OOP-D: Out-of-Process Display Compositor

Chrome GPU Brownbag
2018/03/14
kylechar@chromium.org

What is OOP-D?

Moving the display compositor from the browser UI thread to a dedicated compositor thread in the GPU process.

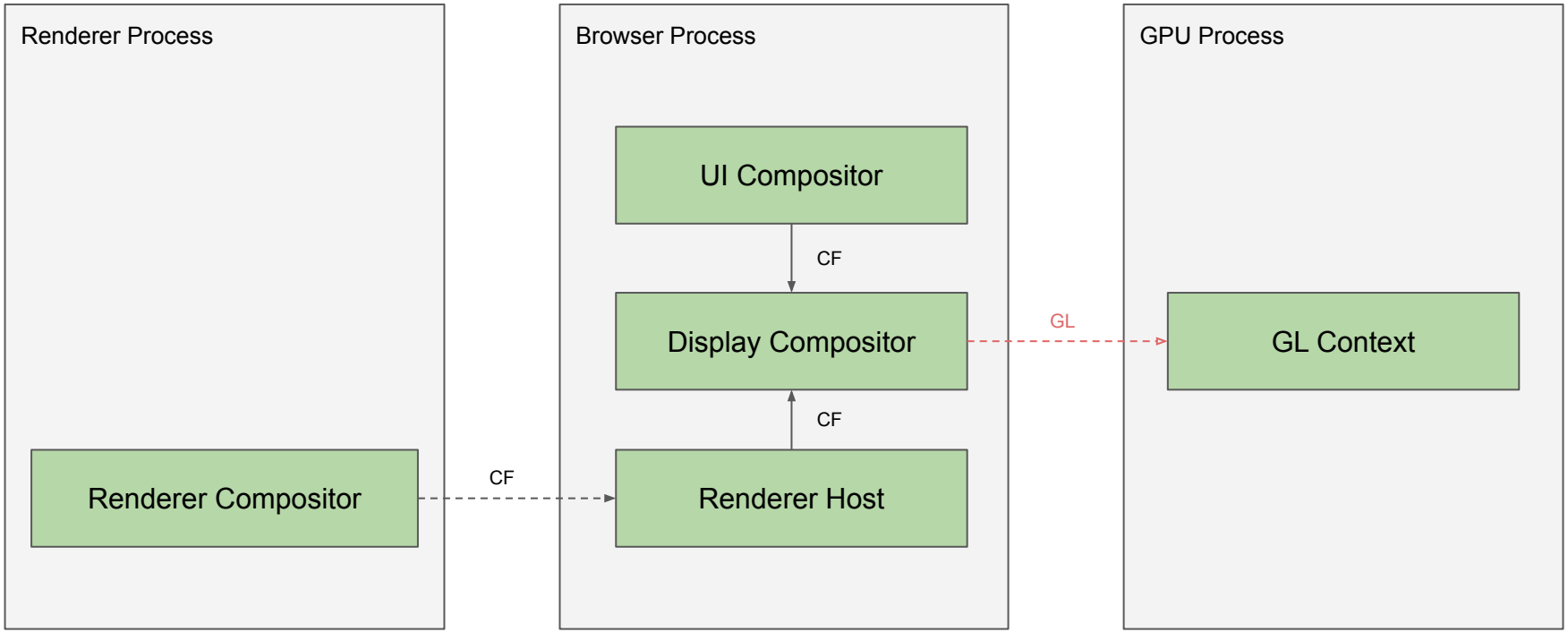
Why OOP-D?

- Enable Vulkan support
  - Vulkan compositing without writing a Vulkan command buffer.
- Enable salamander architecture
  - No renderer impl thread(s), do impl thread work in GPU process.
- Less display compositor GL overhead
  - No IPC and maybe no command buffer.
- More responsive UI
  - Display compositor not impacted by UI thread queuing delay.
  - UI thread runs fewer tasks which lowers queuing delay.
- Fewer browser process crashes

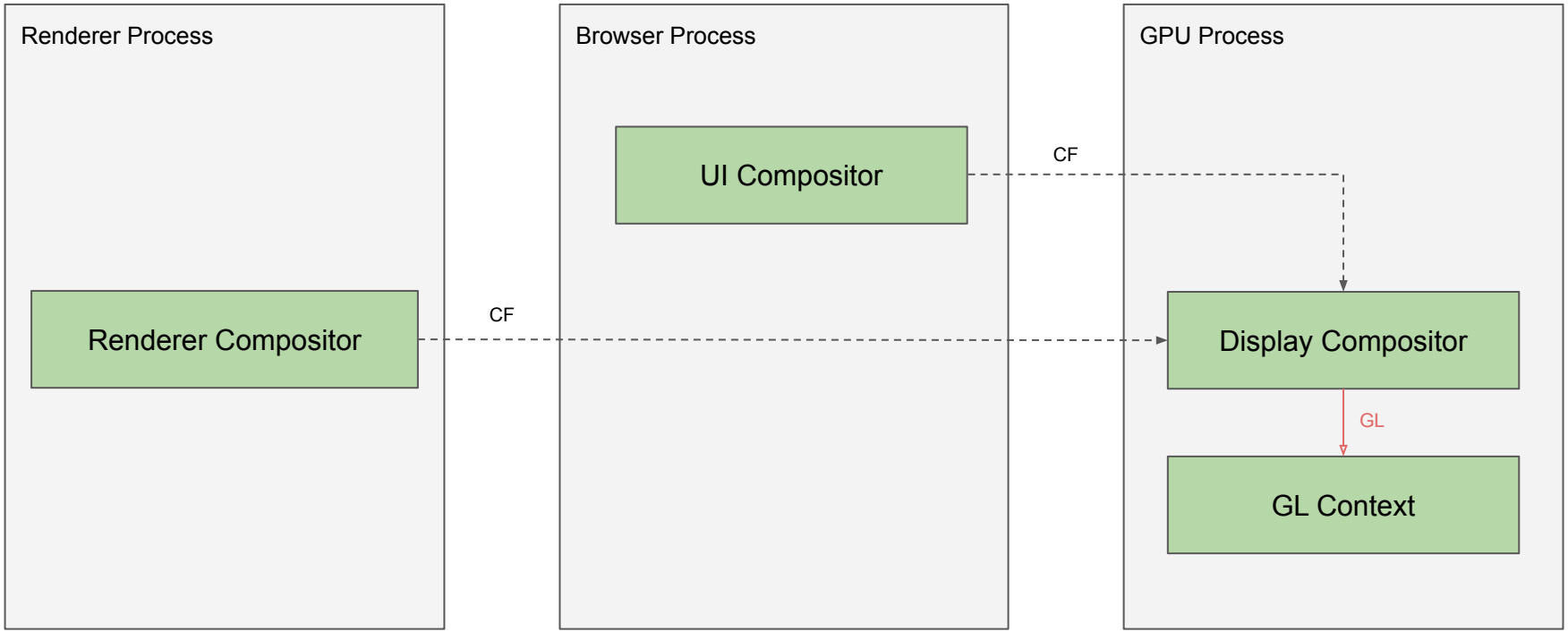
What changes with OOP-D?

1. **Where** the display compositor runs.
2. **How** the browser process controls the display compositor.
3. **How** and **where** the UI compositor submits CompositorFrames.
4. **Where** the renderer compositor submits CompositorFrames.
5. **How** renderer metadata reaches the browser.

Compositing Message Flow (Current)



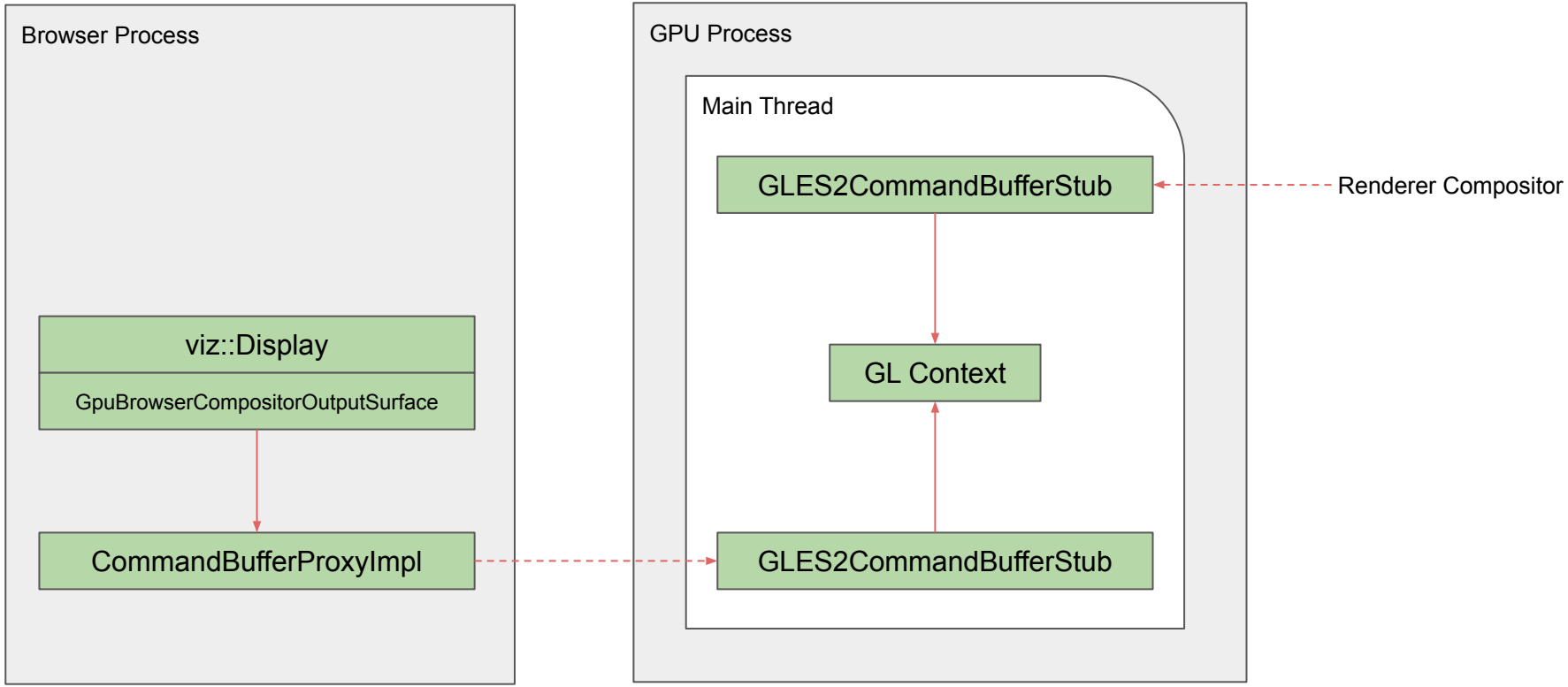
Compositing Message Flow (OOP-D)



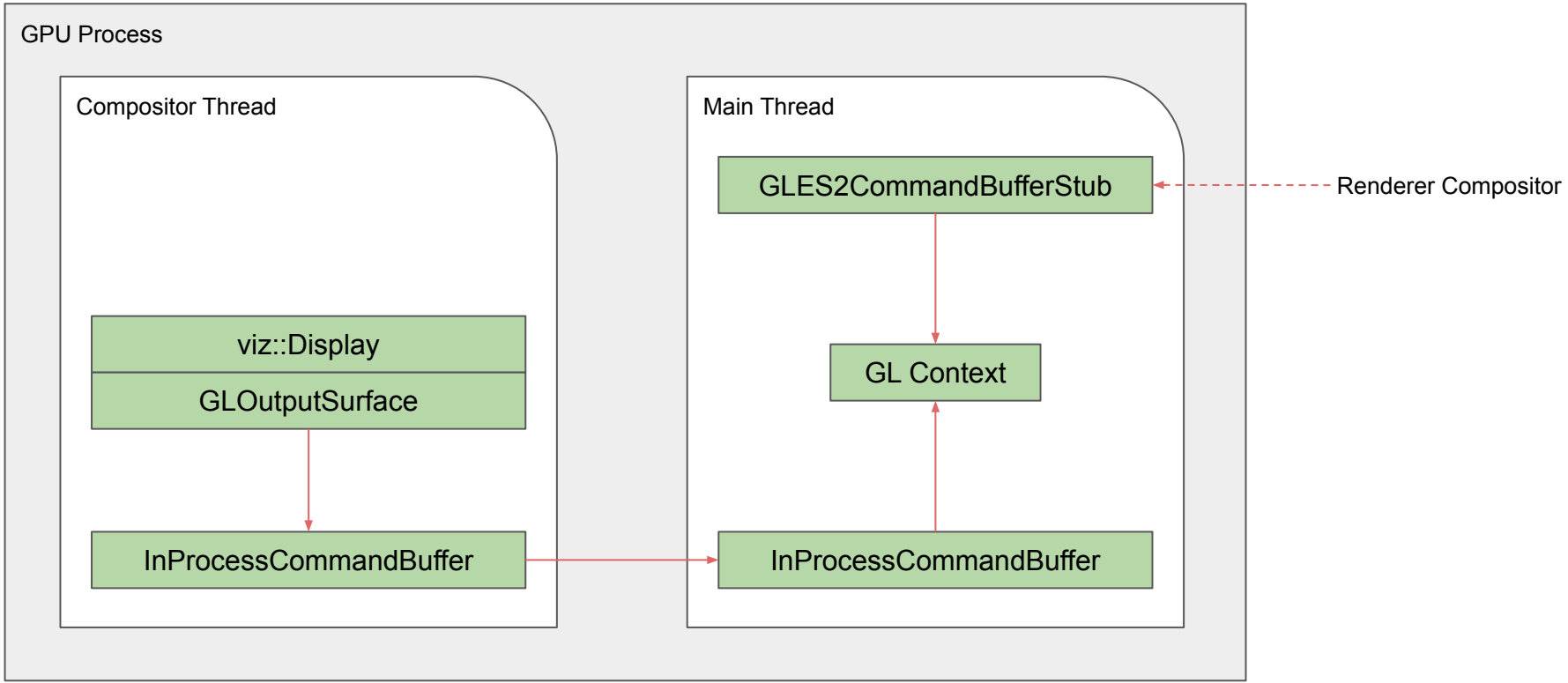
GPU Process Changes

1. Added VizCompositorThread for display compositor
    - FrameSinkManagerImpl (1) + viz::Display (n) + CompositorFrameSinkSupport (m)
2. Display compositor uses InProcessCommandBuffer
    - Sends GL commands from compositor to gpu main thread.
    - Existing code used for Android WebView and tests.
3. Software compositing runs on VizCompositorThread
    - SoftwareOutputSurface instead of GLOutputSurface

GL Command Flow (Current)



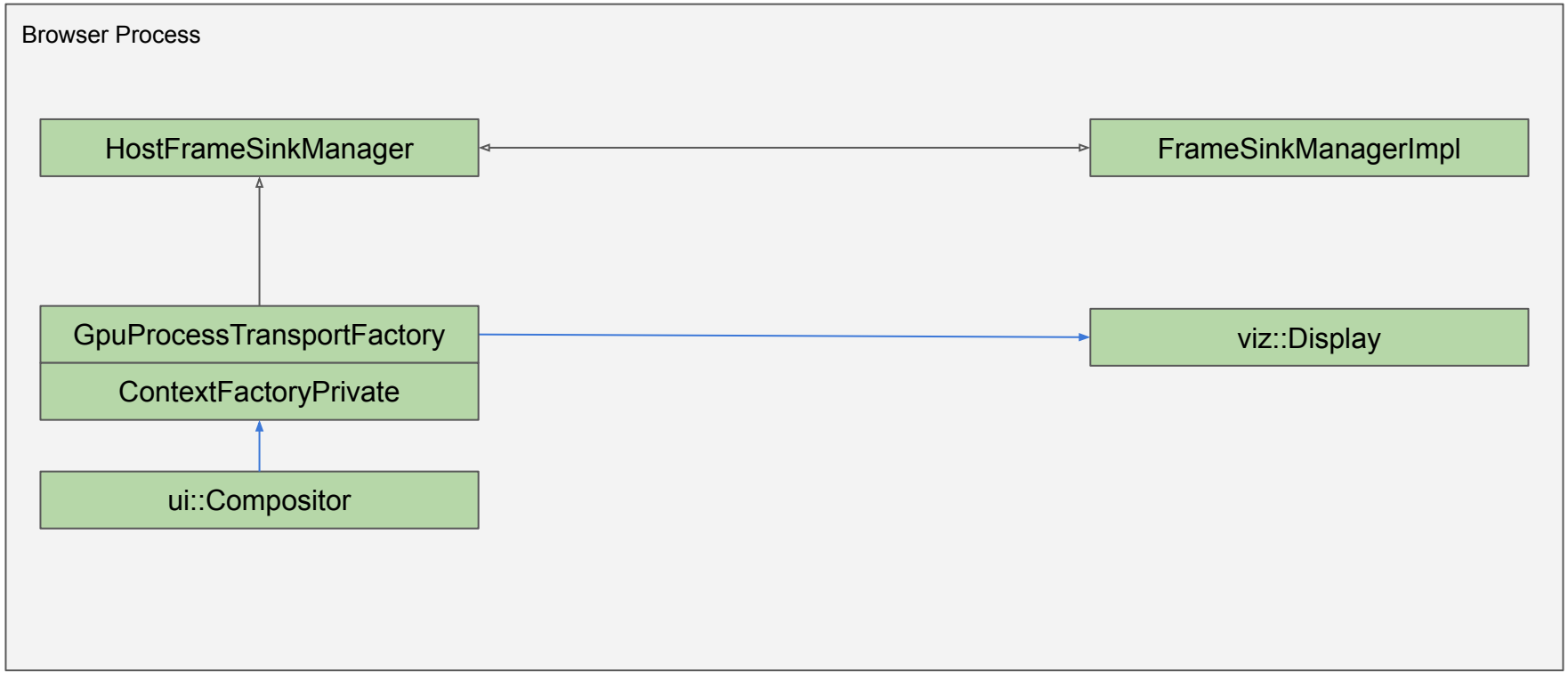
GL Command Flow (OOP-D)



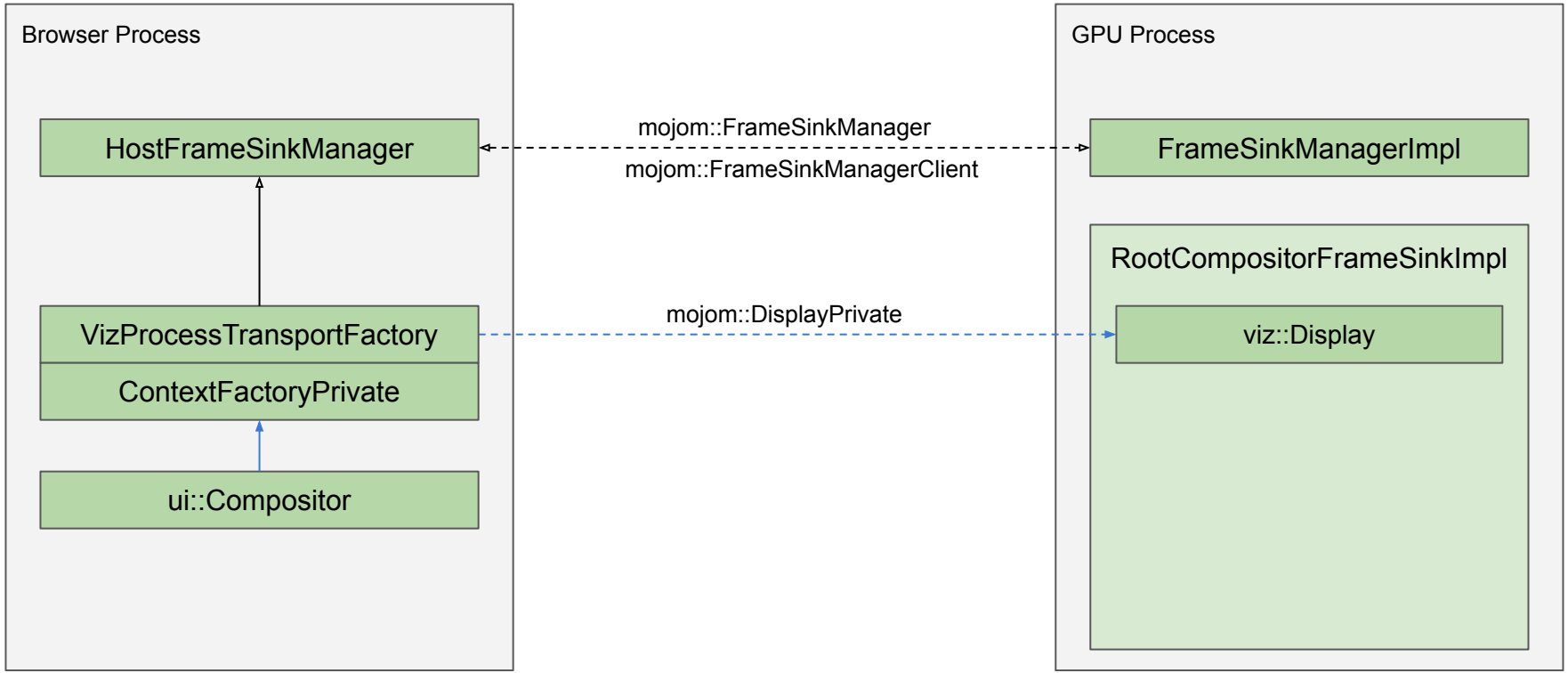
Browser Process Changes

1. No more display compositor.
2. HostFrameSinkManager uses IPC.
    - Sends IPCs to FrameSinkManagerImpl in GPU process.
3. VizProcessTransportFactory controls display compositor.
    - Replaced by GpuProcessTransportFactory.
4. CompositorFrame submission over IPC.

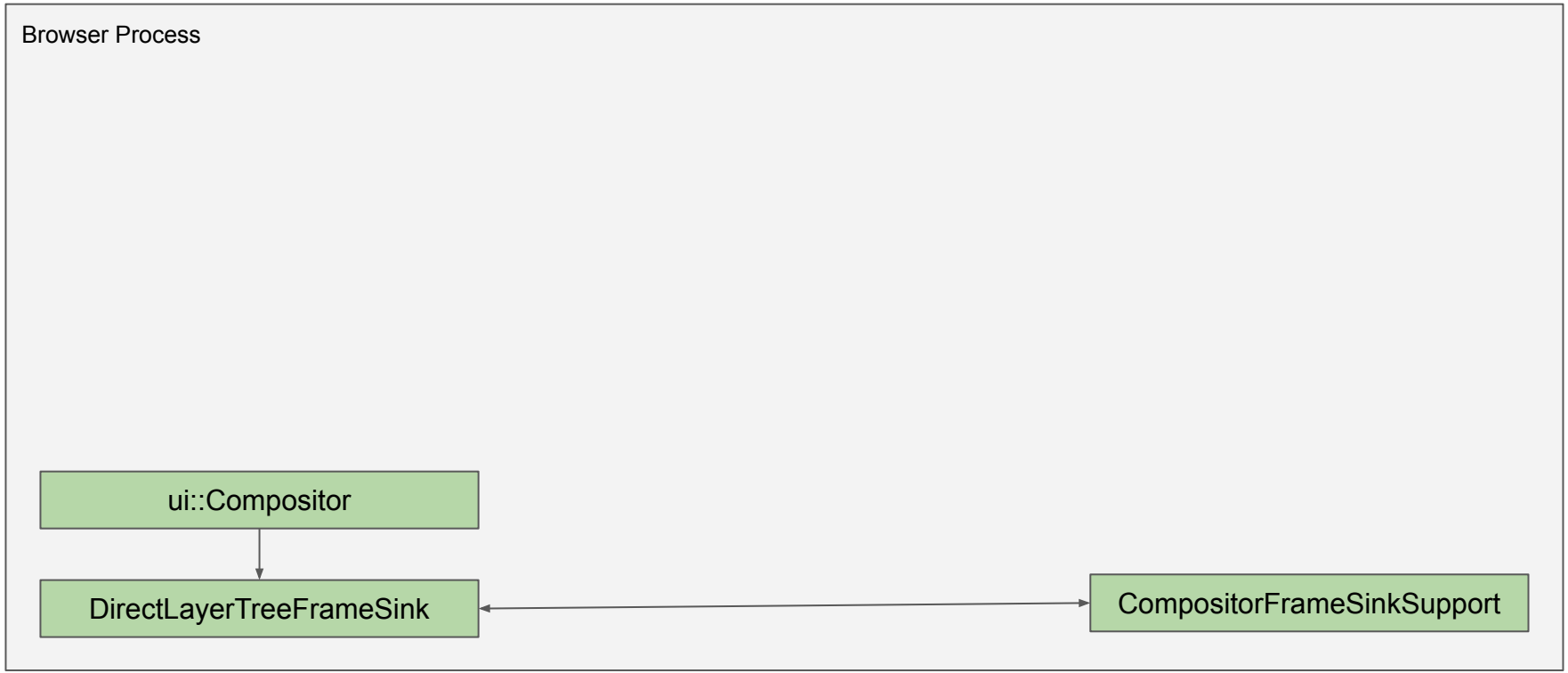
Display Compositor Control (Current)



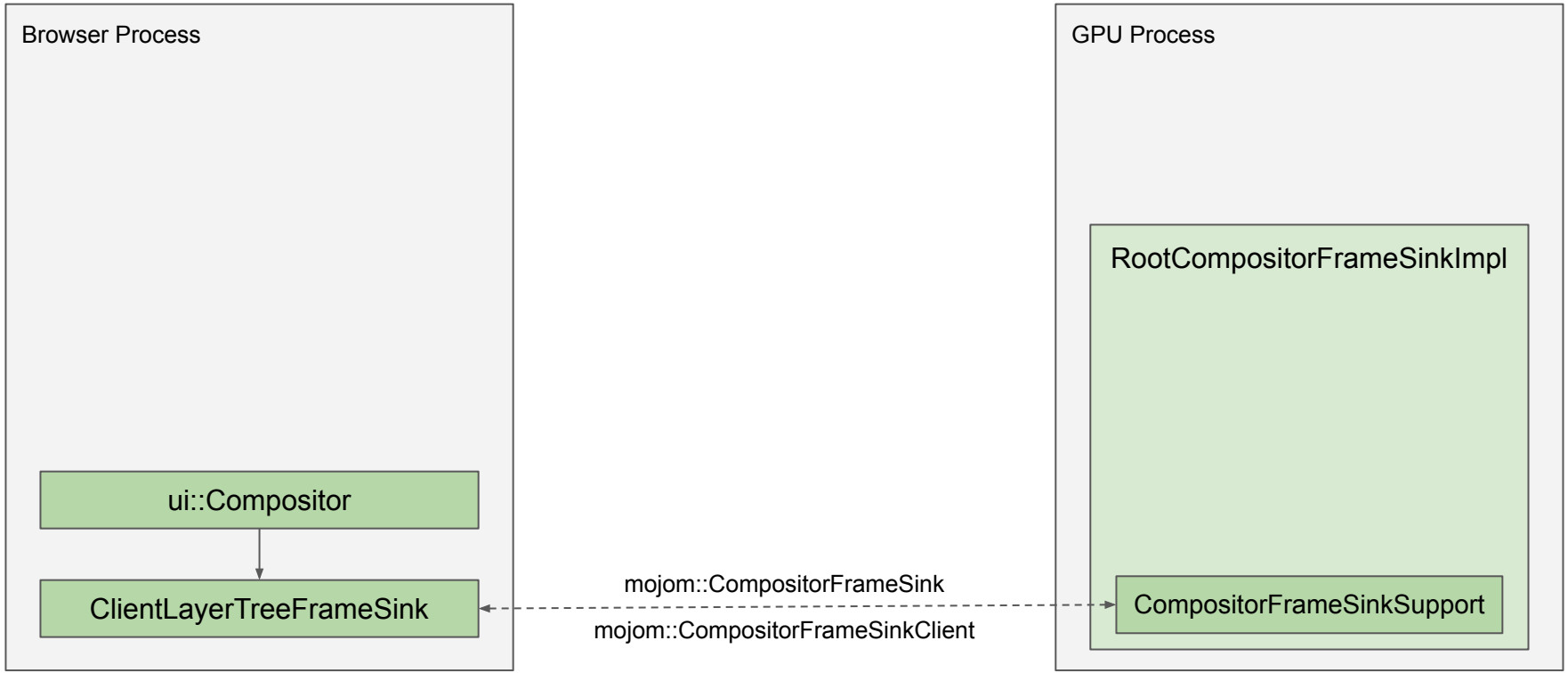
Display Compositor Control (OOP-D)



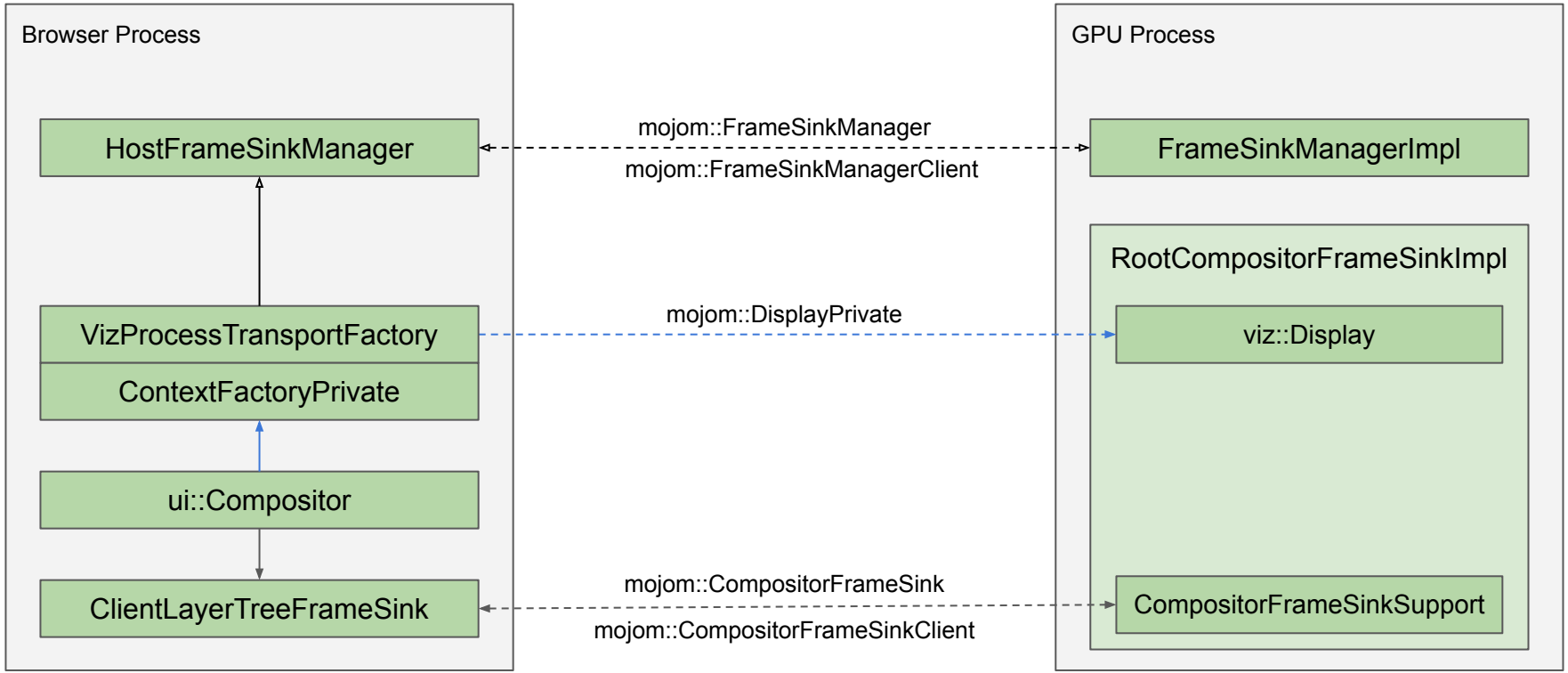
Browser Frame Submission (Current)



Browser Frame Submission (Current)



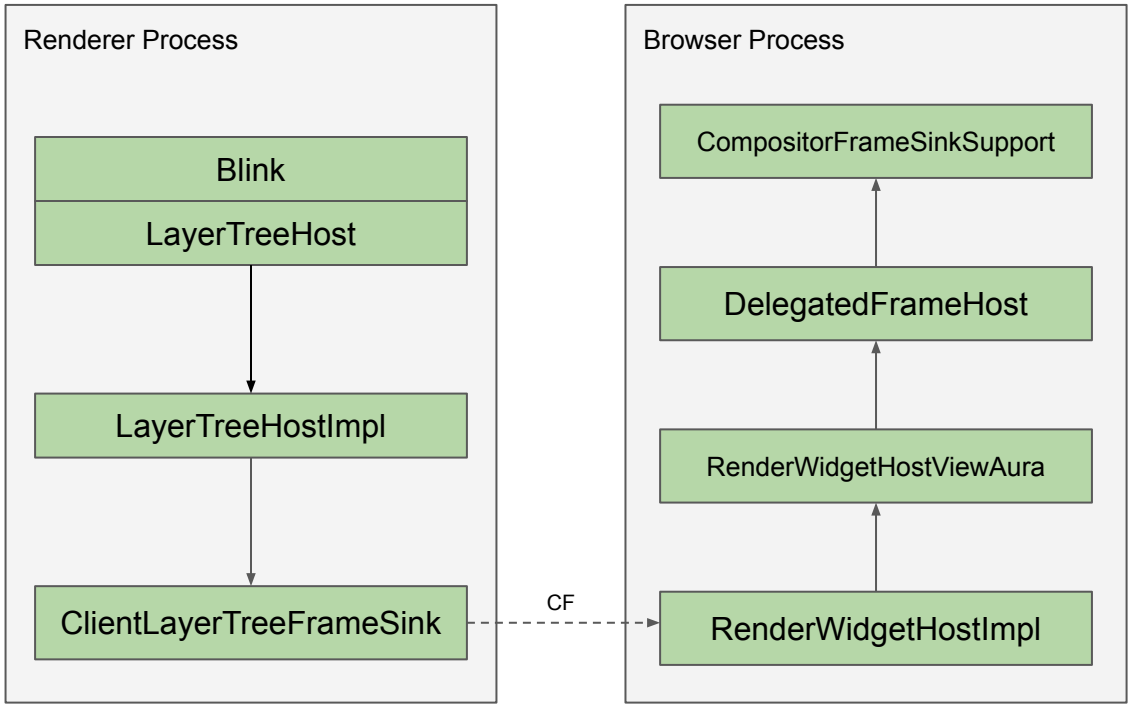
Browser-GPU IPC (OOP-D)



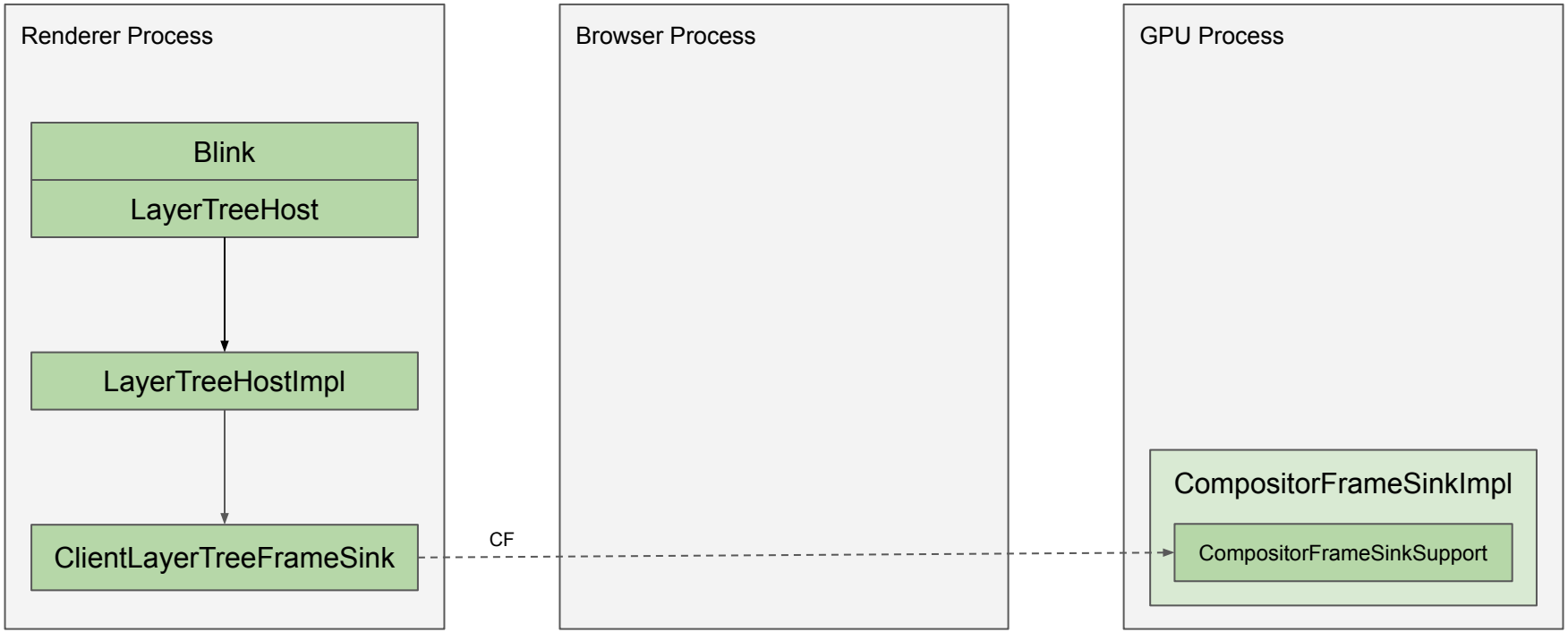
Renderer Process Changes

1. Submits CompositorFrames to GPU.
    - Renderer is unaware anything changed.
2. Browser can't inspect renderer CompositorFrames. Need to send renderer metadata to browser process directly.

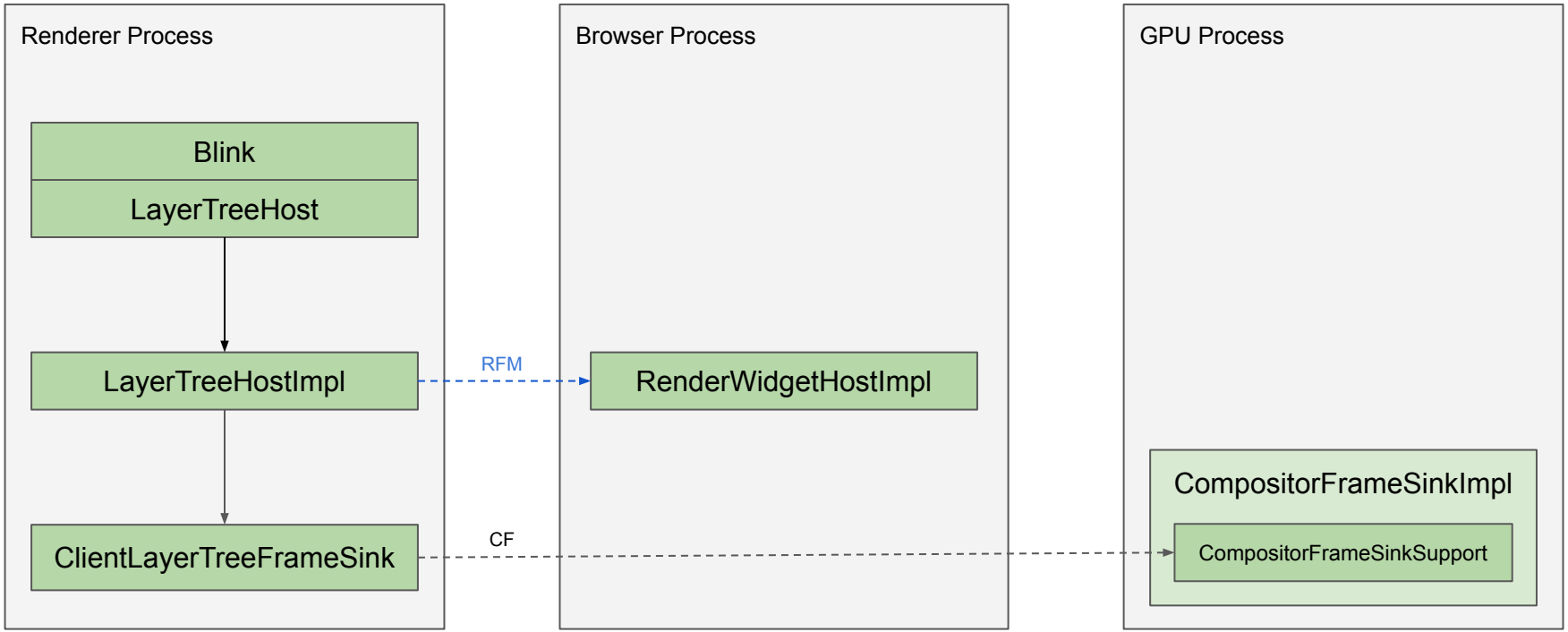
Renderer Frame Submission (Current)



Renderer CompositorFrame Submission (OOP-D)



Renderer CompositorFrame Submission (OOP-D)



OOP-D Status

- Works (mostly) on Windows, Linux and Mac
- Rough edges:
  - Software compositing for video, pepper, etc.
  - DevTools renderer data and screen capture
  - OOPIF hit testing
  - InProcessCommandBuffer
  - Testing
- Guarded by VizDisplayCompositor feature
  - Command line `--enable-features=VizDisplayCompositor`
  - chrome://flags/#enable-viz-display-compositor

OOP-D Tracking

- Hoping to start finch trial in M67
  - Windows and Linux on canary/dev channels
- OOP-D Finch Trial
  - https://crbug.com/787097
- OOP Launch on Desktop
  - https://crbug.com/730193

Questions?