

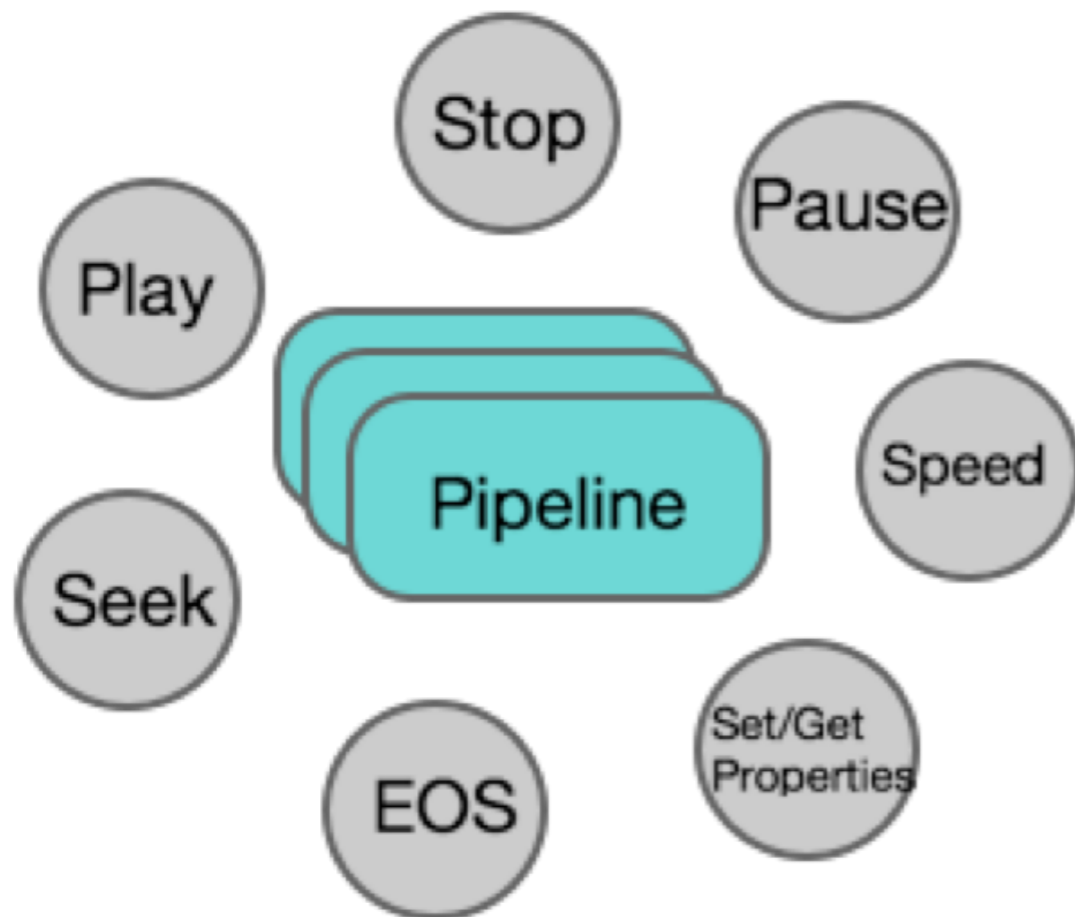
GST Daemon and Interpipes:

A simpler way to get your application done

Melissa Montero Bonilla
melissa.montero@ridgerun.com



Why GStreamer Daemon (GSTD)?



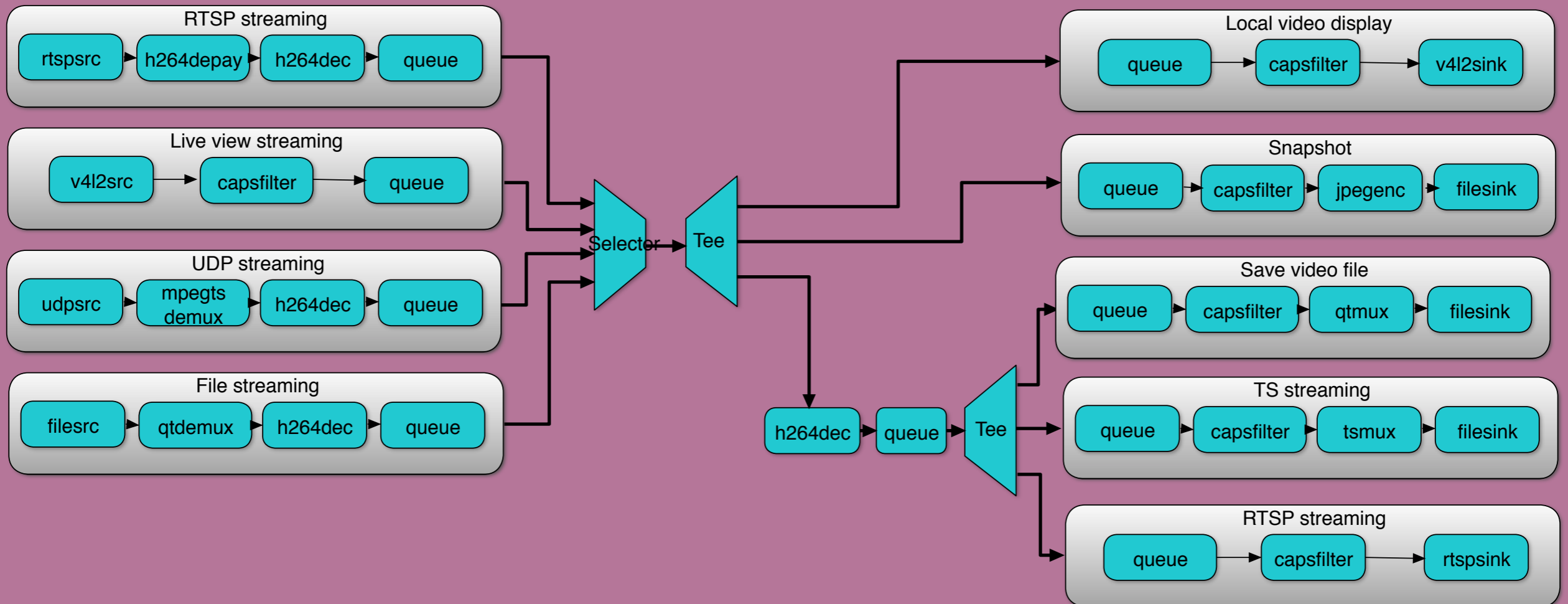
- **Custom applications require common actions.**
- **Replicated code.**

What is GStreamer Daemon (GSTD)?

- **Multi-threaded application to control GStreamer pipelines.**
- **A layer over GStreamer that simplifies user applications development.**
- **Version for gst-1.0 under development, beta version available.**

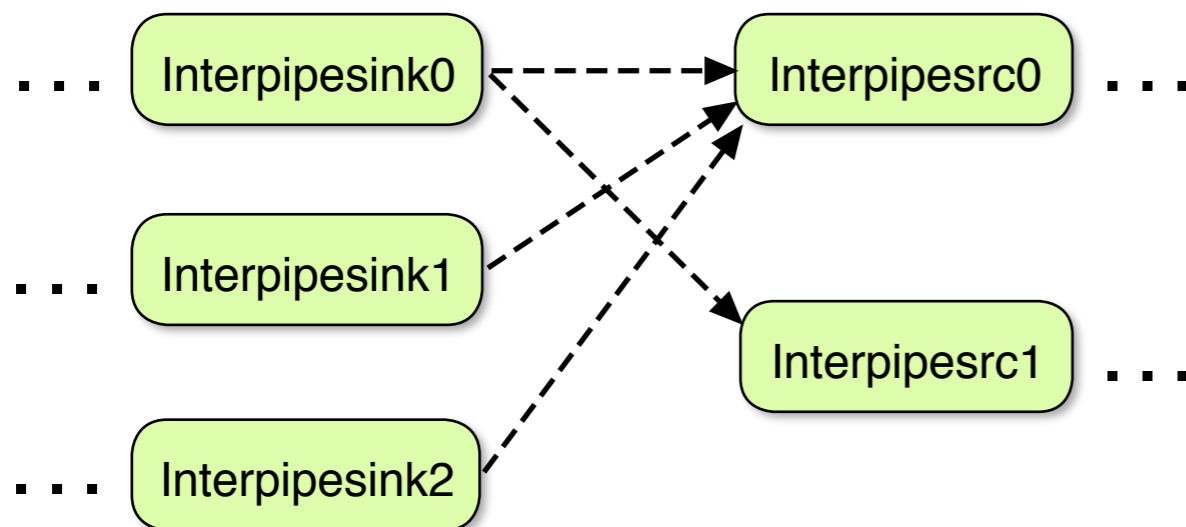
GSTD Features

- **Written in C**
- **May be used as:**
 - **Library linked to application**
 - **Standalone server with multiple IPCs (TCP, DBus ...)**
- **Dynamically create and destroy multiple pipelines.**
- **Play, stop, pause, send eos on specific pipeline.**
- **Seek, change speed, get position, get duration.**
- **Set/Get element properties**
- **Cmdline client to easily test pipelines at early development.**

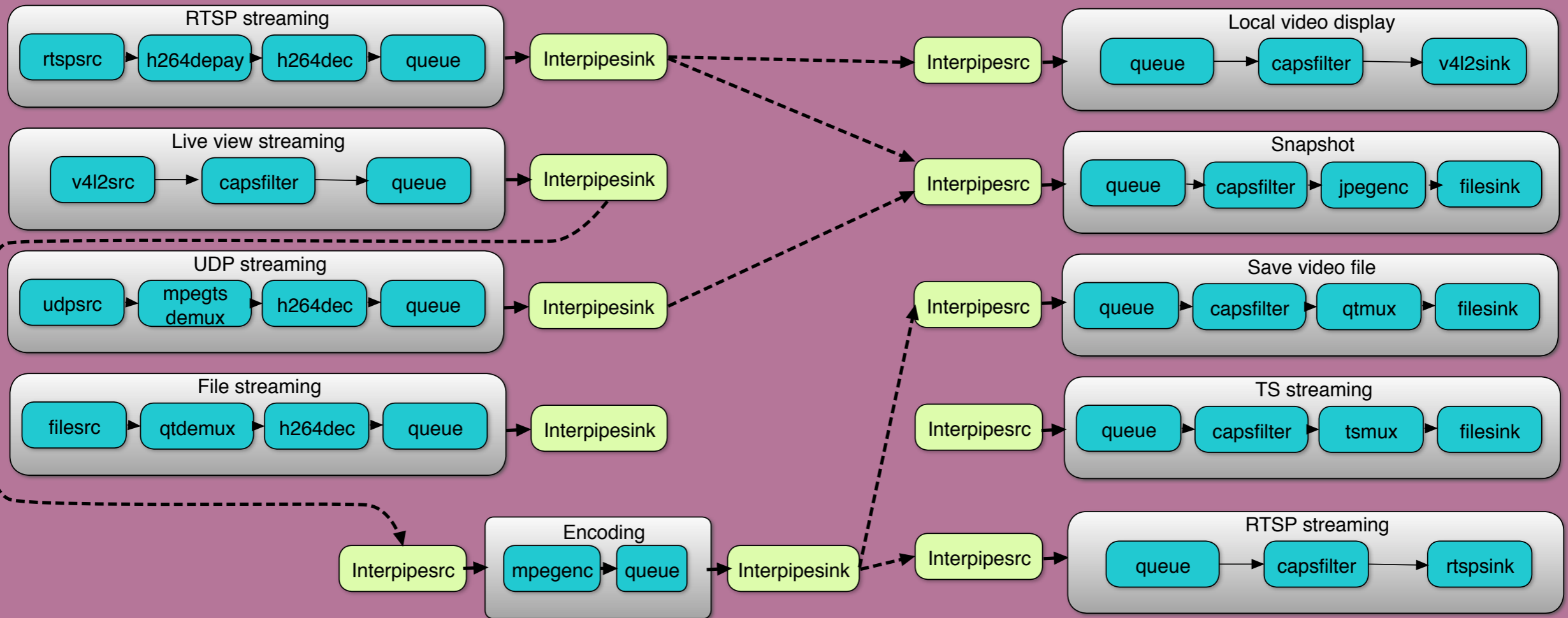


What about complex pipelines?

Interpipes



- Inspired by `intervideosrc/intervideosink`
- Uses `appsrc` and `appsink` as a base
- Generic data handling
- Multiple dynamic `interpipesink - interpipesrc` connections
- Replace tee and selector elements allowing dynamic pipeline flow control with GStreamer Daemon.



Using interpipes

Check out the projects

Interpipes

<https://github.com/RidgeRun/gst-interpipe-1.0>

Gstreamer Daemon (GSTD) 0.10

<https://github.com/RidgeRun/gstd>

Gstreamer Daemon (GSTD) 1.0

<https://github.com/RidgeRun/gstd-1.x>

Thank You

melissa.montero@ridgerun.com

