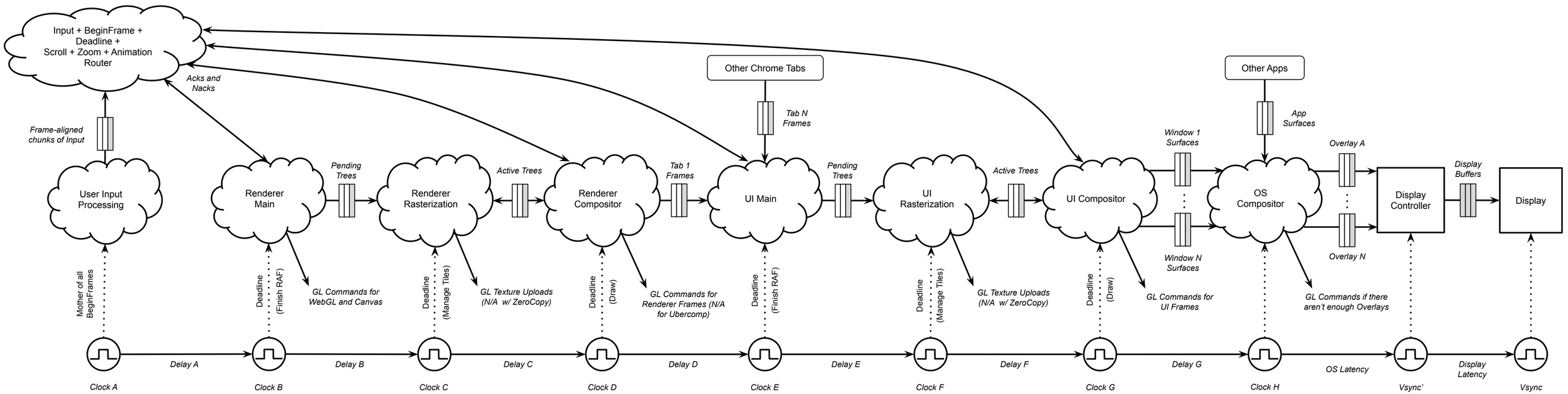
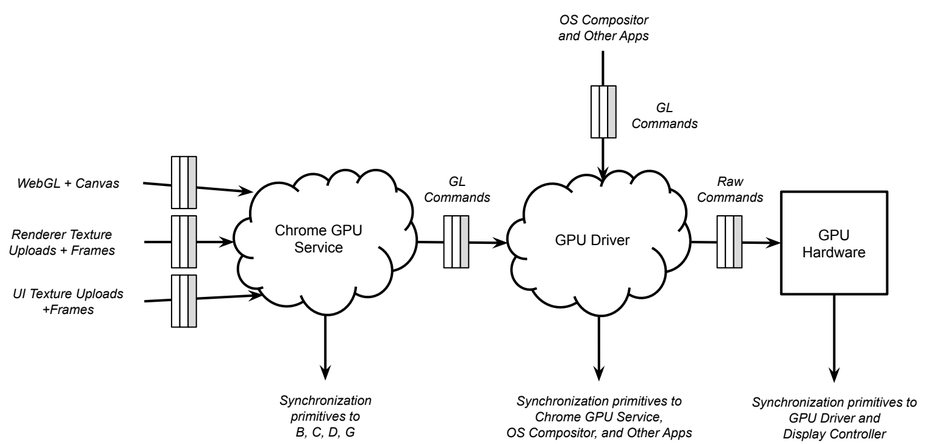
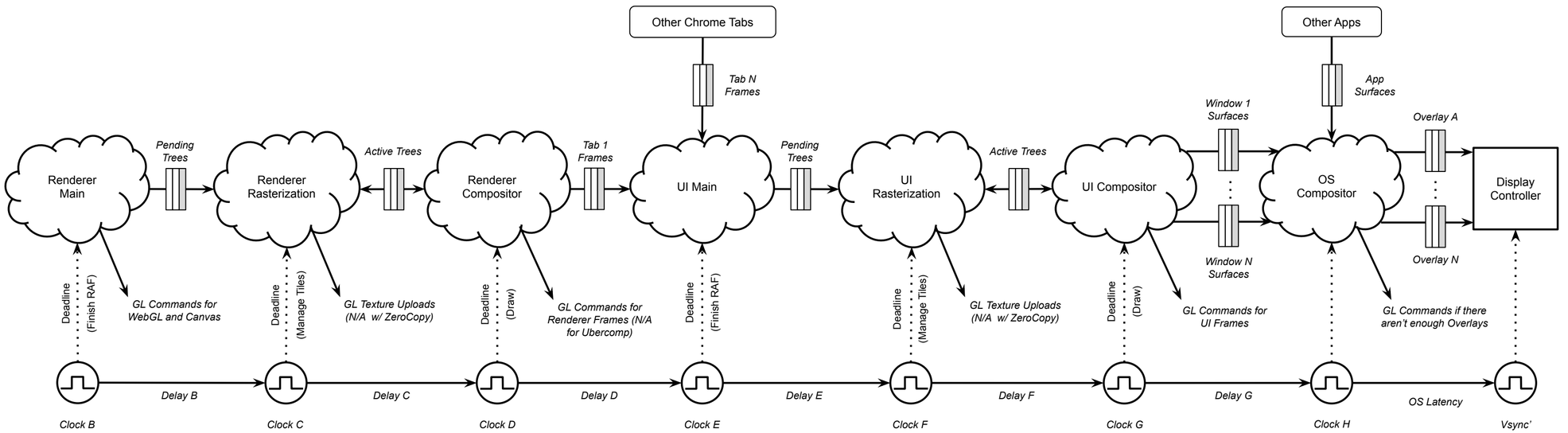
Chrome
Frame Synchronization

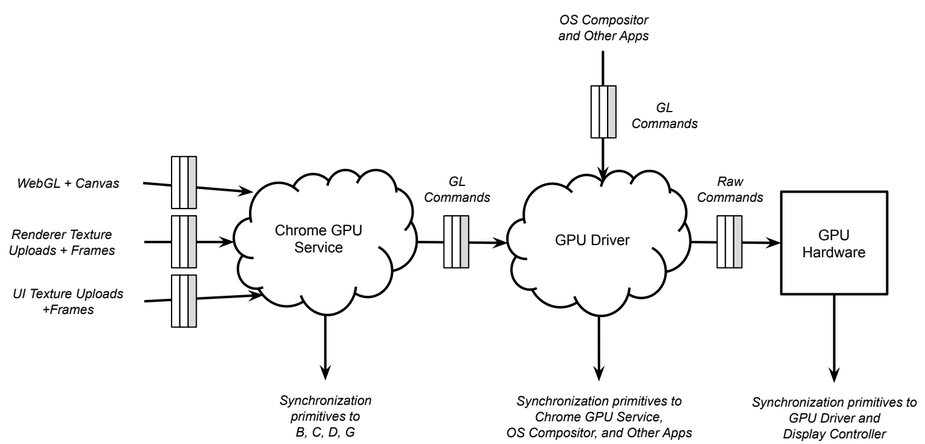
Chrome Rendering Pipeline Overview





Chrome Rendering Pipeline Overview (Pertinent to Frame Synchronization)





Double Buffered WebGL + Impl-Side Painting + Ubercomp

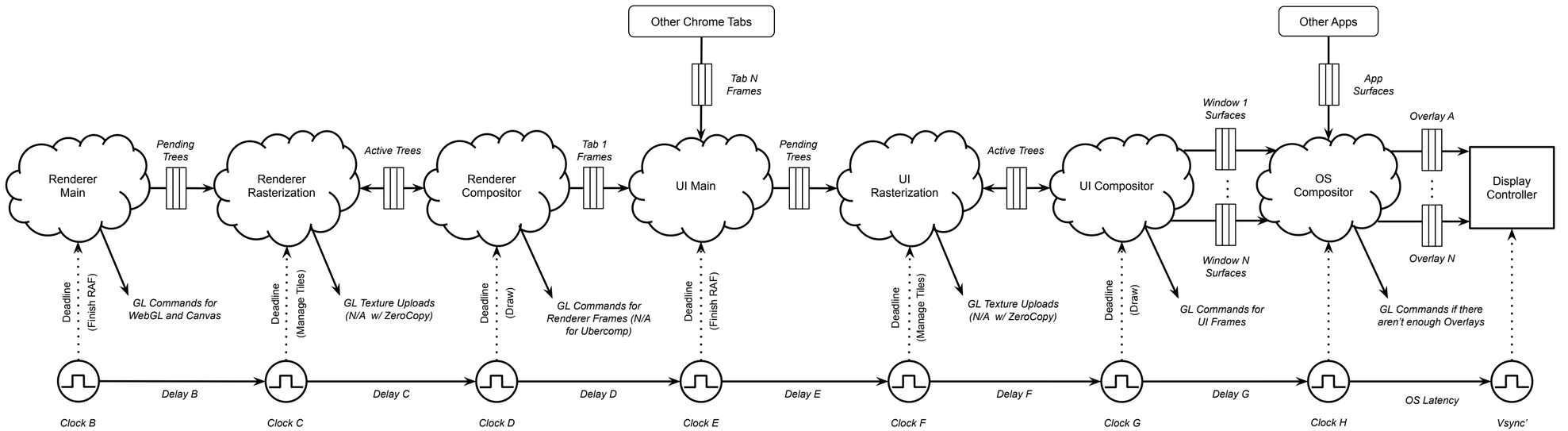
The following is a hypothetical example of frame production and ordering.

It doesn’t reflect what Chrome does today, but is meant to be used as a framework for discussion and serve as an example of the scheduling/ordering issues we are running into.

Ubercomp makes frame synchronization a little more complicated if we want more RAF concurrency.

Each slide represents a progression in time. Although there’s a lot of detail for the sake of completeness, hopefully you don’t need to understand every slide to understand the context of a single slide. Please pay particular attention to the slides with comments.

Double Buffered WebGL + Impl-Side Painting + Ubercomp

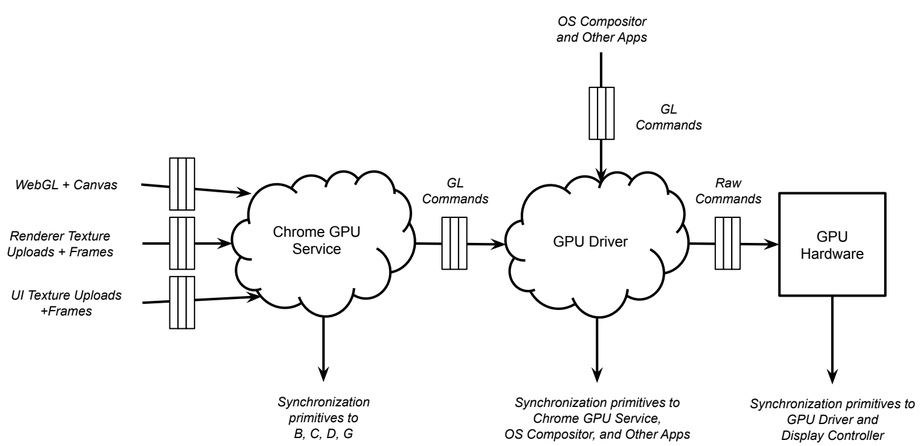


WebGL FB1

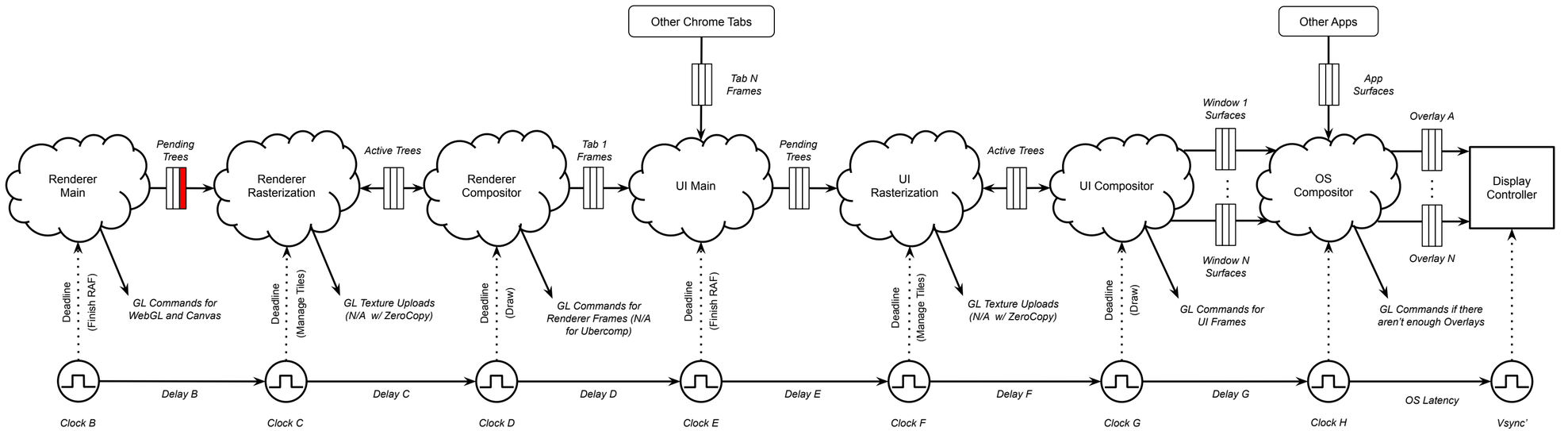
WebGL FB2

Window1 FB1

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

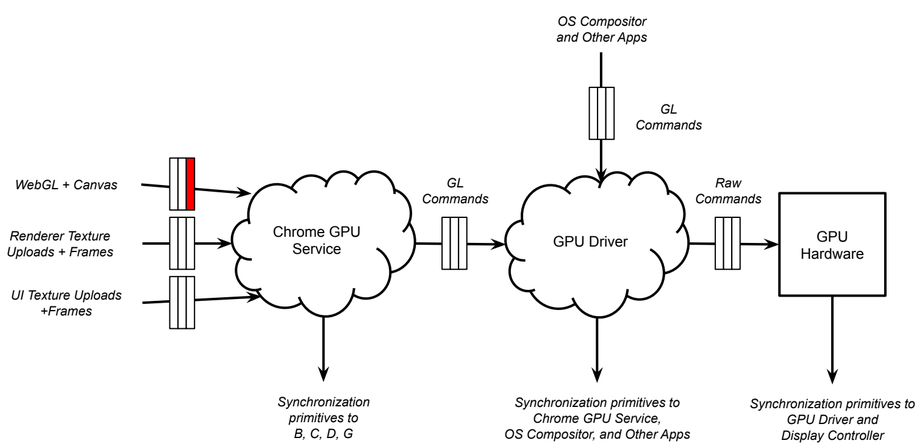


WebGL FB1: RAF sends GL commands to the Chrome GPU Service and submits the DOM as a pending tree to the compositor with one of the layers referencing this buffer.

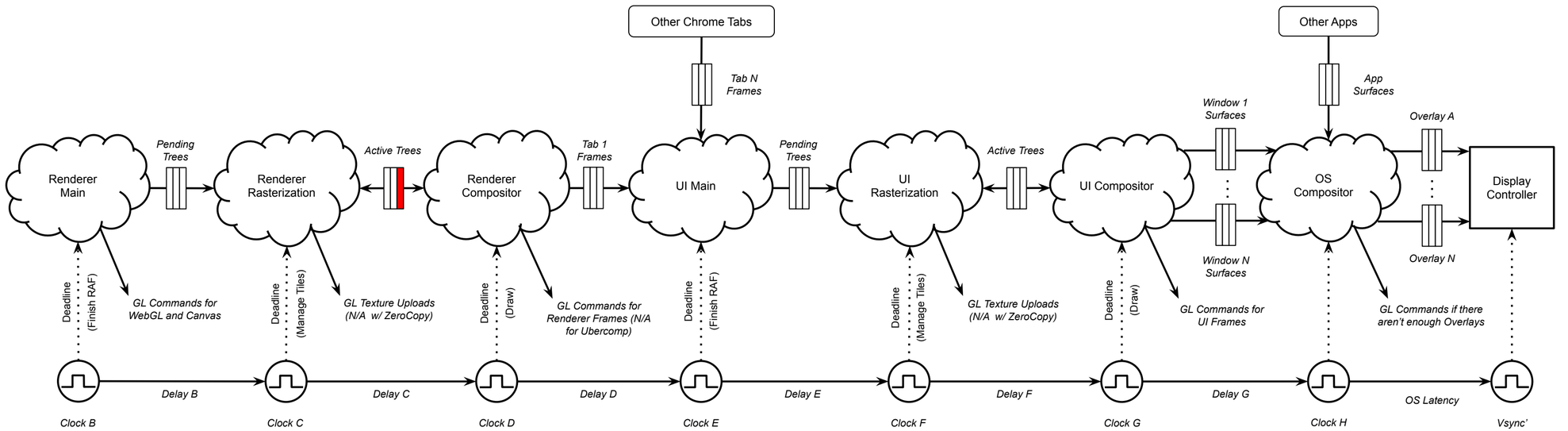
WebGL FB2

Window1 FB1

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

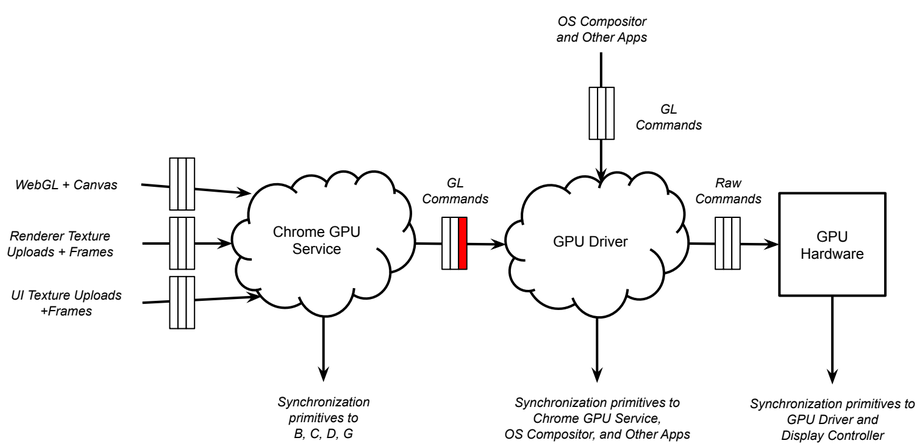


WebGL FB1: The pending tree completes rasterization and is activated. Chrome's GPU service submits the WebGL commands to the driver.

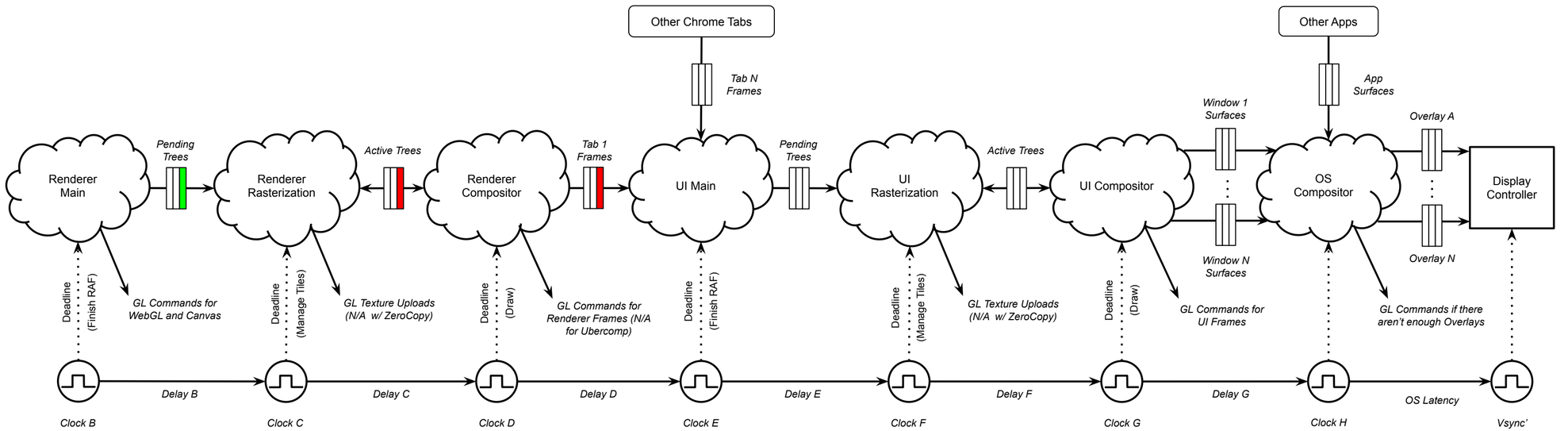
WebGL FB2

Window1 FB1

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

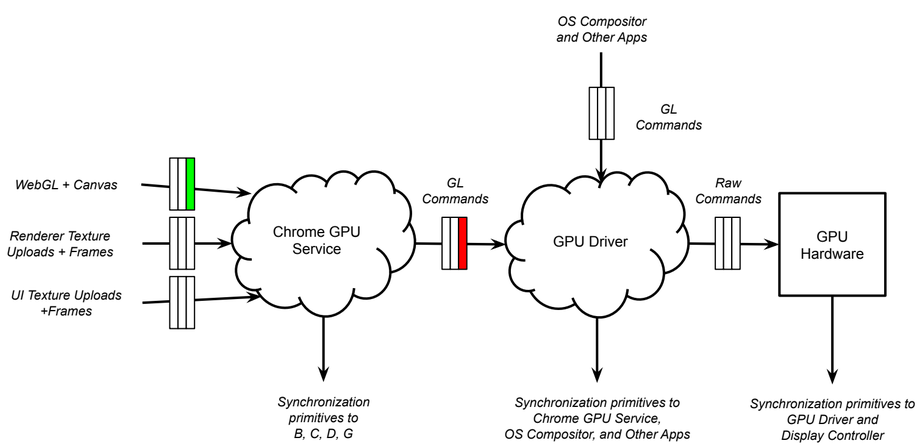


WebGL FB1: The Renderer submits a frame to the Browser with a reference to this buffer.

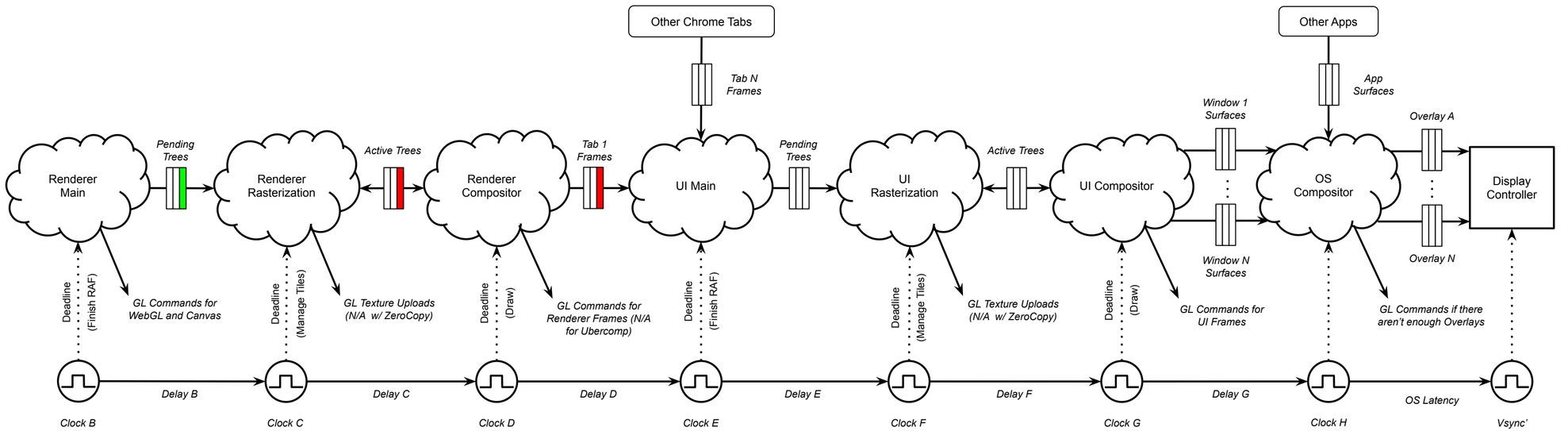
WebGL FB2: Another RAF sends GL commands to the Chrome GPU Service and submits the DOM as a pending tree to the compositor.

Window1 FB1

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

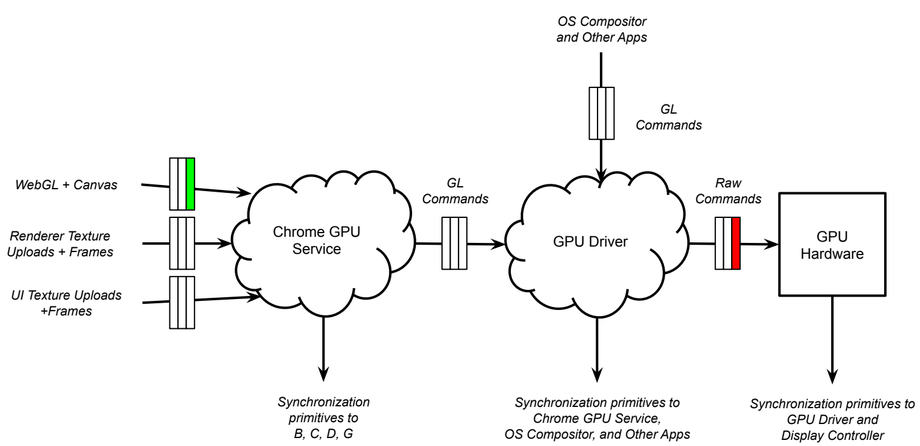


WebGL FB1: The Browser hasn't picked up the first Renderer frame referencing this buffer yet, but the commands have been submitted to the GPU hardware.

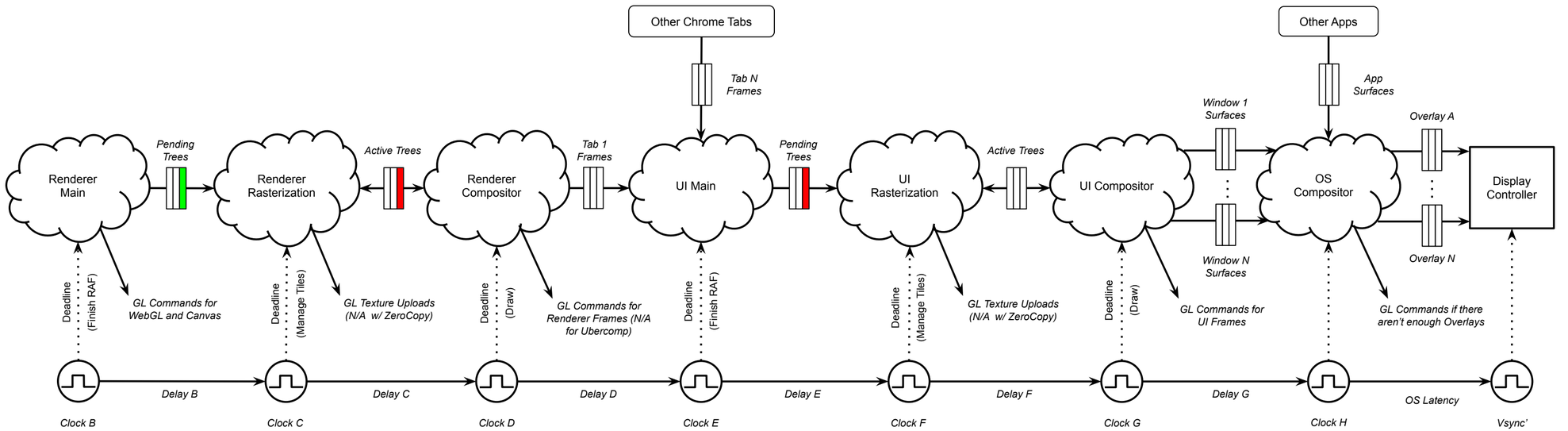
WebGL FB2: Rasterization hasn't completed. The Chrome GPU Service waits to submit this buffer's commands to the driver until notified.

Window1 FB1

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

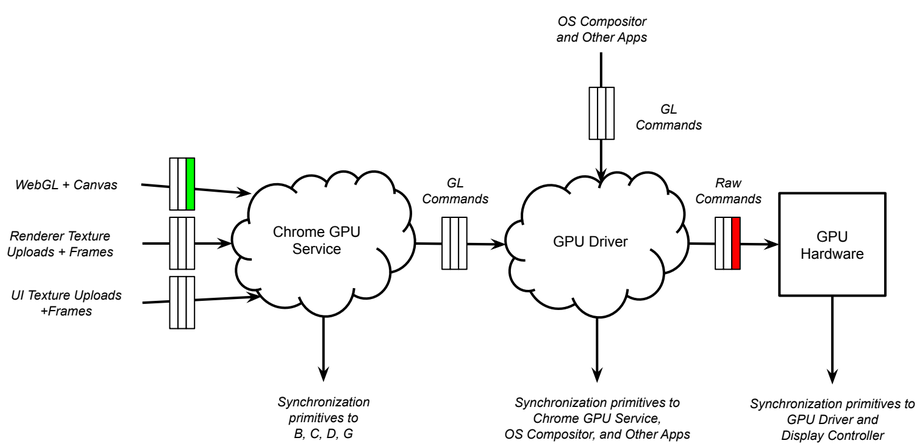


WebGL FB1: The Browser starts to pick up the first Renderer frame referencing this buffer. The GPU hardware is still working to produce this buffer.

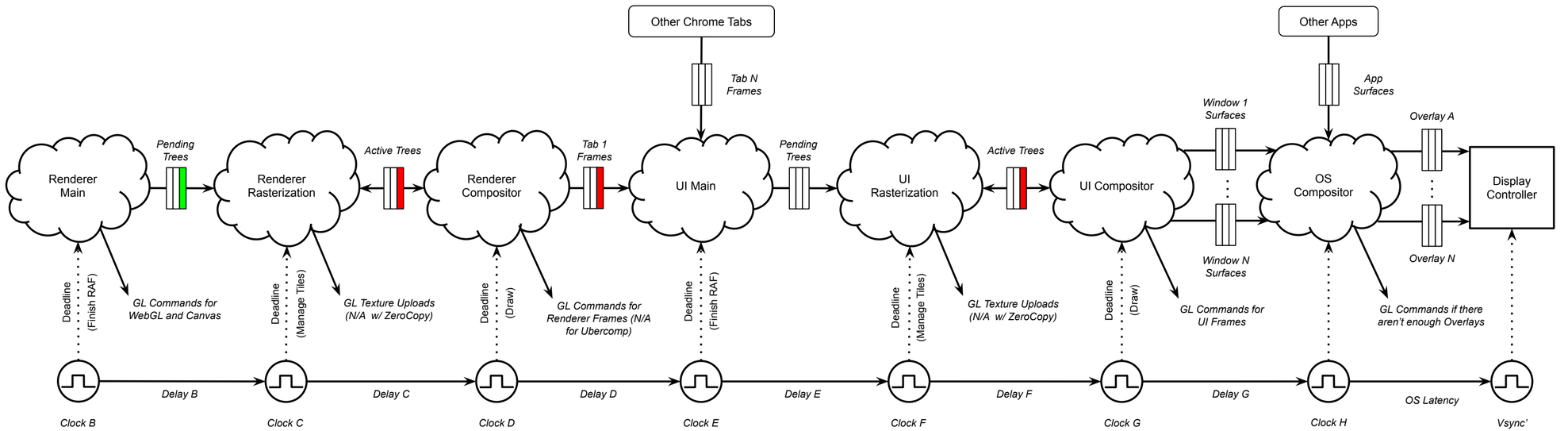
WebGL FB2: Rasterization hasn't completed. The Chrome GPU Service waits to submit this buffer's commands to the driver until notified.

Window1 FB1

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

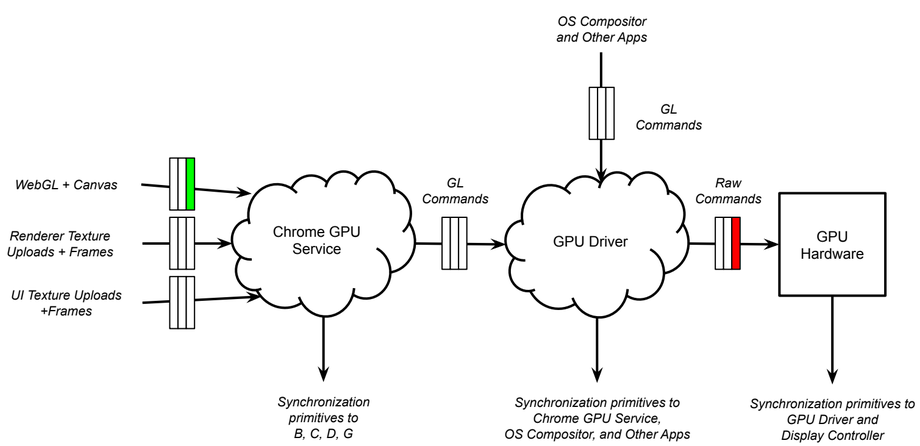


WebGL FB1: The Renderer submits a second frame that has this WebGL buffer. The Browser activates the first frame referencing this WebGL layer. The GPU hardware is still working to produce this buffer.

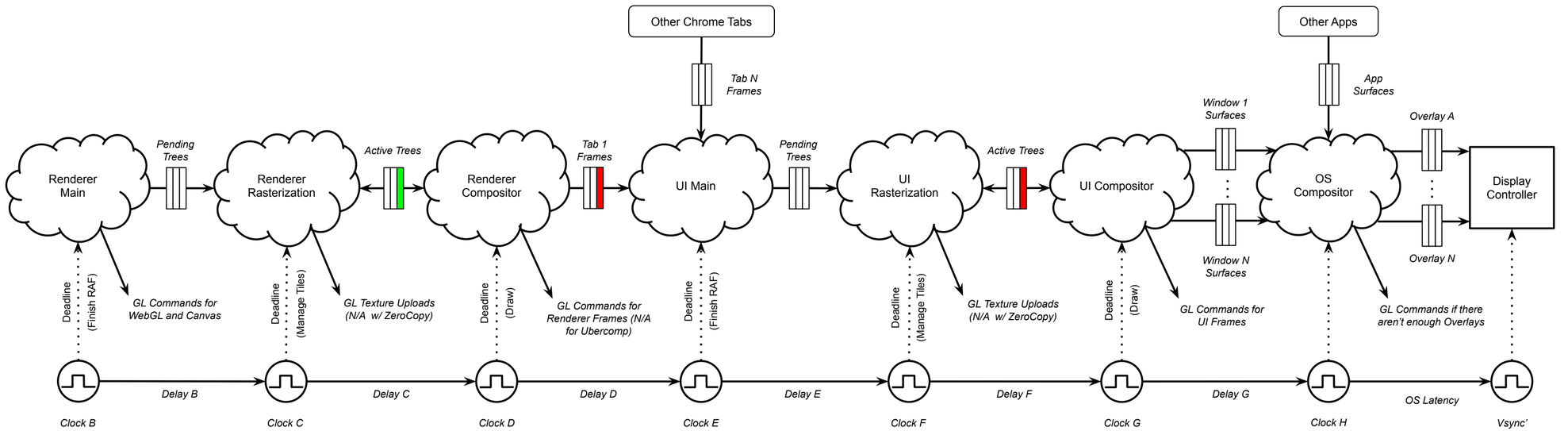
WebGL FB2: Rasterization hasn't completed. The Chrome GPU Service waits to submit this buffer's commands to the driver until notified.

Window1 FB1

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

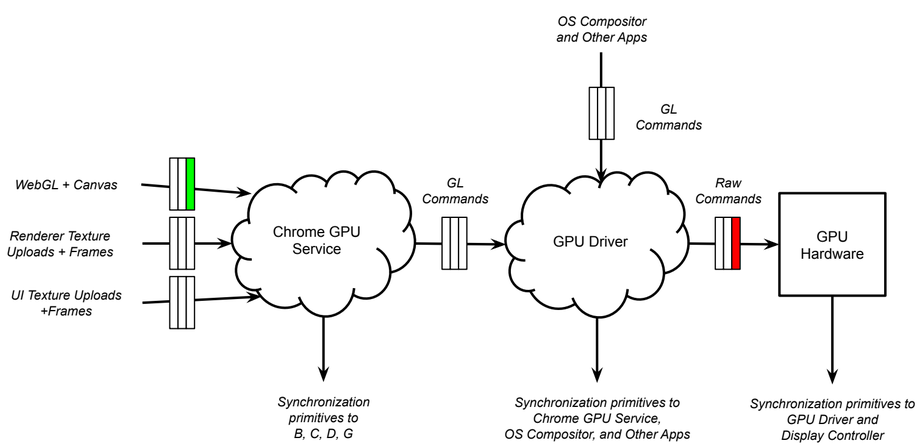


WebGL FB1: The Renderer is no longer referencing this surface because of the pending tree activation. The GPU hardware is still working to produce this buffer.

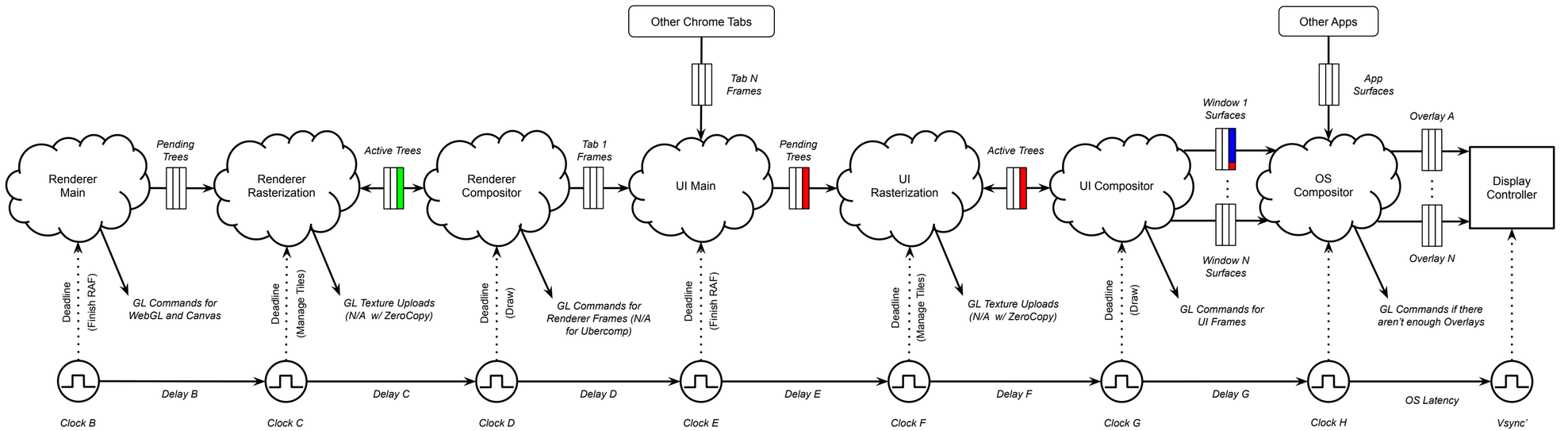
WebGL FB2: Rasterization completes and the pending tree is activated. The Chrome GPU Service waits to submit this buffer's commands to the driver until notified.

Window1 FB1

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

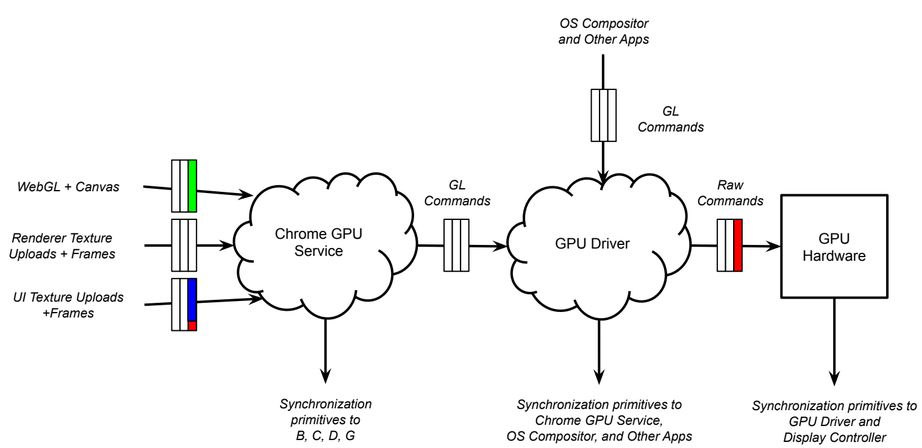


WebGL FB1: The Browser picks up the second frame referencing this buffer. The GPU hardware is still working to produce this buffer.

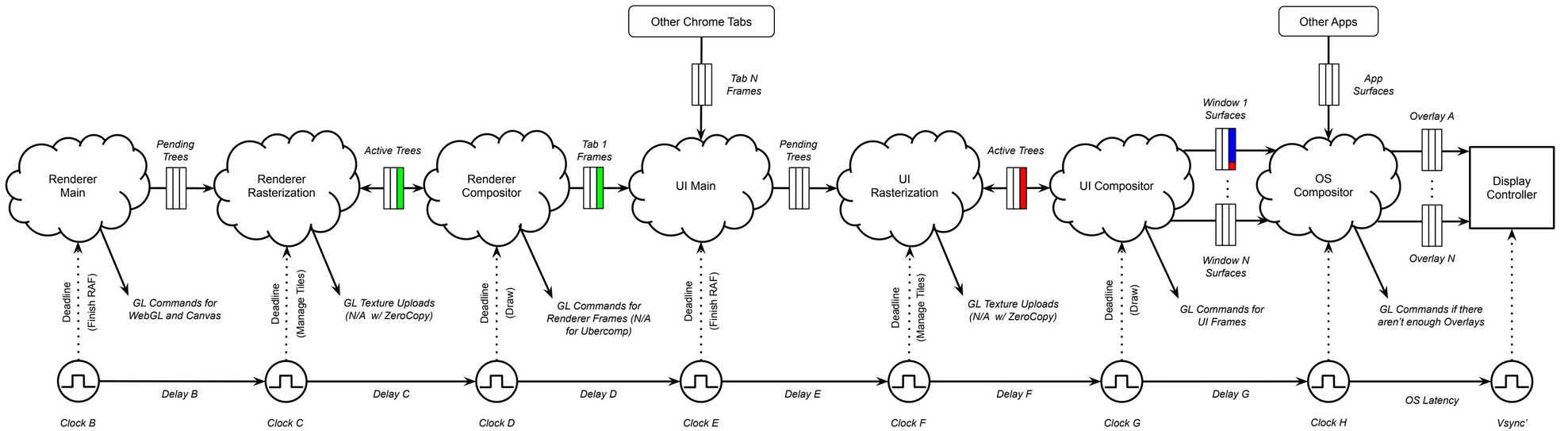
WebGL FB2: Rasterization completes and the pending tree is activated. The Chrome GPU Service waits to submit this frame to the driver until notified.

Window1 FB1: The Browser queues the commands to produce this buffer (with a reference to WebGL FB1) to the Chrome GPU Service and submits a reference to this buffer to the OS Compositor.

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

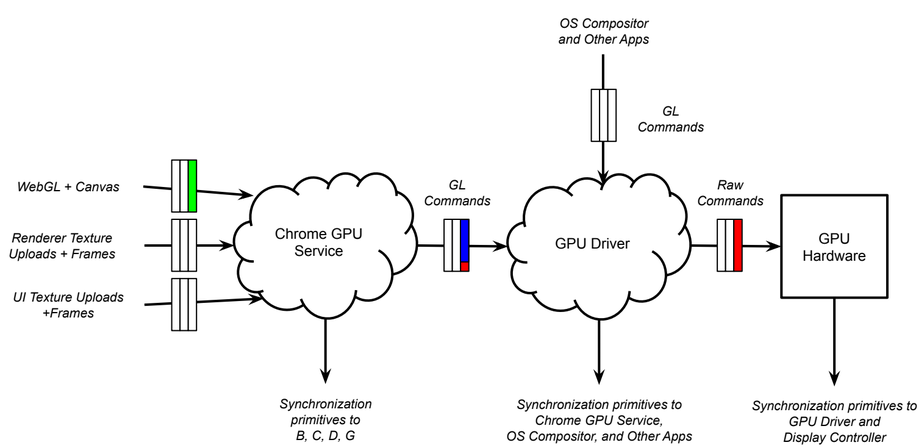


WebGL FB1: The UI activates the second frame referencing this buffer.

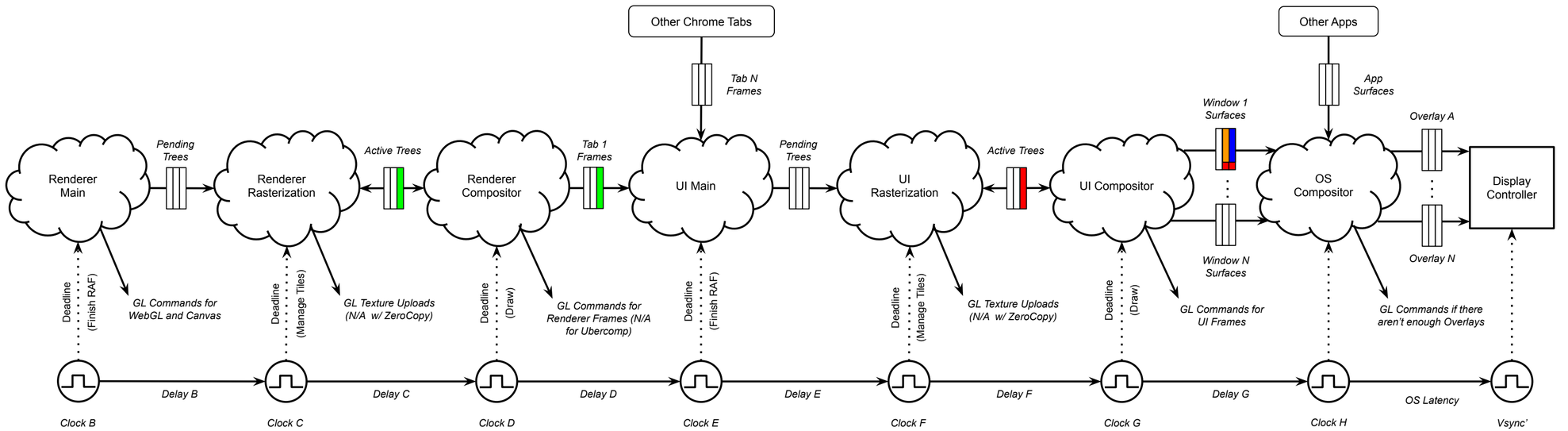
WebGL FB2: The Renderer submits the first frame referencing this buffer to the UI.

Window1 FB1: The commands to produce the buffer are sent to the GPU Driver.

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

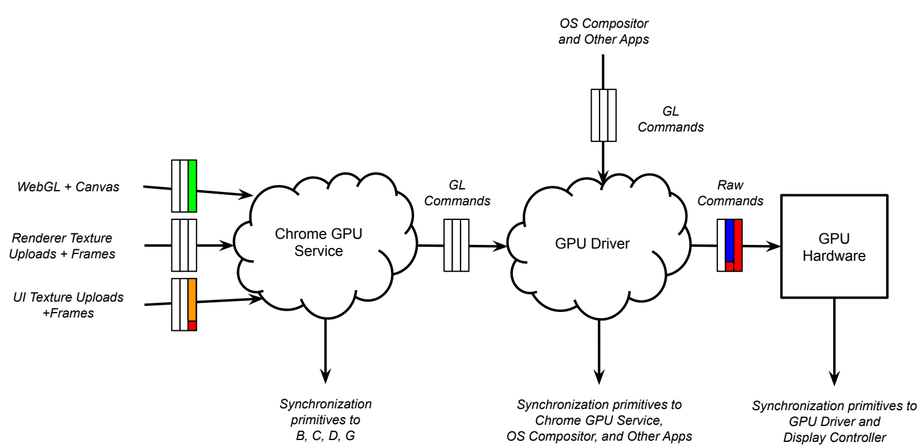


WebGL FB1

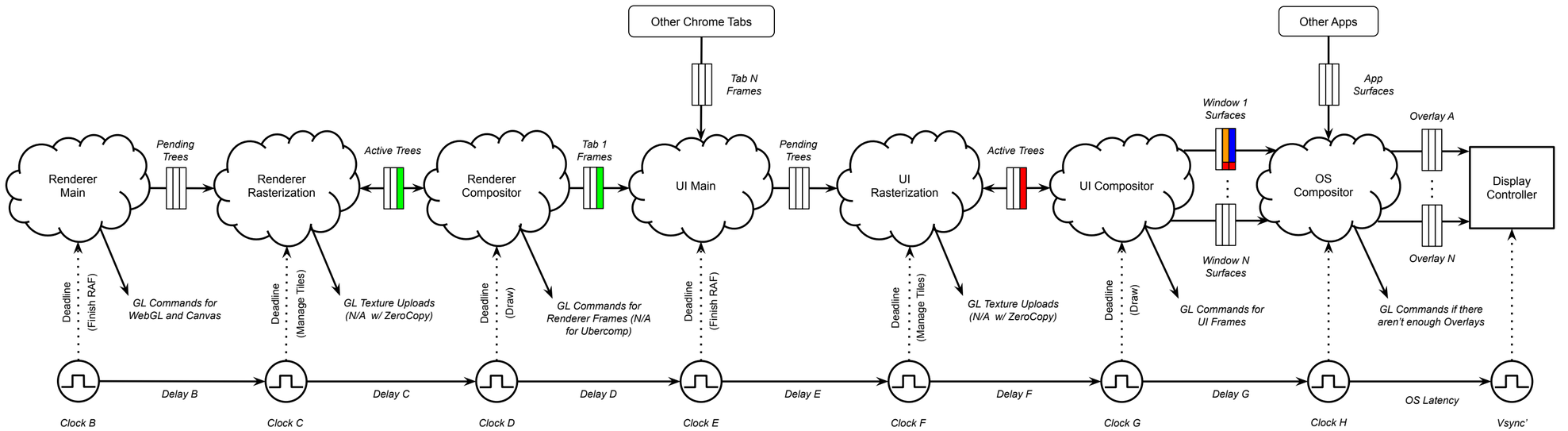
WebGL FB2

Window1 FB1: The GPU hardware is working to produce this buffer.

Window1 FB2: The commands to produce this buffer (with a reference to WebGL FB1) are queued to the Chrome GPU Service and a reference to the buffer is submitted to the OS Compositor.



Double Buffered WebGL + Impl-Side Painting + Ubercomp

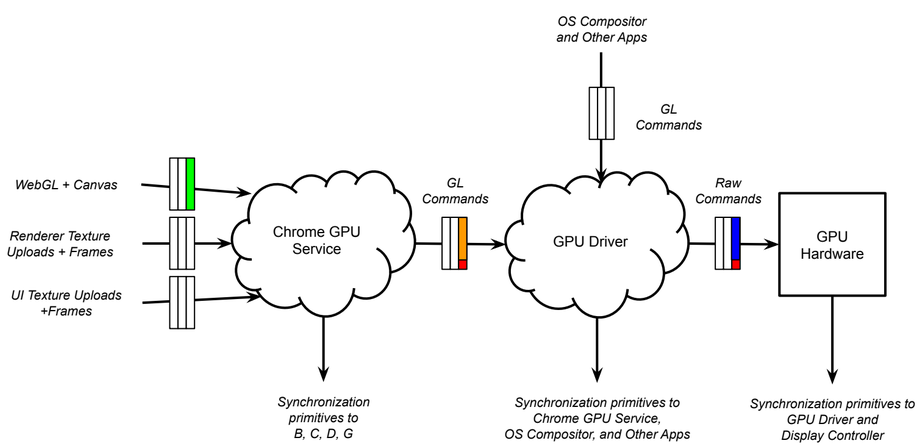


WebGL FB1: Production of this buffer is complete.

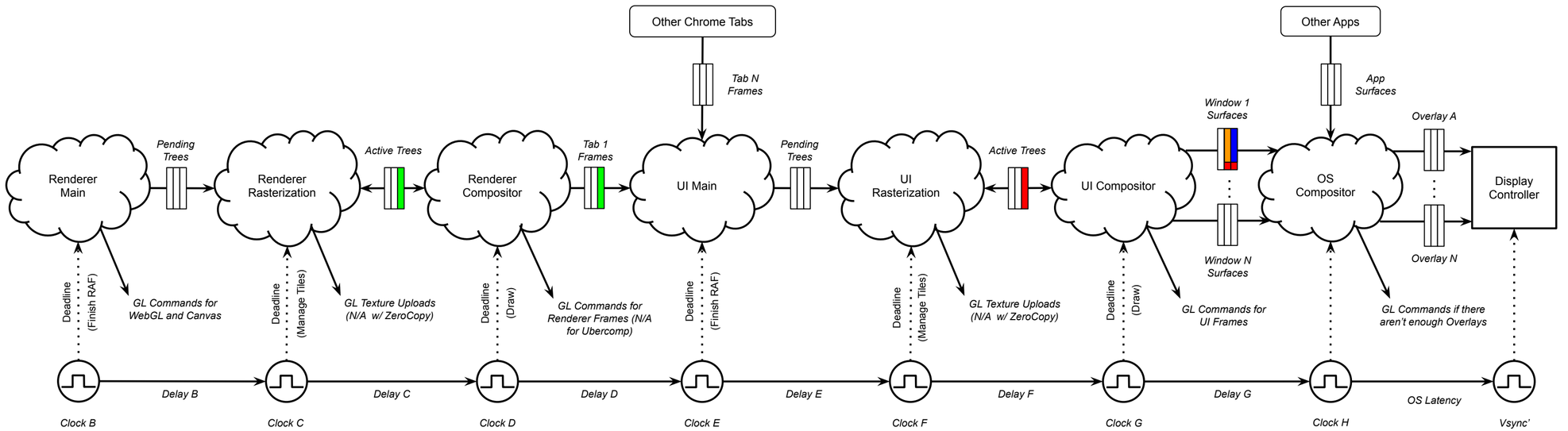
WebGL FB2

Window1 FB1: The GPU hardware is working to produce this buffer.

Window1 FB2: The commands to produce this buffer are sent to the GPU Driver.



Double Buffered WebGL + Impl-Side Painting + Ubercomp

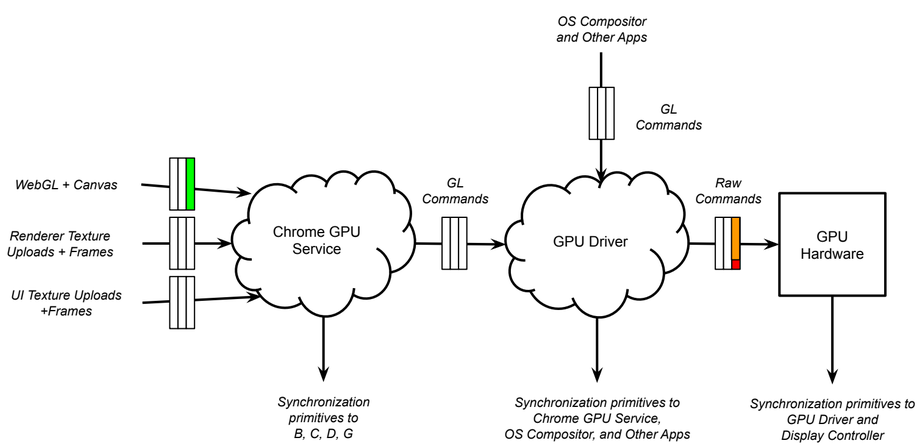


WebGL FB1

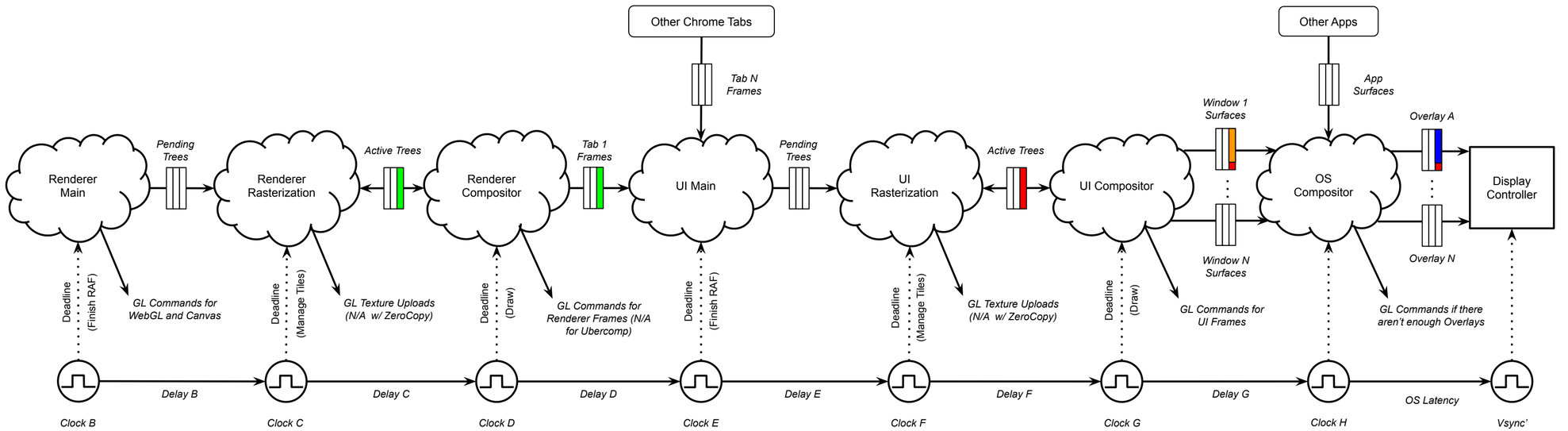
WebGL FB2

Window1 FB1: Production of this buffer is complete.

Window1 FB2: The GPU hardware starts working to produce this buffer.



Double Buffered WebGL + Impl-Side Painting + Ubercomp

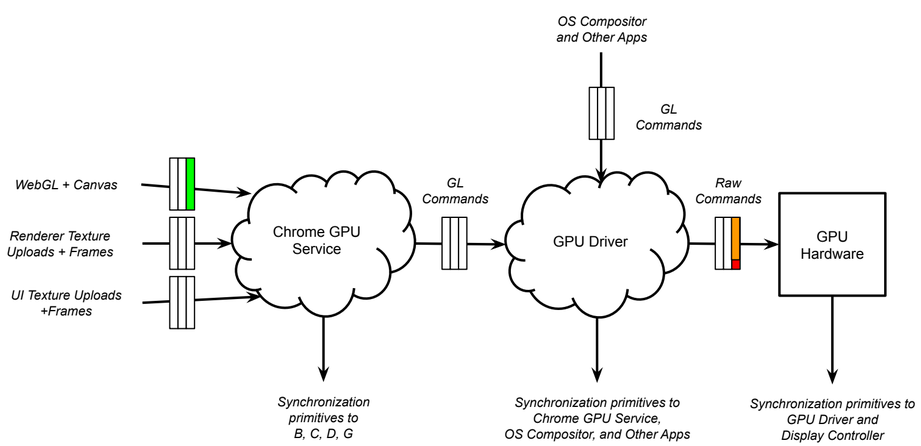


WebGL FB1

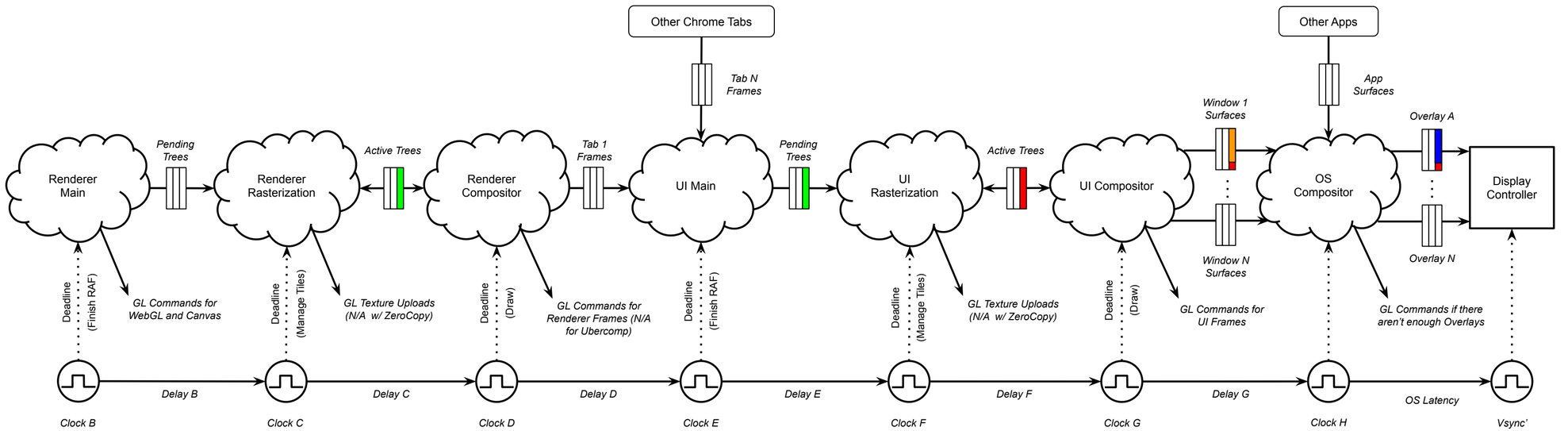
WebGL FB2

Window1 FB1: Now that the buffer is complete, the Display Controller can start displaying it.

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

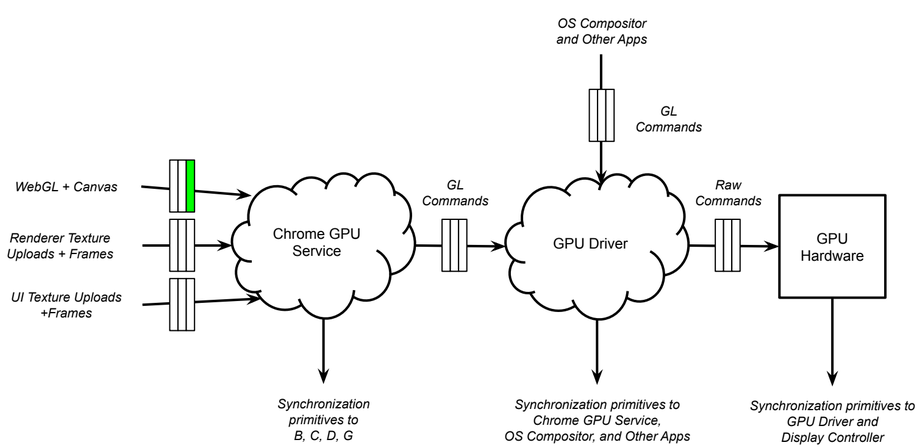


WebGL FB1

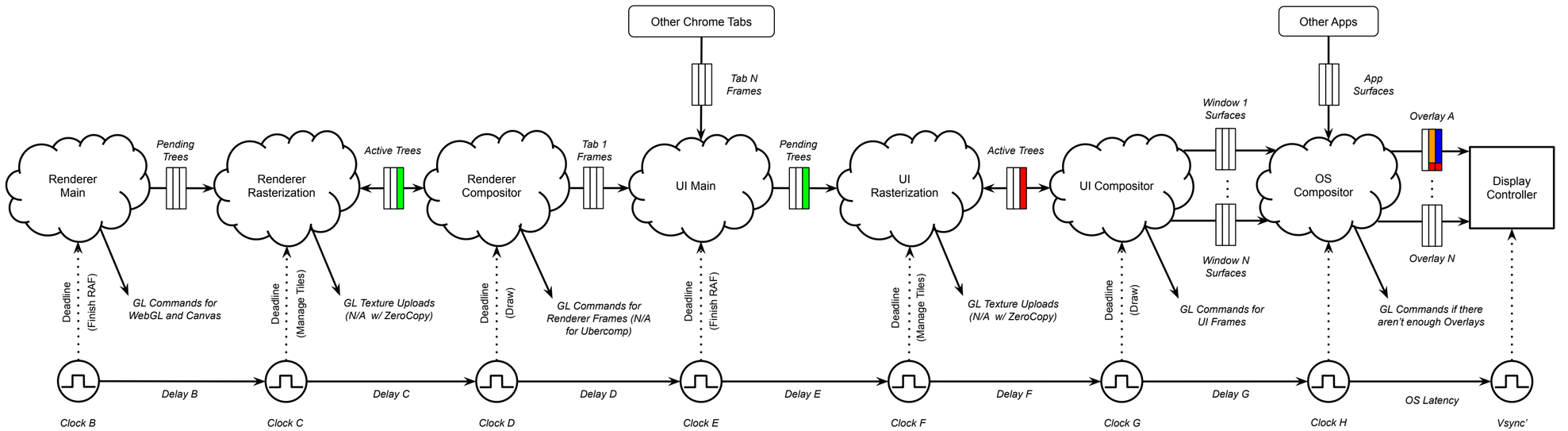
WebGL FB2: The UI picks up the first frame that references this buffer.

Window1 FB1

Window1 FB2: Production of this buffer is complete.



Double Buffered WebGL + Impl-Side Painting + Ubercomp

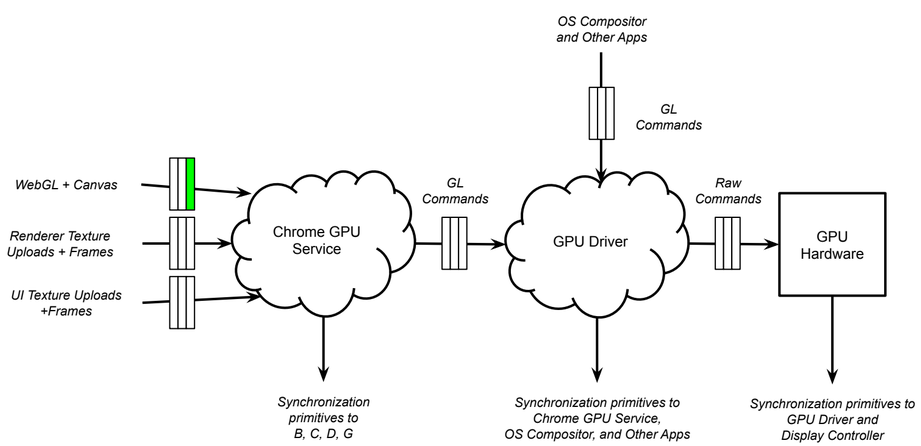


WebGL FB1

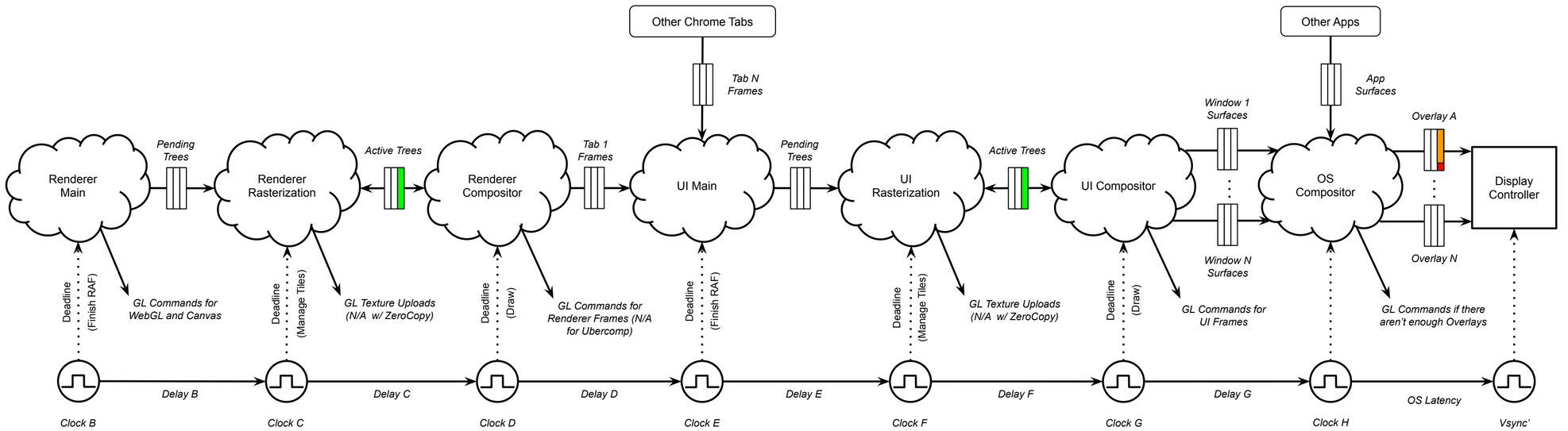
WebGL FB2: Waiting for activation in the UI.

Window1 FB1

Window1 FB2: Now that the buffer is complete, it can be queued to the Display Controller.



Double Buffered WebGL + Impl-Side Painting + Ubercomp

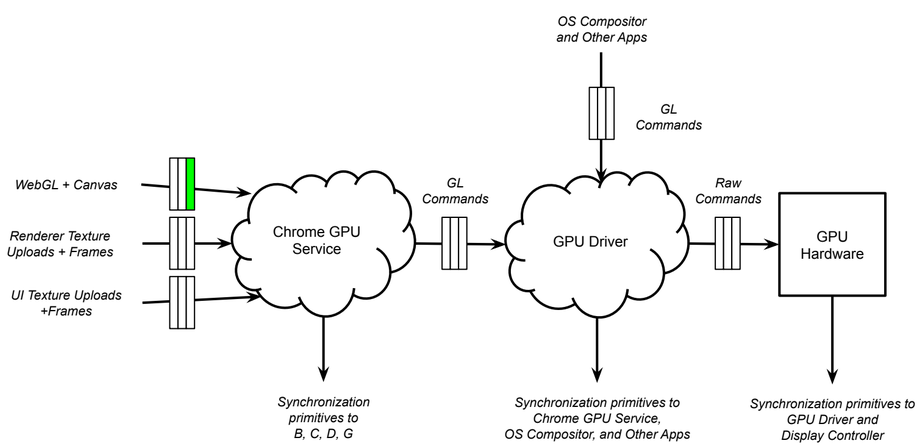


WebGL FB1

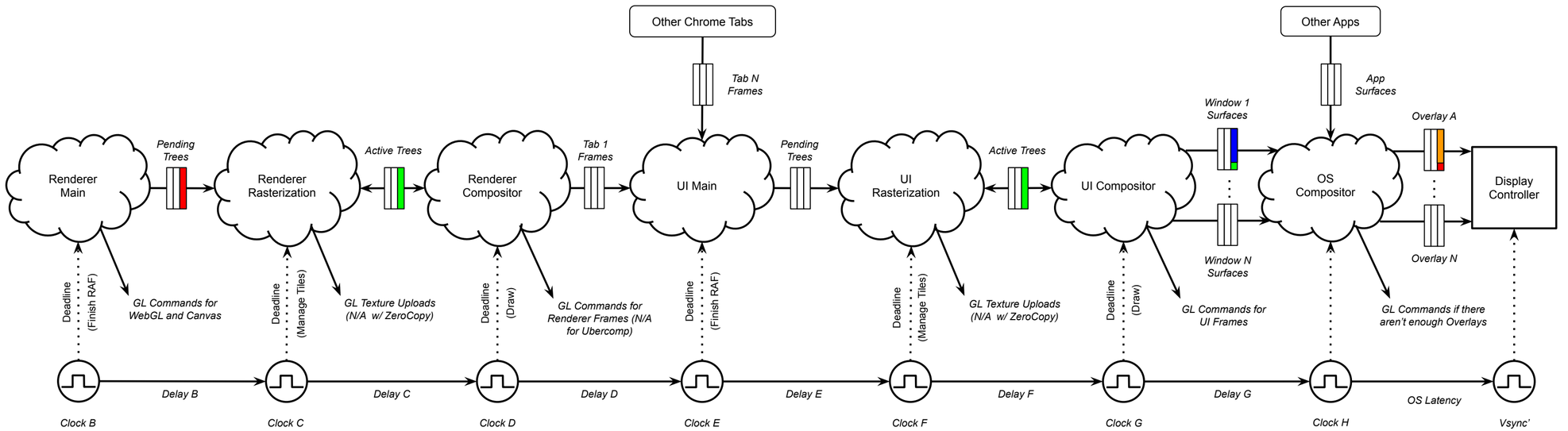
WebGL FB2: The UI activates the frame referencing this buffer.

Window1 FB1: The buffer is available again.

Window1 FB2: This buffer is displayed on the next vsync.



Double Buffered WebGL + Impl-Side Painting + Ubercomp

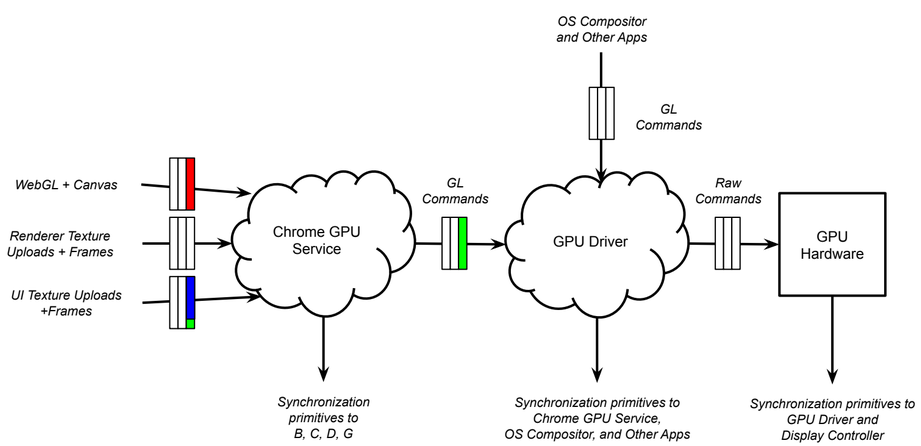


WebGL FB1: WebGL FB2's activation triggers the next RAF.

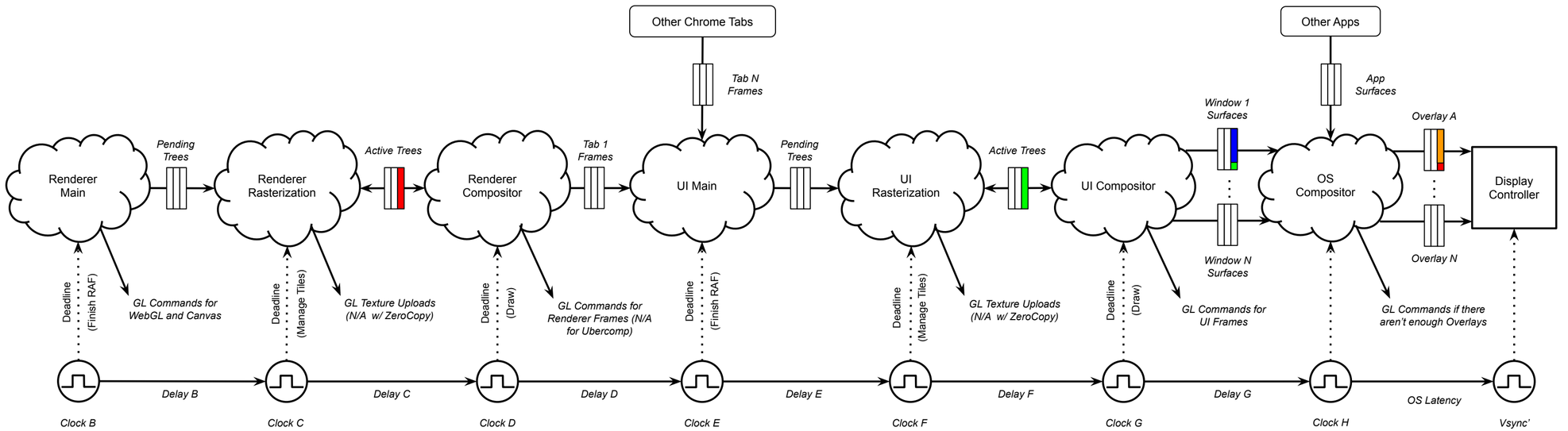
WebGL FB2: This buffer's activation triggers sending of the commands to the GPU Driver.

Window1 FB1: WebGL FB2's activation triggers the queuing of commands for this buffer to the ChromeGPU Service and a reference to the buffer submitted to the OS Compositor.

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

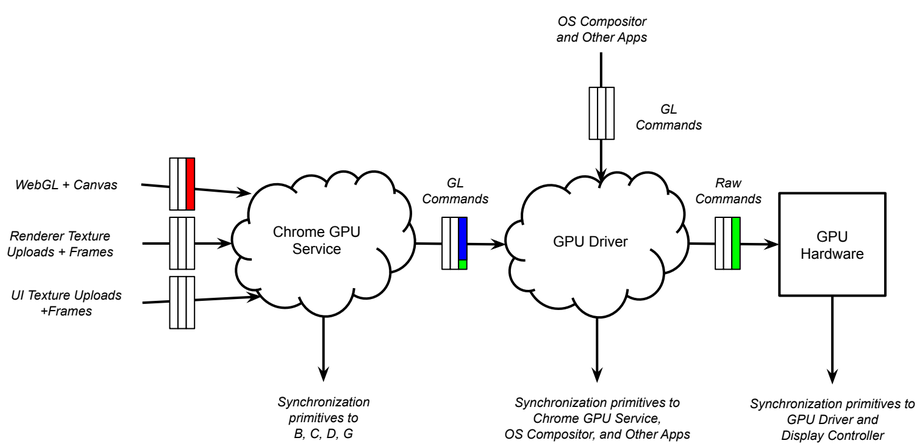


WebGL FB1: The Renderer activates the frame referencing this buffer.

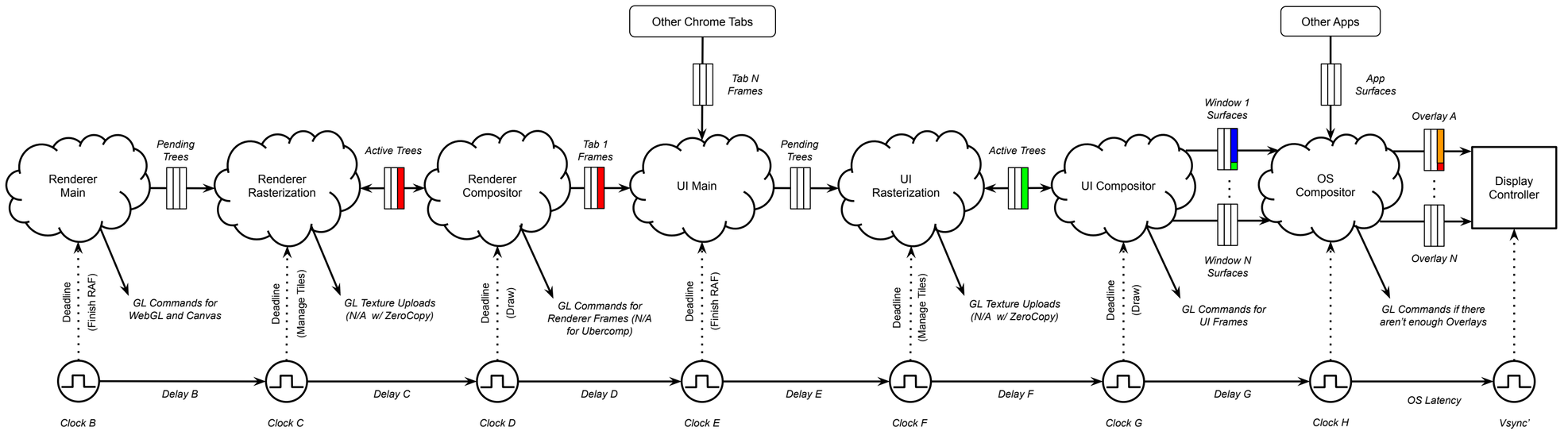
WebGL FB2: The GPU hardware is working to produce this buffer.

Window1 FB1: The Chrome GPU Service submits this frames commands to the GPU Driver.

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

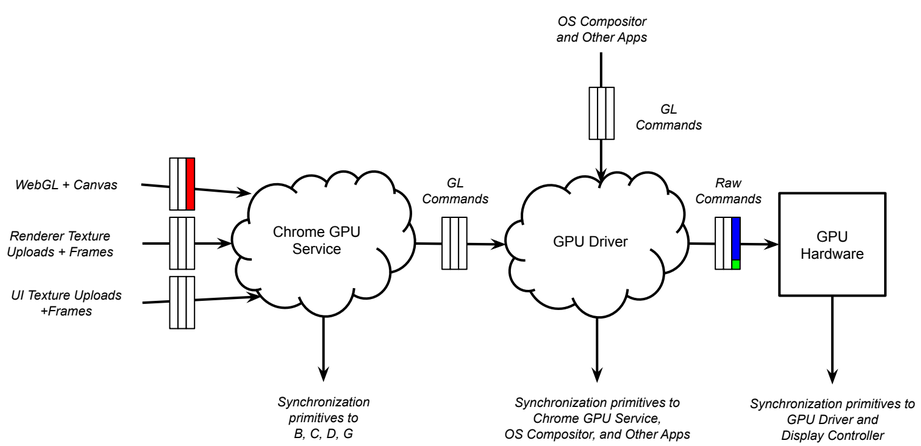


WebGL FB1: The Renderer submits the frame referencing this buffer to the UI.

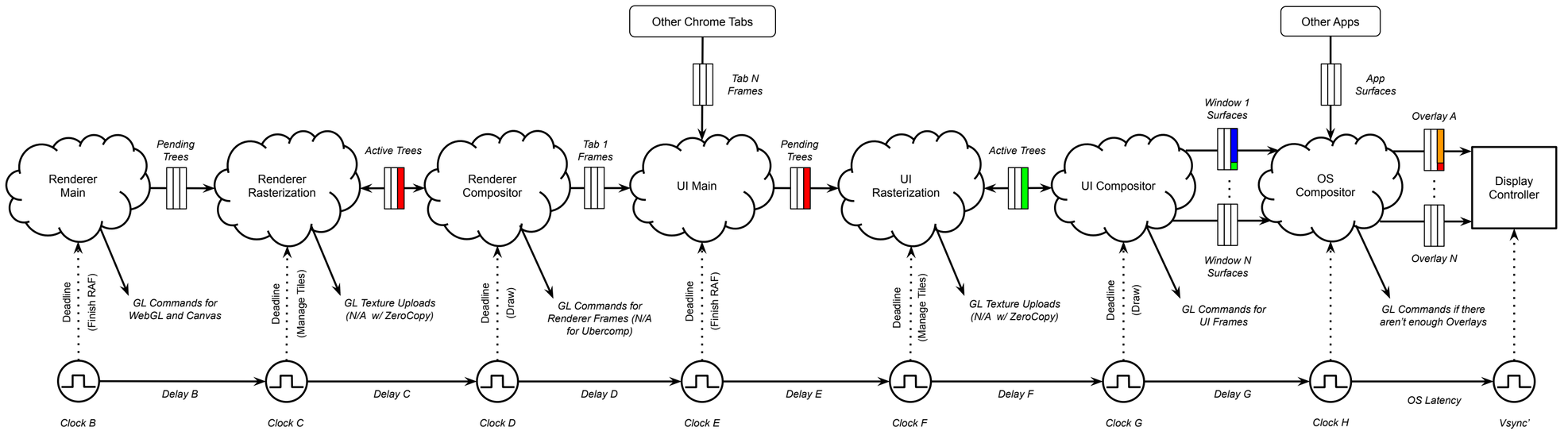
WebGL FB2: Production of this buffer is complete.

Window1 FB1: The GPU Hardware is working to produce this frame.

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp

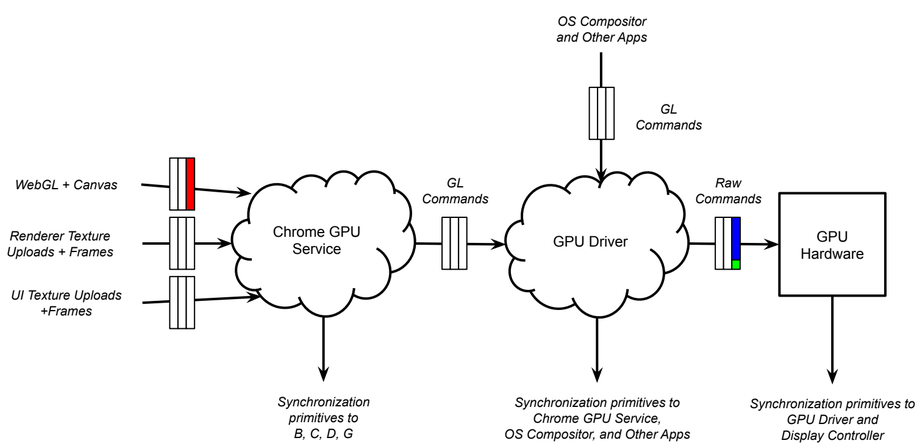


WebGL FB1: The Browser picks up the first frame referencing this buffer.

WebGL FB2:

Window1 FB1:

Window1 FB2



Double Buffered WebGL + Impl-Side Painting + Ubercomp