MOBILE BROWSERS AND CLOUD GAMING MARKET INVESTIGATION

WP3: Access to browser functionalities within the iOS and Android mobile ecosystems

27 June 2024

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 **22nd 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 [✂]. Non-sensitive wording is also indicated in square brackets.

Contents

1. Introduction

1.1 Mobile browsers (otherwise described in this paper as 'browsers') are applications that enable users of mobile devices to access and search the world wide web and interact with content on it. Browsers rely on browser engines to render or transform web page source code into content that users can engage with. [1]

1.2 The two most used mobile browsers are Apple's Safari and Google's Chrome. Apple and Google also run the two main browser engines: all browsers on iOS must run on Apple's Webkit browser engine and Google's Blink engine is widely used on Android, although on Android browsers may use other engines. Other browsers include Mozilla Firefox, Opera, and DuckDuckGo.

1.3 As set out in the Issues Statement for this market investigation, one of the issues being considered in this investigation is whether Apple and Google are using their position in the supply of browser engines to restrict rival browsers' access to functionality which is available in the WebKit and Blink browser engines.[2]

1.4 This working paper considers the extent to which Apple and Google could be using their position in the supply of browser engines and mobile operating systems on iOS and Android devices to restrict access to important functionality for rival browsers, which may limit their ability to compete effectively with Safari and Chrome.[3] In particular, we consider:

(a) Whether Apple is currently providing rival browsers operating on iOS devices with the same level of access to functionality as its own browser, Safari and if it is not, the likely impact that such lack of access has on competition between browsers, by limiting the features and functionality that rival browsers can offer.

(b) Whether Google is currently providing other browsers operating on Android devices with the same level of access to functionality as its own browser, Chrome and if it is not, the likely impact that such lack of access has on competition between browsers, by limiting the features and functionality that rival browsers can offer.

1.5 This working paper also considers the extent of support for browser extensions[4] on iOS and Android mobile devices, and the implications of this for users and developers of web content.

---

[1] As covered in 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine', on iOS browser vendors are required to use a specific version of WebKit controlled by Apple. On Android, although browser vendors can use a browser engine of their choice, most use Blink.
[2] Issues statement, paragraph 27(c).
[3] This workstream relates to the issues set out in the Issues Statement at paragraphs 27(c), 36 and 37.
[4] Browser extensions are additional software applications that can add functionality or features to a browser and enable users to customise their browsing experience.

1.6 This paper should be read alongside ‘WP1 - Nature of competition in the supply of mobile browsers and browser engines’, particularly the section on parameters of competition for browser engines and browsers.

1.7 The issues explored in this paper are in addition and separate to the issues set out in ‘WP2 - The requirement for browsers operating on iOS devices to use Apple’s WebKit browser engine’, which relate to Apple’s requirement that all browsers on iOS devices use a specific version of Apple’s browser engine, WebKit.[5] For example, if Apple is able to restrict rival browsers’ access to operating system and hardware functionalities, then WebKit and Safari may have greater ability to innovate and improve than alternative browsers and browser engines on iOS. Equally, if rival browsers vendors that continue to use WebKit on iOS have restricted access to browser engine functionalities relative to Safari, then they may be less able to innovate and improve their browsers.

1.8 This paper is structured as follows:

(a) Section 2 provides an explanation of what browser functionality is and why it is important in the context of competition between browsers.

(b) Section 3 provides an overview of the evidence we have received to date on access to browser functionality on iOS and the likely impact of this on competition between browsers.

(c) Section 4 provides an overview of the evidence we have received to date on access to browser functionality on Android and on the likely impact of this on competition between browsers.

(d) Section 5 provides an overview of the evidence we have received to date on the extent of support for browser extensions on both iOS and Android.

[5] As described in ‘WP2 – The requirement for browsers operating on iOS devices to use Apple’s WebKit browser engine’, browsers on iOS are required to use a specific version of WebKit, WKWebView, provided as a system framework.

2. Browser functionality in the context of competition between browsers

2.1 References in this paper to 'access to browser functionality' refer to the ability of browsers to access functionality from the relevant browser engine, operating system, or device hardware.

2.2 Access to browser functionality is important in allowing browser vendors to innovate and implement features in their browser, including user-facing features, security features, and privacy features that enable browser vendors to innovate and improve their products. Access to browser functionality may also affect the information available to browser vendors about use of their browser on mobile devices, and therefore potentially impact their ability to optimise and improve the performance of their browser.

2.3 Browser functionalities are often provided by operating system or browser engine suppliers through application programming interfaces (**APIs**). APIs act as a software intermediary that allow two applications to communicate with one another, and to exchange data, features, or functionality. One application sends a request, and the second application provides a response, with the API acting as the connection between the two applications. For example, a social media application sending a request to a camera application to allow a user to take or upload photos.

2.4 Browsers rely on APIs in order to access certain features and functionalities. For example, APIs allow access to device hardware such as the microphone, or can be used to request data on the user's default browser, allowing the browser to prompt the user to change their default.[6] Access to APIs is also important to enable browser vendors to implement features and improvements in their browsers, and is therefore important to innovation and product development.

2.5 This paper considers the following types of features of browsers or other browser functionality that may need to access APIs:

(a) User-facing features – including features relating to the user's experience of a browser. Examples include: full screen API,[7] which allows content to be presented in full screen; push API,[8] which allows browsers to deliver push notifications; and gamepad API,[9] which allows for interaction with gamepads.

[6] Mozilla Developer Network (MDN) provides a list of APIs that may be used in web development.
[7] Full screen API, accessed by the CMA 07 May 2024.
[8] Push API, accessed by the CMA 07 May 2024.
[9] Gamepad API, accessed by the CMA 07 May 2024.

(b) Security features – including product features that improve the security of a browser – for example, process separation or site isolation, which involve running different websites in different processes to improve security.

(c) Privacy features – including product features that impact how data from the user of a browser is used. For example, features which limit tracking of user data, or provide users with control over what data websites have access to (eg location data).

(d) Information availability – including browser access to data or analytics, for example the ability of a browser to access data on a user's default browser.

2.6 Innovations, new features, or improvements to browsers may be implemented at different levels of the software stack within a mobile ecosystem. Many improvements to the performance of a browser happen at the browser engine level ie through changes to the browser engine code, as do many security features such as site isolation[10] (assuming no restrictions are placed at this level, for example the WebKit restriction which prevents browser vendors making changes to the browser engine code on iOS). Additionally, improvements can also happen at the browser level ie within the browser code. For example, changes to the user interface, or features such as password managers can be incorporated at the browser code level. In some cases, browser vendors may have some flexibility in deciding at which level to build a feature. In both cases, adding features may require access to functionality from the operating system or device hardware.

2.7 In the same way, restrictions on access to browser functionalities may occur at different levels. For example, it may be that third-party browsers and browser engines are not granted equal access to operating system or hardware functionalities; or there may be restrictions to access to functionality at the browser engine level alone ie a third-party browser not being granted equal access to functionality within the browser engine. This working paper does not seek to specify the level within the software stack that access may be required for particular browser functionalities, as submissions from parties have not generally specified this, and it does not affect the analysis of the impact on competition set out below.

2.8 Finally, the functionalities that browser vendors require access to so they can improve their browsers are likely to change over time as the capabilities of operating systems and device hardware evolve, and new browser features or innovations are developed. Enabling access to these functionalities in a timely manner may therefore be important to enable browser vendors to innovate.

[10] Site isolation prevents a single browser bug from impacting multiple sites operating in the same tab.

2.9 There may also be instances where, whilst third-party browsers are technically able to access a particular functionality, their access is made more difficult or is delayed, relative to Safari or Chrome. This may limit ability of browser vendors to innovate or improve their products in an equal manner compared to Safari or Chrome.

2.10 Finally, the way in which access to APIs is communicated to developers and documented is important to browser vendors' ability to make use of these APIs, as considered further below.

3. Access to browser functionalities on iOS

3.1 This section considers evidence that we have received to date on whether rival browsers on iOS are able to access browser functionality on iOS and our preliminary assessment of the extent to which this may impact competition between browsers.

3.2 Apple has made general submissions that it does permit substantial differentiation between browsers and allows browser vendors to build features and interfaces on top of its WebKit browser engine, while upholding Apple's stringent privacy and security protections. Apple submitted that it does not dictate what features ship on third-party browsers and that other developers which control third-party browsers are free to build features into their browsers that are not available in Safari.[11]

3.3 Further, Apple submitted that it makes more than 250,000 APIs publicly available to third-party developers on equal terms. Apple submitted these allow developers to offer high-quality apps and services with powerful capabilities. It submitted that it puts substantial effort into maintaining and preserving compatibility from one release to the next, continuing to enhance publicly available APIs, and ensuring those APIs work well for developers across new OS and new hardware releases.[12]

3.4 [✂] Apple submitted that Safari's role as Apple's browser allows Apple to efficiently design, test, revise and ship features, and to ensure that new features do not compromise user privacy and security;[13] and that entitlements[14] are a means by which Apple can provide early access to hardware or software to limited groups of developers in order to test new features.[15]

3.5 The sub-sections below consider evidence from Apple and third parties in relation to access to specific browser functionalities on iOS, relating to: (i) user-facing features; (ii) security features; (iii) privacy features; and (iv) information availability.

User-facing features

3.6 This sub-section covers evidence from third parties on user-facing features and functionalities that third parties submit are supported by Safari but not available to other browsers on iOS.

[11] Apple's response to CMA's information request [✂].
[12] Apple's response to CMA's information request [✂].
[13] Apple's response to CMA's information request [✂].
[14] Entitlements are controls on the iOS operating system resources that may be accessed by apps or other software.
[15] Apple's response to CMA's information request [✂].

3.7 First, several third-party browser vendors submitted that Safari is the only browser on iOS that can make full use of users' saved passwords or have the ability to allow the user to autofill their passwords:[16]

(a) Vivaldi submitted that in 2022, its users could not use their iCloud Keychain[17] passwords in browsers other than Safari. It submitted that users had to copy passwords from iCloud password manager and paste it to other places, which was 'tedious'. It submitted that this led to trends of users opting for Safari since it synced everything together seamlessly.[18] Vivaldi submitted that as of May 2024 this functionality is 'not as restrictive as it once was, since iCloud Keychain passwords can now be used in other browsers besides Safari.'[19]

(b) A browser vendor submitted that creating new credentials in iCloud Keychain is not supported in WKWebView[20] meaning users cannot create passwords from WKWebView. This browser vendor submitted that Safari does offer support for creating new credentials for iCloud Keychain, but not for other credential providers.[21]

3.8 Apple submitted that third-party browsers can use WKWebView for autofill or build their own password managers on WebKit, and store passwords associated with their web domains in their own managers. However, third-party browsers cannot store passwords associated with unaffiliated domains.[22] Apple submitted it has not seen 'sufficient indications of demand in order to prioritize development of a mechanism to allow this feature.'[23] Apple later submitted that third-party browsers can, and do, store passwords from unaffiliated domains in their own password managers using keychain technology.[24]

3.9 Second, a browser vendor submitted that Safari can show a more extensive context menu when the user 'long presses' on an image. Context menus give developers ways to add more menu options and entry points to the Safari app extension. Apple has not shared access to the relevant APIs to enable third-party browsers on iOS to access this feature. This browser vendor stated that this allows Safari to have superior user interaction on this feature.[25]

3.10 Apple submitted that third-party browsers call on a WebKit API to add 'Share' button functionality to the context menu available through a long press on web

[16] Responses to CMA's information requests [✂].
[17] iCloud Keychain is Apple's system that allows users to save passwords, credit cards, and other private information across all Apple devices.
[18] Vivaldi's response to CMA's information request [✂].
[19] Vivaldi's response to CMA's information request [✂].
[20] WKWebView is the system framework provided by WebKit, which all third-party browsers on iOS are required to use.
[21] [✂] response to CMA's information request [✂].
[22] A web domain that is not connected to the browser vendor.
[23] Apple's response to CMA's information request [✂]; Apple's response to CMA's information request [✂].
[24] Apple response to information put back to it [✂].
[25] [✂] response to CMA's information request [✂].

page content. Apple submitted that interaction with the Firefox app demonstrates that it currently makes use of this API.[26]

3.11 Third, several browser vendors submitted that they are unable to see the default browser that a user has selected on their mobile devices, but that Safari is able to track this. This limits third-party browsers' ability to monitor their data and accurately market their browser: [27]

(a) A browser vendor submitted that third-party browser vendors on iOS do not have the ability to see if the user has selected their browser as the default browser. This browser vendor submitted this leads to unnecessary promotion to the user from browsers that they have already set as a default. This browser vendor [✂].[28]

(b) Yandex submitted that Safari is the only browser that is able to see if a user has set it as a default browser.[29]

3.12 Apple submitted that neither Safari nor third-party browsers can track default browser settings by individual users.[30]

3.13 Fourth, Vivaldi submitted that third party browsers are limited in their ability to implement Reader Mode. It stated that Reader Mode provides an optimised way to read articles by stripping away unnecessary content such as ads, sidebars, and other distractions. Vivaldi submitted that whilst it is technically possible for third-party browsers to implement Reader Mode, it is not available in 'standard WebKit'. Vivaldi submitted that Reader Mode also adjusts text size, background colour and layout for better readability.[31]

3.14 Apple submitted that Reader Mode is a Safari feature, and its functionality is specific to Safari. Apple submitted that third-party browsers have the ability to implement their own version of Reader Mode via WebKit APIs.[32]

3.15 Fifth, several third parties submitted that whilst Safari can offer browser extensions on iOS, the same functionality is not available to third party browsers.[33] Browser extensions are additional software applications that can add functionality or features to a browser and enable users to customise their browsing experience. Supporting browser extensions means that a browser allows third-party

[26] Apple's response to CMA's information request [✂].
[27] Responses to CMA's information requests [✂].
[28] [✂] response to CMA's information request [✂].
[29] Yandex's response to CMA's information request [✂].
[30] Apple's response to CMA's information request [✂].
[31] Vivaldi's response to CMA's information request [✂].
[32] Apple's response to CMA's information request [✂].
[33] Whilst this section considers whether Apple limits access to browser extensions for third-party browsers relative to Safari, further background on browser extensions and concerns around the extent of support for browser extensions more broadly on iOS and Android are considered in section 5.

developers to create and offer extensions, and allows users to access a catalogue of these extensions:[34]

(a) A browser vendor submitted that Safari on iOS started supporting extensions from iOS15 (released in 2021). However, third-party browsers are not allowed to offer this functionality despite it being built into WebKit.[35]

(b) Mozilla submitted that Safari supports extensions distributed on the iOS App Store. However, third-party browsers are prevented from offering their own established extension functionality. Mozilla submitted that this prevents Firefox from offering the same functionality as Safari.[36]

(c) Brave submitted that iOS15 offered extensions on mobile version of Safari for the first time, but that third-party browsers do not have access to this functionality.[37]

(d) OWA submitted that only Safari can offer extensions on iOS. It submitted that extensions are used by many users, including to block advertising, and that if third party browsers do not have the ability to set extensions, users may choose to use Safari for the advantage some of these extensions bring.[38]

3.16 Apple has made several points regarding its restriction of browser extensions:

(a) In March 2022, Apple submitted that third-party browsers were not able to offer 'comparable features and functionality' to Safari for browser extensions, as it had not yet determined that this was technically feasible.[39]

(b) In February 2023, Apple submitted that third-party browsers are free to implement web extensions functionality on top of WebKit. Apple also submitted that web extensions give rise to an additional risk because a fourth party is involved and submitted that web extensions present both a security and privacy risk depending on the implementation of the extension model. It stated that many extensions request access to every site that a user visits within a browser, and many require the user to grant the extension access to all websites in order to use the extension at all within the browser, and that this could pose significant privacy risks. It stated that Apple's extension distribution model ensures that Safari users know that the extension developer has access to a specific webpage. For example, if a user is

[34] Responses to CMA's information requests [✂].
[35] [✂] response to CMA's information request [✂].
[36] Mozilla's response to CMA's information request [✂]; See also platform-tilt, accessed by the CMA 18 June 2024.
[37] Brave's response to CMA's information request [✂].
[38] OWA Bringing Competition to Walled Gardens, section 5.3.1, accessed by the CMA 31 May 2024.
[39] Apple's response to CMA's information request [✂].

accessing a banking website and must accept a web extension, that could put private bank account information at risk.[40]

(c) Apple submitted in April 2024 that Safari supports a variety of web extensions through WebKit, and third-party browsers are free to build and implement web extensions functionality on top of WebKit. Apple submitted that third parties can build on top of WebKit in the same way that Safari does, and it pointed to Orion as an example of a third-party browser that has done this.[41] Apple submitted that third-party browsers can use their own extension catalogues with a web-based distribution model.[42]

(d) Apple submitted that it does not currently vet third party extensions unless they are offered in Safari. With respect to the safeguards that could be put in place to ensure users are informed of the implications of this third-party feature, it stated that in theory Apple could mitigate the risk by asking third-party browsers to use WKWebView and provide additional warnings and explanations of risk associated with an unknown fourth party.[43]

3.17 Sixth, several third parties have submitted that on iOS, Safari is the only browser that can install web apps. This prevents third-party browsers from offering the same level of functionality as Safari:[44]

(a) Microsoft submitted that Safari was the only browser that can install (or pin) to an iOS device's home screen. It submitted that this restriction undercut potential competition between Progressive Web Apps (**PWAs**) and the native apps made available by Apple's App Store business, by depriving competing browsers of the ability to offer safe PWAs. Microsoft submitted that this restriction 'was lifted in theory with iOS16.4 but in practice remains'.[45][✂][46]

(b) OWA submitted that web apps cannot be installed by third party browsers and can only be installed by Safari.[47]

3.18 On the issue of installing or adding web apps to the home screen, in 2023 Apple submitted that giving third-party browsers unfettered ability to add web apps to the home screen would present both a security and privacy risk:

[40] Apple's response to CMA's information request [✂].
[41] Apple's response to CMA's information request [✂].
[42] Apple's response to CMA's information request [✂].
[43] Apple's response to CMA's information request [✂].
[44] Responses to CMA's information requests [✂].
[45] Microsoft's response to CMA's information request [✂].
[46] [✂] response to CMA's information request [✂].
[47] OWA Bringing Competition to Walled Gardens, section 5.3.1, accessed by the CMA 31 May 2024.

(a) Apple submitted that there is a security risk because the third-party browser could be compromised and coerced into adding malicious content to a user's home screen without the user's knowledge, among other types of attacks.[48]

(b) Apple submitted that there is also a privacy risk because a third-party browser could install fraudulent web apps that intercept and gather sensitive user data without the user's consent.[49]

(c) Apple submitted that the risk is lower in Safari because Apple has control over security safeguard development standards in Safari. In addition, Apple submitted that the software components for iOS, Safari and WebKit provide Apple with control over both the metadata path to installation and the user interface flow. Apple submitted that this gives it the ability to ensure that users are knowingly making the choice to install a web app despite the risks associated with it. [50]

(d) Apple submitted that it has created an implementation that allows third party browsers to add web apps and websites to a user's home screen. Apple submitted that to mitigate the risks described above, the implementation includes a system user interface that requires users to take affirmative steps, similar to those required in Safari, before having the ability to add a web app to the home screen through a third-party browser. [51]

(e) Apple submitted that it expanded Add to Home Screen functionality to third-party browsers on iOS in 2023.[52] This indicates that despite the risks highlighted above, Apple has ultimately been able to extend this functionality to third-party browsers.

3.19 Seventh, the Guardian submitted that 'universal linking' is only available to Safari. Universal linking is when a native app is launched from a user clicking a link in a browser. The restriction on access to universal linking means that if a user clicks a link in a third-party browser, the link will take them to the website, and not the app. The Guardian submitted that this adds user friction and might show the user messaging that is inconsistent with their expectations.[53]

3.20 In 2023, Apple submitted that it restricts universal linking to Safari because giving third-party developers access would present both a security and a privacy risk. Apple submitted that from a security perspective, if third-party apps could gain knowledge of what apps are installed on a user's device, they could compromise the security of the installed app database. Apple submitted that from a privacy

[48] Apple's response to CMA's information request [✂].
[49] Apple's response to CMA's information request [✂].
[50] Apple's response to CMA's information request [✂].
[51] Apple's response to CMA's information request [✂].
[52] Apple's response to CMA's information request [✂].
[53] The Guardian's response to CMA's information request [✂].

perspective, a third-party browser that is aware of what apps are installed on a user's phone could easily fingerprint[54] a user without their knowledge. [✂.] [55]

3.21 Eighth, a browser vendor submitted that Safari uses a feature that allows it to [✂]. This browser vendor also submitted that in [✂], it filed a request to Apple for an API allowing the browser vendor to do this but by May 2024 had received no response.[56]

3.22 Ninth, a browser vendor submitted that Safari uses a feature that allows it to [✂]. It submitted that its browser is not able to use this feature and that on [✂] it requested for it to be made public. However, as of May 2024, the browser vendor had received no response. This feature is called the [✂] method.[57]

3.23 Tenth, Yandex submitted that Service Worker is a script that a browser runs in the background separately from a webpage, opening the door to capabilities that do not require a webpage or user interaction, such as push notifications and background synchronisation. It submitted that support for Service Worker is limited to Safari and that webpages requiring Service Workers therefore only function in Safari. This limits the number of webpages that can run on browsers other than Safari and makes the browser less attractive to developers.[58]

3.24 Apple submitted that, after introducing support for Service Workers in 2018, Apple expanded access to third-party browsers.[59]

3.25 Eleventh, a browser vendor [✂] submitted that for ten years, there were two versions of WebKit; one version was reserved for Apple's use and another that was slower and only available for third parties. However, the browser vendor [✂] submitted that this restriction has now been lifted and the fast version is available for all.[60]

3.26 Twelfth, OWA submitted that users on Safari are able to make videos full screen but that other browsers are prevented from doing so (except on iPad). OWA also submitted that the inability for third-party browsers to make videos full screen makes them inferior to Safari at delivering video streaming and game streaming services.[61]

3.27 Thirteenth, Epic Games submitted that the latest version of WebKit supports Web Real-Time Communication ('WebRTC') which allows for real-time communications

---

[54] Fingerprinting is when a developer collects data about a device (such as device model, screen size, system fonts, and time zone) and then aggregates and transforms that data to uniquely identify the device.
[55] [✂]'s response to CMA's information request [✂].
[56] [✂] response to CMA's information request [✂].
[57] [✂] response to CMA's information request [✂].
[58] Yandex's response to CMA's information request [✂].
[59] Apple's response to CMA's information request [✂].
[60] Note of meeting with [✂].
[61] OWA Bringing Competition to Walled Gardens, section 5.3.1, accessed by the CMA 31 May 2024.

such as video conferencing and screen sharing, but that this feature is reserved to Safari and cannot be accessed by third-party developers of browsers.[62] We understand that this restriction was in place on 23 November 2022 but has since been removed.[63]

3.28 Fourteenth, Epic Games submitted that WebKit now supports UserMedia, which allows apps to access device hardware such as the camera and microphone. However, Epic Games submitted that only Safari can make use of this feature and third-party browsers cannot.[64] We understand that this restriction was in place on 23 November 2022 but has since been removed.[65]

3.29 Fifteenth, in 2022, Apple submitted that it restricts third-party browsers from being able to download and upload data in the background, without being open. Apple submitted that this is because there is a technical risk to stability as browsers could use up computing resources while running in the background. [66] Apple also submitted that this functionality poses a security risk because it could cause a device to become unstable, lock up, or run out of battery power, which is a common scam to sell fake technical support and/or ransomware. Additionally, Apple submitted that this functionality also creates a privacy risk because permanent background execution enables an app to track a user's location and behaviour over time, which it otherwise would not be able to do. [✂][67] This restriction has not been raised by any third parties.

3.30 Sixteenth, Brave submitted that Apple Pay only used to work on Safari, but that it has had access to Apple Pay resources since iOS16/Safari16 (released in September 2022). Brave submitted that before this third-party browsers did not have access to APIs or entitlements required to implement Apple Pay on their browsers and that the APIs were not open.[68]

3.31 Seventeenth, several third parties submitted that Safari integrates with Apple native apps in a way that other browsers cannot replicate. For example:

(a) A browser vendor submitted that opening a link from iMessage in Safari displays a banner on top with the contact info of the person that sent the user the link, with the ability to quickly write messages back to them. This browser vendor submitted that it cannot replicate this feature on iOS.[69]

---

[62] Epic Games' response to CMA's information request [✂].
[63] Epic Game's response to information put back [✂].
[64] Epic Games' response to CMA's information request [✂].
[65] Epic Games' response to information put back [✂].
[66] Apple's response to CMA's information request [✂].
[67] [✂] response to CMA's information request [✂].
[68] Note of meeting with Brave [✂].
[69] [✂] response to CMA's information request [✂].

(b) Brave submitted Safari uses 'hide my email' and iCloud+ features, but they are not available to third-party browsers. It stated this would add significant consumer value to its browser. [70]

3.32 Finally, some third parties submitted that users cannot import data from Safari into third party browsers:

(a) Ecosia submitted that there are a number of areas where it currently lacks sufficient interoperability with iOS hardware and software features, and the most useful of these would be much of the information that exists within Safari or sits within the Cloud. Ecosia submitted that currently, even if a user were to grant permission for Ecosia to access this key information, Ecosia cannot import the data, which pushes the user back to Safari. Ecosia stated that Apple does not offer the ability for the user to export bookmarks to an HTML file, meaning that users cannot carry their data to a third party such as Ecosia.[71]

(b) A browser vendor submitted that third party browsers cannot import bookmarks from Safari.[72]

Security features

3.33 This sub-section covers evidence from third parties on security features and functionalities that third parties submit are supported by Safari but which Apple does not make available to other browsers on iOS.

3.34 First, Mozilla submitted that for many years, Apple did not make available the WebKit API that is necessary for other browsers to offer the Safebrowsing feature. Mozilla submitted that as a result, only Safari offered this feature. Mozilla submitted that Apple made some changes that extended Safebrowsing to other WebKit browsers, however the implementation is restrictive and prevents third-party browsers from fully controlling how the Safebrowsing service is offered in-product.[73]

3.35 Apple submitted that third-party browsers have equal ability to develop SafeBrowsing functionality for their apps, such as via partnerships with other third parties like Google or Tencent. Apple submitted that it does not prevent browsers from developing this functionality, and others have developed it. Apple submitted that Firefox uses the Google Safe Browsing API.[74]

---

[70] Note of meeting with Brave [✂].
[71] Ecosia's response to CMA's information request [✂].
[72] [✂] response to CMA's information request [✂].
[73] Mozilla's response to CMA's information request [✂].
[74] Apple's response to CMA's information request [✂]; Apple's response to CMA's information request [✂].

3.36 Second, Mozilla submitted that 'Process Separation' is a critical operating system feature that is needed for browser developers which allows for greater stability, quality, and security. It submitted that Safari makes use of this feature, but it is explicitly disabled for third-party browsers.[75]

3.37 Apple submitted that third-party browsers have equal access to process separation through WebKit, which creates a new process for each webpage loaded in order to segregate any instability or bugs and prevents them from affecting the overall performance of iOS.[76]

3.38 Third, Microsoft submitted that Safari is the only browser that can be relied upon to authenticate the user to a network.[77, 78]

3.39 Apple submitted that third-party browsers have equal access to the ability to authenticate users for wireless networks.[79]

3.40 Fourth, Microsoft submitted that Safari is the only browser with direct access to certificates deployed through mobile device management systems. These are commonly used by enterprises for certificate-based authentication.[80, 81]

3.41 Apple has submitted that both Safari and third-party browsers are limited in their ability to access certificates through mobile device management systems on iOS.[82]

3.42 Fifth, a browser vendor submitted that its browser's implementation of 'copy image' on iOS cannot 'grab' the already downloaded image from WKWebView's cache but must re-download it and decode the image through WKWebView (which it submitted presents a potential security vulnerability). The browser vendor submitted this is because Apple restricts access to certain APIs that allow third-party browsers to implement features that Safari is already able to implement on iOS.[83]

3.43 Sixth, a browser vendor submitted that Apple limits its browser's ability to verify the identity of the user for security purposes which also hinders the browser's ability to create a more tailored experience for its users on iOS. Non-Apple apps are unable to interact with the iOS certificate store. This means that installing enterprise

---

[75] Mozilla's response to CMA's information request [✂].
[76] Apple's response to CMA's information request [✂].
[77] Some websites provide, as a service, a secure mechanism for authenticating users. When the user navigates to the site's authentication URL, the site presents the user with a form to collect credentials. After validating the credentials, the site redirects the user's browser, typically using a custom scheme, to a URL that indicates the outcome of the authentication attempt; See 'Authenticating a user through a web service'.
[78] Microsoft's response to CMA's information request [✂].
[79] Apple's response to CMA's information request [✂].
[80] Mobile device management systems allow enterprises or organisations to secure, manage, and monitor employees mobile devices.
[81] Microsoft's response to CMA's information request [✂].
[82] Apple's response to CMA's information request [✂].
[83] [✂] response to CMA's information request [✂].

profiles (ie information on the identity of a user) cannot be done through a third-party browser, including its browser on iOS.[84]

3.44 Finally, some third parties have submitted that only Apple is able to modify the WebKit Just In Time (JIT) compiler and that this limits third-party browsers' ability to compete on performance or the feature set of their JIT compiler. JIT is where the code compilation is done before the execution of the code, unlike with a compiled language. A JIT compiler is important for rendering web content that contains JavaScript code, as most websites do. Apple added a JIT compiler to its browser engine WebKit in 2014, and most modern browser engines use JIT compilers:

(a) Microsoft submitted that because of API restrictions on WebKit, iOS browser developers can only implement a system-wide JIT-free setting which cannot be applied on a per-site or content-aware basis. It submitted that only WebKit and Safari can support sub-processes and configure a sandbox for web content. Microsoft submitted that this prevents it from differentiating its browser on iOS with its strong sandbox for content.[85]

(b) OWA submitted that only Safari is allowed to implement or modify its own JIT compiler. It submitted this means that other browser vendors are unable to compete on performance or the feature set of their JIT compiler.[86]

Privacy features

3.45 This sub-section covers evidence from third parties on privacy features that are supported by Safari but which third parties submit are not available to other browsers on iOS.

3.46 First, some browser vendors submitted that Apple's iCloud Private Relay feature, which routes traffic through a VPN and protects users from IP fingerprinting, is not available to third-party browsers. This limits third-parties browsers' ability to offer the same level of privacy as Safari:

(a) A browser vendor submitted that third-party browsers are not supported and are not able to offer the same functionality. This is because WKWebView does not support customising the network layer.[87]

(b) Microsoft submitted in 2021 that Apple's new Private Relay feature was expected to only be available to Safari.[88]

---

[84] [✂] response to CMA's information request [✂].
[85] Microsoft's response to CMA's information request [✂].
[86] OWA's response to CMA's information request [✂].
[87] [✂] response to CMA's information request [✂].
[88] Microsoft's response to CMA's information request [✂].

3.47 Apple submitted that Private Relay is an iCloud privacy feature and not Safari-specific, meaning that third-party browsers could develop a proxy for themselves to provide a similar offering. Apple submitted that Google One (Google's cloud storage service) currently provides a system-wide VPN offering and Google could choose to make a Chrome-specific offering as well.[89]

3.48 Second, a browser vendor submitted that non-Safari browsers, until the release of iOS 17 in September 2023, were unable to honour cookie storage settings or let users view per-site location permissions. This means that users were presented with more prompts in third-party browsers than in Safari, which could cause an inconvenience for the user by having to select permission on multiple occasions and therefore impact the user experience on third-party browsers on iOS.[90]

3.49 Third, Mozilla submitted that, prior to 2016, browsers were able to offer various features that are necessary for privacy functionality. These features included data saving, cookie settings, multi-profiles, enterprise support and auto-detection encoding. Mozilla submitted that in 2016, Apple made changes that 'broke existing functionality and impeded new feature development'. Mozilla submitted Apple has also not engaged with bug requests from different browser developers seeking to return these APIs.[91]

3.50 Fourth, Mozilla submitted that only Apple had access to Intelligent Tracking Protection from 2017 to 2020, which is a framework to limit cross-site tracking by websites. Mozilla submitted that this left Firefox users on iOS with a disadvantage compared to users of Safari on iOS.[92]

3.51 Fifth, Mozilla submitted that Apple removed support for 'Do Not Track' for third-party browsers in 2016. Mozilla submitted that Apple allowed Safari to keep this feature until 2019, when it also removed it from Safari.[93]

3.52 [✂].[94]

Information availability

3.53 This sub-section covers evidence from third parties on documentation and support for APIs, and access to browser analytics.

[89] Apple's response to CMA's information request [✂].
[90] [✂] response to CMA's information request [✂].
[91] Mozilla's response to CMA's information request [✂].
[92] Mozilla's response to CMA's information request [✂].
[93] Mozilla's response to CMA's information request [✂].
[94] [✂] response to CMA's information request [✂].

Documentation and support for APIs

3.54 Clear guidance or documentation from Apple in relation to the use of browser APIs is important if browser vendors are to be able to make proper use of APIs and add new features into their browsers.

3.55 A browser vendor submitted that its browser on iOS suffers from limited information as compared to Safari, which affects its browser's performance on iOS. The browser vendor stated that there is a category of APIs that are unusable due to low quality support offered by Apple. According to the browser vendor, developer resources such as caniuse.com show the features as supported, which adds more confusion and frustration for developers. For example:

(a) IndexedDB API was first delayed by two years, but when initial support was added, it was 'broken and unusable' and that the implementation was buggy. The browser vendor submitted that IndexedDB is a low-level API for client-side storage of significant amounts of structured data, including files/blobs. This API uses indices to enable high-performance searches of this data.[95]

(b) Apple's incomplete implementation of Fullscreen API. The browser vendor submitted this works for a video element but does not function properly for a [96] and other non-video elements. The browser vendor submitted that this 'restricts gaming and immersive media experiences significantly on iOS' as they cannot benefit from full screen display.[97]

3.56 Opera has submitted that its engineers consider the way the WebKit component is provided on the system to constitute a 'black box' and that it has limited documentation.[98]

3.57 Mozilla submitted that the iOS accessibility documentation is incomplete and many APIs that are needed for a web browser to support accessibility web standards are undocumented. Mozilla submitted that it is difficult to infer how to use them based on WebKit's open-source implementation.[99]

Access to browser analytics

3.58 A browser vendor submitted that Safari benefits from access to a full range of metrics relating to the performance of the browser, but that third-party browsers on iOS are limited in the performance metrics which WebKit allows them to access via APIs. This party stated that this means that third-party browsers on iOS are at a

[95] [✂] response to CMA's information request [✂].
[96] [✂] explained that the tag defines a division or a section of a web page. The tag is used as a container for web page elements and allows similar sets of content to be grouped together on a web page.
[97] [✂] response to CMA's information request [✂].
[98] Note of meeting with Opera [✂].
[99] Mozilla's response to CMA's information request [✂].

disadvantage, compared to Safari, when it comes to analysing and optimising performance because of very limited access to performance metrics.[100]

3.59 Apple has submitted that all browsers on iOS have 'equal access to APIs and analytics on browser performance through WebKit.' For instance, if a developer creates a webpage and it is loaded on WebKit, WebKit APIs can help the developer determine why its webpage is loading slowly, or if a developer wants to debug a particular website, it can use Web Inspector[101] to gain precise information about the website.[102]

Emerging thinking on access to functionality for browsers on iOS

3.60 The evidence above demonstrates that Apple's Safari makes use of several features and functionalities on iOS that third-party browsers do not have the same access to, and which Apple has acknowledged. These features and functionalities include user-facing features such as universal links (see paragraphs 3.19 to 3.20), the ability to download and upload data in the background (see paragraph 3.29), PWA installation (see paragraphs 3.17 to 3.18), and browser extensions (see paragraphs 3.15 to 3.16). Some of these features are likely to be particularly important to users. For example, browser extensions are widely used by users where they are available on desktop (see section 5) and PWA installation is important for making full use of web apps.

3.61 It is not always clear why it is possible for Safari to access these functionalities but not third-party browsers, or why third-party browsers could not be granted equal access with appropriate mitigations. In some cases, eg for PWA installation, Apple has ultimately extended access to third-party browsers after previously highlighting security risks in doing so. In other cases, eg the ability to store users passwords (paragraph 3.8), Apple stated that it has not seen sufficient demand for it to develop the feature. However, it is not clear how Apple assesses demand for features, and the evidence received from third parties indicates that there is demand from third party browser vendors.

3.62 There are also features such as access to the full screen API (see paragraph 3.26) and the ability to modify the JIT compiler (see paragraph 3.44) which were referred to by third-parties but for which we have not yet verified the status of access to the relevant functionality for third-party browsers on iOS.

3.63 For some types of functionality, Apple has stated that access to rival browsers already exists. This includes user-facing features such as Reader Mode (see paragraphs 3.13 to 3.14), security features such as SafeBrowsing (see paragraphs

[100] [✂] response to CMA's information request [✂].
[101] Web inspector helps developers inspect all the resources activity on web pages.
[102] Apple's response to CMA's information request [✂].

3.34 to 3.35) and process separation (see paragraphs 3.36 to 3.37), and access to analytics on browser performance (see paragraphs 3.58 to 3.59).

3.64 Apple has acknowledged that it does sometimes make functionalities available to Safari first, before extending availability to third party browsers. In some cases evidence shows that this has been for a significant period of time, for example around three years with Intelligent Tracking Protection, an important privacy feature (see paragraph 3.50). This may be particularly significant as it relates to newly developed or innovative features, which can be important for attracting users, meaning even a small time advantage for Safari could have an impact.

3.65 Finally, there are credible concerns that the visibility and documentation of APIs that can be accessed by third-party browsers on iOS by Apple is poor. This may increase the cost or difficulty of implementing a feature to third-party browsers or result in third-party browsers not being aware that a given functionality is available.

3.66 As discussed in 'WP1 - Nature of competition in the supply of mobile browsers and browser engines' and 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine', adding features and functionality to their browsers is an important way in which browser vendors can innovate and improve their browsers, and attract users.

3.67 The impact of lack of access to browser functionality should also be considered in the context of Safari's leading position as a browser on iOS (WP1 - Nature of competition in the supply of mobile browsers and browser engines), its WebKit restriction (WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine) and also certain choice architecture practices such as pre-installation and default status, which may also benefit more established browsers such as Safari (which will be discussed further in 'WP5 – The role of choice architecture in the supply of mobile browsers').

3.68 Any limitation on the ability of rival browsers to add features relative to Safari, whether through a complete lack of access, poor visibility and documentation, time delay or additional costs, may adversely impact third-party browsers' ability to attract users. The cumulative impact of missing several of these features may be significant for rival browsers. This will be particularly important for smaller browsers who need to provide users with strong reasons to switch away from more established browsers like Safari.

4. Access to browser functionalities on Android

4.1 This section considers evidence we have received to date on whether rival browsers on Android are able to access to browser functionality on Android and our preliminary assessment of the extent to which this may impact competition between browsers.

4.2 In the context of this analysis, Google has submitted that it has been unable to identify general categories of features that other browsers could not offer based on technical limitations enforced by the Android platform. It submitted that, generally, features Chrome is able to offer or chooses to offer on Android could likewise be implemented by another browser.[103]

4.3 Google submitted that there are limited exceptions (including WebAPKs as described below), but these features are [✂].[104]

4.4 Google submitted that it does not have specific policies regarding the availability of software (including APIs) to third-party browsers. It submitted that in general, when deciding whether to make software available to third parties, Google considers a number of factors, including whether access to software by third parties would be helpful for users and developers, present security or privacy risks, be technically feasible, and expand or diminish the potential for abusive behaviour against users or other services on a device.[105]

4.5 This section considers evidence from Google and third parties on specific functionalities within each of the categories outlined in paragraph 2.5; user-facing features, security features, privacy features, and information availability.

User-facing features

4.6 First, several third parties submitted that Chrome can use an API to create WebAPKs and that this is not available to third-party browsers. WebAPKs are a method of creating applications based on web pages (ie installing web apps). Third parties submitted that this restriction prevents third-party browsers from offering competitively relevant features around the installation of web apps. For example:

(a) Microsoft submitted that Google restricts access to the WebAPK system such that Chrome is uniquely able to offer PWA installation. Microsoft submitted that prevents third-party browsers from offering competing features on web app installation, and limits competition between PWAs and Android native

[103] Google's response to CMA's information request [✂].
[104] Google's response to CMA's information request [✂].
[105] Google's response to CMA's information request [✂].

apps.[106] Microsoft submitted that it has requested updates from contacts at Google at an engineer-to-engineer level but as of 24 May 2024, had not received any indication as to whether or when WebAPKs will be available on Android.[107]

(b) Yandex submitted that only Chrome has the ability to use a private API to create WebAPKs.[108]

(c) OWA submitted that on Android devices running the Google Play store, only Chrome has the ability to mint (create) WebAPKs (except on Samsung devices). OWA submitted this prevents competing browsers from producing viable web apps.[109]

4.7 Google submitted that the WebAPK minting service provides WebAPK minted apps with certain additional functionality.[110] Google submitted that it has not yet deployed a way for other browsers to use the WebAPK minting service. However, the Samsung browser on Samsung devices has access to equivalent functionality provided by the Galaxy Store. Google submitted that [✂].[111] [✂].[112]

4.8 Second, Mozilla submitted that Google Search on Chrome for Android was different from the search experience that was available to Firefox on Android. It submitted that identical terms searched in Firefox showed less information and receive a lower quality design in Firefox than in Chrome. It submitted that this was a significant web compatibility issue that consumers complained about and impacted Firefox usage. We understand that this issue has since been resolved and Google is offering a comparable search experience in Chrome and Firefox on Android. [✂][113]

4.9 Google submitted that the Google Search user experience may vary depending on the capabilities of the browser and that [✂].[114] Google stated that [✂]. Google stated that [✂]. However, Google stated that it is working with [✂].[115]

4.10 Third, Opera submitted that Chrome's one-click login experience to the Google account associated with the device provides Chrome with an advantage over rival browsers.[116]

---

[106] Microsoft's response to CMA's information request [✂].
[107] Microsoft's response to CMA's information request [✂].
[108] Yandex's response to CMA's information request [✂].
[109] OWA Bringing Competition to Walled Gardens, section 5.4.3, accessed by CMA 31 May 2024.
[110] Web apps installed by WebAPK can show up in the app launcher, be listed in Android settings, and process deep links to their content - Google's response CMA's information request [✂].
[111] Note of meeting with Google [✂].
[112] [✂] response to CMA's information request [✂].
[113] Mozilla's response to CMA's information request [✂].
[114] Google's response to CMA's information request [✂]; Google's response to CMA's information request [✂].
[115] Google's response to CMA's information request [✂].
[116] Opera's response to CMA's information request [✂].

4.11 Google has submitted that Chrome and other Google apps benefit from a one-click login experience to the Google account associated with the device, and that this creates an efficient user experience. Google submitted that third party browsers and other apps can also use 'Sign-in with Google' which enables the users to sign-in to the browser and sync user authentication across the developer's websites. The apps from the same developer can achieve the same single sign-on across their apps as available to Google's apps. It submitted that whether a browser supports 'Sign-in with Google' is up to the browser vendor.[117]

4.12 Fourth, Yandex submitted that Chrome uses different mechanisms for creating key processes and that due to the nature of the Android sandbox, leads to Chrome creating processes much faster and loading webpages faster than any other browser.[118]

4.13 Fifth, Brave submitted that Chromium recently added the Read Aloud feature (which converts web page text to audio) but that this is restricted to Chrome and cannot be used by Brave.[119]

Security features

4.14 Yandex submitted that Google controls the technology that allows users to authorise on websites via biometrics and can prevent other browsers from utilising it.[120]

4.15 Google submitted that it is open to other browsers using biometric authentication if they are added to 'trusted lists' maintained by each authenticator application. Google submitted that each authenticator application makes its own decision on what other application to trust for biometric authentication, and that Edge and Firefox are on the list maintained by its own authenticator in Google Play Services and that all legitimate browsers that have applied to be on the list have been accepted.[121]

Privacy features

4.16 We have not received any evidence from Google or third parties on privacy features that Chrome has access to, but that third-party browser vendors do not.

[117] Google's response to CMA's information request [✂].
[118] Yandex's response to CMA's information request [✂].
[119] Brave's response to CMA's information request [✂].
[120] Yandex's response to CMA's information request [✂].
[121] Google's response to CMA's information request [✂]; Google's response to CMA's information request [✂].

Information availability

4.17 We have not received any evidence from Google or third parties on information, data and metrics that are available to Chrome but not to other browsers on Android.

Emerging thinking on access to functionality for browsers on Android

4.18 Overall, the evidence available to date indicates that Google engages in self-preferencing less, in respect of access to functionalities on Android compared to Apple's approach on iOS. Lack of access to WebAPKs, which is essential for installing PWAs, is the main issue highlighted by third parties (see paragraphs 4.6 to 4.7). Whilst Google has acknowledged this restriction, its latest submission to the CMA indicates that it is working to resolve it. Google has in some cases provided justifications for lack of access to functionality being provided or noted that it is working towards providing equal access. For example on the Google Search experience on Firefox (see paragraphs 4.8 to 4.9), both Google and Mozilla stated that they were making progress in resolving the issue. Whilst this may resolve the issue, the delay in granting equal access (described as a significant period of time by Mozilla) may nonetheless have had an impact on Firefox's ability to attract users.

4.19 As discussed in 'WP1 – Nature of competition in the supply of mobile browsers and browser engines' and 'WP2 – The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine', adding features and functionality to their browsers is an important way in which browser vendors can innovate and improve their browsers, and attract users. Chrome has a leading position as a browser on Android (WP1 - Nature of competition in the supply of mobile browsers and browser engines), and other choice architecture practices exist in the browser market, such as pre-installation and default status, which may also benefit more established browsers such as Chrome (which will be discussed further in 'WP5 – The role of choice architecture in the supply of mobile browsers'). Any diminished ability to add features relative to Chrome, whether through a complete lack of access, or time delay or additional costs, may therefore adversely impact the ability of third-party browsers to attract users. This will be particularly important for smaller browsers who need to provide users with strong reasons to switch away from more established browsers like Chrome.

5. Limited support for browser extensions on iOS and Android

5.1 Browser extensions are additional software applications that can add functionality or features to a browser and enable users to customise their browsing experience. Popular extensions add functionality including ad blocking, productivity tools, grammar and spell-checking, amongst others.[122] Browser extensions are generally developed by third parties (ie not the browser vendor themselves).

5.2 Parties have submitted that browser extensions are a key part of the web ecosystem and that most popular browsers support them. For example, Mozilla stated that extensions add functionality to the browser providing increased utility, usability, and interoperability with applications installed on the operating system. Mozilla stated that for distribution, browsers have established extension catalogues that are available on the open web and curated by the browser vendors. As extensions have elevated privileges, developers submit them to be approved to ensure safety and compatibility. Browser vendors make their own decisions about the APIs available to extensions. Extensions for each browser are installed and managed within the browser resulting in a common user experience across platforms.[123]

5.3 On desktop, browser extensions are widely available, including on Chrome[124] and on Safari.[125] For example Chrome offers over 180,000 extensions and nearly half of Chrome desktop users use extensions.[126] For Firefox, around one third of users have installed an extension, and there were 110 million extension installs in 2021.[127]

5.4 The evidence received in this investigation to date shows that, for different reasons, support for browser extensions on iOS and Android is limited compared to the level of support seen for desktop browsers.

5.5 As described in paragraphs 3.15-3.16, although Safari has supported browser extensions on iOS since 2021, Apple has acknowledged that third-party browsers were not able to offer comparable support for browser extensions on iOS in March 2022, and third-party evidence indicates that third-party browsers are still limited in their ability to support browser extensions compared to Safari.

5.6 In addition, some third parties submitted that there is limited support for browser extensions on iOS more broadly, and that this reduces consumer choice, limits

---

[122] Gener8's response to the CMA's Issues Statement, [✂].
[123] Mozilla's response to CMA's information request [✂]; See also platform-tilt, accessed by CMA 18 June 2024.
[124] Chrome Web Store, accessed by CMA 31 May 2024.
[125] App Store Preview – Safari extensions, accessed by CMA 31 May 2024.
[126] Trustworthy Chrome Extensions, by default, accessed by CMA 31 May 2024.
[127] FIREFOX'S MOST POPULAR AND INNOVATIVE BROWSER EXTENSIONS OF 2021, accessed by CMA 31 May 2024.

differentiation between browsers, and holds back a potential entry route into browsers on iOS:

(a) The Coalition for Online Data Empowerment (CODE) submitted that web extensions are being deliberately held back by Apple. CODE submitted that although limited support for browser extensions has been added to Safari on iOS, it is not possible for rival browsers to ship their own extensions due to the WebKit restriction. CODE submitted that this restricts competition and differentiation between browsers and holds back a potential initial entry route into browsers.[128]

(b) Eyeo submitted that Apple technically allows some support for mobile extensions on Safari, however they are so complex to enable that only highly motivated users will succeed. Eyeo submitted that browser extensions are important to allow users to increase accessibility, boost productivity, safeguard privacy, or protect biodiversity, and that availability of extensions could address any existing market constraints.[129]

5.7 Although Safari does support extensions on iOS, evidence suggests this is more limited than on desktop. This limits users from accessing the same extension functionality on iOS that may be available to them on desktop, and prevents them from switching to an alternative browser that might offer greater choice of extensions.

5.8 On Android, although third-party browsers are able to (and in some cases do) support browser extensions[130], some third parties submitted that Chrome's lack of support for browser extensions reduces consumer choice, limits differentiation between browsers, and holds back a potential entry route into browsers on Android:

(a) The Coalition for Online Data Empowerment (CODE) submitted that web extensions are being deliberately held back by Google. CODE submitted that Google allows rival browsers to ship extensions on Android but does not support extensions in Chrome on Android (unlike on desktop). CODE submitted that this acts as a de facto ban on distributing extensions on Android. CODE submitted that this restricts competition and differentiation between browsers and holds back a potential initial entry route into browsers.[131]

(b) Eyeo submitted that Google does not support extensions in Chrome on Android in any way, significantly limiting the available tools for users to take

[128] CODE's response to the CMA's Issues Statement, [✂].
[129] Eyeo's response to the CMA's Issues Statement, [✂].
[130] Both Firefox and Edge support browser extensions on Android.
[131] CODE's, response to the Issues Statement, [✂].

control of their online experience. Eyeo submitted that Google was able to do this on Android given its market power. Eyeo submitted that browser extensions are important to allow users to increase accessibility, boost productivity, safeguard privacy, or protect biodiversity, and that availability of extensions could address any existing market constraints.[132]

(c) Gener8 submitted that it offers a browser extension on iOS but is unable to replicate this on Android 'due to Google's extension restriction.' It submitted that browser extensions are a low-cost entry route for browser vendors, enhance features and functionalities available to users, and are an alternative distribution channel to native apps. It submitted that Chrome's lack of support for browser extensions on Android is a result of its market power.[133]

5.9 Google submitted that it has considered [✂] but concluded that [✂]. Google submitted the following reasons for this:[134]

(a) [✂].

(b) [✂].

(c) [✂].

(d) [✂].

5.10 On Android, although there are no restrictions on the ability of browser vendors to support extensions, and some third-party browsers do support extensions, Chrome, which represents 77% of browser usage[135], does not support extensions. This is in contrast to the position on desktop where Chrome does offer full support for extensions. This limits users from accessing the same extension functionality on Chrome on Android that may be available to them on desktop. Although users are able to switch to another browser, certain choice architecture practices such as pre-installation and default status, may limit them from actually doing so (which will be discussed further in '**WP5** – The role of choice architecture in the supply of mobile browsers').

5.11 The limited support for browser extensions on iOS and Android has implications for browser users, who are less able to customise their browsing experience by using extensions to add features or functionality, relative to desktop. It also has implications for developers, who have less access to a potentially lower cost

[132] Eyeo's response to the Issues Statement, [✂].
[133] Gener8's supplemental response to the Issues Statement, [✂].
[134] Google's response to CMA's information request [✂].
[135] See 'WP1 - The nature of competition in the supply of mobile browsers and browser engines', paragraph 4.10 a.

distribution channel for their applications or content, and less access to a potential entry point into browsers.

5.12 The limited support for browser extensions on iOS and Android may be an outcome of the limited competition between browsers on iOS and between browsers on Android. This may mean that Apple and Google have less incentive to offer full support for this feature relative to desktop where there is more competition amongst browsers.