

RM Noise

AI Noise Reduction for Amateur Radio

Hear More of the Signal ... Not the Noise



Flathead Valley Amateur Radio Club
W7YP – Rick Fletcher
September 15, 2026

What is RM Noise?

- **AI-based noise reduction for amateur radio RX audio**
- **Uses neural nets trained on real signals and noise**
 - Trainable to user's own environment
- **Windows client sends RX audio to neural-net servers**
 - Looking at porting it to run locally on user's PC
- **Processed audio is returned to your PC**
- **AI filters are available for SSB, CW and other conditions**
- **Goal: improve intelligibility and copyability**



RM Noise — Development Team & Evolution

Who developed it?

K5QE

Randy Williams, K5QE

Lead developer /
project administrator

- Public RM Noise posts, documentation and forum development updates are authored by Randy.
- He develops and trains the neural-network filters and maintains the client/server system.
- The project also depends on volunteer server hosts and radio amateurs who contribute real-world noise recordings.

Community-assisted, developer-led project

How RM Noise developed



2023

Training from real ham-radio noise

Users record station/location-specific noise; recordings are screened and incorporated into AI training datasets.



Jan 2024

Multi-server Windows client

v0.21.9.7 added automatic probing, latency/packet-loss scoring and server selection; servers existed in Texas and Switzerland.



2024

Community infrastructure grows

Volunteer-host model expands geographic coverage; Singapore server announced in November 2024 to reduce regional latency.



2025–26

Web client + new AI training

A different web-client audio architecture required modified neural-network/training work and extensive GPU experiments.

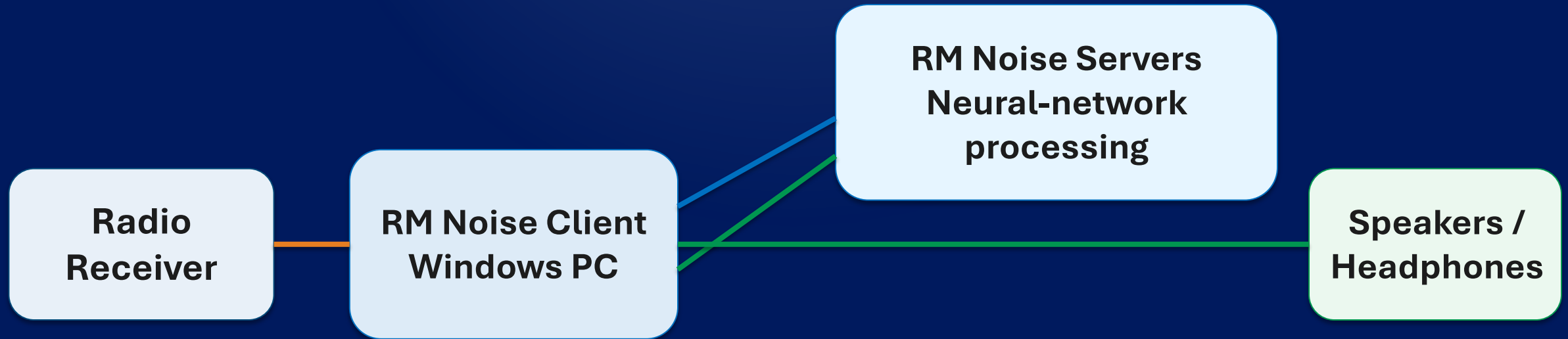


May 2026

Web client v0.421

Development model: real amateur-radio noise → curated recordings → GPU training/experimentation → deployed AI filters → user feedback → next generation

System Architecture

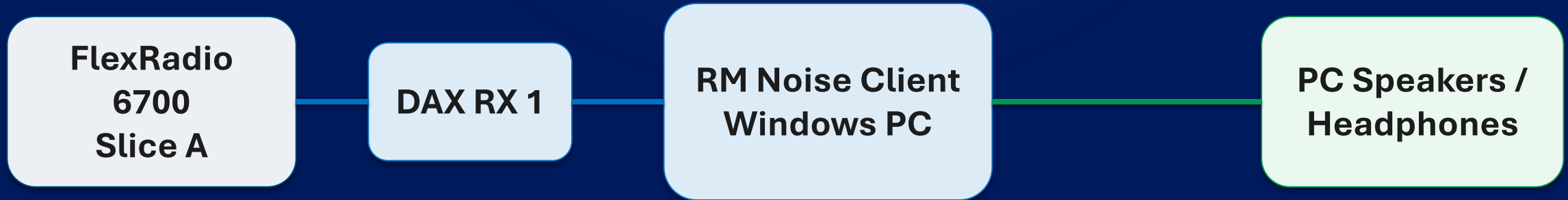


The AI runs on the server — not on the local PC

Getting Started

- 1 Create an RM Noise account
- 2 Install the Windows client
- 3 Log in with your callsign and password
- 4 Select the receiver-audio input and speaker/headphone output
- 5 Choose the AI filter that matches the mode
- 6 Start with a noisy signal and compare Original ↔ Filtered

Connecting a FlexRadio with DAX

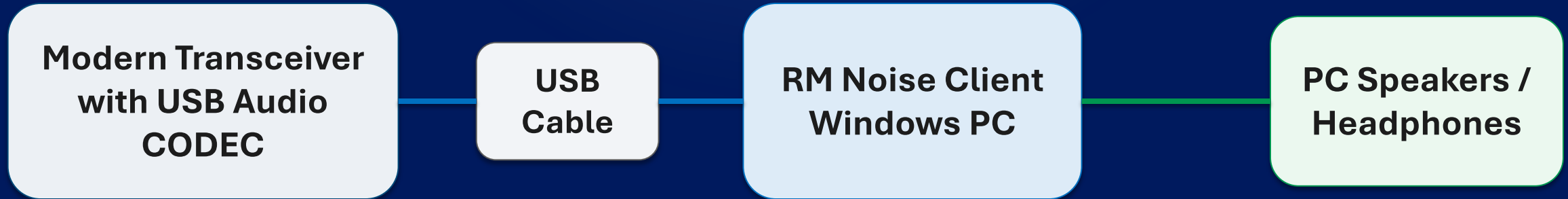


SmartSDR: assign Slice A to DAX channel 1

RM Noise input: MME – DAX RX 1 (FlexRadio DAX)

Avoid a second direct SmartSDR audio path to the same speakers — it creates an echo.

Connecting a Radio with a Built-In USB Audio CODEC



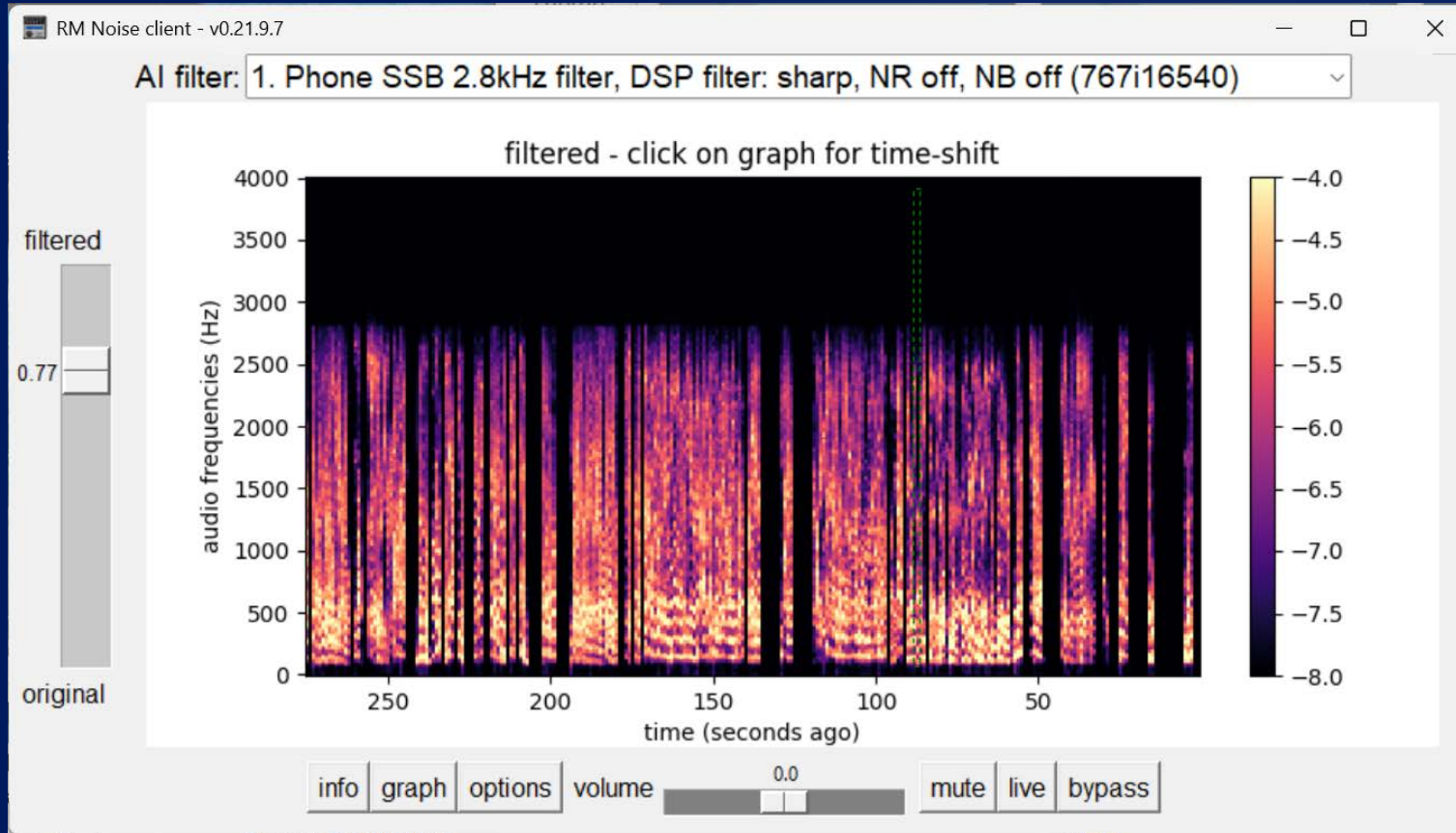
- Windows sees the radio's USB CODEC as an audio input/output device
- Select the radio's receive-audio CODEC as the RM Noise input
- Use the PC sound device for listening to the processed audio
- Examples include many current Icom, Yaesu and Kenwood transceivers

Audio Settings: Give the AI a Clean Signal

- **Start with the radio's own NR/NB/WNB processing OFF unless called for**
- **For SSB, a receive bandwidth around 2.8 kHz is a good starting point**
- **Use enough audio level for a healthy signal without clipping or distortion**
 - **Keep it below -5.5 on the meter**
- **Disable microphone monitoring and other PC audio enhancements**
- **Use headphones or good speakers when comparing subtle changes**

Best results start with clean receiver audio — let RM Noise do the noise reduction.

RM Noise Client – Main Window



What you control:

- AI filter selection
- Original ↔ Filtered mix
- Volume and mute
- Live / bypass
- Time-shift graph
- Input/output devices

Choosing an AI Filter

SSB

CW

**Other /
Experimental**

- **Filters are trained for particular modes, bandwidths and DSP conditions**
- **For phone: start with the current 2.8-kHz SSB filter**
- **For CW: choose the CW model and a receive bandwidth appropriate to the signal**
- **Optional post-filter CW bandpasses can further narrow the recovered tone**
- **Use the Original ↔ Filtered slider to judge the result on the same signal**

What to Expect on CW

ORIGINAL



Noise and competing energy can mask a weak tone

FILTERED



The desired CW tone can become much easier to copy

**RM Noise does not create missing information —
it tries to recover the signal already present**

What to Expect on SSB

ORIGINAL



Speech buried in broadband noise can be tiring or difficult to understand

FILTERED



Background noise often falls dramatically while speech remains intelligible

Prerecorded File Demonstration

1

Select “play a recording”

2

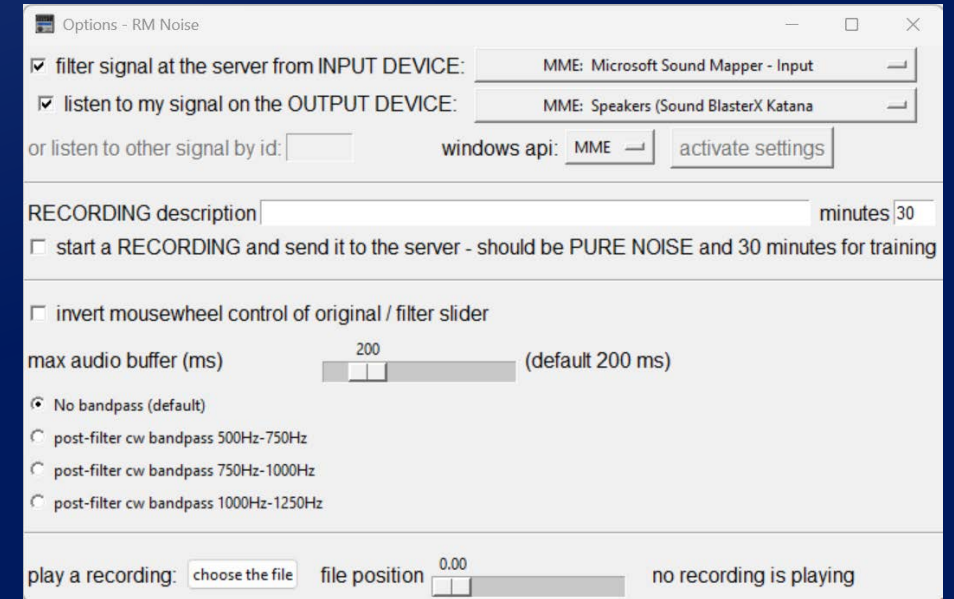
Select the matching RM Noise AI filter

3

Move Original ↔ Filtered while listening

4

Use Bypass for an instant before/after comparison



Original/Filtered allows the comparison of the same instant of received audio

What to Expect

- ✓ Large reduction in many kinds of background noise
- ✓ Improved intelligibility of weak SSB signals
- ✓ CW tones that are easier to hear and copy
- ✓ Marginal signals may become usable
- ✓ Results vary with signal strength, fading and noise type

Think “better copy,” not “perfect audio.”

Limitations and Tradeoffs

**Internet
required**

Audio leaves the PC, is processed remotely, and returns

Latency

Expect a noticeable delay — typically a fraction of a second

**Not all
interference**

Strong QRM, severe fading or distortion may remain difficult

**Filter
dependent**

The wrong model or bandwidth can make results worse

**Receive audio
only**

This is an RX audio enhancement tool, not a TX processor

The RM Noise Server Network

A small distributed GPU (Nvidia RTX 5090) network — geography matters because every filtered audio stream makes a round trip



January 2024

2 servers in Texas
1 server in Switzerland

November 2024

Singapore server added to reduce regional latency and packet loss

Current status page

Four named servers: rmnoise1–rmnoise4
Latest check: 3 normal; rmnoise3 offline

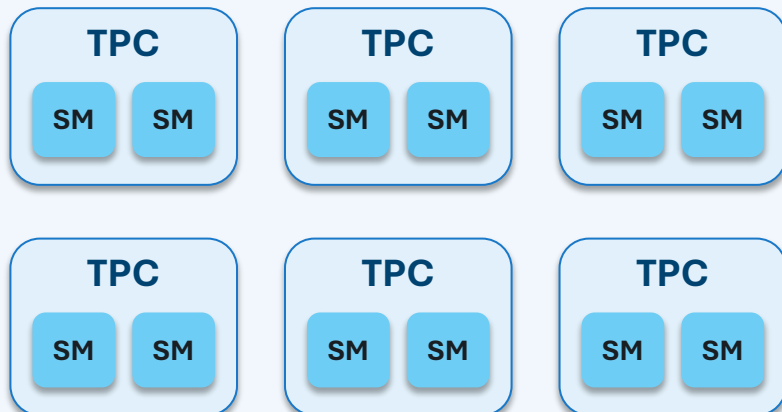
Public RM Noise material does not establish which rmnoise number maps to each geographic location

GeForce RTX 5090 Architecture — NVIDIA Blackwell GB202

Compute hierarchy

RTX 5090: 11 GPCs

Graphics Processing Clusters



85 TPCs • 170 Streaming Multiprocessors

Each SM: 128 CUDA + 4 Tensor + 1 RT core

Inside a Blackwell SM

**128 CUDA Cores
FP32 / INT**

4 × 5th-Gen Tensor Cores
AI • FP4 acceleration

1 × 4th-Gen RT Core
Ray tracing

**L1 / Shared Memory
+ Register File**

170 SMs × these resources

Memory & data movement

32 GB GDDR7

512-bit memory bus

1,792 GB/s bandwidth

128 MB L2 cache

PCIe Gen 5 • DisplayPort 2.1b

21,760 CUDA cores

680 Tensor cores

170 RT cores

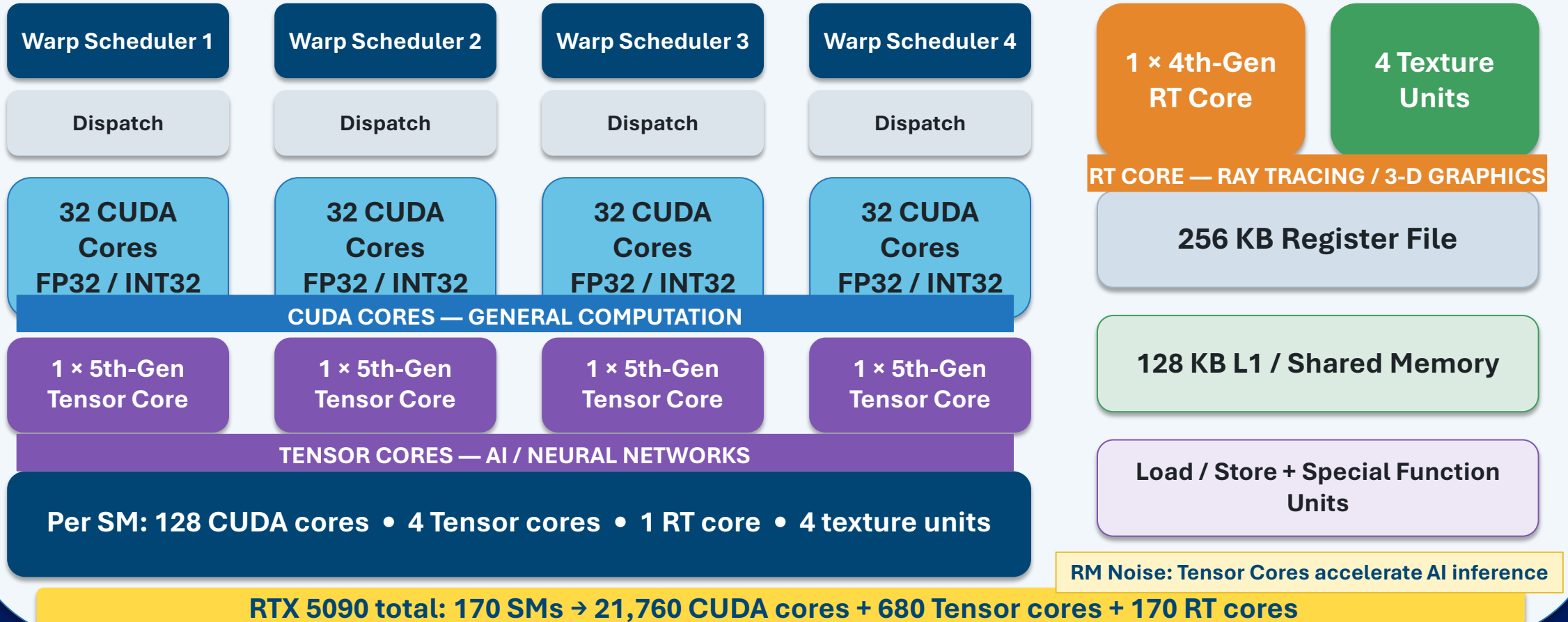
104.8 TFLOPS FP32

**Blackwell • Compute
Capability 12.0**

RTX 5090 — Blackwell Streaming Multiprocessor (SM)

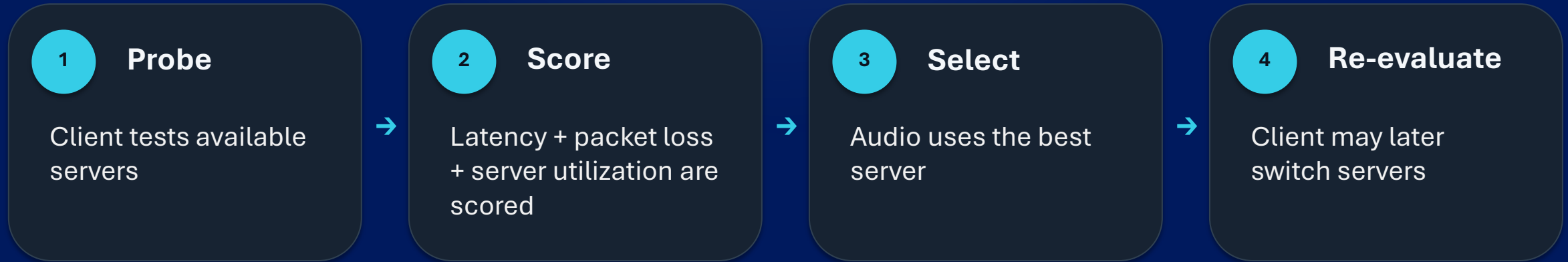
One SM is the fundamental compute block repeated 170 times in the GeForce RTX 5090.

BLACKWELL SM



Why Server Geography Matters

RM Noise chooses among available servers using measured latency and packet loss



Why add more continents?

RM Noise says round-trip Internet latency is a major concern and hopes for at least one centrally positioned server on each continent

Coverage gaps remain

Published locations leave large gaps — Australia/New Zealand, South America, Africa and much of Asia

More servers also mean redundancy, greater user capacity and maintenance without total downtime

Tips for Best Results

- **Start with a good antenna and a reasonably clean receiver setup**
- **Feed RM Noise undistorted audio at a sensible level**
- **Match the AI filter to the mode and bandwidth**
- **Compare Original and Filtered — don't judge by noise level alone**
- **Listen for intelligibility, copyability and fatigue reduction**
- **Keep a demo WAV available for repeatable testing**

RM Noise Summary

AI Noise Reduction for Amateur Radio

Cloud AI

Neural-network processing on remote servers (for now)

Flexible Inputs

FlexRadio DAX, USB CODECs and other PC audio sources

SSB + CW

Mode-specific filters for common ham-radio listening

A/B Comparison

Original ↔ Filtered makes the improvement easy to hear

Same Signals. More Conversations.

Resources and Further Reading

- **RM Noise official site:** [RM Noise – Removing noise with AI](#)
- **Windows Client:** [New client v0.21.9.7 – RM Noise](#)
- **Registration:** [REGISTRATION – RM Noise](#)
- **Official documentation:** [DOCUMENTATION – RM Noise](#)
- **RM Noise forum:** [Removing Noise with AI - RM Noise - Index page](#)
- **Demo recording link appears in the RM Noise documentation**