

Technical Report: Low-Latency ASR Optimization and Scalable Media Pipeline at Streamwise

Eduard Anton
Streamwise
Montreal, Canada

Frédéric Jarjour
Streamwise
Montreal, Canada

1 Motivation

The rapid growth of live streaming has introduced significant challenges in maintaining safe and compliant digital environments. Streamwise [6] addresses these challenges by providing a production-grade platform for low-latency, proactive moderation. Unlike reactive systems that process content after broadcast, this architecture enables real-time, context-aware speech moderation where harmful or non-compliant content can be filtered out before it reaches the destination platform. By integrating high-concurrency media pipelines with advanced language models, Streamwise enables automated monitoring and intervention to preserve platform integrity without introducing perceptible delays for the end-user.

2 System Architecture

The backend platform is built on a high-concurrency infrastructure designed for sub-3-second end-to-end latency. As shown in Figure 1, the pipeline utilizes Simple Realtime Server (SRS) [7] as the primary platform for ingesting Real-Time Messaging Protocol (RTMP) streams. Upon receiving a new request, the system executes an HTTP-based authentication handshake with a separate API server to validate user credentials before allowing the stream to persist.

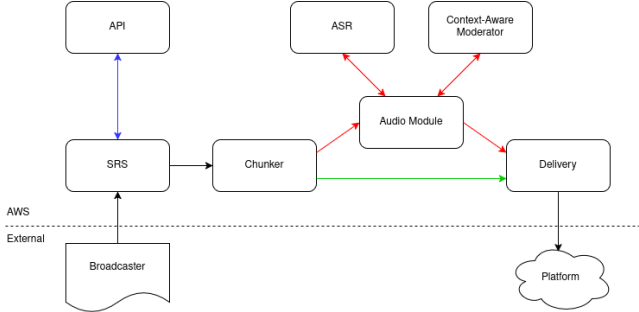


Figure 1. Architecture of the Streamwise media pipeline. Data flows via RTMP (black) to SRS with HTTP (blue) authentication. The Chunker demuxes signals into synchronized video (green) and audio (red) paths for processing.

At the core of the data flow is a Chunker Module, implemented in Go, which reads the incoming stream to demux

and synchronize interleaved audio and video components. This module serves as the primary router, ensuring temporal alignment across the decoupled processing stages. Decoded audio chunks are dispatched via a Round Robin scheduler to an Automatic Speech Recognition (ASR) module for transcription. Following transcription, the resulting text and associated metadata are processed through a proprietary, context-aware moderation pipeline. The delivery module finally receives the processed audio and the raw video stream, re-synchronizing and packaging them into a final RTMP stream for broadcasting to external platforms such as Twitch or YouTube.

3 ASR Model Optimization

To achieve near state-of-the-art performance, the system employs a Conformer-based encoder [5] and a Token-and-Duration Transducer (TDT) decoder [8] architecture within the NeMo PyTorch framework. The development process utilized an NVIDIA NeMo Parakeet-TDT 600M parameter model [3] as the high-fidelity parent baseline. Optimization began with a smaller 360M parameter variant, which underwent an initial supervised training phase using 960 hours of the LibriSpeech dataset [4]. To further refine this model for the nuances of live streaming, a second phase of reinforcement learning was conducted via on-policy distillation [9]. During this stage, the 360M parameter model acted as a student, optimizing its policy by following the reflective thinking traces of the 600M parameter parent model. This distillation process used a custom augmented version of the Mozilla Common Voice dataset [2], incorporating LLM-driven data augmentation and synthetic speech-to-text generation to improve robustness.

4 Results and Performance

System performance was evaluated on the GigaSpeech benchmark across three model iterations [1]. The parent baseline model achieved a Word Error Rate (WER) of 9.59% [3]. In comparison, the student model trained purely on LibriSpeech and the augmented Mozilla dataset without distillation reached a WER of 13.0%. By applying on-policy distillation, the final model improved significantly, reaching a WER of 11.1%. From a business perspective, this approach maximized inference density on the deployment, decreasing infrastructure costs per user by approximately 30–35% while maintaining moderation quality.

References

- [1] Guoguo Chen et al. 2021. GigaSpeech: An Evolving, Multi-Domain ASR Corpus with 10,000 Hours of Transcribed Audio. In *Proc. Interspeech 2021*. 3670–3674. doi:10.21437/Interspeech.2021-1965
- [2] Mozilla. 2021. *Common Voice Dataset*. <https://commonvoice.mozilla.org/>
- [3] NVIDIA. 2024. *Parakeet-TDT-0.6b-v3*. <https://huggingface.co/nvidia/parakeet-tdt-0.6b-v3>
- [4] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *Proc. ICASSP*. 5206–5210. doi:10.1109/ICASSP.2015.7178964
- [5] Dima Rekesh et al. 2023. Fast Conformer with Linearly Scalable Attention for Efficient Speech Recognition. *arXiv preprint arXiv:2305.05084* (2023).
- [6] Streamwise. 2026. *Streamwise: Real-time Media Moderation*. <https://www.streamwise.gg/>
- [7] Winlin et al. 2026. *SRS (Simple Realtime Server)*. <https://github.com/ossrs/srs> Accessed: 2026-03-16.
- [8] Hainan Xu et al. 2023. Efficient Sequence Transduction by Jointly Predicting Tokens and Durations. *arXiv preprint arXiv:2304.06795* (2023).
- [9] Siyan Zhao et al. 2026. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. *arXiv preprint arXiv:2601.18734* (2026). <https://arxiv.org/abs/2601.18734>