Android WebView

Chrome Graphics Meetup

boliu@ April 2019

android.webkit.WebView

A View in Android SDK to display web contents

Hundreds of APIs

Public constructors

`WebView(Context context)`

Constructs a new WebView with an Activity Context object.

`WebView(Context context, AttributeSet attrs)`

Constructs a new WebView with layout parameters.

`WebView(Context context, AttributeSet attrs, int defStyleAttr)`

Constructs a new WebView with layout parameters and a default style.

`WebView(Context context, AttributeSet attrs, int defStyleAttr, int defStyleRes)`

Constructs a new WebView with layout parameters and a default style.

`WebView(Context context, AttributeSet attrs, int defStyleAttr, boolean privateBrowsing)`

This constructor is deprecated. Private browsing is no longer supported directly via WebView and will be removed in a future release. Prefer using `WebSettings`, `WebViewDatabase`, `CookieManager` and `WebStorage` for fine-grained control of privacy data.

Public methods

`void`	`addJavascriptInterface(Object object, String name)` Injects the supplied Java object into this WebView.
`void`	`autofill(SparseArray<AutofillValue> values)` Automatically fills the content of the virtual children within this view.
`boolean`	`canGoBack()` Gets whether this WebView has a back history item.
`boolean`	`canGoBackOrForward(int steps)` Gets whether the page can go back or forward the given number of steps.
`boolean`	`canGoForward()` Gets whether this WebView has a forward history item.

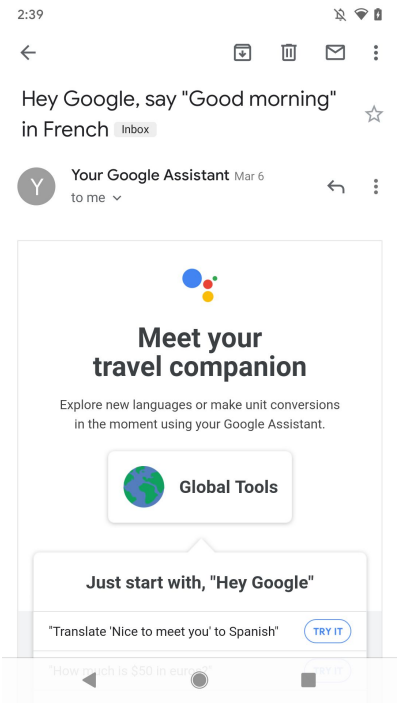
Sixth platform

Blink's intent-to-implement / intent-to-ship template

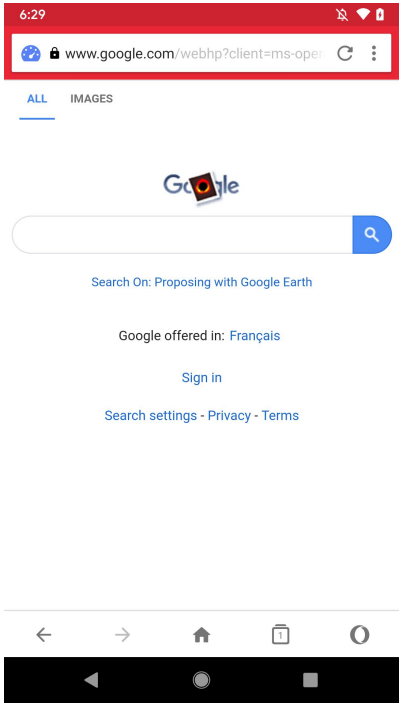
Is this feature supported on all six Blink platforms (Windows, Mac, Linux, Chrome OS, Android, and Android WebView)?

Yes or no. If no, explain why this feature is not supported on these platforms.

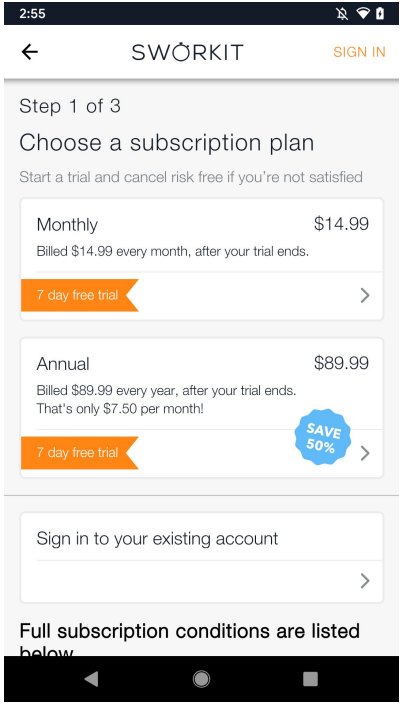
Primary use cases



Embedded into app



Browsers

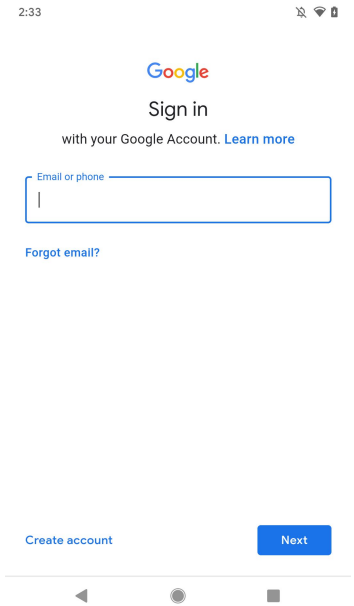


Cordova

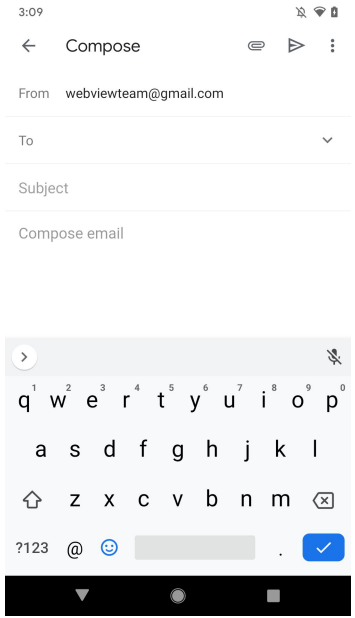
Screenshot missing intentionally

Ads

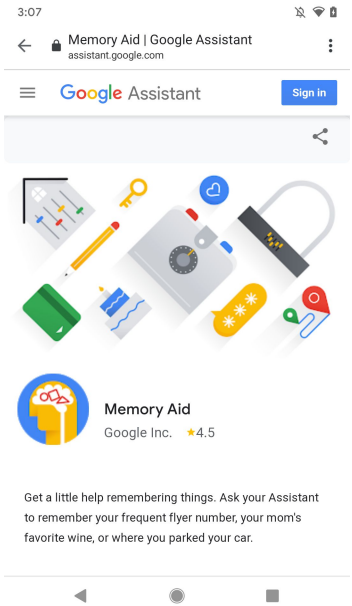
WebView or not?



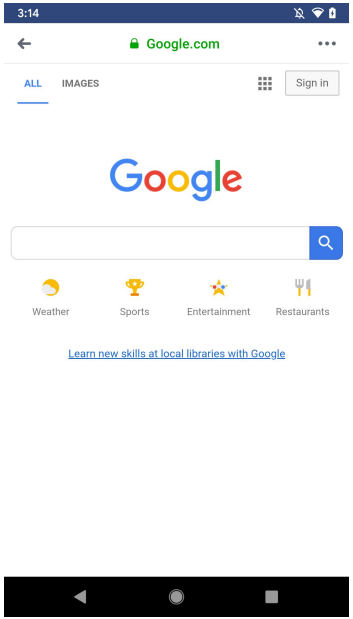
Add Google account on Android



Gmail email compose

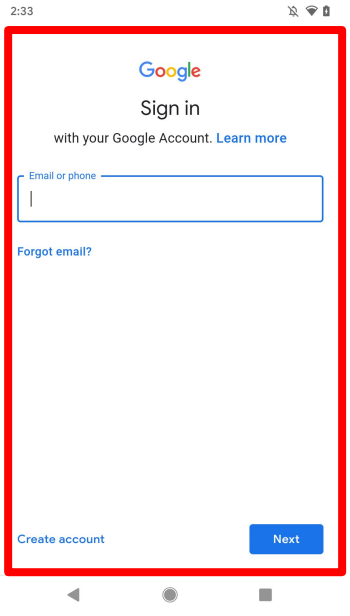


Clicking on a link in Gmail

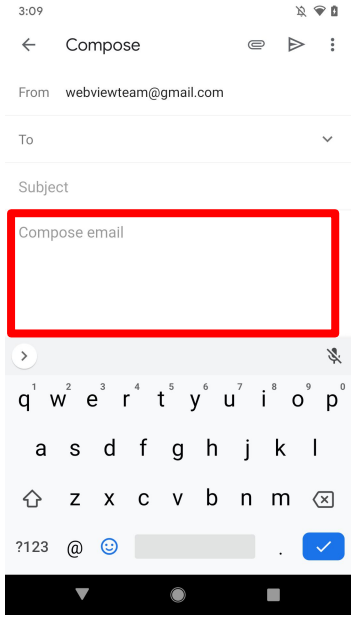


Clicking on a link in Facebook app

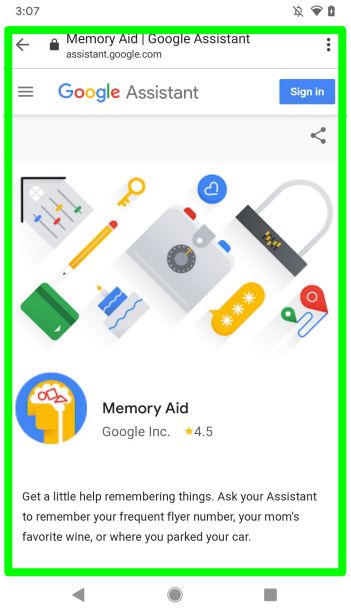
WebView or not?



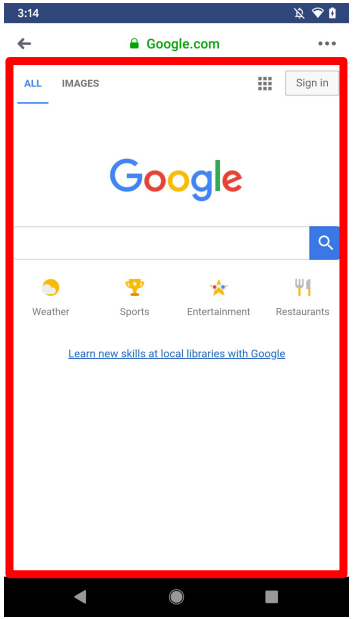
Add Google account on Android



Gmail email compose



Clicking on a link in Gmail



Clicking on a link in Facebook app

WebView’s user base is probably on-par with Chrome on Android

Development background

Browser process is the app process

- Shares code, address space, etc in the same process
- App is “trusted” and has total control over web contents

Each app is ~equivalent to a different installation of chrome (different data dir)

- Does not share cookies or any other profile data with chrome

Each webview is ~equivalent to a tab in chrome

- no multi-profile support

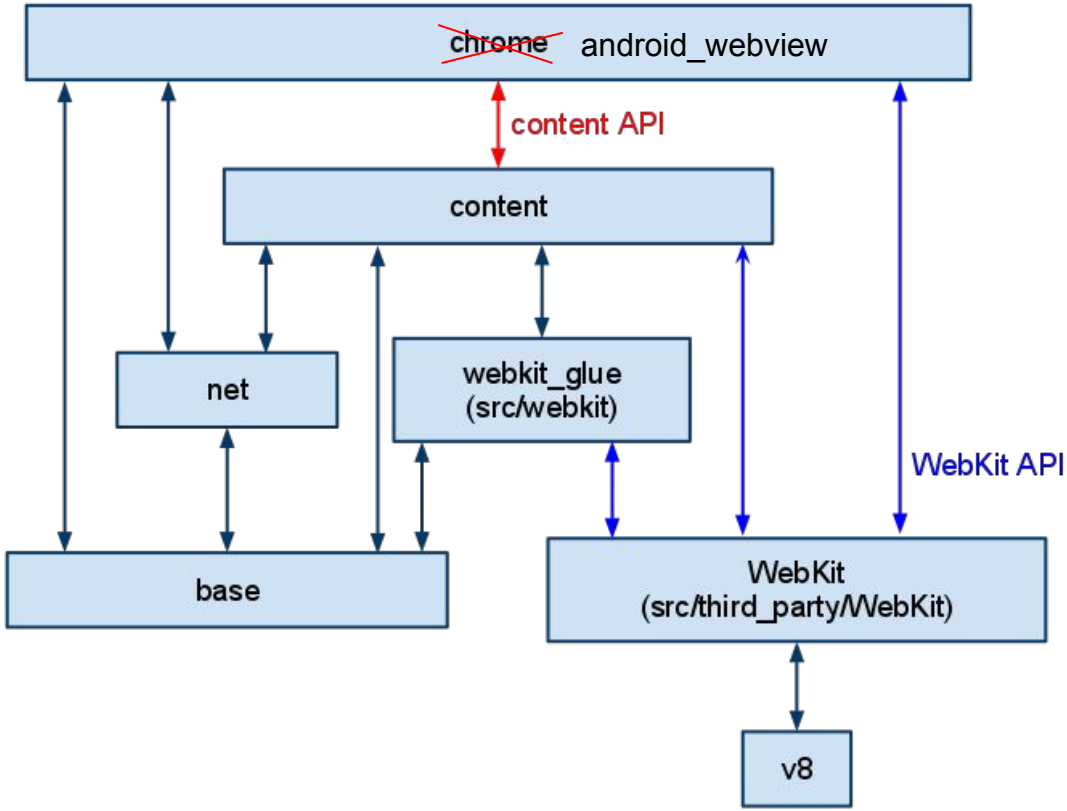
Cannot talk directly to Google services

Source layout

Use OS_ANDROID. No separate compile-time define

Includes src/content and below, but not src/chrome

Rest of diagram fairly out of date ->



Release

Identical release as Chrome on Android

Same version pushed to play store at same time. Same version bundled in Android OS system image

- Respins for one requires respinning both

Supports the same canary, dev, beta, stable channels

- On Android N+
- Population is a lot smaller than Chrome on Android, which is already a lot smaller than desktop

Developer workflow

Historically different (confusing) packaging depending on Android version.
Different build targets is confusing

Webview apk only implements webview APIs

- Also need an embedding app to test.

On Android N+, workflow *almost* as easy as Chrome on Android

- Just need an emulator or a userdebug device
- https://chromium.googlesource.com/chromium/src/+/HEAD/android_webview/docs/quick-start.md

Supports UMA metric collection and Finch experiments

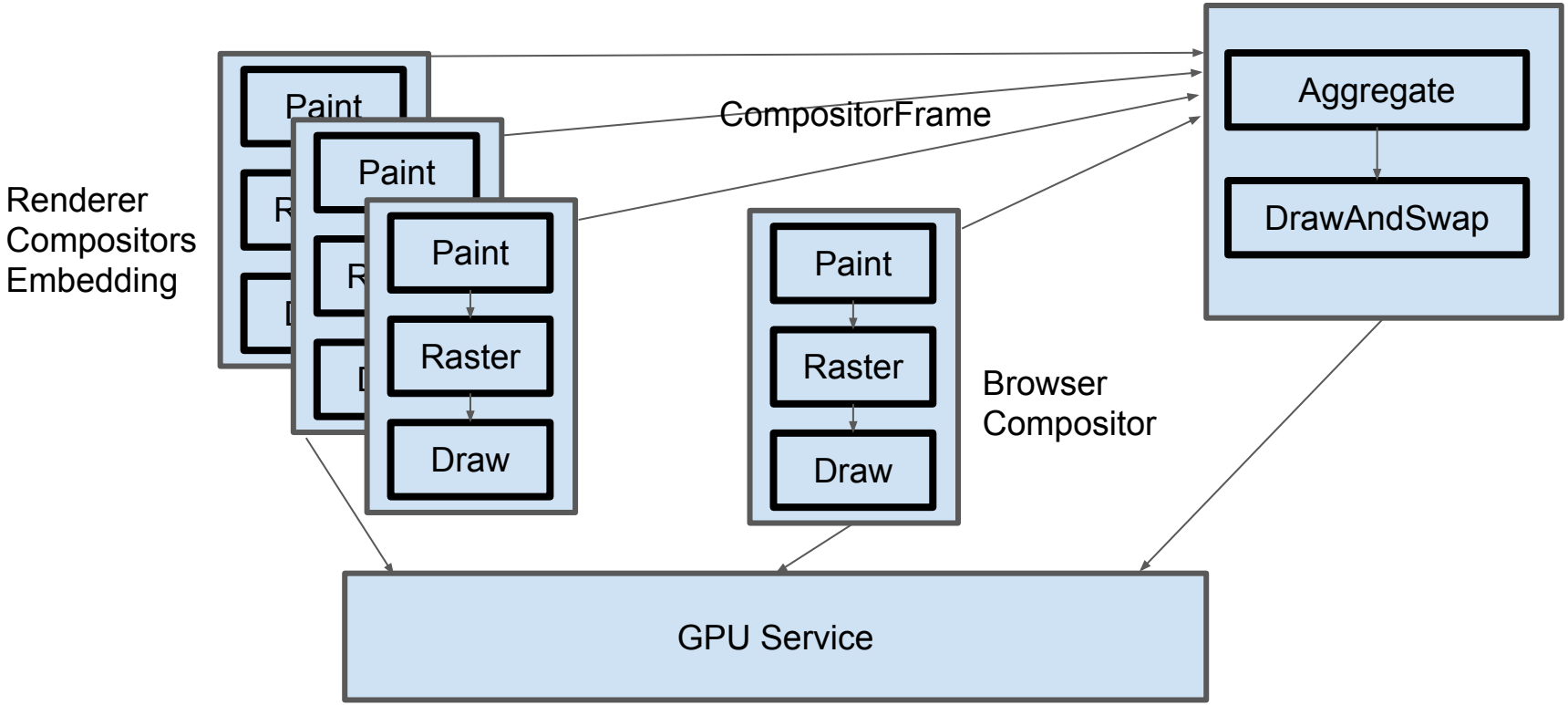
WebView is an Android View

A layer in Android's compositor (hwui)

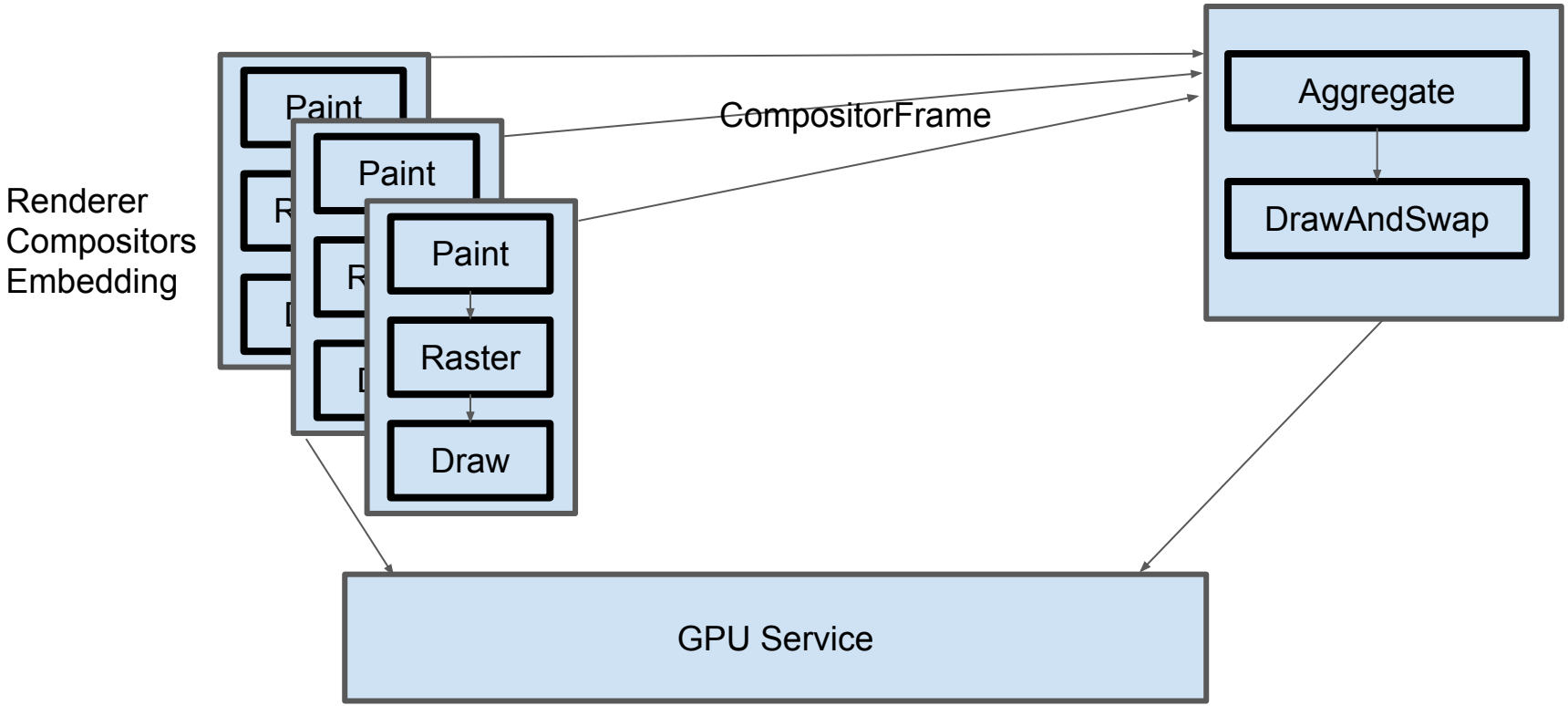
- Analogous to a cc::Layer in chromium compositor
- App or hwui has a control over scheduling, scrolling, etc
- Lots of synchronous (ie blocking) operations
- Hwui owns surface webview renders into

WebView connects chromium's compositing pieces in different ways to satisfy these requirements

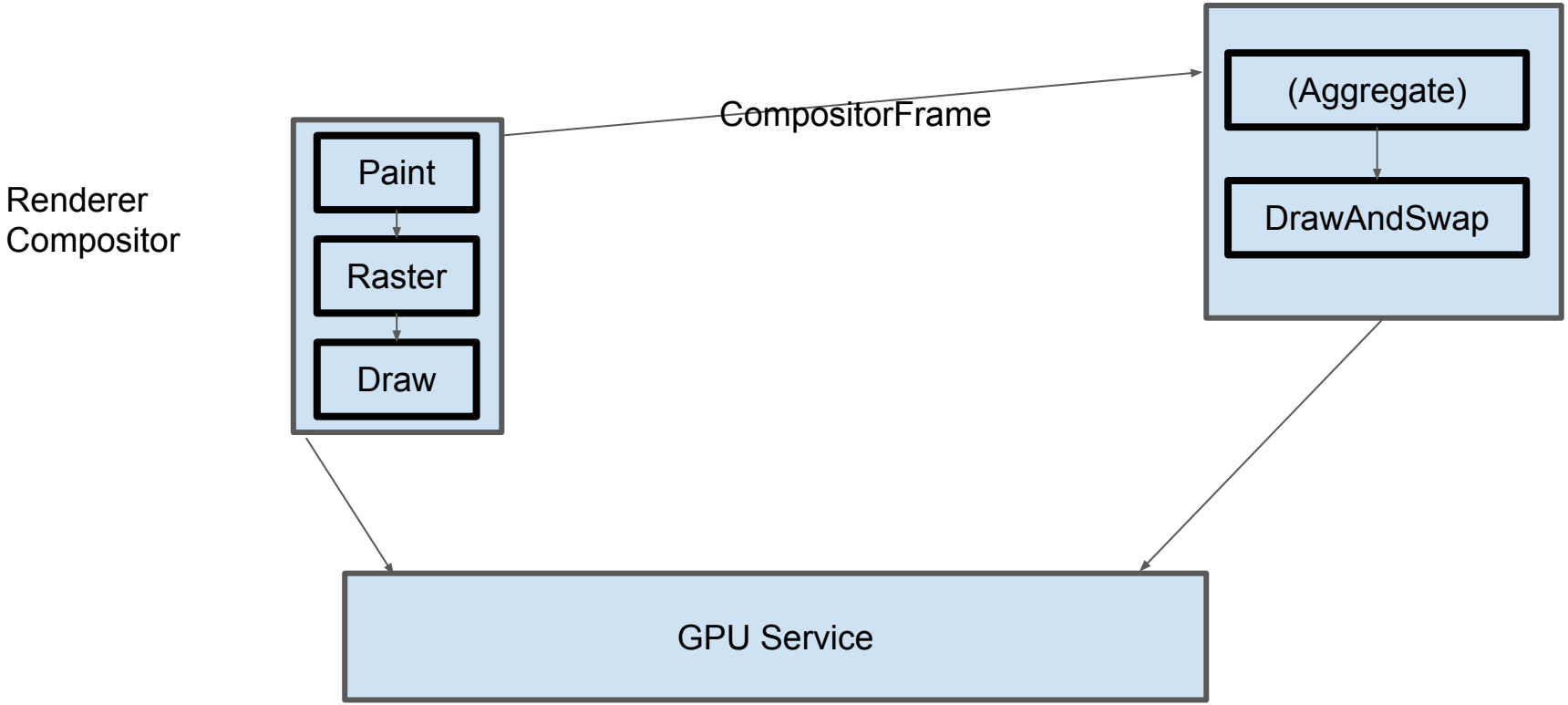
Chrome’s architecture (grossly simplified)



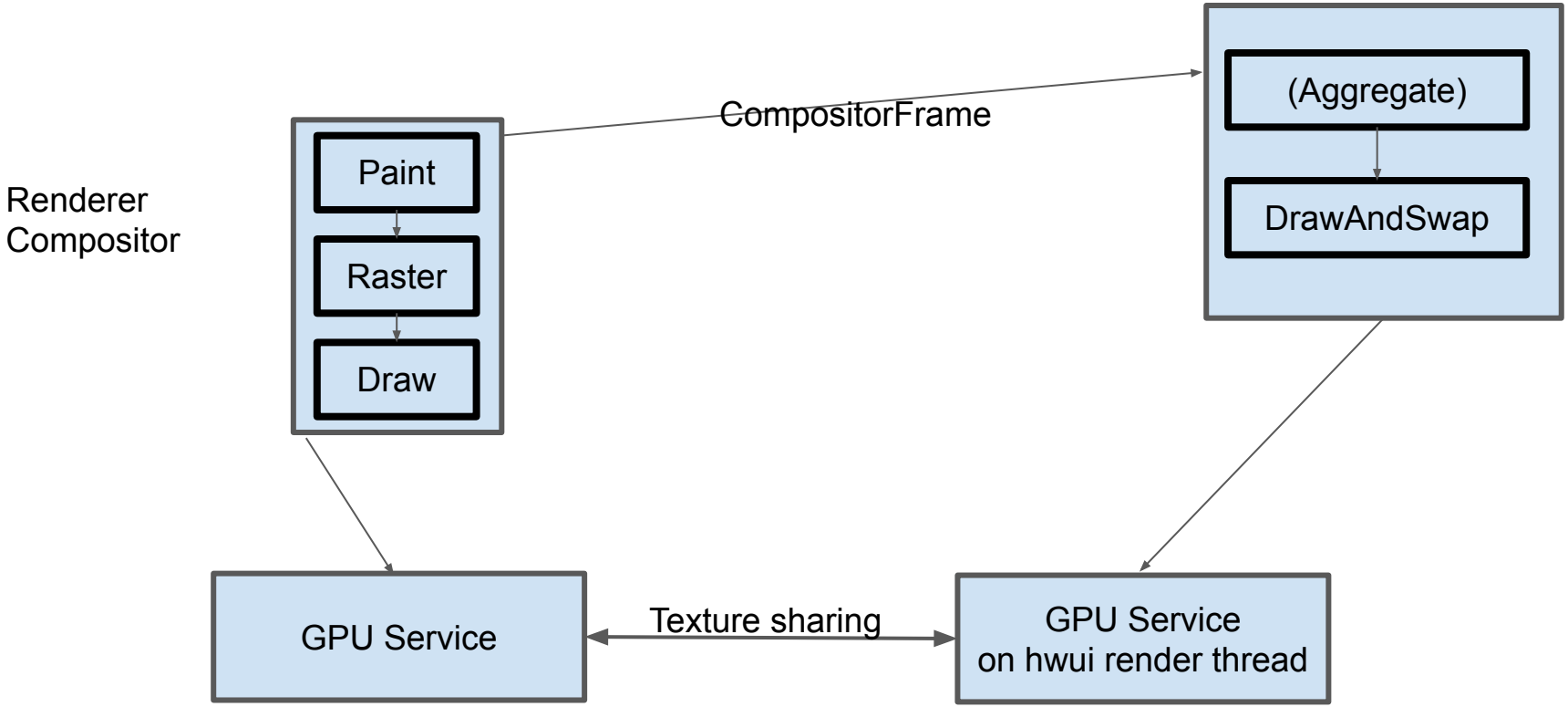
WebView: No browser compositor



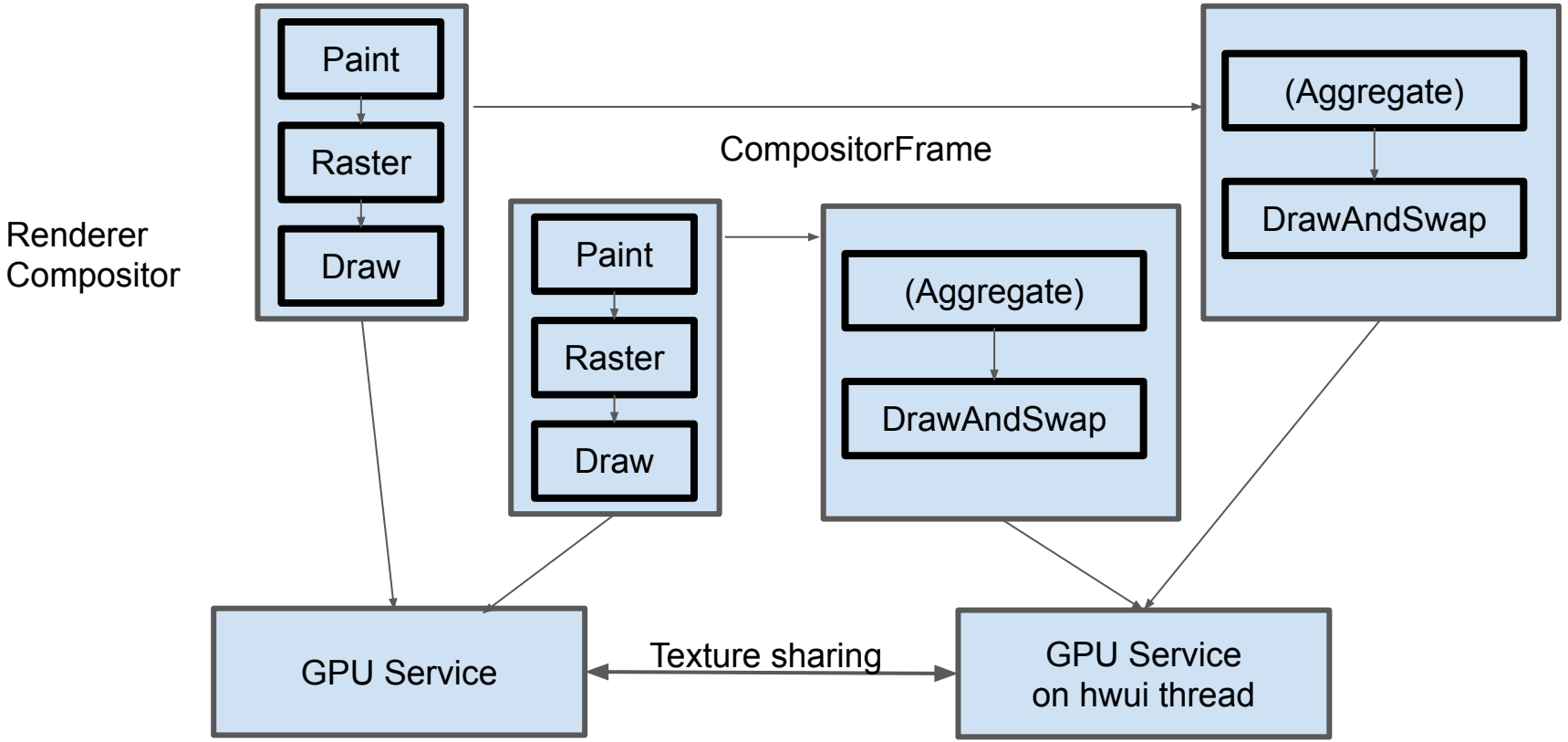
WebView: No embedding support, yet



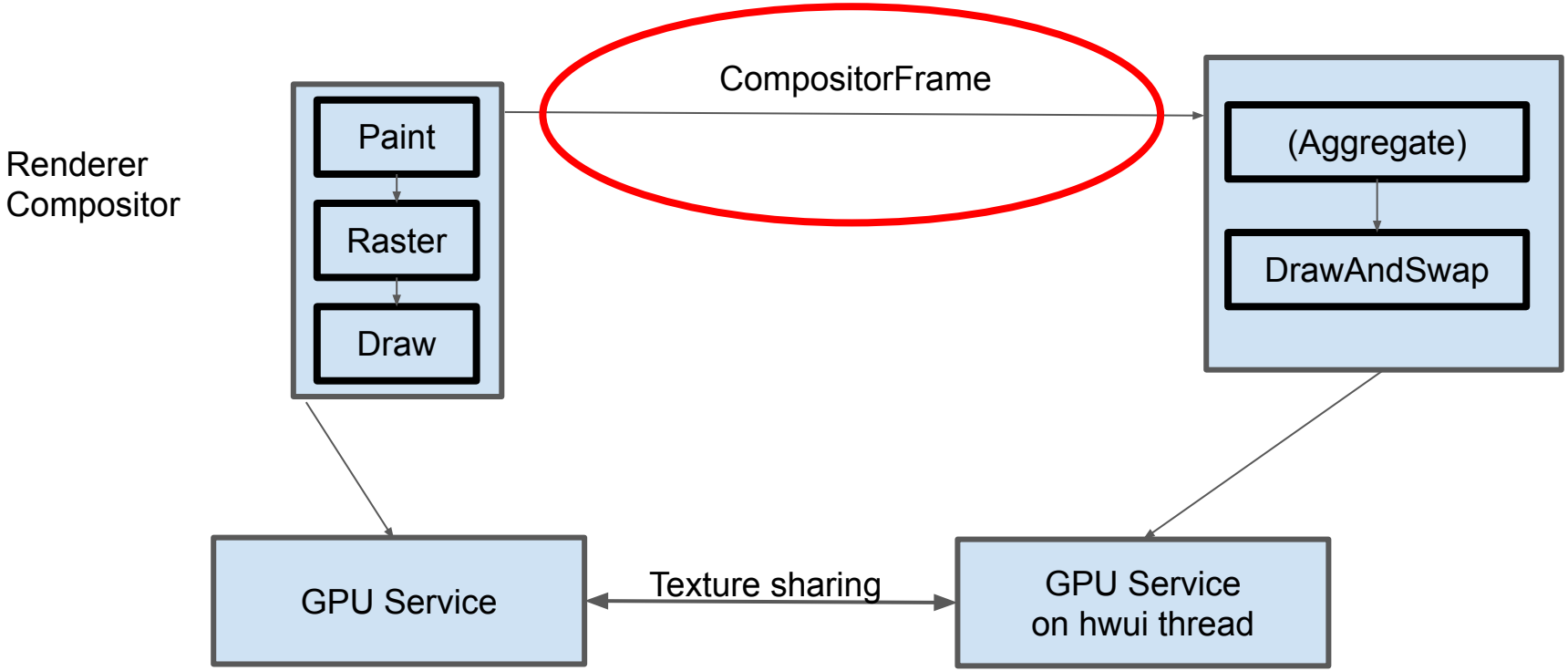
WebView: Two GPU services



WebView: Separate viz::Display, conceptually



WebView: SynchronousCompositor



Other rendering differences

- Hardware overlays are disabled
- Resourceless software draw
- Paint whole document (in some mode)
- Assumptions about viewport size often wrong

Please remember WebView for your features :)