How I Built a Production Voice AI Feature on Android Using Google Gemini Live API

By **Bhavik Sadiwala** · Staff Engineer Android | AI Agents & Automation · Procore Technologies

Most voice AI tutorials show you a demo that works in a quiet room with a good microphone. At Procore, we needed it to work on a construction site — heavy machinery, wind, background chatter, workers wearing gloves who cannot type. That changes everything. Here is exactly how we built it, what we got wrong the first time, and what the architecture looks like in production.

Why We Chose Gemini Live API

The obvious choice would have been a managed voice AI platform that handles the full pipeline out of the box. The problem was latency and control. Construction teams query live project data — "Show me all overdue tasks on Block C" needs a real answer in under two seconds or the user gives up. Gemini Live API gives us one bidirectional stream handling speech-to-text, LLM reasoning, barge-in detection, and text-to-speech simultaneously. No stitching together three separate services. No handoff latency. One stream, one connection, sub-second response.

The Architecture

The mobile apps are intentionally thin clients. Each Android app opens a single authenticated WebSocket to a Procore-owned gateway, which proxies audio into a Google Gemini Live API session. The gateway owns the Gemini session. The Android client speaks a simple JSON wire protocol:

```
{ "type": "start", "session_id": "abc123" }
{ "type": "audio", "data": "<base64_pcm_chunk>" }
{ "type": "interrupt" }
{ "type": "stop" }
```

This design means swapping Gemini for another model is a backend-only change. Zero mobile deployment required.

Audio Pipeline on Android

The Android app captures microphone audio at 16 kHz PCM mono and streams it as base64-encoded frames. Use VOICE_COMMUNICATION as the audio source — it enables hardware noise suppression automatically, which matters enormously in field environments. Response audio comes back at 24 kHz and plays through AudioTrack.

```
audioRecord = AudioRecord(
MediaRecorder.AudioSource.VOICE_COMMUNICATION,
SAMPLE_RATE_HZ, // 16000
AudioFormat.CHANNEL_IN_MONO,
AudioFormat.ENCODING_PCM_16BIT,
bufferSize
)
```

The Hard Problems (Not the Integration)

The WebSocket connection was running in three days. The next three months were spent on real engineering challenges:

1. Barge-in Handling

Users interrupt the agent mid-sentence. Gemini handles barge-in detection server-side, but you still need to stop audio playback immediately on the client. AudioTrack.pause() followed by AudioTrack.flush() — not just pause().

2. Jitter Buffering

Network conditions on construction sites are inconsistent. Audio frames arrive out of order. We buffer 80ms of audio before playback starts. Users do not notice this delay, but choppy audio makes the whole feature feel broken.

3. Field Noise

We tested in the office. It worked. We tested on a real site. It did not. VOICE_COMMUNICATION audio source helped, but the real fix was adjusting VAD (voice activity detection) sensitivity in the Gemini session config so it does not cut off mid-sentence when a drill starts.

> **The integration code is the easy part. The environment is the hard part. Test in real conditions before you call the feature done.**

Results

The feature shipped to production supporting 10+ languages. Support tickets related to the relevant workflows dropped measurably in the first quarter. Workers who previously avoided the app because typing on a tablet mid-task was impractical became active daily users.

Key Takeaways

- Thin client architecture decouples the model from the app — swap models with zero mobile deployment
- Use VOICE_COMMUNICATION audio source for hardware noise suppression
- Buffer 80ms before playback to handle network jitter
- Test in the real environment early — office testing is never representative
- Barge-in on the client requires pause() + flush(), not just pause()

Bhavik Sadiwala is a Staff Engineer Android at Procore Technologies, specialising in AI-powered mobile features and automation. bhaviksadiwala.com | LinkedIn