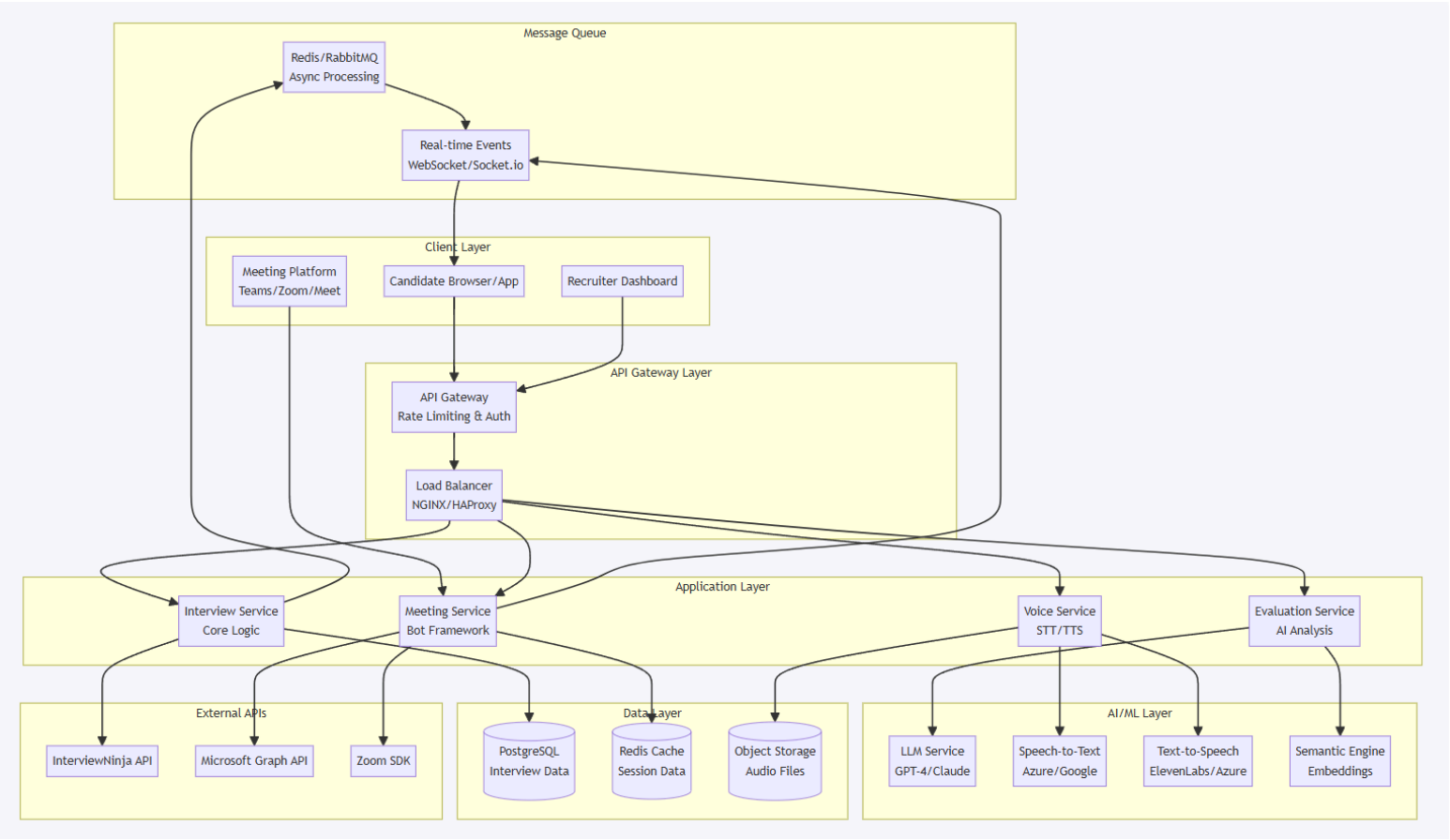
Team Null Pointers

AI-Powered Interview Platform

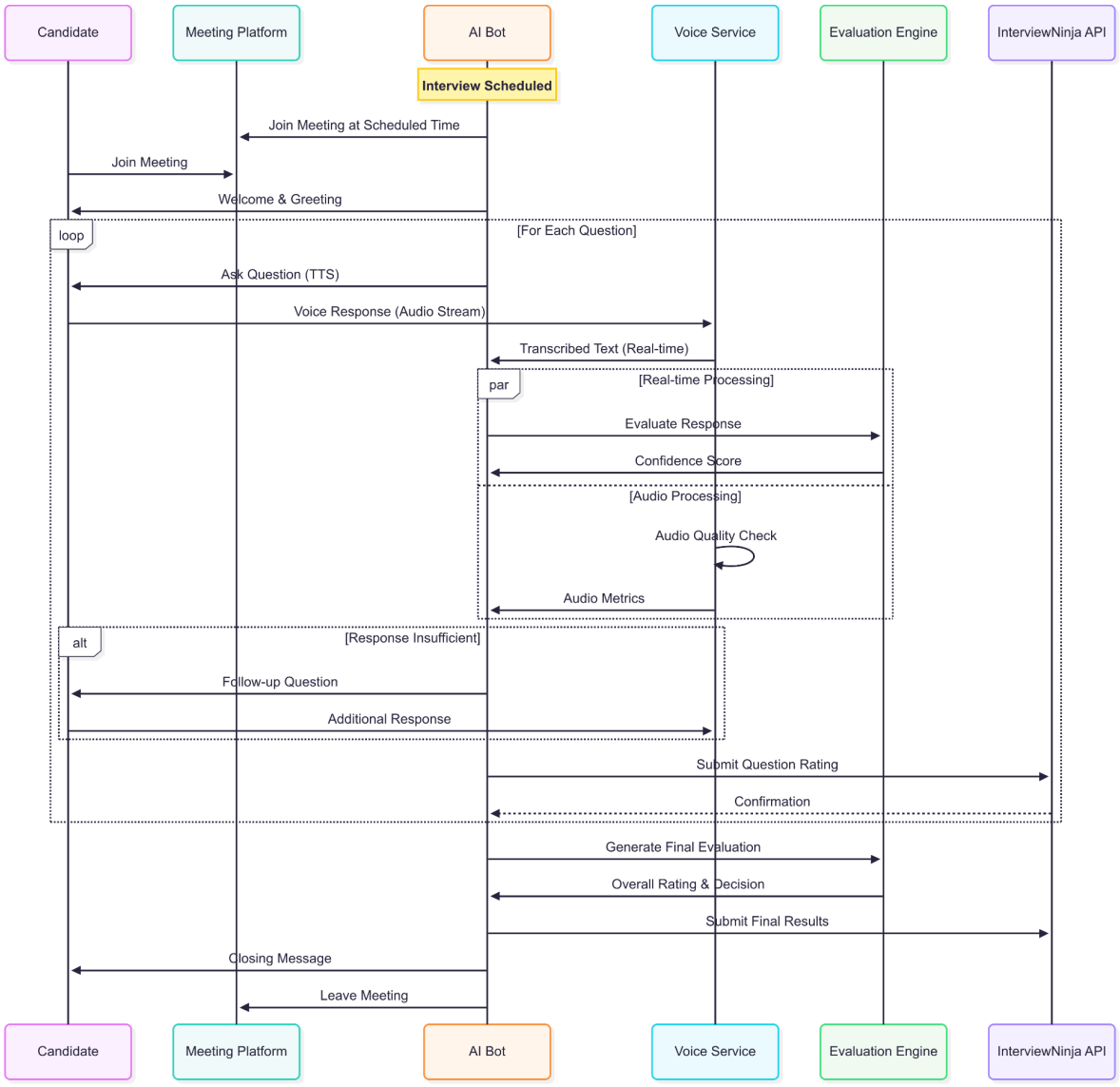
📊 High-Level System Architecture



Team Null Pointers

AI-Powered Interview Platform

📊 Sequence Diagram



Team Null Pointers

AI-Powered Interview Platform

🏗️ Technology Stack & Tools

🎤 Voice Processing

- **STT:** Google Cloud Speech/ Deepgram (Not decided)
- **TTS:** ElevenLabs / Google Cloud TTS for extra voices/accents.
- **Audio:** WebRTC, Opus Codec
- **Streaming:** WebSocket, Socket.io

🤖 AI/ML Services

- **LLM:** OpenAI GPT-5 / Anthropic Claude
- **Embeddings:** OpenAI text-embedding-ada-002/text-embedding-3-small
- **Vector DB:** Pinecone / Weaviate
- **ML Ops:** MLflow, Weights & Biases

🔧 Backend Infrastructure

- **Runtime:** Java / Python (FastAPI)
- **Database:** PostgreSQL + Redis
- **Queue:** Redis Bull / RabbitMQ
- **Storage:** PostgreSQL JSONB

📞 Meeting Integration

- **Teams:** Bot Framework SDK(Azure Communication Services)/Twilio Voice / Vonage API
- **Zoom:** Zoom Video SDK
- **Universal:** Puppeteer / Playwright
- **WebRTC:** Simple-peer, PeerJS

🎯 Key Performance Targets

- **Latency:** < 200ms for voice processing pipeline
- **Throughput:** 1000+ concurrent interviews
- **Scalability:** Auto-scale from 1-100 instances
- **Reliability:** 99.9% uptime with circuit breakers
- **Recovery:** < 30s failover time

Team Null Pointers

AI-Powered Interview Platform

1 Overview

An AI-powered interview bot joins scheduled online meetings (Zoom/Teams/WebRTC) to run structured interviews. It captures audio, transcribes speech in real time, evaluates responses using ML/LLM models, and streams live results to recruiters. Heavy analysis tasks run asynchronously to maintain low latency.

2 Key Components

Client Layer

Candidate app + recruiter dashboard (WebSocket for live updates)

Meeting Platform

Zoom/Teams/WebRTC where the bot joins as a participant

Meeting Service

Orchestrates interview flow, TTS playback, state machine logic

Voice Service

STT/TTS processing, VAD, diarization, audio quality checks

Interview Service

Stores question sets, calls evaluation, aggregates results

Evaluation Service

Real-time + batch scoring using rules, embeddings, and LLMs

Message Queue

Decouples real-time from heavy async processing

Data Layer

PostgreSQL, Redis, Object Storage, Vector DB

External Providers

STT/TTS, LLM, embeddings APIs

Team Null Pointers

AI-Powered Interview Platform

3 Runtime Flow

Pre-interview

1. Recruiter schedules via API/UI → interview metadata stored in PostgreSQL
2. Meeting Service prepares bot session, caches state in Redis, pre-warms AI/STT services

Interview Start

1. Candidate + bot join meeting
2. Bot greets candidate via TTS; recruiter sees `interview_started` event in dashboard

Question Loop

1. Bot asks question (TTS)
2. Candidate responds → Voice Service streams audio to STT → partial transcripts sent to Evaluation Service + dashboard
3. Real-time evaluation decides on follow-up or marks question complete
4. Audio/transcripts saved; embeddings generated asynchronously

Post-interview

1. Evaluation aggregates scores, generates summary & recommendations
2. Final results stored and sent to recruiter dashboard
3. Async workers perform deeper analysis, cleanup, and highlight generation

Team Null Pointers

AI-Powered Interview Platform

4 Data & State

Redis

Ephemeral session data, presence, rate-limit counters

PostgreSQL

Interview definitions, results, scores

Object Storage

Audio, TTS clips, recordings

Vector DB

Response embeddings for semantic similarity

Message Queue

Async jobs (embedding generation, transcript cleanup)

5 Real-Time vs Async

Real-time (<500ms)

STT streaming, partial transcript scoring, live UI updates

Async (seconds–minutes)

Full LLM analysis, embedding generation, report compilation

6 Reliability & Fallbacks

Audio/STT Issues

Re-ask or offline transcription

Meeting Disconnect

Auto-rejoin or record-only mode

LLM Timeout

Fallback to rules-based scoring

Job Processing

Idempotent async jobs to prevent duplicates