

A *gstreamer* backend for chromium

Anton Obzhirov / Julien Isorce

Repository:

github.com/Samsung/ChromiumGStreamerBackend

Authors

Samsung Research UK



- Anton Obzhairov (a.obzhairov@samsung.com, irc: a.obzhairov)
 - Senior Software Engineer, MS degree from FEFU, Russia
 - WebKit committer. One of the TyGL authors.
- Julien Isorce (j.isorce@samsung.com, irc: capOM)
 - Senior Software Engineer, MS degree from ENSEEIHT, France
 - Used to maintain GstGL-0.10. Browser media RPi

More: <http://blogs.s-osg.org/announcing-a-new-gstreamer-backend-for-chromium>

Objectives



- GStreamer - `<video>` tag
- Generic and efficient
- Opensource

Agenda



1. Introduction to GStreamer backend

1. Chromium architecture
2. Existing media backends
3. GStreamer backend

DEMO 1

2. Advanced features

1. Media Source Extension (MSE)
2. Encrypted Media Extension (EME)
3. Zero-copy integration

DEMO 2

3. Roadmap

Chromium architecture

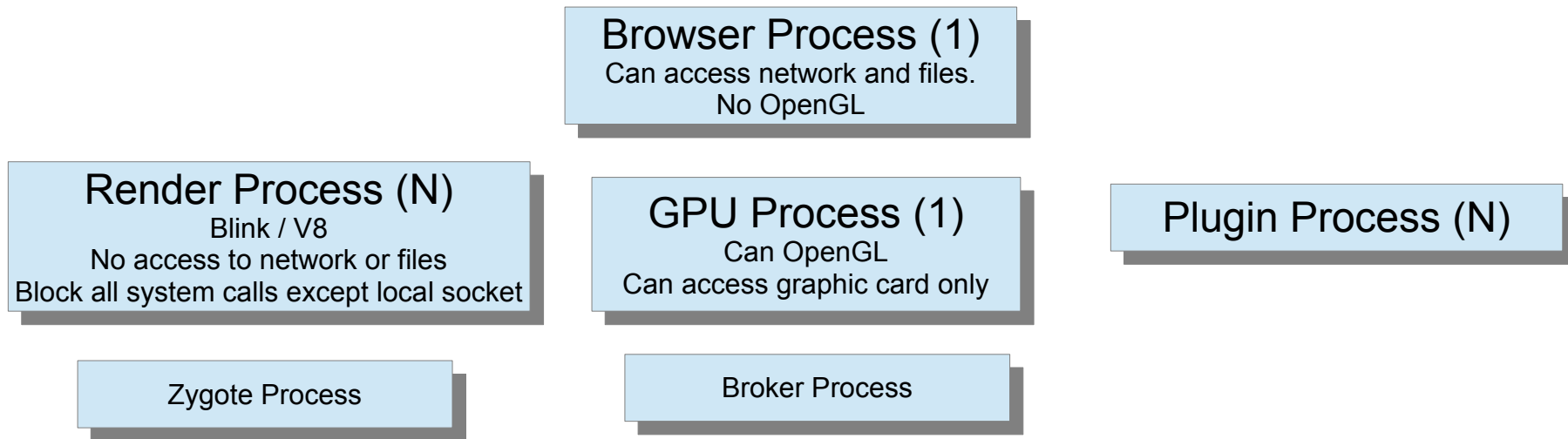
“The credo”



- *“It's nearly impossible to build a browser that never crashes or hangs.”* (chromium.org)
- *“For a codebase as large and diverse as Chromium, reasoning about the combined behavior of all its parts is nearly impossible. Cannot be perfectly secure”* (chromium.org)
- +189 external dependencies in chromium/src/third_party/
- Solution: split the browser into multiple processes
 - For security and stability

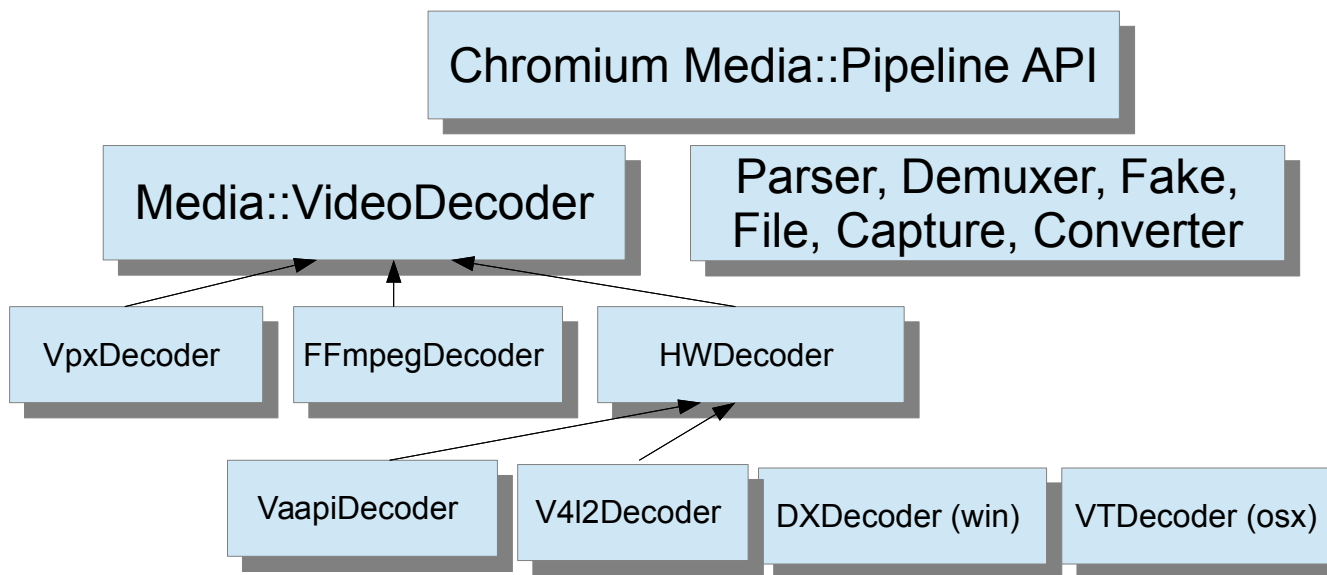
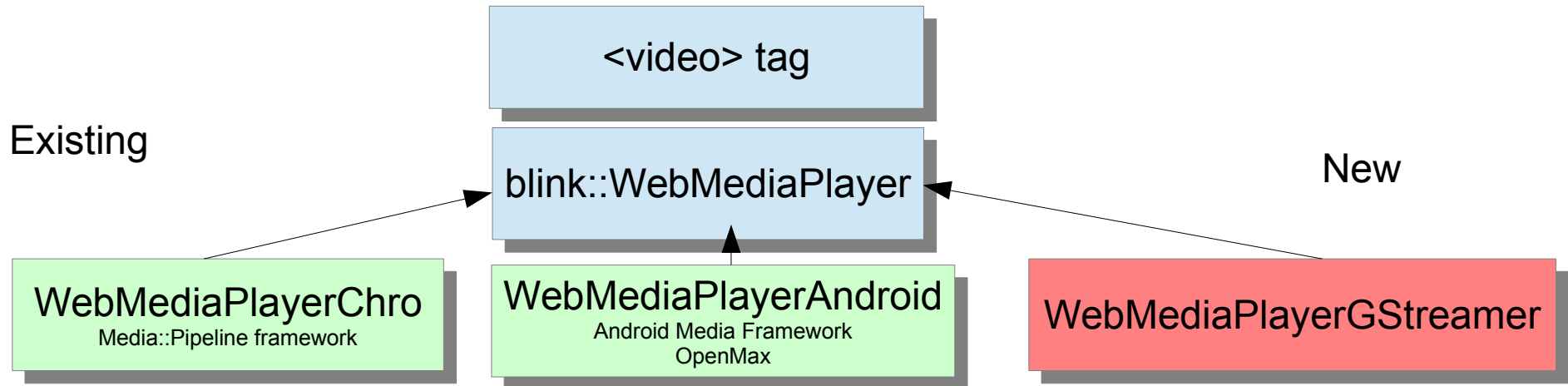
Chromium architecture

Many Processes



- Linux kernel security feature:
 - Seccomp (Secure Computing Mode) (can only call exit)
 - BPF extension: allow / denied particular system calls (read, write, socket, unlink)
- chroot Render Process to empty directory
- New PID namespace for Render Process (kill)

Existing media backends



- 2700 files / 170 000 lines of code
- Only FFmpegDecoder on Linux desktop. (ChromeOS for others)

GStreamer backend Requirements



- Respect sandbox rules
- No video hole
- Not Platform specific (`#if defined(ANDROID)`)
- Handle protected content
- Zero-copy

GStreamer backend Problems



- **Render Process: no.**
 - Problem: cannot load gst plugins. Cannot access HW decoders.
 - Static ? → break rules, duplicate, space, maintenance
- **Browser Process: no.**
 - Problem: no process isolation to handle stability and sandbox.
 - Like Android ? not a long term solution
- **GPU Process: no**
 - Problem: only have access to graphic resources.
 - GStreamer crash would break the whole browser rendering.

GStreamer backend

A new Process

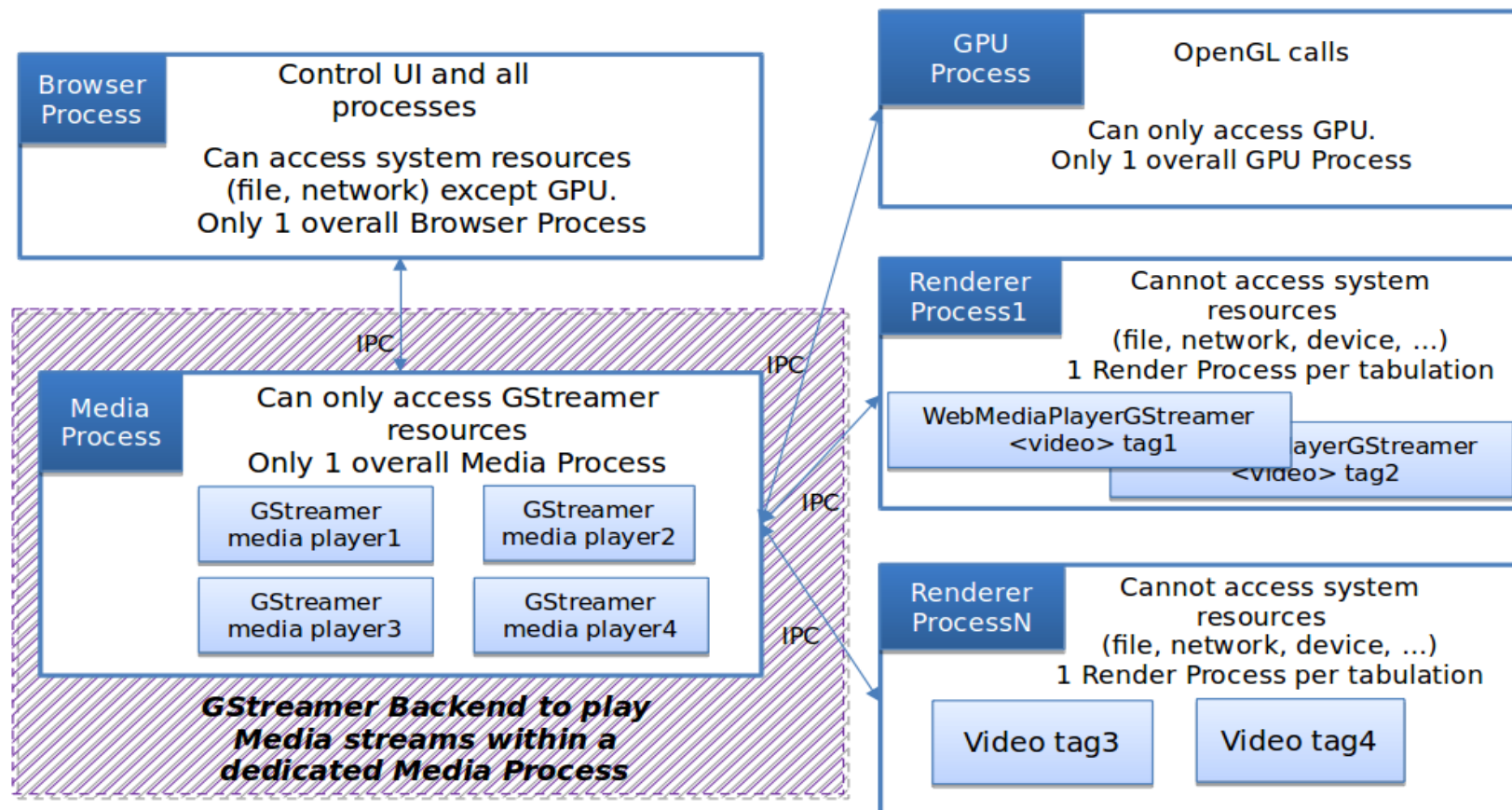


- Sandbox to filter system calls
- No direct access to network and GL
- Policy to load gst plugins
- Zero-copy

GStreamer backend Media Process

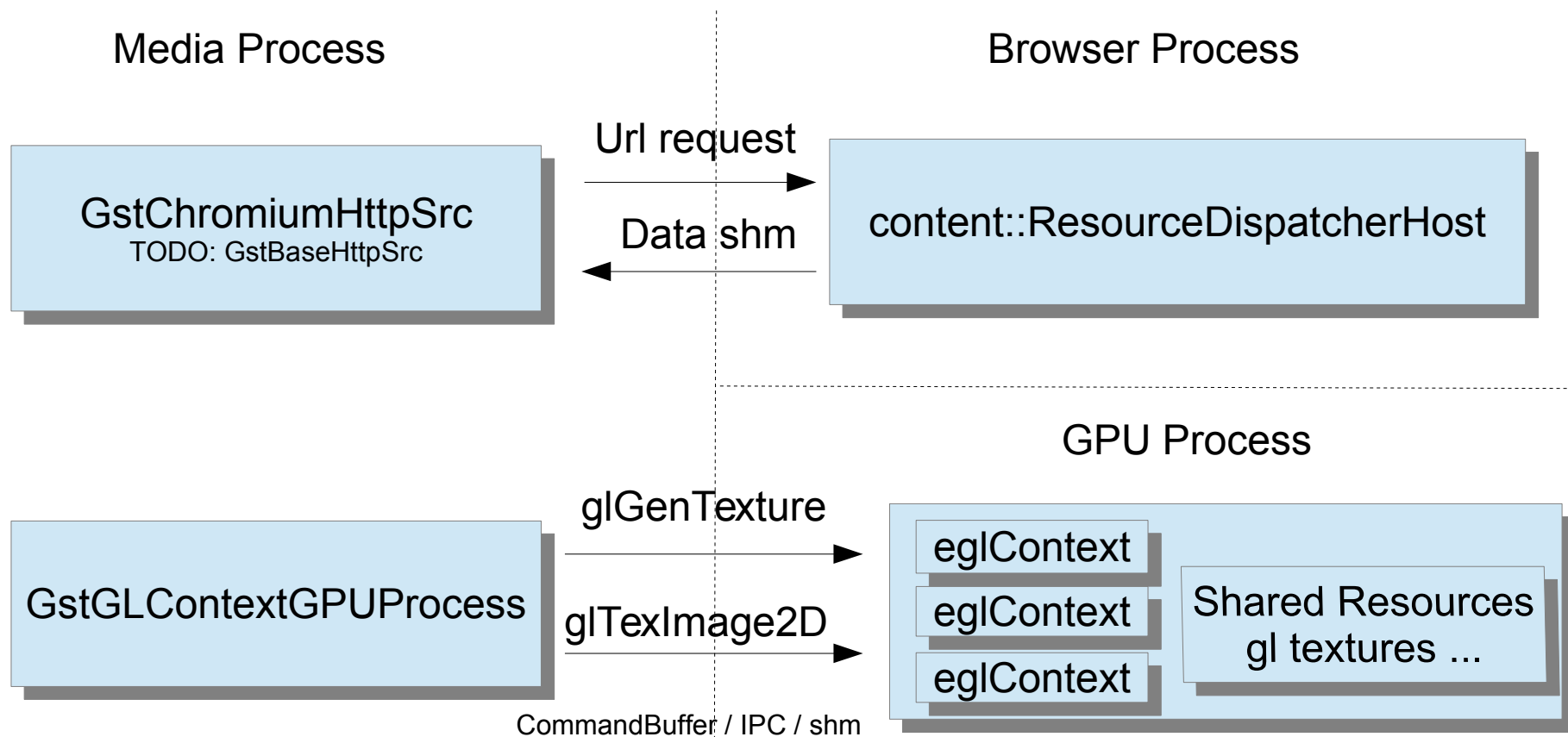


GStreamer pipeline ↔ <video> tag



GStreamer backend

Built-in source and sink



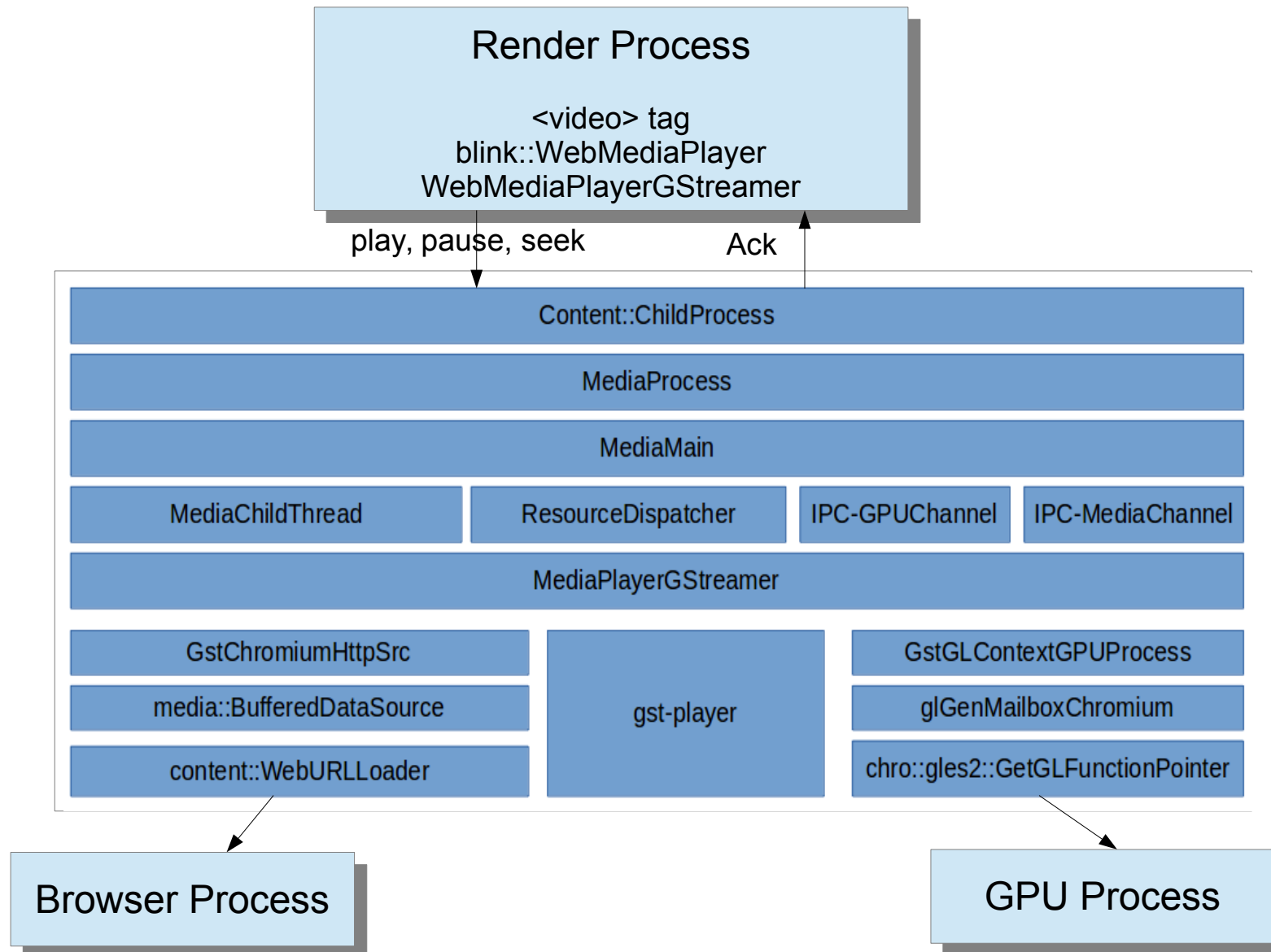
Client: egl/gles2

~~Load libGLSv2.so, load libEGL.so~~

`chromium::gles2::GetGLFunctionPointer` → `GstGLVtable`

Server: egl/gles2, glx/gl, wgl,gl

GStreamer backend Stack

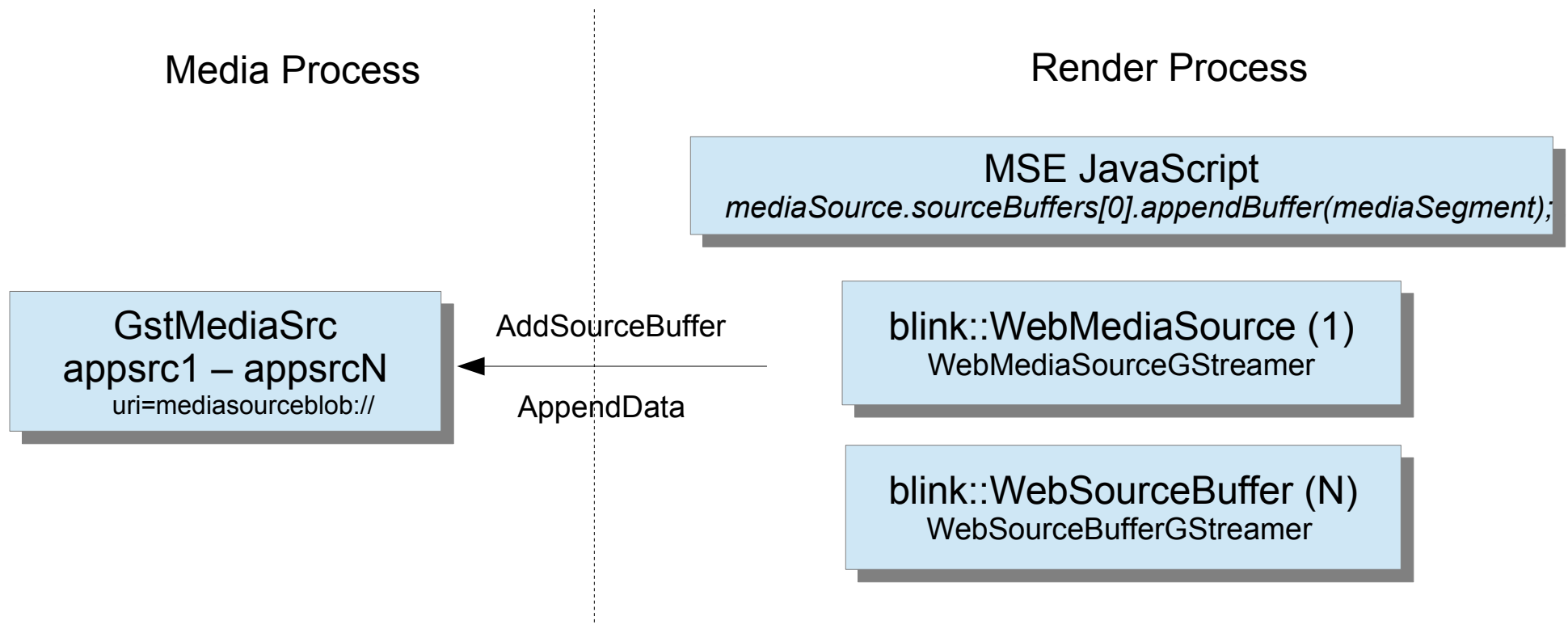


Progressive/adaptive streaming and sandbox

Media Source Extension



- W3C HTML5 extension for <video> tag
- YouTube / Netflix

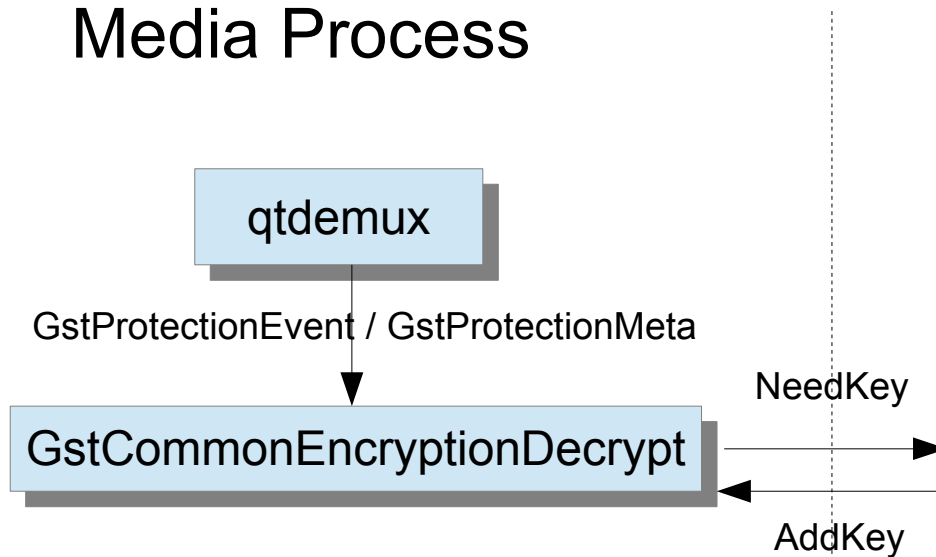


Encrypted Media Extension

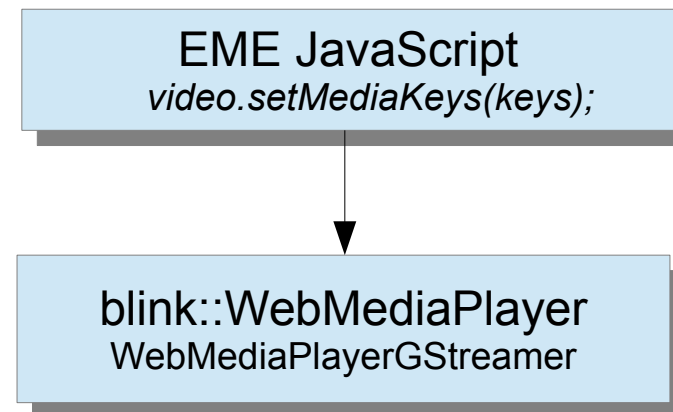


- W3C HTML5 extension for <video> tag (~~flash<object>~~)
- Protected content - DRM (Digital Rights Management)

Media Process



Render Process

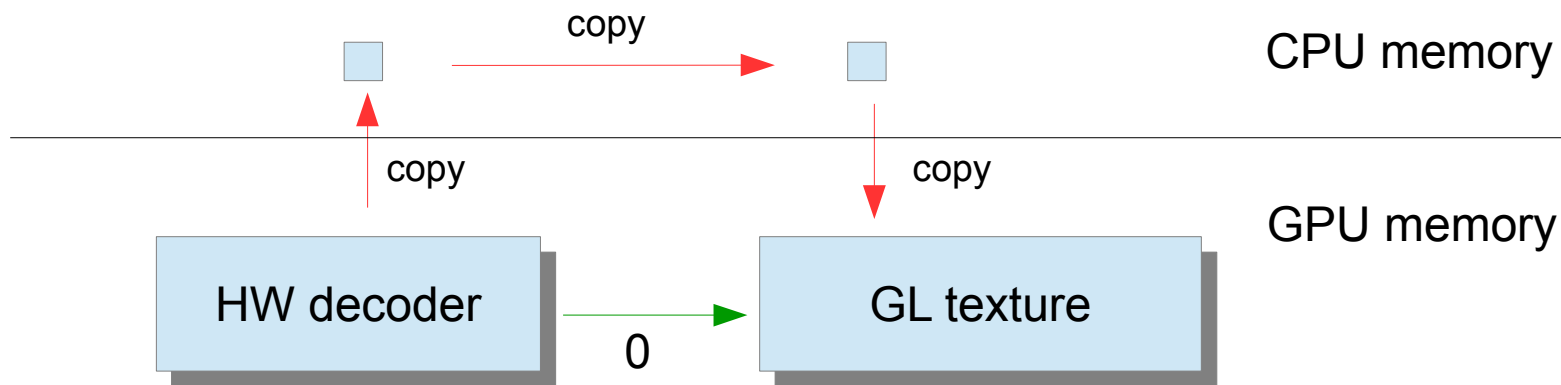


NeedKey

AddKey

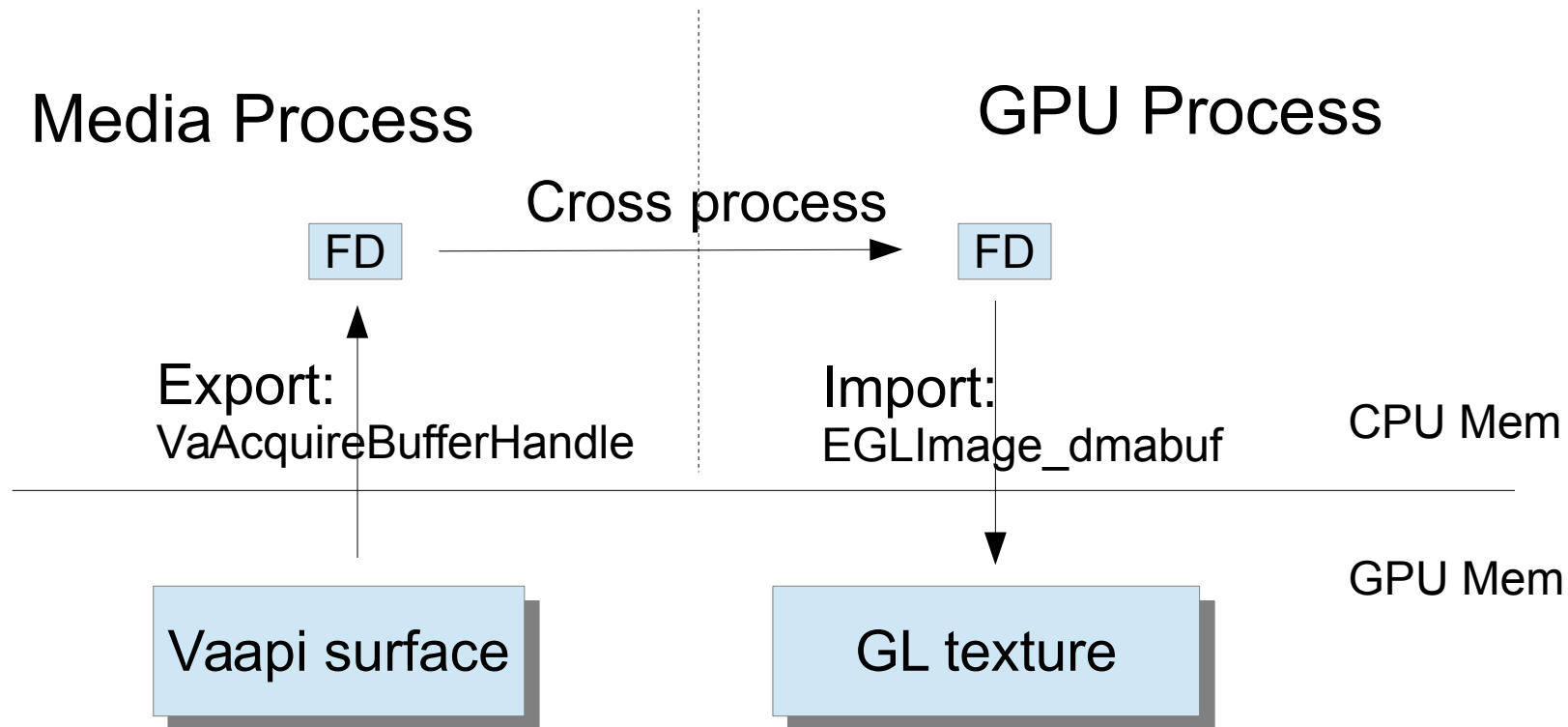
Zero copy

What is it ?



- Need interoperability [VA API / VDPAU / v4l2 / OpenMAX] ↔ GL
- Need cross processes
- Need secure

Zero copy Dma-buf



Zero copy Opensource driver



libdrm
Direct Rendering Manager

libdrm_nouveau
struct nouveau_bo

Mesa / Gallium

PIPE_VIDEO
struct pipe_surface
struct pipe_resource

GL

VA API

VDPAU

OpenMAX

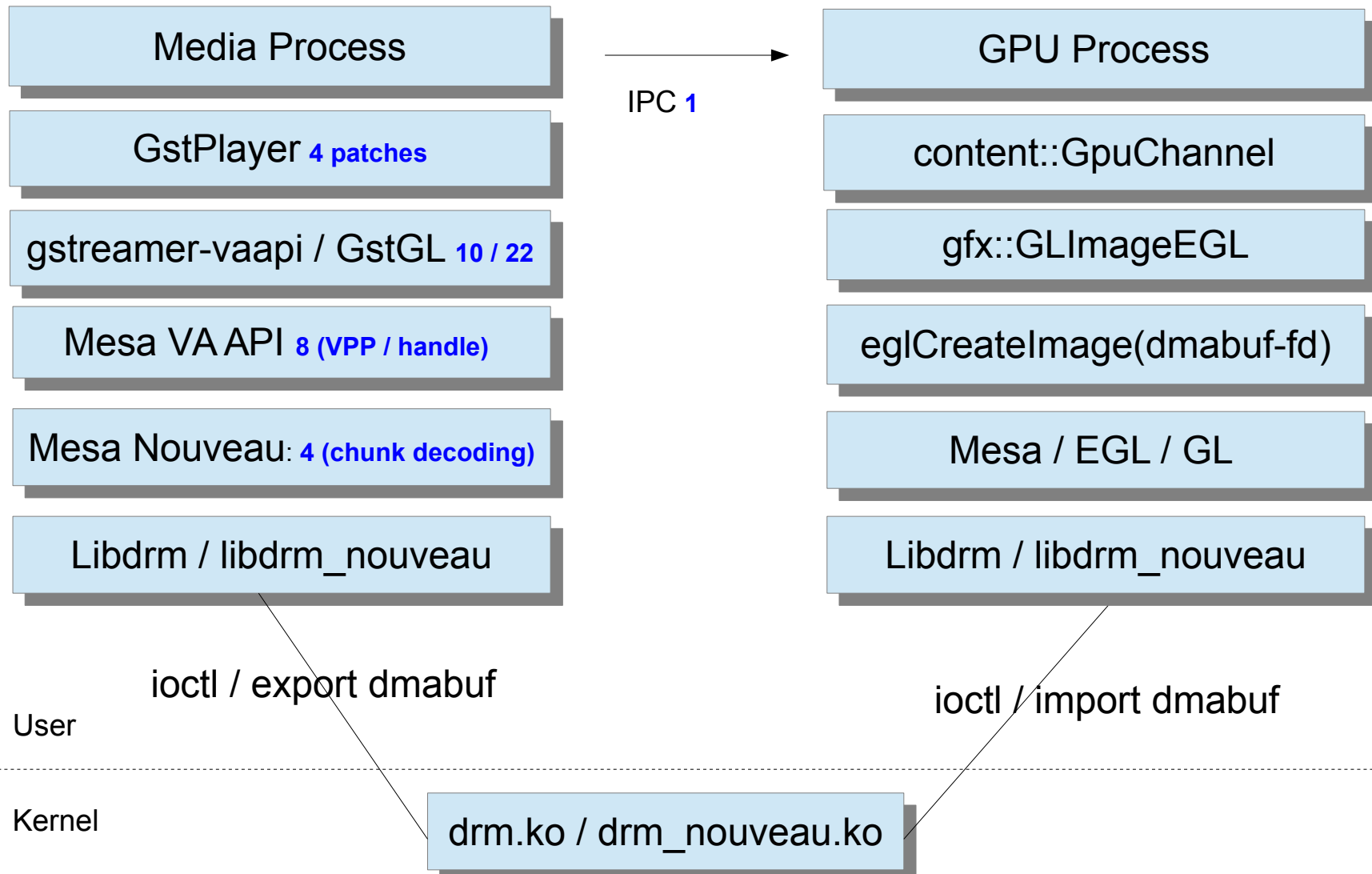
OpenCL

GStreamer

glimagesink

gstvaapidecode

Zero copy Chromium



MSE / EME / Zero copy

Summary

Current supported features



- Progressive streaming (http)
- Adaptive streaming (hls, dash)
- Media Source Extension (YouTube)
- Encrypted Media Extension (Protected content)
- Zero copy (dmabuf / EGLImage / cross process)

Roadmap

We need your contribution



- Player interface in Media Process
- GstMultiAppSrc + seeking
- GstBaseHttpSrc
- gst-omx on desktop
- WebRTC
- WebM encryption
- OP-TEE – secure dmabuf
- Pulseaudio crash with sandbox
- YouTube conformance tests

Thank you. Question ?

Repository:

github.com/Samsung/ChromiumGStreamerBackend

A decorative graphic at the bottom of the slide consisting of several overlapping, translucent blue wavy lines that create a sense of motion and depth.