WebRTC Noise Suppression - Integration Cheat Sheet

Fora Soft Learn · AI for Video Engineering

1 - Three homes for a suppressor (pick exactly one)

A. Built-in (at capture)	getUserMedia noiseSuppression:true - free, weakest
B. Custom, before encoder	RNNoise WASM / Krisp in an AudioWorklet - the right spot
C. Server-side	SFU / AI agent / SIP gateway / NVIDIA Maxine (GPU)

2 - Try the free baseline first

```
navigator.mediaDevices.getUserMedia({ audio: {
  noiseSuppression: true, echoCancellation: true } });
Free, on-device, zero integration. Ship it if complaints stop.
```

3 - The golden rule: raw audio, before the encoder

```
NS reads the ACTUAL sound -> it must run on RAW audio.
YES: Web Audio AudioWorklet, or MediaStreamTrackProcessor (AudioData).
NO : WebRTC Encoded Transform - it only sees Opus-compressed bytes.
```

4 - Never run two suppressors (the double-suppression trap)

```
Built-in + custom = 2x CPU and distorted, robotic voice.
Fix: set noiseSuppression:false when you ship a custom model.
Echo cancellation + gain control may stay on - one NOISE suppressor only.
```

5 - AudioWorklet buffering math (48 kHz)

```
Worklet delivers 128 samples = 2.67 ms; RNNoise wants 480 = 10 ms.
Buffer the small blocks (circular buffer) until 480, denoise, pass on.
Run off the main thread; keep total latency inside the 150 ms G.114 budget.
```

6 - Build vs buy (WebRTC)

RNNoise WASM	free, browser, you wire it; good on steady noise
DeepFilterNet	higher quality, heavier; often better server-side
Krisp AI / ai-coustics	paid, managed, all platforms, BVC, SIP, voice agents

7 - Integration checklist

- [ ] Decide the home (A/B/C) and client-vs-server BEFORE picking a model.
- [ ] If custom: run it in an AudioWorklet on raw audio, before Opus; set noiseSuppression:false.
- [ ] Use one noise suppressor only; verify quality with an ITU-T P.835 (SIG/BAK/OVRL) listen, not a vendor demo.