

On using intersink/src for live pipeline manipulation

Juan Adarve



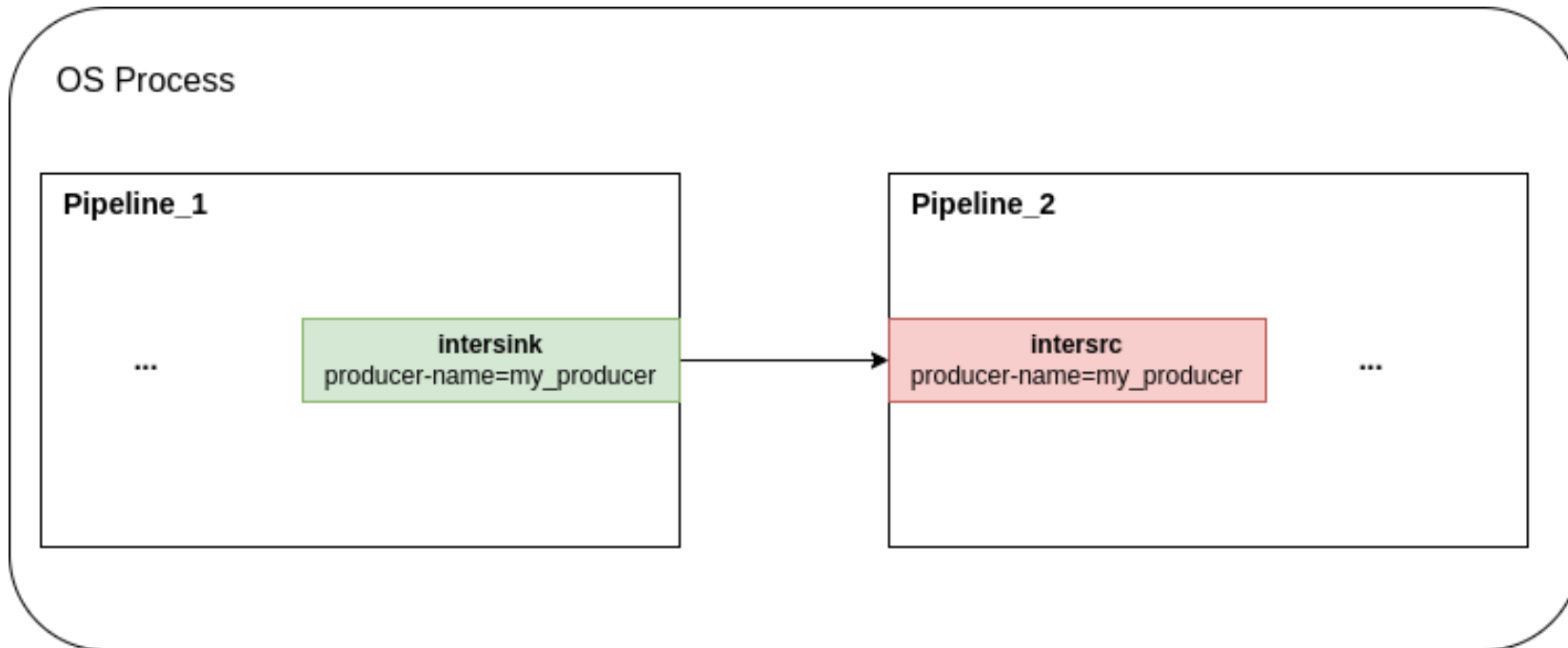
The Problem

Complex GStreamer pipelines taking several responsibilities at once

- Ingesting input stream.
 - Processing (transcoding).
 - Muxing
 - Output
- A failure in any such stages propagates to all others.
 - **Service interruption.**
 - **Ideally**, we should only action on the failing parts of the pipeline, while others might still work.

Intersink/intersrc elements

- Part for the **rsinter** plugin.
- Can connect several GST pipelines running in the same Process.
- Must share a common GstClock and Basetime to run correctly.



Gst-launch

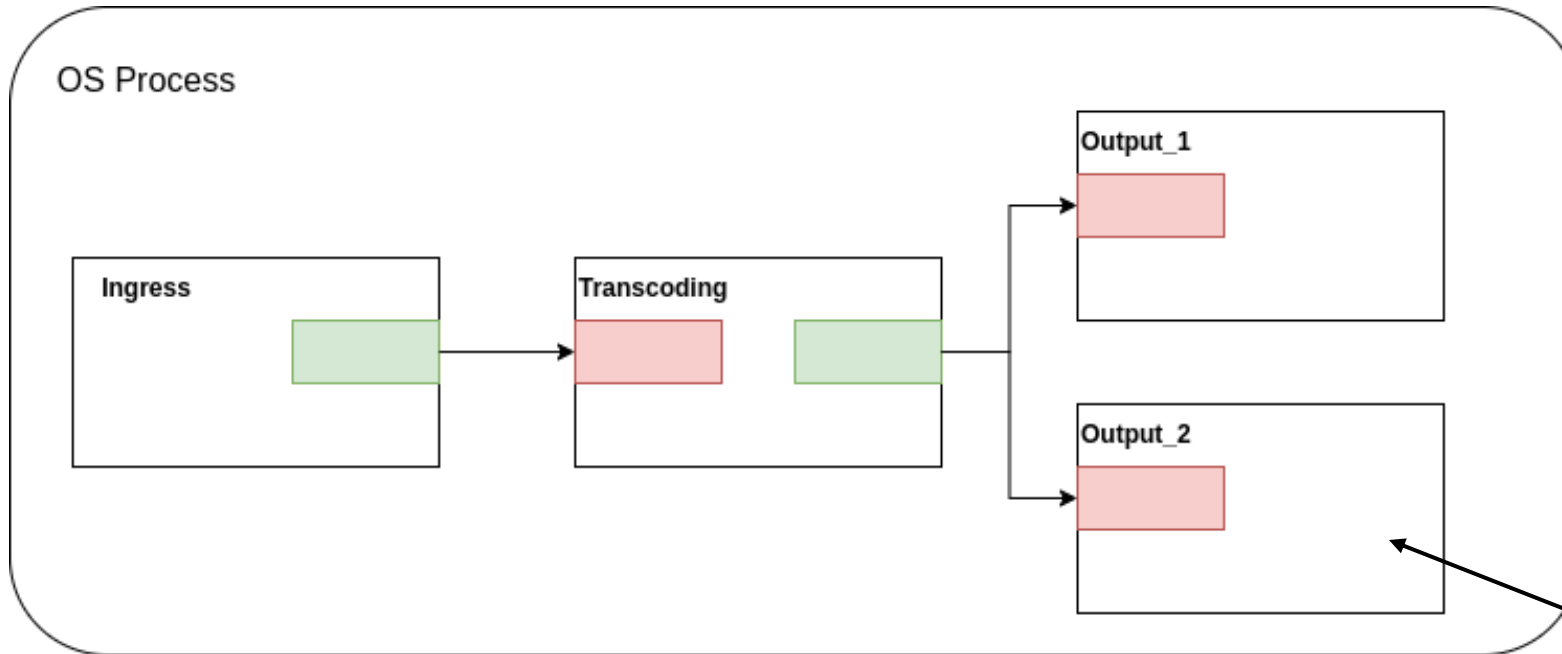
```
cargo run --bin use_intersink -- \
```

```
--pipeline --name testsrc videotestsrc ! video/x-raw ! tee  
name=video_tee ! queue ! videoconvert ! autovideosink video_tee. !  
queue ! intersink name=testsrc_raw_video \
```

```
--pipeline --name display_video intersrc name=testsrc_raw_video !  
queue ! videoconvert ! autovideosink
```

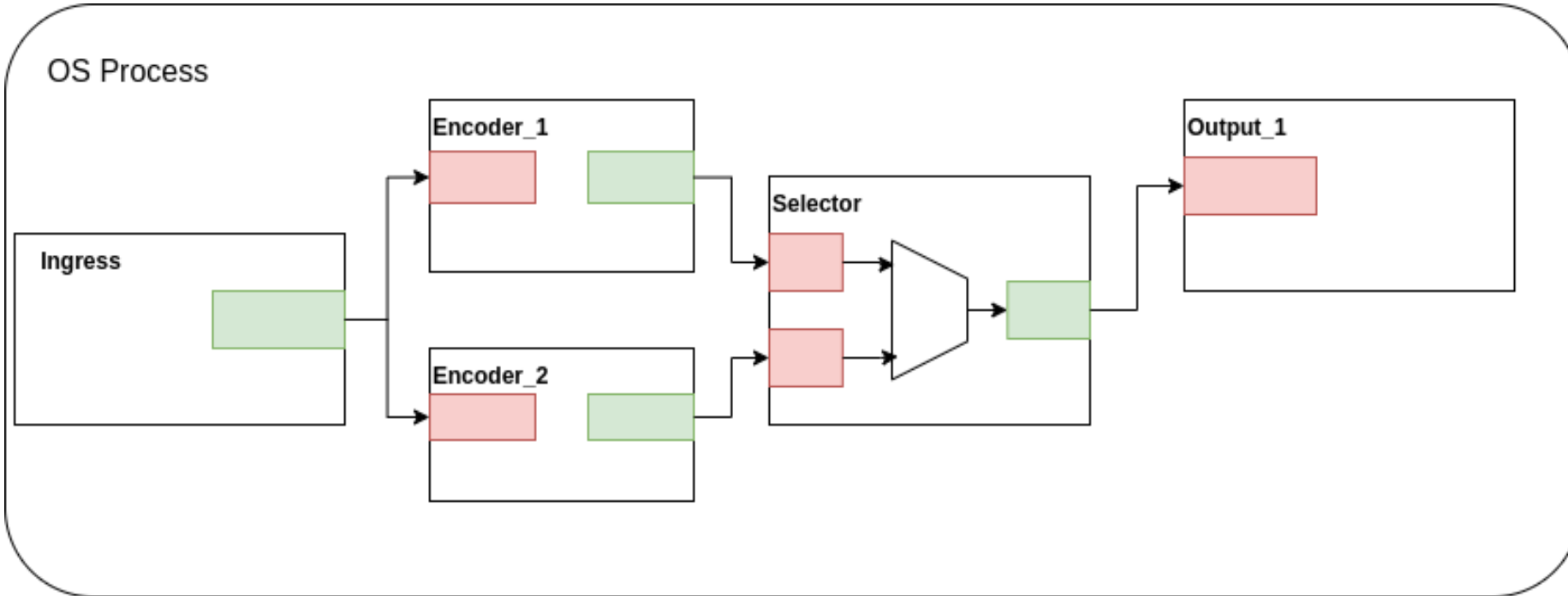
Multi-pipeline design

Each pipeline has its own life-cycle



- Add
- Stop
- Restart
- Remove

Encoder hot-swapping



- **Add** encoder_2
- **Switch** to encoder_2
- **Stop** encoder_1
- **Remove** encoder_1

Thanks!