



Google's Response to Working Paper #4: In-App Browsing Within the iOS and Android Mobile Ecosystems

31 July 2024

Introduction & Summary

1. Google welcomes the opportunity to respond to the CMA's Working Paper on in-app browsing within the iOS and Android mobile ecosystems (**WP4**).
2. We agree with WP4's suggestion that in-app browsers (**IABs**) play a significant role in users' mobile experiences.¹ In a highly competitive native app environment where app developers compete for users' attention, it is important that developers can design their apps to suit their users' needs, including in respect of in-app browsing.²
3. We also welcome WP4's recognition of the benefits of giving app developers choice over how they design their IABs.³ On Android, developers have significant flexibility to customise their IABs so they can meet their users' needs, differentiate themselves, and compete.⁴
4. Unlike on iOS, on Android there is no restriction on the browsers that app developers can use for "remote-tab" IABs (Custom Tabs on Android), or on the browser engine they can use for "webview" or "bundled-engine" IABs. This promotes competition in: (i) browsers, by enabling any browser to provide remote-tab IABs easily, so that they activate whenever their browser is set as the user's default; and (ii) browser engines, as app developers can develop IABs based on any browser engine.
5. Accordingly, WP4 finds that Google's remote-tab and webview IAB policies are not likely to limit competition among mobile browsers on Android as they do not prevent rivals from offering competing products, although there is limited demand among browser engines to create alternative webview IABs.⁵ We agree.

¹ WP4, ¶2.57.

² See also WP4, ¶3.5 ("*[D]evelopers also internalise the demand that they expect from users of their native app*").

³ WP4, ¶5.9.

⁴ WP4, ¶2.57.

⁵ WP4, ¶5.9, 5.18.

6. WP4 suggests, however, that Google’s and Apple’s policies may limit user choice and control in relation to IABs.⁶ On Android, this emerging thinking overlooks that:
- For app developers using Custom Tabs IABs, users’ default browser choice is respected by default.
 - Users can easily exit IABs and open web pages in their preferred browser if they wish to do so. Some apps choose to allow users to “switch off” their IAB by default in a settings menu.
 - Contrary to WP4’s suggestion, the CMA’s consumer research suggests that users are aware that they are viewing content in IABs.
7. In this response, we set out why Google’s IAB policies promote competition in native apps, browsers, and browser engines, as well as user control and choice relating to IABs:
- **Section I** describes how Google’s Custom Tabs IABs promote competition in browsers and browser engines.
 - **Section II** demonstrates how Google’s webview and own-bundled IAB options provide app developers with a range of tools for highly customised IABs.
 - **Section III** explains how Google’s policies promote user control and choice relating to IABs.
8. Ultimately, how IABs are implemented affect several stakeholders on Android and there are trade-offs in offering different options to both developers and users. We believe that we strike the right balance in respecting both developer choice and user choice.
- I. Android Custom Tabs Promotes Competition in Browsers and Browser Engines**
9. We welcome WP4’s finding that our policy on remote-tab IABs is not likely to limit competition in browsers on Android, and its recognition that “*there are benefits in allowing app developers ... to have some degree of choice over the IABs in their app.*”⁷ Below we provide further details on how remote-tab IABs on Android promote browser competition and benefit both developers and users.
10. Android’s remote-tab IAB offering, Custom Tabs, allows app developers to call on a browser app to embed a tab into their IAB. By giving Android developers this tool, we increase browser competition in the following ways:

⁶ WP4, ¶¶5.1 and 5.35.

⁷ WP4, ¶5.9

- First, Custom Tabs provides an attractive option for app developers incorporating IABs into their apps at low cost. This gives browser vendors an opportunity to be the browser that powers users' IAB experiences.
- Second, browsers can compete based on their Custom Tabs implementations. This can make their browser a more attractive option to both users and app developers in relation to IABs.
- Third, browsers Custom Tabs implementations will be powered by the same browser engine as the underlying browser. Because Android offers browser engine choice, Custom Tabs also enhances browser engine competition.

11. We expand on these points below.

12. **Android Custom Tabs is an attractive IAB solution for app developers.** We recommend Android app developers use Custom Tabs when displaying third-party web content to their users.⁸ Custom Tabs is attractive to app developers in the following ways:

- **Choice of underlying browser.** Custom Tabs enables developers to rely on the user's default browser to power their IAB (the default option), or choose for content to open in a specific browser's Custom Tabs implementation. Browsers on Android that support Custom Tabs include Firefox, Vivaldi, Brave, and Edge, as well as Chrome. As the IAB is rendered through the underlying browser app, it inherits the performance, security, and privacy capabilities of the browser app that it calls on.
- **Ease of use.** [*Confidential*]. Custom Tabs IABs are convenient, low-cost ways to incorporate and customise IABs, as WP4 explains.⁹ Developers are not required to write custom code, or manage requests, permission grants, or cookie stores.
- **Customisation options.** For apps that do not have sophisticated requirements for IABs, or do not want to invest in building custom IAB interfaces from scratch, Custom Tabs enables developers to customise their IABs¹⁰ (e.g., entrance and exit animations, colour scheme, and specifying the launch height of the IAB).¹¹ Apple's remote-tab IAB, by contrast, is always powered by Safari. iOS developers cannot offer users as rich or customised

⁸ Chrome for Developers, [Overview of Android Custom Tabs](#). See also WP4, ¶2.28(a) ("Google recommends that app developers use Custom Tabs for opening third-party web content").

⁹ WP4, ¶2.10.

¹⁰ WP4, ¶2.47-2.48.

¹¹ Chrome for Developers, [Overview of Android Custom Tabs](#).

an experience. For example, Pinterest submitted to the CMA that Apple's remote-tab IAB "does not allow it to add certain features [...] i.e. their red 'pin' button," which allows users to directly save content to their Pinterest account from the IAB, resulting in users on iOS being "confused" due to the lack of a consistent experience with Pinterest's interface.¹²

- **Freedom to specify a particular browser.** In some circumstances, developers may wish to specify a particular browser to power their Custom Tabs. This option is an important aspect of developers' control over their app designs and experiences,¹³ allowing further room for app differentiation, and WP4 recognises the benefits this choice can bring.¹⁴ App developers may want their IABs to offer, or not offer, certain features available only through a particular browsers' Custom Tabs. For example, they may want to offer, or not offer, 'dark mode' to users, which is offered by some browsers including Chrome, but not Samsung Internet.

13. **Custom Tabs promotes browser competition.** [Confidential]. On Android, we provide the freedom for any browser vendor to offer a Custom Tabs version of their browser. Because Custom Tabs are popular with app developers, they are likely also to be popular investments for browsers to make.
14. As WP4 explains, offering remote tab IABs to app developers can enable browser vendors to support their users more effectively and increase the time users spend in their browser.¹⁵ As a result, "an increasing number of browser vendors currently offer this product on Android,"¹⁶ and would be interested in doing the same on iOS if it were technically possible.¹⁷ Browser vendors told the CMA that "they consider their remote tab IAB as a feature of their wider competitive offering to users."¹⁸
15. We agree. [Confidential].
16. As part of the competitive process that Custom Tabs facilitates between browsers on Android, different browsers' implementations of Custom Tabs offer different features. [Confidential]. Accordingly, we agree with WP4's preliminary conclusion

¹² As explained by WP4, ¶4.10.

¹³ As recognised by WP4, at ¶2.53: "[I]f the developer wanted to ensure that specific features were supported by the IAB, then it might choose a browser that it knew offers these features."

¹⁴ WP4, ¶5.8.

¹⁵ WP4, ¶2.36; see also ¶4.8.

¹⁶ WP4, ¶4.7.

¹⁷ WP4, ¶4.14 ("[B]rowser vendors would be interested in offering a remote tab IAB on iOS to improve the quality of their offering and to better support their users."). [Confidential].

¹⁸ WP4, ¶2.36.

that our support for remote tab IABs on Android does not give rise to an adverse effect on competition.¹⁹

17. **Custom Tabs promote browser engine competition.** On Android, a browser is free to use the browser engine of their choice. This choice extends to browsers' implementations of Custom Tabs. This means that, for example, Firefox Custom Tabs uses the Gecko browser engine. In this respect, our support for Custom Tabs on Android benefits browser engine competition as well as browser competition.

II. **Webview and Bundled-Engine IABs Provide a Valuable Opportunity for Customisation**

18. WP6 recognises that it is possible for browser engines to offer alternatives to Android WebView on Android. It comments, however, that this may be “*difficult*,” although noting that the impact of this is unclear.²⁰ Below, we explain how Android offers several options allowing developers to provide bespoke IABs. That said, not all developers have a commercial need to develop bespoke IABs, and accordingly the CMA should not attribute low uptake of this option to any restrictions imposed by Google.

19. **Android provides options for developers to provide bespoke IABs.** Some app developers require a higher degree of customisation to build specific features from scratch than Custom Tabs offers. For example, as webview IABs allow for more control and customisation for app developers, they are able to build functions to receive further insights into how users interact with their IAB if they choose to.²¹ This engagement data may be used for improving the user interface and monetisation opportunities, among others.²² Options for building bespoke IABs on Android include:

¹⁹ WP4, ¶¶5.1(a); 5.9.

²⁰ WP4, ¶5.18.

²¹ Custom Tabs API also enables app developers to measure user engagement signals: (i) a “CustomTabsCallback” function for tracking basic navigation events, such as when a user has navigated to a certain page; and (ii) an “EngagementSignalsCallback” function for tracking page specific user engagement, such as scroll direction or scroll percentages. Both APIs can be supported by any browser’s version of Custom Tabs. The “CustomTabsCallback” API is an API that Google understands is widely supported by other browser versions of Custom Tabs. The “EngagementSignalsCallback” API was introduced more recently. Google does not have visibility over which browser vendors support this API in their versions of Custom Tabs. See Chrome for Developers, [Measure user engagement](#).

²² Monetisation opportunities for third parties in Custom Tabs may benefit from the user’s “shared state” (*i.e.*, cookies, payment methods, browsing history and login details) between the browser powering the Custom Tabs IAB and the standalone browser app, as the website inventory can be effectively monetised through advertising (including through both search and display ads). This may enable more tailored ad content in both the Custom Tab IAB and the user’s default browser, increasing opportunities for ad providers in both the app’s IAB and the default browser app (as both benefit from the additional browsing history in the user’s “shared state”).

- Use a ‘webview’ browser engine solution to build and design the interface of their IABs from scratch.²³ On Android, developers can use any browser engine to power their customised IABs. As well as Android WebView, they can choose Mozilla’s webview (called GeckoView) based on its browser engine, Gecko.
 - Use a ‘bundled-engine’ IAB, which involves building their own custom or forked browser engine for their IAB, as in the case of the Facebook app.²⁴
20. WP4 recognises that Google’s webview IAB policy allows browser engines other than Blink to offer alternative webviews, although notes that there may be limited demand to do so.²⁵ But it considers that it may still be “*difficult for browser engines other than Blink to offer attractive alternatives to Android WebView.*”²⁶ WP4 cites submissions that Android WebView “*comes pre-installed on Android devices*” and that “*app developers looking to offer a webview IAB based on an alternative webview (such as GeckoView) can face certain frictions.*”²⁷
21. These concerns are misplaced. Providing tools such as WebView to developers is a critical part of our role as supplier of the Android OS. Components such as WebView cannot properly be characterised as “defaults” (or, at least, as defaults of the type that can give rise to inertia bias). We enable developers to easily use alternatives. The only additional step involved for app developers is incorporating the third-party browser engine’s library (e.g., GeckoView’s library) into their app.
22. WP4 considers that alternative webviews can mean “*the app takes up more storage on a user’s device, which may reduce its attractiveness to users.*” This difference does not affect the ability of alternative webviews to compete, as:
- Additional storage is a trade off that an app developer would have to consider for any additional functionality. Re-building an API or functionality that the OS provides by default will involve a change in the app’s size. This has not prevented app developers from building these functionalities. In fact, many app SDKs (software development kits) which provide apps additional functionality have been very successful on Android, despite any increased storage.
 - In any event, additional storage taken up would be minimal, and would not affect the attractiveness of the app to users.

²³ WP4, ¶2.48(b).

²⁴ WP4, ¶¶2.16-2.17.

²⁵ WP4, ¶5.18.

²⁶ WP4, ¶5.18.

²⁷ WP4, ¶5.14.

23. **Bespoke IABs make little commercial sense for some app developers.** There is a wide range of different app developers on Android. For some, development and maintenance costs mean that investing in webview or own-bundled engine IABs may not make commercial sense. This does not, however, mean that developing a bespoke IABs is unviable, as shown by the fact that several apps, spanning a spectrum of categories, have built such IABs.
24. As WP4 recognises, webview or bundled-engine IAB options are more attractive for app developers with higher customisation requirements and incentives to build their own interface or even, in the case of bundled engines, incentives to fork and maintain their own browser engine.²⁸ In particular, app developers also take on responsibility for maintaining certain security and privacy capabilities. [Confidential].²⁹ This lack of uptake is not, however, attributable to the “default” status of Android WebView. It is the outcome of a simple tradeoff for developers between customisation and development cost/effort.
25. **A wide range of apps have chosen to build customisable webview IABs.** Bespoke IABs are nonetheless popular with app developers in a range of categories, including healthcare, travel, job search, gambling, as well as social media apps. For example, popular apps in the UK in terms of time spent in WebView include: the NHS app, the Jet2 Holidays app, the National Lottery app, the Indeed job search app, as well as gambling (e.g., Ladbrokes), and social media (e.g., Instagram and Telegram) apps.³⁰
26. [Confidential].³¹
27. In sum, our approach to facilitating bespoke IABs for developers that prioritise customisation over maintenance costs cannot plausibly give rise to an AEC on Android.

III. Google Facilitates User Control and Choice of IABs

28. WP4 suggests that users on Android (and iOS) have limited choice and control in relation to: (i) which browser is used for IAB implementations; or (ii) whether IABs are used in general. This is based on WP4’s findings that: (i) a user’s default may not be applied in IABs; (ii) users may be unaware they are in an IAB versus a browser; and (iii) users are generally not provided the same level of control over their IAB experience via an IAB menu.³²

²⁸ WP4, ¶¶2.12 and 2.19.

²⁹ [Confidential].

³⁰ [Confidential].

³¹ [Confidential].

³² WP4, ¶5.34.

29. This concern does not apply to IABs on Android, where users have a large degree of choice over the browser they open web content in from native apps, for three main reasons:

- First, Custom Tabs API respects user choice by default.
- Second, app developers usually give users the option to switch off or exit IABs.
- Third, WP4’s concern around users’ lack of awareness that they are in IABs is not supported by evidence.

30. We expand on these points below.

31. **Custom Tabs API respects user choice by default.** As WP4 recognises, Google supports user choice of their default browser by setting Custom Tabs to rely on the user’s default browser unless the developer decides to override this.³³ Indeed, most app developers implement Custom Tab IABs without specifying a particular browser. As WP4 finds, Custom Tab IABs are “*set by most native app developers to call on the user’s default browser.*”³⁴ On Android, major UK apps such as ██████████ incorporate Custom Tab IABs and respect the user’s default browser.

32. **App developers usually give users the option to switch off or exit IABs.** On Android, users can and do use controls provided by app developers to allow them to open links in their default browser rather than an IAB when they want to. Users have two principal means of controlling the browser that links to third-party content from a native app open in:

- First, many apps provide an option for users to open links in their default browser once they have opened the link in an IAB. As the provider of the Custom Tabs system, Google recommends that developers allow users to include an option in the application for the user to open links in the default browser instead of using a Custom Tab.³⁶

For Chrome Custom Tabs, the mechanism that lets users open a webpage in their default browser is hardwired and not removable, meaning that when in Chrome Custom Tabs IABs, users will always have the option to open a link in their external default browser. Google understands that major browser vendors (e.g., Samsung Internet, Edge, and Firefox) offer users the same

³³ WP4, ¶5.20(b).

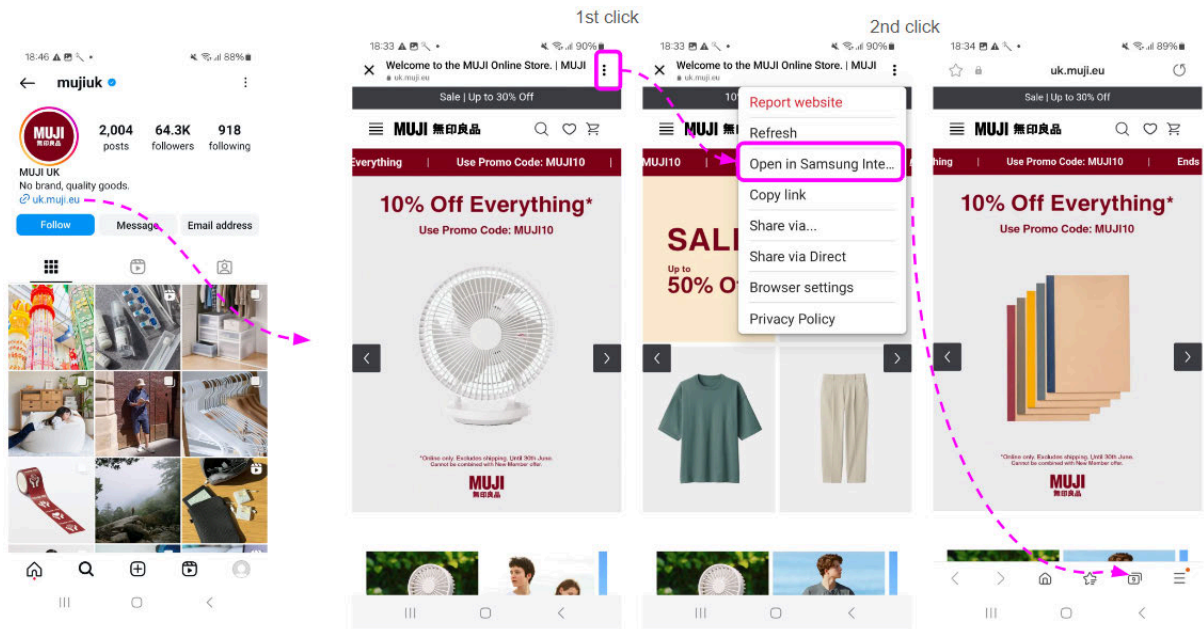
³⁴ WP4, ¶2.34. See also ¶2.10(b) (“*In most cases, apps invoke the user’s default dedicated browser in Custom Tabs mode*”).

³⁵ [Confidential].

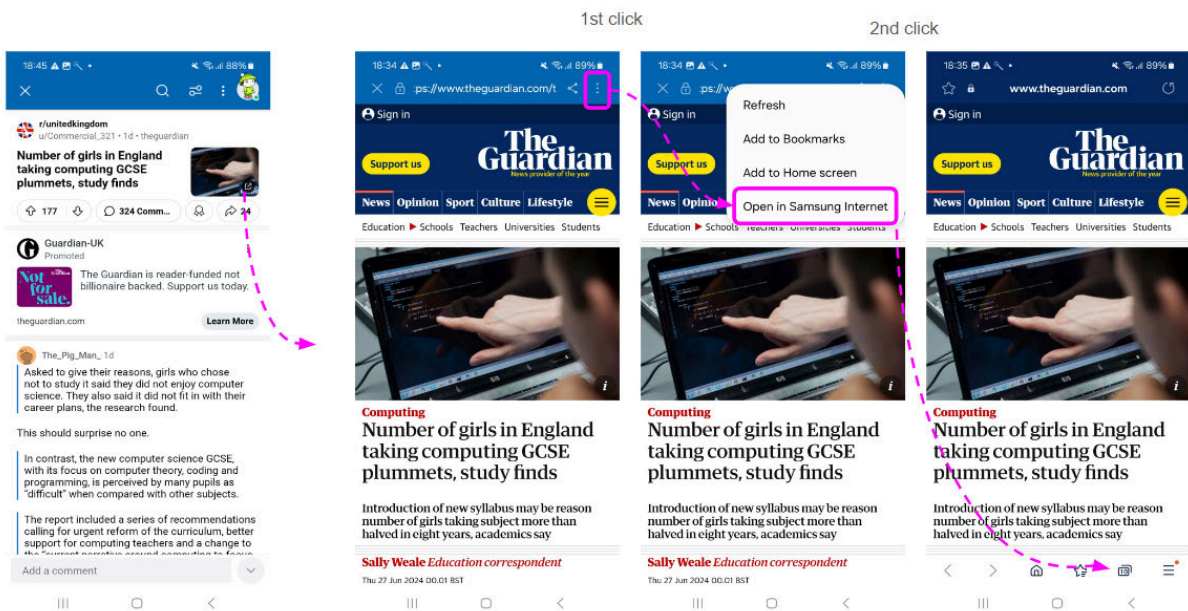
³⁶ Medium, [Best Practices for Custom Tabs](#). See also WP4, ¶2.56(b) (“*Google said that adding opt-out of in-app browsing is recommended by Android Custom Tabs as best practice*”).

option to open links in the user's external default browser instead. Many popular apps in the UK allow the user to exit the IAB in two taps, as illustrated below.

User Journey When Exiting Instagram's IAB



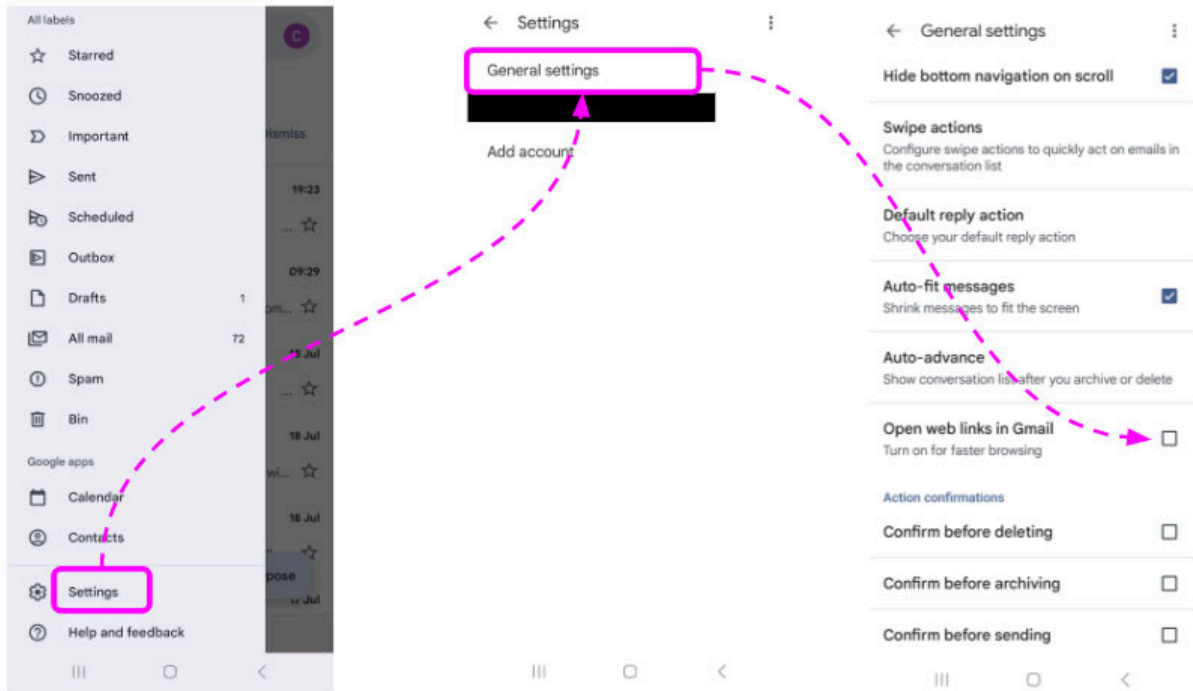
User Journey When Exiting Reddit's IAB



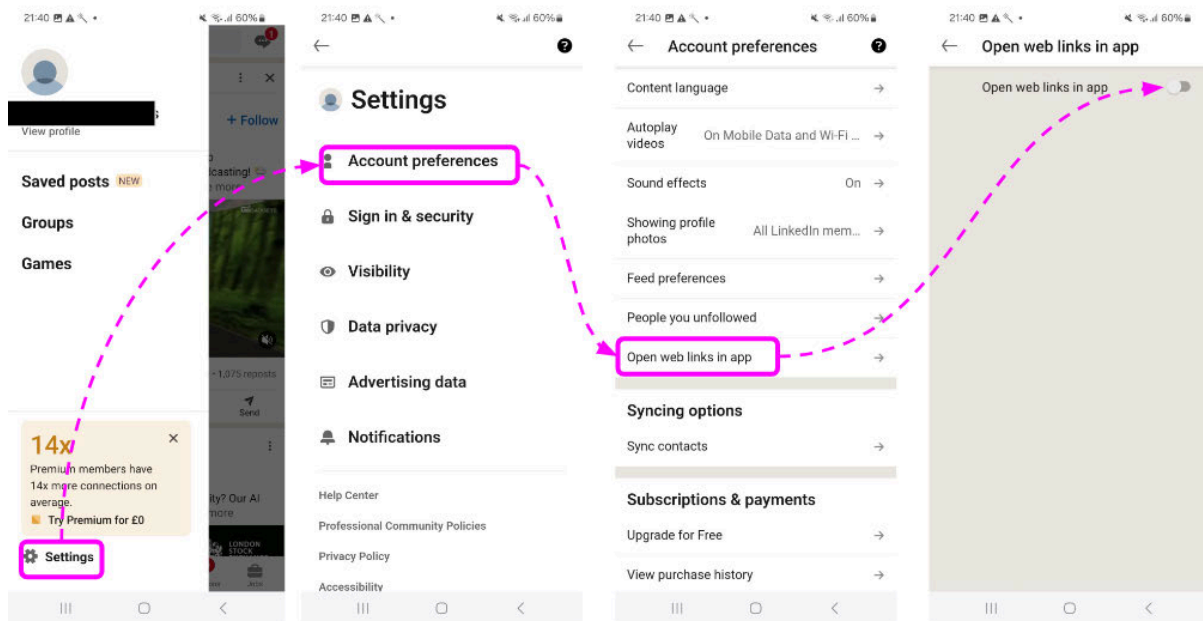
- Second, some apps allow users to “switch off” their IAB by default in a settings menu, which means all links to third-party content will open in the user’s default standalone browser. As WP4 recognises, this option is

available in the Gmail app, as demonstrated below.³⁷ It is also possible to disable in-app browsing in the LinkedIn app.

Disabling In-App Browsing in Gmail App



Disabling In-App Browsing in LinkedIn App



³⁷

33. **The CMA’s consumer research confirms user awareness of IABs.** WP4 raises a potential concern that IABs’ interfaces on Android resemble the Chrome browsing experience, which could contribute to lower levels of user awareness of in-app browsing. It notes that users may not be aware they are in an IAB, which could mean that users are being tracked without awareness or consent when browsing in an IAB, or less likely to control their IAB settings.³⁸ This is not supported by the evidence available:
- Verian’s qualitative consumer research demonstrates that users are generally aware that they are not in their dedicated browser app and had remained within the same app after clicking a link. Users responded that they are aware “*it does open a browser but doesn’t open in Chrome,*” and respondents could distinguish between leaving the app “*for a browser*” versus remaining within the app.³⁹
 - Verian’s qualitative findings show high user awareness that they could potentially be “tracked” in an IAB by the native app, with the most frequent guess about who could access data from entering an IAB being a social media company (*i.e.*, the app). The survey found a “*general reticence about clicking on links within social media apps*” as a result.⁴⁰
34. On Android, users have multiple ways to exercise their choice of IAB. Google’s practices cannot plausibly give rise to an AEC.

Conclusion

35. How IABs can be implemented involves a delicate balance taking into account the needs of several stakeholders within the Android ecosystem, principally app developers and users. As WP4 recognises, the flexibility on Android for app developers in incorporating IABs in their app brings several benefits, which in turn give users a better app experience.⁴¹ [*Confidential*].
36. We agree with WP4’s emerging finding that Google’s remote tab and webview IAB policies are not likely to limit competition among mobile browsers on Android as they do not prevent rivals from offering competing products, although there is limited demand among browser engines to create alternative webview IABs.⁴²

³⁸ WP4, ¶¶5.21.

³⁹ Verian consumer research presentation of key qualitative findings, p. 31.

⁴⁰ *Ibid.*

⁴¹ See, e.g., WP4, ¶¶ 3.8, 3.10 and 5.9.

⁴² WP4, ¶¶5.9; 5.18.

37. When considering user choice and control in relation to IABs, WP4 overlooks the fact that users can easily exit IABs and open web pages in their preferred browser if they wish to do so. Some apps even choose to allow users to “switch off” their IAB by default in a settings menu. And for apps using Custom Tabs IABs, user choice of browser is respected by default. Android users cannot therefore be described as lacking choice over the browser they use when opening links from native apps.

* * *