Real-Time Speech Translation - Earbuds & WebRTC - Cheat Sheet

1 - The four-step pipeline

```
Hear -> ASR (speech to text) -> MT (translate) -> TTS (text to speech) -> deliver.
Same pipeline whether the audio is from earbuds or a video call.
```

2 - 'Translation earbuds' = an audio endpoint in front of the pipeline

```
On-device (Apple Live Translation, iOS 26): private, offline, hardware-capped.
Phone + cloud (Pixel / Galaxy): Gemini-class, 70+ languages, needs a network.
```

3 - Cascade vs direct (how to build the brain)

```
Cascade (ASR->MT->TTS): readable text -> captions, glossary, debug, audit. More delay.
Direct S2ST (Seamless / GPT Realtime / Gemini Live): lower delay, keeps the voice.
Regulated + caption-heavy -> cascade. Latency-critical + natural voice -> direct. Often both.
```

4 - Why it rewrites itself + the latency budget

```
Languages reorder words -> early translation is a guess. REPLACE the line until final.
Stay ~2-3 s behind (ear-voice span / wait-k). Speech path waits longer than captions.
Budget: ~1 s machine (TTS biggest) + ~2-3 s wait. <800 ms feels live; AIIC ceiling 3-5 s.
```

5 - In a WebRTC call: translate once at the SFU, fan out per language

```
Tap speaker audio at the SFU, VAD-gate it, translate once per listener language.
Captions -> data channel (RFC 8831). Audio -> injected Opus track (RFC 6716), duck original.
Cost ~ languages x minutes: 3 langs x 60 min x $0.12 = $21.60/hr. VAD gates the rest.
```

6 - Build-vs-buy checklist

- [ ] Group call you record / audit? -> cascade at the SFU (text for captions + record).
- [ ] Two-person, natural-voice feel? -> direct S2ST model, expressive variant.
- [ ] Self-host the call? -> mediasoup / Janus / LiveKit (own the tap + track injection).
- [ ] Pick ASR + MT + TTS (cascade) or one S2ST model; swap on first-chunk latency + accuracy.
- [ ] <60 concurrent streams -> cloud API wins; far above -> self-host (compliance decides).
- [ ] Always: VAD-gate, add a glossary for proper nouns, plan voice-clone consent + a record.