[redacted]

From: Andreas Bovens [redacted]
Sent: 07 February 2022 13:50
To: Mobile Ecosystems
Subject: Comment on the CMA's Mobile Ecosystems Market Study Interim Report

Dear Madam/Sir,

This is a response with comments on the CMA's Mobile Ecosystems Market Study Interim Report (https://www.gov.uk/government/publications/mobile-ecosystems-market-study-interim-report/interim-report).

First, a brief word about my background. At present, I am a UK resident, and over the last decade, I have worked in various product leadership roles for Opera Software (maker of the Opera browser) and for Mozilla (maker of the Firefox browser), which has provided me with good insights into the mobile browser landscape. I currently work as a Lead Technical Product Manager at Whereby (https://whereby.com/), which is a Norwegian company creating browser-based video conferencing solutions, and which also has an entity in the UK. However, I want to stress that the opinions expressed below are mine, and do not necessarily represent my current employer nor any past employers. If needed, it is fine for these comments to be published quoting my name, Andreas Bovens.

My comments on the interim report are as follows:

1. In my role at Whereby, I'm regularly confronted with the limitations of the WebKit engine on iOS, and the lack of engine diversity due to Apple's App Store rule 2.5.6 (https://developer.apple.com/app-store/review/guidelines/). Certain aspects of Whereby's browser-based video conferencing solution rely on web APIs that, although standardised, are not or only partially supported by WebKit, and the lack of alternative browser engines on iOS means that Whereby cannot always serve its iOS customer base in an optimal manner. Indeed, certain product features remain unavailable for Whereby's iOS customers due to a lack of standards support, and the Whereby engineers spend a not insignificant amount of their time debugging and working around WebKit-specific issues. Requiring Apple to allow alternative web engines on iOS would likely improve the current situation, as it would give users a choice, and in the long run, increase competition and browser engine quality on the platform.

2. I want to make a clarifying comment about Android WebView in the section "Restrictions on browser engine choice for in-app browsers" (https://www.gov.uk/government/publications/mobile-ecosystems-market-study-interim-report/interim-report#restrictions-on-browser-engine-choice-for-in-app-browsers).
It's worth pointing out here that while Android app developers can indeed use a browser engine other than Blink to power their app, they have to bundle this engine with each Android app they ship (e.g. Mozilla bundles its Gecko engine with its Firefox app for Android). However, if these app developers don't want to bundle a full engine with their app, their only choice is to hook into the preinstalled Android WebView, which is powered by Chrome; indeed, it is not possible for users to install an alternative WebView component on their Android phones (or for developers to hook into such a component), which puts third-party browser engines at a disadvantage, and limits app developers' options.
Requiring Google to allow alternative WebView components to be installed on Android, which then could be used across multiple apps if app developers so desire, would be good for user choice and browser engine competition.

3. The report does not appear to contain any reference to Custom Tabs, which Google's Android developer docs (https://developer.chrome.com/docs/android/custom-tabs/) describe as follows: "Custom Tabs is a browser feature, introduced by Chrome, that is now supported by most major browsers on Android. It [gives] apps more control over their web experience, and [makes] transitions between native and web content more seamless without having to resort to a WebView." Essentially, Custom Tabs can be thought of as a fast-loading single tab experience, which can be launched from a native app like e.g. Twitter or Slack to quickly show linked web content. App developers can typically customise the Custom Tab's colour and action button, and unlike Android WebView, Custom Tabs typically share their cache, cookies and permissions with the browser they're powered by, allowing users to share session info between their browser and any Custom Tabs they launch.

To its credit, Google allows other browser makers to create their own Custom Tabs integrations, which is something e.g. Firefox has done in its Android app: when a user sets Firefox as the default browser on Android, Custom Tabs opened from apps like Twitter and Slack will be powered by Firefox' Custom Tabs implementation.
However, it's worth pointing out here that Google's highly visible Google app, which can be launched when performing a search through Android's built-in Google search box (https://play.google.com/store/apps/details?id=com.google.android.googlequicksearchbox), forces search result links to always open in Chrome's own Custom Tabs implementation, regardless of what the user has set as their default browser. In other words, Google gives the tools to browser makers to ship their own Custom Tabs implementation and integrate with Android users' interaction patterns and workflows, but then prevents this implementation from properly integrating with the Google app, which is arguably one of Google's most important apps on Android.
I recommend the CMA to look deeper into the Custom Tabs mechanism, and the role it plays in the Android ecosystem. Requiring Google to respect user choice when launching links from its own apps in Custom Tabs would level the skewed browser playing field on Android, and it would benefit browser engine competitiveness on Android.

Thanks for your consideration of my comments,

Andreas Bovens