

Whither GStreamer 2.0?

Tim-Philipp Müller

tim@centricular.com

Prague, 11 October 2026



Why?



Why?

- GStreamer 2.0 is something GStreamer developers occasionally talk about at hackfests and conferences, but details of that rarely make it out into the wider community.



Why?

- Good to provide some more context to the wider community of what GStreamer 2.0 might look like, or at least what direction it might be going, and gather thoughts and input early in the process.



Why?

- Toward that end, there are first a couple of talks, followed by a freeform panel discussion.

Disclaimer

- These are my opinions only
- No decisions have been taken
- No plan or even consensus yet

Important

! Don't panic !



Predictions



GStreamer 2.0 will be a native Rust API



There will be no GObject
or GLib as part of that
native Rust API



There will be no GObject or GLib as part of that native Rust API

We may have to reinvent some of the functionality GObject provided.



There will still be a C API



There will still be a C API

- It will be on top of the native Rust API
- It may be more limited and not expose the full API



There will still be a place for GObject

- for bindings
- for C plugins (?)

There will still be a place for GObject - Why?

- Bindings are one of our strengths
- People building frameworks on top of us are one of our strengths (NVIDIA, Amazon, Intel, etc.)



A lot of things we will want to do differently

- Memory + buffer handling ?
- Clocks and Timing ?
- Scheduling ?
- Stronger-typed caps ?

But a lot of things will be
the same



We will take inspiration from others

(PipeWire, Blender, CasparCG etc.)



But the world doesn't
need a slightly different
PipeWire clone



Our flexibility is our strength

- Caps
- Plugin system
- Easy customisation
- Extensibility (metas etc)

How do we get to 2.0 ?



How do we get to 2.0 ?

- it's a marathon not a sprint



How do we get to 2.0 ?

- it's a marathon not a sprint
- we need more than a PoC or a MVP

How do we get to 2.0 ?

Reimplementing existing
GStreamer core APIs in Rust is not enough



It's not just a coding problem

There are many open questions that we don't know how they can or should be solved yet, even if there was already agreement on the overall streaming architecture.



My ~~Plan~~ Proposal



My ~~Plan~~ Proposal

Keep calm and carry on.

Port all the major plugins to Rust, in 1.0



My ~~Plan~~ Proposal

Keep calm and carry on.

Port all the major plugins to Rust, in 1.0.

**And experiment with ideas and see what works
and doesn't.**



Then find funding for the
last part



How can you help?

Sponsor Rust porting work for plugins.



How to follow the discussion?

- There's a [GStreamer 2.0 wishlist](#) in hedgedoc
- Should probably create a Matrix channel
- There is a Next API/ABI label in GitLab
- Come to the spring hackfest!



Let's hear from other people

