

MAXIMIZING THE PERFORMANCE OF HTML5 VIDEO IN RPI2

Gwang Yoon Hwang

yoon@igalia.com



WHAT IS THE HTML5 VIDEO?

```
<video width="1280" height="720" autoplay loop>
  <source src="big_buck_bunny.mp4" type="video/mp4" />
</video>
```

WEBKIT - A PORTABLE WEB RENDERING ENGINE

- PCs, phones, TVs, IVI, smartwatches, **Raspberry Pi**
- Mac, iOS, EFL, **GTK**

WEBKITGTK+

- Full-featured port of the WebKit rendering engine
- Useful in a wide rage of systems from desktop to embedded

WEBKITFORWAYLAND

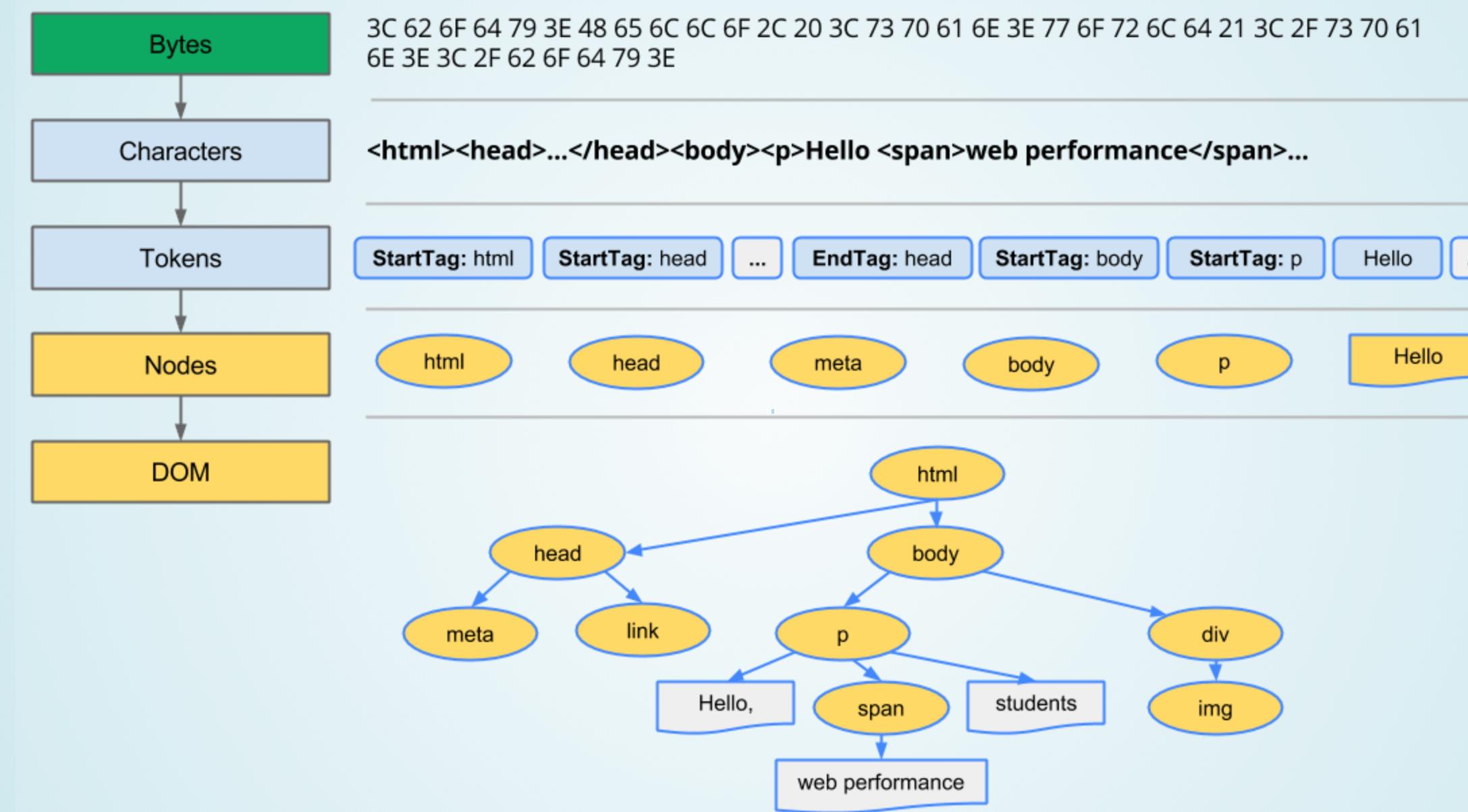
- Avoids using any toolkit
- Defaults to EGL, OpenGL ES for accelerated rendering of Web content
- Optimized for displaying the fullscreen web content -- Youtube TV, Information displays, IVI, STBs..

HOW WEBKIT RENDERS WEBPAGE

STEPS FOR RENDERING

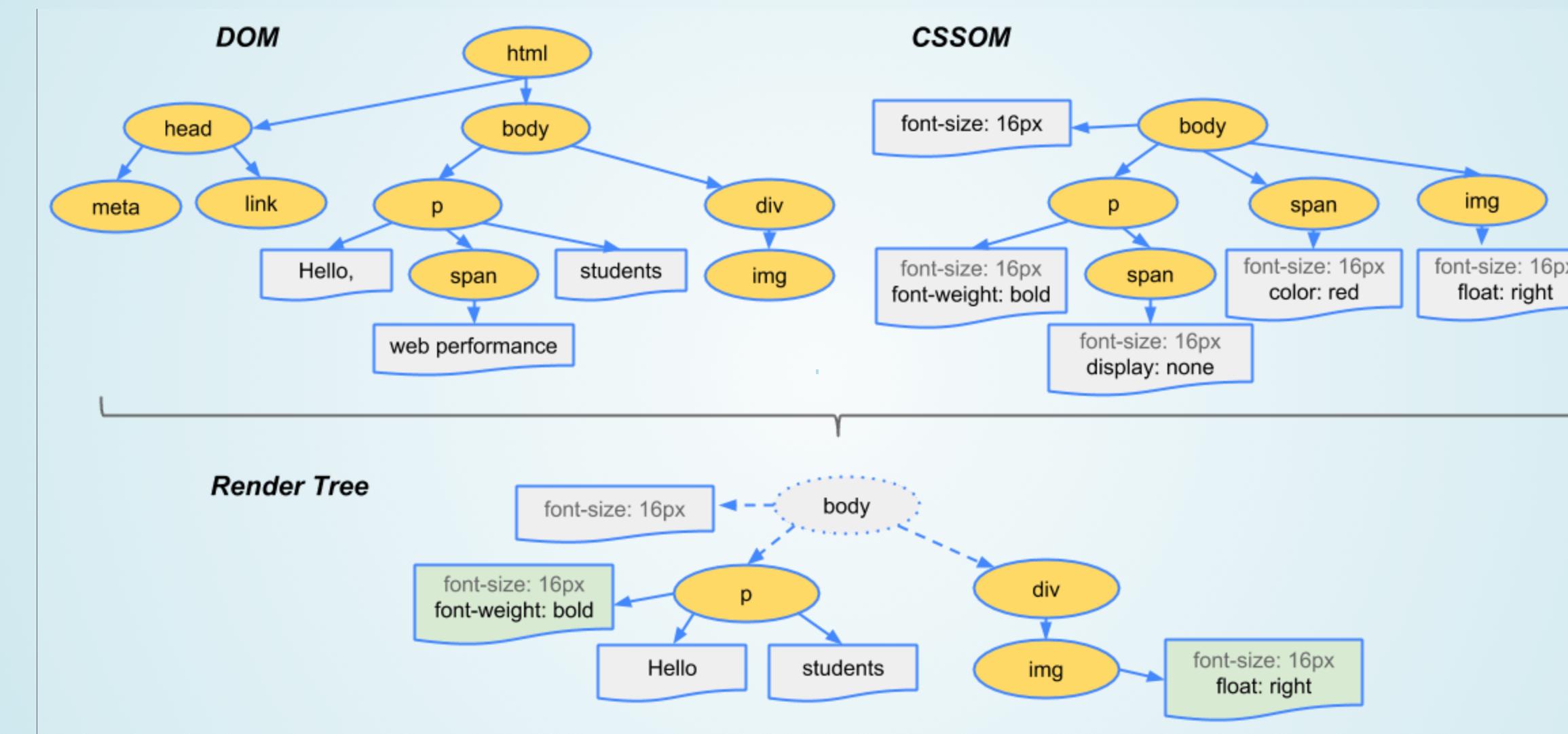
- Parsing: Nodes to DOM Tree
- Constructing RenderObject Tree to RenderLayer Tree

PARSING: CREATES THE DOM TREE FROM SOURCE



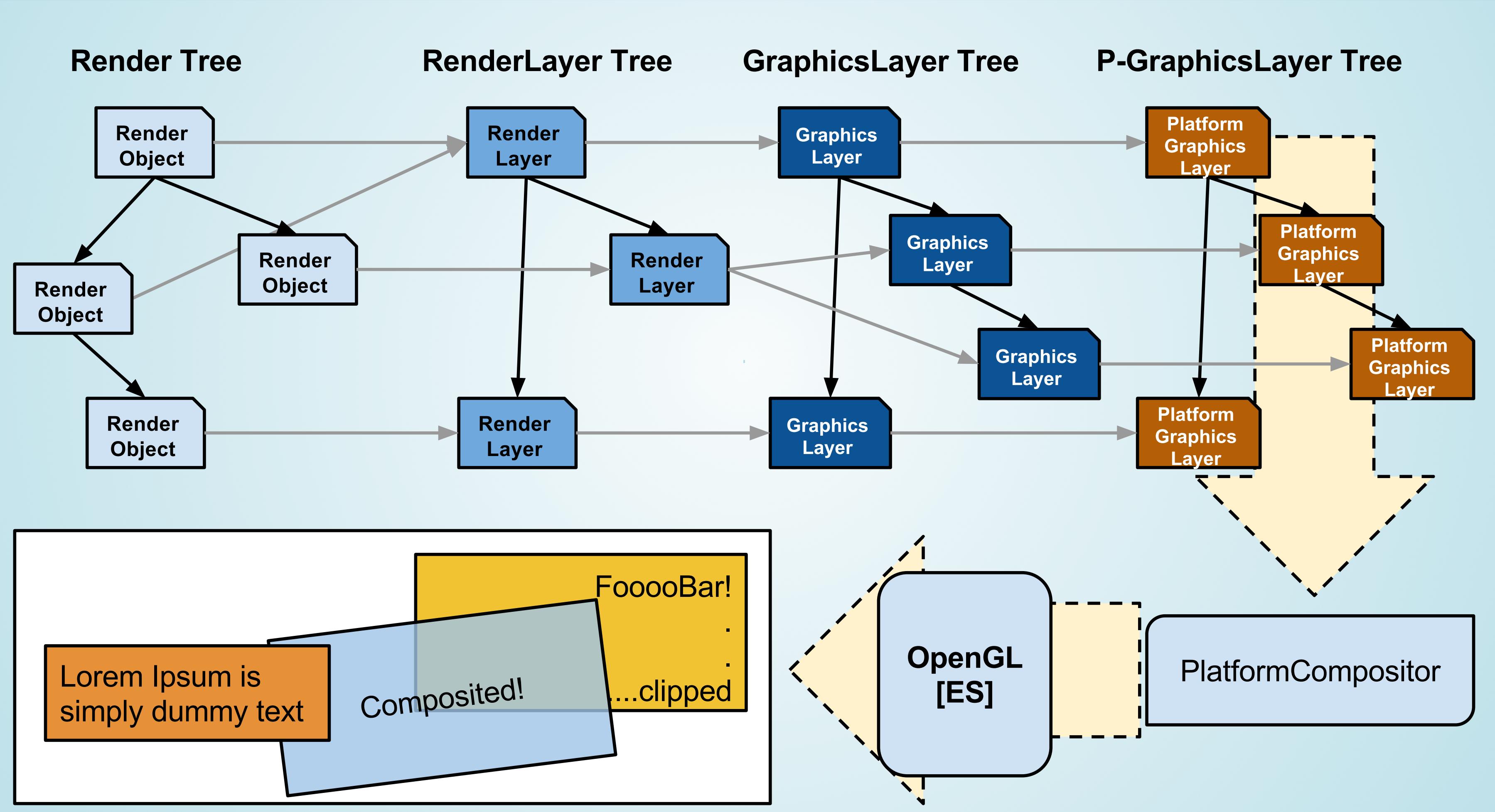
From: <https://developers.google.com/web/fundamentals/performance/critical-rendering-path/>
Under the Creative Commons Attribution 3.0 Licenses

CREATING THE RENDER TREE



From: <https://developers.google.com/web/fundamentals/performance/critical-rendering-path/>
Under the Creative Commons Attribution 3.0 Licenses

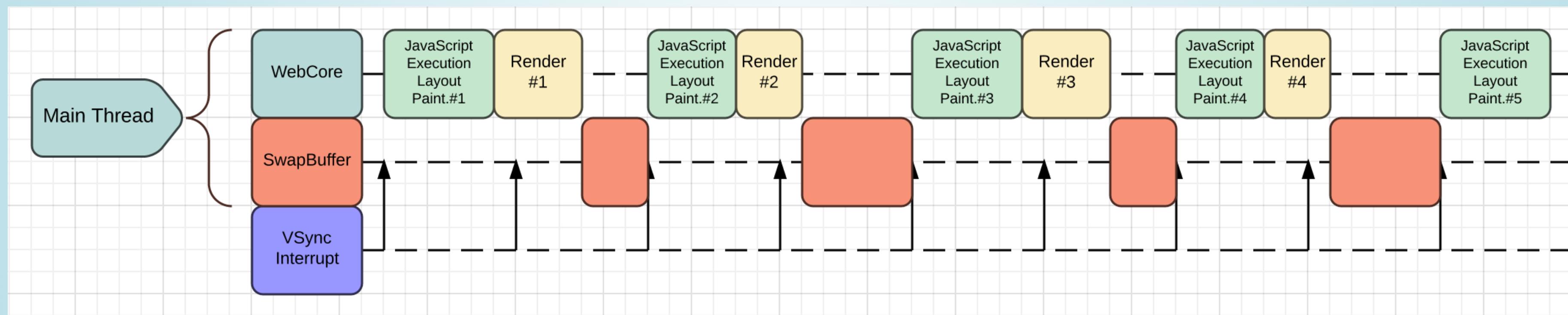
CREATING THE GRAPHICSLAYER TREE AND COMPOSING IT



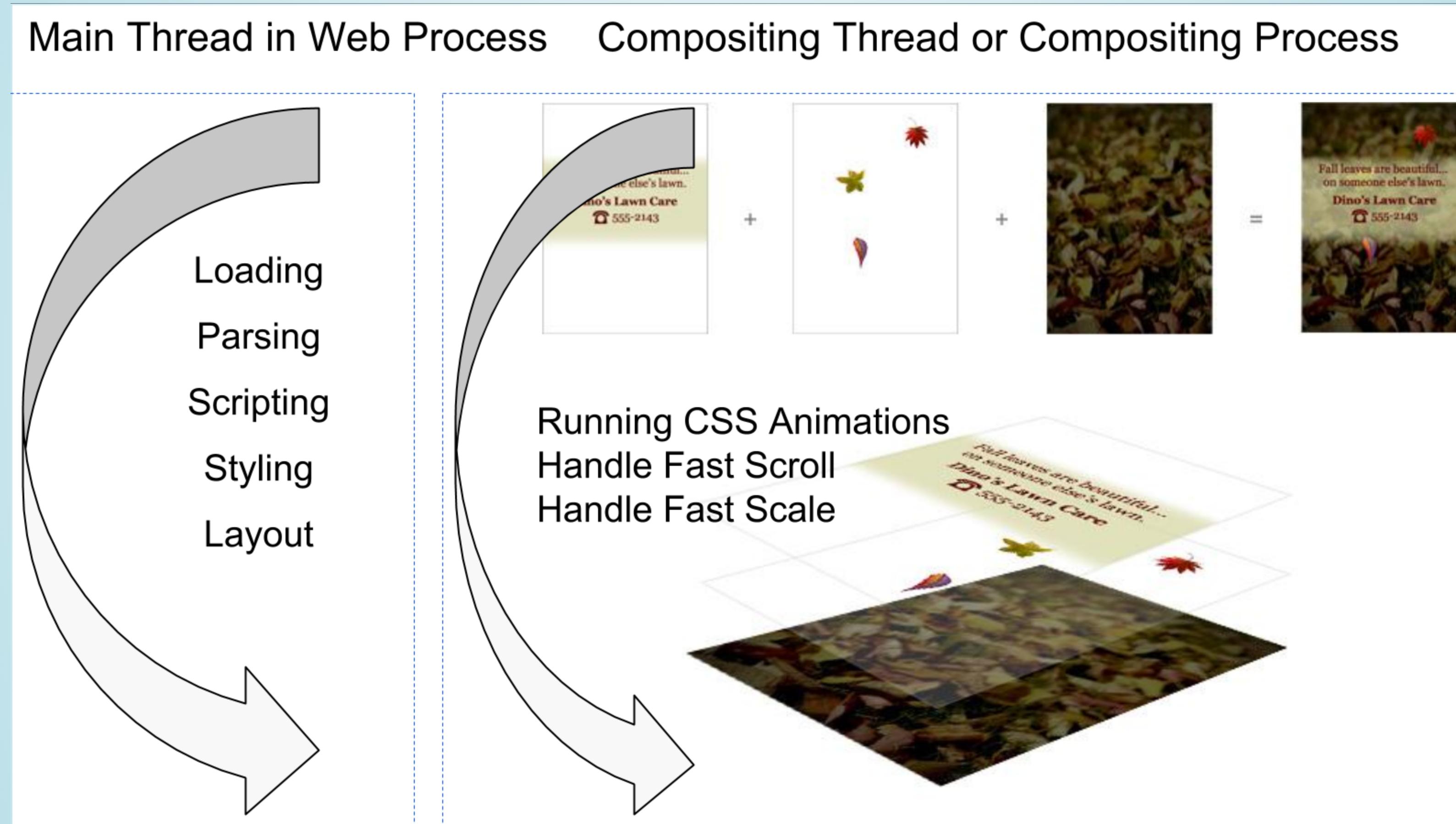
PROBLEMS

- The main-thread is always busy (Parsing, Layout, JS ...)
- The main-thread can be blocked by VSync
- And we want awesome webpages which uses HTML5 Video

VSYNC : WORST CASE

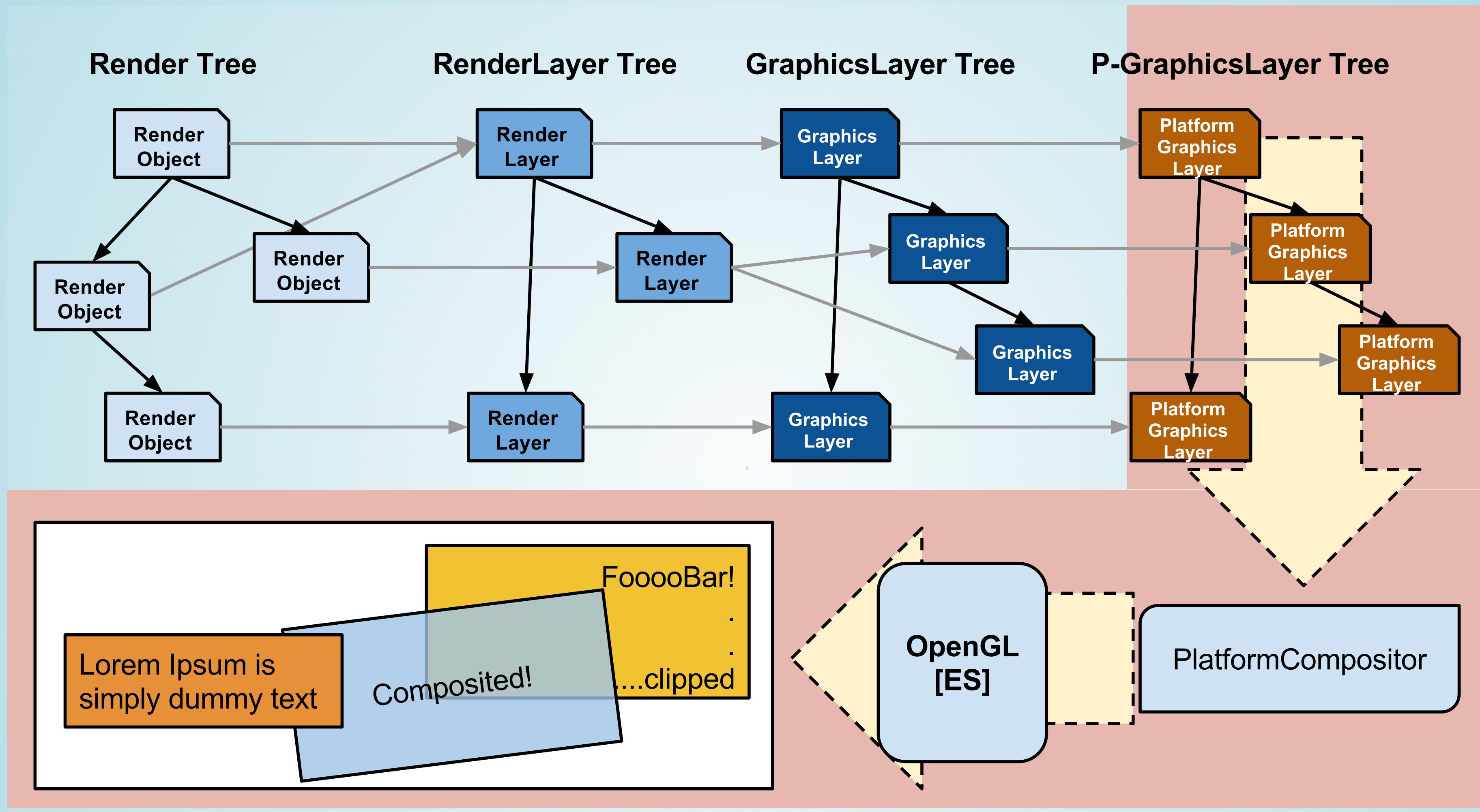


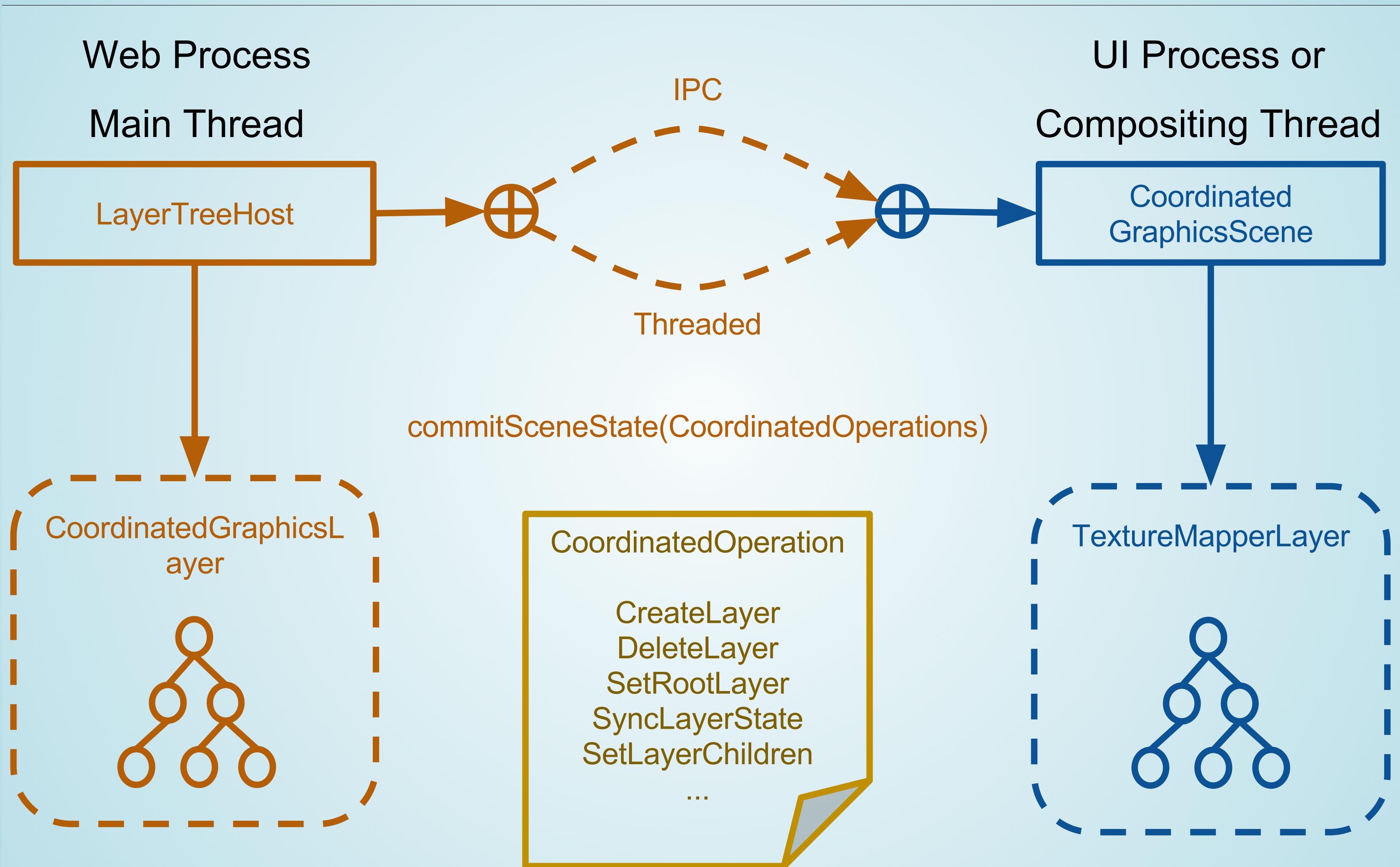
OFF-THE-MAIN-THREAD COMPOSITING



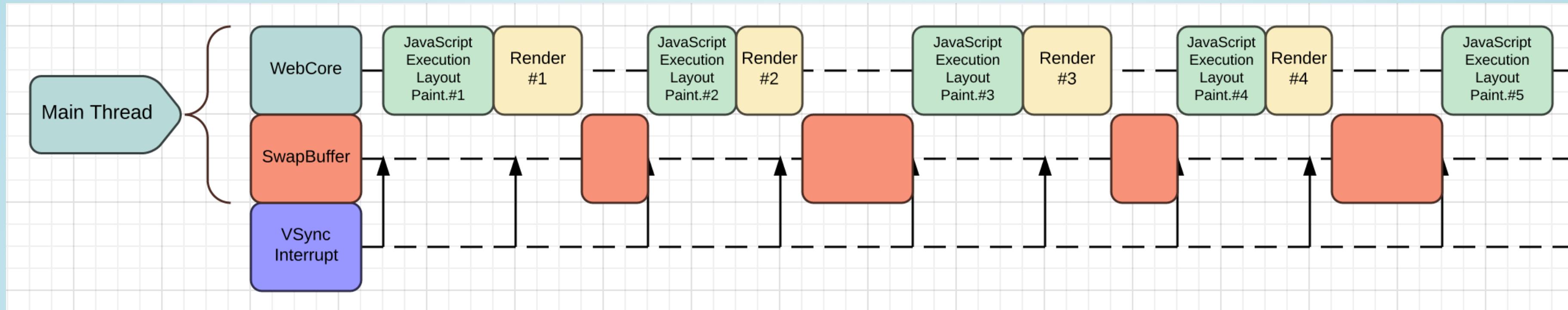
COMPOSITING IN THE DEDICATED THREAD OR PROCESS

- Free the main-thread from the Vsync and compositing operations
- It shows more smooth CSS animations, zoom, and scale operations.

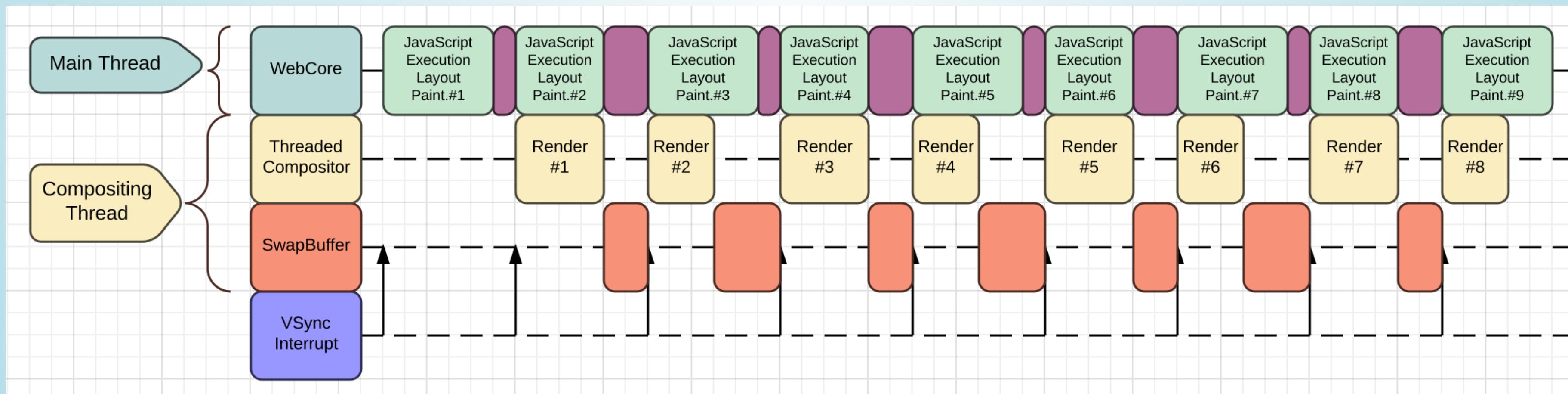




WORST CASE



SAME CASE WITH DEDICATED COMPOSITING THREAD



WHAT WE HAVE DONE : WEBKITGTK / WEBKITFORWAYLAND

- Split compositing operations into the dedicated thread
- Utilize multi-core CPUs and GPU
- Play CSS Animation off-the-main-thread
- Reduce latencies of scrolling and scaling operations

VIDEO RENDERING

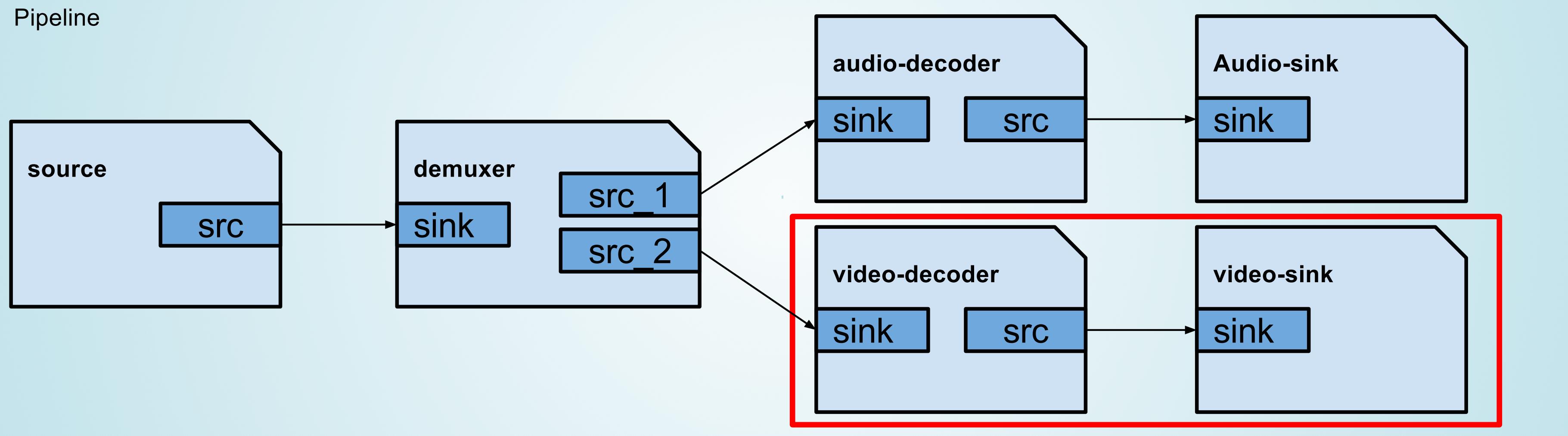
GSTREAMER

- Open source media framework for multimedia playback.
- GStreamer constructs graphs of media-handling components.
- Supports playback, streaming, complex audio mixing and non-linear video processing
- Can handle muxers/demuxers and codecs transparently.
- Add codecs/filters by writing plugins with a generic interface
- A major version is API and ABI stable.

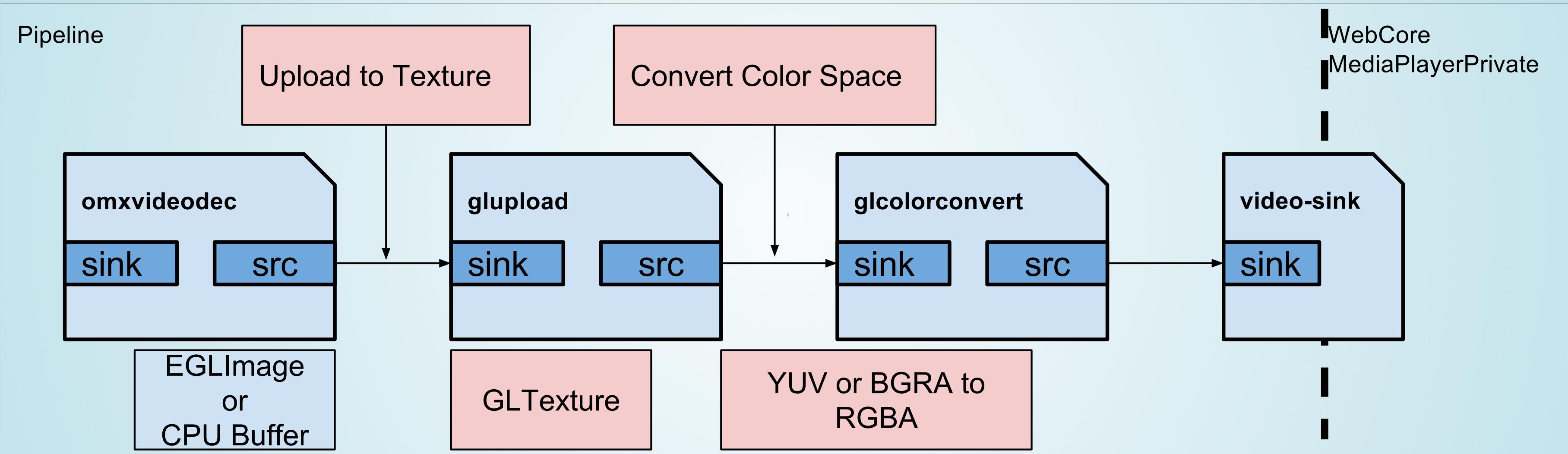
GSTREAMER OPENMAX (GST-OMX)

- Hardware decoders for H.264, VP8, Theora
- meta:EGLImage
- Custom audio sinks: HDMI and analog
- Very good integration within playbin

GSTREAMER PIPELINE



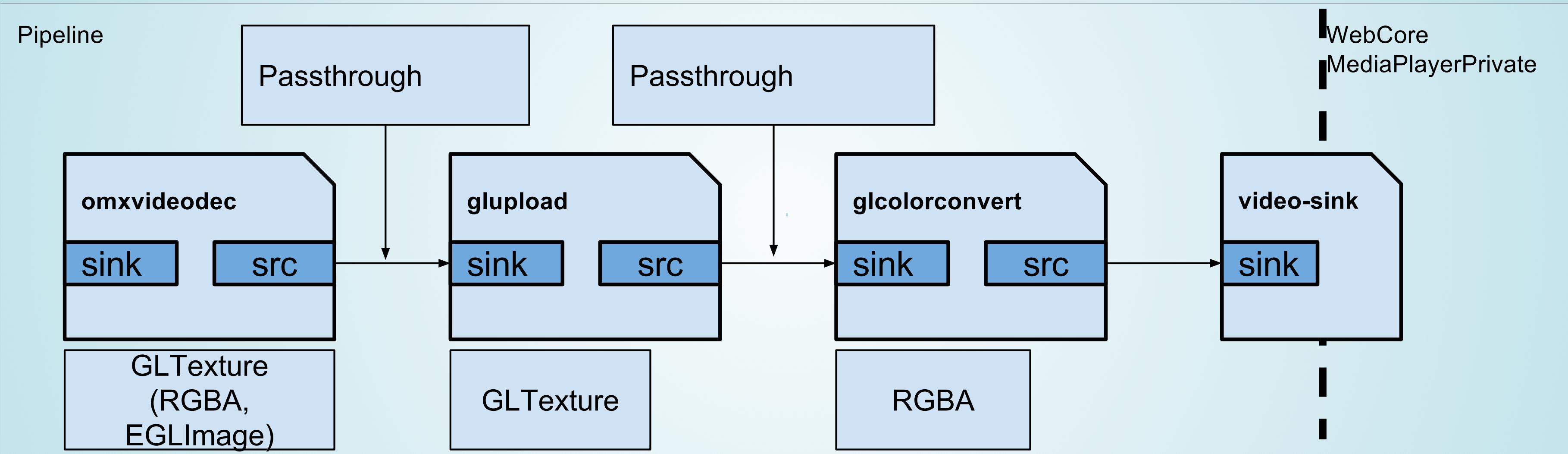
GSTREAMER PIPELINE - INEFFICIENT



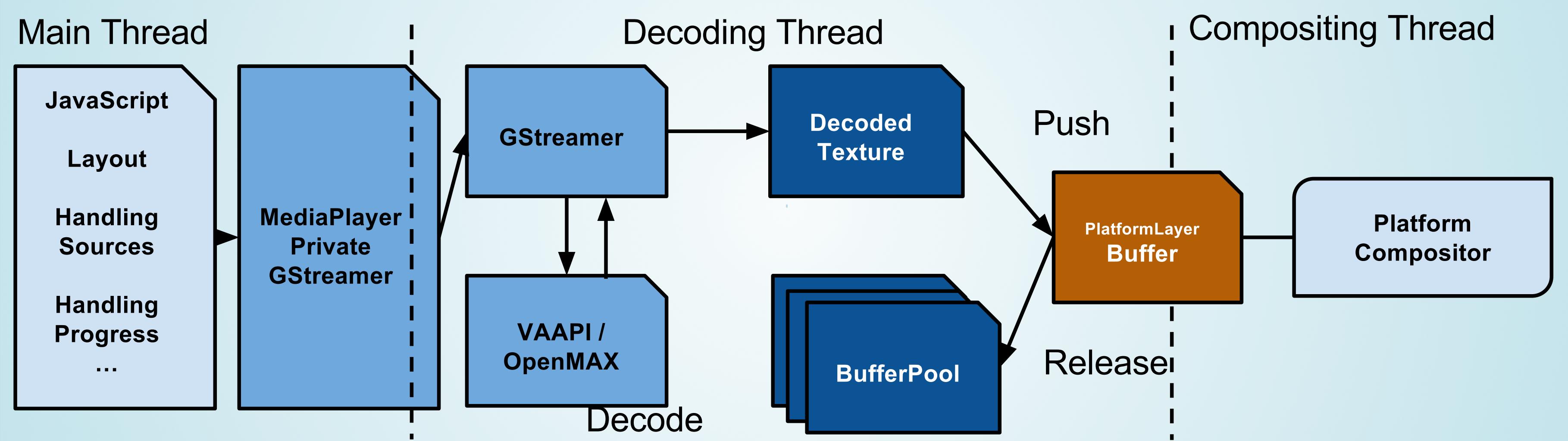
POLISH GSTOMX AND GSTGL TO REMOVE OVERHEADS

- GstGLMemoryEGL (EGLImage + GLMemory)
- Remove additional texture allocations and copy operations
- **Passthrough**

GSTREAMER PIPELINE - EFFICIENT

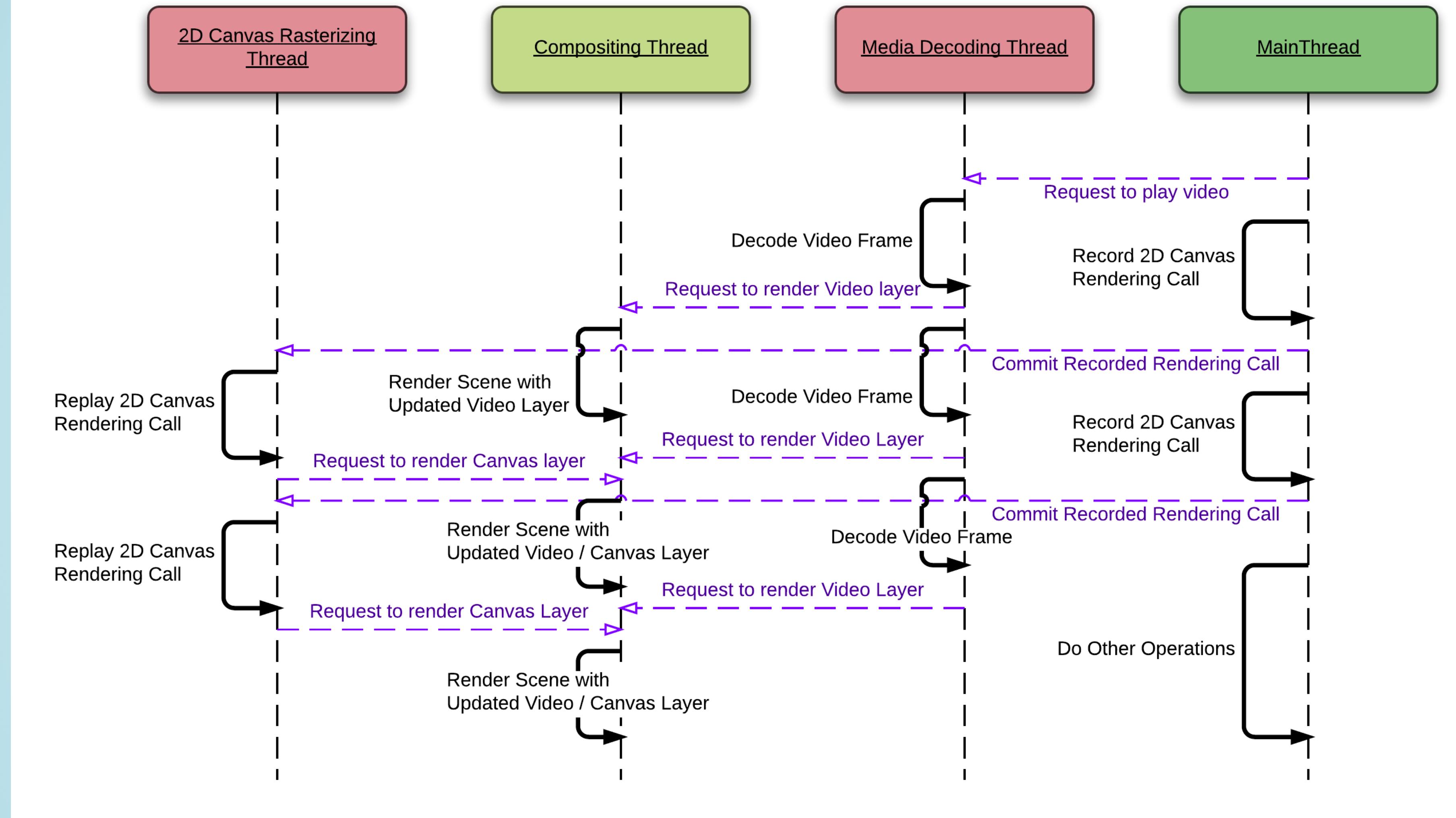


COMPOSITE DECODED FRAME



COMPOSITE DECODED FRAME

- Pass the decoded frame to the compositor directly
- Compositor composites the video without waiting main-thread



TEST RESULTS ON THE RASPBERRY PI2

- Targeting 720p, 1080p
- 30 FPS on HTML5 Video playback
- 40-50 FPS with a 720p HTML5 Video and WebGL at same time
- Reduced memory consumption
- Still, needs to reduce ghost copies of decoded frame

THANK YOU
QUESTIONS?