diagnostic-style reporting** with risk level indicators
- **Automated session scheduling** based on assessment severity
- **Calendar integration** for appointment management

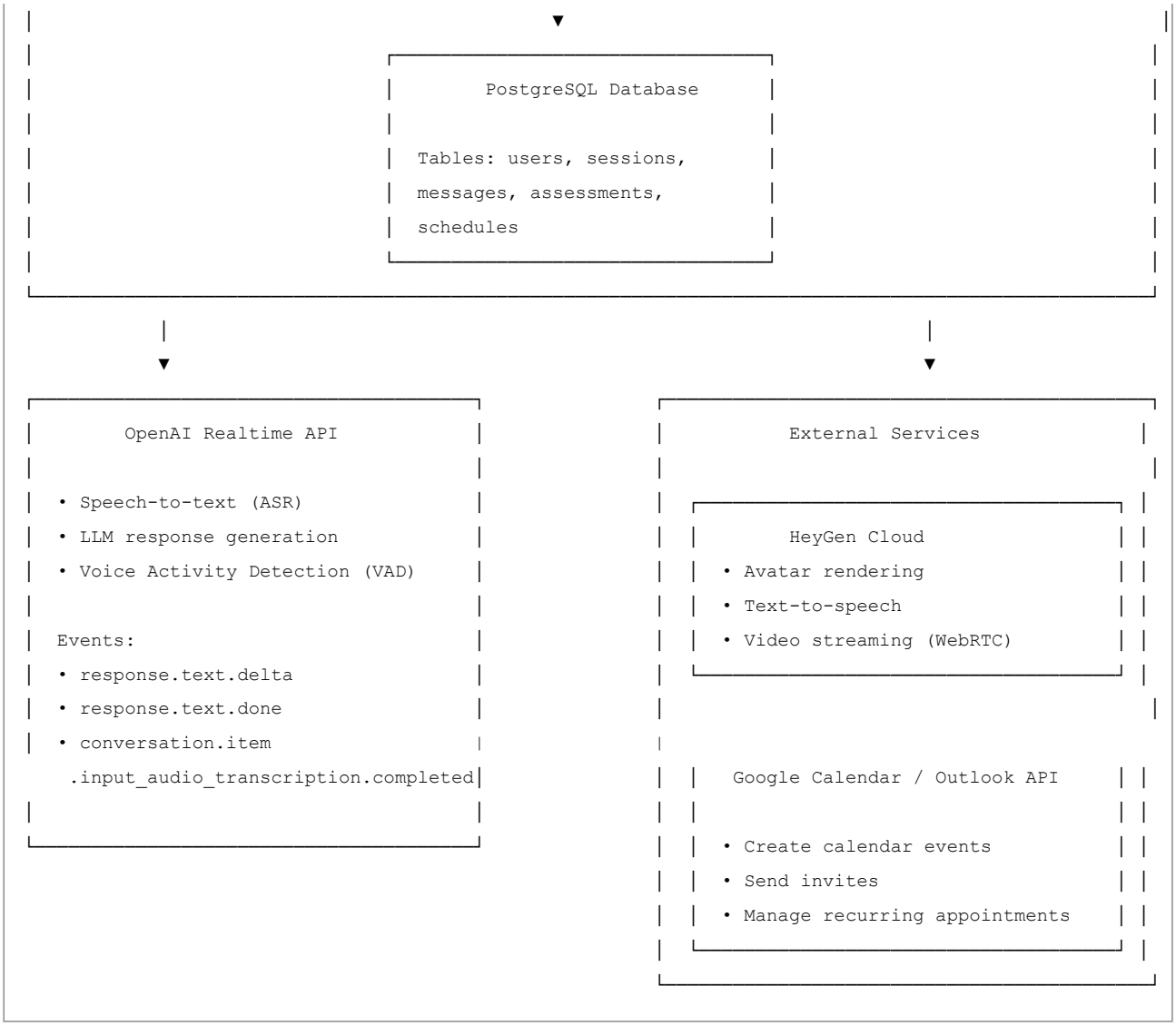
Technology Stack

Layer	Technology
Frontend	React 19, HeyGen Streaming Avatar SDK, WebSocket
Backend	FastAPI (Python), AsyncIO, WebSockets
Database	PostgreSQL with SQLAlchemy ORM
AI Services	OpenAI Realtime API, OpenAI GPT-4
Avatar	HeyGen Streaming Avatar (WebRTC)
Calendar	Google Calendar API / Outlook API

Complete Architecture Diagram


FRONTEND (React)
App.js
React Router: "/" and "/assessment"
Home.js
(Interactive Session)
• Start/End Session
• Microphone control
• Avatar greeting
• Status indicators
• Conversation display
AssessmentView.js
(Transcript Analysis)
• File upload (drag & drop)
• Transcript validation
• Submit for analysis
AssessmentResults.js
• Overall score + risk badge
• Domain scores (visual bars)
• Recommendations
• Scheduling options ◄── NEW
• Print functionality
useOpenAIWebSocket
• Record audio
• PCM16 @ 24kHz
• WS streaming
useHeygenAvatar.js
• StreamingAvatar SDK
• avatar.speak(text)
• Video rendering
assessmentService.js
POST /assessments/analyze
GET /assessments/{id}
GET /schedules
◄── NEW
WebSocket
ws://
WebRTC
REST API
https://
BACKEND (FastAPI)
API Layer (app/api/v1/)


websocket.py
/ws/session/{id}
assessments.py
/assessments/analyze
sessions.py
/sessions/
schedules.py
NEW
/schedules/
calendar.py
NEW
/calendar/invites
Service Layer (app/services/)
openai_service.py
• connect()
• send_audio()
• _listen_from_llm
assessment_service.py
• analyze_transcript
• calculate_scores
• generate_report
schedule_service.py
• auto_schedule()
• get_next_slot()
calendar_service
• create_event
• send_invite
DiagnosticReportService
• summarize_pillars()
• determine_severity()
• recommend_schedule()
Model Layer (app/models/)
User
id
email
full_name
Session
id
user_id
status
started
ended
Message
id
session_id
role
content
encrypted
Assessment
id
session_id
*_score (x4)
*_feedback
risk_level
overall_*
Schedule.
id
assessment_id
user_id
scheduled_at
recurrence_rule
calendar_event_id
status


PostgreSQL Database
Tables: users, sessions, messages, assessments, schedules
OpenAI Realtime API
• Speech-to-text (ASR)
• LLM response generation
• Voice Activity Detection (VAD)
Events:
• response.text.delta
• response.text.done
• conversation.item .input_audio_transcription.completed
External Services
HeyGen Cloud
• Avatar rendering
• Text-to-speech
• Video streaming (WebRTC)
Google Calendar / Outlook API
• Create calendar events
• Send invites
• Manage recurring appointments