WebRTC Echo Cancellation - Integration Cheat Sheet

Fora Soft Learn · AI for Video Engineering

1 - Where echo cancellation sits (and the one rule)

Audio Processing Module order	1) Echo cancellation (AEC3) 2) Noise suppression 3) AGC
The reference / render signal	the far-end audio BEFORE any mixing - feed it correctly
#1 cause of WebRTC echo	a broken or fed-back reference wire, not a weak model

2 - The browser switches you control (getUserMedia)

```
echoCancellation: true   -> ON by default (Chrome, Firefox, Safari) = AEC3.
echoCancellationType     -> 'browser' (AEC3 software) | 'system' (OS/hardware).
Default picks hardware AEC when good, else falls back to AEC3 software.
```

3 - Mobile: the phone usually does the work

iOS / iPadOS	Voice-Processing I/O unit via AVAudioSession .voiceChat mode
Android	AcousticEchoCanceler - quality varies wildly by device model
Consistency trick	run AEC3 in software on Android for the same behaviour everywhere

4 - The number to log when debugging: ERLE

```
ERLE = 10 x log10(echo power before / after). Higher = more echo removed.
Healthy: climbs to 20-40 dB in 1-2 s.  Below 10 dB = filter not converging.
```

5 - The stacking trap (the most common self-inflicted bug)

```
Two cancellers at once -> half-duplex 'walkie-talkie', your voice cut off.
Fix: run EXACTLY ONE. Adopt a vendor SDK? Set browser echoCancellation=false.
```

6 - Decision checklist

- [ ] Ship browser AEC3 first (echoCancellation: true); verify with an ERLE log, not one test call.
- [ ] Stubborn echo? Check the reference wire and delay estimate BEFORE blaming the model.
- [ ] Non-linear echo only (loud/cheap speaker, double-talk, reverb)? -> add a neural canceller.
- [ ] Devices you control -> on-device SDK (Krisp) via AudioWorklet; phone-ins -> server-side (Maxine / SFU).
- [ ] Always one canceller in the chain - turn the others off.