# MOBILE BROWSERS AND CLOUD GAMING MARKET INVESTIGATION

## WP4: In-app browsing within the iOS and Android mobile ecosystems

05 July 2024

CMA
Competition & Markets Authority

This is one of a series of consultative working papers which will be published during the course of the investigation. This paper should be read alongside the Issues Statement published on 13 December 2022 and other working papers published.

These papers do not form the inquiry group's provisional decision report. The group is carrying forward its information-gathering and analysis and will proceed to prepare its provisional decision report, which is currently scheduled for publication in October 2024, taking into consideration responses to the consultation on the Issues Statement and responses to the working papers as well as other submissions made to us.

Parties wishing to comment on this paper should send their comments to browsersandcloud@cma.gov.uk by **29th July 2024**.

The Competition and Markets Authority has excluded from this published version of the working paper information which the inquiry group considers should be excluded having regard to the three considerations set out in section 244 of the Enterprise Act 2002 (specified information: considerations relevant to disclosure). The omissions are indicated by [✂]. Some numbers have been replaced by a range. These are shown in square brackets. Non-sensitive wording is also indicated in square brackets.

# Contents

## Figures

# 1. Introduction

1.1 In-app browsing refers to the situation where a user accesses web content while they are already in a native app that is not a dedicated browser. This might occur, for example, when a user is viewing a news article after clicking on a link within a social media app and – instead of being directed to their dedicated browser app – they view the article from within the social media app itself.

1.2 As set out in the Issues Statement for this market investigation, this investigation is considering whether the handling of hyperlinks and the implementation of in-app browsers (IABs) in native apps on iOS and Android may weaken browser and browser engine competition.

1.3 This paper considers how IABs operate on iOS and Android; and how Apple's and Google's policies for different implementations of IAB within native apps may limit competition between browsers and browser engines for offering IAB implementations.

1.4 This paper (**WP4**) is one of a series of working papers being published in this market investigation which consider competition in mobile browsers and mobile browser engines. Therefore, it should be read alongside other working papers produced as part of the market investigation. It closely relates to:

(a) 'WP1 – Nature of competition in the supply of mobile browsers and browser engines';

(b) 'WP2 – The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine';

(c) 'WP5 – The role of choice architecture on competition in the supply of mobile browsers'; and

(d) 'WP7 – Potential remedies'.

1.5 This paper is structured as follows.

(a) Section 2 sets out key facts and background to in-app browsing on mobile devices and the different ways that this can be implemented.

(b) Section 3 sets out our emerging thinking on the relevant market.

(c) Section 4 sets out Apple's key policies for how in-app browsing can be implemented as a feature within native apps on iOS devices and the extent of their impact on how browsers, browser engines and IABs compete.

(d) Section 5 sets out Google's key policies for how in-app browsing can be implemented as a feature within native apps on Android devices and the extent of their impact on how browsers, browser engines and IABs compete.

# 2. Background

## What is in-app browsing and how does it work?

2.1 In-app browsing refers to a situation where a user views web content in an IAB within a native app instead of being taken to a separate browser app on their mobile device (referred to in this paper as a 'dedicated browser').

2.2 IABs are typically simpler than dedicated browsers, offering fewer functionalities to their users. IABs are generally designed for users to view one or two web pages before returning to their previous activity in the native app. For this reason, they usually lack features for enabling users to navigate the web – eg unlike dedicated browsers, IABs typically lack a search bar and browsing tabs.[1]

2.3 Web content can be displayed within native apps in a variety of formats, which include IABs.[2] When a native app displays a link to web content, the native app developer can:

(a) direct the user out of the app to a dedicated browser or an alternative native app;

(b) keep users inside the app and display the content via an IAB; or

(c) show a choice screen of the above options.

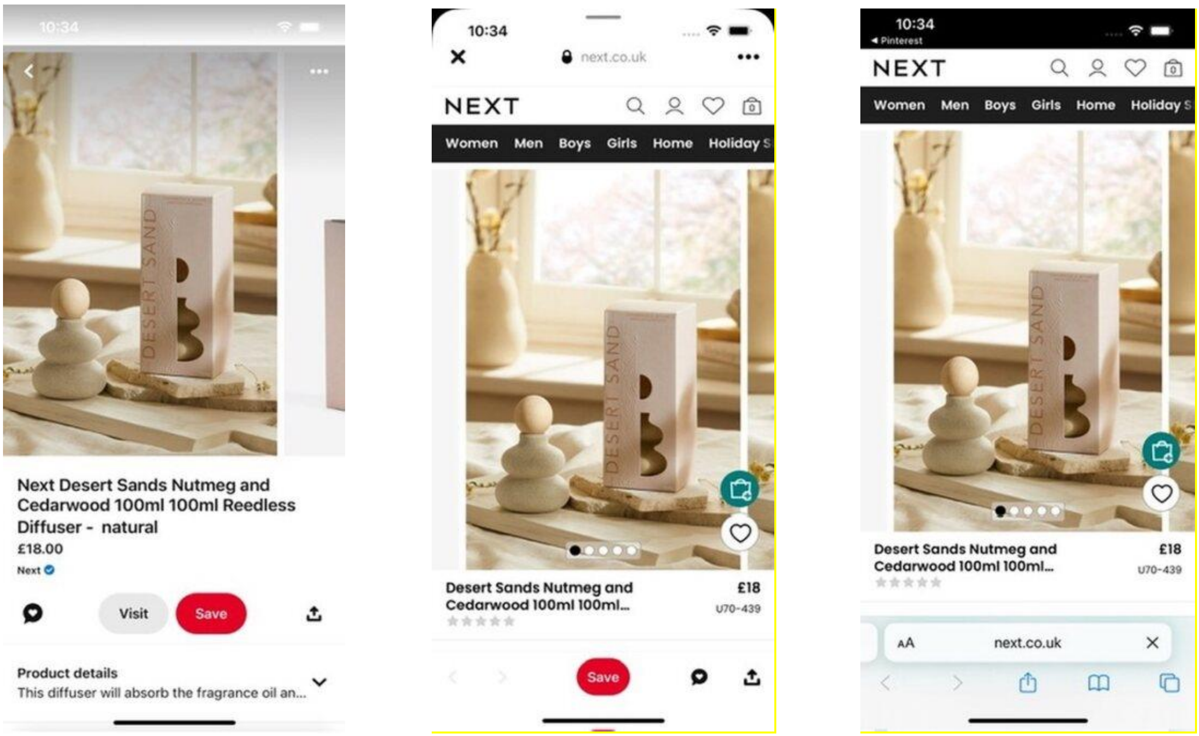
2.4 In-app browsing in the context of this paper refers to option (b) above. More specifically, this paper focuses primarily on the use of IABs to display third-party web content. This is because most of the evidence we have heard so far on in-app browsing relates to the display of third-party web content in an IAB. Third-party web content is developed by an outside, unaffiliated party and not owned by the native app. A classic example of this is the display of a news article when a user clicks on a link within a social media app.

2.5 Native apps may also use the same in-app browsing technology to display first-party web content which is their own – an example of this would be a display of the terms and conditions section of their own website.

[1] A search bar is the box where a user can type words to get weblinks as search results. Browsing tabs are the browsing sessions which can typically be open at the same time while in a dedicated browser.

[2] For example, apps might integrate web content in the form of a small rectangular advertisement at the bottom of the screen. For in-app browsing, apps display full web pages, resembling a simplified version of a dedicated browser.

**Figure 2.1: Screenshots of web content owned by a third party viewed in an app, an IAB and a dedicated browser**


10:34
Next Desert Sands Nutmeg and Cedarwood 100ml 100ml Reedless Diffuser - natural
£18.00
Next
Visit
Save
Product details
This diffuser will absorb the fragrance oil an...
10:34
next.co.uk
NEXT
Women Men Boys Girls Home Holiday S
Desert Sands Nutmeg and Cedarwood 100ml 100ml...
£18
U70-439
Save
10:34
◂ Pinterest
NEXT
Women Men Boys Girls Home Holiday S
Desert Sands Nutmeg and Cedarwood 100ml 100ml...
£18
U70-439
AA
next.co.uk

**Pinterest app.**

**Link opened in an IAB within Pinterest.**

**Link opened in dedicated browser.**

2.6 The above images show how web content can appear differently depending on whether it is opened in an IAB or a dedicated browser app. When the link is opened in the Pinterest IAB (middle image), it has the option in the top left to close the screen and return straight to the app. It also keeps the save button at the bottom of the page and has a comment bubble along with Pinterest's own custom share button. When the link is opened in the dedicated browser app (right image), the webpage no longer features any Pinterest specific buttons and there is no close button at the top of the page to return directly to the app.

2.7 There are different ways that an app can integrate an IAB (these are referred to as IAB 'implementations'). The main types are 'remote tab', 'webview' and 'bundled engine' implementations. An overview of these implementation types on iOS and Android is provided below.

## Remote tab

2.8 A remote tab implementation of in-app browsing (ie a 'remote tab IAB') is essentially where an app calls upon a dedicated external browser to display web content. When a user taps a link within an app, the app calls on a dedicated browser installed on the device to load and present web content. This could be likened to a 'window' to a browser, opened from within the app. Remote tab IABs are different from instances where the user is sent to an external dedicated browser because the user remains inside the original app.

2.9 App developers have less scope to customise and control remote tab IABs compared to other types of IAB implementations. For example, app developers generally cannot monitor user activity in the IAB but they can change the colour of the toolbar and introduce certain customisations, such as the proportion of the screen that is covered by the remote tab. We understand that remote tab IABs may exhibit certain security and privacy vulnerabilities that are generally not present in dedicated browsers.[3]

2.10 Remote tab IABs are relatively easy and not costly to implement for the app developer compared to other IAB implementation types as they rely on dedicated browsers installed on the device, including for ensuring the security (and privacy) of the in-app browsing experience. An app developer does not need to build on or maintain the IAB. App developers may choose to rely on a specific dedicated browser for the remote tab IAB, although the level of choice available to them is different between iOS and Android:

(a) On iOS, the only available remote tab IAB is called **SFSafariViewController**,[4] which relies on Apple's mobile browser Safari and is powered by Apple's browser engine WebKit.[5] This means that Safari is the only browser available for in-app browsing where a remote tab implementation is chosen by the app developer on iOS. If a user sets a browser other than Safari as their 'default' dedicated browser, apps with remote tab IABs on iOS would still call on Safari for in-app browsing.

(b) On Android, browsers can be called upon for remote tab in-app browsing via a system called **Custom Tabs**.[6] Any browser that offers a version of Custom Tabs and is installed on the user's device can be used as a remote tab IAB on Android devices. In most cases, apps invoke the user's default dedicated browser in Custom Tabs mode. As explained above, this implementation

[3] See cct_oakland24.pdf (lindorfer.in), accessed by the CMA on 26 June 2024.
[4] Apple's description of SFSafariViewController, from response to CMA's information request [✂]: 'SFSafariViewController is designed to allow developers to provide in-app browsing without the need to create their own browsing view. It presents a self-contained web interface inside an app. SFSafariViewController is essentially a "plug-and-pay" offering, providing system level functionality for developers that may be less focused on browsing and therefore less able or willing to design their own browsing webview from scratch using WKWebView.' For clarity, Apple submitted that 'WKWebView is a general API that developers use to enable internet browsing or to display web content in their apps. As WKWebView is designed to allow developers to create their own browsing experience, it is highly customizable. Developers can present a full or partial view of web content directly in their apps by loading a view that leverages existing HTML, CSS, and JavaScript content or create their own view if they prefer. A third party could develop and offer an SDK that allowed developers to embed within their apps an in-app browsing interface using WebKit. This would mean that developers could offer an in-app browsing UI simply by integrating that SDK within their app.'
[5] See Apple developer Documentation- SFSafariViewController, accessed by the CMA on 2 July 2024.
[6] Note the browser needs to have opted into offering Custom Tabs – there are some browsers that do not offer this feature. If an app tries to call on a version of Custom Tabs that does not exist, the CMA understands that the app will then specify a 'back-up' option, which might be a different in-app browser or a dedicated browser app. The browser vendors that told the CMA they offer Custom Tabs are listed in paragraph 2.35. Note also that Custom Tabs is sometimes referred to as Android Custom Tabs (ACT).

allows users' preferences (including over privacy and security) to be carried across from the dedicated browser to the Custom Tabs IAB (eg from the Chrome browser to Chrome Custom Tabs, or from the Firefox browser to Firefox Custom Tabs).

### Webview

2.11 An app developer can build its own IAB on top of a readily available ('packaged') browser engine[7] called a webview. We refer to an IAB built this way as a 'webview IAB'. A webview IAB enables the native app developer to embed and render web content within its app in a highly customisable way and the app itself can interact with web content displayed via a webview – eg it can introduce features such as 'auto-fill' for login details and monitor user activity inside the IAB. The webview IAB uses an underlying browser engine to render web content. We refer to webview IABs as those where the browser engine is 'packaged' such that the app developer cannot access the core rendering engine – the developer only has access to certain APIs that sit on top of this engine. By contrast, we refer to 'bundled engines' (see below) as those where the app developers can access and modify the core rendering engine.

2.12 Webview IABs do not rely on any dedicated browser apps installed on the device and are distinct from those. The browsing experience within a webview IAB would not typically inherit any settings the users may have selected for their dedicated browser and would not sync with the user's browsing history. The fact that webview IABs are highly customisable and do not rely on a dedicated browser already present on the device means that they require more effort from app developers who need to develop and maintain the webview IAB alongside their app. Developers building webview IABs have more control over the level of security (and privacy) of the in-app browsing experience relative to remote tab IABs, for which such control is retained by the mobile browser the IAB links to.

2.13 Submissions from several parties indicate that webview IABs could have weaker security and privacy protections relative to remote tab IABs and dedicated browsers. These submissions suggest that, either through developer neglect or misuse, webview IABs may be more prone to security and privacy risks.[8] Indeed, webview IABs allow the modification of webpage content through JavaScript injection, which can be misused to modify webpages in ways designed to track or mislead users.[9] The OWA submitted that webview IABs do not inherit privacy- and security-protecting settings and extensions a user may have chosen in their

[7] A browser engine is the underlying technology which applications (including browsers) on devices use to transform web page source code into content with which users can engage.

[8] Responses to the CMA's information requests [✂].

[9] Unlike webview IABs, dedicated browsers and remote tab IABs are 'sandboxed', meaning they operate in a secure, isolated environment separate from the native app. This isolation prevents the modification of webpage content.

dedicated browser.[10] Separately, we have seen evidence from a browser vendor that indicates webview IABs can lack some functionality and be somewhat more exposed to security risks given the degree of developer flexibility and customisation they allow compared to general purpose browsers, which can increase the risk of personal data loss.[11]

2.14 To implement a webview IAB, app developers build on top of a specific webview that operating system providers or browser engines themselves offer:

(a) On iOS, Apple provides **WKWebView**, which is based on the WebKit browser engine.[12] There are no alternative webview options on iOS based on alternative browser engines due to the WebKit restriction – meaning the fact that Apple only allows its own browser engine on iOS devices – which we consider in 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine'.

(b) On Android, Google provides **Android WebView**, which is based on the Blink browser engine.[13] On Android, app developers can also choose alternative webview options. For example, the browser engine provider and browser vendor Mozilla has developed an alternative webview called GeckoView, which is based on the Gecko browser engine. In practice, it may be difficult for app developers to adopt alternative webviews such as GeckoView due to the setup of the Android platform – this issue is discussed in more detail in section 4.

2.15 We are not aware of any other webview options with significant reach. The two operating system webviews – WKWebView and Android WebView – are the two main webviews in mobile platforms. These webviews are integrated into iOS and Android, so their strong reach appears linked to Apple and Google's duopoly in mobile operating systems.[14]

### Bundled engine

2.16 For a bundled engine implementation of in-app browsing (referred to as a 'bundled engine IAB' as well as 'custom browser engine IAB' in this paper), the app developer builds upon its own custom (or forked) browser engine to create an IAB.

2.17 It is currently only possible to implement bundled engine IABs on Android in the UK. We understand that Meta is introducing a bundled engine IAB for the

[10] OWA: DMA interventions for In-App Browsing, paragraph 4.1.3, accessed by the CMA on 3 July 2024.
[11] [✂] response to CMA's information request [✂].
[12] Note that all alternative browsers on iOS (ie other than Safari) are built upon WKWebView. These browsers cannot access the core browser engine WebKit.
[13] Also referred to as Android System WebView. Note that some browsers on Android are built on Android WebView (eg DuckDuckGo).
[14] MEMS, Final Report, page 28.

Facebook app on Android.[15] We are only aware of three app developers that have bundled engine IABs [✂].[16]

2.18 On iOS, bundled engine IABs are not available due to the WebKit restriction that prevents any browser engines other than WebKit from being used by native apps (considered in 'WP2 – The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine'). This means that app developers must implement a webview IAB based on Apple's webview engine (WKWebView) or use a remote tab IAB that calls on Safari as the browser.

2.19 Implementing a bundled engine IAB is resource-intensive because the app developer essentially develops and maintains not only a browser built on top of a readily available engine (like in the webview IAB case, where the webview is usually integrated with the operating system) but also a browser engine in addition to the native app. We understand this means app developers have complete control over the performance, user experience, security and privacy within a bundled engine IAB. We also understand that the main benefits of offering a bundled engine IAB for app developers result from this control and the potential to incorporate a wider range of features in the IAB (see section 3 for more detail).

## How different stakeholders interact with IAB and their incentives

2.20 This section addresses how different stakeholder groups interact with in-app browsing, focusing on **OS providers, browser vendors, browser engines, app developers, web developers and users**.

### OS providers

2.21 Apple and Google essentially provide the platform for other stakeholders to offer and implement in-app browsing. They provide tools for stakeholders to develop IABs, set the rules for how IABs can be implemented and maintain documentation and guidance for app developers looking to implement an IAB in their respective operating systems.

2.22 Apple and Google provide the two largest mobile operating systems for mobile devices in the UK – iOS and Android. The CMA MEMS report found that Apple and Google have a de facto duopoly in respect of supplying mobile operating systems.[17] The two firms control the main gateways for digital content within their mobile ecosystems because they operate the largest app distribution channels

[15] Launching a new Chromium-based WebView for Android - Engineering at Meta (fb.com), accessed by the CMA 7 June 2024.
[16] Note of meeting with Google [✂].
[17] MEMS, Final Report, page 33 and 150.

(Apple’s App Store and Google’s Play Store) as well as the largest mobile browsers (Apple’s Safari and Google’s Chrome).[18]

2.23 Therefore, [✂] we understand that how Apple and Google enable and implement in-app browsing within their OSs can add value to their wider ecosystems. For example, their in-app browsers can allow for increased use of their first-party browsers and browser engines, and their first-party apps can use IABs to display web content. We have seen some evidence relating to the extent to which Apple and Google may be able to monitor and gather data from usage of IABs in their OSs. We intend to look further into this question.

### Apple

2.24 As explained above, Apple mandates that all IABs on iOS are based on the WebKit browser engine via a requirement in the App Store Review Guidelines.[19] Apple provides two different implementations to apps: a remote tab IAB based on Safari called SFSafariViewController and webview IABs based on WKWebView.[20]

2.25 Apple told us that its approach to enabling the implementation of in-app browsing on iOS strikes an appropriate balance between developer freedom and discretion over the in-app browsing experience while still affording users choice.[21] Apple produces guidance for developers looking to implement IAB and recommends different implementations based on the use case (ie what the app developer wants to get out of its IAB product). More specifically:

(a) Apple recommends developers use SFSafariViewController when they want to display websites within the app without sending users to Safari. Apple suggests it is best used for ‘interactive web experiences on websites you don’t own’.[22]

(b) Apple recommends developers use WKWebView if they need to customise or control the display of web content or to interact with the content itself.[23] Further, it recently introduced a feature called App-Bound Domains (ABD) to

[18] Indeed, as of 2021, the combined share of supply for Apple’s and Google’s browsers on mobile devices in the UK is around 90%, with Safari having a share of close to 50% and Chrome around 40%. MEMS, Final Report, page 141 and 150.
[19] Clause 2.5.6 of the App Store Review Guidelines specify that ‘apps that browse the web must use the appropriate WebKit framework and WebKit Javascript’. App Store Review Guidelines - Apple Developer, accessed by the CMA 19 April 2024.
[20] See ‘WP2 – The requirement for browser apps on iOS to use Apple’s WebKit browser engine’ for more detail on the WebKit restriction. Also, see Should I use WKWebView or SFSafariViewController for web views in my app? - Discover - Apple Developer, accessed by the CMA 11 June 2024.
[21] Apple’s response to CMA’s information request [✂].
[22] Should I use WKWebView or SFSafariViewController for web views in my app? - Discover - Apple Developer, accessed by the CMA 7 May 2024.
[23] Should I use WKWebView or SFSafariViewController for web views in my app? - Discover - Apple Developer, accessed by the CMA 7 May 2024.

enhance user privacy in webview IABs.[24] ABD is an opt-in feature for native app developers using WKWebView that limits how much visibility the app has over user activity in the IAB. Apple claims ABD may protect users from 'intrusive' tracking methods.[25]

2.26 Apple told us it [✂]. However, it told us traffic to SFSafariViewController helps Apple to improve the performance, security and web compatibility of Safari and WebKit [✂].[26]

2.27 We asked Apple to provide any internal documents related to Apple's strategy on how in-app browsing is implemented on iOS and any changes that Apple is considering for how in-app browsing may be implemented in the future. Apple did not identify any documents that were relevant to this request, [✂]. Apple explained that it publishes materials and makes available tools to deliver in-app browsing experiences and to assist developers in determining which option for in-app browsing would be most appropriate for their needs.[27]

### Google

2.28 Google told us that in-app browsing is becoming increasingly 'common' and that IABs are beneficial to users and app developers on Android.[28] Google maintains and updates Android WebView and the Custom Tabs system, as well as Google's own version of Custom Tabs – Chrome Custom Tabs.[29] Google allows third-party webviews (based on browser engines other than its own browser engine Blink) and bundled engine IABs to be used for in-app browsing on Android. Google also provides guidance for developers looking to implement IABs:

(a) Google recommends that app developers use Custom Tabs for opening third-party web content. Google advises app developers that Custom Tabs offer a 'better user experience' than opening in an external browser because users remain within the app.[30]

[24] Apps that opt-in to ABD can specify up to ten domains for which they are able to use certain features that may put user privacy at risk. Apple describes these features as 'intrusive ways to communicate with known trackers seeking to collect and aggregate personal information about users'. These features include JavaScript injection, custom style sheets, cookie manipulation and event handlers. They can reveal to the app which images a user pauses on, what content they copy and paste and which sections of pages they reach while scrolling. Source: https://webkit.org/blog/10882/app-bound-domains/.

[25] Source: https://webkit.org/blog/10882/app-bound-domains/.

[26] Apple's response to CMA's information request [✂].

[27] Apple's response to CMA's information request [✂].

[28] For example, Google claims that IABs can remove friction and streamline user journeys. They also enable app developers to offer a wider range of functionality and innovative experiences. Note of meeting with Google [✂].

[29] For example, Google recently changed the conditions for updates to WebView in response to developer feedback to increase the proportion of devices on which WebView is up to date. Google's response to CMA's information request [✂].

[30] Overview of Android Custom Tabs | Web on Android | Chrome for Developers, accessed by the CMA 13 May 2024; Web-based content | Views | Android Developers, accessed by the CMA 7 May 2024.

(b) Google advises that developers only use Android WebView for displaying first-party web content or if the developer needs to inject JavaScript directly from their app. This is because web content ‘may not be displayed in the way the developer intended’ in Android WebView.[31] This is also because webview IABs do not share state with the browser (eg users’ browsing history is not shared between the IAB and their dedicated browser) and they are more costly to maintain.[32]

2.29 Evidence from Google suggests it was [✂]. Google has submitted that webview IABs [✂].[33] Google [✂][34] told us that it will continue to work on improvements to [✂] Android WebView.[35]

2.30 Google submitted [✂].[36]

2.31 We have seen evidence that suggests specific features of Chrome Custom Tabs can drive advertising conversion for app developers. [✂].[37]

## Browser vendors

2.32 In this section we summarise what we have heard from browser vendors on why they offer IABs (or not) and what they typically monitor in relation to in-app browsing.

2.33 As explained above, browser vendors can offer their dedicated browser to be called upon by a native app on Android as a remote tab IAB. On iOS, this is not possible because only Safari can be used as a remote tab IAB. Note that browser vendors generally do not interact with webview or bundled engine IABs because these implementations are controlled by the app developer and do not rely on or link to a dedicated browser (although some browser vendors have considered offering alternative webviews to app developers, which is explained in the browser engine providers section below).

2.34 On Android, browser vendors can choose to offer an implementation of Custom Tabs, which is set by most native app developers to call on the user’s default browser. Therefore, by offering Custom Tabs on Android, browser vendors choose to further support their users for whom they are the default, displaying web content for these users within apps (as well as in their dedicated browser app).

[31] Web on Android | Articles | web.dev, accessed by the CMA 13 May 2024.
[32] Overview of Android Custom Tabs | Web on Android | Chrome for Developers, accessed by the CMA 13 May 2024.
[33] Google’s response to CMA’s information request [✂]; Google’s response to CMA’s information request [✂].
[34] [✂] See Google’s response to CMA’s information request [✂].
[35] Google’s response to CMA’s information request [✂].
[36] [✂]; Google’s response to CMA’s information request [✂].
[37] Google’s internal document [✂].

2.35 Many browser vendors we have gathered evidence from told us that they offer a version of Custom Tabs on Android:

(a) Chrome (Google);

(b) Firefox (Mozilla);

(c) Vivaldi;

(d) Brave; and

(e) Edge (Microsoft).[38]

2.36 Each browser vendor's version of Custom Tabs is based on the same browser engine as their respective dedicated browsers. Most of these browser vendors told us that they did not consider offering Custom Tabs to be particularly costly or resource-intensive[39] and that they consider their remote tab IAB as a feature of their wider competitive offering to users. From stakeholders' submissions, we understand there are two main benefits of offering remote tab IABs (which are explained in more detail in paragraph 4.8):

(a) Browser vendors can support their users more effectively. Browser vendors want to be able to display web content and offer their features (eg tracker blockers) to their users for in-app browsing, as well as in the dedicated browser app.

(b) Offering a remote tab IAB increases the time users spend on their browser, which indirectly benefits the browser vendor. This benefit materialises via feedback received on any issues encountered in the browser and via web compatibility effects (for more detail on indirect network effects arising from web compatibility, see 'WP1 – Nature of competition in the supply of mobile browsers and browser engines', paragraphs 2.45 to 2.54).

2.37 One browser vendor told us that while offering remote tab IAB would further support its users, it has a relatively small team so this has not been an immediate priority. However, it planned to introduce a version of Custom Tabs later this year.[40]

2.38 Another browser vendor submitted that it does not offer a remote tab IAB because it did not see what immediate benefits could be gained from it. While a remote tab IAB could be valuable to its users, there was no way to monetise IAB traffic. It also

[38] Google also submitted that as of May 2021, it was aware of several additional browsers offering Custom Tabs. Google's response to CMA's information request [✂].
[39] Responses to CMA's information requests [✂].
[40] Note of meeting with [✂].

said it may look into offering a remote tab IAB in the future and are exploring what value it could derive from it.[41]

2.39 Browser vendors told us they do not get direct monetisation from offering IABs. We understand that browser vendors mainly monetise their dedicated browsers through search revenue-sharing agreements, but this is not possible in IABs in which users do not search the web.[42] IABs usually lack a search bar that would enable users to search (and, in turn, that would generate revenue for the browser vendor). One browser vendor told us it could be possible to monetise traffic to its IAB,[43] but we are not aware of any browser currently doing this.

2.40 We understand that browser vendors can monitor user activity and time spent in their remote tab IAB and some do track this (see paragraphs 2.64 and 2.65 for more detail).[44] Some browser vendors which market themselves as privacy-oriented (eg Vivaldi and [✂]) do not track time spent in their remote tab IAB.[45]

2.41 Some browser vendors appear less engaged on the topic of in-app browsing than others. As a browser vendor and the provider of WebView on Android, we understand that [✂]. Indeed, Google recognises the growth of IAB and is very active in this area.[46] Other browser vendors may be less cognisant of the size of IAB and, in turn, may be less incentivised to develop a competitive offering on Android. For example, browser vendors that do not track or monitor time spent or user activity may have limited awareness of the size of their Custom Tabs IAB relative to their dedicated browser apps.

### Browser engine providers

2.42 Browser engine providers might choose to provide a version of their browser engine for native apps to incorporate within an in-app browser – as per the above, alternative browser engines can currently only be used for IABs on Android. From the evidence we have seen so far, the main incentive for browser engine providers to do this would be that the additional traffic from IABs creates benefits for the browser engine providers that are explained in paragraph 2.43 below. The main beneficiary of this effect may be Mozilla, which provides an alternative browser engine to Blink on Android.

2.43 More specifically, evidence we have seen so far suggests that browser engine providers could benefit from increased usage of their browser engine that takes place via in-app browsing. This benefit may arise in three ways:

[41] Note of meeting with [✂].
[42] Responses to CMA's information requests [✂].
[43] Note of meeting with [✂].
[44] [✂] response to CMA's information request [✂].
[45] Note of meeting with [✂]; Note of meeting with Vivaldi [✂].
[46] Google's response to the CMA's information requests [✂].

(a) **Feedback from usage:** The browser engine provider receives feedback from any issues and problems encountered in the browser engine, which allows it to improve the engine's performance and security.

(b) **Web compatibility:** Usage of the browser engine is a signal to web developers, who are more likely to develop their sites to be compatible with that browser engine. Although, the effect of this channel may be relatively small, especially given that we understand web compatibility has become less of a problem in recent years.[47]

(c) **Incentives for improvement efforts:** Google submitted that increased traffic 'may increase a browser developer's incentive to invest in performance and security improvements generally in order to remain competitive and retain users.'[48]

2.44 Mozilla has offered a webview called GeckoView for app developers to incorporate in their apps and build upon for in-app browsing.[49] A different browser vendor has considered offering a similar product in the past that would have been based on the chromium browser engine, but ultimately decided this would not be 'rewarding' for its business.[50] We understand that benefits to Mozilla (and particularly its browser engine Gecko) may also come from usage of remote tab IABs based on Gecko. We are only aware of Firefox Custom Tabs to be based on the Gecko browser engine on Android.

2.45 We understand from a recent Mozilla submission that competing as a replacement for the system webview engine on Android for browser developers is something they are not prioritising for a number of reasons. Mozilla would instead prefer app developers to call upon Firefox as a remote tab IAB in instances when it was the user's default. This would use Mozilla's own engine and give the browser vendor (rather than the app developer) greater control over the user experience (see section 4 for more detail on Mozilla's view).[51]

### App developers

2.46 App developers implement IABs within their apps to allow users to view web content within the app in a convenient manner. IABs enable app developers to expand the functionality of their app, enhance user engagement (because the user

[47] For example, the analysis from the web developer research conducted as part of this market investigation suggests the time spent by developers on ensuring web compatibility has declined in recent years and there were few mentions of this issue. [✂]. Additionally, Google submitted that increased traffic to a browser does not necessarily affect its compatibility 'as browsers follow a common set of web standards'. Source: Google's response to CMA's information request [✂].

[48] Google's response to CMA's information request [✂].

[49] See Geckoview - GeckoView (mozilla.github.io), accessed by the CMA 10 June 2024.

[50] Note of meeting with [✂].

[51] Mozilla's response to CMA's information request [✂].

is kept within the app when viewing web pages) and support their advertising models.[52]

2.47 App developers’ incentives vary depending on their app and business model. For example, based on evidence we have seen so far, social media apps and other apps that generate revenue from advertising are incentivised to use IABs to enhance engagement and user time spent within their app. Some apps might invest more in developing and adding features to their IABs – for example, Pinterest offers a ‘Pin’ feature that allows users to directly save content to their Pinterest account from the IAB. Other apps may have less sophisticated requirements for in-app browsing, focussing on a convenient way to display the web to their users. Some apps may not have a requirement for displaying third-party web content in the app at all – these apps choose to send users to an external browser.[53]

2.48 For app developers that choose IABs, the implementation type they choose depends on the following:

(a) App developers may choose to implement a **remote tab IAB** because they have less need to customise or control the IAB. This implementation is also less costly to implement and requires less effort from the app developer.

(b) App developers may choose to implement a **webview IAB** if they want to have more control over the IAB. App developers can interact with web content in a webview IAB – for example, they can inject JavaScript to track user activity or to protect users from malicious weblinks.

(c) App developers may choose to implement a **bundled engine IAB** if they want complete control over the IAB, including security, privacy, performance and user experience. As explained above, in this case the app developer would need to maintain not only their native app and the webview it incorporates, but also the browser engine it is based on.

2.49 We understand that IABs are often used for advertising purposes. IABs can support in-app weblinks displayed by advertisers because IABs enable users to view and complete a purchase in an advertiser’s webpage after clicking on a promoted link without leaving the original app. App developers also submitted that using an IAB allows them to collect data on users’ web activity. This data enables

[52] For example, Google submitted that IABs allow developers to ‘expand the functionality of their apps because they can introduce features and logic across different OSs via the web’. Google also told us that ‘choice of in-app browsing is important for app developer differentiation; app developers can build innovative in-app browser experiences. Note of meeting with Google [✄].

[53] Note some app developers may still use IAB technology to display first-party web content - any app developers use it for displaying terms and conditions, settings, FAQs, and the login page without having to interrupt the user experience. We do not consider this use case to align with ‘browsing the web’ in an app, which is the focus of this paper.

the app to personalise and enhance their advertising and recommendation models.[54]

2.50 Apps may opt for different implementations of IAB depending on what will most benefit their advertising business. We understand that some app developers face a trade-off – they have greater control and visibility over user activity in webview IABs but can deliver better outcomes for their advertisers (eg higher ad conversion rates) if they send users to remote tab IABs or dedicated browsers.

(a) An app developer told us better outcomes can be achieved from ads that send users to remote tab IABs or out of the app to a dedicated browser. This may be because users remain logged in and have a more stable experience in remote tab IABs and dedicated browsers relative to webview IABs. [✂].[55]

(b) One app developer told us that on iOS, it prefers to use a webview IAB to open links that are not promoted content because it can personalise the user interface more than it can with SFSafariViewController. For advertising links, however, this app developer uses SFSafariViewController. The app developer has no visibility over what users do when they click on the advertising links but submitted that its most important advertisers prefer to have their websites opened on a remote tab IAB over a webview because the advertisers can monitor traffic coming from the app better in dedicated browsers and remote tab IABs such as Chrome and Safari.[56]

(c) One app developer told us it chooses webview IABs on iOS and Android for links within ads. For non-ad links on Android, it primarily uses Chrome Custom Tabs. It avoids using a remote tab on iOS because SFSafariViewController does not provide any information to the host app about the in-app browsing session.[57]

### Web developers

2.51 Web developers develop content that might be rendered within IABs.[58] Therefore, the quality and functionality of an IAB impacts users' experience of a developer's website, which in turn may affect web developers' businesses.

2.52 We have heard evidence that web developers care about the quality of the web as a platform for their business. Parties told us that IABs can distort and disrupt users' experience of a website, which may reflect poorly on the web developers

[54] Responses to CMA information requests [✂].
[55] Note of meeting with [✂].
[56] Note of meeting with [✂].
[57] [✂] response to CMA information request [✂].
[58] Web and web app developers may have a view on how their web content is best presented. If they think webview IABs will not present their content optimally, they could try and direct users to a different browser. Advertisers may also advocate for native apps to open links to their website in a dedicated browser (ie if they think this would mean the user is more likely to be logged in).

themselves due to low user awareness of being in an IAB.[59] Additionally, IABs can complicate web development efforts, which may hold some web developers back from offering new features to ensure their websites work well regardless of how they are being accessed. For example:

(a) **Microsoft** submitted that web developers care about the reliability of the web as a platform. Microsoft submitted that some IABs can hamper web developers' ability to use the web as a reliable platform.[60]

(b) **OWA** published that IABs create a poorer user experience for viewing the web, which affects developers' businesses because users attribute blame for any issues to the website owner.[61]

(c) **A web developer** who participated in our research undertaken for this MI submitted that IABs contribute to there being 'a lot more to think about' when developing new features for the Android platform.[62]

### Users

2.53 The evidence set out below demonstrates that users have no effective choice and limited control over which browser is used for in-app browsing and may not be aware of which browser they are taken to, when they use in-app browsing. App developers will decide how IAB is configured and made available to a user within an app. In some instances, on iOS and Android devices, an app developer may choose to call upon the user's default browser for in-app browsing but this is not always the case. There are few differences between the two operating systems in relation to the user's default browser choice:

(a) On iOS there is no variation between SFSafariViewController and webview implementations of in-app browsing. A user's default browser choice makes no difference to the remote tab implementation of IAB, which always calls upon Safari for SFSafariViewController. The webview implementation of in-app browsing, similarly to Android, would not call on a user's default browser choice, as webview IABs do not rely on any browser apps installed on the device.

(b) On Android this may vary between the Custom Tabs, the webview and bundled engine implementations of in-app browsing. In contrast to iOS, by default, as provided by the OS provider, the Custom Tabs implementation of in-app browsing will run the user's default browser in Custom Tabs mode.

[59] User awareness of IABs is discussed in more detail in sections 3 and 4. Note of meeting with OWA [✂].
[60] Note of meeting with Microsoft [✂].
[61] See: OWA - DMA Interventions - In-App Browsers (To Publish) (open-web-advocacy.org), accessed by the CMA on 2 July 2024.
[62] CMA and Jigsaw, MI web developer research report, 13 May 2024.

However, this can be altered by the app developer, who can specify a version of Custom Tabs – for example, if 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. Similarly to iOS, the webview or bundled engine implementations of in-app browsing on Android do not rely on any dedicated browser apps installed on the device and will not therefore call on the user's default browser.

2.54 There are two possible ways for users to control in-app browsing:

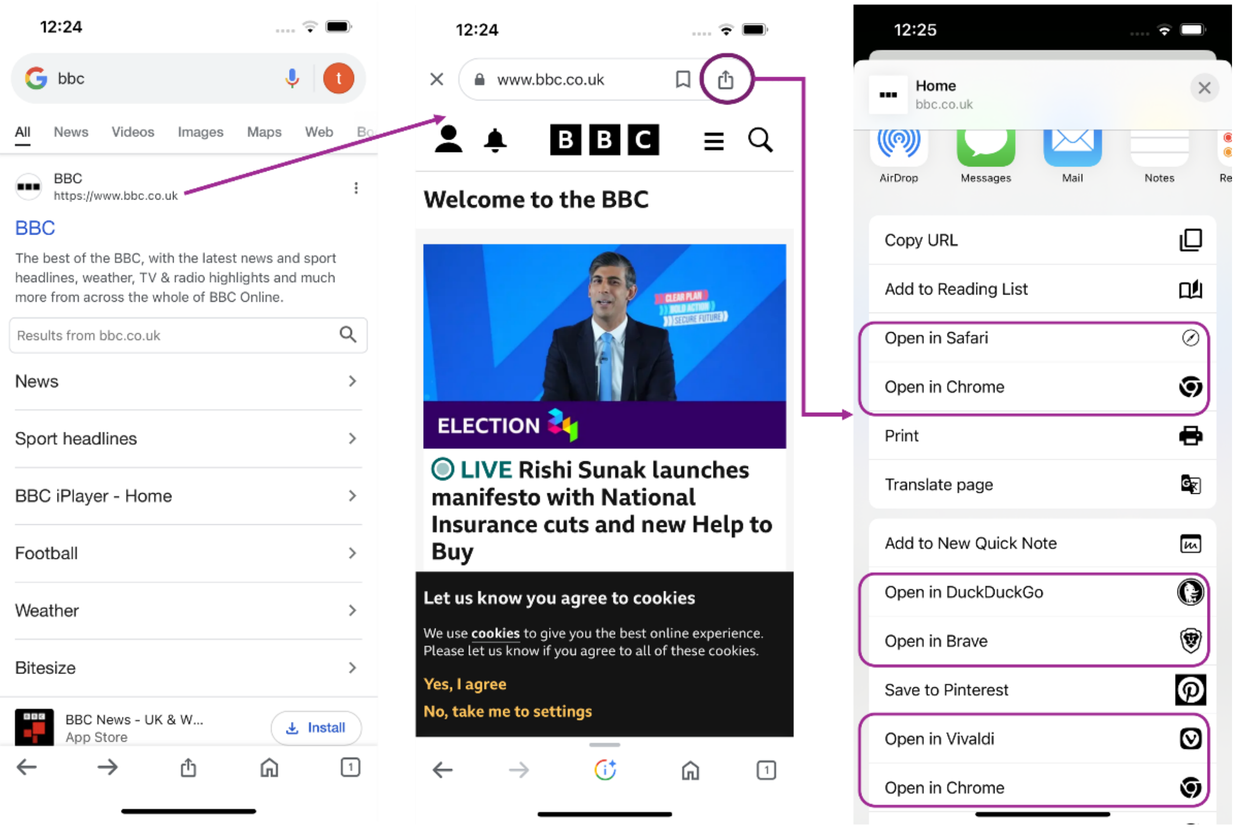
(a) at the point of using a native app; and

(b) at the device settings level.

2.55 App developers can provide users with an option to open weblinks in a separate browser app instead of the app's built-in IAB, at the point of using the native app. This option can be accessed by navigating the 'IAB choice menu' from within the IAB, which allows users to switch from in-app browsing to their default dedicated browser or other installed browsers. This feature is available across different IAB implementations on Android and iOS. An 'IAB choice menu' may display a list of all browsers installed on the device where the link could be opened (see Figures 2.2, 2.3. and 2.4). However, users may not be aware of this option as it is usually placed discretely in the top corner of the screen. Apps can be configured in different ways by app developers, and they are able to customise the in-app browsing interface.[63]

[63] [✂] response to the CMA's information request [✂].

**Figure 2.2: Option to open webpage in separate browser in the Google Search app for iOS ('IAB choice menu', 'share' icon).**



BBC

The best of the BBC, with the latest news and sport headlines, weather, TV & radio highlights and much more from across the whole of BBC Online.

Welcome to the BBC

LIVE Rishi Sunak launches manifesto with National Insurance cuts and new Help to Buy

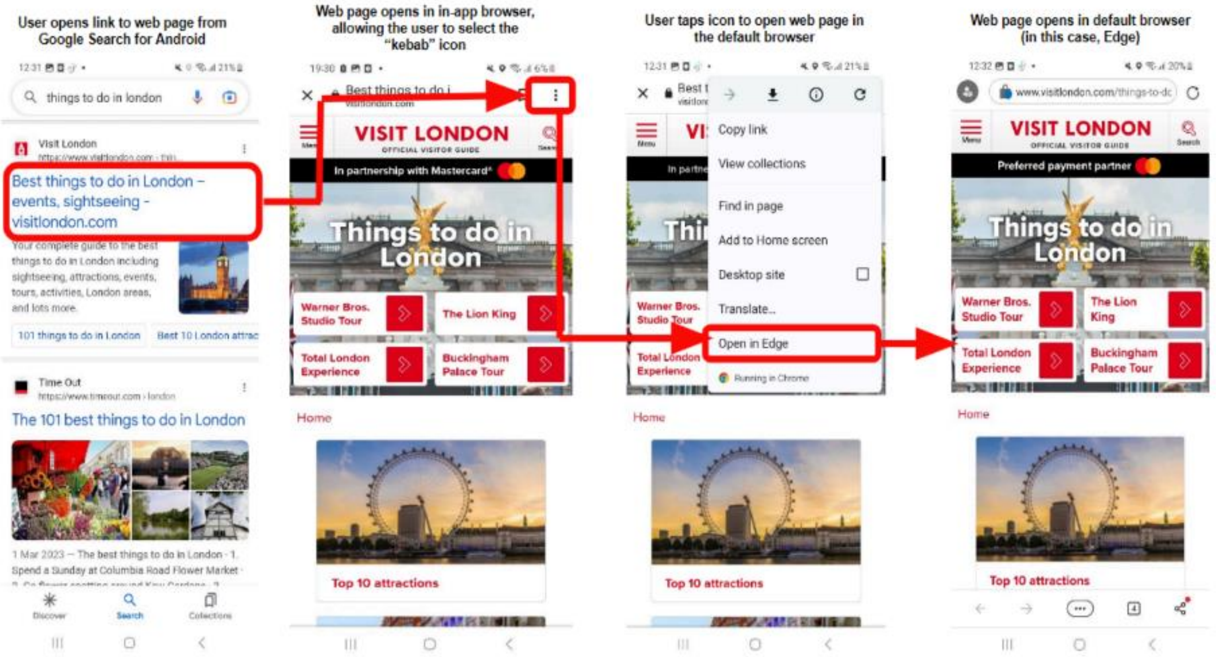
Let us know you agree to cookies

We use cookies to give you the best online experience. Please let us know if you agree to all of these cookies.

Yes, I agree

No, take me to settings

**Figure 2.3: Options to open webpage from Google Search app on Android in separate browser ('IAB choice menu', vertical ellipsis icon).**



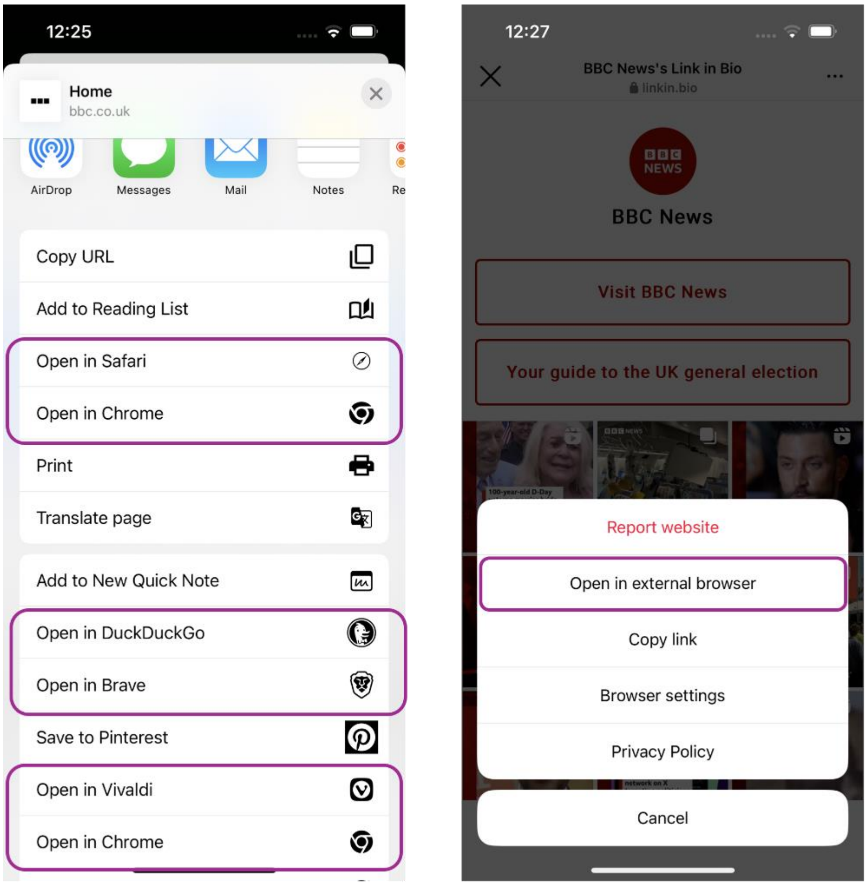
User opens link to web page from Google Search for Android

Web page opens in in-app browser, allowing the user to select the "kebab" icon

User taps icon to open web page in the default browser

Web page opens in default browser (in this case, Edge)

**Figure 2.4: Opening link in external browser on Google Search app (left) and Instagram (right) on iOS.**



**Opening link in external browser on Google Search App**

**Opening link in external browser on Instagram**

2.56 Users may, but are not usually able to, turn in-app browsing off for a particular app at a device settings level and have limited control over in-app browsing experience.[64]

(a) On iOS, Apple submitted that there is no centralised set of controls to disable or enable in-app browsing for all native apps on iOS devices. The choice of whether and how to enable in-app browsing functionality is dictated by the app developer, not Apple or the user.[65] For example, Gmail on iOS either uses SFSafariViewController or prompts the user to choose which browser to open the link, including the default browser (see 'WP5 – The role of choice architecture in the supply of mobile browsers'), giving users some degree of control of in-app browsing when using Gmail app. However, we found evidence that users can disable IAB on the Guardian app.[66]

[64] An example of an app where users can disable IAB for external links on iOS and Android is the Guardian. GMG response to the CMA's information request [✂].
[65] Apple's response to the CMA's information request [✂].
[66] GMG response to the CMA's information request [✂].

(b) On Android, in-app browsing user control is enabled for Google's Gmail and Google Search apps, which use the Custom Tab API for in-app browsing. Google said that adding opt-out of in-app browsing is recommended by Android Custom Tabs as best practice.[67] Microsoft submitted that it is possible to disable non-advertising in-app browsing on LinkedIn on Android, but not on iOS.[68] This seems to be related to LinkedIn using primarily [✂] for non-advertising links on Android and [✂] for advertising in-app browsing on Android and all in-app browsing on iOS.[69]

## The significance of IAB as a proportion of overall mobile browsing

2.57 Given the sparsity of available data, it is difficult to reliably estimate overall time spent within IABs on iOS and Android. Based on the evidence we have seen so far, while no stakeholders have a full picture of time spent browsing web content within an IAB, it appears likely to be significant. This section attempts to give an indication of the prevalence of in-app browsing, compiling evidence from OS providers, browser vendors and app developers.

### Evidence from OS providers

2.58 [✂][70] Apple has some visibility over the number of apps that query in-app browsing APIs on iOS using SFSafariViewController. [✂][71] For example, Apple submitted that [over 20,000] non-browser apps use SFSafariViewController on iOS. Apple has explained that the data does not allow Apple to know whether apps that were identified as incorporating calls to the SFSafariViewController API have actually used that API during a given period. Additionally, apps may call on the SFSafariViewController in order to deliver web-based content for discrete uses, such as linking to a privacy policy, support page, or when prompting users to undertake log-in or account authentication flows. The fact that an app includes calls to SFSafariViewController does not therefore indicate whether in-app browsing is a significant feature of the app or included for an edge case or something in between.[72]

2.59 Google has some visibility over the usage of Android WebView and its version of Custom Tabs (Chrome Custom Tabs). It has no visibility on usage of alternative webviews (eg GeckoView) or different browser vendors' implementations of Custom Tabs.[73]

[67] Google's response to the CMA's information request [✂].
[68] Microsoft's response to the CMA's information request [✂].
[69] Note of call with Microsoft [✂].
[70] [✂] response to the CMA's information request [✂].
[71] Apple response to CMA's information request [✂].
[72] Apple response to the CMA's information request [✂].
[73] Google's response to CMA's information request [✂].

2.60 Evidence submitted by Google was not conclusive on time spent by Android users viewing web content via in-app browsing technology (ie viewing web content displayed via a webview or Custom Tabs) compared to dedicated browser apps. In any event, this does not reveal much about the relative time users spend in-app browsing (ie viewing third-party web content within a native app) because a large proportion of this time is likely to be spent viewing first-party web content within native apps (eg viewing a shopping app's search results, or search results within the Google Search app).[74]

2.61 Evidence from Google was not conclusive on whether remote tab IABs are used more or less than webview IABs (based on Android WebView). This evidence also indicates that [a significant amount] of the time spent in remote tab IABs on Android is attributed to the Google Search App, which uses Chrome Custom Tabs to display web pages.[75]

2.62 Google estimates that several native apps on Android may account for the majority of the time spent on web content rendered by Android WebView. [✂].[76] As explained above, a large portion of this time might be spent viewing first-party web content owned by the app rather than browsing third-party web pages.

### Evidence from app developers

2.63 Native apps that offer an in-app browsing experience can often track how much time users spend in the IAB.[77] It is unclear how many apps choose to do this, and each app will see different levels of usage. This means an individual app will have little visibility over the prevalence of in-app browsing relative to dedicated browsing. Evidence we have seen on time spent within a given app's IAB ranged from less than 1% to as high as 15% of overall time spent in the app.[78]

### Evidence from browser vendors

2.64 Based on evidence we have seen so far, browser vendors offering a remote tab IAB can collect data on how much time is spent within it. Google submitted that it could not provide accurate data to reflect time spent in Chrome Custom Tabs across the population,[79] but internal documents indicate that it has [✂].[80] Microsoft submitted an estimate that in the past year, the global daily number of instances where Edge Custom Tabs was invoked has been as high as nearly [✂].[81]

[74] Google's response to the CMA's information request [✂].
[75] Google's response to CMA's information request [✂]; Google response to CMA's information request [✂].
[76] Google's response to CMA's information request [✂].
[77] [✂] response to the CMA's information request [✂].
[78] Responses to the CMA's information requests [✂].
[79] Google's response to the CMA's information request [✂].
[80] Google's internal document [✂]; Google's response to the CMA's information request [✂].
[81] Microsoft's response to CMA's information request [✂].

2.65 Many browser vendors do not track time spent in Custom Tabs.[82] Browser vendors have no visibility over in-app browsing taking place through a webview IAB or other remote tab IABs that are not using their browser.

[82] Responses to CMA information requests [✂].

# 3. Market definition

3.1 A separate paper published in this market investigation entitled 'WP1 – Nature of competition in the supply of mobile browsers and browser engines' considers the extent to which certain native apps, particularly those that support in-app browsing, exert a competitive constraint on standalone mobile browsers and should therefore be considered as part of the same relevant market. This section sets out our emerging thinking on market definition in relation to in-app browsing, building on the 'Nature of competition' working paper, by considering whether different IAB implementations may be considered as separate markets or separate segments of an IAB market.

3.2 The focal products and services that we are considering in this assessment are:

(a) Remote tab IABs offered by browser vendors.

(b) Webview IABs offered by browser engine providers (albeit built upon and customised by app developers).

(c) Bundled engine IABs built by app developers.

3.3 We are considering the extent of the constraints between the following:

(a) Different IAB implementations – meaning whether different IAB implementations (ie remote tab, webview and bundled engine IABs) are in separate markets or segments of a broader market.

(b) IABs and dedicated browsers – meaning whether for each OS (ie iOS and Android) there is a separate market for the supply of IABs or whether IABs form a segment of the wider mobile browser market on that OS.

(c) Search apps that include IABs and dedicated browsers – meaning whether there are separate markets for search apps that incorporate an IAB and dedicated browsers (eg the Google Search App, which uses Chrome Custom Tabs to display web pages, and dedicated browsers such as Firefox).

**Different IAB implementations**

3.4 In this section we consider the extent to which different IAB implementations exert a constraint on each other.

3.5 From the users' perspective, we note that users can switch between different versions of remote tab IABs on Android. That is, users may choose a dedicated browser as their default, and this browser will appear as a Custom Tab within certain apps that have chosen to rely on the user's default. However, users cannot choose between different IAB implementations (which is a choice that resides with the app developer) and the evidence we have seen suggests they are not very engaged or even aware of different IAB implementations (see sections 4 and 5 'Evidence from consumer research' for evidence on user awareness of IABs). We therefore consider that the competitive process for IABs relies on the choices of app developers. Indeed, app developers represent the demand side for 'inputs' (provided by browser vendors and browser engines) that they use to incorporate IABs into their apps, but developers also internalise the demand that they expect from users of their native app.

3.6 There is evidence of demand-side substitutability from the app developers' perspective between different IAB implementations within each operating system. App developers choose IAB implementations based on their features, cost and the specific use case. We understand that switching between different IAB implementations does take place but it can require effort from app developers. It may be costly to develop a webview IAB and bundled engine IABs are costly, such that we understand only particularly large app developers could afford to switch towards implementing them (see paragraphs 2.16 to 2.19 for more detail).

3.7 App developers use webview, remote tab or bundled engine IABs for similar use cases. For example, as explained in section 2, the CMA has heard from app developers that use different IAB implementations to support advertising within their apps. Some app developers we heard from have switched, or considered switching, between implementations. For example:

(a) One large app developer submitted that its Android apps switched to Chrome Custom Tabs from Android WebView due to the security benefits of Chrome Custom Tabs.[83]

(b) [✂] told us that it has switched from a webview IAB to a bundled engine IAB on its Android [✂] app. [✂]. [✂] did not consider this switch to a bundled engine IAB in its [✂] app to be costly or resource intensive.[84]

3.8 However, there is also evidence that app developers do not always consider the different IAB implementations to be substitutes because they offer different features and varying levels of customisability that may be less suitable for certain use cases. For example:

(a) Google told us Custom Tabs cannot meet all app developers' needs for development and innovation, which is why it offers Android WebView alongside Custom Tabs. [✂][85]

[83] [✂] response to CMA's information request [✂].
[84] Note of meeting with [✂].
[85] Note of meeting with Google [✂].

(b) One app developer told us it would prefer to use just one IAB implementation for all weblinks within its iOS app. However, it uses both SFSafariViewController and WKWebView. This is because the app developer's advertising business customers prefer SFSafariViewController for opening ad links, but the app developer has more customisability over its WKWebView IAB, which is used for all other in-app weblinks.[86]

(c) [✂] submitted it considers that IABs are more suitable for apps that require more integrated experiences with both native and web content. [✂] also submitted that cloaking protections are only possible with webview IABs.[87] [✂].[88]

3.9 When considering what browser vendors supply to app developers for the purpose of building or incorporating IABs, it appears that supply-side substitution between webview, remote tab and bundled engine IABs is limited by the technical set-up of iOS and Android and the respective OS providers' policies. More specifically:

(a) Third-party browser vendors who offer remote tab IABs have the option to offer alternative webview IABs, but this is only possible on Android in the UK. Further, we are not aware of a browser vendor which is actively investing in offering a webview product on Android. We understand this is likely to be because there are technical issues with offering an attractive third-party webview on Android that relate to the default position of the OS-provided Android WebView (see section 5 for more detail).

(b) On iOS, Apple's policies on in-app browsing mean that only Apple can provide IABs – there is therefore no scope for any party other than Apple to substitute between these products (see section 4 for more detail).

3.10 In summary, IAB implementations appear to lie on a spectrum for app developers from low cost and limited customisability (remote tab IABs) to more cost and customisability (webview IABs) and then to higher cost and complete customisability (bundled engine IABs). This means that they support a range of use cases for app developers, but there does not seem to be a straightforward or unique classification between different use cases, customers and their corresponding preferences for IABs.

3.11 As a result, for the purpose of our current analysis, we have considered different IAB implementations as likely part of the same market. However, we are still considering this issue and will take into account the important differences between the IAB implementations as we continue with our assessment of this market.

[86] Note of meeting with [✂].
[87] Note of meeting with [✂].
[88] [✂] response to CMA's information request [✂].

### IABs and dedicated browsers

3.12 In 'WP1 – Nature of competition in the supply of mobile browsers and browser engines', we set out the evidence in relation to competition between IABs and dedicated browsers. Our emerging thinking is summarised below.

3.13 The extent to which native apps with IABs can be considered as substitutable to standalone mobile browsers from a demand-side perspective appears limited. Indeed, native apps with IABs have a fairly specific use case, with IABs often used to view just one or two websites before returning to the native app, whereas dedicated browsers are used to navigate the web. We do acknowledge that users can choose to leave IABs to open a dedicated browser and view the same web content in that browser, which indicates there may be some indirect constraint between the two products.

3.14 Further, IABs typically lack certain browser functionalities. For example, they generally do not have a URL bar or a search function, cannot access browsing history or sync it with the user's history on a standalone browser, and do not have password saving features or tabs.

3.15 As explained in 'WP1 – Nature of competition in the supply of mobile browsers and browser engines', evidence from browser vendors on the extent to which standalone browsers and IAB exert a constraint on each other is mixed, with some browser vendors expressing the view that IABs do not compete closely with dedicated browsers and others that IAB can pose a constraint to standalone browsers under some circumstances.

3.16 From a supply-side perspective, we consider there to be some substitutability between native apps incorporating an IAB and standalone browsers. For example, browser vendors offer both remote tab IABs and dedicated browsers on Android and we heard from browser vendors that offering a remote tab IAB on Android alongside their browser was not too costly.[89]

3.17 Based on evidence seen so far, we consider it likely that IABs should be regarded as either separate markets to standalone mobile browsers, or a sub-segment of a wider mobile browsing market. We keep considering this topic further as our analysis progresses.

### Search apps with an IAB and browsers

3.18 In section 3 of 'WP1 – Nature of competition in the supply of mobile browsers and browser engines', we considered the extent to which mobile search apps exert a constraint on mobile browser apps. Based on evidence seen so far, our emerging

[89] Responses to the CMA information requests [✂].

view is that there is some evidence pointing towards search apps likely being in a separate market than mobile browsers – as mobile browsers and search apps are distinct products and supply-side substitutability is limited between the two. However, we are continuing to consider whether they are more appropriately considered to be part of the same market.

3.19 When it comes to search apps exerting a constraint on IAB specifically, it is worth noting that the two products may interact from a technical perspective. For example, we understand that when a user clicks on a link in the Google Search App, web content is displayed for the user in Chrome Custom Tabs (see section 4 for more detail). In this case, IABs may therefore be considered as a feature of a search app that is incorporated into the app to display web content.

3.20 On the relationship among search apps, IABs and standalone browsers, we have heard from a browser vendor that if search apps do not launch the user's browser of choice (ie the 'default' browser) to show the search results, it is reasonable to expect that users might use that browser less frequently over time and therefore in that case they can be seen as direct competitors to that browser.[90] This means that whether search apps display weblinks to the user via an IAB or re-direct the user to a standalone browser which they have installed on their device may influence how much usage either can get. We explore whether Google's policy on how the Google Search App links to Chrome Custom Tabs may impact competition among mobile browsers on Android further below (see section 5).

3.21 We are still considering the implications of the above for our emerging thinking in relation to the extent of the constraint from search apps to mobile browsers and, separately, IABs as we continue with our assessment.

[90] Note of meeting with [✂].

# 4. The impact of Apple’s policies in relation to in-app browsing on iOS

4.1 This section considers whether Apple’s policies for in-app browsing may be limiting competition between browsers and browser engines for offering IABs on iOS. Apple’s three key policies and their potential impacts are summarised below:

(a) **Apple’s policy on remote tab IABs:** Apple does not allow third-party browser vendors to offer remote tab IABs, which means that all other browsers are prevented from offering in-app browsing on iOS. Only Apple’s Safari browser can be used as a remote tab IAB on iOS (ie via SFSafariViewController).[91] This may limit competition both in IABs and dedicated browsers.

(b) **Apple’s policy on webview and bundled engine IABs:** Apple does not allow third-party browser engines for webview and bundled engine IABs as part of Apple’s wider ban on alternative browser engines on iOS, referred to in this market investigation as the ‘WebKit restriction’ (see ‘WP2 – The requirement for browsers operating on iOS devices to use Apple’s WebKit browser engine’). Third parties cannot offer webviews based on browser engines other than WebKit for app developers to build upon for in-app browsing, so they are prevented from offering this product to app developers. Additionally, app developers cannot implement bundled engine IABs – they cannot build an IAB from scratch using their own choice of browser engine and must instead use SFSafariViewController or WKWebView.

(c) **Apple’s policies that affect user choice and control in relation to IAB**: Users do not appear to be able to exercise choice in relation to which browser is used for IAB and there is a lack of awareness on the part of users about IAB in general – meaning that users currently provide a weak constraint on providers of IABs.

4.2 In paragraph 41 of the Issues Statement for this market investigation, we outlined a potential concern that Apple may restrict the customisability and functionality of IABs through changes that restrict the use of certain implementations (ie webview implementations).[92] The evidence available to date suggests this is a relatively minor concern for app developers. Therefore, we are not currently planning to explore this topic (ie that Apple might restrict the customisability and functionality of IABs) further as part of this market investigation.

[91] The WebKit restriction also applies here. If third-party browsers were allowed for remote tab IAB, the WebKit restriction would prevent these browsers from using their own browser engines. This restriction is addressed in full in ‘WP2 – The requirement for browsers operating on iOS devices to use Apple’s WebKit browser engine’.

[92] Issues statement (publishing.service.gov.uk), paragraph 41.

## Apple’s policy on remote tab IABs

4.3 This section summarises the evidence considered to date in relation to Apple’s policy on remote tab IABs and our emerging thinking on how this policy may be limiting competition between browser vendors that might offer remote tab IABs.

### Apple’s submissions

4.4 Apple submitted that technical limitations and the benefits of existing IAB options on iOS were the key reasons for not allowing third-party browsers to offer remote tab IABs.

4.5 Apple submitted that it does not allow third-party browsers to link to native apps for remote tab IAB on iOS because of technical limitations relating to the set-up of the operating system. [✂]. A third party could develop and offer an SDK that allowed developers to embed within their apps an in-app browsing interface using WebKit. In the EU, a developer could ship an in-app browsing SDK based on an alternative browser engine. This approach specifically allows for browser vendors to compete to offer in-app browsing experiences for non-browser apps using alternative browser engines, by providing an SDK.[93]

4.6 Additionally, Apple submitted that it provides developers with multiple options to offer users access to web content on iOS. At one end of the scale, developers can choose to switch out of the app and use the default browser to view selected web content. At the other end of the scale, developers who have the capability and desire to do so can use WKWebView to create their own highly customisable in-app webview experience. As a middle option, Apple offers SFSafariViewController to developers as an option to implement a webview experience when they do not need or wish to customize or interact with the web content. When an app uses SFSafariViewController, interactions with the web content occur solely within the view controller, which ensures that the security and privacy standards to which users are accustomed are maintained. SFSafariViewController also provides users the option to link out to their default browser to view selected web content.[94]

### Evidence from third parties

4.7 Browser vendors have complained about Apple’s restriction on remote tab IABs.[95] While browser vendors we heard from generally do not seem to directly monetise remote tab IABs, they told us that they can derive some benefit from offering this functionality and many of them would therefore like to offer it on iOS.[96] Indeed, an

[93] Apple’s response to CMA’s information request [✂].
[94] Apple’s response to CMA’s information requests [✂].
[95] Responses to CMA information requests [✂].
[96] Responses to CMA information requests [✂].

increasing number of browser vendors currently offer this product on Android (see paragraph 2.35 for a list of browser vendors that offer remote tab IABs).

4.8 Several browser vendors submitted evidence on the benefits of offering remote tab IABs. In summary:

(a) Browser vendors submitted that offering remote tab IABs enables them to support their users more holistically, displaying web content within apps as well as in the dedicated browser.[97] One browser vendor submitted it can offer a more 'consistent and differentiated experience'.[98]

(b) Offering remote tabs makes the browser offering more 'sticky'. One browser vendor told us that for a browser without a remote tab IAB, there is a risk that its users may switch to an alternative browser with a remote tab IAB because the latter browser will keep appearing on the user's device (eg on iOS the user may switch from using their chosen default to Safari because SFSafariViewController keeps appearing when they tap on weblinks).[99]

(c) Users can benefit from the features and their settings (eg security, privacy, and login details) in the browser they choose for dedicated browsing while in-app browsing. Microsoft submitted that this enables a user's chosen browser to 'meaningfully [be] the user's browser everywhere'.[100]

(d) Browser vendors emphasised that their users' privacy can be better protected when using a remote tab IAB. Mozilla submitted the native app has less control over tracking and modifying web activity for remote tab IABs relative to where they use webview IABs.[101] Browser vendors which market themselves as privacy-focused submitted they could protect users' privacy with features such as tracker blockers in remote tab IABs.[102]

(e) Browser vendors told us additional traffic that goes to the remote tab IAB can benefit their wider browser offering via web compatibility effects and an increased volume of user feedback on issues encountered in the IAB.[103] As explained above (see paragraph 2.39), the evidence on the extent to which it

[97] To illustrate this benefit, consider the following: as a result of Apple's policy an iOS, user who chooses a third-party browser as their default (eg Chrome or Brave etc) would face a disjointed browsing experience on their mobile device, since Safari would keep appearing from weblinks within apps. A user choosing Safari as their default does not face this issue, as their browser remains consistent no matter how they are accessing the web.

[98] [✂] response to the CMA's information request [✂].

[99] [✂] of meeting with [✂].

[100] Note of meeting with Microsoft [✂].

[101] Mozilla's response to the CMA's [✂].

[102] Responses to CMA information requests [✂].

[103] Responses to CMA information requests [✂]. Web compatibility effects that result from additional traffic to a browser refers to the following phenomenon: additional traffic to a browser via the remote tab IAB increases the browser's market share. The browser's market share signals to web developers the importance of developing features that will work with that browser. This affects the quality of the user's experience within the browser. See 'WP1 – Nature of competition in the supply of mobile browsers and browser engines', paragraphs 2.45 to 2.54, for more detail.

may be possible for browser vendors to monetise IAB traffic suggests that this is not something that can be done currently.

4.9 Parties' submissions indicate that Apple's policy on remote tab IABs impacts app developers because they have less choice and customisability over remote tab IABs on iOS. App developers submitted that SFSafariViewController offers less customisation than the remote tab option on Android (ie Custom Tabs).[104]

4.10 Pinterest submitted that users may be confused on iOS because SFSafariViewController does not allow it to add certain features that are included in their webview and Chrome Custom Tabs IABs ie their red 'pin' button. Pinterest would prefer users to have a more familiar and consistent experience across all IABs in its app. It submitted that compared to iOS, this is 'simplified' on Android by Custom Tabs for the app developer, users and business advertising customers.[105]

4.11 Several parties submitted that this policy limits choice in relation to in-app browsing on iOS. Browser vendors submitted that SFSafariViewController does not allow users to use their default browser for in-app browsing on iOS.[106] OWA has published that SFSafariViewController 'always invokes and renders web pages in Safari, subverting browser choice entirely'.[107]

4.12 Evidence from [✂] suggests this policy might favour Apple's products in adjacent markets. [✂] submitted that SFSafariViewController disadvantages [✂] compared to Apple Pay. Users who choose to pay with [✂] are taken out of SFSafariViewController (and WKWebView) to pay via the [✂], which creates a disjointed user experience that does not arise for Apple Pay.[108]

4.13 Finally, one browser vendor submitted that this policy is 'bad for competition' and there is no inherent security challenge with building a system for remote tab IAB on iOS similar to Custom Tabs on Android.[109]

### Summary of emerging thinking on Apple's policy on remote tab IABs on iOS

4.14 Apple prevents all rival browser vendors from offering remote tab IABs on iOS**.** This means native apps cannot call on a browser other than SFSafariViewController for a remote tab implementation of in-app browsing. This is likely to be limiting rival browsers' ability to compete against Safari on iOS because they lack the functionality of displaying web content within an app. We

[104] Responses to CMA information requests [✂].
[105] Note of call with [✂].
[106] Responses to CMA information requests [✂].
[107] OWA - DMA Interventions - In-App Browsers (To Publish) (open-web-advocacy.org) accessed by the CMA 4 June 2024.
[108] [✂] response to CMA's information request [✂].
[109] Note of meeting with [✂].

understand browser 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. [110]

## Apple’s policy on webview and bundled engine IABs

4.15 This section assesses Apple’s policy on webview and bundled engine IABs. It sets out Apple’s key submissions in relation to this policy and evidence from third parties on its impact. We then summarise our emerging thinking on how this policy may be impacting competition between browser engine providers that might offer IABs.

### Apple’s submissions

4.16 Apple submitted that security risk is the main reason for not allowing browser engines other than WebKit for webview or bundled engine IABs – in particular:

(a) The WebKit restriction is necessary for reasons of security, privacy, and performance; and

(b) The WebKit restriction is part of how Apple ensures high levels of security, privacy and performance on iOS devices and this drives competition between ecosystems ie iOS devices competing with Android devices.[111]

4.17 Apple told us it does not allow third-party browser engines on its platform at all and that its reasoning for this is the same for IAB as for dedicated browsers. The exception to this is the recent change in the EU, which has been mandated by the requirements of the Digital Markets Act (DMA).[112]

4.18 In the EU, Apple offers the Embedded Browser Engine Entitlement which Apple has stated: ‘allows browser vendors to develop an in-app browsing SDK based on an alternative browser engine that can be used to provide an in-app browsing experience for non-browser apps. This approach specifically allows for browser vendors to compete to offer in-app browsing experiences for non-browser apps using alternative browser engines, by providing an SDK. As Apple has explained, however, the use of alternative browser engines for in-app browsing, including via in-app browsing SDKs, poses substantial risks given the significantly larger volume of non-browser apps and the fact that developers of non-browser apps do not generally focus on or have an expertise in addressing the complex security

[110] [✂] response to CMA’s information request [✂].

[111] See paragraph 5.2 of ‘WP2 – The requirement for browsers operating on iOS devices to use Apple’s WebKit browser engine’.

[112] Regulation (EU) 2022/1925 of the European Parliament and of the Council of 14 September 2022 on contestable and fair markets in the digital sector and amending Directives (EU) 2019/1937 and (EU) 2020/1828 (Digital Markets Act).

risks and volume of security issues associated with operating a browser or browser engine.’[113]

4.19 Apple told us that allowing alternative browser engines for in-app browsing as it has done in the EU creates significant risk to security. To comply with the DMA, Apple is providing certain functionalities on iOS for browser vendors wishing to base their browsers on an alternative browser engine in the EU. Apple will be reserving certain features to browser developers that are not important for app developers with bundled engine IABs and would substantially exacerbate security and privacy risks if afforded to non-browser apps. Apple told us non-browser app developers are generally not practiced in maintaining browser engine security and do not have the same ability as browser vendors to tackle security or privacy risks in the IAB – eg app developers may not have a security vulnerability disclosure process.[114]

4.20 Apple submitted that allowing alternative browser engines for bundled engine IABs poses a significant security risk. This is because non-browser app developers are less likely than browser app developers to have the necessary experience and capabilities for maintaining and updating browser security. Browsers face many more vulnerabilities than the most popular non-browser apps. As a result, browser developers have a higher level of sophistication in identifying, assessing and responding to security threats than non-browser developers.[115]

4.21 Apple submitted that the security risk is further amplified by the vastly larger number of apps providing in-app browsing relative to dedicated browsers. [Over 50] browser apps are available in the UK App Store, compared to [over 20,000] apps that use SFSafariViewController. This vast difference in volume makes the attack surface from the in-app browsing use case many orders of magnitude greater than that of the dedicated browser use case.[116]

4.22 Additionally, Apple told us that users may not be aware of or understand the security and privacy protections and policies within an IAB that uses an alternative browser engine or how these might differ from those applied by dedicated browsers.[117]

## Evidence from third parties

4.23 Evidence from third parties suggests that browser engine providers would benefit from the additional traffic to their browser engine that would come through webview IAB implementations due to effects on web compatibility (ie web

[113] Apple’s response to the CMA’s information request [✂].
[114] Note of meeting with Apple [✂].
[115] Apple’s response to the CMA’s information request [✂].
[116] Apple’s response to the CMA’s information request [✂].
[117] Note of meeting with Apple [✂].

developers are more likely to develop certain features to be compatible with their browser engine). Browser engine providers could also benefit from increased feedback and enhanced incentives to invest in performance and security improvements for their engine. See paragraph 2.43 for further detail on the benefits of additional traffic to a browser engine.

4.24 However, the evidence we have seen also suggests there is only limited interest among browser engine providers and browser vendors to offer alternative webviews (ie a webview IAB based on a browser engine different from WebKit) for app developers to build upon for in-app browsing on iOS. We have heard from two browser vendors and browser engine providers that have considered offering a webview for IAB:

(a) Mozilla told us that in the past it hoped to be able to offer its engine to other third-party browsers using GeckoView.[118] Mozilla submitted that non-browser apps loading third-party web content should call on a remote tab IAB that uses the user's default browser instead of a webview IAB. Moreover, Mozilla submitted that the App Store should prevent the use of webview IABs to display third-party web content rather than attempting to enable alternative webview engines such as GeckoView. Mozilla also told us that it does not believe that app developers should be forced to use an alternative to the system-provided browser engine (for example, the default browser) when rendering first-party web content owned by an app (eg where an app renders a Help or Settings screen). This would be an 'extremely complex' process and might create friction in the user experience.[119]

(b) One browser vendor submitted that it has considered offering an alternative webview for app developers to adopt for in-app browsing, but ultimately decided offering a webview product would not be rewarding for its business growth. The browser vendor told us that this was because webviews do not provide a great user experience relative to dedicated browsers. The browser vendor also cited Apple's App Store rules being 'too restrictive' (ie Apple's ban on alternative browser engines) as one of the reasons for not attempting to launch this product.[120]

4.25 The above evidence is further supported by past entry on Android, where Google allows alternative webviews to be offered for in-app browsing. We understand that past entry for webviews on Android has been very limited and we are not aware of third-party providers other than Mozilla (ie with GeckoView).

4.26 Most app developers who engaged with this market investigation have not expressed interest in using alternative webviews on iOS or Android. It may be that

[118] Note of meeting with Mozilla [✂].
[119] Mozilla's response to the CMA's information request [✂].
[120] Note of meeting with [✂].

demand for alternative webviews such as GeckoView is low. However, we understand that app developers stand to benefit from the option to use alternative browser engines for bundled engine IABs if this were possible on iOS and indeed (as described above), we are aware of some app developers (eg Meta) offering a bundled engine on Android.

4.27 Parties told us that Apple’s policy on bundled engine IABs limits app developers’ ability to customise and introduce certain features within their IAB.

(a) Google submitted that bundled engine IABs have benefits for competition and can help app developers tailor their IAB for specific use cases.[121]

(b) One app developer submitted the potential benefits of custom browser engine IABs for users on Android - including improved security, user experience, functionality, stability and performance - are not possible on iOS due to Apple’s WebKit restriction, which ‘shows how artificial Apple’s restrictions have been.’[122]

4.28 Some app developers told us bundled engine IABs could present benefits to their businesses. For example:

(a) A large app developer submitted that it would introduce a bundled engine IAB on iOS if this were possible. It submitted that it had planned to introduce a custom browser engine IAB for iOS in the EU, but the only reason it has not yet done so is that Apple’s requirements are too prohibitive.[123]

(b) An app developer told us that implementing and maintaining a bundled engine IAB, could have benefits for its business, customising the user interface, improving stability and tracking user activity. [✂][124]

4.29 However, we understand that the majority of app developers would not be interested in developing a bundled engine IAB on iOS. This includes large app developers that are relatively engaged in developing their IAB. These app developers are largely satisfied with the functionality of WKWebView.[125,126]

4.30 We have received the following evidence from third parties on the security risk of allowing bundled engine IABs on iOS:

[121] Note of meeting with Google [✂].
[122] Note of meeting with [✂].
[123] Note of meeting with [✂].
[124] Note of meeting with [✂].
[125] For example, [✂].
[126] Responses to the CMA’s information requests [✂].

(a) Google submitted that building and maintaining a bundled engine IAB is a big investment, with security a key consideration.[127]

(b) [✂] submitted that its custom browser engine IAB on Android has ‘improved security compared to the Android System WebView’. [✂] submitted that it can control the lifecycle of tabs better with its own browser engine on Android as compared to WebKit on iOS. Additionally, [✂] can ensure that users are receiving important security updates more promptly when its embedded engine is used.[128]

(c) OWA published that bundled engine IABs present significant privacy and security concerns. It published that bundled engine IABs typically have many unique bugs and issues and that their security may be poorer quality given they are maintained and tested by the native app developer rather than ‘a dedicated browsing team’.[129]

### Summary of emerging thinking on Apple’s policy on webview and bundled engine IABs

4.31 Apple prevents rival browser engines from offering webview IABs to app developers on iOS. This policy directly prevents browser engine providers from competing against WKWebView with their own webview on iOS. Based on the evidence we have seen to date, the impact of this policy on the ability of browser engine providers and IABs to compete is unclear.

4.32 From a supply side perspective, there is only limited interest among browser engine providers and browser vendors to offer alternative webviews, and they are generally more concerned by Apple’s policy on remote tab IABs. The same policy also limits the options available for app developers implementing IABs on iOS, as they cannot choose an alternative engine to WebKit for IABs, either adopting one from third parties or implementing their own bundled engine IABs. This may impact their ability to innovate and improve their apps. However, while some app developers told us bundled engine IABs could present benefits to their businesses, levels of interest amongst app developers in this option appears to be low. Further, we acknowledge that allowing bundled engine IABs may introduce security risks such that only app developers with significant resources could offer these in a secure manner.[130]

[127] Note of meeting with Google [✂].

[128] Note of meeting with Meta [✂].

[129] OWA - DMA Interventions - In-App Browsers (To Publish) (open-web-advocacy.org) accessed by the CMA 5 June 2024, pages 16 and 37.

[130] See paragraphs 4.20, 4.21 and 4.30 above.

## Apple's policies that affect user choice and control in relation to IAB

4.33 This section considers the extent to which users of apps are able to make effective choices about IAB implementations on iOS devices. It addresses the extent of user awareness of in-app browsing generally and whether this is likely to exacerbate the effect of Apple's other IAB policies on competition between browsers.

### IAB interface and settings on iOS

4.34 As set out above, there are two types of IAB implementations on iOS (see paragraph 2.53a).

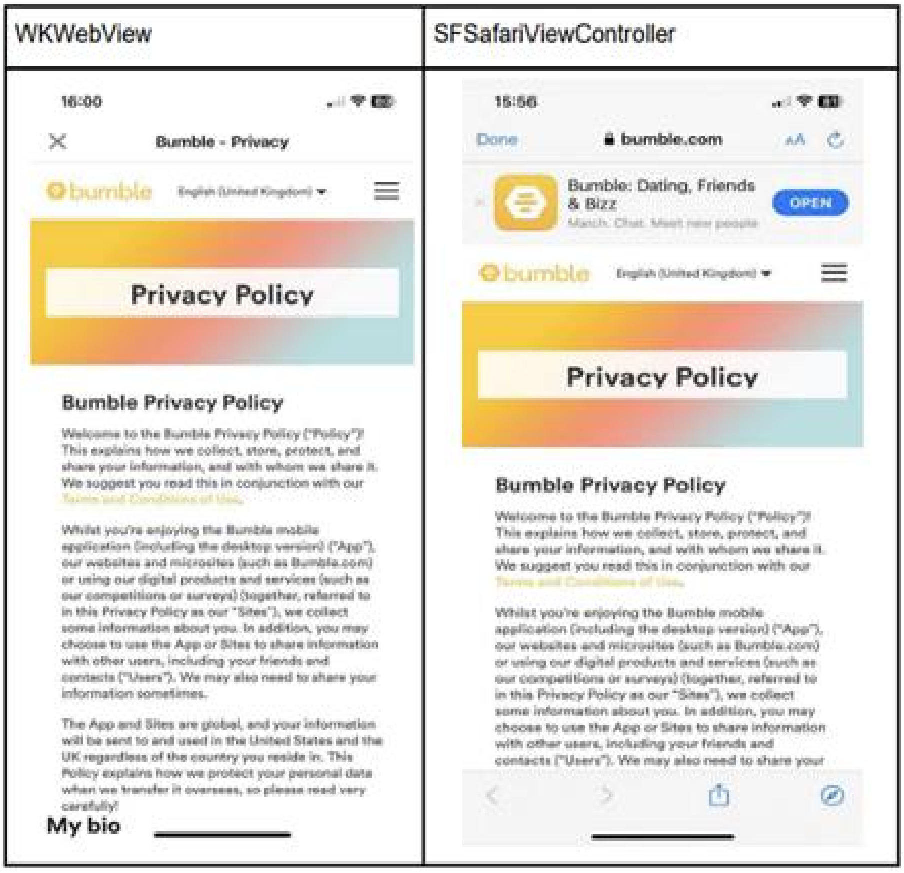
(a) The only remote tab IAB implementation ('SFSafariViewController') on iOS is Safari. Apple also provides WKWebView which is based on the Webkit browser engine. This means that where a user's default browser is not Safari, any default that a user has set for dedicated browsing is not applied for in-app browsing on iOS.

(b) If a user is not aware that they are in IAB based on WKWebView or SFSafariViewController implementation, and they offer different security and privacy settings to their default browser, this may mean that a user is being unknowingly and unwillingly tracked when browsing the web within an application. The lack of user awareness of IAB may also make users less likely to control their IAB settings (eg by leaving the IAB for an external dedicated browser) so that users are likely to be a very limited competitive constraint on providers of IABs on iOS.

4.35 Apple's control over the design of IAB implementations impacts how the IAB visual interface is designed and configured. As a result, the two IAB interfaces on iOS are similar and mimic the Safari browsing experience, meaning that consumers might not be aware which product they are using (see Figure 4.1).[131] This, in turn, may contribute to low levels of user awareness and engagement with IAB technology.

[131] Aesthetic manipulation (ie 'type of choice architecture') may be being used to create an impression of familiarity which hides the fact that users may not be using their dedicated browser. See Evidence Review of Online Choice Architecture (paragraph 3.6; table 1).

**Figure 4.1: WKWebView and SFSafariViewController in-app browsers’ visual interfaces on iOS.**



Bumble Privacy Policy

Welcome to the Bumble Privacy Policy ("Policy")! This explains how we collect, store, protect, and share your information, and with whom we share it. We suggest you read this in conjunction with our Terms and Conditions of Use.

Whilst you're enjoying the Bumble mobile application (including the desktop version) ("App"), our websites and microsites (such as Bumble.com) or using our digital products and services (such as our competitions or surveys) (together, referred to in this Privacy Policy as our "Sites"), we collect some information about you. In addition, you may choose to use the App or Sites to share information with other users, including your friends and contacts ("Users"). We may also need to share your information sometimes.

The App and Sites are global, and your information will be sent to and used in the United States and the UK regardless of the country you reside in. This Policy explains how we protect your personal data when we transfer it overseas, so please read very carefully!

Bumble Privacy Policy

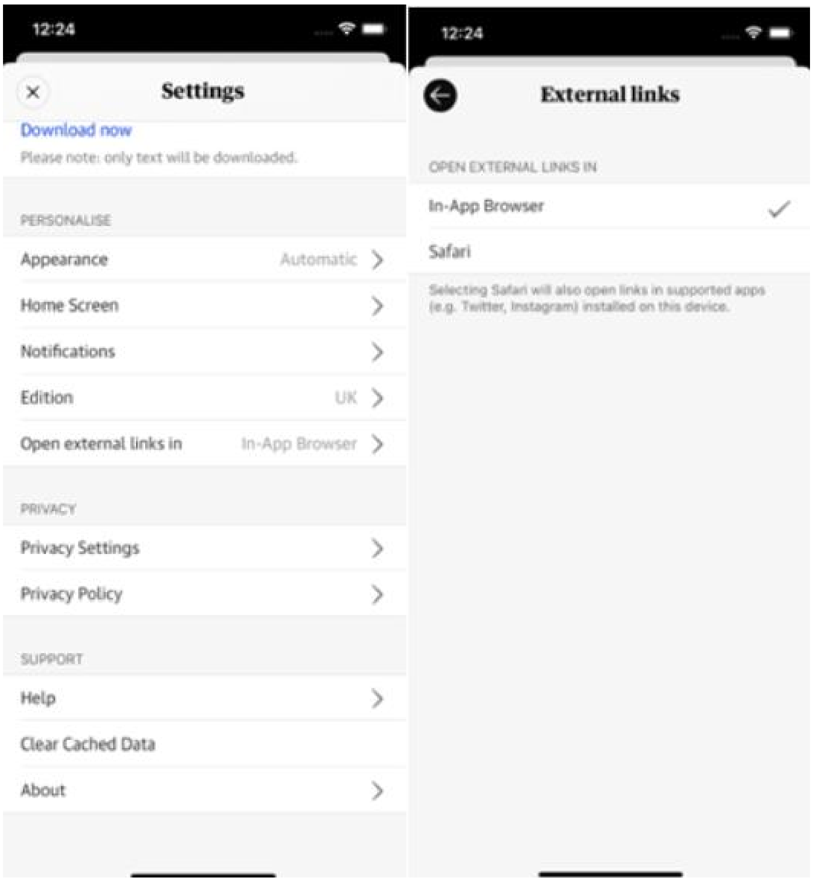
Welcome to the Bumble Privacy Policy ("Policy")! This explains how we collect, store, protect, and share your information, and with whom we share it. We suggest you read this in conjunction with our Terms and Conditions of Use.

Whilst you're enjoying the Bumble mobile application (including the desktop version) ("App"), our websites and microsites (such as Bumble.com) or using our digital products and services (such as our competitions or surveys) (together, referred to in this Privacy Policy as our "Sites"), we collect some information about you. In addition, you may choose to use the App or Sites to share information with other users, including your friends and contacts ("Users"). We may also need to share your

4.36 Finally, the option for users on iOS to disable or enable in-app browsing in their device or app settings is usually not supported across apps. However, we found evidence that users can disable IAB on the Guardian app (see Figure 4.2).[132]

[132] GMG response to the CMA’s information request [✄].

**Figure 4.2: It is possible to disable in-app browsing on the Guardian app on iOS.**



12:24

× Settings

Download now

Please note: only text will be downloaded.

PERSONALISE

- Appearance — Automatic >
- Home Screen >
- Notifications >
- Edition — UK >
- Open external links in — In-App Browser >

PRIVACY

- Privacy Settings >
- Privacy Policy >

SUPPORT

- Help >
- Clear Cached Data
- About >

12:24

External links

OPEN EXTERNAL LINKS IN

- In-App Browser ✓
- Safari

Selecting Safari will also open links in supported apps (e.g. Twitter, Instagram) installed on this device.

## Apple's submissions

4.37 Apple submitted that the options provided for in-app browsing on iOS are designed to 'enable developers to offer that in-app browsing experience in the way the developer considers most effective' and if users were to choose the browser to be used for in-app browsing in all circumstances, this would mean that 'app developers could not be certain what features are offered and what limitations are imposed by an IAB that they did not choose'.[133] For example, where developers use in-app browsing for sign-in flows, the user experience may be disrupted if the developer cannot control the IAB. The IAB may break the flow or cause additional friction in the sign-in process if the developer does not retain control over the in-app browsing functionality.[134]

4.38 Apple also submitted that users control whether to continue using SFSafariViewController or to view the web content in their default browser. Apple mandates that users be given the option 'by way of an easy to spot UI element' to view the web content in their default browser.[135] See also paragraph 2.55 and Figure 2.2 outlining how app developers can design the 'IAB choice menu icon'

[133] Apple's response to the CMA's information request [✂].
[134] Ibid.
[135] Ibid.

(eg in the form of a vertical ellipsis or share icon). There is no uniform approach to the design of this UI element.

## Evidence from third parties

4.39 We asked a range of third parties to express their views on the extent of user control, choice and awareness within in-app browsing on iOS.

4.40 OWA submitted that IABs as a technology do not acknowledge user choice of default browser (ie a type of a dark pattern)[136], which, in turn, 'harms legitimate browsers of all sorts, but particularly small players'. OWA suggests that 'the only sort of IAB that does not present outsized user and competition risks are remote-tab IABs that steadfastly respect user's default browser choice'.[137] OWA submitted that app developers are not only misleading consumers but can also grant the app the ability to manipulate the third-party websites, leading to privacy and security breaches.[138] Moreover, OWA submitted that the quality of the user experience could be reduced because accessibility settings that users have in their default browser may not be carried over to the IAB.[139]

4.41 [✂] submitted that allowing users to set their default in-app browser would make it 'more difficult to offer deeply embedded browsing experiences'. Using [✂] custom IAB improves user experience through the ability to deliver more integrated content. For example, operating its own engine means that [✂] can offer new features such as an improved WebShare API experience - meaning an integrated sharing API which would have [✂] logos (for example, this would make it easier for users to share a link within the [✂] app).[140]

4.42 TikTok submitted that the decision to use WKWebView on iOS comes from its willingness to create a 'more seamless experience for customers by ensuring that they can access web content from within the TikTok app'. Other reasons included ensuring the safety of both users and the content they access.[141]

4.43 An app developer told us that using both WKWebView and SafariViewController on iOS enables a 'cohesive' experience within the app [✂]. However, the same app developer also stated that these IABs may have disadvantages compared to the default browser where a user seeks to make a purchase because the default browser has their cookies, cache and log-in data, payment details and user history stored. [✂].[142]

[136] Note of a meeting with OWA [✂]. Dark patterns refer to the set of (deliberately) manipulative practices that harm users online.
[137] OWA's response to CMA's information request [✂].
[138] OWA, In-App Browsers: The worst erosion of user choice you haven't heard of, accessed by the CMA 2 July 2024.
[139] Note of meeting with OWA [✂].
[140] Note of meeting with [✂].
[141] Call note from meeting with TikTok [✂].
[142] Note of meeting with [✂].

4.44 Finally, we received research from a third-party app provider on user preferences and satisfaction relating to IAB.[143] The research showed that most users preferred to stay within the app when they were browsing content in which they had a surface level interest. Respondents expressed that staying within the app allowed them to get back to their task more easily and they didn't have to have an unnecessary number of browser windows open. However, users who were interested in purchasing an item preferred to be taken to the native browser outside the app. Respondents stated that the native browser felt 'safer', and purchasing was more streamlined because they could access their saved payment details from within the native browser application. The research also found that over half of respondents (51%) would prefer to be informed and given the choice of the type of browser they would be taken to. Reasons given for this preference included 'feels transparent' and 'feel in control of experience'. A further 13% voiced that although they do not need to choose the type of browser they would be taken to, they would like to be informed. Reasons given for this preference included 'knowing ahead of time where the link is going can help [you to] assess if you truly want to visit' and 'feel in control of experience'.[144]

### Evidence from consumer research

4.45 As there is limited available evidence from parties or publicly available literature on users experience and behaviour with IAB, we commissioned Verian to conduct primary research with smartphone users. The research comprised two phases: a qualitative phase to explore users' awareness, understanding and behaviour in relation to mobile browsers and in-app browsers[145]; and a quantitative phase to assess the degree of user's awareness, understanding and behaviour related to browsers and in-app browsers, with a specific focus on choice architecture elements.[146]

4.46 The qualitative phase comprised 40 depth interviews and observations of participants undertaking a number of set tasks on their mobile, including going to a social media application and scrolling through that app until they found an external link or advert, clicking on the link or advert and reading through the content. The research was designed to include a range of participants with regard to operating system, age and self-assessed technical confidence. The quantitative phase comprised a consumer survey with a representative sample of UK smartphone users drawn from an online panel that used a random probability-based approach to recruitment. (see 'WP5 – The role of choice architecture in the supply of mobile

[143] Submission from [✂]
[144] [✂] response to the CMA's information request [✂].
[145] Verian Group UK (2023), Mobile Browsers Qualitative Consumer Research.
[146] Verian Group UK (2024), Mobile Browsers Quantitative Consumer Research.

browsers' for more information regarding the qualitative research sample and the quantitative survey sample recruited by Verian).

4.47 The findings from the Verian qualitative research and the Verian survey apply across both iOS and Android devices.

4.48 Overall, this research suggests that there is a lack of user awareness in relation to the following:

(a) that in-app browsing is a different experience to browsing the web on a dedicated browser app;

(b) the subtle differences between the in-app browsing visual interface (remote tabs vs webview) and their designated browser; and

(c) that an IAB may not use the same browser as the default browser being used on a dedicated browser app.

4.49 Verian's qualitative research with users showed that overall users have very low levels of awareness of in-app browsing.[147] Respondents had not thought about in-app browsing before or whether they were using a browser. Respondents also did not normally think about in-app browsing or what was happening operationally 'behind the scenes' when they viewed web content within an app.

4.50 Users' understanding of how IAB works is poor. When probed to think about IAB in the Verian qualitative research, users thought that they were 'just using the app', visiting 'the app's version of the website', using their 'main' browser or using a partial version of a browser or an extension. Some users considered that there were minor differences between browser apps and IABs. For example, users reported that IABs were slower, had more ads (more tracked), had no tabs or browser history, seemed clunkier and had no address bar.

4.51 During the observed tasks to assess users' technical ability, conducted as part of Verian's qualitative research, users were asked to go onto a social media app they often use on their device, to scroll through the app until they saw an external link or advert, follow the link to the content, and finally read the content.￼ ￼ Users did not always find it easy to complete this task and there was general reticence about clicking on links within social media apps, to avoid targeted advertising, potential scams and interruptions or in case they lost where they were on the app. This

[147] In particular, the quantitative research comprised a sample of UK smartphone users drawn from an online panel that used a random probability-based approach to recruitment. Recruitment for the qualitative consumer interviews included consumers with a range of perceived technical literacy. including those with lower levels of digital literacy, given the importance of understanding how potentially more vulnerable users navigate browsers on their phones. Verian Group UK (2023), Mobile Browsers Qualitative Consumer Research, slides 30 to 32.

revealed users' distrust of third-party app links and to the preference to maintain a more seamless app experience.[148]

4.52 When asked to identify which of a set of statements relating to in-app browsing were true or false, almost half of respondents thought it was true that different apps on their smartphone may use different web browsers, 13% of respondents incorrectly believed that different apps did not use different web browsers, while the remaining 40% did not know.[149]

4.53 The Verian survey also found that 52% of iOS users incorrectly believed that when clicking on a link within an app it will always open in their default web browser. This percentage was significantly smaller for Android users – 43% of Android users provided an incorrect answer. Only 18% of iOS users correctly identified that this is not the case with the remaining 30% responding that they did not know.[150]

### Summary of emerging thinking on Apple's policies that affect user control and choice relating to in-app browsing

4.54 Overall, users have limited choice and control in relation to which browser is used for IAB implementations in native apps that they use and IAB in general because:

(a) Where a user's default browser is not Safari, any default that a user has set for dedicated browsing is not applied for in-app browsing on iOS.

(b) Apple's control over the design of IAB implementations, impacts how the visual interface is designed and configured on iOS (eg mimics 'actual' Safari browsing experience). This may result in users finding it difficult to understand when they are in their default browser and when they are in an in-app browser.

(c) Linked to this, consumer research indicates that there is low awareness of IAB, in that users will not often be aware that they are in an IAB, or that their default browser, which they may have actively chosen because it offers certain features, is not being used for that purpose. As a result, users might

[148] In addition, when users were asked to think about data collection linked to IAB, users were more sensitive to the 'first click' on weblinks (ie an initial reaction from users when presented with an unknown or unfamiliar weblink) – with 'browsing data' less front of mind. When probed to think about who might have access to the data, users would mention: (i) the social media company (as it was delivering further advertising); (ii) the company of the website they visited; (iii) the browser company (eg Google or Apple); and (iv) their phone manufacturer (eg Samsung).

[149] Verian Group UK (2024), Mobile Browsers Quantitative Consumer Research. In addition, understanding that different smartphone apps use different web browsers was correlated with age (younger respondents were more likely to know this), experience (those who spent more time on their phone were more likely to respond correctly) and perceived confidence in using smartphones and apps (those with higher self-reported confidence were more likely to respond correctly).

[150] The understanding that weblinks do not always open in the default browser was higher for younger users (16-24 years old, 55%) and the most technically literate users (59% of those with high confidence and 47% of those with medium confidence). See Verian Group UK (2024), Mobile Browsers Quantitative Consumer Research, slide 59, for the overview across both iOS and Android users.

not realise which browser features are not carried over (eg privacy and security features, or ‘saved’ password information).

(d) Finally, users are often not offered the same levels of control over their in-app browsing experience. This can vary across apps (eg ‘IAB choice menu’) and also across device level settings (eg users mostly cannot turn off IAB on iOS devices)**.**

4.55 Overall, this means that users’ control and choice is likely to be a very limited constraint on the extent of competition between providers of IABs and browsers.

## 5. The impact of Google's policies in relation to in-app browsing on Android

5.1 This section considers whether Google's policies for in-app browsing may be limiting competition between browsers and browser engines for offering IABs on iOS. Google's three key policies and their potential impacts are described below:

(a) **Google's policy on remote tab IABs:** We are considering whether Google has prevented rivals from offering competing remote tab IABs on Android (or disadvantaged them). We have not received strong evidence to date suggesting that Google's policy on remote tab IABs has an adverse impact on competition.

(b) **Google's policy on webview IABs:** In the Issues Statement, the CMA suggested that default settings and preinstallation on Android make it difficult for app developers to use IABs based on alternative webviews.[151] This has been supported by evidence we have seen – alternatives to Android WebView are likely to be unattractive to app developers because third-party webviews increase an app's size and are more burdensome to maintain.

(c) **Google's policies that affect user control and choice relating to IAB:** users cannot actively select their in-app browsing implementation on Android and there is lack of awareness about IAB more broadly. This means that users currently provide a weak constraint on providers of IABs.

### Google's policy on remote tab IABs

5.2 This section explains our emerging thinking on whether Google restricts competition between browser vendors for providing remote tab IABs. The evidence we have seen to date suggests that the level of customisation options available in the Custom Tabs system is likely to be the same for all browser vendors, but browser vendors can choose whether to support these features. We understand that Google is investing in the Chrome Custom Tabs product, and this seems to be an important part of Google's strategy in relation to in-app browsing.[152] We are aware that Chrome Custom Tabs offers certain features that others do not. For example, Chrome Custom Tabs provides insights to app developers on user activity in the IAB.[153]

5.3 Submissions with third-party browser vendors indicate that they are typically less engaged than Google on Custom Tabs IABs.[154] Google has some visibility over

[151] CMA's Issues statement, page 9

[152] Note of meeting with Google [✂].

[153] Google's response to the CMA's information request [✂].

[154] Responses to CMA information requests [✂].

the use of Android WebView IABs as well as Chrome Custom Tabs.[155] Google may therefore be more incentivised to invest and develop Chrome Custom Tabs relative to other browser vendors. Indeed, Google told us that it considers that IAB on mobile is becoming ‘more critical and common’ as a way that people browse.[156]

5.4 We have received evidence on a separate issue relating to Custom Tabs – that is, whether Google favours its own version of Custom Tabs in the Google Search App in a manner that impacts on competition. The Google Search App allows users to search the web to generate ranked search results. Google also ships a ‘widget’ (a tool on Android users’ device home screens) with the same function.[157] We understand that when users click on a search result from the Google Search app or widget, the link takes them to browse the webpage in Chrome Custom Tabs (unless Chrome is disabled on the mobile device).[158]

5.5 Third parties have raised the following concerns in relation to this policy.

(a) One browser vendor submitted that Google ‘self-preferences’ its own browser by ‘hard-coding’ Chrome in the Android Google Search App.[159]

(b) OWA published that the Google Search app on Android being ‘locked to Google Chrome’ is a ‘clear example of undermining the user’s choice of default browser’ (ie where Google Chrome is not the user’s default).[160]

5.6 We note that in this case Google’s position as the OS provider and app developer for the Google Search App may impact how much usage rival providers of remote tabs IABs get on Android. This is because when Google chooses Chrome Custom Tabs for traffic generated through the Google Search App on Android, Google is effectively limiting the ability for rival browsers to potentially get traffic that goes through the Google Search App.

5.7 Indeed, Google submitted that the Google Search App was the app with the [✂] globally on Android devices in March 2024.[161] An internal document from Google also estimates that traffic from the [✂].[162] Further, we have found that Android device manufacturers are incentivised to pre-install the Google Search App on their devices because they can earn payments from Google by entering into agreements relating to the placement of this app.[163]

[155] Google’s response to the CMA’s information request [✂].
[156] Note of meeting with Google [✂].
[157] See The Google app – Download the app for Android and iPhone, accessed by the CMA 10 June 2024.
[158] Note of meeting with Google [✂].
[159] [✂] response to the CMA’s information request [✂].
[160] OWA - DMA Interventions - In-App Browsers (To Publish) (open-web-advocacy.org), page 41, accessed by the CMA 2 July 2024
[161] Google’s response to CMA’s information request [✂].
[162] Google’s internal document [✂].
[163] MEMS, Final report, paragraph 6.62 and footnote 496.

5.8 While we note that the fact that weblinks in the Google Search App sends users to Chrome Custom Tabs may be providing an advantage to Chrome Custom Tabs in terms of usage, we also recognise that there are benefits in allowing app developers such as Google to have some degree of choice over the IABs in their app.

### Summary of emerging thinking on Google's policies in relation to remote tab IABs

5.9 Google's policy on remote tabs does not appear to have a clear impact on competition between browsers because it is not preventing rivals from offering competing products. Further, while we note that Google linking its Search app with Chrome Custom Tabs may be resulting in advantages in terms of the latter's usage, we recognise there are benefits in allowing app developers such as Google to have some degree of choice over the IABs in their app. Therefore, based on the evidence we have seen so far, we do not consider this policy on remote tab IABs is likely to limit competition among mobile browsers on Android.

## Google's policy on webview IABs

5.10 This section considers Google's policy on webview IABs. It sets out Google's rationale for this policy and evidence from third parties on its impact. We then summarise our emerging thinking on how this policy may be impacting competition between browser engine providers that might offer IABs.

### Google's submissions

5.11 Google submitted that it offers browser engine choice for IABs on Android and that the Android platform is set up to be attractive and easy to use for developers.[164]

5.12 In the CMA's Mobile Ecosystems Market Study, we were concerned that 'default settings make it difficult to use a browser engine other than Blink' for IABs on Android. In response, Google submitted that this concern was 'misplaced' and that app developers can 'incorporate a different in-app browsing technology if they choose to', including GeckoView.[165]

5.13 Moreover, Google submitted that it provides 'easy access' to software development kits (SDKs), libraries, and application programming interfaces (APIs), such as Android WebView – access to these tools is 'a necessary feature of a successful app development platform'. In Google's view, 'it is doubtful that these

[164] Google's Response to the CMA's consultation, page 3-4.

[165] MEMS, Final report, paragraph 5.83 ; Google's Response to the CMA's Consultation, pages 3-4, accessed by the CMA 6 June 2024.

tools can properly be characterized as "defaults" – or at least, defaults of the type that can give rise to inertia bias.'[166]

## Evidence from third parties

5.14 Android WebView (based on the Blink browser engine) comes pre-installed on Android devices and is the default option for app developers looking to implement a webview IAB. Based on evidence we have seen so far, app developers looking to offer a webview IAB based on an alternative webview (such as GeckoView) can face certain frictions.

5.15 For example, using an alternative webview increases the size of the app, so the app takes up more storage on a user's device, which may reduce its attractiveness to users. Google told us that app developers using alternative webviews such as GeckoView on Android would need to incorporate the web browser engine's library into their app, but otherwise work similarly as for Android WebView in that they call on an API within their app.[167]

5.16 We understand that Mozilla is the main browser engine provider that has attempted to offer an alternative webview engine on Android (Mozilla offers GeckoView, which is based on the Gecko browser engine). Mozilla submitted that the preinstallation of Android WebView makes it difficult to offer an alternative webview engine for app developers to build upon for in-app browsing on Android.[168] Mozilla told us it had initially hoped that GeckoView would be an alternative to Android WebView. However, it subsequently chose not to prioritise the provision of GeckoView as a competing webview on Android because of the frictions described above (ie the additional storage required for apps to use alternative webviews and the lack of automatic updates) and that Android WebView was very difficult to compete with due to its default status and easy availability to developers on Android.[169]

5.17 While there is no blanket ban on alternative webviews on Android, the frictions outlined above may make it difficult for rival browser engine providers to compete effectively with Android WebView. Further, as explained in section 4, it seems interest among browser engine providers to offer alternative webviews is low. Indeed, Mozilla told us that it no longer prioritises competing with the system webview, given it submitted that remote tab IABs are better suited to support in-app browsing for third-party web content instead of webview IABs.[170]

[166] Google's Response to the CMA's Consultation, pages 3-4
[167] Google's response to the CMA's information request [✂].
[168] Note of meeting with Mozilla [✂].
[169] Note of call with Mozilla [✂].
[170] Mozilla's response to the CMA's information request [✂].

### Summary of emerging thinking on Google’s policy on webview IABs

5.18 The evidence suggests that, while allowed, it may be difficult for browser engines other than Blink to offer attractive alternatives to Android WebView to app developers on Android. However, the impact of this difficulty is unclear. On the one hand, we understand interest among browser engine providers (ie Mozilla) as well as app developers for offering and using alternative webviews to the system webview on Android is low. On the other, the limited attractiveness of offering such alternatives is partly influenced by the set-up of the operating system.

## Google’s policies that affect user control and choice relating to in-app browsing

5.19 This section addresses the extent to which users are able to make effective choices about IAB on Android; the extent of user awareness of IAB and whether this is likely to exacerbate the effects of Google’s other policies on IAB on competition between browsers in general.

### IAB interface and settings on Android

5.20 There are three types of IAB implementation on Android (see paragraph 2.53b).

(a) Users have limited choice over their in-app browsing implementation and cannot select their default browser for IAB. This is because webview or bundled engine implementations of in-app browsing do not rely on any dedicated browser apps installed on the device and will not therefore call on the user’s default browser.

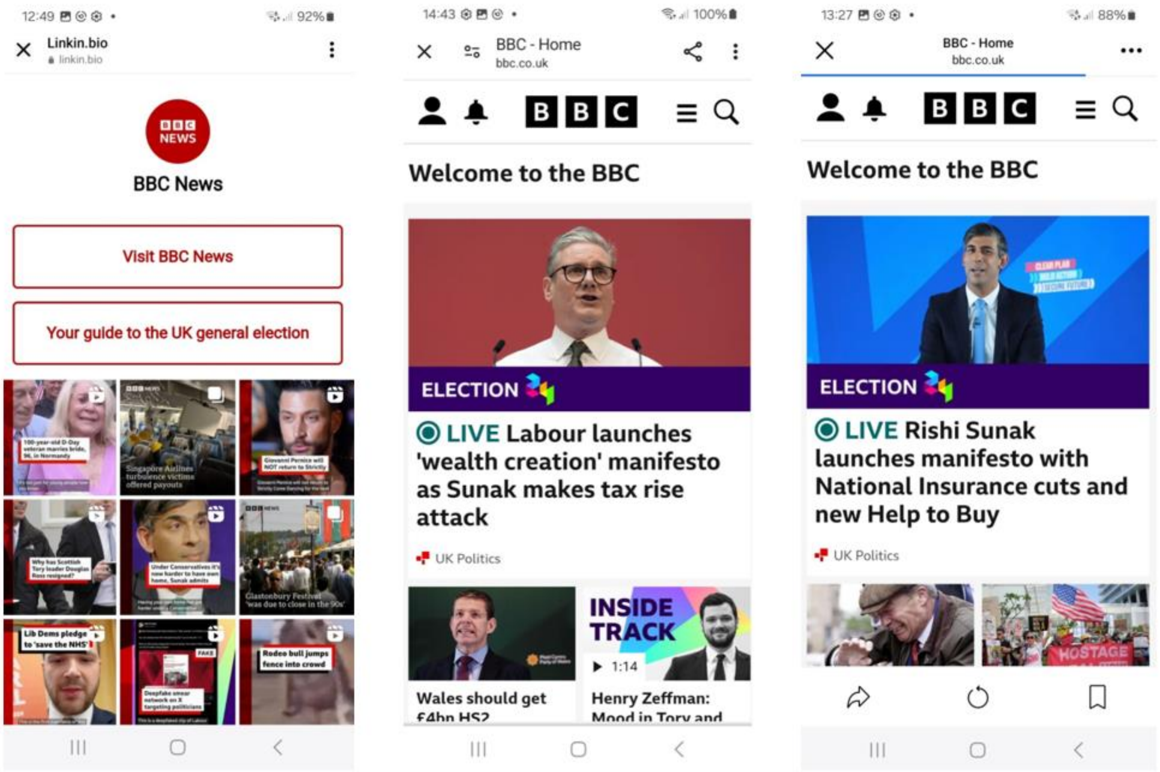
(b) However, the Custom Tabs IAB implementation when opening third-party content will rely on the user’s dedicated browser by default but that can be changed by app developers if they choose to select a specific browser.

5.21 The in-app browsing visual interfaces on Android, similarly to iOS, mimic the ‘actual’ Chrome browsing experience with some visual differences between the different IAB implementations (see Figure 5.1).[171] This, in turn, may contribute to low levels of user awareness of in-app browsing. If a user is not aware that they are in an IAB and the IAB offers different security and privacy settings to their default browser, this might mean that users are being tracked without awareness or consent, while navigating the web within an app. The lack of user awareness of IAB may also make users less likely to control their IAB settings (eg by leaving the

[171] We consider this as an example of aesthetic manipulation (ie ‘type of choice architecture’) See Evidence Review of Online Choice Architecture (paragraph 3.6; table 1).

IAB for an external dedicated browser) so that users are likely to be a very limited competitive constraint on providers of IABs on Android.

**Figure 5.1: (i) Instagram's webview IAB, (ii) Google's Search app's remote tab IAB, and (iii) Meta's custom browser engine IAB on Android.**



12:49 92%

Linkin.bio
linkin.bio

BBC News

Visit BBC News

Your guide to the UK general election

14:43 100%

BBC - Home
bbc.co.uk

Welcome to the BBC

ELECTION

LIVE Labour launches 'wealth creation' manifesto as Sunak makes tax rise attack

UK Politics

INSIDE TRACK 1:14

Wales should get £4bn HS2

Henry Zeffman: Mood in Tory and

13:27 88%

BBC - Home
bbc.co.uk

Welcome to the BBC

ELECTION

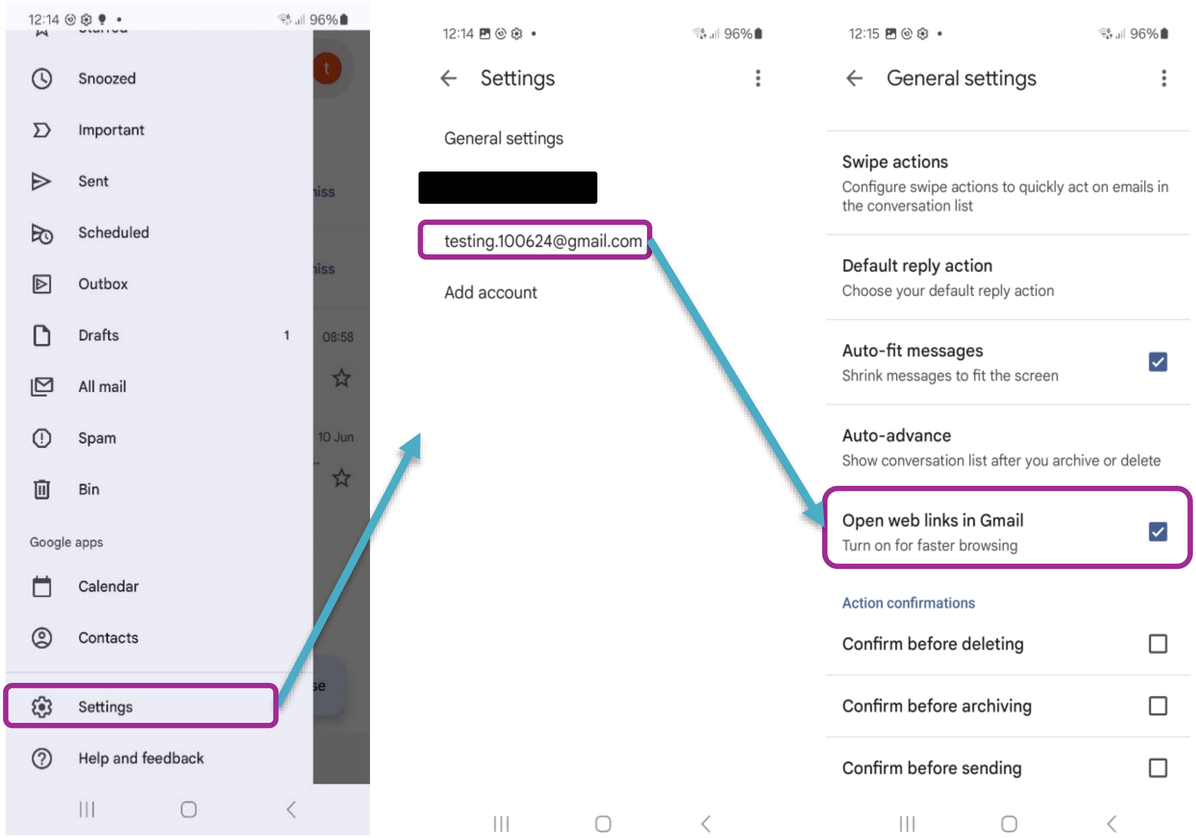
LIVE Rishi Sunak launches manifesto with National Insurance cuts and new Help to Buy

UK Politics

5.22 Android users can change their in-app browsing settings if they wish to in some apps such as the Gmail and Google Search apps (see Figure 5.2).[172] However, we are aware that not all apps allow such controls for users.

[172] Google's response to the CMA's information request [✄].

**Figure 5.2: User journey to disable IAB on Gmail app on Android.**



12:14 96%

Snoozed

Important

Sent

Scheduled

Outbox

Drafts 1

All mail

Spam

Bin

Google apps

Calendar

Contacts

Settings

Help and feedback

12:14 96%

Settings

General settings

[redacted]

testing.100624@gmail.com

Add account

12:15 96%

General settings

Swipe actions
Configure swipe actions to quickly act on emails in the conversation list

Default reply action
Choose your default reply action

Auto-fit messages ☑
Shrink messages to fit the screen

Auto-advance
Show conversation list after you archive or delete

Open web links in Gmail ☑
Turn on for faster browsing

Action confirmations

Confirm before deleting □

Confirm before archiving □

Confirm before sending □

*Note. If 'Open weblinks in Gmail' box is 'ticked', weblinks in Gmail will open in Gmail's webview. If it is 'unticked', weblinks will open in the user's dedicated browser.*

5.23 An app developer told us that it does not generally allow a user to turn off in-app browsing within an app, because the in-app browsing experience is deeply integrated within the app and because user expectations are different as a result.[173] In addition, Microsoft submitted that it is possible to disable non-advertising IAB on LinkedIn on Android (but not on iOS).[174]

## Google's submissions

5.24 Google submitted that it aims to support app developers' ability to customise in-app browsers and balance this with respecting user choice of default browser.[175] Google submitted that it is important for Google to provide a range of options and provide the ability for developers to innovate on top of that. Bespoke solutions (for app developers) as well as the respect for user choice aim to strike the appropriate balance between user and developer choice. Google also submitted that developer and user needs are mostly aligned but acknowledges this is not always

[173] Note of meeting with [✂].
[174] Microsoft's response to CMA's information request [✂].
[175] For example, by default Custom Tabs is set to call the user's default browser, but the app developer can specify a browser and override this default. Note of meeting with Google [✂].

the case.[176] Google considers that 'less developer choice would be bad for innovation and competition as developers would be less able to differentiate their apps – some apps and browser vendors have invested in full development teams for in-app browsing and removing developer freedom could stunt the ecosystem'.[177] Therefore, Google allows app developers to implement webview and bundled engine IABs as bespoke solutions to IAB or Custom Tabs. By default, Custom Tabs respect the user's choice of preferred browser, but the user's default choice can be changed by app developers if they choose to select a specific browser.

5.25 Google submitted evidence suggesting that users are generally aware that they are viewing web content within the app developer's environment, rather than in a dedicated browser.[178] Google highlighted that app developers can customise their in-app browsing interface in Custom Tabs, for example, choose how a Custom Tabs in-app browser is opened or closed, the colour scheme and the content of the in-app browser's toolbar and menus.[179] Google also submitted that it allows app developers to specify a version of Custom Tabs – other than relying on the user's default choice in all instances – because developers may have a preference and certain features may not be available across all versions of Custom Tabs.[180]

5.26 Google submitted that when the core Android apps Google licenses under the European Mobile Application Distribution agreement (plus the Google Search app), as well as their counterparts for iOS (Play, Google Search, Maps, Gmail, YouTube, Drive, YouTube Music, Google TV, and Photos), open links in an in-app browser, all of these apps also present the user with options/settings to open the content in one or more separate browser app, including their default browser.[181] Google also submitted that the same prompts are shown to the users within Custom Tabs as they would see when browsing in the underlying browser, such as Chrome. This might apply when users are in-app browsing and these prompts might encourage users to change their default browser.[182]

5.27 Google commented that users care about having a seamless experience within an app and sometimes this requires developer freedom to customise in-app browsing to suit the needs of their app.[183] Evidence submitted by Google suggests that some users and developers care about consistency and Custom Tabs allows developers to offer the same built-in features across web experiences.[184] In

[176] Note of meeting with Google [✂].
[177] Note of meeting with Google [✂].
[178] Google submitted that it explored [✂]. Google's response to the CMA's information requests [✂].
[179] Google's response to the CMA's information requests [✂].
[180] For example, Google identified two Custom Tab providers that do not support the feature 'dark mode'. Google's response to the CMA's information request [✂].
[181] Google's response to the CMA information request [✂].
[182] Note of meeting with Google [✂].
[183] Note of meeting with Google [✂].
[184] Google's internal document [✂].

particular, Google conducted a small scale qualitative study which showed that [✂].[185]

5.28 Google highlighted that the extent to which user knows they are using the default browser when in-app browsing is dependent on the app developer or, for Custom Tabs IABs, the vendor of the underlying browser. For example, on Android, users can see they are in Custom Tabs by tapping the three dots (vertical ellipsis IAB choice menu – see paragraph 2.55 and Figure 2.3) at the top right of their screen, where it might say ‘Open in Samsung Internet’. Chrome Custom Tabs also has the ‘Running in Chrome’ branding to inform the user.[186]

### Evidence from third parties

5.29 We asked third parties about their views on users’ choice, control and awareness within in-app browsing on Android. Overall, third parties submitted that app developers on Android have more flexibility in selection of IAB implementations.

5.30 Snap reported that taking the user to their dedicated default browser has advantages for the user’s experience (eg pre-saved personal, payment and login information).[187]

5.31 [✂] submitted that it did not consider allowing users to set their own IAB default, as the in-app browsing experience is deeply integrated within the app. [✂] custom browser IAB improves user experience through the ability to deliver more integrated content. For example, operating its own engine means that [✂] can offer new features such as an improved WebShare API experience - meaning an integrated sharing API which would have [✂] logos (for example, this would make it easier for users to share a link within the [✂] app).[188]

5.32 On Android, OWA is particularly concerned about users’ default choice not being carried over and its impact on privacy and security in relation to bundled engine and webview implantation of in-app browsing. OWA also highlighted the fact users may be unaware that in-app browsers allow for JavaScript injections and this has the potential for unwanted user surveillance.[189]

### Evidence from consumer research

5.33 As described in the section on Apple’s policies that affect user control and choice in relation to in-app browsing, the CMA commissioned qualitative research and a quantitative survey conducted by Verian on users’ awareness, understanding and

[185] Google submitted that it ran [✂]. Google’s response to the CMA’s information request [✂].
[186] Note of meeting with Google [✂].
[187] Note of meeting with Snap [✂].
[188] Note of a meeting with [✂].

[189] Note of a meeting with OWA, [✂].

behaviour in relation to mobile browsers and in-app browsers. The findings from Verian consumer research apply across both iOS and Android devices (see paragraphs 4.45 to 4.53).

### Summary of emerging thinking on Google's policies that affect user control and choice relating to in-app browsing

5.34 Overall, users have limited choice and control in relation to which browser is used for IAB implementations in native apps that they use and IAB in general because:

(a) If a user's default browser is not Chrome, any settings they have for browsing in their dedicated browser might not be applied to in-app browsing on Android.

(b) The in-app browsing visual interfaces on Android mimic the 'actual' Chrome browsing experience with some visual differences between the different IAB implementations. Consequently, users often struggle to distinguish whether they are using their default browser or an in-app browser.

(c) Related to this, consumer research showed very low levels of awareness and limited understanding of in-app browsing among users, who frequently did not realise they were using an IAB instead of their chosen browser. This lack of awareness extends to the features they prefer, such as privacy and security settings.

(d) Ultimately, users are generally not provided with the same level of controls over their in-app browsing experience, which can vary across different apps (eg the presence of an 'IAB menu'). Additionally, at the device setting level, users often cannot turn off IAB on Android.

5.35 This means that users' control and choice is likely to be a very limited constraint on the extent of competition between providers of IABs and browsers.