WebViews
Usages and Challenges
Who are we?

Rayan Kanso
rayankans@google.com
(cochair of WebViewCG)

Peter Conn
peconn@google.com
Outline

- WebView Community Group
- WebView ecosystem
- CG: Usages & Challenges
Goal

The WebView Community Group aims to identify, understand, and reduce the issues arising from the use of **WebViews**
Scope

- Identify representative use cases of where WebViews are being used, regardless of the platform and type of device they're used on
- Identify the issues that arise from these usages
- Determine whether these issues can be addressed through improvements to the Web Platform, or the surrounding ecosystem
- Raise awareness about the impact of WebViews for Web technologies in relevant communities
Charter

You can find more information at the WebView CG charter
So what is a WebView?

Software components that are used to render Web technology-based content outside of a Web browser.
WebViews on Android

iOS

Windows
What is Android WebView?

- Display web contents in your app.
- The native app can interact with the web contents by injecting JavaScript, scrolling, zooming, searching, etc.
- No navigation controls or address bar.
What isn’t Android WebView?

- A full browser:
  - Lacks full web platform support.
  - Lacks fancy browser features.

- An isolated browsing context:
  - The embedding app can see everything.
  - The embedding app can modify everything.

- A comprehensive toolkit for making browsers.
  - No helpers for URL bars.
Aside: You can highlight Android WebView

1. Install [WebView Beta](#).

2. Enable [Android Developer Options](#).

3. Set [WebView Beta](#) as your [WebView Implementation](#).

4. Open the [WebView DevTools](#) app, go to [Flags](#) and enable [highlight-all-webviews](#).
What is a Custom Tab?

- Launches the page in the browser...
  - Shares state.
  - Access to browser features like autofill, translate, etc.

- ...but with a different UI.
  - Close button to return to app.
  - No access to tabs.
  - Somewhat customizable.

- No* interaction between app and web contents.
  - *some limited navigation timing info.
  - *query parameters.
Why did we make Custom Tabs?

- **WebView:**
  - Poor web API support.
  - Poor user privacy/security.

- **Just launching Browser:**
  - User has left the app.

- **Custom Tabs:**
  - Keep the user in the app.
  - Give the user better security and web API support.
What isn’t a Custom Tab?

● It isn’t a single implementation:
  ○ Features will depend on the user’s browser.

● It doesn’t allow web/native mixing:
  ○ Can’t add JavaScript.
  ○ Can’t live alongside native views.
  ○ Can’t be customized as much as a WebView.

● It isn’t clear to the user where their data is.
Summary: Android WebView vs Custom Tabs

**WebViews:**
- Embedder can see/monitor everything.
- Tight integrations between native/web.
- Poor web platform support.
- No advanced browser features.

**Custom Tabs:**
- Embedder has very limited insight.
- Limited integrations between native/web.
- Browser-level web platform support.
- Browser features available (autofill, translate, etc).
WebViews on Android
iOS
Windows
WKWebView

- Allows you to embed web content into your app as a seamless part of your app’s UI.
- Lets you customize or control the display of web content, or interact with the content itself.

https://developer.apple.com/news/?id=trjs0tcd
WKWebView and App-Bound Domains

- App-Bound Domains were released in iOS 14.0 (Sept 2020).
- A limit of 10 domains per app for powerful features.
- Feature is currently opt-in.
- The link is one-way.
- Browsers get an exception.

https://webkit.org/blog/10882/app-bound-domains/
SFSafariViewController

● The same browsing experience as Safari.
  ○ Password autofill
  ○ Reader
  ○ Secure browsing

● SFSafariViewController is best used when you need to display interactive web experiences on websites you don’t own.

https://developer.apple.com/news/?id=trjs0tcd
WebViews on Windows
WebView2

• Lets you embed web technologies in your native apps.

• Provides many powerful features:
  ○ JavaScript execution.
  ○ Messaging between native and web.
  ○ Object sharing between native and web.
  ○ Cookie management.
  ○ Customizing browser features such as printing, autofill, downloads.

https://docs.microsoft.com/en-us/microsoft-edge/webview2/
Non-Normative Reports on WebView use cases and limitations

- You can find a first draft of the report [here](#).
- You can find issues submitted by the Community Group [here](#).

Join us on Friday for the Community Group [meeting](#).
WebView

Status

Usages

Challenges
Types of WebViews

"fully-fledged" WebViews

- Android WebView, WKWebView, and WebView2
- Access to Web content
- UX flexibility
- Isolated state

"browser-like" WebViews

- Custom Tabs,
- SFSafariViewController
- No access to Web content
- No UX flexibility
- Shared state with browser
Hybrid Apps

- Web Apps
- In-App Browsers
Browsers

- Browser customization
Mini apps and super apps

- https://web.dev/mini-app-super-apps/
And many more

- Advertisements
- epub Readers
WebView

Status

Usages

Challenges
Challenge #1 - Performance

- Loading latency
- Pre-caching
- Native integration
Challenge #2 - Inconsistencies

- Web Platform support
- Locally hosted content
- JavaScript injection
- Network interception
- Debuggability
Challenge #3 - Control

- Features
- Permissions
- 3P resources
- Web content
- ...

...
Join the WebView Community Group → link

Add your own issues → link

Join the CG meeting on Friday → link