

Victor Gaydov

Real-Time Audio Software Engineer



- Email: victor@enise.org
- Website: gavv.net
- LinkedIn: [linkedin.com/in/victor-gaydov](https://www.linkedin.com/in/victor-gaydov)
- Upwork: [upwork.com/o/profiles/users/~01205fd34b306ddfd6](https://www.upwork.com/o/profiles/users/~01205fd34b306ddfd6)

Summary

I've been working in software engineering for over 15 years.

Nowadays I help companies build latency-sensitive audio systems and performance-critical native applications for embedded, desktop, and mobile platforms.

My core expertise is turning complex low-level requirements into reliable and scalable architectures and implementations.

Open Source Work

Below is a list of the most significant open-source projects that I've built and maintain. You can find the complete list on my website.

Roc Streaming • Co-founder

2015–Present · github.com/roc-streaming

Roc Streaming is an open-source ecosystem for real-time audio streaming. I am one of the founders and core maintainers.

It consists of reusable components of different abstraction levels: libraries, services, sound server integrations, virtual devices, and end-user apps.

The ecosystem core, Roc Toolkit, implements real-time audio streaming with guaranteed latency, loss recovery, adaptive adjustment to network properties, and other essential features.

Other technically challenging projects worth mentioning here are Roc VAD (network virtual audio device for macOS) and rocd (audio streaming daemon).

libASPL • Author

2021–Present · github.com/gavv/libASPL

libASPL is a C++17 library for creating virtual audio devices for macOS. It saves you thousands of lines of code and makes it much easier to get started with Core Audio HAL plugins.

Signal Estimator • Author

2020–Present · github.com/gavv/signal-estimator

signal-estimator is an audio engineering tool for measuring signal latency and its other characteristics.

I often build project-specific software+hardware test setups and use this tool to measure end-to-end system latency in different scenarios: local I/O, network streaming, virtual device I/O, synchronization accuracy, etc.

WebRTC CLI • Author

2019–Present · github.com/gavv/webrtc-cli

webrtc-cli is a command-line tool for streaming to and from audio devices and files via WebRTC. It supports PulseAudio, Opus (with FEC and PLC), and has simplistic clock drift compensation.

Founder Work

ViveSound • Co-founder

2024–Present · vivesound.com

ViveSound builds a commercial audio engine for system-wide audio routing and processing. It gives control over audio flows between apps, devices, and effects.

I initially joined the company as a freelance engineer, designed and implemented its core audio engine, and helped shape the product's technical direction. Later I became an equity partner.

Responsibilities:

1. Technical direction and architecture.
2. Turning product goals into technical plans and priorities.
3. Core audio engine development.
4. Release planning, acceptance criteria, and quality-control coordination.

Freelance Projects

Below is a list of the most interesting freelance contracts that I've completed. You can find most of them, and some others, on my Upwork profile.

Probabilistic NAT traversal for WebRTC

2025

Implemented NAT Hole Punching algorithm based on birthday paradox probing.

- Establishing P2P connection in non-trivial scenarios when standard STUN doesn't work.
- Custom STUN client stack.
- Integration with WebRTC / libjuice.
- Targeted embedded Linux.

Real-time audio streaming over Internet

2023–2025

Implemented real-time audio streaming daemon for embedded Linux.

- Guaranteed latency below 40 ms over Internet.
- Streaming high-quality audio.
- Forward error correction (recovery of lost packets).
- Packet loss concealment (masking of unrecoverable losses).

Airborne audio synchronization

2023

Implemented real-time audio synchronization software based on embedded audio signatures.

- Embedding and extraction of audio signatures into acoustic (over-the-air) signal.
- Aligning network playback with the in-air stream with a precision of 1-2 milliseconds.
- Targeted embedded Linux with ALSA.

Distributed playback synchronization

2023

Developed proof-of-concept software for synchronized playback across multiple independent network stream receivers.

- Worked reliably across high-jitter Internet connections.
- Achieved sub-millisecond synchronization precision.

Speech-to-text cloud server

2023

Developed a small speech-to-text web server using OpenAI Whisper.

Audio processing injection using virtual devices

2020–2023

Designed and implemented a solution for audio processing injection on macOS and Windows.

- macOS VAD (Virtual Audio Device) and Windows APO (Audio Processing Object).
- Low-latency audio I/O and IPC.
- Robust multi-service architecture.
- Plugin management system.

Software hearing aid for Android and iOS

2019–2020

Designed and implemented a software hearing augmentation system for Android (Oboe/AAudio) and iOS (CoreAudio, AVFoundation).

Fake Bluetooth headset with WebRTC streaming

2019

Designed and implemented a real-time audio streaming daemon with REST API.

- Real-time streaming between desktop (PulseAudio), mobile (Android), and cloud storage using WebRTC / Opus (using Go and Pion).
- Fake headset (using PulseAudio) for nearby Bluetooth devices with sound capture/injection.
- Low-level control over Bluetooth devices.

Employment History

Team lead • TradingView

2016–2019

Developed a high-load concurrent HTTP proxy for an internal time-series protocol, written in Go (stream multiplexing, routing, balancing, caching).

Responsibilities:

- Leading a team of 5-10 developers and testers.
- Architecture and code review (this part took most of the time).
- Development and performance testing.
- Planning sprints and releases.

Software Engineer • Zodiac Interactive

2014–2016

Developed software for embedded Linux for set-top boxes (ARM, MIPS).

1. Developed a library stack for video storage on the set-top boxes (for DVR and TimeShift functions). Implemented writing and reading real-time video streams, on-disk ring buffers, multi-process synchronization.
2. Participated in development of a platform abstraction layer for a legacy system. Worked on caching, asynchronous fetching, and a download scheduler.
3. Participated in porting of Chromium to Linux/MIPS.

Software Engineer • InfoTeCS, JSC

2012–2014

Developed software for desktop Linux and MikroMedia SoC.

1. Developed a client-server X11-like graphical stack, with a server running on for MikroMedia board with a touchscreen, and client running on Linux PC.
2. Developed software serial-over-ip for Linux, using CUSE (character device in userspace).
3. Maintained a C++ library for ASN.1 and crypto tokens (ruToken, eToken).

4. Ported legacy codebase to PowerPC.

Software Engineer • NTMR, LLC

2010–2012

Developed software for Linux, Windows, and UCOS-II for a millimeter-wave scanner of an access control system.

1. Developed firmware for Altera SoC with UCOS-II RTOS in C and Verilog.
2. Developed desktop libraries for peripheral control (TCP, CAN, LPT) and real-time gigabit measurements.
3. Developed GUI apps using Qt, Qwt (for real-time plots), OpenCV (for video capture and marker detection), and OpenGL (for raycast renderer).
4. Implemented random-forest ML algorithm with C++/MPI running on computing cluster.
5. Implemented various MEX files and MATLAB scripts.

Technical Writing

You can find all my articles and blog posts here: gavv.net/articles

Below is the list of the most interesting ones.

- **PulseAudio under the hood**

Long-form (150 pages) in-depth article about PulseAudio architecture, interfaces, algorithms, and more. Was well received and nowadays is cited from official PulseAudio docs.

- **Working on a new network transport for PulseAudio and ALSA**

A pre-announcement for the first release of Roc Toolkit (a real-time streaming library) with some interesting technical details. It was also well received on Reddit and HN.

- **File locking in Linux**

Low-level details about different file-locking mechanisms available in POSIX and Linux.

- **Decoding and playing audio files in Linux**

A tutorial to various media libraries on Linux, partially outdated nowadays.

Expertise

Audio engineering

- Building scalable, optimizable architectures that sustain real-time guarantees as the system evolves.
- Low-latency audio systems: audio streaming, audio I/O, synchronization, virtual audio devices.
- Cross-platform audio systems: Linux (ALSA, PulseAudio, PipeWire), macOS/iOS (CoreAudio, AVFoundation, HAL plug-ins), Windows (WASAPI, Windows APO), Android (Oboe, AAudio).
- Strong knowledge of real-time media protocols: WebRTC, RTSP, RTP/RTCP, FECFRAME.
- Experience with media frameworks and libraries: FFmpeg, GStreamer, Opus, SpeexDSP, OpenFEC.

Systems programming

- Strong Linux and POSIX systems programming experience: networking, asynchronous I/O, performance.
- Strong experience with multithreading, concurrency, and lock-free algorithms.
- Strong experience with embedded systems development on Linux and non-Linux platforms, targeting ARM, MIPS, RISC-V, AVR.
- Experience with backend development and high-load web services.
- Experience with distributed and scientific computing.
- Proficient in C++, C, Go, Python, and UNIX scripting. Familiar with Rust, Java, C#, and Objective-C.