Android WebView 101

Chrome University 2019
ntfschr@ (he/him)

(based on a previous talk by boliu@)

Bring the best of Chromium to the Android platform

Sixth platform

Blink’s intent-to-implement / intent-to-ship template:

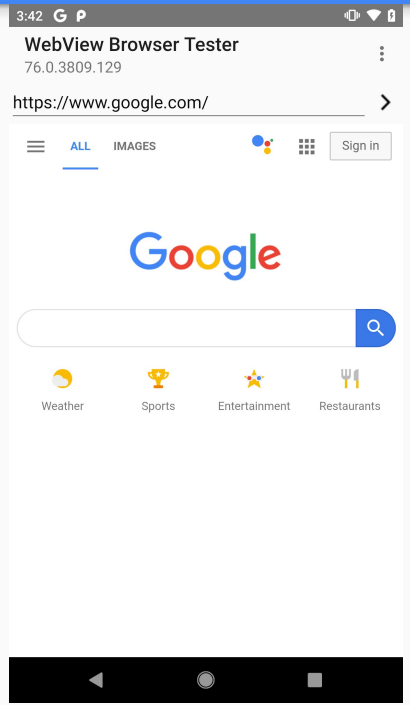
*Is this feature supported on all six Blink platforms (Windows, Mac, Linux, Chrome OS, Android, and **Android WebView**)?*

What is it?

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.

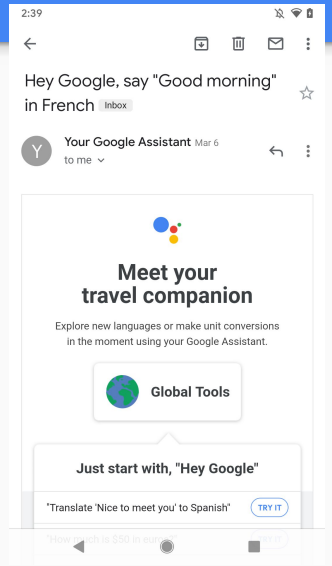
`WebView(Context context, AttributeSet attrs, int defStyleAttr, boolean privateBrowsing)`

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.

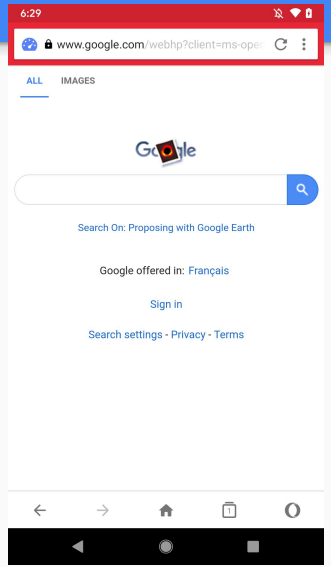
Public methods

`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.

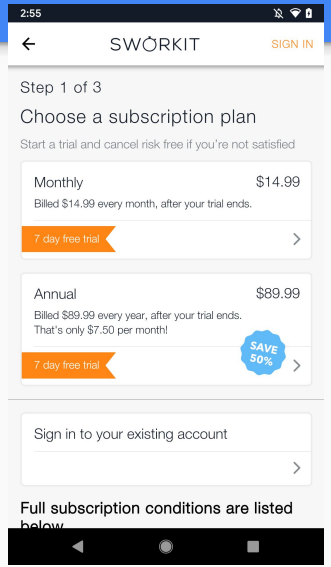
Primary use cases



Embedded into app



Browsers

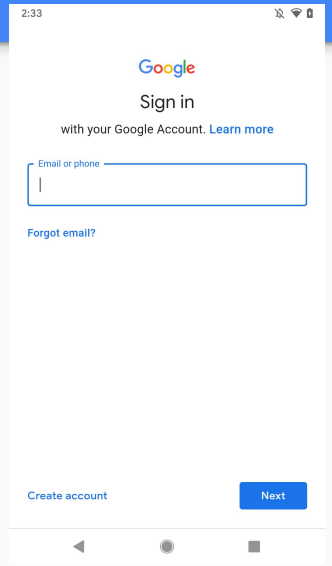


Cordova

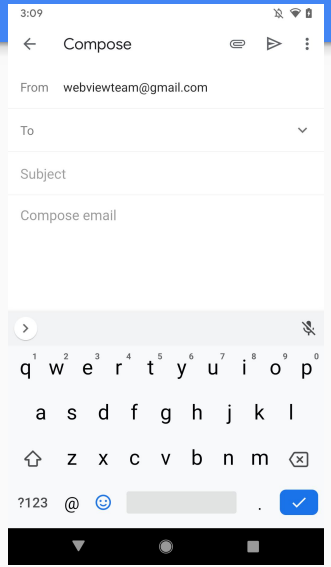
Screenshot missing intentionally

Ads

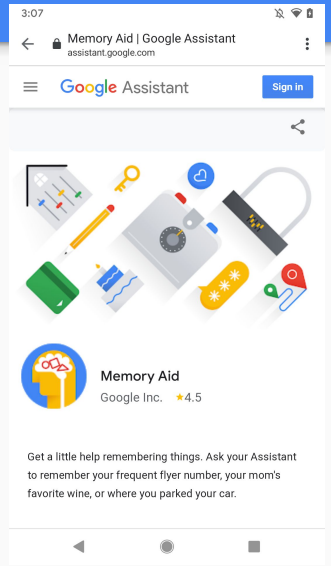
WebView or not?



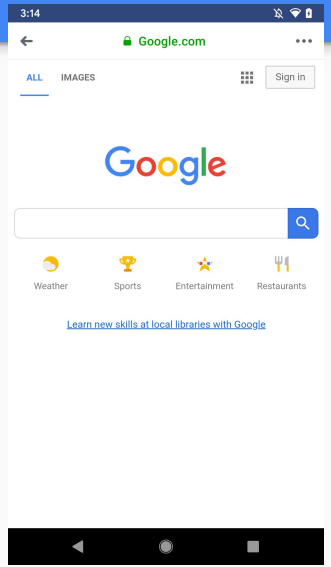
Add Google account on Android



Gmail email compose

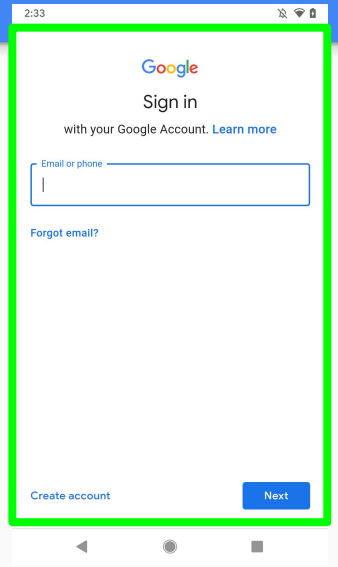


Clicking on a link in Gmail

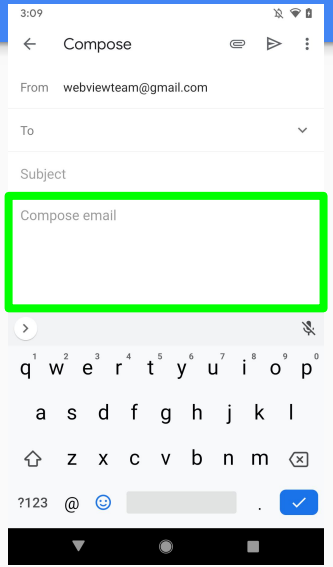


Clicking on a link in Facebook app

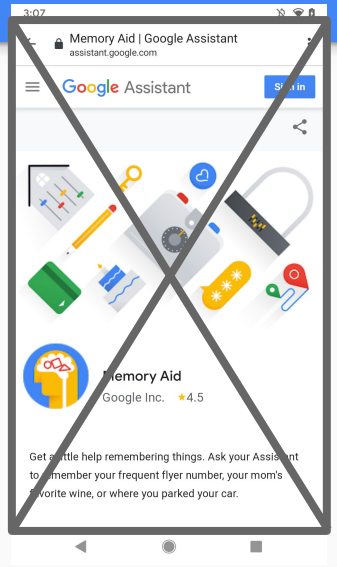
WebView or not?



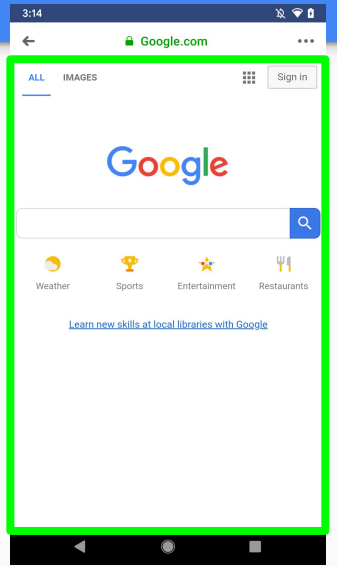
Add Google account on Android



Gmail email compose



Clicking on a link in Gmail



Clicking on a link in Facebook app

WebView is used by apps on every Android device worldwide

What can it do?

loadUrl() ~ navigate in omnibox

onPageStarted/onPageFinished() callbacks ~ update omnibox while navigating

onReceivedError()/onReceivedHttpError() callbacks ~ error page

What can it do?

evaluateJavascript()/@JavascriptInterface ~ interact with web content

setCookie()/getCookie() ~ interact directly with cookie database

shouldInterceptRequest() ~ modify or intercept arbitrary network requests

Technical guts

Trust model

Browser process is the app process

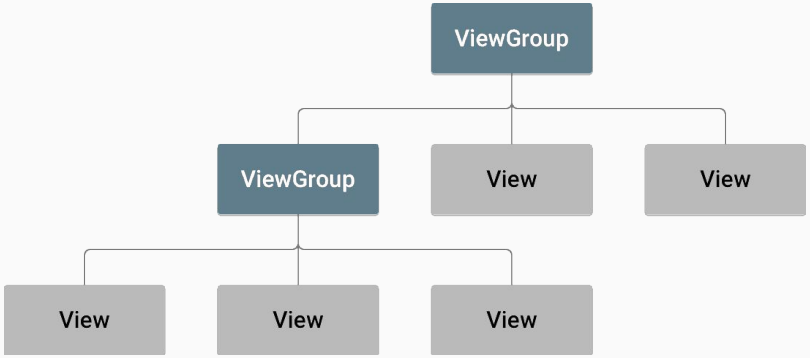
- Shares code, address space, etc. in the same process
- App is “trusted” and has total control over web contents

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

WebView is an Android View



Draws synchronously into View system (lots of blocking graphics operations)

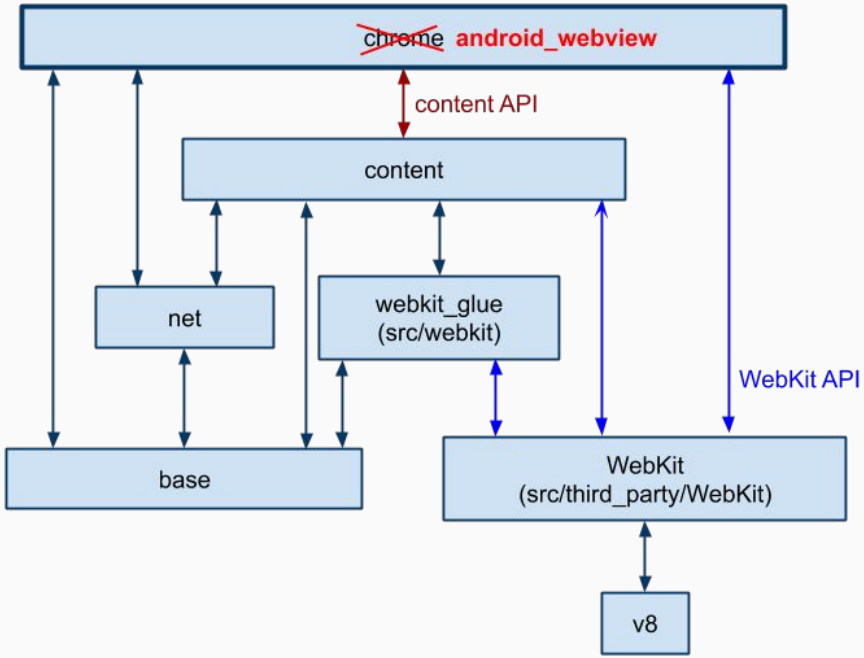
Supports Android platform features (smart text selection, autofill)

APIs to control web platform (ex. ServiceWorkerController)

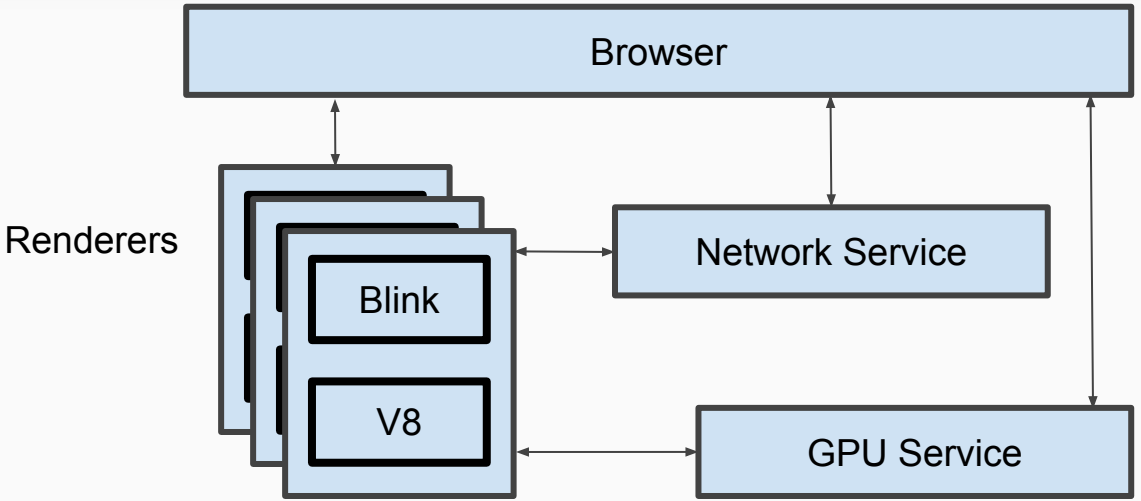
WebView is a Chromium embedder

Embeds content layer, affected by all changes in lower layers

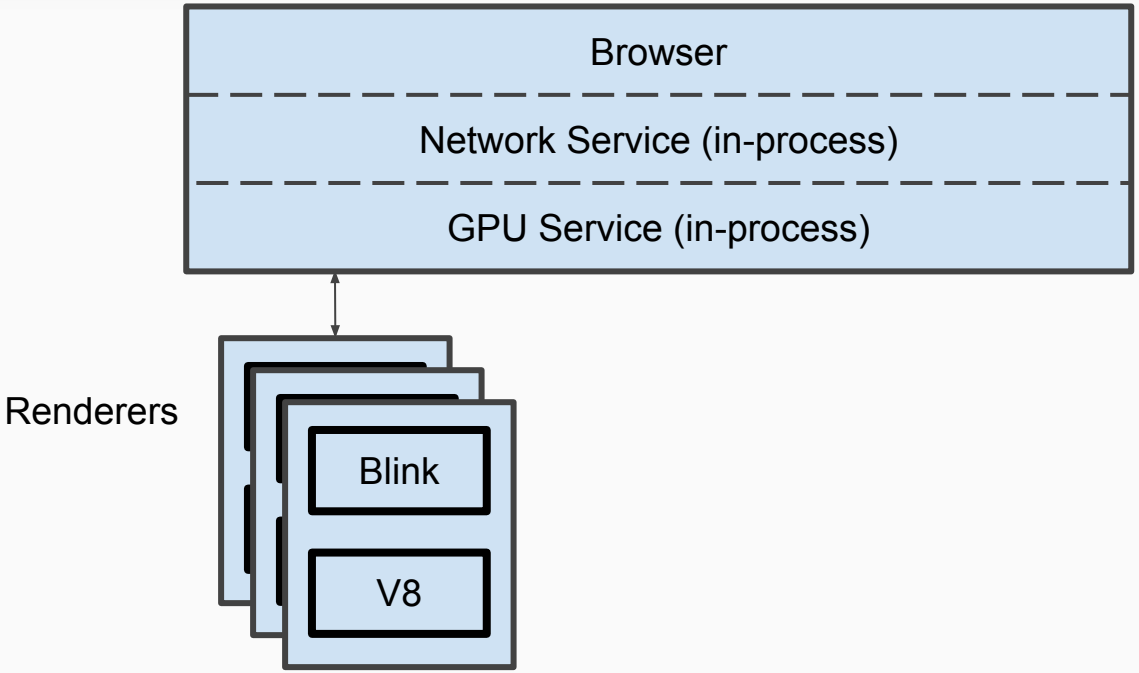
Must support architectural changes (ex. Network Servicification, crashpad)



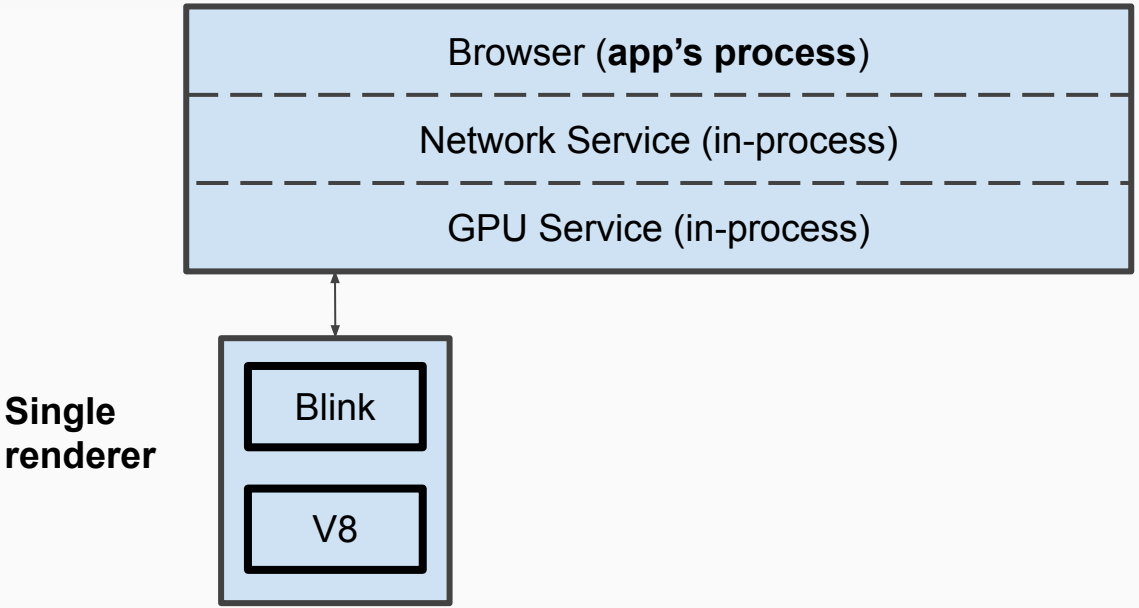
Chrome's architecture (simplified)



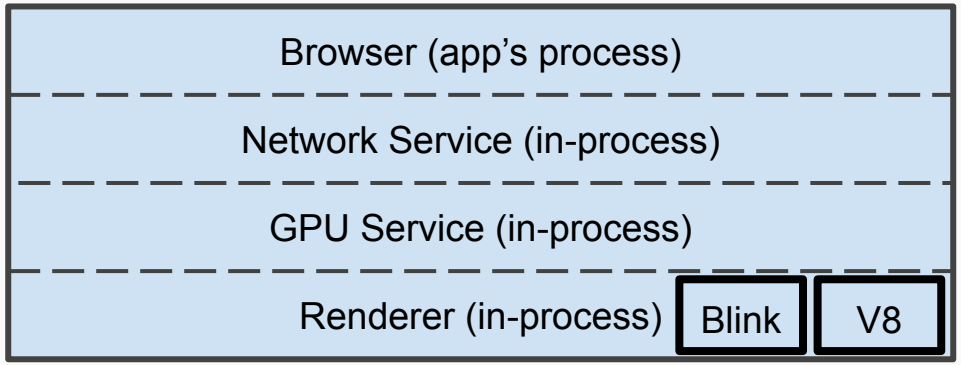
Chrome's *low-memory* architecture



WebView’s architecture (Android O+)



WebView’s architecture (Android L-N)



Working with WebView

Developer workflow

Workflow *almost* as easy as Chrome for Android

- Just need an emulator or a userdebug device
- go/webview-docs/quick-start.md (or, //android_webview/docs/)

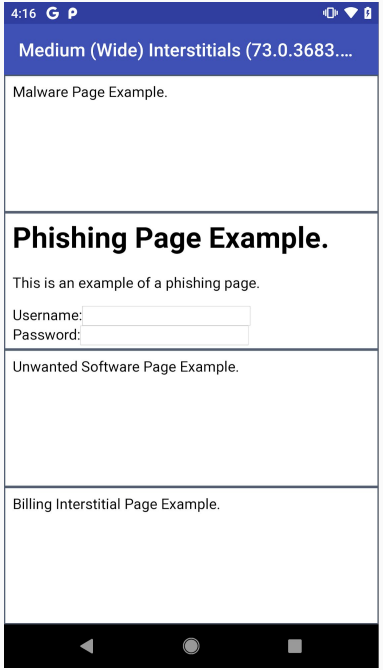
WebView APK only implements WebView APIs

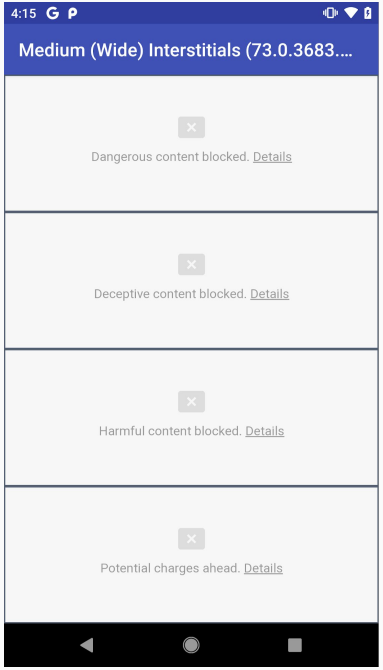
- Also need an embedding app to test
- go/webview-docs/webview-shell.md

Design Challenges

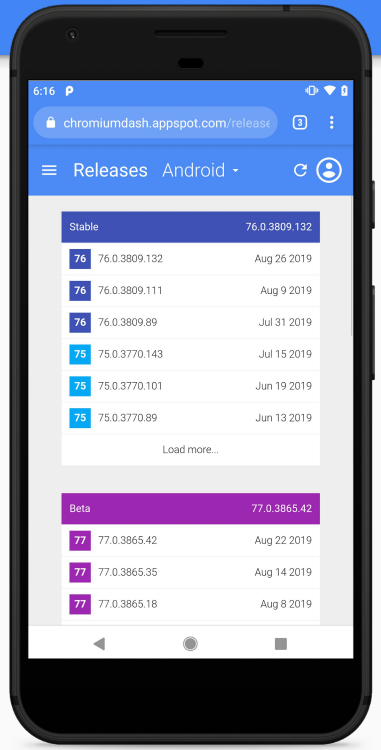
Weird sizes: small, giant

Weird restrictions: no internet, can’t create popups





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

Experiments & metrics

Supports metrics (platform == Android WebView)

Supports field trials (platform == ANDROID_WEBVIEW)

Knows about base::Features and flags in blink, content, etc.

Please remember WebView for your features :)