

Multimedia using Rust and GStreamer

Sanchayan Maity

Who?

- ▶ Consultant Software Engineer @ **asymptotic**

Who?

- ▶ Consultant Software Engineer @ **asymptotic**
 - ▶ Open source consulting firm based out of Toronto, Bangalore & Hyderabad

Who?

- ▶ Consultant Software Engineer @ **asymptotic**
 - ▶ Open source consulting firm based out of Toronto, Bangalore & Hyderabad
 - ▶ Work on low level systems software centred around multimedia

Who?

- ▶ Consultant Software Engineer @ **asymptotic**
 - ▶ Open source consulting firm based out of Toronto, Bangalore & Hyderabad
 - ▶ Work on low level systems software centred around multimedia
 - ▶ GStreamer, PipeWire, PulseAudio

Who?

- ▶ Consultant Software Engineer @ **asymptotic**
 - ▶ Open source consulting firm based out of Toronto, Bangalore & Hyderabad
 - ▶ Work on low level systems software centred around multimedia
 - ▶ GStreamer, PipeWire, PulseAudio
- ▶ Embedded Systems background

Who?

- ▶ Consultant Software Engineer @ **asymptotic**
 - ▶ Open source consulting firm based out of Toronto, Bangalore & Hyderabad
 - ▶ Work on low level systems software centred around multimedia
 - ▶ GStreamer, PipeWire, PulseAudio
- ▶ Embedded Systems background
- ▶ C, Rust and Haskell

Who?

- ▶ Consultant Software Engineer @ **asymptotic**
 - ▶ Open source consulting firm based out of Toronto, Bangalore & Hyderabad
 - ▶ Work on low level systems software centred around multimedia
 - ▶ GStreamer, PipeWire, PulseAudio
- ▶ Embedded Systems background
- ▶ C, Rust and Haskell
- ▶ Organizing Rust and Haskell meetup Bangalore since 2018

Agenda

- ▶ Introduction to GStreamer

Agenda

- ▶ Introduction to GStreamer
- ▶ Why Rust

Agenda

- ▶ Introduction to GStreamer
- ▶ Why Rust
- ▶ Rust and GStreamer

GStreamer

- ▶ Multiplatform Pipeline based multimedia framework

GStreamer

- ▶ Multiplatform Pipeline based multimedia framework
- ▶ Bindings for various languages

GStreamer

- ▶ Multiplatform Pipeline based multimedia framework
- ▶ Bindings for various languages
- ▶ Supported on Linux, macOS, Android and Windows

GStreamer

- ▶ Multiplatform Pipeline based multimedia framework
- ▶ Bindings for various languages
- ▶ Supported on Linux, macOS, Android and Windows
- ▶ Allows building complex media processing workflows

GStreamer

- ▶ Multiplatform Pipeline based multimedia framework
- ▶ Bindings for various languages
- ▶ Supported on Linux, macOS, Android and Windows
- ▶ Allows building complex media processing workflows
- ▶ Some applications

GStreamer

- ▶ Multiplatform Pipeline based multimedia framework
- ▶ Bindings for various languages
- ▶ Supported on Linux, macOS, Android and Windows
- ▶ Allows building complex media processing workflows
- ▶ Some applications
 - ▶ GstLAL (gravitational wave data analysis)

GStreamer

- ▶ Multiplatform Pipeline based multimedia framework
- ▶ Bindings for various languages
- ▶ Supported on Linux, macOS, Android and Windows
- ▶ Allows building complex media processing workflows
- ▶ Some applications
 - ▶ GstLAL (gravitational wave data analysis)
 - ▶ PiTiVi (Video Editor)

GStreamer

- ▶ Multiplatform Pipeline based multimedia framework
- ▶ Bindings for various languages
- ▶ Supported on Linux, macOS, Android and Windows
- ▶ Allows building complex media processing workflows
- ▶ Some applications
 - ▶ GstLAL (gravitational wave data analysis)
 - ▶ PiTiVi (Video Editor)
 - ▶ amaroK, Banshee, Clementine (audio players)

GStreamer

- ▶ Multiplatform Pipeline based multimedia framework
- ▶ Bindings for various languages
- ▶ Supported on Linux, macOS, Android and Windows
- ▶ Allows building complex media processing workflows
- ▶ Some applications
 - ▶ GstLAL (gravitational wave data analysis)
 - ▶ PiTiVi (Video Editor)
 - ▶ amaroK, Banshee, Clementine (audio players)
 - ▶ Empathy (VOIP and video conferencing)

GStreamer

- ▶ Multiplatform Pipeline based multimedia framework
- ▶ Bindings for various languages
- ▶ Supported on Linux, macOS, Android and Windows
- ▶ Allows building complex media processing workflows
- ▶ Some applications
 - ▶ GstLAL (gravitational wave data analysis)
 - ▶ PiTiVi (Video Editor)
 - ▶ amaroK, Banshee, Clementine (audio players)
 - ▶ Empathy (VOIP and video conferencing)
 - ▶ Rygel (DLNA streaming server and renderer)

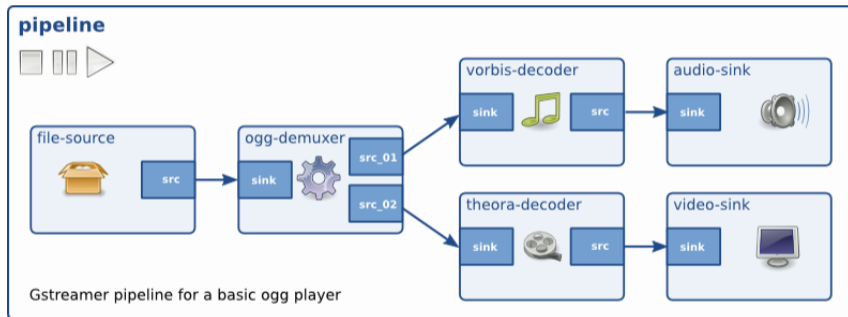
GStreamer

- ▶ Multiplatform Pipeline based multimedia framework
- ▶ Bindings for various languages
- ▶ Supported on Linux, macOS, Android and Windows
- ▶ Allows building complex media processing workflows
- ▶ Some applications
 - ▶ GstLAL (gravitational wave data analysis)
 - ▶ PiTiVi (Video Editor)
 - ▶ amaroK, Banshee, Clementine (audio players)
 - ▶ Empathy (VOIP and video conferencing)
 - ▶ Rygel (DLNA streaming server and renderer)
 - ▶ Showtime, Clapper, Totem (Media players for desktop)

Simple pipeline

```
gst-launch-1.0 videotestsrc ! autovideosink  
gst-launch-1.0 audiotestsrc ! autoaudiosink
```

Media pipeline¹

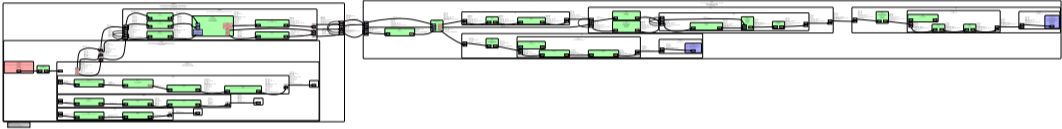


¹Dynamic Pipelines

Playback pipeline

```
gst-play-1.0 https://devstreaming-cdn.apple.com/videos/streaming/examples/  
img_bipbop_adv_example_ts/master.m3u8
```

Playback pipeline graph



Custom elements



```
gst-launch-1.0 filesrc location=bunny.mp4 ! decodebin ! videoconvert !  
roundedcorners border-radius-px=100 ! videoconvert ! gtsink
```

Why Rust?

- ▶ Codec implementations in pure Rust (Rust Audio, Xiph AV1, Symphonia)

Why Rust?

- ▶ Codec implementations in pure Rust (Rust Audio, Xiph AV1, Symphonia)
- ▶ Things we care about

Why Rust?

- ▶ Codec implementations in pure Rust (Rust Audio, Xiph AV1, Symphonia)
- ▶ Things we care about
 - ▶ **Low cognitive overhead**

Why Rust?

- ▶ Codec implementations in pure Rust (Rust Audio, Xiph AV1, Symphonia)
- ▶ Things we care about
 - ▶ **Low cognitive overhead**
 - ▶ Immutability

Why Rust?

- ▶ Codec implementations in pure Rust (Rust Audio, Xiph AV1, Symphonia)
- ▶ Things we care about
 - ▶ **Low cognitive overhead**
 - ▶ Immutability
 - ▶ Expressive type system

Why Rust?

- ▶ Codec implementations in pure Rust (Rust Audio, Xiph AV1, Symphonia)
- ▶ Things we care about
 - ▶ **Low cognitive overhead**
 - ▶ Immutability
 - ▶ Expressive type system
 - ▶ Memory safety and concurrency

Why Rust?

- ▶ Codec implementations in pure Rust (Rust Audio, Xiph AV1, Symphonia)
- ▶ Things we care about
 - ▶ **Low cognitive overhead**
 - ▶ Immutability
 - ▶ Expressive type system
 - ▶ Memory safety and concurrency
 - ▶ Foreign Function Interface

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject
 - ▶ GstObject

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject
 - ▶ GstObject
 - ▶ GstAllocator

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject
 - ▶ GObject
 - ▶ GstAllocator
 - ▶ GstBufferPool

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject
 - ▶ GObject
 - ▶ GstAllocator
 - ▶ GstBufferPool
 - ▶ GstBus

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject
 - ▶ GstObject
 - ▶ GstAllocator
 - ▶ GstBufferPool
 - ▶ GstBus
 - ▶ GstClock

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject
 - ▶ GObject
 - ▶ GstAllocator
 - ▶ GstBufferPool
 - ▶ GstBus
 - ▶ GstClock
 - ▶ GstDevice

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject
 - ▶ GObject
 - ▶ GstAllocator
 - ▶ GstBufferPool
 - ▶ GstBus
 - ▶ GstClock
 - ▶ GstDevice
 - ▶ GstDeviceMonitor

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject
 - ▶ GObject
 - ▶ GstAllocator
 - ▶ GstBufferPool
 - ▶ GstBus
 - ▶ GstClock
 - ▶ GstDevice
 - ▶ GstDeviceMonitor
 - ▶ GstDeviceProvider

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject
 - ▶ GObject
 - ▶ GstAllocator
 - ▶ GstBufferPool
 - ▶ GstBus
 - ▶ GstClock
 - ▶ GstDevice
 - ▶ GstDeviceMonitor
 - ▶ GstDeviceProvider
 - ▶ GstElement

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject
 - ▶ GObject
 - ▶ GstAllocator
 - ▶ GstBufferPool
 - ▶ GstBus
 - ▶ GstClock
 - ▶ GstDevice
 - ▶ GstDeviceMonitor
 - ▶ GstDeviceProvider
 - ▶ GstElement
 - ▶ GstPad

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject
 - ▶ GObject
 - ▶ GstAllocator
 - ▶ GstBufferPool
 - ▶ GstBus
 - ▶ GstClock
 - ▶ GstDevice
 - ▶ GstDeviceMonitor
 - ▶ GstDeviceProvider
 - ▶ GstElement
 - ▶ GstPad
 - ▶ ...

²GstObject

Why Rust?

- ▶ Bindings/abstractions over GLib/GObject²
- ▶ Provides a root for the object hierarchy tree filed in by the GStreamer library
- ▶ Gives basic reference counting, parenting functionality and locking.
- ▶ GObject
 - ▶ GObject
 - ▶ GstAllocator
 - ▶ GstBufferPool
 - ▶ GstBus
 - ▶ GstClock
 - ▶ GstDevice
 - ▶ GstDeviceMonitor
 - ▶ GstDeviceProvider
 - ▶ GstElement
 - ▶ GstPad
 - ▶ ...
- ▶ GStreamer bindings

²GstObject

Why immutability and types matter?

```
let caps: gst::Caps = gst::Caps::builder("video/x-raw")
    .field("width", crop_w)
    .field("height", crop_h)
    .field("pixel-aspect-ratio", gst::Fraction::new(1, 1))
    .build();
let s = caps.remove_structure(0);
```

Why immutability and types matter?

```
warning: unused variable: `s`
--> video-bin/src/imp.rs:152:13
|
152 |         let s = caps.remove_structure(0);
|           ^ help: if this is intentional, prefix it with an
|               underscore: `_s`
|
| = note: `#[warn(unused_variables)]` on by default
error[E0596]: cannot borrow data in dereference of `gstreamer::Caps`
as mutable
--> video-bin/src/imp.rs:152:17
|
152 |         let s = caps.remove_structure(0);
|           ~~~~~~ cannot borrow as mutable
|
| = help: trait `DerefMut` is required to modify through a dereference,
|         but it is not implemented for `gstreamer::Caps`
```

Why immutability and types matter?

```
let mut caps: gst::Caps = gst::Caps::builder("video/x-raw")
    .field("width", crop_w)
    .field("height", crop_h)
    .field("pixel-aspect-ratio", gst::Fraction::new(1, 1))
    .build();
let _s = caps.remove_structure(0);
```

Why immutability and types matter?

```
warning: variable does not need to be mutable
```

```
--> video-bin/src/imp.rs:147:13
```

```
147 |         let mut caps: gst::Caps = gst::Caps::builder("video/x-raw")
```

```
        ----^^^^
```

```
        |
        | help: remove this `mut`
```

```
= note: `#[warn(unused_mut)]` on by default
```

```
error[E0596]: cannot borrow data in dereference of `gststreamer::Caps`
  as mutable
```

```
--> video-bin/src/imp.rs:152:18
```

```
152 |         let _s = caps.remove_structure(0);
```

```
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ cannot borrow as mutable
```

```
= help: trait `DerefMut` is required to modify through a dereference,
but it is not implemented for `gststreamer::Caps`
```

Why immutability and types matter?

```
let caps: gst::Caps = gst::Caps::builder("video/x-raw")
    .field("width", crop_w)
    .field("height", crop_h)
    .field("pixel-aspect-ratio", gst::Fraction::new(1, 1))
    .build();
let caps = caps.get_mut().unwrap();
let _s = caps.remove_structure(0);
```

Why immutability and types matter?

```
error[E0596]: cannot borrow `caps` as mutable, as it is not declared
             as mutable
   --> video-bin/src/imp.rs:152:20
      |
147  |         let caps: gst::Caps = gst::Caps::builder("video/x-raw")
      |         ---- help: consider changing this to be mutable:
           `mut caps`
...
152  |         let caps = caps.get_mut().unwrap();
      |         ^^^^^^^^^^^^^^^^^^ cannot borrow as mutable
```

For more information about this error, try `rustc --explain E0596`.

Why immutability and types matter?

```
let mut caps: gst::Caps = gst::Caps::builder("video/x-raw")
    .field("width", crop_w)
    .field("height", crop_h)
    .field("pixel-aspect-ratio", gst::Fraction::new(1, 1))
    .build();
if let Some(caps) = caps.get_mut() {
    let _s = caps.remove_structure(0);
}
```

GStreamer & Rust

- ▶ Some stats³

³GStreamer & Rust: What has happened over the last 5 years

GStreamer & Rust

- ▶ Some stats³
 - ▶ gstreamer-rs: ~2700 commits, gst-plugins-rs: ~2600 commits

³GStreamer & Rust: What has happened over the last 5 years

GStreamer & Rust

- ▶ Some stats³
 - ▶ gstreamer-rs: ~2700 commits, gst-plugins-rs: ~2600 commits
 - ▶ gstreamer-rs: ~85 contributors, gst-plugins-rs: ~110 contributors

³GStreamer & Rust: What has happened over the last 5 years

GStreamer & Rust

- ▶ Some stats³
 - ▶ gstreamer-rs: ~2700 commits, gst-plugins-rs: ~2600 commits
 - ▶ gstreamer-rs: ~85 contributors, gst-plugins-rs: ~110 contributors
 - ▶ gst-plugins-rs: ~ +180k SLOC / -37k SLOC

³GStreamer & Rust: What has happened over the last 5 years

GStreamer & Rust

- ▶ Some stats³
 - ▶ gstreamer-rs: ~2700 commits, gst-plugins-rs: ~2600 commits
 - ▶ gstreamer-rs: ~85 contributors, gst-plugins-rs: ~110 contributors
 - ▶ gst-plugins-rs: ~ +180k SLOC / -37k SLOC
 - ▶ gst-plugins-rs: Overall 47 plugins, 149 elements

³GStreamer & Rust: What has happened over the last 5 years

GStreamer & Rust

- ▶ Some stats³
 - ▶ gstreamer-rs: ~2700 commits, gst-plugins-rs: ~2600 commits
 - ▶ gstreamer-rs: ~85 contributors, gst-plugins-rs: ~110 contributors
 - ▶ gst-plugins-rs: ~ +180k SLOC / -37k SLOC
 - ▶ gst-plugins-rs: Overall 47 plugins, 149 elements
- ▶ In relation to the GStreamer monorepo

³GStreamer & Rust: What has happened over the last 5 years

GStreamer & Rust

- ▶ Some stats³
 - ▶ gstreamer-rs: ~2700 commits, gst-plugins-rs: ~2600 commits
 - ▶ gstreamer-rs: ~85 contributors, gst-plugins-rs: ~110 contributors
 - ▶ gst-plugins-rs: ~ +180k SLOC / -37k SLOC
 - ▶ gst-plugins-rs: Overall 47 plugins, 149 elements
- ▶ In relation to the GStreamer monorepo
 - ▶ 1.22 cycle: ~33% commits / MRs in Rust modules

³GStreamer & Rust: What has happened over the last 5 years

GStreamer & Rust

- ▶ Some stats³
 - ▶ gstreamer-rs: ~2700 commits, gst-plugins-rs: ~2600 commits
 - ▶ gstreamer-rs: ~85 contributors, gst-plugins-rs: ~110 contributors
 - ▶ gst-plugins-rs: ~ +180k SLOC / -37k SLOC
 - ▶ gst-plugins-rs: Overall 47 plugins, 149 elements
- ▶ In relation to the GStreamer monorepo
 - ▶ 1.22 cycle: ~33% commits / MRs in Rust modules
 - ▶ 1.24 cycle: ~25% commits / MRs in Rust modules

³GStreamer & Rust: What has happened over the last 5 years

Resources

- ▶ [GObject subclassing in Rust](#)

Resources

- ▶ [GObject subclassing in Rust](#)
- ▶ [GStreamer bindings for Rust](#)

Resources

- ▶ [GObject subclassing in Rust](#)
- ▶ [GStreamer bindings for Rust](#)
- ▶ [Rust GStreamer Plugins](#)

Resources

- ▶ [GObject subclassing in Rust](#)
- ▶ [GStreamer bindings for Rust](#)
- ▶ [Rust GStreamer Plugins](#)
- ▶ [Using GStreamer](#)

Resources

- ▶ [GObject subclassing in Rust](#)
- ▶ [GStreamer bindings for Rust](#)
- ▶ [Rust GStreamer Plugins](#)
- ▶ [Using GStreamer](#)
- ▶ [How to get started with GStreamer](#)

Resources

- ▶ [GObject subclassing in Rust](#)
- ▶ [GStreamer bindings for Rust](#)
- ▶ [Rust GStreamer Plugins](#)
- ▶ [Using GStreamer](#)
- ▶ [How to get started with GStreamer](#)
- ▶ [GStreamer for your backend services](#)

Questions?

- ▶ Rust Bangalore

Questions?

- ▶ Rust Bangalore
 - ▶ Meetup: <https://hasgeek.com/rustbangalore>

Questions?

- ▶ Rust Bangalore
 - ▶ Meetup: <https://hasgeek.com/rustbangalore>
 - ▶ Telegram: t.me/RustIndia

Questions?

- ▶ Rust Bangalore
 - ▶ Meetup: <https://hasgeek.com/rustbangalore>
 - ▶ Telegram: t.me/RustIndia
- ▶ Reach out on

Questions?

- ▶ Rust Bangalore
 - ▶ Meetup: <https://hasgeek.com/rustbangalore>
 - ▶ Telegram: t.me/RustIndia
- ▶ Reach out on
 - ▶ email:
 - me@sanchayanmaity.net
 - sanchayan@asymptotic.io
 - hello@asymptotic.io

Questions?

- ▶ Rust Bangalore
 - ▶ Meetup: <https://hasgeek.com/rustbangalore>
 - ▶ Telegram: t.me/RustIndia
- ▶ Reach out on
 - ▶ email:
 - me@sanchayanmaity.net
 - sanchayan@asymptotic.io
 - hello@asymptotic.io
 - ▶ Mastodon: sanchayanmaity.com

Questions?

- ▶ Rust Bangalore
 - ▶ Meetup: <https://hasgeek.com/rustbangalore>
 - ▶ Telegram: t.me/RustIndia
- ▶ Reach out on
 - ▶ email:
 - me@sanchayanmaity.net
 - sanchayan@asymptotic.io
 - hello@asymptotic.io
 - ▶ Mastodon: sanchayanmaity.com
 - ▶ Blog: sanchayanmaity.net