

GStreamer

State of the Union

Tim-Philipp Müller

tim@centricular.com

Prague, 10 October 2026



Welcome to Prague!

Nice to see everyone again!



What have we been up to?



Stats

- **monorepo**: 2915 commits
- **plugins-rs**: 771 commits
- **rust bindings**: 340 commits
- **cerbero**: 490 commits
- **Total** 4516 commits
 - ~28% in Rust

... since the last conference



Releases

- **1.22:** January 2023
- **1.24:** March 2024
- **1.26:** March 2025
- **1.28:** January 2026
- **1.30:** ??

1.28 stable series

- **8 bug-fix releases** since end January
 - **every ~3-4 weeks**
- 571 backported merge requests
- 1024 backported commits
 - almost 6 per working day!
 - **~ 1 per hour!**

Upcoming 1.30 stable

- Target: **December 2026**, on track so far:
 - Had **several 1.29 development snapshots**
 - Next one after the hackfest
- Next steps:
 - **feature freeze**: ~ 15 Nov ?
 - **API freeze**: ~ 1 Dec ?

Let's look at some things
people have been
working on



Codecs

- AC-4 audio parsing
- HDR10+ metadata from H.265 and AV1
- Improved support for codecs with alpha channel
 - hardware-accelerated paths
- Lots of smaller improvements across the board



Video encoders

- **adaptive presets** with resolution-dependent properties
- implemented in x264enc and nvh264enc so far

Closed Captions

- Generic **h264seiinsertter** and **h265seiinsertter** elements for **closed captions** and unregistered user data **SEIs**

Adaptive streaming

- HLS, CMAF muxing improvements
- **HLS WebVTT sink** element
- New **DASH CMAF sink**

Speech, Text, Translations

- New **whisper**-based STT element for **local use**
- **translationbin**: additional latency fine-tuning controls
- **llama.cpp**-based text transformation



Digitally Signed Content (DSC)

- Support for **H.266 + H.265** video
 - parse bitstreams with DSC SEI data
 - inject DSC metas into the bitstream
- **DSC** plugin and **CP2A helper library** (in gitlab)
 - Content Provenance and Authenticity

Analytics

Lots of work happening in this area!

New Mtds, more models, tensor decoders.

Lots of talks too!



Analytics: ONNX

- More execution providers:
 - DirectML
 - MlGraphX
 - Windows ML
 - Vitis AI
 - AMD HIP
 - Android NNAPI + OpenVINO (in gitlab)

Network elements

- **SRT** connection **multiplexing** support
- New **udpsink2** and **multiudpsink2** elements
- New **PCAP** parser element in Rust
 - with **IPv6** support

RTP

- **rtpbin2**: RFC7273 support in receiver
- **JPEG XS** (de)payloader
- **Grayscale support** in **raw video** (de)payloader

RTP II

- New **Rust RTP (de)payloaders** for MPA audio, MPEG-2 video, raw video, Vorbis
- Bump ranks of the new Rust RTP (de)payloaders to PRIMARY
- Default to **MTU 1200** for new Rust payloaders

RTSP

- Rust **rtspsrc2** gained many missing features
 - authentication, SRTP
 - HTTP tunneling, keep-alive
 - stream selection, TLS validation
 - latency configuration
- **Future:** audio backchannel + VOD support

WebRTC

- **webrtcbin2**: New scalable WebRTC bin
 - **native Rust** DTLS, ICE and RTP session management
 - uses **fewer threads**, async runtime

WebRTC II

- **webrtc sink + src**: make custom signalling protocol extendable
- **webrtc sink**: web_server_embedded feature
- support for **RTP-carried application media**
 - e.g. KLV



Hardware-acclerated alpha codec

- New **OpenGL glalphacombine** element for RGBA inputs
- **alphacombine: Vulkan/OpenGL** memory support

Vulkan

- Improve **barrier synchronisation** with **multiple consumers**
- Lots of work on **Vulkan Video**

Direct3D12

- Element for color lookup operations using **Adobe .cube LUT files**
- Hw-accelerated decoding of **VP9 with alpha**
- Support for **resource residency management**

VA-API

- New **VA-API overlay compositor**



Video4Linux

- **v4l2src**: adds **capture timestamps** to buffers
- **video encoders**: “bitrate” and “gop-size” properties

AMD HIP support

- **GstHipHostMemory** implementation
- Support for **dmabuf importing and caching** on AMD

Tracing and Debugging

- tracing: **custom spans and events API**
- integration with the **Rust tracing** ecosystem
- **dots-viewer**: General enhancement of the web app
- tracer for **per-pipeline debug logging**



Editing Services

- Makes better use of **hardware-accelerated converters**
- Take advantage of **multi-threaded** usage in more places



Windows

- Ship the **AJA** plugin in our binary packages
- filesink: Add support for **Windows file sharing mode** to allow reading of a file while it's being written to

macOS + iOS

- **memory:IOSurface** support library
- **vtdec** now exposes per-codec elements
- Ship the **AJA** plugin in our binary packages
- **Device monitor** can report changes for video and screen devices

Android

- use **memory:AHardwareBuffer** for more efficient decode into vulkan, glupload
- New **source element for Android assets**
- We now require **Android Oreo** (API level 26)

Start-up performance and deployment

- Support for **pre-generated registry caches**
- Lower **start-up time** on Windows/macOS
- Useful also for **embedded systems**

Lots of new Rust plugins

- **webrtcbin2**, RTP (de)payloaders
- **Minus-1 audio mixer**
- image-rs based **image overlay, decode, encode** elements (to replace gdkpixbuf ones)
- **whisper transcriber**



Challenges ...



Challenges

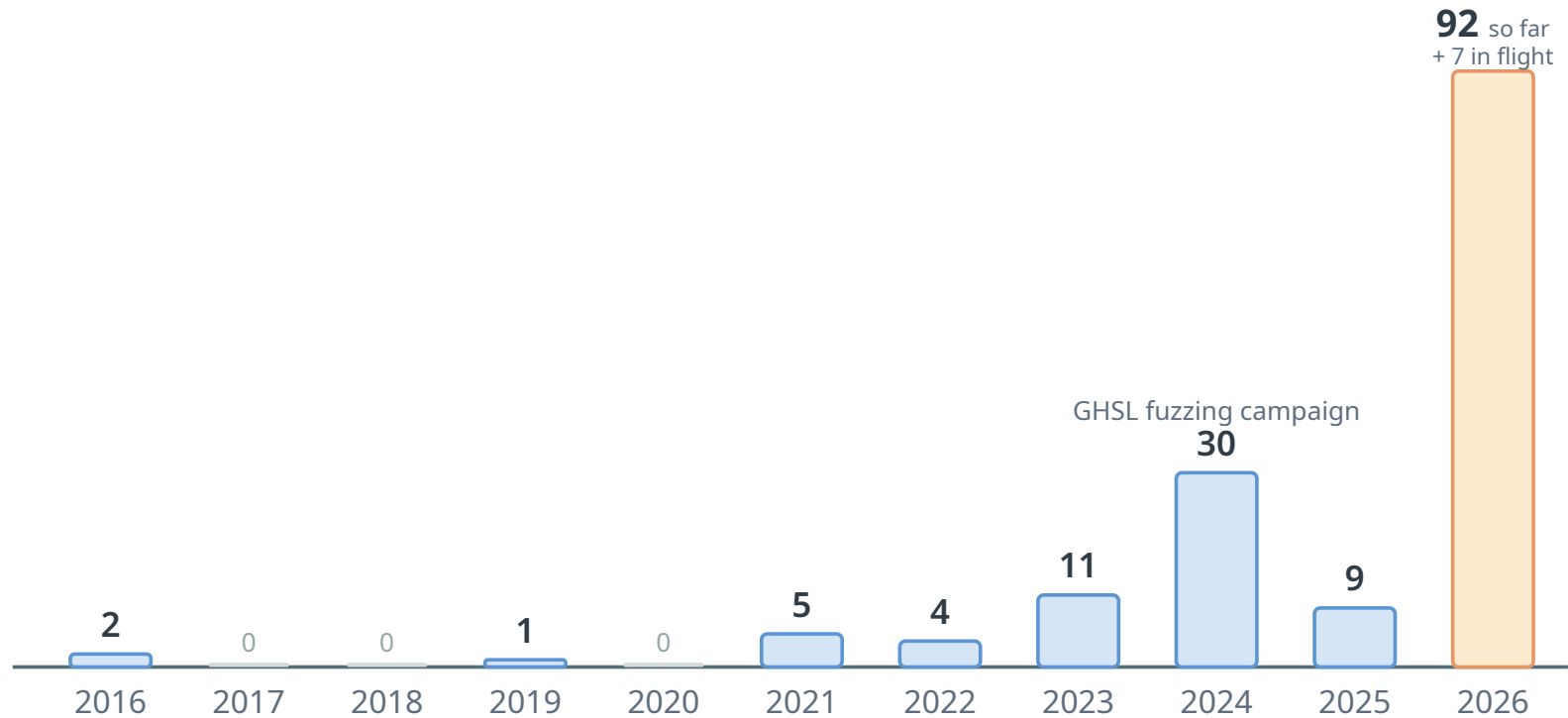
- Security
- Maintenance
- LLMs

Challenges

You're absolutely right to raise that, Tim!

What a brilliant observation!

Security



Security

- New security report every 2-3 days
- In batches of course
- Most are minor and trivial
- Have to play the security theatre game

Security

- Extremely time consuming
 - super verbose reports
 - review surrounding code
 - come up with minimal fixes
 - co-ordinate release

Security: Full Disclosure

In June we were ready to throw in the towel.



Security: Full Disclosure

In June we were ready to throw in the towel:

- Most security issues are bugs like all other bugs
- We routinely fix similar issues on a daily basis
- Can't justify crowding out everyday maintenance
- Have work to do too from time to time



Security

But we got lucky!

Sovereign Tech Agency

to the rescue!

But: When funding runs out, it's back to square one.



Security: Questions

Who here cares about security?



Security: Questions

Who has made resources available to open source maintainers for security maintenance?



Security: Questions

Who relies on their base distro for security fixes?
(spoiler: you can't)



Security - what to do?

- Rewrite in a memory-safe language
 - already doing
- Proactively look for issues
 - ENOTIME
- Go see Sebastian's talk

Let's talk about Maintenance



Maintenance: stats in perspective

- 4516 commits in 9 months
- ~500 commits per month
- ~24 commits per work day
- ~3-4 commits per work hour
- **~ 1 commit every 15-20 minutes!**

Main branch only.

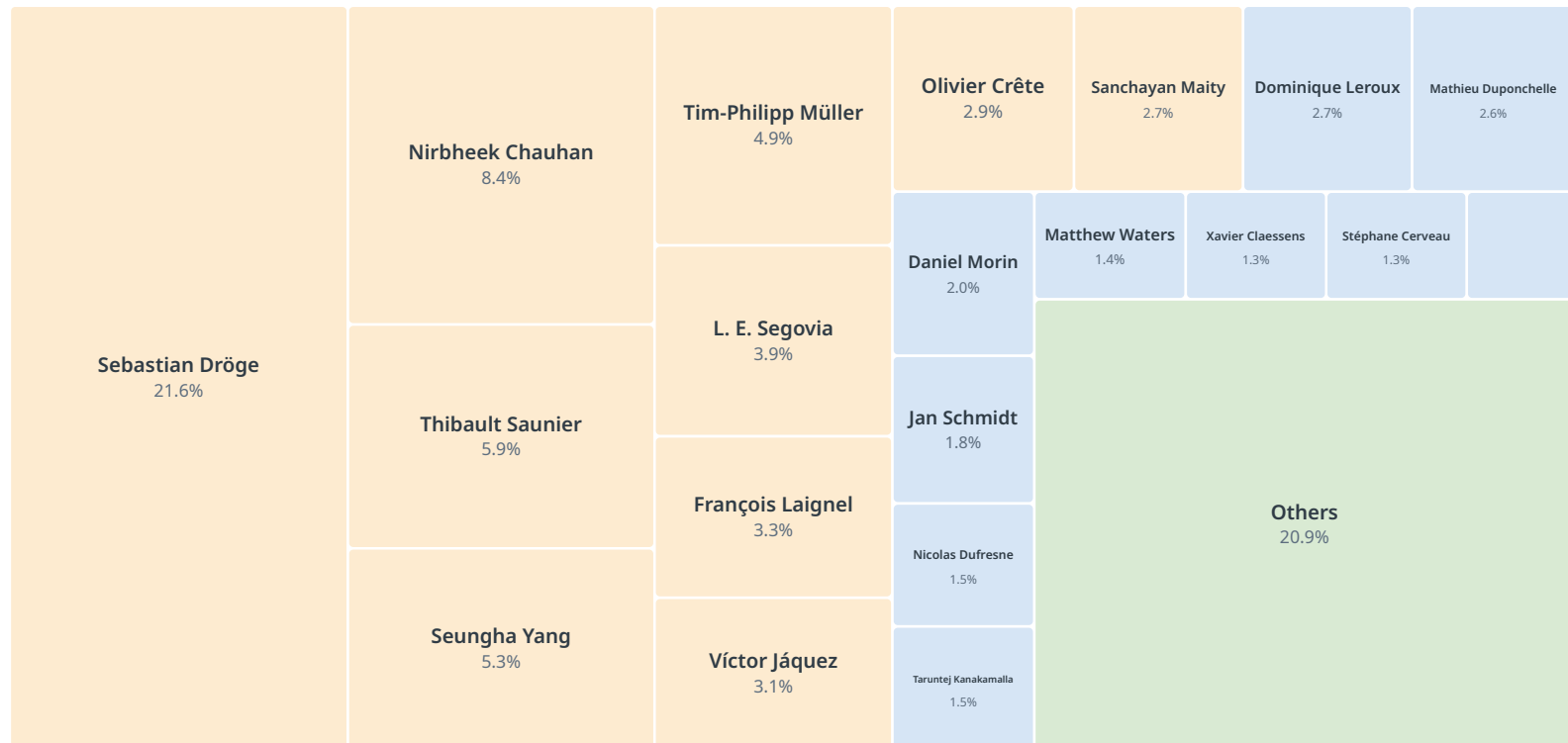


Contributors - Individuals

GStreamer: top contributors by commits

monorepo, plugin-rs, rs and cerbero

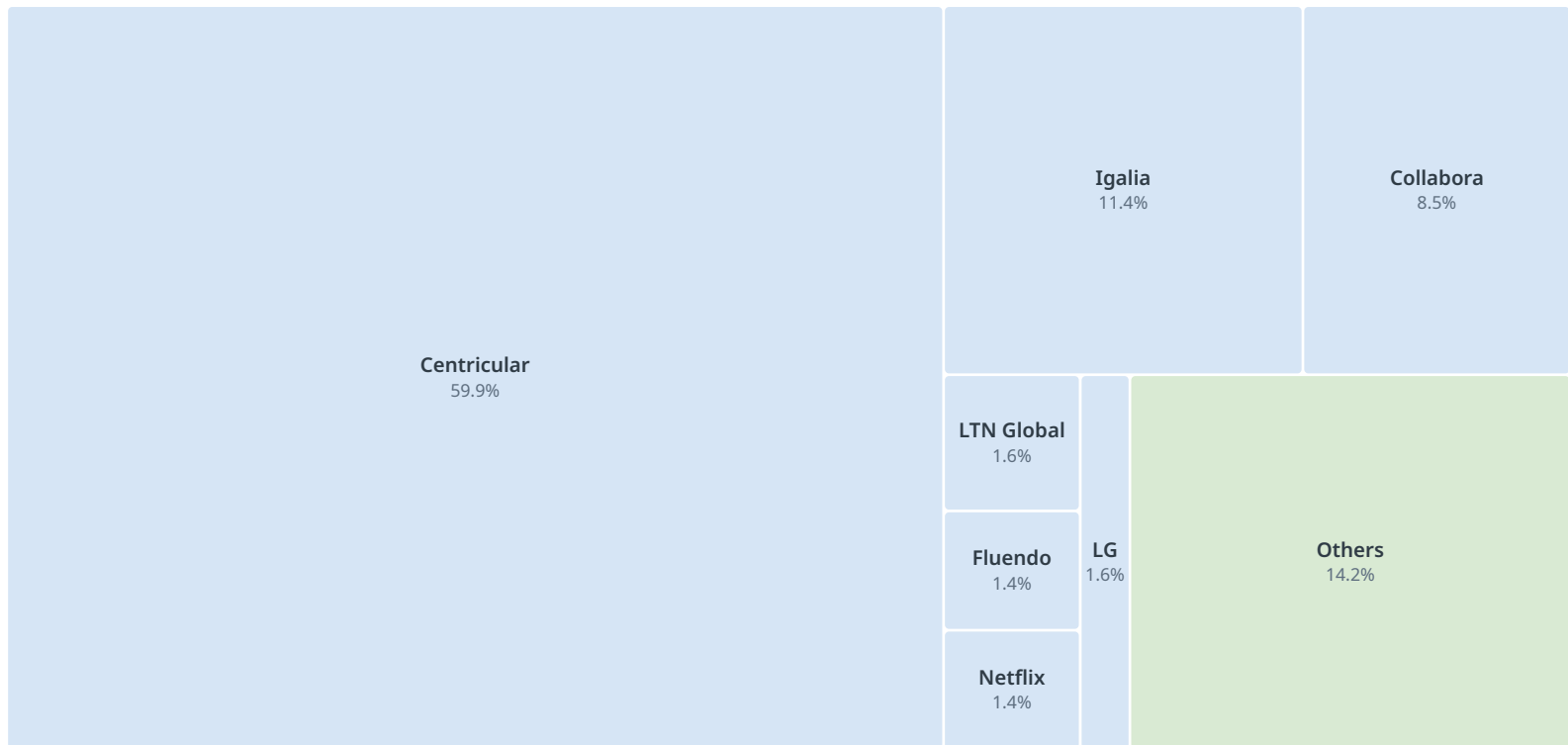
top 10 ranks 11+ others



Contributors - By Affiliation

GStreamer: top companies by commits

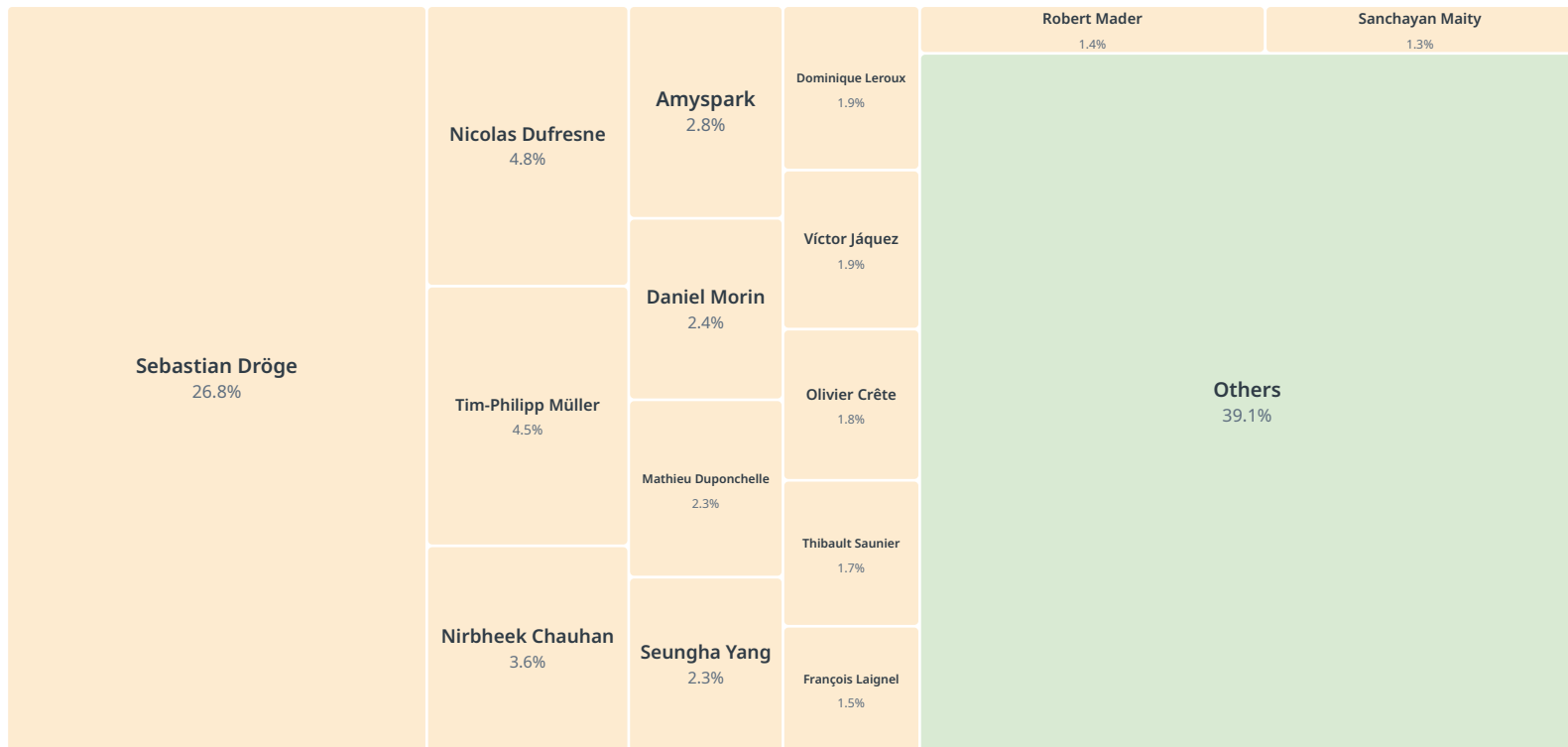
monorepo, plugin-rs, rs and cerbero



GitLab activity - Individuals

GStreamer: top contributors by comments

monorepo, plugin-rs, rs and cerbero



Maintenance Summary

- Very **concentrated**, long tail
- **Sebastian does too much**
- Maintenance **pressures are increasing**
 - LLMs in general, verbosity, volume, quality
 - Extra load from Security issues
- Consultancies are being squeezed
- **No one pays for maintenance**

The Future ...



More Analytics

- **Preprocessing** of model input
- **Postprocessing** of model output
- **Multi-modal models:**
 - image, text, audio input
 - text output
- **Cascading / chaining** of models

moar Rust



Media over QUIC (MoQ)

- Builds on top of QUIC (d'oh)
- WebTransport (server-client model)
- Simpler replacement for WebRTC for certain low-latency use cases
- MoQ publisher and subscribe in plugins-rs MR



MSE source goes Rust



New MPEG-TS demuxer in Rust



More Rust rewrites!

- Parsers
- Demuxers
- Muxers
- More RTP elements

... thanks to the **Sovereign Tech Agency!**



It's not enough

We need to be more radical!



Accelerating our move to Rust

- **Rust plugins will no longer be optional**
- We need to **remove C code**
- Provide **drop-in replacements for C plugins**



Accelerating our move to Rust

- Monorepo modules will **require a Rust compiler**
- We have **documented** the most **essential** Rust plugins for distributors



Accelerating our move to Rust

We will need **your help** with testing!



That's all folks!



Thank You!

Questions?



Enjoy the Conference!

