the agile monkeys.

Voice Translation Service High Level Architecture

Version: 1

Date: Dec 22, 2025

Overview

The Voice Translation Service enables real-time, two-way voice translation for live phone conversations by joining the call as a neutral third participant. Callers continue to place and receive calls exactly as they do today; the service integrates seamlessly with existing signaling and media infrastructure without requiring any changes to client behavior or devices.

The system is designed to deliver low-latency, intelligible translated speech while supporting multiple participants, overlapping speech, and variable call quality typical of telephony environments. Each participant can receive translated audio in their preferred language and voice, enabling natural, bi-directional conversations across language barriers. The architecture emphasizes isolation, resilience, and extensibility to ensure the service can scale reliably as usage and supported providers grow.

Service Architecture

At a high level, the service is structured around a clear separation of concerns between **external connectivity**, **session orchestration**, and **translation providers**. A single WebSocket connection anchors each call and acts as the ingress and egress point for all audio and control messages. From this connection, the system establishes an internal session that is responsible for routing audio, managing participants, and coordinating translation workflows.

Within a session, each participant's audio is processed through a dedicated translation pipeline when required. This design allows the system to maintain independent language targets, voices, and latency profiles per participant, while also avoiding limitations imposed by some translation backends that support only a single output language per session. Pipelines are created dynamically as participants begin speaking and are torn down independently when participants disconnect, ensuring efficient use of resources and minimizing the impact of failures.

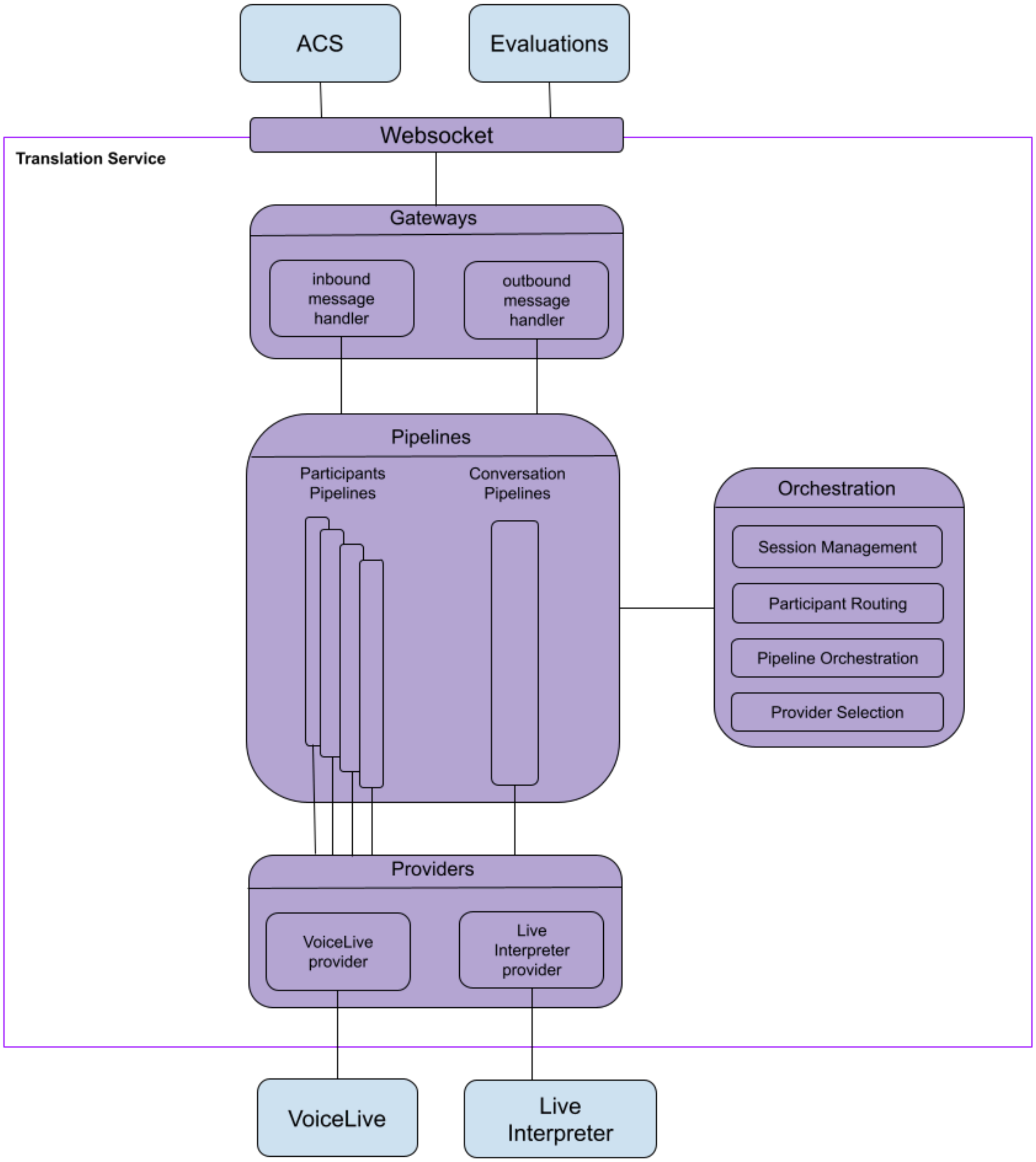
External protocols and message formats are terminated at the system boundary by gateway components. These gateways are responsible for accepting inbound audio and metadata, converting them into internal envelopes, and emitting translated results back into the call. They intentionally remain thin, focusing on protocol translation and event flow rather than business logic, which keeps the system adaptable to future ingress mechanisms beyond the current telephony integration.

Translation and speech synthesis are handled by provider components that encapsulate the specifics of individual backend services. Each participant pipeline maintains its own provider session, allowing concurrent use of different providers or configurations within the same call. This isolation is critical both for correctness—ensuring each caller receives audio in the correct language—and for resilience, as issues with one provider session do not affect other participants or the overall call.

Overall, the architecture balances real-time performance with operational robustness. By isolating participants, constraining backpressure to individual pipelines, and maintaining a consistent internal event model, the service can scale horizontally, recover gracefully from partial failures, and evolve to support new providers and capabilities without disruptive changes to the core system.

Architecture Diagram

High level Architecture Overview


ACS
Evaluations
Websocket
Translation Service
Gateways
inbound message handler
outbound message handler
Pipelines
Participants Pipelines
Conversation Pipelines
Orchestration
Session Management
Participant Routing
Pipeline Orchestration
Provider Selection
Providers
VoiceLive provider
Live Interpreter provider
VoiceLive
Live Interpreter