MOBILE BROWSERS AND CLOUD GAMING MARKET INVESTIGATION

WP2: The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine

27 June 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 **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 [✂].

Contents

Figures

Appendix

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.

1.2 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, this investigation is considering whether Apple is using its position in the supply of mobile operating systems to restrict the ability of competing browser vendors to develop competitive features for their browsers, in particular by requiring that all browsers on iOS use Apple's WebKit browser engine.[1]

1.4 This working paper provides an overview of our emerging thinking on the requirement for mobile browsers operating on iOS devices to use Apple's WebKit browser engine (the WebKit restriction).

1.5 We have developed the following three hypotheses in order to provide a framework for our investigation of this issue, against which we continue to consider available evidence:

(a) Apple bans alternative browser engines operating on iOS and iPadOS. This restriction forms part of Apple's App Store Review Guidelines. Specifically, Apple's clause 2.5.6 of its App Store Review Guidelines obliges third-party browsers to use WebKit: *'…Apps that browse the web must use the appropriate WebKit framework and WebKit Javascript…'*.[2] This excludes all competition between browser engines on iOS and may reduce Apple's incentive to improve WebKit.

(b) The Webkit restriction may have an impact both on browser vendors, by limiting their ability to innovate and to improve their browsers by adding competitive features for users, and also on web developers[3] by limiting their ability to develop features for their websites and web apps, given WebKit's

[1] Issues statement, paragraph 27(b).
[2] App Store Review Guidelines - Apple Developer, accessed by the CMA 19 April 2024.
[3] Web developers develop websites and web apps, which are accessed by users via different browsers and browser engines.

lack of functionality relative to other browser engines.[4] This in turn may lead to worse outcomes for consumers.

(c) Apple has submitted that the WebKit restriction improves the security, privacy, and performance of iOS devices and that this promotes competition at an ecosystem level between iOS and Android. To the extent that any benefits arise from the Webkit restriction, any such impact may not be sufficient to counteract the negative effects on competition in the browser market.

1.6 This paper includes a preliminary analysis of a range of evidence submitted by Apple, Google, and a range of third parties; qualitative web developer research conducted by Jigsaw Research, commissioned for this market investigation; evidence from the CMA's Mobile Ecosystems Market Study (MEMS) report; and an appendix comparing browsers and browser engines across several key metrics (Appendix A).

1.7 This paper should be read alongside 'WP1 - Nature of competition in the supply of mobile browsers and browser engines'.

1.8 This paper is structured as follows:

(a) Section 2 provides background on the WebKit restriction;

(b) Section 3 considers the implications of the WebKit restriction for browser vendors;

(c) Section 4 considers the implications of the WebKit restriction for web developers; and

(d) Section 5 considers Apple's justifications for the Webkit restriction.

[4] Web apps are applications built based on open standards and accessible through a browser on the open web. Differently from native apps, web apps are designed to be agnostic to the operating system in use.

2. Background on Apple's WebKit restriction on iOS and iPadOS

2.1 In this paper, references to Apple's 'WebKit restriction' are to Apple's requirement that all browsers on iOS and iPadOS use a mandated version of Apple's browser engine WebKit as underlying technology for the browser they offer on iOS. WebKit is also the browser engine used by Apple's browser Safari.

2.2 The WebKit restriction is specified in Apple's App Store Review Guidelines. Specifically, Apple's clause 2.5.6 from App Store Review Guidelines requires third-party browsers to use WebKit: *'…Apps that browse the web must use the appropriate WebKit framework and WebKit Javascript…'*.[5] This clause has been in place since the launch of Apple's App Store in 2008. Although Apple now permits iOS apps to use alternative browser engines in the EEA since March 2024, the WebKit restriction continues to apply in the UK and the rest of the world.[6]

2.3 WebKit is described as open-source, meaning that its source code can be taken and used by anyone to build software and it can benefit from contributions from a range of stakeholders who participate collectively in its development. Apple employs a significant portion of WebKit contributors and hosts and maintains WebKit's public-facing interfaces and documentation for the WebKit framework on Apple platforms. [✂].[7] However, it is Apple as WebKit's steward, which decides which changes to WebKit are incorporated into the head project. Additionally, as owner of the iOS and macOS operating systems (which are not open-source) Apple also retains control over the features and functionalities included in the versions of WebKit offered on macOS and iOS.[8]

2.4 Therefore, although browser vendors may submit changes to the WebKit open-source project, it is not fully 'open-source' in that Apple controls which changes are incorporated, and which changes are used for the iOS version. Browsers on iOS are restricted to using the same version of WebKit provided as a system framework[9] (WKWebView) and are therefore prevented from using modified versions or 'light forks'[10] of the browser engine, which would provide a mechanism for browser improvements and differentiation.[11]

---

[5] App Store Review Guidelines - Apple Developer, accessed by the CMA 19 April 2024.
[6] Using alternative browser engines in the European Union, accessed by the CMA 19 April 2024.
[7] Apple response to the CMA's information request [✂].
[8] Apple stated that different contributors are responsible for individual ports of WebKit. For example, Apple is responsible for the macOS and iOS ports of WebKit, Igalia is responsible for the GTK port, and Sony is responsible for the PlayStation port. Port owners have the final decision on the features and functionality that ship on their ports of WebKit. See Apple response to the CMA's information request [✂].
[9] A system framework is a pre-defined collection of code that is bundled together for reuse by other apps or frameworks. System frameworks are stored at the system-level instead of being embedded within a specific app.
[10] A fork is another version/copy of an open-source browser engine that has separated from the main branch of code. Light forks may retain most of the original code. Light forks may also be referred to as 'soft forks'.
[11] Several browsers use 'light forks' of Blink on Android and desktop.

Figure 2.1: WebKit restriction timeline

Source: CMA elaboration

2.5 Apple does not have an equivalent restriction for its desktop operating system MacOS, where rival browsers running on browser engines other than WebKit are allowed. MacOS differs from iOS in how native apps can be distributed with iOS only allowing native apps to be downloadable via Apple's official App Store, while MacOS also allows so-called 'sideloading', meaning the direct download of an app package from a website, without Apple's intermediation.

2.6 As noted in 'WP1 - Nature of competition in the supply of mobile browsers and browser engines',[12] browser engines transform web page source code into web pages (or web apps) that users can see and engage with. Therefore, they largely determine the performance and overall capability of a browser, as perceived by users. This includes the speed of loading of a webpage, the ability to render specific content (eg video format), the stability of navigation (ie absence of crashes and freezes while browsing), security protections (eg degree of vulnerability to malicious attacks while browsing) and some privacy features (eg option to block ads).[13, 14]

2.7 Until 2019, iOS was the operating system used on all Apple mobile devices, including its range of tablets, iPads. However since 2019 Apple rebranded the variant of iOS running on iPads as iPadOS. The WebKit restriction applies on both iOS and iPadOS, and evidence-gathering during this market investigation has defined 'mobile devices' as including both mobile phones and tablets, and defined 'iOS' as including both iOS and iPadOS. The evidence and views presented in this working paper therefore apply to both iOS and iPadOS.

[12] Paragraph 2.40.
[13] Privacy features can also be added at the browser level.
[14] In a public report titled 'Five Walled Gardens', Mozilla stated that browser engines 'can determine the speed, quality and features of a browser, as well as its security and privacy characteristics, including vulnerabilities'.

3. Implications of the WebKit restriction for browser vendors on iOS

3.1 The section below sets out evidence from Apple and rival browser vendors (including Google) in relation to the potential impact of the WebKit restriction on browser vendors. In particular (i) limiting the ability of browser vendors to innovate and improve their browser by adding competitive features for users; (ii) increased costs for browser vendors, which arise as a result of having to develop and maintain an additional version of their browser based on WebKit; (iii) delays to browser vendors being able to implement new innovative features or fixes[15] as a result of Apple's allegedly slow engagement.

Evidence from Apple

3.2 As detailed below, Apple submitted that the WebKit restriction does not restrict browser competition on iOS. It submitted that it is incentivised to allow for competition amongst browsers on iOS, and that browsers can compete effectively with the WebKit restriction. It stated that features are made available to rival browsers through WebKit, and that rival browsers can differentiate through building features on top of the browser engine.

3.3 Apple submitted that it allows other browsers to differentiate from Safari as WebKit permits them to build features on top of the engine while upholding privacy and security protections. Apple also stated that browser vendors are free to build features into their browsers that are not available in Safari within the constraints of the iOS ecosystem. Apple mentioned Brave shipping Web Authentication and Global Privacy Controls[16] on iOS before these were available via WebKit as an example of differentiation happening on top of the engine.[17]

3.4 Additionally, Apple submitted that browser vendors can build user interface (UI) features such as tab interfaces, bookmarks, history, downloads, and autofill of saved user information to differentiate themselves and gave the example of Chrome shipping Voice Search and Translation on iOS, which it built on top of WebKit.[18]

3.5 Apple submitted that it had added more than 300 features to WebKit from iOS 16.4 (released 28 March 2023) to 17.4 (released 5 March 2024) including web push, badging, and screen orientation (in beta). Apple referenced a public website that lists an improvement in 'browser engine score' for WebKit up 39 points, compared

[15] 'Fixes' include updates to resolve bugs or security issues.
[16] A feature which allows users to signal that they do not want to be tracked online.
[17] Apple, response to MEMS Interim Report, paragraphs 103-105.
[18] Apple, response to MEMS Interim Report, paragraph 105.

to 23 for Chromium and 24 for Gecko.[19] We note that despite this reported improvement WebKit has the lowest score of the three major browser engines (see the Appendix A).

3.6 Apple's justifications for the WebKit restriction are discussed in further detail in section 5 below.

Evidence from browser vendors

3.7 This section sets out evidence submitted to the CMA from browser vendors regarding the impact of the WebKit restriction, covering: (i) evidence from browser vendors on their strategies to differentiate their browsers from competitors on iOS and how the WebKit restriction affects this; (ii) additional costs that may arise for browser vendors as a result of having to maintain a different version of their browser using WebKit on iOS; (iii) delays to the implementation of features and fixes as a result of Apple's alleged slow engagement.

Limitations on browser improvements and innovations

3.8 This section covers evidence from browser vendors on the impact of the WebKit restriction on their ability to differentiate by innovating and improving their browsers on iOS. It first considers general evidence on how browser vendors are less able to innovate and improve their browsers on iOS relative to Android, before considering evidence on specific features or improvements that they are unable to implement, grouped into four categories, namely security, privacy, performance, and other features or innovations.

3.9 As described in 'WP1 - Nature of competition in the supply of mobile browsers and browser engines',[20] browser vendors submitted that their products compete by offering differentiated features to browser users, with smaller browser vendors in particular focusing on specific product features that may attract users to download and use a different browser than the default browsers offered on iOS and Android.

3.10 Browser vendors which offer browsers on both iOS and Android (accounting for the vast majority of the UK browser market) stated that they are less able to innovate and improve their products on iOS than on Android. In particular, they stated that the WebKit restriction limits their ability to add new features on iOS relative to Android and that this results in differences between their iOS and Android offerings (examples of specific features are described in the following sections):[21]

[19] Note of meeting with Apple [✂]; Apple response to the CMA's information request [✂].
[20] Paragraph 2.31.
[21] Responses to the CMA's information requests [✂].

3.11 We note that browser vendors, particularly the smaller ones, are limited in their ability to differentiate to some extent by the need to ensure that their browsers are compatible with websites and web apps. As described in paragraphs 4.20 and 4.21, most web developers test for compatibility against the major browsers, namely Safari and Chrome. Most other browsers use Blink which reduces compatibility issues, however some submitted that they may still face compatibility issues if they differentiate too far from Safari and Chrome, and have to test for this as they develop new features.[22] However, the evidence below demonstrates that the WebKit restriction further restricts differentiation for these browser vendors, and there are several specific examples of features that it prevents them from implementing.

Security

3.12 Several browser vendors described security-related features that they were unable to implement on iOS to increase the level of protection on their browsers. They also described how the WebKit restriction means that only Apple can implement security fixes for browsers on iOS, which can prevent other browser vendors from implementing fixes that are important to them quickly:[23]

(a) Microsoft submitted that it cannot support innovations it has introduced in the versions of Edge for other operating systems in the iOS version of Edge, such as Enhanced Security Mode (which applies stricter security settings on unfamiliar sites),[24] and that it is not able to implement more secure and private networking on iOS as well as security features such as Content Security Policy (CSP) Violation Reports and Trusted Types.[25]

(b) One browser vendor [✂] stated that browsers [✂] on iOS do not include site isolation, the development of which required the browser vendor [✂]. This browser vendor [✂] explained that without site isolation a single browser bug could allow multiple sites operating in the same tab (eg when an advertisement, video, payment widget etc. is embedded within a web page) to attack each other but with site isolation a single browser bug is insufficient as the operating system provides an additional layer of protection. This browser vendor [✂] cited a study according to which the enforcement of site isolation is 'one of the most important security features that a browser should have today' and noted that it has been able to implement it on all platforms except iOS, due to the WebKit restriction and that WebKit does not have a site isolation equivalent.[26]

---

[22] Responses to the CMA's information requests [✂].
[23] Responses to the CMA's information requests [✂].
[24] Microsoft response to the CMA's information request [✂].
[25] Microsoft response to the CMA's information request [✂].
[26] [✂]; X41 – Browser Security White Paper, page 9, accessed by the CMA 18 June 2024.

3.13 Overall, evidence we have seen so far suggests that the WebKit restriction limits the ability of browser vendors to improve their browsers by adding additional security improvements or features on iOS, compared to other platforms, including Android. It may therefore decrease competition between browsers on security features on iOS. In this context, Apple submits that its control over browser engines on iOS contributes to high levels of security overall, particularly as it controls security updates in a centralised way and therefore ensures that every browser on the platform has an up-to-date and secure browser engine. This is considered further in section 5.

Privacy

3.14 Several browser vendors highlighted privacy features or improvements that they were unable to implement, or were not able to implement as effectively, on iOS due to the WebKit restriction. These include various features to prevent user tracking[27] or to block ads:[28, 29]

(a) Brave submitted that it is limited in the privacy protections it can implement on the iOS version of its browser compared to its browsers based on Chromium and this depends on its inability to add or modify APIs in WKWebView, as well as other restrictions associated with WebKit such as its inability to change the rendering logic in WKWebView. Brave said that while Apple builds in features with a competitive level of privacy on WKWebView and possibly higher than can be found the default levels in Chromium, these are not at the same level as Brave can offer on Android. As a result, Brave stated that on iOS it cannot achieve the level of privacy Brave offers on Android.[30] Brave provided several examples of privacy-enhancing features it cannot provide on iOS or cannot replicate to the same standard as on Android, including:

(i) WKWebView restricts Brave to using Apple's Content Blocker system, which is much more constrained and limited than the far more featured Adblock-rust library Brave uses on desktop and Android.[31]

(ii) Fingerprint[32] randomisation (or 'farbling'), which is a technique Brave uses on its Android and desktop browser to protect users from being

[27] Preventing websites or web apps from gathering data about user activity on the web.
[28] Preventing websites or web apps from displaying advertisements to users.
[29] Responses to the CMA's information requests [✂].
[30] Note of meeting with Brave [✂].
[31] Brave response to the CMA's information request [✂]; Brave stated that Adblock-rust library allows Brave to apply far more (and more narrowly tailored) rules, replace privacy-harming requests with alternatives, and considering DNS information (eg CNAMEs) when making blocking decisions.
[32] Browser fingerprinting consists of using semi-identifying characteristics to identify users.

identified via certain characteristics on the web but that it can only implement in a weaker form on iOS.[33]

(iii) HTTP header modifications[34] which Brave uses on Android and desktop to fully implement the Global Privacy Control standard and reduce the possibility of fingerprinting.[35]

(iv) Query parameter stripping, which Brave uses on Android and desktop to automatically remove some trackers and identifiers from URLs[36] but which it cannot replicate fully on iOS. Even when it can implement it this comes with risks of sites breaking, which results in Brave doing it less on iOS.[37]

(v) Storage management, via which Brave partitions storage by site, allowing users to have different cookie settings for each, which is in some cases not possible at all on iOS, or not possible without prohibitive performance cost or compatibility risks.[38]

(vi) ServiceWorker customisation which allows Brave to ensure that sites do not circumvent its privacy protections on Android and desktop and which it cannot replicate via WKWebView.[39]

(b) Vivaldi explained that it uses Chromium on Windows, macOS, Linux, and Android where it can modify it and add functionality, particularly in relation to privacy, but that this is not possible on iOS for which it would have to rewrite code even when it has already done the work on the other platforms it supports, including macOS. As an example of the functionality Vivaldi adds to Chromium, Vivaldi noted that it provides its own tools for ad tracker blocking, which it had to build going down to the engine level. Vivaldi stated that on iOS it could not do this as WebKit has its own tracker blocker built in (ITP) which Vivaldi considers 'significantly flawed'.[40]

3.15 This evidence suggests that the WebKit restriction limits the ability of browsers to improve their browsers by adding privacy features on iOS, by either preventing the implementation of features that are available on other platforms, or making implementation more difficult. These includes features to prevent user tracking or

---

[33] Brave response to the CMA's information request [✂].
[34] An HTTP header is a field of an HTTP request or response in the exchange between a browser and a server that passes additional context and metadata. For example, a request message from a server to a browser can use headers to indicate its preferred media formats.
[35] Brave response to the CMA's information request [✂].
[36] Brave said it is not able to change the URL in an existing request in WKWebView but can only cancel a request and issue a new request with the modified URL which breaks some sites and this results in Brave being far more cautious about which parameters we remove in iOS.
[38] Brave response to the CMA's information request [✂].
[38] Brave response to the CMA's information request [✂].
[39] Brave response to the CMA's information request [✂].
[40] For example, ITP only blocks trackers after the initial tracker has been loaded once and cannot detect it until it has been loaded enough times, whereas Vivaldi blocks trackers from the outset; Note of meeting with Vivaldi [✂].

to block ads. It may therefore decrease competition between browsers on privacy features on iOS. Nevertheless, there may be differing views on what is meant by privacy for users in the context of browsers, and this is considered further, alongside Apple's submissions on the benefits of the WebKit restriction to privacy on iOS, in section 5.

Performance

3.16 Some browser vendors submitted that the performance of WebKit is inferior to other browser engines, and that in turn, the WebKit restriction and a lack of access to necessary APIs limits performance improvements they can make to their browsers on iOS:[41]

(a) One browser vendor [✂] stated that speed and performance is largely determined by the browser engine and thus improvements on iOS are limited to components outside of WKWebView. It explained that performance is a low-level consideration (meaning it is determined at the browser engine level) therefore not having access to the core low-level rendering engine limits potential improvements, for example [✂].[42]

(b) One browser vendor [✂] also submitted that it is required to implement features in a way that impacts performance due to the WebKit restriction. [✂].[43]

3.17 This evidence suggests that the WebKit restriction limits the ability of browser vendors to improve the performance of their browsers on iOS, and may therefore decrease the ability of browsers to compete on performance features on iOS.

3.18 In this context, Apple argues that the WebKit restriction leads to higher performance of browsers on iOS overall – these arguments are considered in section 5. Apple has submitted 2023 and 2024 results of performance tests for Safari on iOS compared with Chrome and Firefox on an Android mobile device which indicate that Safari outperformed Android browsers.[44]

3.19 Notwithstanding evidence that WebKit on iOS may, on the tests carried out, outperform other browser engines on other operating systems on metrics such as loading speed, it is still the case that the WebKit restriction prevents browser vendors from adding their own performance improvements and being able to compete on such features.

---

[41] Responses to the CMA's information requests [✂].
[42] [✂] response to the CMA's information request [✂].
[43] [✂] submission to the CMA [✂].
[44] Apple response to the CMA's information request [✂]; Apple response to the CMA's information request [✂].

Other features or innovations

3.20 Several browser vendors highlighted particular browser features or innovations that they could not implement on iOS due to the WebKit restriction. Two particular areas that have been highlighted are the lack of support available through WebKit for features which are important (i) to web apps, which are an increasingly important way for web developers to create content for mobile users, and which many browser vendors would like to support, and (ii) for accessibility ie ensuring that websites and web apps are accessible to all users, for example, individuals with vision, mobility, hearing, or cognitive issues.[45]

(a) Microsoft submitted that web apps built using WebKit do not have access to certain key APIs necessary to offer full functionality. It stated that WebKit does not support certain WebRTC APIs necessary to create streaming functionality and has not implemented features such as Web Codecs,[46] Web Transport,[47] Web Share Target amongst others.[48] It also stated that it was not able to offer features such as Image SuperResolution on iOS[49] which are supported by every other operating system. These limitations mean that Microsoft cannot create web apps that are as capable as native apps on iOS.[50]

(b) Mozilla submitted that it was unable to implement certain accessibility APIs on iOS. This means information to allow developers to make their apps accessible to all users. For example, people with vision, mobility, hearing or cognitive issues. It stated that this meant 'Firefox becomes less attractive for users with accessibility needs'.[51]

3.21 As described in paragraph 3.5 Apple has recently added some of these features to WebKit, notably push notifications and full screen API functionality for web apps. However, some of the features highlighted by rival browser vendors such as WebTransport are not currently available from WebKit.[52] Further, the delay in implementation of these features in WebKit relative to other browser engines means that iOS users have not benefitted from functionality of web apps for some time, and is an indication of Apple being slower to implement new features in WebKit, which could continue in the future.

3.22 As noted in paragraph 4.35, there may be security risks to implementing additional features in a browser engine. There may therefore be security benefits to limiting

[45] Responses to the CMA's information requests [✂].
[46] Which enables better performance for video conferencing applications.
[47] Which enables faster page loading on poor and unreliable networks.
[48] Microsoft response to the CMA's information request [✂].
[49] A feature which automatically enhances images.
[50] Microsoft response to the CMA's information request [✂].
[51] Mozilla response to the CMA's information request [✂].
[52] Mozilla Developer Network - Web Transport API, accessed by the CMA 17 June 2024.

the implementation of new features in WebKit. Arguments around the security benefits of the WebKit restriction are discussed in section 5.

3.23 The above evidence is supported by our analysis of different measures of feature support and compatibility for the different browsers and browser engines in Appendix A. This indicates that WebKit has lagged behind other browser engines on some measures, but has been closing the gap recently.

3.24 The support available through WebKit for web apps is also considered in section 4. This indicates that lack of support for web app features on WebKit has had an impact on web developers, particularly those with specific business models reliant on exploiting web apps, for example Microsoft and its cloud gaming service.

Additional costs

3.25 Browser vendors submitted that they incur additional costs to maintain versions of their browsers in two engines, which would not be necessary if they were able to use a browser engine of their choice on iOS.[53] They also stated that some features need to be developed in a different way on WebKit, incurring additional costs:[54]

3.26 Several browser vendors stated that the WebKit restriction has delayed or prevented their entry in browsers on iOS. One vendor stated that this has also restricted its entry on Android as it does not make sense to ship a product to only Android users:[55]

(a) Mozilla submitted that it delayed listing Firefox on the iOS App Store because Firefox is built using Mozilla's Gecko browser engine and listing on iOS required redeveloping Firefox on Apple's WebKit engine.[56]

(b) Gener8 submitted that it has a suite of products, including a desktop browser built using Blink. It stated that due to the WebKit restriction, it is not technically possible for it to ship its browser for iOS devices, and it does not make sense for it to 'ship a product that is only accessible by up to half the market', therefore it does not currently offer a mobile browser on Android either.[57]

3.27 Other browser vendors however did not cite the WebKit restriction as a reason for not developing a browser for iOS.[58]

[53] Although there would still be additional costs to supporting a browser an iOS, the cost is greater given the requirement to use a different browser engine compared to other platforms.
[54] Responses to the CMA's information requests [✂].
[55] Responses to the CMA's information requests [✂].
[56] Mozilla response to the CMA's information request [✂].
[57] Gener8, Supplemental Response to Issues Statement, page 1.
[58] Responses to the CMA's information requests [✂].

3.28 Overall, browser vendors have raised concerns that the WebKit restriction increases costs, as it requires them to develop and maintain an additional version of their browser, based on WebKit, to serve iOS users. There are also additional costs associated with finding ways to implement features within WebKit, given browser vendors are not able to alter the browser engine source code. According to the evidence from browser vendors, these increased costs have in some cases deterred or delayed entry of rival browsers on iOS.

Delays implementing features and fixes

3.29 Browser engines largely determine the performance and overall capability of a browser. As WebKit is the only permitted browser engine on iOS, browser vendors must engage with Apple (which controls the version of WebKit available on iOS) regarding issues with WebKit or requests for new features to be implemented in WebKit. Evidence from most browser vendors indicates that Apple is slow to engage and often does not respond to such requests, leading to delays in the implementation of new features or fixes. They submitted that such delays can deter investment in browsers and prevent browser vendors from developing innovations on iOS as they do not have certainty that features will be implemented:[59]

(a) One browser vendor [✂] submitted that it regularly notifies Apple of bugs arising out of the WebKit restriction, including through its WebKit bug tracker, Apple feedback bugs, email, and a WebKit Slack channel. It stated that when it [✂] files feedback on bugs with Apple's internal system, Apple typically does not act on these reports and the file remains open indefinitely. This browser vendor [✂] provided examples of requests submitted to Apple in relation to implementing features or APIs on iOS and WebKit which did not receive a response.[60] These included:

(i) [✂].

(b) [✂].[61]

3.30 One browser vendor [✂] however was more positive about its engagement with Apple. It stated that it generally faces fewer issues on iOS relative to Android as Apple has a more established process to engage with it and to escalate bugs.[62]

[59] Responses to the CMA's information requests [✂].
[60] [✂] response to the CMA's information request [✂].
[61] [✂] response to the CMA's information request [✂].
[62] Note of meeting with [✂].

Summary of emerging thinking on implications of the WebKit restriction for browser vendors

3.31 The evidence above indicates that the WebKit restriction limits the ability of rival browsers to innovate and improve their browsers on iOS. It also increases their costs as a result of having to maintain a separate version of their browser on WebKit, and delays or deters the implementation of new features and fixes.

3.32 This has a direct impact on consumers as features and improvements that would otherwise be available on iOS, and that often are available on Android, cannot be implemented due to the WebKit restriction. Furthermore, the inability to differentiate by innovating and improving browsers may weaken competition between browsers on iOS, leading to worse outcomes for consumers. These impacts mean that the restriction could lead to worse levels of security, privacy, performance, and feature support for browsers on iOS.

3.33 As discussed below Apple seeks to justify the WebKit restriction on the basis that it improves the level of security, privacy and performance of iOS mobile browsers and devices, and that this is important to ecosystem competition between iOS and Android. These submissions are considered further in Section 5 below.

4. Implications of the WebKit restriction for web developers

4.1 The WebKit restriction has implications for web developers as it means a substantial proportion of mobile users ie all iOS users, will be using a WebKit-based browser. This may have a negative impact on web developers if WebKit, as a result of not facing competition from alternative browser engines on iOS, performs worse than other browser engines on parameters such as (i) web compatibility and feature support,[63] (ii) support for web apps, or (iii) the extent of bugs and security issues. These issues are explored further below.

4.2 By way of context, web compatibility is a key issue for web developers, as they develop websites and web apps, which are accessed by users via different browsers and browser engines. Any websites or web apps they develop must therefore be compatible with whichever browser and browser engine their customers are using.

4.3 A lack of support for features may limit the quality of websites and web apps that web developers can create, and in turn may impact users by reducing availability of features or the quality of websites and web apps they use.

4.4 Also by way of context, web developers have raised that, as a result of Apple's control of the iOS operating system, it is able to hold back the development of web apps as a method of users accessing content, meaning that developers are less likely to focus their efforts on developing web apps compared to native apps through the Apple App Store. It is possible that the revenue Apple receives through commissions on in-app sales made through apps listed on the App Store may create an incentive for it to hold back the development of web apps.

Evidence from Apple

4.5 As detailed below, Apple submitted that it is incentivised to provide features and functionality to web developers to ensure the attractiveness of iOS devices. It submitted that it has added functionality to WebKit and that any delays to implementing functionality can be a result of prioritisation, or security, privacy, or performance considerations.

Compatibility and feature support

4.6 First, on web compatibility and feature support, Apple submitted that it has already implemented or is in the process of implementing many features and functionalities such as Screen Orientation functionality, TouchEvents,[64] WebGL

[63] Web compatibility refers to the extent to which consumers can visit any webpage and use any web application from any operating system and any browser.
[64] To provide quality support for touch-based user interfaces.

2.0,[65] File and Directory Entries API,[66] and Service Workers.[67] It further stated that browser quality should not be judged by the length of a browsers list of features, or the speed with which they are introduced, and that Apple implements new features in a way that allows device security, privacy, and performance to be preserved.[68]

4.7 When asked about its decision not to implement, or to delay the implementation of, a range of features in WebKit which appeared earlier in other browser engines (and in some cases have been part of web standards for several years) Apple disagreed with the characterisation of unavailability as a delay, noting that product development is characterised by decisions on where and how to allocate resources based on relative product priorities. Apple noted that this is especially true of browser feature development that prioritises expedience over quality or that involves substantial compromises on performance, privacy, or security. Apple stated that, to the extent that certain features are not available at a given time, this can be the result of differences in product development priorities, time and resource constraints, lack of third-party demand for features or technical barriers with making features widely available without compromising security, performance, or privacy.[69]

4.8 As described above, Apple submitted that it had added more than 300 features to WebKit from iOS 16.4 (released 28 March 2023) to 17.4 (released 5 March 2024) including web push, badging, and screen orientation (in beta). Apple referenced a public website that lists an improvement in 'browser engine score' for WebKit up 39 points, compared to 23 for Chromium and 24 for Gecko.[70] We note that despite this improvement WebKit has the lowest score of the three major browser engines (see Appendix A).

4.9 Several Apple internal documents indicate that [✂].[71]

4.10 Other Apple internal documents are informative of its approach to incorporating new features for use in web development. As described in section 2, WebKit's lack of support for features has an impact on browser vendors who might want to support these features in their browser. It also impacts web developers who would want to implement these features in their websites or web apps. Internal documents suggest that Apple takes account of standards bodies, public feedback from web developers, and interoperability measures, as well as its own strategy, when deciding on whether to implement features.[72]

---

[65] Improves the visual fidelity of 3D applications on the web, including games.
[66] Simulates a local file system that web apps can navigate within and access files in.
[67] API which enables modern, reliable offline web experiences and progressive web apps.
[68] Apple, response to MEMS Interim Report, paragraph 106-107.
[69] Apple response to the CMA's information request [✂].
[70] Note of meeting with Apple [✂]; Apple response to the CMA's information request [✂].
[71] Apple internal documents [✂].
[72] Apple internal documents [✂].

Support for web apps

4.11 Apple submitted that over the last few years it has added new functionality to WebKit enabling greater features and functionality for web apps but had to balance this with assurances that any new functionality provided to web apps would not compromise user privacy and data security.[73] In relation to this, Apple submitted that on iOS 11 it introduced support for key web app and progressive web app (PWA)[74] technologies, such as Service Workers, Web Authentication API,[75] and WebRTC.[76] Apple also stated that it continues to work on introducing PWA features to WebKit, such as prompts and web app manifest icon support.[77]

4.12 Apple submitted that it has always supported web apps and that it created the concept of web apps in 2007 and originally intended for third parties to develop web apps for iPhone rather than native apps. However, the development was slow and overtaken by the development of native apps when Apple launched the App Store in response to developer demand. Despite this, Apple submitted in 2021 that web-apps have made a resurgence with HTML5.[78]

Bugs and security issues

4.13 We have not received specific submissions from Apple on the implications for developers of any bugs or security issues in Safari and WebKit. However, as described in section 5 Apple has submitted that WebKit has privacy and security features built in and is optimised for high performance on iOS devices.

Evidence from web developers

4.14 This section summarises evidence from web developers on the implications of the WebKit restriction for web development on the three key issues of (i) web compatibility and feature support; (ii) support for web apps, and (iii) the extent of bugs and security issues.

4.15 The section includes the views of large web developers gathered through RFI responses and calls. It also includes submissions from the Open Web Advocacy ('OWA') group which campaigns for a more open web and is made up of developers from several companies, and submissions from several individual developers who responded to the CMA's MEMS Interim Report. Finally, it includes evidence from the qualitative web developer research commissioned by the CMA

[73] Apple response to the CMA's information request [✂].
[74] PWAs are particular versions of web apps which aim to create an experience even more comparable to a native app compared to a normal web app.
[75] Enables strong authentication with public key cryptography, enabling password free authentication and secure multi-factor authentication (MFA).
[76] Real time network protocol for enabling videoconferencing, desktop sharing, and game streaming applications.
[77] Apple, response to MEMS Interim Report, paragraphs 98-99 and 108-109; Apple, response to consultation on market investigation reference proposal, paragraph 3;
[78] Apple response to the CMA's information request [✂].

and conducted by Jigsaw Research, which consisted of detailed interviews with individual web developers. The research conducted by Jigsaw Research was commissioned to gather evidence from a different set of web developers than the CMA might normally receive evidence from through RFI response or submissions ie those who may be less engaged with issues in the industry and therefore less likely to proactively submit evidence. It was therefore anticipated that the views expressed by respondents to this research might differ from those of other web developers who have previously submitted evidence to the CMA during the CMA's MEMS or during this market investigation.

4.16 We note that web developers are a diverse group, and often have differing views about the impact and importance of issues. Where possible we have considered and highlighted where certain types of developers eg larger developers might have a certain viewpoint or be impacted in a certain way.

Compatibility and feature support

4.17 This section first considers how web developers manage differing levels of support for features that they would like to implement, and therefore ensure that their websites and apps can be accessed by users. It then considers evidence on whether WebKit is behind other browser engines in terms of feature support, and therefore whether the WebKit restriction increases the cost of ensuring compatibility or prevents developers implementing features because they are not supported by WebKit.

4.18 As detailed below, the evidence shows that ensuring compatibility with the main browsers (and therefore browser engines) is an important consideration for web developers. There is some evidence that Apple is slower to introduce new features on WebKit, relative to other browser engines, making it more difficult for developers to ensure compatibility, and increasing their costs. However, many developers did not consider that the WebKit restriction was an issue for compatibility.

4.19 OWA submitted that, given that browsers have different codebases, the features they support differ, meaning web content and features may be rendered differently or in some cases may not be compatible with certain browsers or browser engines.[79]

4.20 As a result, several developers submitted that they ensure that their websites are rendered correctly across the most used browsers and that their teams work to ensure that features are supported consistently by those. Web developers

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

submitted that they optimise their websites and web apps in different ways to achieve compatibility.[80]

4.21 The analysis from the qualitative web developer research conducted by Jigsaw Research suggests that for the developers interviewed:

(a) Developers initially develop in a browser of their choice, often in Chrome given its market share, but also in other browsers such as Safari or Firefox depending on developer preferences or client needs. They then check compatibility with the other main browsers namely Chrome, Safari, and sometimes Firefox, Brave or Edge.[81]

(b) Most web developers see compatibility as a small part of the job taking up an estimated 5-10% of their time however some estimated the time taken was outside this range, with a few saying it took very little or even a negligible amount, and a few others that it took 20-25% of their time.[82] The time spent on compatibility has declined over the last five to ten years due to the use of frameworks, increased standardisation amongst browsers, and the decline of Internet Explorer. The level of detail of checks may depend on client requirements or developer resource available.[83] Browser compatibility was therefore not cited as a major issue and there were few mentions of browser engines differences.[84]

(c) The web development environment was seen to be constantly changing requiring them to learn and adapt, consider new ways to develop sites, and fix issues with existing sites.[85] There was a noted trend towards 'mobile-first' development, greater use of web apps and PWAs, and uptake of AI.[86]

4.22 There is some evidence (gathered from RFI responses and calls) that the WebKit restriction may increase the cost to developers of ensuring compatibility. There is evidence that it also holds back web development as WebKit supports fewer features, limiting the features that web developers can implement, and meaning that websites and web apps are therefore less advanced. Some respondents referred to sources in the public domain which they stated show that WebKit is lagging behind other browser engines in terms of support for features that could be

[80] Responses to the CMA's information requests [✂]
[81] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p22, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.
[82] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p42, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.
[83] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p43, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.
[84] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p34, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.
[85] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p25, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.
[86] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p23, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.

used by web developers. These and other data sources are assessed in more detail in Appendix A:

(a) One party [✂] submitted that Apple either delays the introduction of technical changes to WebKit that facilitate these improved experiences or chooses not to implement them at all, thereby perpetuating a bifurcation of the web for developers. This party [✂] referred to a public benchmark measuring the performance of different browser engines on compatibility, which it submitted demonstrates the poor compatibility of Safari.[87] Apple however submitted that this public benchmark is inconsistent in terms of test quality, is vulnerable to gaming by browser vendors, and focuses on compatibility with web specifications, not on other attributes, including quality, performance, stability, and privacy. Apple also stated that it focuses on 'metrics based on the total number of tests run, rather than the importance of those tests'.[88]

(b) OWA submitted that compatibility issues may be exacerbated by WebKit lagging other browser engines on support for features. It stated that 'it is well known in the web-development industry that Safari is far behind on critical web-features'. It also highlighted two public benchmarks for compatibility which it claimed show that 'Safari is objectively lagging the competition'.[89]

4.23 Some web developers submitted that they face costs from ensuring their websites are compatible with WebKit given its limitations with respect to functionality:[90]

(a) One party [✂] submitted that optimising websites and web apps for WebKit is costly because WebKit does not support the latest browser engine features and technology. This party [✂] stated that due to cost considerations it usually builds its websites and web apps to the 'lowest common denominator', WebKit. This party [✂] stated that when it decides to add a feature to its websites and web apps which Android and desktop browsers support, but iOS browsers do not, this requires building separate versions of the site, which entails higher costs.[91] This party [✂] estimated that to ensure compatibility with Safari, it has to multiply its efforts by 1.5 times.[92]

(b) Microsoft submitted that developing a single codebase entails lower development and maintenance costs for its websites and web apps.[93] As a result, Microsoft stated that missing WebKit functionality means web developers face a difficult choice between providing a reduced set of features

[87] [✂] response to the CMA's information request [✂]; Web platform tests dashboard, accessed by the CMA 18 June 2024.
[88] Apple response to the CMA's information request [✂].
[89] OWA, Bringing Competition to Walled Gardens, section 5.4; Web platform tests dashboard, accessed by the CMA 18 June 2024; Progressive Web App Feature Detector, accessed by the CMA 18 June 2024.
[90] Responses to the CMA's information requests [✂].
[91] [✂] response to the CMA's information request [✂].
[92] [✂] response to the CMA's information request [✂].
[93] Microsoft response to the CMA's information request [✂].

(either only to users of certain browsers or to all users), recommending users switch browsers, or recommending users switch to native apps.[94]

4.24 However, many web developers submitted that the cost of ensuring that their websites are compatible with different browsers is limited and not burdensome, and that it largely stems from the presence of different browsers and browser engines as opposed to issues affecting a specific browser engine:[95]

(a) The Guardian submitted that compatibility costs are ad hoc and the presence of certain software to run code through ensures that its websites are accessible via most browsers and therefore ensures that developers working at the Guardian do not need to worry about compatibility on a day-to-day basis.[96]

(b) DMG Media submitted that the majority of costs for ensuring compatibility derive from testing functionality, and that it did not identify any specific limitations of browser engines which increase testing time.[97]

4.25 The qualitative web developer research conducted by Jigsaw Research found that respondents provided few explicit mentions of the WebKit restriction leading to limitations. Only a minority of developers in the research said that WebKit had some specific limitations.[98] Some noted that Apple is slower to take up new features in WebKit relative to other browser engines and uses different formats (eg for video) and that this may mean choosing not to use a feature or using a workaround.[99]

4.26 Overall (considering evidence gathered through RFI responses and calls, and the qualitative web developer research conducted by Jigsaw Research), there is mixed evidence regarding the extent to which the WebKit restriction has an impact on web compatibility and features support. Whilst certain developers felt that WebKit is lagging behind other browser engines on support for new features and therefore making compatibility more difficult, others considered that compatibility issues could not be attributed to a specific browser or browser engine. For some developers the lack of feature support in WebKit has added to their costs or limited their ability to include new innovative features in their websites or web apps, therefore reducing the features available to users.

---

[94] Microsoft response to the CMA's information request [✂].
[95] Responses to the CMA's information requests [✂].
[96] GMG response to the CMA's information request [✂]; note of meeting with GMG [✂].
[97] DMG Media response to the CMA's information request [✂].
[98] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p34, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.
[99] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p35, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.

Support for web apps

4.27 This section considers evidence on WebKit's support for web apps (or PWAs). It first considers whether WebKit has lagged in support for web apps relative to other browser engines, and whether this has held back development of web apps more broadly. It then considers evidence on the impact of this on web developers in terms of increased costs or inability to provide web apps with certain features.

4.28 Several developers submitted that the requirement for mobile browsers to use WebKit on iOS has limited or significantly delayed the capabilities of web apps and PWAs across platforms due to Apple's slow adoption of features important for web apps in WebKit:[100]

(a) In an article titled 'Progress Delayed Is Progress Denied', Alex Russell (Microsoft) criticises Apple's consistent delays in the delivery of important features for web apps and determines that these can never be 'a credible alternative to its proprietary tools and App Store.'[101]

(b) One party [✂] provided a list of features and functionalities showing that Apple has been slower at implementing these on WebKit compared to implementation of the same features in Blink, some of which are important for PWAs. This includes features and functionalities which Apple implemented on WebKit years after Blink, such as push notifications, Service Workers[102] and WebRTC;[103] and features which Apple has to date not committed to supporting on WebKit, such as A2HS Prompt,[104] WebGPU, and WebXR.[105] This party [✂] stated that this slows the adoption of PWAs across platforms as the functionality of PWAs is reduced 'to the lowest common denominator'.[106]

(c) OWA listed APIs important for PWAs and for gaming on the web, submitting that many of these APIs are still not supported by WebKit.[107]

[100] Responses to the CMA's information requests [✂].
[101] Progress Delayed Is Progress Denied - Infrequently Noted, accessed by the CMA 20 May 2024.
[102] API which enables modern, reliable offline web experiences and progressive web apps.
[103] Real time network protocol for enabling videoconferencing, desktop sharing, and game streaming applications.
[104] Allows a user to install a web app.
[105] Provides augmented reality and virtual reality input and scene information to web applications.
[106] [✂] response to the CMA's information request [✂].
[107] OWA, Bringing competition to walled gardens, accessed by the CMA 17 June 2024, section 5.4.3.

Figure 4.1: State of web app support on iOS (December 2022)

iOS					
Feature	**Native**	**Safari**	**Chrome**	**Firefox**	**Edge**
Install Prompts	✓	✗ Rejected (7+ years)	✗ Apple Browser Ban	✗ Rejected	✗ Apple Browser Ban
Notifications	✓ (2009)	~ In Development (7+ years)	✗ Apple Browser Ban	✗ Apple Browser Ban	✗ Apple Browser Ban
Deep OS integration	✓	✗ No Signal (5+ years)	✗ Apple Browser Ban	✗ Apple Browser Ban	✗ Apple Browser Ban
App Store Support	✓ (2008)	✗ Rejected (14+ years)	✗ Apple Browser Ban	✗ Apple Browser Ban	✗ Apple Browser Ban
Fullscreen API	✓ (2008)	✗ In Development? (11+ years)	✗ Apple Browser Ban	✗ Apple Browser Ban	✗ Apple Browser Ban
Badging	✓	✗ No Signal (5+ years)	✗ Apple Browser Ban	✗ Apple Browser Ban	✗ Apple Browser Ban
Deep Links	✓	✗ No Signal (7+ years)	✗ Apple Browser Ban	✗ Apple Browser Ban	✗ Apple Browser Ban
Screen Orientation Lock	✓	✗ In Development? (10+ years)	✗ Apple Browser Ban	✗ Apple Browser Ban	✗ Apple Browser Ban
Bluetooth	✓	✗ Rejected (5+ years)	✗ Apple Browser Ban	✗ Rejected	✗ Apple Browser Ban
NFC	✓	✗ Rejected (1+ years)	✗ Apple Browser Ban	✗ Rejected	✗ Apple Browser Ban

Source: OWA, Bringing Competition to Walled Gardens, section 5.4.3.

(d) Figure 4.1 illustrates the features as of December 2022 that were available to native apps on iOS but not available to web apps. It indicates that for some features Apple has not implemented support in Safari on iOS for several years after the feature became available on other platforms. It also indicates that the WebKit restriction prevents third-party browsers on iOS from implementing these features.

(e) Several individual developers submitted that WebKit lacks support for certain features, including important features for PWAs such as push notifications, or full screen,[108] and that missing features can cause developers to make native apps for iOS instead of web apps.[109]

4.29 Other developers also highlighted web app features that were not available on WebKit, including offscreen canvas, APIs for rendering graphics, and access to Bluetooth.[110]

4.30 Some respondents to the qualitative web developer research conducted by Jigsaw Research referred to Apple being slower to allow for the development of web apps in WebKit relative to other browser engines. There was however a sense from one respondent that this was improving.[111]

---

[108] Alister Shepherd, response to MEMS Interim Report, Developer G, response to MEMS Interim Report, Jack Peterson, response to MEMS Interim Report, Developer E, response to MEMS Interim Report, Developer I, response to MEMS Interim Report, Jesper van den Ende, response to MEMS Interim Report, Andy Cowan, response to MEMS Interim Report, Thomas Allmer, response to MEMS Interim Report, Developer A, response to MEMS Interim Report, Developer C, response to MEMS Interim Report, Luca Casonato, response to MEMS Interim Report, Chris Haynes, response to MEMS Interim Report, Mark Johnson, response to MEMS Interim Report, Andreas Bovens, response to MEMS Interim Report, Kimberly Blessing, response to MEMS Interim Report, Thomas Steiner, response to MEMS Interim Report.
[109] Developer A, response to MEMS Interim Report, Developer B, response to MEMS Interim Report, Developer C, response to MEMS Interim Report, Developer E, response to MEMS Interim Report, Developer G, response to MEMS Interim Report, Developer I, response to MEMS Interim Report, Jesper van den Ende, response to MEMS Interim Report, Thomas Allmer, response to MEMS Interim Report, Bradley Taylor, response to MEMS Interim Report, Kimberly Blessing, response to MEMS Interim Report.
[110] Responses to the CMA's information requests [✂].
[111] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p37, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.

4.31 As noted above, Apple has recently added some of these features to WebKit, notably push notifications and full screen API (see paragraph 4.8). This appears to have closed the gap between WebKit and other browser engines on support for web apps. However, whilst this resolves some of the concerns raised by web developers above, other features such as Web Transport are not currently available from WebKit.[112] The delay in implementation of these features in WebKit relative to other browser engines may have adverse implications for web developers.

4.32 Several developers submitted that they incur additional costs or are unable to develop certain products or features as a result of WebKit's alleged lack of support for web apps.[113]

(a) Microsoft submitted that WebKit missing key WebRTC APIs limits its ability to provide a compelling browser-based cloud gaming experience on iOS relative to Android.[114]

(b) Several individual developers expressed concerns in relation to WebKit's lack of support for web apps and the impact on their business. Respondents stated that WebKit lacks support for certain features that can require developers to make native apps for iOS, therefore incurring additional costs.[115]

4.33 The qualitative web developer research conducted by Jigsaw Research indicated less concern from web developers about web apps features than is suggested by the evidence referred to above. Although a few respondents mentioned Apple being slower to support web apps in WebKit relative to other browser engines (see paragraph 4.30), the general view expressed by respondents to the research is that most browsers today have similar features and functionalities,[116] and respondents showed little awareness or concern around the WebKit restriction.[117] Respondents to the research noted a shift from native apps towards web apps, with use of web-apps in development increasing.[118]

---

[112] Web Transport API, accessed by the CMA 17 June 2024.
[113] Responses to the CMA's information requests [✂].
[114] Microsoft response to the CMA's information request [✂].
[115] Alistair Shepherd, Response to MEMS Interim Report; Andy Cowan, Response to MEMS Interim Report; Jack Peterson, Response to MEMS Interim Report; Jesper van den Ende, Response to MEMS Interim Report; Kimberley Blessing, Response to MEMS Interim Report; Luca Casonato, Response to MEMS Interim Report; Mark Johnson, Response to MEMS Interim Report; Thomas Allmer, Response to MEMS Interim Report; Developer A, Response to MEMS Interim Report; Developer B, Response to MEMS Interim Report; Developer E, Response to MEMS Interim Report; Developer G, Response to MEMS Interim Report; Developer I, Response to MEMS Interim Report.
[116] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p47, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.
[117] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p39, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.
[118] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p23, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.

4.34 We note that evidence from the CMA's MEMS report suggests that most app developers submitted that PWAs remain an important channel for many online content providers, even if they are not regarded as viable substitutes for native apps, in particular due to the limited discoverability of web apps compared to native apps that are catalogued, and discoverable through a curated app store.[119]

4.35 Web apps may create higher security risks than native apps, as they are not curated and reviewed in the same way as native apps. Some third-parties therefore raised security concerns around bringing web apps closer in functionality with native apps. For instance:

(a) RET2 stated that allowing web apps to access more APIs increases the 'attack surface' and can therefore degrade the current level of security or privacy for those apps and the device at large.[120]

(b) Mozilla submitted that it has actively decided not to add some APIs used by native apps to Gecko due to security and privacy concerns.[121]

4.36 These arguments are consistent with Apple's submissions that it is necessary to balance support for web apps with assurances that any new functionality provided does not compromise user privacy and data security (see paragraph 4.7).

4.37 Overall, the evidence shows that WebKit has lagged other browser engines in support for web apps, although this has improved recently. There is some evidence that this has created additional costs for developers, particularly those with specific business models reliant on exploiting web apps, for example Microsoft and its cloud gaming service. However not all developers were concerned about it, and some either raised security concerns around web apps or considered web apps would not be effective substitutes for native apps for broader reasons (in particular, the easier discoverability of native apps which are catalogued and discoverable through a curated app store).

Bugs and security issues

4.38 This section considers evidence from web developers on the extent of bugs or security issues in Safari and WebKit relative to other browsers and browser engines, and the implications of this for developers. The existence of bugs and security issues can lead to a worse user experience if websites are not rendered correctly or create security risks, and may create additional costs for web developers who are required to fix them. This issue is also considered in Appendix A.

---

[119] MEMS final report, paragraphs 4.131-4.132 and 5.139.
[120] RET2 advice to the CMA [✂].
[121] Note of meeting with Mozilla [✂].

4.39 Some developers highlighted issues related to bugs on WebKit, and indicated that Apple is slow to resolve issues:[122]

(a) OWA submitted that many developers had complained about the presence of bugs on Safari and provided links to several complaints.[123]

(b) One party [✂] submitted that WebKit suffers from bugs and technical issues, including in the implementation of certain APIs, and that Apple does not indicate whether it is preparing a fix for a specific issue or provide timelines for the release of a fix. This party [✂] also stated that fixes can take time to reach users as updates to WebKit only happen through full iOS updates.[124]

(c) Several individual developers responding to the CMA's MEMS Interim Report also expressed concerns in relation to WebKit specific issues.[125]

4.40 Some developers also highlighted security concerns with WebKit, submitting that Apple is slow at fixing security issues and expressing concerns over Apple's approach to releasing security updates:[126]

(a) OWA submitted that WebKit is slower than Blink or Gecko at fixing security issues based on Project Zero data (which is maintained by Google).[127] It also stated that, according to a public database of reported vulnerabilities, Safari has had more Common Vulnerabilities and Exposures (CVEs) than Chrome and Firefox between 2014 and 2021.[128, 129]

(b) Several individual developers expressed concerns in relation to users not being able to switch to a browser which uses a different engine on iOS to protect themselves from security issues affecting WebKit before they are patched.[130] However, we note that in practice users are unlikely to be sufficiently informed about security vulnerabilities to take action, even if the option were available to them.

---

[122] Responses to the CMA's information requests [✂].
[123] OWA, Bringing competition to walled gardens, accessed by the CMA 17 June 2024, section 5.6.
[124] [✂] response to the CMA's information request [✂].
[125] Matt Perry, response to MEMS Interim Report, Alistair Shepherd, response to MEMS Interim Report, Jack Peterson, response to MEMS Interim Report, Developer I, response to MEMS Interim Report, Developer K, response to MEMS Interim Report, Patrick Grey, response to MEMS Interim Report, Developer H, response to MEMS Interim Report, Developer C, response to MEMS Interim Report, Andreas Bovens, response to MEMS Interim Report, Kimberly Blessing, response to MEMS Interim Report, Gopal Venkatesan, response to MEMS Interim Report, Chris Haynes, response to MEMS Interim Report
[126] RET2 advice to the CMA [✂].
[127] OWA, Bringing competition to walled gardens, section 8.2.1; Project Zero, accessed by the CMA 17 June 2024.
[128] OWA, Bringing competition to walled gardens, section 8.2.3; CVE details, accessed by the CMA 17 June 2024.
[129] This data only includes reported fixes.
[130] Developer D, response to MEMS Interim Report, Jesper van den Ende, response to MEMS Interim Report, Andy Cowan, response to MEMS Interim Report, Developer K, response to MEMS Interim Report, Paul Neave, response to MEMS Interim Report, Niels Leenheer, response to MEMS Interim Report, Developer C, response to MEMS Interim Report, Developer J, response to MEMS Interim Report, Luca Casonato, response to MEMS Interim Report, Chris Haynes, response to MEMS Interim Report.

(c) OWA submitted that Apple's policy of pairing WebKit and iOS updates can negatively impact security as browser engine exploitation risk increases when engines are not updated regularly. This is because users may delay installing updates given iOS updates make the device unusable for several minutes. In addition, OWA submitted that Apple does not update iOS on older devices, which means that older iPhone devices use outdated versions of WebKit.[131]

4.41 However, some developers did not highlight security concerns with WebKit. The qualitative web developer research conducted by Jigsaw Research also found that for respondents, iOS was perceived as being more secure than Android, driven by more stringent guidelines and permissions.[132]

4.42 We received some evidence from browser vendors on the relative stability of their browsers on iOS and Android. We have requested more recent data from browser vendors and will consider this question further.

Summary of emerging thinking on implications of WebKit restriction for web developers

4.43 Overall, there is less clear evidence of a significant impact on web developers than on browser vendors. In particular, evidence from some web developers suggests that compatibility costs are either not very significant (eg because of software tools enabling them to ensure compatibility) or are not WebKit-specific, and simply arise because of the presence of multiple browsers and browser engines.

4.44 There is evidence that WebKit has been behind other browser engines in its support for web apps, and that this has impacted certain web developers through increased costs or limiting their ability to offer products for a non-trivial period of time. However, there is also evidence that WebKit has improved its support more recently (since 2022), and many web developers we heard from (including through the qualitative web developer research conducted by Jigsaw Research) did not raise this as a significant issue. There may also be security concerns around the implementation of some features that may in some cases justify Apple's decision not to implement them in WebKit.

4.45 A minority of web developers stated that WebKit's performance was worse than that of alternative browser engines and that this creates additional work for them.

4.46 It is possible that the concerns raised by some developers regarding feature support and compatibility have resulted from the WebKit restriction, and this in turn

[131] OWA, Bringing competition to walled gardens, accessed by the CMA 17 June 2024, section 8.2.1-8.2.2.
[132] Jigsaw Research (2024), Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines, p56, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.

has led to worse outcomes in terms of the ability of web developers to implement certain features in their websites and web apps.

4.47 We also note that evidence received during this market investigation from web developers so far (both through RFI responses and the qualitative web developer research conducted by Jigsaw Research) has indicated less concern about the WebKit restriction amongst developers compared to the evidence received during the CMA's MEMS.

4.48 Overall, the mixed nature of the evidence regarding the impact on web developers is likely to reflect the fact that a key priority for web developers is ensuring compatibility between different browsers, and the WebKit restriction likely has little direct impact on this as they would still have to ensure compatibility with the same major browsers and browser engines even if alternative browser engines were permitted on iOS.

5. Apple's justification for the WebKit restriction

5.1 This section considers Apple's justification for the WebKit restriction. It considers evidence from Apple on its rationale for the restriction and a preliminary assessment of this justification.

5.2 Apple submitted that:

(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.

5.3 For context, while Apple refers to security, privacy and performance as three parameters of competition between mobile ecosystems (and as three benefits associated with the WebKit restriction), we have focused in particular on Apple's arguments on security. This is because:

(a) We have seen evidence (see paragraph 5.342) that privacy may be interpreted differently by different stakeholders, and this is reflected in the variety of privacy-preserving features available in the mobile browser market. For example, some stakeholders focus on limits to 'tracking' while others on giving users control over their data (which may entail getting compensated for allowing tracking). As a result, we currently consider privacy to be a quality parameter over which different stakeholders may compete 'horizontally' (ie by offering different versions of it) as well as 'vertically' (by offering more or less privacy) and in relation to which consumers may have different preferences, depending on whether they align with the interpretation/mission of a specific stakeholder.

(b) We would expect the performance of a browser (including its speed, drain on the device battery, etc) to be, at least to some extent, reflected in a given user's preference for a specific mobile browser. Further, performance is likely to be a parameter that users are reasonably well placed to evaluate and express a preference on, including by potentially trading it off against other parameters. For example, Google submitted that there can sometimes be a trade-off between performance and security as security technologies may use more of a device's memory or require additional runtime checks.[133]

(c) We have seen evidence suggesting that security is the foundation for privacy and there is more agreement in the industry as to what a baseline for security

[133] Google response to the CMA's information request [✂]; Runtime checking allows developers to automatically detect errors such as memory access errors and memory leaks.

is (compared to what privacy is) and more convergence on best practices. For example, Apple submitted that, in terms of industry recognition and standards, there is more clarity and alignment on security standards, whereas the issue of privacy is running a number of years behind.[134] Further, it is likely to be more difficult for users to compare the security level of different devices.

5.4 We currently consider the two main questions to assess Apple's argument to be:

(a) the extent to which the WebKit restriction:

(i) improves the security, privacy, and performance of mobile browsers on iOS (and iOS devices more generally);

(ii) leads to greater competition between the iOS and Android mobile ecosystems;

(iii) in turn drives greater competition in browsers.

(b) if so, we would then consider (i) whether any such benefits offset any negative impacts on competition in the browsers market resulting from the restriction; and (ii) whether there is a less restrictive way to achieve those benefits. This latter question is largely relevant to remedy design. Therefore, we are not addressing it in detail in this paper.

5.5 The section below considers a range of evidence, including:

(a) Apple and third-party submissions;

(b) the external security expert advice commissioned during the CMA's MEMS from the security consulting firm RET2;

(c) our assessment of how the major browser engines compare across a range of metrics (see Appendix A);

(d) results from the smartphone user survey commissioned by the CMA during the CMA's MEMS; and

(e) qualitative consumer research commissioned by the CMA for this market investigation, conducted by Verian (formerly Kantar Public).

[134] Note of meeting with Apple [✂].

Evidence from Apple

Rationale for the WebKit restriction

5.6 Apple submitted that the WebKit restriction is needed to ensure the high standards of privacy, security, and performance that users have come to expect of its devices. It submitted that it requires that apps browsing the web use the WebKit browser engine, which has privacy and security features built in and is optimised for high performance on iOS devices.[135]

5.7 Apple submitted that the WebKit restriction is complementary to its policies around the App Store as it ensures that iOS devices are private, safe, and secure when browsing the web.[136] More specifically, Apple submitted that it uses the App Review and restrictions on sideloading to guard against native applications that might violate a user's privacy or security, or undermine device performance but that it must solely rely on device-based protections for web content, as this does not go through App Review and that WebKit is a critical component of these device-based protections.[137, 138]

5.8 Apple submitted that it integrated WebKit into iOS as a 'critical system feature to ensure that iOS devices are private, safe, and secure while maintaining outstanding browsing performance'.[139] More specifically:

(a) Apple stated that it leverages integration of WebKit with Apple's processor (Apple Silicon) and operating system iOS to improve security. Apple gave examples of security features resulting from such integration, including WebKit's 'customised sandbox profile',[140] Pointer Authentication Codes (PACs)[141] and access limitations to the just-in-time compiler (JIT).[142]

---

[135] Apple submission to the CMA [✂]; Apple response to MEMS Interim Report, paragraph 87-90.
[136] Apple submission to the CMA [✂]; Apple, response to MEMS Interim Report, paragraph 87-90.
[137] Apple submission to the CMA [✂]; Apple, response to MEMS Interim Report, paragraph 87-90.
[138] Apple submitted that browser applications, and the browser engines on which they are based, are an important entry point for users to access web-based experiences, but they are also a critical threat vector, particularly for iOS devices. Because Apple restricts sideloading, malicious actors have a vastly limited opportunity to attack iOS devices via native apps; this incentivises them to turn more of their attention to web-based attacks to try to infiltrate iOS devices. In particular, iOS's stricter controls in these other domains makes browser engines, which use technologies like just-in-time compilation on untrusted and potentially malicious content from the web, a more attractive and higher volume target. Apple submission to the CMA [✂].
[139] Apple submission to the CMA [✂].
[140] Apple stated that WebKit on iOS supports a customised sandbox profile that represents 'a decade's worth of security improvements' and which is more stringent than the sandbox for native iOS apps. Apple stated that WebKit's sandbox profile restricts the attack surface from which malicious actors can attack iOS processes. Apple also stated that it regularly updates its sandbox and develops new sandbox technology in response to evolving threats. See: Apple, response to MEMS Interim Report, paragraph 88.
[141] Apple submitted that it implements Pointer Authentication Codes (PAC) to prevent attackers from gaining code execution outside of the JIT. PACs provide cryptographic signatures and authentication to function pointers and return addresses to protect against the exploitation of memory corruption bugs. PACs provides protection against the exploitation of memory corruption bugs. See: Apple, response to MEMS Interim Report, paragraph 88.
[142] Apple stated that the JIT allows apps browsing the web to quickly and efficiently render JavaScript content, which is valuable for users but also exposes a vulnerability that malicious actors can exploit. To mitigate the risks posed by the

(b) Apple also stated that the WebKit restriction allows Apple to address malware attacks and security vulnerabilities quickly and effectively to all apps via WebKit updates.[143]

(c) Apple submitted that WebKit has been designed and optimised for use on iOS devices, and that this allows iOS devices to outperform competitors on web-based browsing benchmarks, while also achieving industry-leading power efficiency and battery performance.[144]

(d) Apple stated that by integrating WebKit into iOS, it is able to guarantee robust user privacy protections for every browsing experience on iOS. Apple gave the example of privacy-enhancing features it integrates into WebKit, including third party cookie blocking by default, storage and service worker partitioning (to ensure secure offline access of web pages), private browsing, requiring a user permission for websites to access the device orientation or motion APIs, and prevention of fingerprinting of device microphones or cameras.[145]

(e) Apple submitted that its approach to WebKit has given the iPhone significant advantages, including a reduced attack surface, risks can be managed at platform level, Apple can control the content and cadence of security updates, robust privacy features can be built into every build of iOS, and browsers on iOS can take advantage of new features immediately as Apple innovates.[146]

5.9 Apple also submitted that device performance, security, and privacy are key to the competitive differentiation between iOS and Android. Removing the WebKit restriction would therefore essentially remove differentiation with Android and diminish competition between iOS and Android devices.[147] Apple submitted that evidence from its [✂] found that security and privacy are amongst the most important product features for iPhone buyers, being described as extremely important by [✂] of respondents.[148]

5.10 Further, Apple submitted that 'any erosion of the WebKit model on iOS' would result in the degradation of iOS privacy or performance to the detriment of users and competition, and that malware and other attacks on iOS devices would

JIT, WebKit leverages tight integration with iOS hardware. Apple employs a highly effective hardware security extension (APRR) to prevent attackers gaining access to the JIT. See: Apple, response to MEMS Interim Report, paragraph 88.
[143] Apple, response to MEMS Interim Report, paragraph 32 and 88.
[144] Apple, response to MEMS Interim Report, paragraph 90.
[145] Apple, response to MEMS Interim Report, paragraph 89.
[146] Note of meeting with Apple [✂].
[147] Apple, response to Market Investigation Reference consultation, paragraph 30.
[148] Apple response to the CMA's information request [✂].

increase, bringing iOS 'closer to Android's wild west-like levels of device threats.'[149]

Apple's views on allowing alternative browser engines on iOS

5.11 Apple submitted that browser engines create significant security risks because they operate on untrusted and unvetted content with no review process and are by far the most common vector for operating system security exploits. Apple stated that significant exploits typically occur at least once a month in established browser engines, and that exploits are well compensated due to the extent of access granted from a successful attack and the difficulty to accomplish the exploit relative to other forms of attacks.[150]

5.12 Apple also submitted that, if each browser uses its own browser engine (as would be the case if third-party browser engines were allowed on iOS), each would introduce delays in responding to security vulnerabilities (known as patch gaps). Apple stated that, on Android, as of January 2024, an assessment of browser apps in the Google Play Store showed that in 2023 there were nine browsers with more than five million lifetime downloads that had out-of-date browser engines, and as of March 2024, an assessment of browser apps in the Google Play Store showed that in 2024, there are six browsers with more than five million downloads with out-of-date browser engines. In contrast on iOS, due to the WebKit restriction, Apple can update all browsers simultaneously.[151]

5.13 As a result of the above, Apple submitted that mandating Apple to allow apps to use third-party browser engines on iOS would break the integrated security model of iOS devices, reduce their privacy and performance, and ultimately harm competition between iOS and Android devices. This is because, according to Apple, users trust that their iOS devices offer 'world-class security and privacy, as well as all-day battery life, out-of-the-box', all qualities that substantially enhance iOS device appeal compared to Android devices.[152]

5.14 Apple referred to changes that it is making in the EU in order to comply with provisions of the Digital Markets Act (DMA),[153] which requires that designated 'gatekeepers' shall not require the use of their own web browser engine in the context of services provided using that gatekeeper's designated 'core platform services'.[154] Apple stated that these changes have introduced significant risks to

[149] Apple submission to the CMA [✂].
[150] Note of meeting with Apple [✂].
[151] Note of meeting with Apple [✂].
[152] Apple, response to MEMS Interim Report, paragraph 91.
[153] 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).
[154] Digital Markets Act, Article 5.7

the platform and for users, and Apple believes this degrades the user experience of iOS and iPhones in the EU.[155]

Preliminary views on Apple's justification

5.15 This section considers and assesses Apple's justification for the WebKit restriction. In doing this, we consider the evidence on:

(a) the extent to which the WebKit restriction improves the security, privacy, and performance of mobile browsers on iOS (and iOS devices more generally);

(b) the extent to which the WebKit restriction leads to greater competition between the iOS and Android mobile ecosystems;[156]

(c) the extent to which the WebKit restriction in turn drives greater competition in browsers – which could offset any potential harmful effects on competition between browsers that operate on iOS.

Extent to which the WebKit restriction improves the security, privacy and performance of iOS and iPadOS devices relative to Android devices

Security

5.16 Overall, we have not yet seen clear evidence that the WebKit restriction confers a significant improvement in security compared to a situation where other browser engines would be allowed on iOS. The WebKit restriction may create security benefits for Apple devices (albeit these could be potentially extended, at least to some extent, to rival browser engines), through enabling closer integration between WebKit and device hardware, and providing Apple with greater control over browser engines such that it can ensure browsers use an up-to-date browser engine and do not implement features that may create a security risk. However, as explained in Section 3 above, the restriction limits the ability for browser vendors to improve their browsers and differentiate themselves from Safari, including on security.

5.17 Appendix A considers several metrics relating to security vulnerabilities and bugs identified in each browser engine, including the time taken to fix the most severe issues, and the frequency of browser updates available to users. Whilst WebKit generally had fewer identified vulnerabilities than Blink or Gecko, the time taken to fix vulnerabilities and bugs in WebKit was longer, and updates to WebKit were less frequent. However, limitations around measuring vulnerabilities and comparability

[155] Note of meeting with Apple [✂].
[156] For this to be true, there needs to be competition between iOS devices and Android devices. Therefore, we first consider the extent of any ecosystem competition.

of publicly available bug data mean that it is difficult to draw firm conclusions on the relative security outcomes of different browser engines. See Appendix A for additional detail.

5.18 Apple's submissions in relation to security may have some merit as integration between hardware and software grants Apple more control over iOS compared to Google's control over Android, and allows it to limit the addition of browser features that might compromise security (eg by providing websites with potentially harmful access to device storage). However, we note that there might be an inherent difference in how closed systems (such as Apple's) approach security compared to open ones (such as Google's). In this respect, Google stated that while historically closed systems had been considered more secure than open ones, experts are now saying that the two are on par.[157] Google provided an externally commissioned research report [✂].[158]

5.19 Consistent with Apple's submissions, we have heard that Apple's control over the hardware can allow it to adopt hardware-specific security features, which can help to provide a high level of protection.[159] However, as part of the measures Apple has announced in response to the DMA, Apple has made some of these security features available to other browser engines, such as Pointer Authentication Codes,[160] demonstrating that benefits of hardware integration could potentially be extended to other browser engines.

5.20 Apple has submitted that its ability to update all browsers simultaneously makes browsing on iOS more secure. In contrast, the greater fragmentation (ie different browsers using different versions of a browser engine) that could be created by allowing browsers to incorporate their own browser engine could lead to browsers using outdated browser engines, creating a security risk.

5.21 However, several stakeholders submitted that when a security flaw is found in WebKit, consumers are unable to protect themselves by switching to a browser based on a different browser engine and are therefore vulnerable until a fix is deployed to WebKit (which can take several weeks).[161, 162] As noted in paragraph 4.40(b), users may be unlikely to be sufficiently informed about security vulnerabilities to take this action, even if the option were available to them.

---

[157] Note of meeting with Google [✂].
[158] Google internal document [✂].
[159] RET2 advice to the CMA [✂].
[160] Improving control flow integrity with pointer authentication | Apple Developer Documentation, accessed by the CMA 14 May 2024.
[161] Submissions to the CMA: [✂], [✂], [✂], question 1; Jesper van den Ende, Andy Cowan, Developer K, Paul Neave, Niels Leenheer, Developer C, Luca Casonato, Chris Haynes, responses to MEMS Interim Report.
[162] Five respondents to the MEMS Interim Report submitted that Apple took more than a month to patch a serious vulnerability associated to the IndexDB API which compromised the privacy of browsers based on WebKit. iOS users remained vulnerable when browsing the web until the patch was shipped. Source: Jesper van den Ende, Niels Leenheer, Developer C, Luca Casonato, Chris Haynes, responses to MEMS Interim Report.

5.22 One party [✂] also submitted that Apple's approach to updating WebKit creates security risks:

(a) It [✂] stated that, as Apple bundles WebKit updates with iOS system updates, this leads to larger and less frequent updates.[163]

(b) It [✂] also stated that user uptake of WebKit updates is slower compared to other browser engines as updates cannot happen automatically in the background. Google said that as of [✂], [✂] of weekly active Chrome on Android users have been updated [✂].[164]

5.23 Public data shows that during 2022 and 2023, Chrome and Firefox released 24 and 26 browser updates respectively, while Safari released 13.[165] A blog article from Exodus Intelligence suggested that, when Chrome went from releasing security updates every six weeks to every two weeks or less, n-day[166] exploits on Chrome had become impractical.[167]

5.24 A 2021 Apple internal document stated [✂].[168] Apple submitted that it has [✂].[169]

5.25 Apple also stated [✂]. [✂].[170]

5.26 Apple pointed to a 'patch gap' issue on Android, meaning the time between when a vulnerability is discovered and patched by an individual mobile browser, during which attackers may create exploits to target users of the relevant app and system. This is because each browser developer on Android updates its browser apps at a frequency it chooses.[171]

5.27 Google acknowledged that it was possible to exploit inactive, outdated browser apps, but said it had recently introduced changes to make such exploits harder. Google said these changes have made it harder for an app to open a browser app through the intents system, depending on the precise operation the app wants to perform.[172,173]

---

[163] [✂] response to the CMA's information request [✂]; note of meeting with [✂].
[164] Note of meeting with [✂].
[165] Appendix A, paragraph 2.52.
[166] An n-day exploit is an exploit that remains unfixed n days after a vulnerability is made public.
[167] Exodus Intelligence, A EULOGY FOR PATCH-GAPPING CHROME - Exodus Intelligence, accessed by the CMA 17 June 2024.
[168] Apple internal document [✂].
[169] Apple response to the CMA's information request [✂].
[170] Apple response to the CMA's information request [✂].
[171] Note of meeting with Apple [✂].
[172] Note of meeting with Google [✂]
[173] One party response to the CMA's information request [✂]. noted that similar exploits can happen in iOS, even in absence of intents, as an app can register to open itself automatically in response to different URLs. Source: note of meeting with [✂].

5.28 Independent research has also shown that several hundred browsers are available on Android, a number of which have security and privacy flaws.[174] This may demonstrate how the large number browsers operating on different versions of browser engines on Android may contribute to security vulnerabilities and potentially expose users to harm.

5.29 As described in Section 3 above, the WebKit restriction may also decrease device security by limiting the ability of browser vendors to deliver security enhancing features or improvements. For example, one browser vendor [✂] stated that 'site isolation', a key security feature it developed [✂], is available on its browser [✂] on all operating systems apart from iOS, due to the WebKit restriction because WebKit does not offer this feature.[175] One browser vendor [✂] also told us that the Strict Content Security Policy (CSP) variant it had implemented on its browser [✂] six years earlier, only became available on Safari on iOS recently. In order to implement this CSP variant on iOS, this browser vendor [✂] had to [✂] to build this security feature in WebKit.[176]

5.30 According to the security consulting firm RET2, from which the CMA commissioned advice during the CMA's MEMS, there is no way to effectively measure how many vulnerabilities software contains. As not all fixed vulnerabilities are reported, the number of vulnerabilities found in a piece of software depends on the scrutiny that the software receives, which in turn may depend on resources allocated to this. A higher number of security fixes could therefore indicate that a piece of software has more vulnerabilities but also that more effort is being devoted to identifying and eliminating them.[177] Public data from the Cybersecurity and Infrastructure Security Agency shows that between 2022 and 2023 there were over 50 browser vulnerabilities known to have been exploited, including 15 on WebKit, 3 on Chrome (Blink), 24 on all Chromium based browsers, and 8 on Firefox (Gecko).[178] The high number of exploits for Chromium based browsers may be a result of the greater fragmentation of Chromium/Blink ie the number of browsers using different versions of the browser engine, some of which may not be updated or patched frequently (see paragraph 5.26).

5.31 Overall, the WebKit restriction may provide some security benefits to iOS devices, through allowing greater integration between the browser engine and device hardware. It also enables Apple to have control over browser engines used on iOS, ensuring that all browsers use an up-to-date browser engine for which Apple can control which features are available. However, it is not clear from the evidence available to date that WebKit has better security outcomes compared to other

[174] Pradeep A et al. (2022) Not Your Average App: A Large-scale Privacy Analysis of Android Browsers, accessed by the CMA 17 June 2024.
[175] Note of meeting with [✂].
[176] Note of meeting with [✂].
[177] RET2 advice to the CMA [✂].
[178] Appendix A, Table 2.5.

browser engines. At the same time, the WebKit restriction may also have a negative impact by limiting the ability of browser vendors to deliver security enhancing features or improvements.

Privacy

5.32 Overall, we have not to date seen clear evidence that the WebKit restriction improves privacy on iOS devices. As explained in Section 3 above, the Webkit restriction limits the ability for browser vendors to innovate and differentiate themselves from Safari, including on privacy. Different stakeholders appear to interpret privacy differently, compared to security, in relation to which there is more alignment across stakeholders. As a result, consumers may have different preferences on privacy, or the importance of privacy relative to other parameters, and therefore browser vendors may differentiate 'horizontally'[179] ie by implementing different features or privacy policies to cater to certain customer preferences, as well as 'vertically', meaning positioning their products on a scale of more or less privacy.[180]

5.33 When discussing the WebKit restriction in written submissions, Apple has often grouped together privacy and security. [✂]. When asked about the difference between the two, Apple clarified that security is critical to protect users and serves as the foundation for privacy.[181] However, Apple has also stated that, in terms of industry recognition and standards, broadly speaking, there is more clarity and alignment on security standards, whereas while there is increasing consensus and international standards on privacy, this is still not at the same level as on security.[182]

5.34 While privacy and security are connected, with a security issue potentially having implications for privacy, and may not necessarily always be distinguishable by users, the two can be distinct from a supply side perspective. Evidence indicates that different browser vendors compete on privacy but do not necessarily agree on what is meant by it and on what is the best way to grant it. For example:

(a) Apple stated that 'the defining principle for Apple is that the user is empowered to choose how their data is treated and is given sufficient information and options to allow them to make an active choice.'[183]

[179] Horizontal differentiation caters to different customer preferences, over which there is no objective ranking eg the colour of a car, where customers will have different preferences. It is in contrast to vertical differentiation where customers have aligned preferences eg the fuel efficiency of a car, where better fuel efficiency is preferred by all.
[180] Indeed, there may be a trade-off between privacy and browser capabilities, with [✂]. Google internal document [✂].
[181] Apple submission to the CMA [✂]; Apple response to the CMA's information request [✂].
[182] Note of meeting with Apple [✂].
[183] Apple, response to MEMS interim report, paragraphs 28-30.

(b) Mozilla has described privacy as 'the act of giving users the right to control how their data is collected, stored, and used, and not using it irresponsibly'.[184]

(c) Brave appears to consider privacy mostly being about giving users control over their data and even the ability to monetise ads and tracking if they want, including through its Brave Rewards feature.[185]

5.35 The above suggests that, whilst there may be some element of vertical differentiation in relation to privacy, companies may also differentiate their products 'horizontally', by offering features to cater to certain customer preferences.

5.36 It is true that privacy can be implemented at the browser engine level, and Apple's Intelligent Tracking Prevention (ITP)[186] is a successful example of this, therefore the WebKit restriction allows Apple to include privacy controls at the browser engine level. It also allows Apple to limit browser features that might harm user privacy eg by enabling tracking or monitoring location data.

5.37 However, browser vendors can also compete on privacy by adding privacy enhancing features to their browsers and are already doing this on iOS, for example Brave's Global Privacy Controls feature (see paragraph 3.3). As explained in Section 3 above, browser vendors submitted that the WebKit restriction is limiting their ability to differentiate their browsers further on privacy features as it does not allow them to offer certain protections that go beyond what WebKit grants, or that work differently to WebKit.

5.38 Our emerging thinking, based on our assessment of the evidence so far suggests that, although the WebKit restriction offers a method of guaranteeing a baseline level of privacy protection through controlling the access that browsers have to user data at the operating system level, the WebKit restriction prevents browsers from offering further privacy features that could potentially better meet the preferences of certain users, and limits the ability of browser vendors to compete on privacy features.

Performance

5.39 We have not yet seen clear evidence that the WebKit restriction improves performance of browsers on iOS (or iOS devices more generally) overall.

---

[184] Mozilla – Privacy on the web, accessed by the CMA 19 April 2024; This is distinct from security which Mozilla described as 'the act of keeping private data and systems protected against unauthorized access.'
[185] The Brave Privacy Glossary, accessed by the CMA 17 April 2024 states that 'privacy means that your personal data isn't seen by anyone whom you don't want to see it, and isn't used by anyone in ways you don't approve of'; See also Brave Rewards, accessed by the CMA 18 April 2024.
[186] John Wilander-Intelligent Tracking Prevention, WebKit blog post, accessed by the CMA on 17 May 2024.

5.40 It is reasonable to expect the integration between hardware and software could result in performance advantages (eg including in terms of browser speed, drain on the device battery, etc), given Apple would design both to be optimised for each other. Integrating WebKit with the operating system therefore is clearly a design choice by Apple which may partially be aimed at making sure that WebKit works well on Apple devices. By requiring that all browsers on iOS use WebKit, Apple can ensure that all achieve this level of performance.

5.41 However as described in section 3, the restriction also limits browser vendors' ability to improve their browsers and differentiate themselves from Safari, including on performance of the browser. This may prevent browsers from improving performance further than the level provided by WebKit. It also prevents browsers from competing on performance on iOS.

5.42 Further, we would expect performance to be, at least to some extent, reflected in a given user's preference for a specific mobile browser. In addition, performance is likely to be a parameter that users are reasonably well placed to evaluate and express a preference on, including by potentially trading it off against others. Users are therefore likely to be better placed to make informed choices on browser performance and have less need for platform level restrictions that ensure a given performance level, compared to a parameter such as browser security, where users are likely to be less well informed.

Extent to which the WebKit restriction may increase ecosystem competition

5.43 In this section, we assess the extent to which the WebKit restriction increases competition between iOS and Android devices. Therefore, we first consider the extent of ecosystem competition, before considering the importance of security, privacy, and performance to ecosystem competition.

Extent of ecosystem competition

5.44 The CMA has recently considered ecosystem competition in its MEMS report and concluded that iOS and Android hold an effective duopoly in the UK, with a roughly even split of active devices, and this has been stable over time.[187] While it is reasonable to expect iOS and Android to exert some constraint on each other (as 'out-of-market constraints'), evidence shows that:

(a) Pricing between iOS and Android devices is clearly segmented, with Android accounting for 100% of devices sold for £300 or less (lower-priced devices) in 2021 and Apple accounting for 77% those sold at more than £300 (higher-

[187] MEMS final report, paragraph 3.176-3.179.

price devices) in the same period. This suggests limited price competition between iOS and Android.[188]

(b) Results from a smartphone user survey commissioned by the CMA during the CMA's MEMS show that most users purchase a 'replacement device' meaning that they are not purchasing their first mobile device. Among those, the vast majority do not switch to a different operating system. More specifically, 90% of the surveyed iOS users purchased an Apple device when changing phone while 91% Android users purchased an Android device. Across both groups, 87% said they did not consider switching.[189]

(c) The perceived barriers that surveyed users identified as reasons for limited switching, appeared more significant with respect to switching from iOS to Android than from Android to iOS.[190] This suggests that there is an asymmetry in the constraint iOS and Android exert on each other, with users finding it more difficult to switch from the iOS ecosystem than from the Android ecosystem and therefore Apple being less constrained by Google in mobile operating systems than vice versa.

5.45 Overall, the evidence suggests that competition between the iOS and Android ecosystems is relatively limited.

Importance of security, privacy, and performance to ecosystem competition

5.46 As described above, Apple has submitted that device performance, security, and privacy are key to the competitive differentiation between iOS and Android and that, as a result, the WebKit restriction improves the ability of iOS devices to compete with Android devices.

5.47 However, results from the consumer survey conducted during the CMA's MEMS suggest that, whilst security, privacy, and performance are factors considered by some consumers when choosing a mobile device, they are not the most important:

(a) Only 29% of consumers on iOS and 22% on Android named security and privacy as a factor which was important to their decision to choose their

[188] MEMS final report, paragraph 3.79; Such price segmentation is also consistent with Apple and Google respective business models, with Apple predominantly monetising through selling high-end devices and Google via advertising and traffic, and therefore having an incentive to have its OS on as many devices as possible. In terms of overall volume, devices sold at 300 pounds or less account for much higher volume than those sold for over 300.
[189] Consumer purchasing behaviour in the UK smartphone market for the CMA's Mobile Ecosystem Market Study, Accent, pages ii and iii.
[190] These were (i) learning costs associated with using a device from a different ecosystem; (ii) costs associated with the transferring of data and apps; and (iii) costs associated with the usage of first party apps or the ownership of other devices from the same ecosystem; Consumer purchasing behaviour in the UK smartphone market for the CMA's Mobile Ecosystem Market Study, Accent, pages 39, 41.

current smartphone, making security and privacy the eighth most named factor for both sets of consumers.[191]

(b) 42% of iOS consumers and 51% of Android consumers named battery life (an element of performance) as a factor, making this the fourth most named factor for iOS consumers, and the third most named factor for Android users.[192] It is however likely that the device itself is also important for battery life, as well as the operating system.

(c) However, for iOS consumers, 66% named brand as an important factor (making it the most listed factor), whilst 40% named operating system (making it the fourth most named factor). It is possible that that this is based on a perception that the Apple brand and iOS offer better security, privacy, and performance, although this cannot be determined from the survey data.[193]

5.48 Further, consumers may not be able to accurately assess factors such as the security and privacy of an ecosystem when making their first decision of choosing a mobile device, which effectively determines which operating system they will be using as well as which browser engine (in the case of iOS).

Extent to which ecosystem competition (in turn) drives competition in browsers

5.49 For any potential benefits of the WebKit restriction to offset any potential harmful effects on competition between browsers, such benefits would need to increase competition in the same market, which we currently consider to be the supply of mobile browsers on iOS.

5.50 This section considers: (i) the link between competition between mobile ecosystems and mobile browsers; (ii) evidence of competitive interactions between browsers and browser engines across ecosystems; and (iii) evidence on Apple's investment in WebKit.

Importance of browsers in competition between mobile ecosystems

5.51 If browsers are an important element of competition between the iOS and Android ecosystems, then it is possible that Apple and Google may be incentivised to improve their respective browsers on their platform to increase the appeal of iOS and Android to users.

---

[191] Consumer purchasing behaviour in the UK smartphone market for the CMA's Mobile Ecosystem Market Study, Accent, Figure 5.
[192] Consumer purchasing behaviour in the UK smartphone market for the CMA's Mobile Ecosystem Market Study, Accent, Figure 5.
[193] Consumer purchasing behaviour in the UK smartphone market for the CMA's Mobile Ecosystem Market Study, Accent, Figure 5.

5.52 However, the results of the consumer survey cited in the CMA's MEMS report suggest that app store conditions and availability of a certain app on a device are not the most important factor driving a user's choice of mobile device.[194] As browsers are only a single app among many that a smartphone user would use, it is unlikely to be an important determinant of device choice.

5.53 As described in 'WP1 - Nature of competition in the supply of mobile browsers and browser engines'[195] qualitative consumer research commissioned as part of this market investigation conducted by Verian found that there is low engagement with mobile browsers by users, and awareness of different mobile browsers is low.[196] This also suggests that browsers likely have very limited influence over a user's decision to purchase a given device.

5.54 On this basis, it is unlikely that ecosystem competition between Apple and Google would be a significant driver of competition between Safari and Chrome.

Evidence of competition between browsers across ecosystems

5.55 Evidence of competition between browsers or browser engines across the iOS and Android ecosystems could also indicate that competition between ecosystems may drive competition between browsers.

5.56 Apple has submitted that there is robust competition between browser engines and that suppliers 'are constantly introducing new features, many of which mirror features that competitors have introduced.' It highlighted the examples of Google and Mozilla implementing similar features following Apple's introduction of ITP, and Apple implementing Web RTC in 2017 after it was first introduced in Chrome. [✂].[197] With reference to the implementation of push notifications and pointer lock API in WebKit, Apple also stated that it had devoted significant resources to achieving feature parity while maintaining its stringent security, performance, and privacy protections.[198]

5.57 While this may indicate some competition between browser engines across iOS and Android devices, it is also possible that some of this competition may take place between desktop browsers, where WebKit and Blink compete on the same platform, rather than on mobile where WebKit is only present on iOS, and Blink is only present on Android.

---

[194] Only 14% of iOS users and 15% of Android users considered the range and quality of mobile apps available on a device as an important factor in their decision to buy their current smartphone. See: Accent Report 'Consumer purchasing behaviour in the UK smartphone market for the CMA's Mobile Ecosystems Market Study' dated June 2022, Figure 5.
[195] Paragraph 2.16.
[196] Verian Group UK (2024), Mobile Browsers Qualitative Consumer Research, p10, 16, 17 and 23-25, conducted for the CMA as part of the Mobile Browsers and Cloud Gaming Market Investigation.
[197] Apple response to the CMA's information request [✂].
[198] Apple response to the CMA's information request [✂].

5.58 We are still considering the extent to which innovations in desktop browsers may generate competition in mobile browsers. On the whole, as explained in 'WP1 - Nature of competition in the supply of mobile browsers and browser engines',[199] we consider desktop browsers to only exert a limited constraint on mobile browsers, including from a supply-side perspective, as they need to be tailored for a different use case.

5.59 For other features, including security features or performance improvements, Apple has not to date provided evidence that such features were developed to make sure WebKit offered similar or superior features to Blink, or Safari offered similar or superior features to Chrome, so that users would keep buying Apple devices.

5.60 When asked about the extent of competition between browsers across ecosystems, Google stated that it thinks it is probably [✂].[200]

Evidence of Apple's investment in WebKit

5.61 Evidence of Apple's level of investment in WebKit (particularly on iOS) may also give an indication of the extent of competition between browsers on iOS and Android. While we have requested evidence from the three main browser engine providers, we have not been able to obtain data that would allow for a useful comparison between their respective levels of investment in their browser engines.

5.62 Some Apple internal documents suggest that [✂].[201]

Summary of emerging thinking on Apple's justification for the WebKit restriction

5.63 Apple has argued that the WebKit restriction, together with other iOS restrictions on the sideloading of apps, is needed to ensure the high standards of privacy, security, and performance on iOS devices, which in turn drives competition between iOS and Android devices.

5.64 The WebKit restriction potentially provides some benefits for the security, privacy, and performance of iOS devices for example, because of the integration between WebKit and the operating system, the centralised control that Apple has over WebKit, and by ensuring all browsers provide a baseline level of security, privacy, and performance.

5.65 However, the Webkit restriction also limits rival mobile browser vendors from innovating and improving their browsers on iOS, including on security, privacy, and

[199] Paragraphs 3.28-3.40.
[200] Note of meeting with Google [✂].
[201] Apple internal documents [✂].

performance (as explained in section 3 above) above this baseline. This prevents browser vendors from achieving potentially higher levels of security or performance than would be possible absent the restriction, or from further differentiating on privacy protections for users, as all browsers are largely restricted to the levels determined by WebKit.

5.66 It is unlikely that any benefits arising from the WebKit restriction lead to greater competition between the iOS and Android ecosystems, given the limited evidence of security, privacy and performance driving consumer's choice of mobile device.

5.67 Further, there appears to be a weak link between competition between ecosystems and competition between browsers (which is the market affected by the WebKit restriction) on the basis that mobile browsers are do not appear to be a key factor driving users' choice of device.

5.68 It is therefore our emerging view that the WebKit restriction is unlikely to have a positive impact on competition in browsers on iOS that would offset the negative impacts on competition described in section 3.

6. Emerging thinking on the WebKit restriction

6.1 The requirement that all browsers on the iOS operating system use a specific version of the WebKit browser engine controlled by Apple, means that there is no competition between browser engines on the platform. Browser vendors cannot switch to an alternative browser engine or make changes to the version of WebKit used on iOS. Similarly, consumers are unable to switch to a browser based on an alternative browser engine. We consider that the lack of competitive pressure is likely to reduce Apple's incentives to improve WebKit.

6.2 The evidence shows that the WebKit restriction has significant implications for browser vendors. Given the importance of the browser engine to a browser's features and performance, the inability to use an alternative browser engine limits the ability of browser vendors to innovate and improve their browsers on iOS. Browser vendors are less able to add features and improvements to their browsers on important parameters such as security, privacy, performance, and innovations on iOS relative to less restricted platforms such as Android or desktop. This reduces the features available to consumers and limits effective competition on these parameters.

6.3 In addition, we have obtained evidence indicating that browser vendors incur additional costs from having to develop and support a version of their browser based on WebKit, which they would not do if the restriction were not in place. Evidence from browser vendors also indicates that Apple is difficult to engage with regarding requests for fixes or the addition of new features to WebKit on iOS.

6.4 The implications for web developers are less clear cut, and this may be a result of their different priorities. The main concern of web developers with respect to browsers is ensuring that the websites and web apps that they develop are compatible with the most commonly used browsers. Evidence indicates that this has become easier in the last five to ten years as more tools have become available, and major browsers have become more similar. Web developers therefore often had little concern and even awareness of the WebKit restriction.

6.5 However, there was some evidence from web developers that WebKit was slower to support new features, particularly in relation to web apps. This may be because a lack of competition between browser engines on iOS has reduced Apple's incentives to keep pace with other browser engines. There is some evidence that this has held back web development more broadly as some developers have avoided using features that are not supported by WebKit. This is a particular concern for developers interested in more innovative features such as those for web apps.

6.6 Whilst Apple has submitted that the WebKit restriction is necessary to ensure the security, privacy, and performance of iOS devices, and that this is an important

aspect of competition between iOS and Android devices, the evidence we have seen to date does not support this conclusion. In particular, while it is likely true that requiring all browsers on iOS to use WebKit guarantees that all browsers meet a baseline level of security, privacy, and performance in a way that can be controlled by Apple, the WebKit restriction also limits browser vendors from innovating and offering competitive features to users beyond this baseline level.

6.7 In any event, consumer evidence indicates that mobile browsers and the features available within them play a very limited role in consumers' choice when purchasing a mobile device – and therefore Apple's restrictions are unlikely to contribute materially to greater competition between browsers or across ecosystems.