# MOBILE BROWSERS AND CLOUD GAMING

## Appendix D: Remedies not taken forward in this market investigation

12 March 2025

CMA
Competition & Markets Authority

*Website: www.gov.uk/cma*

# Remedies not taken forward in this market investigation

## Introduction

1. This appendix describes the potential remedies which we have considered during this market investigation but which we have decided not to take forward. They are summarised in Table 1 below.

2. This appendix should be read in conjunction with Section 11 which sets out the remedy we have chosen to adopt following this market investigation, namely to recommend to the CMA Board that, if it decides to designate Apple and/or Google with Strategic Market Status (SMS) in their respective digital activities in mobile ecosystems as a result of the SMS investigations opened on 23 January 2025, the CMA Board should consider imposing appropriate interventions, such as those we have considered in this Appendix.

3. Our reasoning for deciding to make a recommendation, rather than using our remedy-making powers under the Enterprise Act 2002 (EA02), is set out in that section. In summary, while we consider that the potential remedies we have considered are, in principle, capable of addressing certain features we have identified as restricting competition, we have identified a number of significant risks to their effectiveness if taken forward under those powers.

**Table 1: Summary of potential remedies we have decided not to take forward**

| | Description of the potential remedy | Relevant AEC |
|---|---|---|
| **Potential remedy 1** | A requirement for Apple to allow use of alternative browser engines on iOS with access granted to iOS to browser vendors using alternative browser engines on equivalent terms to that made available to WebKit, Safari or third-party applications. | AEC1 (the supply of mobile browser engines on iOS)<br><br>AEC2 (the supply of mobile browsers on iOS) |
| **Potential remedy 2** | An interoperability requirement mandating Apple to: (i) grant equivalent access to functionality used by Safari to browser vendors using the version of the WebKit engine as specified by Apple on iOS; and (ii) grant such access within a reasonable timeframe. | AEC2 (the supply of mobile browsers on iOS) |
| **Potential remedy 3** | Remedy 3a: A requirement for Apple to: (i) allow native app developers on iOS in the UK to use their choice of browser engine for in-app browsing within their native app (a 'bundled engine'); and (ii) provide interoperability with bundled engines for in-app browsing ('bundled engine IAB') | AEC1 (the supply of mobile browser engines on iOS)<br><br>AEC2 (the supply of mobile browsers on iOS)<br><br>AEC3 (the supply of in-app browsing technology on iOS) |
| | Remedy 3b: A requirement for Apple to allow sufficient cross-app functionality to enable native apps to invoke third-party browsers (regardless of the browser engine they use) to support in-app browsing. | |
| **Potential remedy 4** | Prohibition of the Chrome Revenue Share. | AEC2 (the supply of mobile browsers on iOS) |
| **Potential remedy 5** | 5a - A requirement for Apple to ensure the use of a browser choice screen at device set-up. | AEC 2 (the supply of mobile browsers on iOS) |
| | 5b - A requirement for Apple to ensure the placement of a default browser selected by the user in the 'application dock'/'hotseat'[1] or on the default home screen[2] at device set-up. | |
| | 5c - A requirement for Apple to ensure the use of a browser choice screen after device set-up. | |
| | 5d - A requirement for Apple to share user data on default browser settings with browser vendors. | |
| | 5e - A requirement for Apple to ensure that the frequency of default browser prompts and notifications is limited across multiple access points. | |
| **Potential remedy 6** | 6a - A requirement for Google to ensure the use of a browser choice screen at device set-up. | AEC4 (the supply of mobile browsers on Android) |
| | 6b - A requirement for Google to ensure the placement of a default browser selected by the user in the 'dock'/'hotseat' or on the default home screen at device set-up. | |
| | 6c - A requirement for Google to ensure the use of a browser choice screen after device set-up. | |
| | 6d - A requirement for Google to ensure that the frequency of default browser prompts and notifications is limited across multiple access points. | |

*Source: CMA*

4. In our discussion of each potential remedy below, we set out:

[1] The 'hotseat' or 'application dock' position refers to the positioning centrally in the row of apps placed at the bottom of the home screen. Apps located in the 'hot seat' remain visible even when the user moves away from their default home screen to another screen on their device. This is explained in Section 8: The role of choice architecture in mobile browsers.
[2] The 'default home screen' refers to the initial screen that the user sees when unlocking their device.

(a) A description of the potential remedy.

(b) How the potential remedy would seek to address the AECs and any associated customer detriment.

(c) Key remedy design considerations.

(d) Conclusions on that potential remedy, including as to the effectiveness risks that would arise if the potential remedy was to be implemented through the remedy-making provisions of the EA02.

5. Our consideration of these potential remedies takes account of the recent entry into force of the DMCC Act and other key developments internationally. In particular, the other key legislation outside of the UK that applies specifically to the supply of mobile browsers and browser engines is the European Union's Digital Markets Act (DMA), which entered into force in 2022.[3]

6. The DMA establishes a set of criteria to identify and designate 'gatekeeper' firms. Gatekeepers are large digital platforms providing, as defined in the DMA, core platform services which (if they are designated) must comply with the obligations and prohibitions listed in the DMA. Apple and Google are both designated gatekeeper firms for the purposes of the DMA.[4]

7. Articles 5(7), 6(3) and 6(4) of the DMA contain provisions that may apply in connection with the supply of mobile browsers and browser engines.

8. We have considered the policies announced by each of Apple and/or Google to comply with these provisions of the DMA, where they are relevant to the design and/or implementation of the potential remedies referred to in this appendix.

9. During the course of our investigation, we received representations regarding the implications of the potential remedies set out in this appendix, in particular in response to Working Paper 7: Potential remedies (WP7)[5] and the PDR[6].

10. For each of the potential remedies set out below, we have provided a summary of the key submissions which we have received on those potential remedies. However, given that the submissions relate to remedies which we have decided not to take forward in this market investigation, we have not sought to provide an exhaustive list of the submissions made to us, nor have we sought to respond to them in detail.

---

[3] 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.
[4] European Commission, Digital Markets Act - Gatekeepers.
[5] CMA, Working paper 7: Potential remedies.
[6] CMA, Mobile browsers and cloud gaming Provisional decision report dated 22 November 2024.

# Potential remedy 1 to address AECs 1 and 2

## Description of potential remedy 1

11. A potential remedy to address AECs 1 and 2 would be to require Apple to:

    (a) allow the use of alternative browser engines on iOS – by removing current clause 2.5.6 from Apple's App Review Guidelines, which requires third-party browsers to use WebKit, and refraining from introducing any guidelines with similar effect in the future; and

    (b) provide 'equivalent access' to iOS as that which WebKit, Safari or third-party applications have to iOS on fair, reasonable and non-discriminatory (FRAND) terms to browser vendors choosing to use browser engines other than WebKit ('alternative browser engines').

12. High-level parameters that could be used to assess equivalence of access to functionality include:

    (a) enabling access in a way which respects the technical architecture of alternative browser engines;

    (b) enabling access to all of the current operating system-level features and functionalities that WebKit and Safari currently use;

    (c) enabling access to all other current operating system-level features and functionalities that exist on iOS and are available for use by third-party applications, but which WebKit and Safari currently do not use[7];

    (d) enabling access to future operating system-level features and functionalities available to WebKit, Safari, or third-party applications, whether or not WebKit and Safari choose to use them;

    (e) enabling access to the required iOS functionality to allow browser vendors using alternative browser engines to install and manage progressive web apps (PWAs)[8] using alternative browser engines[9]; and

[7] As noted in Section 4: The requirement to use Apple's WebKit browser engine on iOS, WebKit currently does not support a number of features that are important to browsers, for example WebBluetooth. The underlying functionalities necessary to implement such features are generally available to non-browser apps. As part of this potential remedy, we would expect these underlying functionalities to be made available to browser vendors choosing to use alternative browser engines.
[8] Apple often refers to PWAs as Home Screen Web Apps (HSWAs).
[9] Browser vendors' ability to install and manage PWAs using alternative browser engines would require Apple to provide browser vendors with the ability to configure PWA install prompts.

(f) enabling access to the required functionality to allow browser vendors using alternative browser engines to check whether their mobile browser has been set as default.

13. This potential remedy would enable browser vendors to incorporate alternative browser engines, such as Gecko or Blink, as well as variations of Apple's WebKit, into their mobile browser; and would involve Apple taking the requisite steps to provide equivalent functionality to them for the purpose of using an alternative browser engine.

14. Potential remedy 1 would directly benefit browser vendors wishing to use alternative browser engines but would also indirectly benefit browser engine providers. As noted in Section 2: Nature of competition in mobile browsers, browser engines and in-app browsing, mobile browser engine providers also typically supply mobile browsers. There are numerous mobile browsers, compared to just three widely used mobile browser engines. Therefore, addressing the remedy at browser vendors would ensure that the remedy is applied widely.

15. Additionally, since browser engines are open-source, browser vendors may choose to make changes to the engine being used to differentiate their browser from others.

16. We do not expect that, in order to implement this potential remedy, Apple would be required to degrade any currently available functionality made available to WebKit and Safari.

## How potential remedy 1 would seek to address the AECs and customer detriment

17. As set out in Section 10: Decisions on AEC(s) in the supply of mobile browsers, browser engines and in-app browsing, we have found that the WebKit restriction is a feature which, individually or in combination with other features, gives rise to AECs in the markets for:

(a) mobile browser engines on iOS, and

(b) mobile browsers on iOS.

18. The aim of potential remedy 1 would be to allow alternative browser engines to enter and compete in the relevant market, providing browser vendors with greater choice and enabling greater diversity of features and functionalities for the benefit of users by placing greater competitive pressure on Apple to improve WebKit. In doing so, this potential remedy would also enable greater competition between mobile browsers on iOS, by allowing browser vendors to choose a browser engine

to best meet their needs in terms of implementing features and improvements in their mobile browsers and reducing their overall costs. In particular:

(a) alternative browser engines would be able to compete with WebKit by offering functionality to browser vendors which may not be present in WebKit;

(b) browser vendors would have a choice of browser engines they can use on iOS – a choice which currently does not exist;

(c) browser vendors would be able to innovate and offer mobile browser features to iOS users which are currently either not available or restricted – in turn improving the browser experience for iOS users; and

(d) browser engines and browser vendors would be able to compete by offering features and functionalities to web developers such as those important to web apps. This would allow developers to make greater use of web apps which could be a lower cost alternative to native apps for developers which in turn could benefit consumers in the form of higher quality apps or lower prices.[10]

19. However, we consider that there are a number of risks to the effectiveness of this potential remedy if implemented through the remedy-making provisions of the EA02. Taken together, they amount to a significant risk to the effectiveness of this potential remedy in addressing the AECs and resulting customer detriment. We set this out in further detail in the section below.

## Key remedy design considerations

20. We set out below an assessment of whether potential remedy 1 would be effective and the key remedy design considerations that would be relevant in this respect. In particular, an effective remedy would require:

1. adequate specification of what equivalence of access to functionality would need to encompass; including objective criteria to assess and monitor compliance by Apple;

2. a mechanism for assessing the terms and conditions imposed by Apple on parties seeking to use (or applying for an entitlement to use) alternative browser engines; and

[10] The CMA's MEMS report noted that the main advantage to web developers of developing web apps rather than native apps is that a developer only has to develop one app for all operating systems. Additionally, web app support can lead to savings for developers which may be passed on to consumers in the form of higher quality apps or lower prices, (see paragraph 7.25 of the MEMS report).

3. a clear process for third-party browser vendors to request access to functionality and a mechanism for resolving disputes between Apple and browser vendors should these arise.

## 1. Specification of equivalent access to functionality and associated criteria for assessing and measuring compliance

### Criteria for determining equivalent access to functionality

21. In order to be effective, any requirement on Apple to provide 'equivalent access' to browser vendors would need to be clearly specified.

22. A remedy of this nature would require specific criteria to be established for determining equivalent access to functionality, in order to manage the risk of differing interpretations and to mitigate possible circumvention risks.

23. We have set out above in Description of potential remedy 1 sub-section how equivalence could be determined through objective criteria.

24. For example, 'equivalence of access' would need to include enabling third-party browsers using alternative browser engines to install and manage PWAs (rather than relying on WebKit to support parts of this process), including enabling mobile browsers using alternative browser engines to implement installation prompts for PWAs. This is a point that multiple stakeholders[11] and individual web developers[12] have noted as a key parameter for competition between browser engines and between mobile browsers.

25. Apple would not be expected to develop significant PWA functionality that is not already made available by its operating system (iOS). WebKit and iOS currently support PWAs, albeit to a more limited extent than is available on other platforms.

26. However, we note there would be an information asymmetry between Apple and third parties relating to iOS architecture and considerable integration of both WebKit and Safari with the operating system. A high-level obligation on Apple without specified criteria for measuring equivalence would place a high burden on browser vendors seeking to establish whether certain functionality is available; and would be challenging to monitor. On the other hand, where more specific criteria are established for measuring whether equivalent access to functionality has been provided, these may need to be iterated over time.

[11] Responses to the CMA's information requests: [✂].
[12] Summary of Individual Responses to WP7 Submitted to the CMA, 15 November 2024.

**Specific requirements for Apple to demonstrate equivalent access**

27. A separate consideration is the technical method by which Apple provides access to functionality. Two different ways of achieving such access could be (i) to leave it open to Apple to choose to create new APIs for third parties, replicating the functionalities and features made available to WebKit and Safari; or (ii) Apple could give access to existing private APIs that exist as internal interfaces within iOS. We note that due to the significant integration that exists at present between WebKit, Safari and iOS, extending existing APIs[13] to third-party browsers may be insufficient to achieve equivalent access to functionality.

28. It is our view that Apple should be able to determine the way in which technical access to its operating system is made available to third-party browser vendors under potential remedy 1. Apple's dual role as the device manufacturer and operating system provider means it is best placed to determine how the required level of access can be granted to third parties considering any security and privacy considerations that need to be incorporated.

29. We received submissions from various parties on the significance of establishing what level of access would be required and how it should be specified to make the remedy effective in response to WP7:

    (a) Apple submitted that it found it difficult to determine what the CMA meant by 'equivalent access', both in terms of access to iOS and access to APIs.[14] Further, Apple questioned whether it would be required under this remedy to make changes to WebKit to enable functionality that Safari does not have in order to address 'unsubstantiated' complaints from third-party browser vendors, noting that this would be disproportionate.[15]

    (b) Mozilla submitted the importance of achieving the specification of access to technical functionality at the right level.[16] Similarly, Google submitted that uncertainty over what constitutes 'equivalent' features and functionalities risks undermining the potential remedy's effectiveness, but the risk could be mitigated through established mechanisms such as public scrutiny.[17]

    (c) Google further submitted that a remedy enabling use of alternative browser engines on iOS should prohibit policies or technical limits which restrict browsers from accessing APIs available to other non-browser apps (even if

[13] Existing APIs would include both public and private APIs that would be required by third-party browser vendors to be able to use alternative browser engines.
[14] Apple's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 30.
[15] Apple's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 30.
[16] Mozilla's response to Working Paper 7: Potential Remedies, 8 August 2024, page 3.
[17] Google's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 10.

not used by Safari), if browsers routinely access and use such APIs on other platforms.[18]

(d) A large app developer submitted that it would be important that the remedy not only provides access to APIs used by WebKit and Safari (to establish a level playing field), but also that those APIs that are made available allow developers to innovate how they deliver browsing experiences on iOS using alternative browser engines. The developer submitted that those APIs should provide the means, but not the ends, of delivering browsing experiences on iOS using alternative browser engines.[19]

(e) Mozilla submitted that with high-level remedies, the burden tends to fall on the challenger firm seeking to provide choice and competition to demonstrate why a particular proposal from dominant platforms is unworkable.[20]

30. In response to the PDR, we received a range of views on potential remedy 1, in relation to the significance of establishing what level of access would be required and how it should be specified, in particular:

(a) Apple submitted specifically in relation to potential remedy 1 that:

(i) The potential remedy would require enabling access in a way which respects the technical architecture of alternative browser engines. Apple does not know the architecture of third-party browser engines, what their technical architecture may require or if it would be possible to support them.[21]

(ii) Apple already provides extensive documentation, and WebKit is open source and therefore it is unclear what is insufficient about its current documentation and what further would be required.[22]

(b) Apple submitted in relation to potential remedies 1 and 2 that:

(i) The requirements to provide 'equal access' to iOS and WebKit, respectively are extraordinarily broad, with no limiting principle contemplated beyond a "vague" exception for 'significant' new PWA functionalities and integration of first-party services (under potential remedy 1). Apple submitted both potential remedies are disproportionate because they do not account for third parties already having the ability to build most material functionality.[23]

[18] Google's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 10.
[19] [✂] response to Working Paper 7: Potential Remedies, [✂].
[20] Mozilla's response to CMA's Working Paper 7: Potential Remedies, 8 August 2024, page 3.
[21] Apple's response to the CMA's provisional decision report dated 22 November, paragraph 191.
[22] Apple's response to the CMA's provisional decision report dated 22 November, paragraph 191.
[23] Apple's response to the CMA's provisional decision report dated 22 November, paragraph 191.

(ii) As currently described, the potential remedies are an open-ended obligation on Apple to make substantial ongoing investments to develop every feature demanded by third parties and who are unwilling to invest in themselves. Such specification of the requirement, Apple submitted, limits Apple's ability to legitimately commercialise its platform and recoup its development and maintenance costs of new features offered to third parties.[24]

(iii) The specification of potential remedies 1 and 2 would cause market distortions through free-riding and underinvestment from third parties and by adversely impacting Apple's incentives to invest in development of new features and technologies, particularly a requirement that results in Apple not being able to recoup the development costs of new technologies.[25]

(c) In relation to the geographic scope of potential remedies 1, 2 and 3 Apple submitted that:

(i) There is no basis for remedies to extend beyond the UK for them to be effective.

(ii) It would be disproportionate to require fundamental changes to the iOS architecture on a worldwide basis to address UK-specific concerns.

(iii) The extra-territorial application of remedies would also impose the CMA's views on markets outside the UK.

(iv) Well-established principles of comity would argue strongly against such an approach.[26]

31. Movement for an Open Web (MOW) submitted that potential remedy 1 can be future-proofed, and therefore its effectiveness increased if the definition of browser functionality was to be tied to W3C standards.[27]

32. Mozilla submitted it supports the introduction of potential remedy 1.

(a) It noted that a provision which prohibits Apple from introducing guidelines with a similar effect to the existing guideline 2.5.6 would be important to ensure effectiveness of the remedy.

[24] Apple's response to the CMA's provisional decision report dated 22 November, paragraph 191.
[25] Apple's response to the CMA's provisional decision report dated 22 November, paragraph 191.
[26] Apple's response to the CMA's provisional decision report dated 22 November 2024, pages 36 – 37.
[27] Movement for an Open Web (MOW)'s response to the CMA's provisional decision report dated 22 November 2024, page 8.

(b) It is important that the remedy allows browser vendors to use alternative versions of WebKit to that developed by Apple, in addition to using alternative browser engines.

(c) It broadly agrees with the high-level parameters that could be used to assess equivalence of access to functionality and proposes that the detail of how these parameters might be applied in practice could be provided through guidance which should not be considered as an exhaustive list of what 'equivalence of access' means.

(d) In relation to allowing Apple to decide how it will provide access to iOS and WebKit for browsers using alternative browser engines, Mozilla submitted that should Apple decide to create new APIs for third parties, the CMA would be required to monitor the timeliness of the provision of these. In Mozilla's view, there should be a clear list of APIs used by WebKit and Safari (including private and public) and a separate list of the APIs that are made available to third-party browsers.[28,29]

(e) Mozilla submitted that Apple should offer explanation of what 'affordances' Apple has made/will make for third parties using alternative browser engines and the estimated timing for implementing those 'affordances'.[30]

(f) Where access has been withheld in relation to an API, Mozilla submitted that Apple should set out the specific reasons for withholding access to any APIs or functionality which is available to Safari.[31]

33. Open Web Advocacy (OWA) submitted that removing the WebKit restriction through potential remedy 1 and potential remedy 2 would be both proportionate and effective.[32]

34. An effective remedy would also require that the terms on which Apple provides access to iOS for third parties are also equivalent to that of Safari. For example:

(a) Documentation or guidance provided by Apple on APIs would need to be clear, complete and up to date to ensure browser vendors' ability to make effective use of the APIs.

(b) Service-level support for third-party browsers should be available which is equivalent to that provided for WebKit and Safari. In particular, this should involve Apple providing a complete set of up-to-date APIs (and any other technical updates and implementations to its operating system) in a timely

[28] Mozilla's second response to the CMA's provisional decision report dated 22 November 2024, pages 2—3.
[29] Mozilla's second response to the CMA's provisional decision report dated 22 November 2024, page 4.
[30] Mozilla's second response to the CMA's provisional decision report dated 22 November 2024, page 4.
[31] Mozilla's second response to the CMA's provisional decision report dated 22 November 2024, page 4.
[32] Open Web Advocacy (OWA)'s response to the CMA's provisional decision report dated 22 November 2024, page 10.

manner; in a way which enables third-party browser vendors to implement relevant functionalities and features fully. Overall, this support should ensure that browser vendors are not delayed in implementing the desired features or when using the available functionalities.

(c) Browser vendors using alternative browser engines on iOS would need to be given access to the range of iOS and device metrics[33] that are available to Apple for assessing Safari's performance, to enable browser vendors to measure and assess the performance of their own respective browsers on iOS (for example data to facilitate debugging of the browser app or monitor its stability).

**2. The terms and conditions (or application criteria) that Apple may impose on browser vendors using alternative browser engines**

35. A further area of effectiveness risk may arise from any terms and conditions imposed by Apple on parties seeking to use (or applying for an entitlement to use) alternative browser engines. Such terms and conditions could undermine the viability of using such alternative engines and introduce circumvention risk.

36. We note that multiple stakeholders submitted that the terms Apple has attached to its proposed Web Browser Engine Entitlement (WBEE), which it has introduced in response to the DMA obligations, have precluded them from considering using alternative browser engines on iOS in the EU[34].

37. Particular concerns could arise if Apple were to introduce conditions such as:

(a) requiring users of mobile browsers which use alternative browser engines to uninstall their existing mobile browser and install a new version of the app, creating potential friction or confusion (the 'separate binary' requirement); and

(b) Apple imposing terms on browser vendors on the location of where testing and development of mobile browser apps using alternative browser engines should take place (for example, that testing and development of a UK browser app using an alternative browser engine should be done in the UK only).

(c) Disproportionate security and privacy considerations.

[33] Metrics for assessing performance include access to telemetry APIs, which enable measurement and transmission of data on application performance, health and security.
[34] Responses to the CMA's information requests: [✂].

**Separate binary requirement**

38. In the EU, Apple requires browser vendors choosing an alternative browser engine to do so by providing a separate app to that which currently uses WebKit (referred to in Apple's documentation as a 'separate binary').

39. In this context, we note that alternative options may be available. For example, a large app developer submitted that browser vendors using alternative browser engines could have region-specific binaries enabling the browser vendor to retain a single App Store entry and feature updates.[35] This suggests that there are ways to limit the use of alternative browser engines to a specific geographic location, making possible risks avoidable or manageable.

40. Additionally, Mozilla submitted that there are some user-related as well as financial implications of imposing a separate binary requirement, which would negatively impact the effectiveness of the remedy:

    (a) When deploying an application that is as critical as a web browser, Mozilla relies on its A/B testing[36] infrastructure to ensure the quality of its product and rolls out major changes in stages to a representative set of users. Apple's separate binary requirement implies that any transition must be a hard switchover with no way of going back. This is unnecessarily disruptive to users and risky for browser developers.

    (b) Browsers based on alternative browser engines could encounter problems migrating users, leading to unnecessary friction and confusion for users.

    (c) There may be breakages and compatibility issues when browsing Gecko on iOS (due to web developers currently only needing to make their websites compatible with WebKit on iOS) at the outset and having the ability to fall back to WebKit could potentially help to mitigate this issue.

    (d) Browser vendors would be forced to maintain two versions of their app for the UK, leading to increased development costs.[37]

41. The separate binary requirement appears unduly onerous, and the evidence indicates that there are alternative means of allowing browser vendors to use alternative browser engines on iOS.

42. In response to the PDR, Apple submitted the following in relation to its separate binary requirement under its compliance with the DMA:

---

[35] Note of meeting with [✂].
[36] A/B testing involves comparing two different versions of a design to see which performs better. It assists in developing an understanding of how the differences between the two versions affect users' behaviour and outcomes. A/B testing: comparative studies - GOV.UK.
[37] Mozilla submission to CMA [✂]

(a) Separate binaries are commonly used to address different technical or regulatory requirements and do not impose an undue burden on developers.

(b) Separate binaries are necessary to ensure that a remedy imposed in the UK would not deliberately or inadvertently be used in other regions by developers choosing to implement the features globally.

(c) Separate binaries ensure that users are informed of the change in engine.[38]

43. Further, we consider that any requirement imposed on browser vendors as regards the location of where testing and development of mobile browser apps using alternative browser engines should take place would undermine the effectiveness of this potential remedy and does not appear to be necessary.

44. On the other hand, as set out below, we do consider that it may be necessary for Apple to impose appropriate security and privacy requirements on browser vendors choosing to use alternative browser engines on iOS. Further, Apple should be able to amend such requirements to ensure they remain up-to-date and reflect the latest security threats.

**Security and privacy considerations**

45. Apple made a number of submissions in response to WP7 in relation to the potential adverse security and privacy implications of a remedy requiring it to allow use of alternative browser engines on iOS:

(a) Apple submitted that no requirements Apple could impose on browser developers (or browser engine developers) would be sufficient to fully mitigate the harms that would arise from removal of the WebKit requirement;[39]

(b) Apple further submitted that there is a residual risk from allowing alternative browser engines to meet Apple's security requirements on iOS – though, in describing such residual risk, Apple acknowledged that every browser engine has vulnerabilities, including WebKit;[40]

(c) Apple objected to the CMA specifying security requirements that Apple would be entitled to impose on browser vendors and browser engine vendors. Apple submitted that such specification would be inappropriate and unworkable;[41]

[38] Apple's response to the CMA's provisional decision report dated 22 November 2024, page 39.
[39] Apple's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 40.
[40] Note of meeting with Apple, [✂].
[41] Apple's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 41.

(d) Apple submitted that setting static security requirements would create very significant risks for Apple, developers, and users, and that Apple should be allowed to determine what security requirements should be deployed in response to threats as Apple sees them;[42] and

(e) Apple submitted there would be a security risk from allowing a browser app to migrate to use an alternative browser engine without a separate binary requirement. Apple explained that the alternative browser engine would then be present in the browser app binary worldwide, and Apple would have no means of controlling that the engine code would not be executed outside of the jurisdiction where the remedy was being imposed. Apple explained this means an attacker could potentially execute that code to access low-level capabilities of the system.[43]

46. In response to the PDR, Apple made the following security and privacy submissions:

(a) Apple welcomed the acknowledgement that Apple should be entitled to set out minimum security and privacy requirements for the introduction of third-party browser engines.[44]

(b) Apple submitted that it should be given sufficient leeway to determine exactly what third-party developers must demonstrate in terms of capability, intention and accountability before they can be allowed to offer alternative browser engines or use them in their apps on iOS.[45]

(c) Apple submitted it had concerns based on the findings of the DSIT survey showing very few app developers in the UK were aware of the voluntary code of practice on mobile app security and privacy for app developers, app store operators and platform developers, introduced in 2022.[46]

(d) On PWAs, Apple submitted that there are three layers of risks associated with PWAs:

(i) the underlying browser engine (this is the same for both PWAs and websites that the user is browsing);

(ii) specific security risks involved with the complexity of the solutions required to make PWAs work; and

[42] Apple's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 41.
[43] Apple, Main Party Hearing summary note, 18 September 2024, paragraphs 43-44.
[44] Apple's response to the CMA's provisional decision report dated 22 November 2024, paragraph 199.
[45] Apple's response to the CMA's provisional decision report dated 22 November 2024, paragraph 200.
[46] Apple's response to the CMA's provisional decision report dated 22 November 2024, paragraph 73.

(iii) user-facing risks, eg trust and safety risks (whether an app is real or malicious).[47]

47. Apple submitted that due to architectural challenges there is "no simple switch" to enable PWAs to run using alternative engines on iOS, and making changes to enable such a feature would create security risks for iOS.[48]

48. Apple noted that it [✂] all possible security issues [✂] given the complexity of the issue and the fact [✂]. Nevertheless, Apple stated that bringing this feature to iOS can, in theory, be done but that the risk would [✂] and create a security risk for all users globally, [✂].[49]

49. Google submitted that exposing access to the same lower-level iOS features that Safari and WebKit have to third-party browser vendors may create security risks that do not exist if these are only used by WebKit and Safari. To adequately mitigate this risk, Google submitted that Apple may need to enable equivalent or indirect access to such features by exposing alternative APIs in a safe and secure manner.[50]

50. One browser vendor [✂] submitted that browser engine choice on a mobile platform can facilitate greater competition on security, privacy, and performance between mobile browsers and between browser engines.[51]

51. The UK National Cyber Security Centre (NCSC) submitted that a vendor which produces both the operating system and the browser is potentially able to offer better security as it is able to modify the operating system, sandbox, and browser to provide the best overall security, for example by moving components into or out of the browser or modifying the interactions the operating system allows – whereas a third-party browser vendor can only modify its browser and the sandbox profile that it requests.[52] NCSC further submitted that:[53]

(a) costs of enabling use of alternative browser engines would likely fall on an operating system vendor which would need to document or modify its operating system security features such as a sandbox, as well as parties looking to make best use of those security features;

(b) the overall security of a product on a platform depends on the vendor of the product, and not all vendors will necessarily be willing or able to provide high

[47] Note of meeting with Apple, [✂].; Apple's response to the CMA's provisional decision report dated 22 November 2024 paragraph 201.
[48] Note of meeting with Apple, [✂].
[49] Note of meeting with Apple, [✂].
[50] Google's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 15.
[51] [✂] response to the CMA's provisional decision report [✂].
[52] NCSC, submission to the CMA [✂].
[53] NCSC, submission to the CMA [✂].

levels of security through a mixture of their own controls and making best use of the platform's security features;

(c) vendors of a product may need to work with, or be supported by the platform vendor, to get the maximum security benefits from the platform's features; and

(d) the technical challenge is in allowing browser engines in such a way that they benefit fully from protections within the operating system. This is likely to require re-work by either the browser vendor or operating system vendor. There is potential for difficulty in exposing the underlying operating system components, particularly with regards to properly sandboxing the engine to the same standard.[54]

52. The NCSC also submitted that:

(a) the length of gap between a vulnerability being known and a patch being issued presents opportunity for attackers, and so it is important that vendors promptly issue updates. Browser vendors using engines they have not created, or vendors who have not sufficiently prioritised security, may take longer to issue updates;[55] and

(b) browser and platform vendors with greater knowledge, maturity and resources are likely to be more capable of building features securely.[56]

53. RET2 submitted that only those browser vendors that are best equipped to manage and operate an alternative engine appropriately be allowed to use third-party engines and permitted special platform rights, and RET2 expected Apple to set security and privacy requirements.[57]

54. Regarding PWA security, the NCSC submitted that a PWA is unlikely to pose more risk to a device than visiting the website of the organisation producing the PWA in the relevant browser. NCSC further submitted that any risks and mitigations around PWAs would need to be around the browser engines themselves and aspects such as sandboxing that run these applications.[58,59]

55. In relation to the potential security risks of enabling alternative browser engines to support PWAs, we consider that many of the possible risks that PWAs carry can

[54] NCSC's response to the CMA's information request [✂]
[55] NCSC, submission to the CMA [✂].
[56] NCSC, submission to the CMA [✂].
[57] RET2's advice to the CMA, [✂]. RET2 Systems Inc. is a computer security consulting firm that was commissioned by the CMA in 2022 to give expert technological advice to as part of the Mobile Ecosystems Market Study.
[58] NCSC, submission to the CMA [✂].
[59] The NCSC noted that this submission is not based on knowledge of any specific platform, and as such does not address issues posed by potential architectural changes that might be required to implement PWA interoperability with alternative browser engines on iOS in particular.

be adequately mitigated through security and privacy requirements that Apple could impose on browser vendors choosing an alternative browser engine on iOS.

56. We acknowledge that Apple has raised concerns in relation to greater fragmentation that could be created by allowing browsers to incorporate their own browser engine leading to browsers using outdated or insecure engines, creating a security risk.[60] However, in relation to this issue, we note that such risk is not unique to mobile browsers and can be managed.[61] We expect that some security requirements would specifically address the issue of fragmentation.

57. Overall, we consider that security and privacy requirements would be a necessary mitigation to ensure that potential security risks which can arise from mobile browsers, and multiple browser engines, could be actively managed and addressed. We consider that the best approach to managing such security risks would be to enable Apple to impose minimum security and privacy requirements. However, such security and privacy requirements would need to be objectively required and proportionate to mitigate the risks highlighted above.

**3. A clear process for third-party browser vendors to request access to functionality; and a method for resolving disputes**

58. In order to be effective, browser vendors would need to have a clearly specified system for requesting access to functionality and for disputes to be resolved in a timely manner, with independent scrutiny.

59. Furthermore, Apple holds a powerful position as the owner of the iOS operating system. There is a risk of information asymmetry as Apple is in an advantageous position regarding its knowledge of its own operating system compared to the CMA and to market participants.

60. In response to the PDR, stakeholders submitted the following in relation to monitoring of the potential remedy 1:

(a) MOW submitted that an oversight and monitoring committee could be put in place to be funded by Apple, a measure implemented in other competition cases. MOW also submitted that a monitoring trustee could be used to carry out periodic review of the remedy and its effects on the market.[62]

(b) Mozilla submitted that for the remedy to be effective the following processes would need to be established as part of the remedy:

[60] Note of meeting with Apple, [✂] Paragraphs 21—22.
[61] Google, response to Working Paper 2: The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine, paragraphs 37 and 41.
[62] Movement for an Open Web's response to the CMA's provisional decision report dated 22 November 2024, page 8.

(i) a mechanism for assessing the terms and conditions that Apple could seek to impose on parties who may apply for entitlements to use an alternative browser engine;

(ii) strict transparency obligations to reduce the burden on the access seekers, the DMU and other stakeholders; and

(iii) a clear process for third-party browser vendors to request access to functionality and a mechanism for resolving disputes between Apple and browser vendors if they arise.[63]

61. The risks associated with information asymmetry could be mitigated through:

(a) a clearly specified process by which third parties interested in developing an alternative browser engine on iOS could engage with Apple, which would be required to provide relevant information in a timely manner to enable the development and deployment of alternative browser engines on iOS;

(b) an independent dispute resolution mechanism enabling browser vendors to raise concerns that Apple is not providing sufficient information or access to enable such third parties to develop or deploy alternative browser engines on iOS and to establish a satisfactory resolution to the concerns in a timely manner; and

(c) a mechanism enabling browser vendors to report any instances in which concerns raised with Apple have not been resolved satisfactorily or within an acceptable time frame.

## Conclusions on potential remedy 1

62. As noted above under the discussion of key remedy design considerations, there are a number of risks to the effectiveness of this remedy if implemented through the remedy-making provisions of the EA02.

63. These relate to:

(a) **Specification**: it would be important to specify clearly what is required from Apple in order that it provides access to iOS at a level equivalent to that obtained by WebKit, Safari and other third-party applications. This is because there is a high risk of circumvention in relation to high level or static requirements. We also note there is an information asymmetry between Apple and other parties in relation to the working of iOS architecture, the

[63] Mozilla's second response to the CMA's provisional decision report dated 22 November 2024, page 3.

availability of functionality and what terms and conditions are necessary for access.

(b) **Circumvention, monitoring and enforcement:** any requirements in connection with this potential remedy would need to be monitored closely on an ongoing basis to ensure that equivalent access, including terms and conditions imposed by Apple on browser vendors seeking to use alternative browser engines do not undermine the effectiveness of the potential remedy.

In general, the implementation of this potential remedy would require ongoing monitoring and oversight and the requirements on Apple may need to be iterated and revised in light of technological developments. As noted above, there would need to be a process for third parties to make access requests and a mechanism for resolving disputes for the duration of the potential remedy.

64. We conclude that, taken together, these risks mean that there is a significant risk to the effectiveness of potential remedy 1 in addressing AECs 1 and 2.

# Potential remedy 2 to address AEC 2

## Description of potential remedy 2

65. A potential remedy to address AEC 2 would require Apple to: (i) grant equivalent access to functionality used by Safari to browser vendors using the version of the WebKit engine as specified by Apple on iOS;[64] and (ii) grant such access within a reasonable timeframe.

66. We consider that browser vendors using WebKit should be able to offer features and functionalities equivalent to those offered by Safari. This would result in browser vendors using the version of WebKit specified by Apple being able to compete on a level-playing field with Safari on iOS.

67. High-level parameters for granting equivalent access to functionality used by Safari to all mobile browsers using the version of WebKit specified by Apple could include:

    (a) enabling access to all WebKit or operating system-level features and functionalities that Safari currently uses, on a request-basis;

    (b) enabling access on fair, reasonable and non-discriminatory (FRAND) terms;

    (c) for any future features and functionalities to be used by Safari, stopping use of private interfaces/APIs (unless required solely for integration with Apple's own first-party services on iOS) and designing future APIs for equivalent access by default; and

    (d) enabling access to all future WebKit or operating system-level features and functionalities that Safari uses free of charge, in a timely manner.

68. We do not expect that, in order to comply with the potential remedy as set out above, Apple would be required to degrade any currently available functionality made available for WebKit and Safari.

## How potential remedy 2 would seek to address the AEC and customer detriment

69. As set out in Section 10: Decisions on AEC(s) in the supply of mobile browsers, browser engines and in-app browsing, we have found that Apple provides greater access to functionality to Safari compared to rivals and that this is a feature which,

[64] Potential remedy 2 addresses potential issues for third-party browser vendors using the version of WebKit specified by Apple on iOS. It would not apply to browser vendors who would use their own version of WebKit as their alternative browser engine under potential remedy 1.

individually or in combination with other features, gives rise to an AEC in the market for mobile browsers on iOS.

70. This potential remedy would support browser competition on iOS by enabling any browser vendors who had decided against using an alternative browser engine to compete with Safari on a level-playing field. This remedy would help address AEC 2 and the resulting customer detriment that may be expected to result, by ensuring that:

(a) browser vendors are able to innovate and offer mobile browser features and functionalities to iOS users which were previously either not available or restricted – in turn improving the browser experience for iOS users;

(b) browser vendors are able to effectively compete with Safari if they choose to use the version of WebKit specified by Apple instead of an alternative browser engine; and

(c) browser vendors can access existing and future features and functionalities available to Safari without incurring costs or unreasonable delays.

71. However, we consider that there are a number of risks to the effectiveness of this potential remedy if implemented through the remedy-making provisions of the EA02. Taken together, they amount to a significant risk to the effectiveness of this potential remedy in addressing the AEC and resulting customer detriment. We set this out in further detail in the section below.

## Key remedy design considerations

72. We set out below an assessment of whether the potential remedy described above would be effective and key considerations that are relevant in this respect. In particular, an effective remedy would require:

1. a clear articulation of what is meant by ‘equivalence’ of access to features and functionality and a set of objective criteria to measure whether Apple is providing equivalent access (which would need to include a requirement for Apple to articulate how it would facilitate and manage access to APIs by third parties);

2. relevant terms and conditions pursuant to which such access would be granted; and

3. a clear process for third-party browser vendors to request access to functionality and a mechanism for resolving disputes.

## 1. Specification of equivalent access to functionality and associated criteria for assessing and measuring compliance

### Criteria for determining equivalent access to functionality

73. An effective remedy would require Apple to provide equivalent access (to that which is provided to Safari by the iOS) to existing features and functionalities on a request basis from individual browser vendors. Similar to potential remedy 1, in order for this potential remedy to be effective, any requirement on Apple to provide 'equivalent access' to third-party browser vendors using the version of WebKit specified by Apple would need to be sufficiently clear and understood.

74. In particular, a potential remedy would need to include specific criteria for determining equivalent access to functionality, in order to manage the risk of differing interpretations and to mitigate possible circumvention risks. We note in particular that there is an information asymmetry between Apple and third parties relating to iOS architecture and the considerable integration of both WebKit and Safari with the operating system. A high-level obligation on Apple without specified criteria for measuring equivalence would place a high burden on browser vendors seeking to establish whether certain functionality is available and would be challenging to monitor.

75. We have set out above in Description of potential remedy 1 sub-section how it may be possible to specify 'equivalence' of access using objective criteria.

76. In response to WP7 stakeholders submitted the following:

(a) Apple submitted that it found it difficult to determine what the CMA meant by 'equivalent access', both in terms of access to iOS and access to APIs.[65]

(b) Mozilla highlighted the importance of achieving the specification of access to technical functionality at the right level.[66]

(c) Similarly, Google submitted that uncertainty over what constitutes 'equivalent' features and functionalities risks undermining the potential remedy's effectiveness, but the risk can be mitigated through established mechanisms such as public scrutiny.[67]

77. In response to the PDR, stakeholders submitted the following in relation to potential remedy 2:

(a) Apple submitted that the existing description of the potential remedy, which would require Apple to provide access to all future WebKit or iOS features

[65] Apple's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 30.
[66] Mozilla's response to Working Paper 7: Potential Remedies, 8 August 2024, page 3.
[67] Google's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 10.

that Safari uses free of charge, does not allow Apple to commercialise its platform.[68,69]

(b) Apple noted that potential remedy 2 would require access to 'any features and functionalities used by Safari by default in a timely manner'. Apple submitted that this could be read as reducing Apple's ability to test new functionality to fix bugs and ensure that security and privacy are not impacted.[70]

(i) Similar to potential remedy 1, Apple submitted that potential remedy 2 as currently described places an open-ended obligation on Apple to develop every feature that third parties wish, which makes the potential remedy disproportionate. Apple submitted that the potential remedy does not take into account that third parties have the ability to build most material functionality.[71]

(c) Mozilla submitted it agrees that potential remedy 2 would support browser competition on iOS provided that concepts of 'equivalent WebKit access' and 'interoperability requirement' are described in enough detail.

(i) Mozilla submitted that it may be best to refer to 'all browsers using the version of WebKit provided by Apple on iOS' rather than 'all Webkit-based browsers', which it considered could lead to confusion.

(ii) Mozilla submitted that similar to potential remedy 1, transparency (around the APIs that exist and those that are made available to third parties) and provision of access in a timely manner are essential.

(iii) Mozilla submitted that where access has been withheld in relation to an API, Apple should set out the specific reasons for withholding access that is made available to Safari.[72]

(d) MOW submitted that it agreed with our suggestion that high level parameters for Apple granting equivalent access to functionality used by Safari should include granting access on FRAND terms.[73]

---

[68] Apple's response to the CMA's provisional decision report dated 22 November 2024, page 36.
[69] We note that Apple made some submissions which concern multiple potential remedies, including potential remedy 2. Those submissions are recorded elsewhere in this section: see sub-section Specific requirements for Apple to demonstrate equivalent access above for Potential remedy 1.
[70] Apple's response to the CMA's provisional decision report dated 22 November 2024, page 36.
[71] Apple's response to the CMA's provisional decision report dated 22 November 2024, page 37.
[72] Mozilla's response to the CMA's provisional decision report dated 22 November 2024, page 4.
[73] Movement for an Open Web's response to the CMA's provisional decision report dated 22 November 2024, page 11.

**Specific requirements for Apple to demonstrate equivalent access**

78. In comparison to potential remedy 1, we consider that the work that would be required by Apple to enable equivalent access for browser vendors using the version of WebKit specified by Apple to be less complex and extensive. Such mobile browsers already use the WebKit engine and the security and privacy protections offered by WebKit to these browsers would continue. The features and functionalities which Apple would need to make available to third-party browser vendors are the same that are available to Safari. Therefore, this would enable Apple to readily determine how the same features and functionalities could be securely extended to other WebKit-based browsers.

79. To enable access to WebKit and operating system-level features and functionalities, Apple already provides third-party developers access to several categories of APIs on iOS:

    (a) public APIs;

    (b) APIs extended under public entitlements; and

    (c) APIs extended under managed entitlements.

80. We envisage that third-party browser access to new APIs created by Apple or existing APIs that would be made public under this potential remedy could be managed through these existing access categories. Therefore, the implementation of access would not be expected to represent a significant technical challenge.

81. However, an obligation on Apple to ensure equivalence of access to future WebKit and/or operating system level features or functionalities that Apple is yet to develop for Safari would require Apple to consider the obligations of this potential remedy in its future design and development processes.

82. This would likely result in Apple ceasing the use of private APIs for this purpose (unless required solely for integration with its own first-party services), documenting all the features and functionalities available to Safari and making them available through entitlements. Browser vendors using the version of WebKit specified by Apple would therefore have access to future features and functionalities by design and would only need to submit requests for access to existing features and/or functionalities.

83. In summary, under this potential remedy browsers using the version of WebKit specified by Apple on iOS would need to be able to make requests to Apple for access to features and functionalities which they do not currently have equivalent access to, and for Apple to provide such access in a timely manner.

84. Equivalent access could be demonstrated through the provision of requirements imposed on Apple relating to the following (similar to those set out in the Specific requirements for Apple to demonstrate equivalent access sub-section for potential remedy 1):

(a) Quality documentation maintained by Apple, which is clear, detailed and kept up to date.[74]

(b) The level of service support made available by Apple to third-party browsers to ensure that access enabled to the operating system allows browser vendors to operate their browsers on equivalent terms to those of WebKit and Safari.

85. Apple would be required to provide equivalent access to the requested features and functionalities in a timely manner. We would expect support (in relation to documenting and providing access) to be provided within the same timescales as the support provided to Safari.

86. In addition to providing access, Apple would be required to ensure third party browsers can customise and configure these features and functionalities to the same level of detail as Safari.[75]

87. Any new APIs created by Apple or existing private APIs that were made public under this potential remedy would need to be documented, kept up to date and maintained to a similar level and standard to APIs used by Safari at no additional cost to browser vendors, and fully supported to ensure ongoing compatibility following any updates to iOS and WebKit.

88. Third-party browser developers would also need to be given equal opportunity to fully test new features and functionalities at the same time as Safari, with access to the same or directly comparable test environments as Safari, including hardware testing.

89. As with potential remedy 1, we do not consider that access to develop and test new features or functionalities should not be limited by any geographical constraints.

90. Apple submitted that it 'opens up access to features and functionalities as widely and quickly as possible, subject to the overriding need to protect the integrity and performance of the platform as a whole.'[76]

---

[74] Quality documentation would need to include, but not be limited to, details of API maintenance, release cycles for updates, communication channels with relevant Apple teams, update and deprecation processes, feature prioritisation criteria and processes for access to new or changed features.
[75] For APIs, this would include the ability to use all its available parameters. API parameters are mandatory or optional settings that can be applied to influence the result of using the API.
[76] Apple response to CMA's Working Paper 7: Potential Remedies, 8 August 2024, paragraph 34.

91. Apple also noted the importance of testing APIs, which it does in part by using its own apps as 'guinea pigs' to judge performance and stability, submitting that APIs must be 'stable, well-tested and long-lived before being released because once released, third-party developers rely on the underlying functionality of the APIs always being there to power their own apps'.[77]

**2. Terms and conditions pursuant to which such access would be granted**

92. We do not consider that Apple should be able to impose any additional terms and conditions on parties accessing or using features and functionalities made available by this potential remedy, beyond the existing App Store terms and conditions relevant to mobile browsers.

**3. A clear process for third-party browser vendors to request access to functionality; and a mechanism for resolving disputes.**

93. A key issue in remedy implementation is the extent to which the CMA is able to effectively monitor compliance with the requirements of the remedy and enforce against any non-compliance. Similar to potential remedy 1, we have identified circumvention and specification risks in this respect. For instance, Apple could document the APIs inaccurately or not respond to requests in a timely manner or provide insufficient support. These risks could be mitigated to an extent through a robust dispute resolution process and enhanced CMA monitoring and enforcement.

94. The design of the potential remedy means that Apple would be able to decide how to technically implement equivalent access to WebKit and iOS for third party browsers using the version of WebKit specified by Apple. However, given that it is the owner of the iOS operating system, Apple has an advantageous position regarding its knowledge of its own operating system. Allowing Apple to determine how to implement access to features and functionalities currently not made available to other browsers using the version of WebKit specified by Apple would give rise to circumvention risk due to information asymmetry.

95. This risk could be mitigated through a similar process as set out in the A clear process for third party browser vendors to request access to functionality; and a method for resolving disputes sub-section above for potential remedy 1.

[77] Apple's response to MEMS Annex D, paragraph 103.2.

## Conclusions on potential remedy 2

96. As noted above under the discussion of key remedy design considerations, there are a number of risks to the effectiveness of this remedy if implemented through the remedy-making provisions of the EA02.

97. These relate to:

(a) **Specification:** we have set out above that it would be important to specify clear criteria for measuring equivalent access to functionality for third-party browsers using Apple's version of WebKit, compared to Safari. This is because there is a high risk of circumvention in relation to high level or static requirements. We also note there is an information asymmetry between Apple and other parties in relation to the working of iOS architecture, the availability of functionality and on what terms and conditions are necessary for access.

(b) **Circumvention, monitoring and enforcement**: any requirements (or criteria) in connection with this potential remedy, such as those relating to whether Apple was providing equivalent access, would need ongoing monitoring and the need for iteration to reflect technological change. Any requirements in connection with this potential remedy would need to be monitored closely on an ongoing basis to ensure that equivalent access, including terms and conditions imposed by Apple on browser vendors continuing to use Apple's version of WebKit are applied equally to Safari and do not raise similar effectiveness risks as those set out under potential remedy 1.

Any potential disputes between browser vendors and Apple in relation to potential non-compliance with access requirements would require an ongoing independent dispute resolution mechanism.

98. We conclude that, taken together, these risks mean that there is a significant risk to the effectiveness of potential remedy 2 in addressing AEC 2.

## Potential remedy 3 to address AECs 1, 2 and 3

99. As set out in Section 10: Decisions on AEC(s) in the supply of mobile browsers, browser engines and in-app browsing, AEC 3 comprises multiple features relating to in-app browsing technology.[78] Particularly relevant to AEC 3, and to this potential remedy, are the following features:

    (a) Apple restricts the use of alternative browser engines for in-app browsing on iOS; and

    (b) Apple does not permit the use of remote tab IABs for in-app browsing on iOS.

### Description of potential remedy 3

100. There are two parts to this potential remedy which seek to address the two features (noted above) which contribute to AEC 3 in in-app browsing:

    (a) a requirement for Apple to: (i) allow native app developers on iOS in the UK to use their choice of browser engine for in-app browsing within their native app (a 'bundled engine'); and (ii) provide interoperability with bundled engines for in-app browsing ('potential remedy 3a'); and

    (b) a requirement for Apple to allow sufficient cross-app functionality to enable native apps to invoke third-party browsers (regardless of the browser engine they use) to support in-app browsing ('potential remedy 3b').

101. We describe each of these two parts below in more detail. We set out key remedy design considerations and then we describe risks that may impact the effectiveness of this potential remedy if implemented through the remedy-making provisions of the EA02.

**Potential Remedy 3a: Requirement for Apple to: (i) allow native app developers on iOS in the UK to use a bundled engine; and (ii) provide interoperability with bundled engines for in-app browsing.**

102. This potential remedy aims to achieve a similar outcome to potential remedy 1 – namely to allow the use of alternative browser engines on iOS for in-app browsing within a developer's native app.

103. The approach to enabling access for mobile browsers, set out in potential remedy 1, could be applied to non-browser apps, including any security and privacy obligations placed on native app developers.

[78] Some of those features also contribute to AECs 1 and 2.

104. Potential remedies 1 and 3 in combination would result in native apps on iOS being able to choose which browser engine to use to facilitate browsing, rather than being limited to using Apple's version of WebKit.

105. From a technical perspective, the potential remedy would enable native apps (that are not dedicated mobile browsers) to use an alternative to Apple's iOS WKWebView, which is currently the only webview available on iOS.[79]

106. Depending on the use case, such developers could use their own custom browser engine in-app browser (IAB) (referred to as a 'bundled engine IAB'), incorporate a forked browser engine, or use an alternative webview option provided by a third party (eg GeckoView).

107. The specification of how this potential remedy would work in practice carries many similarities to potential remedy 1. For example, this potential remedy would require Apple to:

(a) Remove clause 2.5.6 from Apple's App Review Guidelines, which requires native apps to use WebKit (and refraining from introducing any guidelines with similar effect in the future); and

(b) Provide access on fair, reasonable and non-discriminatory (FRAND) terms to app developers choosing to use browser engines other than the version of WebKit specified by Apple ('alternative browser engines').

108. High-level parameters such as those set out as part of potential remedy 1 in Description of potential remedy 1 sub-section above could similarly apply to this potential remedy. However, we consider that the level of access to iOS would not need to be 'equivalent' to that of WebKit and Safari, for native apps (that are not mobile browsers) to be able to implement a bundled engine IAB.

109. In addition, we note that, under this potential remedy, native apps that choose to bundle a browser engine would continue to be able to use all operating system features and functionalities available to third-party native apps on iOS. In other words, some provisions[80] set out above in the Description of potential remedy 1 sub-section already apply to all native apps and we would expect these to remain and not to be degraded if a native app developer chose to bundle a browser engine.

[79] See Section 7: In-app browsing.
[80] Provisions set out under Potential remedy 1 such as those requiring 1) enabling access to all other current operating system-level features and functionalities that exist on iOS and are available for use by third-party applications, but which WebKit and Safari currently do not use and 2) enabling access to future operating system-level features and functionalities available to WebKit, Safari, or third-party applications, whether or not WebKit and Safari choose to use them.

110. Therefore, we would expect that high-level parameters for assessing the adequacy of access to functionality should include the following:

**Specific to this potential remedy:**

(a) enabling native app developers to bundle their own engine in a way which allows them to implement the security, privacy, performance and other features they require;

**Similar to the parameters proposed under potential remedy 1 (see Description of potential remedy 1 sub-section above):**

(b) enabling access in a way which respects the technical architecture of alternative browser engines;

(c) enabling access to the necessary operating system-level features and functionalities that WebKit and Safari currently use;

(d) continuing to enable access to all other current operating system-level features and functionalities that exist on iOS and are available for use by other third-party applications, but which WebKit and Safari currently do not use; and

(e) enabling access to future operating system-level features and functionalities available to Safari, WebKit or other third-party applications, whether or not WebKit and Safari choose to use them.

111. Under this potential remedy, Apple would be able to withhold certain iOS features and functionalities that WebKit and Safari have access to, or provide them in a more restricted manner, but only where Apple could demonstrate that this is necessary for security or privacy reasons. We consider that Apple should be required to make its reasoning public if it withheld such iOS features and functionality.

112. Similar to potential remedy 1 (see the Description of potential remedy 1 sub-section), this potential remedy would:

(a) require Apple to provide quality documentation and/or guidance, service-level support and access to a range of performance metrics; and

(b) enable Apple to impose security and privacy requirements for app developers choosing alternative browser engines to ensure that the access to iOS to native app developers is facilitated in a way that takes account of security and privacy considerations.

113. We consider that the entitlement, or other access mechanism, aimed at browser vendors should be maintained separately from that aimed at non-browser apps.

114. For the avoidance of doubt, the following high-level parameters would not be used to assess the required access to functionality as part of this potential remedy:

(a) enabling access to the required functionality to allow browser vendors using alternative browser engines to install and manage PWAs using alternative browser engines; and

(b) enabling access to the required functionality to allow browser vendors using alternative browser engines to check whether their browser has been set as default.

**Potential remedy 3b: A requirement for Apple to allow sufficient cross-app functionality to enable third-party browsers to provide in-app browsing in native apps**

115. This potential remedy would require Apple to provide in-app browsing functionality enabling mobile browsers to be invoked by a native app in an IAB. Apple would be required to enable cross-app functionality for all mobile browsers on iOS, irrespective of the browser engine being used.

116. The potential remedy would require that Apple:

(a) allows mobile browsers to support in-app browsing functionality which relies on the functionality of mobile browsers. The functionality would allow the sharing of resources (eg data and memory) between the IAB and the corresponding mobile browser (see Section 7: In-app browsing); and

(b) provide native apps with a straightforward protocol to access, and choose between, multiple in-app browsing options, including an option to invoke the user's default mobile browser or to use a specific mobile browser on the user's device.

## How potential remedies 3a and 3b would seek to address the AECs and customer detriment

117. The aim of implementing both parts of this potential remedy in combination would be to improve competition between providers of in-app browsing technology by allowing alternative options for in-app browsing on iOS to compete with Apple, which is currently the only provider of in-app browsing technology on iOS. Further, implementing both parts in combination would also improve competition between browser engines on iOS by enabling alternative browser engines to compete for in-app browsing traffic.

118. Section 7: In-app browsing notes that an app developer with more control over the browser engine could even make certain improvements to the security and privacy of its IAB that would not be possible for those using the OS-provided in-app browsing implementations.

119. As explained in Section 7: In-app browsing, even if very few native apps developed and bundled their own browser engines, this could significantly increase competition in the browser engine and in-app browsing technology markets on iOS – given that Apple is currently the only provider. Innovation within bundled engine IABs may further positively impact on competition between browser engines, for example, if a native app developer decided to offer its browser engine to third parties, or if it published its bundled engine's features so that third parties could adopt them (eg by contributing code to the open-source community).

120. Separately, if alternative browser engines were permitted on iOS, additional traffic via remote tab IABs may contribute to increased web compatibility for them and therefore allow them to compete more effectively– as well as increased brand awareness and engagement.

121. As set out in Section 7: In-app browsing (Conclusions on Apple not permitting remote tab IABs on iOS sub-section), while not many browser vendors appear interested in providing remote tab IABs, a small number of browser vendors consider this product to be important for their ability to compete and have expressed interest in offering it. In the current context of a total lack of rivalry, we consider that even limited entry would result in a significant effect on competition.

122. Additionally, implementing both parts of the potential remedy in combination would improve competition between browser vendors on iOS by enabling such mobile browsers to compete for in-app browsing traffic. Being able to support in-app browsing would allow browser vendors to grow and serve their existing customers better, providing a more consistent experience on the device and offering users features (eg tracker blockers) while the user is browsing from within an app, as set out in further detail in Section 7: In-app browsing. Remote tab in-app browsing implementation would also increase the traffic to the mobile browser, which would benefit the browser vendor and browser engine provider.

123. Accordingly, this potential remedy would aim to address AEC 3 directly, as well as AECs 1 and 2 indirectly and the resulting customer detriment that may be expected to result from the AECs we have found, by ensuring that:

    (a) app developers are able to innovate and use in-app browsing technology which was previously either not available or restricted – in turn improving the in-app browsing experience for iOS users;

(b) app developers have choice of browser engine they can use for their native app on iOS – a choice which currently does not exist;

(c) browser vendors are able to access traffic from in-app browsing;

(d) browser vendors are able to benefit from increased engagement and brand awareness with their browsers;

(e) browser engine providers are able to access traffic from in-app browsing;

(f) browser engine providers are able to benefit from increased investment from web developers into web compatibility; and

(g) Apple is no longer the sole provider of in-app browsing technology on iOS and faces competitive pressure from competing suppliers.

124. However, we consider that there are a number of risks to the effectiveness of this potential remedy if implemented through the remedy-making provisions of the EA02. Taken together, they amount to a significant risk to the effectiveness of this potential remedy in addressing the AECs and resulting customer detriment. We set this out in further detail in the section below.

## Key remedy design considerations

### Potential remedy 3a: Requirement for Apple to allow native app developers on iOS in the UK to use a bundled engine, and to require interoperability with bundled engines for in-app browsing

125. We set out below an assessment of whether the potential remedy described above would be effective and the key considerations that would be relevant in this respect. In particular, an effective remedy would require:

1. adequate specification of what level of access to functionality would be required, including objective criteria to assess and monitor compliance by Apple;

2. a mechanism for assessing the terms and conditions imposed by Apple on parties seeking to use (or applying for an entitlement to use) alternative browser engines; and

3. a clear process for app developers to request access to functionality and a mechanism for resolving disputes.

### 1. Specification of the level of access to operating system features and functionalities required to support bundled engine IABs and objective criteria to measure whether Apple is providing sufficient access

126. A key consideration for this potential remedy is that any requirement on Apple to provide access to native app developers would need to be sufficiently clear as regards the level of access provided.

127. For this potential remedy, we do not consider that native apps would require 'equivalent' access (as described in potential remedy 1) to implement bundled engine IABs. This would mean that native apps using a bundled engine could have access to fewer operating system features and functionalities, compared with mobile browsers using alternative engines – if Apple demonstrates that restricting access to certain features or functionalities is necessary for security or privacy reasons. We consider that Apple should be required to make its reasoning public if it withheld such iOS features and functionality.

128. The level of access which Apple grants to its operating system features and functionalities would need to be sufficient to enable, at the minimum, native app developers to implement in-app browsing using their own browser engine.

129. This would require the setting of objective criteria (see 1. Specific requirements for Apple to demonstrate equivalent access sub-section for potential remedy 1) to determine whether Apple is providing the required access to its operating system features and functionalities, in particular to take into account potential developments over time.

130. In response to WP7 Apple submitted in relation to potential remedy 3a that:

(a) IABs using alternative engines would be less secure and private due to developers not having as much experience as browser vendors in dealing with complex issues associated with accessing the web. These developers would not generally prioritise dealing with these issues, or they might not have the resources to do so.[81]

(b) Reflecting this, Apple submitted that, in the EU, it provides differing levels of access to iOS functionality to support alternative engines in the case of native apps, compared with browser apps due to the greater level of risk associated with IABs. For example, Apple submitted that it has provided additional functionality (such as dynamic code generation also known as JIT) to browser vendors, as this can be important for the browser use case and is not material for the in-app browsing use case.[82]

[81] Apple's response to the CMA's information request [✂].
[82] Note of meeting with Apple [✂].

131. In response to the PDR, Apple submitted in relation to potential remedy 3 that:

(a) Evidence shows that app developers are generally content with the current IAB implementations on iOS.

(b) There is a clear lack of demand for custom-tabs type implementation or a bundled engine implementation on iOS, with the exception of Meta.

(c) Potential remedy 3 would lead to significant privacy and security risks on iOS, which are more significant for bundled engine IABs, including the impact of the patch gap issue.[83]

132. In response to the PDR, Meta[84] submitted that it supports potential remedy 3a and noted that if this remedy was to be designed correctly it could effectively promote competition and innovation in the markets for IABs, mobile browsers and browser engines in the UK.

133. Meta[85] submitted that potential remedy 3a should not exclude PWA functionality and existing functionality allowed to PWAs should not dictate what functionality IABs can offer in the future. Meta[86] further submitted that nascent interest from web developers in using PWAs on its IAB already exists.

### 2. The terms and conditions (or application criteria) that Apple may impose on native app developers choosing to use a bundled engine

134. In line with potential remedy 1, we consider that it may be necessary for Apple to impose appropriate security and privacy requirements (through terms and conditions or application criteria) on native app developers choosing to use alternative browser engines on iOS. Further, Apple should be able to amend such requirements to ensure they remain up-to-date and reflect the latest security threats.

135. However, similar to that described in The terms and conditions (or application criteria) that Apple may impose on browser vendors using alternative browser engines sub-section above, we do not consider that Apple should be able to impose other requirements which may hinder third-party native app developers' ability to use alternative browser engines for in-app browsing and therefore undermine the effectiveness of this potential remedy.

[83] Apple's response to the CMA's provisional decision report dated 22 November 2024, paragraph 191.
[84] Meta's response to the CMA's provisional decision report dated 22 November 2024, page 1
[85] Meta's response to the CMA's provisional decision report dated 22 November 2024, page 1
[86] Meta's response to the CMA's provisional decision report dated 22 November 2024, page 1

136. A number of stakeholders made submissions in response to WP7, in relation to the possible implications for security and privacy of potential remedy 3a should native apps be allowed to use alternative engines for in-app browsing:

(a) Apple submitted that a remedy to enable alternative engines would produce adverse effects which are disproportionate to the remedy's aim which, in its view, was a mere theoretical enabling remedy outweighed by clear and serious risks to users.[87] Apple pointed to risks in terms of security, privacy and reliability.[88]

(b) Apple submitted that if an app developer offering in-app browsing adopts an alternative browser engine, that will cause significant risks including the 'patch gap' problem[89] and the fact that users may not understand security and privacy on that service.[90]

(c) OWA submitted that such a remedy allowing alternative webviews could encourage behaviour from developers that is against consumer interests, for instance by way of increased user tracking.[91]

(d) Conversely, Meta submitted that bundled engine IABs can allow for potential benefits to users, such as faster patching and fewer crashes, compared to a native webview.[92]

(e) A large app developer submitted that its custom engine IAB contains a number of technologies which combat [security risks] [✂] [✂][93]

137. In response to the PDR, Apple submitted that very few app developers in the UK are aware of the voluntary code of practice on mobile app security and privacy for app developers introduced in 2022. Apple submitted this highlights there is a lower level of specialism or understanding among app developers of security and privacy compared to dedicated browser apps.[94]

138. Relatedly, we note that in the context of the MEMS, RET2's view was that allowing all native apps to bundle their own browser engines would lead to fragmentation and a less secure ecosystem with many apps using outdated engines.[95]

---

[87] Apple's response to CMA Working Paper 7: Potential Remedies dated 8 August 2024, paragraph 61.
[88] Apple's response to CMA's Working Paper 1 to 5, published on the CMA's case page on 3 September 2024, para 173.
[89] The 'patch gap' problem refers to where a user runs an outdated version of a browser engine, thus exposing users to known but unmitigated security risks.
[90] Note of meeting with Apple [✂].
[91] OWA's response to CMA Working Paper 7: Potential Remedies dated 8 August 2024, section 4.4, page 19.
[92] Meta's response to Working Paper 4 In-app browsing within the iOS and Android mobile ecosystems dated 5 July 2024, paragraph 3.4.2.
[93] [✂], submission to CMA [✂].
[94] Apple's response to the CMA's provisional decision report dated 22 November 2024, paragraph 174.
[95] RET2's advice to the CMA [✂], provided as part of the Mobile Ecosystems Market Study.

139. We consider that enabling Apple to impose security and privacy requirements would address the above concerns described by Apple and other stakeholders.

140. We note that, in the EU, Apple has set near-identical security and privacy requirements for its entitlement aimed at dedicated mobile browsers (Web Browser Engine Entitlement) and its entitlement for native apps, EBEE.[96] We consider that it is reasonable to apply minimum security and privacy requirements on native app developers similar to those applicable to browser vendors, given the similarity in the risks that would apply when enabling use of alternative browser engines. As mobile browser developers are more likely to be familiar with the security risks of a browser engine, compared to native app developers in general, we consider that, as an additional mitigation to such security and privacy concerns, the level of access which Apple grants to iOS features and functionality for native app developers could be limited compared to browser apps – if Apple demonstrates that restricting access to certain iOS features or functionalities is necessary for security or privacy reasons.

141. In this context, we consider Apple's EBEE privacy requirement to block third-party cookies by default, unless users opt in, might be a reasonable way to limit the risk of increased user tracking compared with WKWebView-based IABs.[97]

142. Nevertheless, as set out in Section 7: in-app browsing (Conclusions on Apple's ban on alternative browser engines for bundled engine and webview IABs sub-section), we do not consider that bundled engine IABs (or webviews based on alternative browser engines) would necessarily be less secure or offer lower privacy levels for users than dedicated browsers (with appropriate mitigations put in place, such as entitlements discussed above). Absent the ban, app developers would gain greater control over the browser engine that powers the IAB, such that they could introduce new engine-level features to strengthen the IAB's security.

143. Ongoing compliance with such security and privacy requirements would likely mean that only a limited number of native app developers – those with appropriate expertise and sufficient resources – would choose to bundle a browser engine as part of this potential remedy. However, we consider that such requirements and restrictions relating to security and privacy may be necessary to protect users and that a limited uptake of this potential remedy would not take away from its effectiveness.

144. This is because, even if only a limited number of developers bundled a browser engine, this would result in a significant increase in competition amongst browser engines, considering the current total lack of rivalry. On the one hand, allowing in-app browsing implementations to run on alternative browser engines would place

[96] See Using alternative browser engines in the European Union - Support - Apple Developer, accessed 4 October 2024.
[97] This requirement is part of Apple's EBEE in the EU. See Using alternative browser engines in the European Union - Support - Apple Developer, accessed 10 October 2024.

greater competitive pressure on Apple to improve its own in-app browsing technology. On the other, it would also impose a potential constraint on the adjacent markets for mobile browsers and browser engines.

*Other conditions*

145. Considerations on other conditions would be the same as those set out in relation to potential remedy 1 (see The terms and conditions (or application criteria) that Apple may impose on browser vendors using alternative browser engines sub-section). We note the following related stakeholder submissions regarding Apple's separate binary requirement:

(a) A large app developer submitted that it had planned to introduce a custom browser engine IAB for iOS in the EU, but the only reason it has not yet done so is that Apple's requirements are too prohibitive, particularly the separate binary requirement. It also considered that some of the restrictions related to third-party cookies and how the web should work are extremely limiting and do not reflect the latest industry standards.[98]

(b) A browser vendor submitted that it would be interested in bringing a [✂] webview solution to iOS if it were viable to do so (which, we understand, native apps could potentially bundle). However, it submitted that the restrictive nature of Apple's requirements in the EU have meant that this has not been explored in detail.[99]

146. In response to the PDR, Meta submitted the following in relation to the separate binary requirement and the importance to developers of the ability to perform A/B testing:

(a) An alternative to Apple's separate binary requirement could be the implementation of region-specific binaries, allowing browser vendors to retain single App Store entries and feature updates.

(b) Apple already has the technical capability to send different binaries to different groups of users demonstrating that iOS is already capable of supporting the selective distribution of app code.

(c) In the Netherlands, Apple has enabled use of alternative payments mechanisms for dating apps through entitlements without requiring [the apps] to create and use separate binary.

(d) It is critical for app developers to have the ability to conduct A/B testing; without the confidence that A/B tests have provided, Meta would have lacked

[98] Note of meeting with [✂].
[99] [✂] response to the CMA's information request [✂].

the confidence to make its custom browser engine IAB available broadly or to invest the resources required to develop its custom browser engine IAB in the first instance.[100]

147. As is the case for potential remedy 1, we consider that a separate binary requirement appears unduly onerous. The evidence indicates that there are alternative means of allowing browser vendors to use alternative browser engines on iOS. Similarly, we consider that restrictions on the location of where testing and development of native apps using alternative engines may take place do not appear necessary.

**3. A clear process for app developers to request access to functionality; and a mechanism for resolving disputes.**

148. Considerations on monitoring and enforcement of the potential remedy are similar to those set out in relation to potential remedy 1 in the A clear process for third party browser vendors to request access to functionality; and a method for resolving disputes sub-section above.

**Conclusion on potential remedy 3a – Requirement for Apple to allow native app developers on iOS in the UK to use a bundled engine, and to require interoperability with bundled engines for in-app browsing**

149. As noted above under the discussion of key remedy design considerations, there are a number of risks to the effectiveness of this remedy if implemented through the remedy-making provisions of the EA02.

150. These relate to:

(a) **Specification**: it would be important to specify clearly the level of access to operating system features and functionalities that native app developers would require in order to implement a bundled engine IAB. This is because there is a high risk of circumvention in relation to high level or static requirements. We also note there is an information asymmetry between Apple and other parties in relation to the working of iOS architecture and the availability of functionality.

(b) **Circumvention, ongoing monitoring and enforcement:** there is a high risk of circumvention if the specifications described above are set at too high a level, are insufficiently clear or are too static. Any requirements in connection with this potential remedy would need to be monitored closely on an ongoing basis to ensure that access, including terms and conditions imposed by

[100] Meta's response to the CMA's provisional decision report dated 22 November 2024, pages 2—3.

Apple on native app developers seeking to implement bundled engine IAB do not undermine the effectiveness of the potential remedy.

151. In general, the implementation of this potential remedy would require ongoing monitoring and oversight and the requirements on Apple may need to be iterated and revised in light of technological developments. As noted above, there would need to be a process for third parties to make access requests and a mechanism for resolving disputes for the duration of the potential remedy.

152. We conclude that, taken together, these risks mean that there is a significant risk to the effectiveness of potential remedy 3a in addressing AECs 1, 2 and 3.

**Potential remedy 3b: Requirement for Apple to allow sufficient cross-app functionality and technical support to enable third-party browsers to provide in-app browsing in native apps**

153. We set out below an assessment of whether potential remedy 3(b) (as described above) would be effective and the key considerations that are relevant in this respect. In particular, an effective remedy would require:

1. adequate access to the required functionality, and technical support, by Apple to browser vendors and native app developers so that mobile browsers can be invoked in an IAB;

2. adequate documentation and service support by Apple to native app providers; and

3. choice being given to app developers to implement in-app browsing by either invoking a user's default mobile browser or a mobile browser chosen by the developer (provided that mobile browser is installed on a user's device and supports in-app browsing).

**1. Provision of required functionality and technical support to enable mobile browsers to provide in-app browsing on iOS**

154. On iOS, SFSafariViewController provides users with an in-app browsing experience that may be comparable with their browsing experience in Safari. However, SFSafariViewController does not call on a mobile browser to implement in-app browsing.

155. The capability for a native app to invoke a mobile browser in an IAB is not currently available on iOS. As part of this potential remedy, Apple would need to enable such functionality, which may involve further developing cross-app functionality on iOS.

156. Apple would be required to make the functionality available to third-party apps at the same time as it becomes available to Apple's first-party apps.

157. A process would need to be in place to ensure the functionality is developed in a way that is fit for purpose.

158. Apple would be required to make this capability available to all native app developers and browser vendors, with no specific obligations placed on them, including no obligation to use the version of WebKit engine as specified by Apple. The remedy would not require native app developers to bundle a browser or browser engine within their app's binary.

*Engineering considerations regarding the required functionality*

159. Apple submitted that iOS [✂].[101] Apple also submitted that [✂].[102]

160. Both Mozilla[103] and MOW[104] submitted that they agreed that any potential remedy enabling IAB requires sufficient cross-app functionality on iOS (such as the sharing of resources in relation to data and memory) between the IAB and the corresponding mobile browser to ensure that user experience is not compromised.

161. MOW submitted that Apple's de-identified 'random identifier'[105] could be made available to rivals to support the interoperability (between mobile browsers and IAB) including new prohibitions or reidentification by any recipient of this common match key.[106]

162. [✂][107]

163. In this context, we note that functionality that allows mobile browsers to provide in-app browsing in native apps is available on Android devices and is referred to as Android Custom Tabs. This relies on Android's cross-app functionality, the so-called 'intents' system, which enables an app to call another.[108]

164. On iOS, there are examples of functionality enabling the launch of an app from within another app, namely 'universal links' and a URL-scheme.[109] We consider that it would in principle be feasible for Apple to support sufficient cross-app

---

[101] Apple's response to the CMA's information request issued [✂].
[102] Apple's response to the CMA's information request issued [✂].
[103] Mozilla's response to the CMA's provisional decision report dated 22 November 2024, page 6.
[104] Mozilla's response to the CMA's provisional decision report dated 22 November 2024, page 6.
[105] Random identifiers are unique codes which are assigned to processes, users or devices to enable data collection without personal information being exposed. See Privacy - Features - Apple (UK). Accessed 31 January 2025
[106] Movement for an Open Web's response to the CMA's provisional decision report dated 22 November 2024, page 11.
[107] [✂] response to the CMA's provisional decision report dated 22 November 2024, [✂.].
[108] See Overview of Android Custom Tabs | Views | Android Developers; Intent | Android Developers. Accessed 21 October 2024.
[109] See Allowing apps and websites to link to your content | Apple Developer Documentation. Accessed 21 October 2024.

functionality as part of this potential remedy by further developing cross-app functionality on iOS.

165. In a more recent submission, Apple submitted that it originally removed statesharing from first-party and third party apps. Apple later reinstated data-sharing between Safari and SFSafariViewController, for first-party apps and settings, because of technical issues affecting first-party apps, [✂].[110]

166. Based on Apple's latest submission, a form of cross-app data-sharing functionality on iOS does exist and could provide the functionality that a third-party app IAB would require to share resources (eg data and memory) with a browser app.[111]

167. This suggests that the technical capability to enable in-app browsing which uses browsers to complete the browsing action may not require the extent of additional development or engineering and financial investment as was originally estimated.

*Security and privacy considerations*

168. Facilitating cross-app functionality to enable third-party browsers to provide in-app browsing may carry some security risks as functionality enabling the launch of an app from within another app can enable security exploits which would need to be mitigated.[112,113]

169. In relation to cross-app functionality and security and privacy considerations, stakeholders made the following submissions:

(a) Apple submitted that for Apple to implement cross-app functionality, it would be a fundamental, significant architectural question. It submitted that this would not be something trivial to implement as the architecture of the iOS platform does not support one app running inside of another app space and would have very significant aspects to work through for security and privacy.[114]

(b) Apple submitted that SFSafariViewController has security benefits over a remote tab implementation as it isolates the browsing session state. SFSafariViewController is a private sandbox container that offers a firewalled

[110] Apple's supplementary submission to the CMA's provisional decision report dated 22 November 2024 [✂].
[111] Apple's supplementary submission to the CMA's provisional decision report dated 22 November 2024 [✂].
[112] Holmberg, A. (2022) iOS vs Android: Security of Inter-App Communication.
[113] See Defining a custom URL scheme for your app | Apple Developer Documentation. Accessed 21 October 2024.
[114] Apple hearing with the CMA [✂].

webview. This means that neither the third-party app, nor Safari, gain access to browsing session state.[115,116]

(c) Apple further submitted that its approach with SFSafariViewController avoids exposing users to the 'patch gap' problem on Android – where a user runs an outdated version of a browser engine, thus exposing users to known but unmitigated security risks.[117]

(d) Google submitted that it had recently introduced changes on Android, which 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. Google submitted that similar exploits can happen on iOS, even in absence of intents, as any app can register to open itself automatically in response to different URLs. The issue is therefore not limited to browser apps but any vulnerable app that is opened automatically without express user intent.[118]

170. In response to the PDR, Apple submitted that it has identified significant issues with offering a custom tabs-type approach, as this would expose communications between the relevant native app and browser app to potential exploitation by an attacker.[119]

171. We have limited evidence available to determine specific types of mitigations that could be put in place to adequately address any security or privacy risks resulting from enabling mobile browsers to provide in-app browsing on iOS.

172. We consider that Apple is best placed to identify appropriate mitigations and should be allowed to design the required functionality in a way that minimises the risks discussed above, whilst effectively enabling native app developers to invoke a mobile browser in an IAB.

173. However, we note that this approach introduces a circumvention risk, as Apple's design could limit in-app browsing functionality and render this potential remedy ineffective.

174. Regarding the security and privacy of any mobile browser called upon for in-app browsing:

[115] Many IABs 'share state', meaning that the IAB shares data, resources and users' preferences with either the app or a browser on the device.
[116] Apple's response to Working Paper 7: Potential Remedies dated 8 August 2024, paragraph 57-59. As noted in a more recent submission, Apple originally removed statesharing from first-party and third party apps. Apple later reinstated datasharing, as between Safari and SFSafariViewController, for first-party apps and settings, because of technical issues affecting first-party apps.
[117] Apple's response to Working Paper 7: Potential Remedies dated 8 August 2024, paragraph 59.
[118] Note of meeting with Google, [✂].
[119] Apple's response to the CMA's provisional decision report dated 22 November 2024, page 30.

(a) mobile browsers using the system-provided WKWebView would benefit from WebKit's security and privacy protection; and

(b) in the case of mobile browsers using alternative browser engines, the considerations set out in relation to security and privacy in respect of potential remedy 1 apply.

175. Overall, while the outcome will depend on the precise implementation, we do not consider that remote tab IABs would necessarily be less secure or offer lower privacy levels for users than dedicated browsers (see Section 7: In-app browsing).

*Objective criteria for assessing the required functionality*

176. The following high-level criteria could be used to determine whether Apple is providing the required functionality:

(a) Apple allows mobile browsers to support in-app browsing functionality which relies on the functionality of mobile browsers. The functionality would allow the sharing of resources (eg data and memory) between the IAB and the corresponding mobile browser (see Section 7: In-app browsing); and

(b) Apple allows native apps to have access to and choose between multiple in-app browsing options, including an option which could invoke the user's default mobile browser or use a mobile browser on the user's device.

177. Apple could delay and/or create technical barriers for native app developers attempting to use this new functionality and this raises a circumvention risk.

**2. Apple to provide adequate documentation and support**

178. Similar to potential remedy 1, potential remedy 3b would require Apple to provide up to date quality documentation and guidance, service-level support and access to a range of performance metrics.

**3. The relevance of a user's default browser for in-app browsing**

179. Apple submitted that creating a remote tab implementation that extends the users' default browser choice in all circumstances would remove app developers' choice and control over in-app browsing.[120]

180. In response to the PDR, Mozilla submitted it would welcome requirements in respect of IABs which remove Apple's restriction over in-app browsing. Mozilla further submitted it would support a remedy for IABs which honours a user's choice of default browser when browsing in-app but acknowledged the needed

[120] Apple's response to CMA's Working Paper 7: Potential Remedies dated 8 August 2024, paragraph 68.

balance between respecting user choice and putting choice in the hands of app developers.[121]

181. OWA submitted that potential remedy 3b should instead mandate that Apple upgrades SFSafariViewController to respect the user's browser choice.[122]

182. We note that the aim of this potential remedy is to increase choice for native app developers in how they implement in-app browsing on iOS rather than to prescribe the use of specific in-app browsing technologies. We consider therefore that this potential remedy should require Apple to offer functionality that would enable app developers to either invoke a user's default mobile browser in an IAB (if the mobile browser supports such in-app browsing implementation on iOS) or the developer's choice of mobile browser.

**Conclusion on potential remedy 3b**

183. As noted above under the discussion of key remedy design considerations, there are a number of risks to the effectiveness of this potential remedy, if implemented through the remedy-making provisions of EA02.

184. These relate to:

(a) **Specification:** it would be important to specify clearly the requirements to be placed on Apple to achieve the requisite level of cross-app functionality, documentation and technical support, including in relation to any security and privacy conditions which Apple seeks to impose.

Future iteration of the remedy requirements is likely to be necessary to address the risk that innovation or technological developments enable the potential remedy to be circumvented or otherwise become ineffective.

(b) **Circumvention, ongoing monitoring and enforcement:** there is a high risk of circumvention in relation to any of the requirements which form part of this potential remedy, and which are at too high a level or which are too static. We also note that there is an information asymmetry advantage between Apple and other parties in relation to the working of iOS architecture and availability of functionality.

Any requirements placed on Apple (or on third parties) in connection with this potential remedy would need to be closely monitored on an ongoing basis to ensure that they remained effective and were being adhered to.

[121] Mozilla's response to the CMA's provisional decision report dated 22 November 2024, pages 5 – 6.
[122] Open Web Advocacy (OWA)'s response to the CMA's provisional decision report dated 22 November 2024, page 14.

As noted above, there would need to be a process for app developers to make access requests to Apple and a mechanism for resolving disputes for the duration of the remedy.

185. We conclude that, taken together, these risks mean that there is a significant risk to the effectiveness of potential remedies 3a and 3b in addressing AECs 1, 2 and 3.

# Potential remedy 4 addressing AEC 2

## Description of the potential remedy

186. A potential remedy to address AEC 2 would be to prohibit the contractual provisions in the ISA pursuant to which Google shares revenue derived from Chrome on iOS with Apple (Chrome Revenue Share). Further, Apple and Google would be prohibited from entering into any agreement of equivalent effect pursuant to which Google shares its search advertising revenue with Apple derived from Chrome on iOS (including agreements in relation to any other future product that performs the equivalent functions of a dedicated mobile browser).

187. However, the potential remedy would not restrict the parties from sharing revenues in respect of search traffic derived through Safari.

## How potential remedy 4 would seek to address the AEC and customer detriment

188. As set out in Section 10: Decisions on AEC(s) in the supply of mobile browsers, browser engines and in-app browsing, we have found that the Information Services Agreement (ISA), individually or in combination with other features, gives rise to an AEC in the market for mobile browsers on iOS.

189. Prohibiting Google from sharing search advertising revenues with Apple derived from Chrome on iOS would significantly increase Apple's and Google's financial incentives to compete against each other for user traffic on their respective browsers, which are by far the two most established mobile browsers iOS.

190. The aim of this potential remedy, in combination with potential remedy 1 (enabling alternative browser engines on iOS), potential remedy 2 (enabling equivalent access to features and functionalities for all WebKit-based browsers on iOS) and potential remedy 5 (a combination of choice architecture remedies), would be to increase Apple's and Google's incentives to compete more vigorously on iOS, which would likely benefit consumers through increased innovation in mobile browsers, resulting in additional features and functionalities being introduced that otherwise would not occur.

191. However, we consider that there are a number of risks to the effectiveness of this potential remedy if implemented through the remedy-making provisions of the EA02. Taken together, they amount to a significant risk to the effectiveness of this potential remedy in addressing the AEC and resulting customer detriment. We set this out in further detail in the section below.

## Key remedy design considerations

192. We set out below an assessment of whether the potential remedy described above would be effective and the key considerations that would be relevant in this respect.

### The connection between the Chrome Revenue Share and revenue-sharing arrangements under the Safari Agreement

193. In response to WP7, Apple submitted the following in relation to the effectiveness of potential remedy 4:

(a) Imposing a prohibition of the Chrome Revenue Share would significantly chill innovation in relation to search and would potentially call into question a wide array of platform business arrangements.[123]

(b) Prohibition of the Chrome Revenue Share could lead to significant harms to browser competition and the user experience of browsing on iOS [✂].[124]

(c) Linked to this, Apple submitted that the Chrome Revenue Share is the most efficient and proportionate way to remove the [✂] and by addressing these underlying incentives, it also removes the need for monitoring. [✂].[125]

194. Additionally, Apple submitted that without the [✂].[126]

195. In response to the PDR, Apple submitted the following in relation to potential remedy 4:

(a) the proposed ISA remedy would exceed what is necessary to remedy any AEC in the UK;

(b) a global remedy would be inconsistent with well-established principles of international comity and would give rise to significant risk that the CMA would pre-empt regulatory or other action in jurisdictions outside the UK;

(c) a prohibition of the Chrome revenue share without altering the other ISA obligations would unbalance the commercial relationship between Apple and Google creating a serious risk of distortion;

[123] Apple's response to [✂].
[124] Apple's response to [✂]. See also Apple's response to the CMA's provisional decision report dated 22 November 2024, paragraph 193.
[125] Apple's response to [✂].
[126] Apple's response to [✂].

(d) imposing such a remedy would provide a financial windfall to Google and would likely strengthen Google's dominant position in internet search in the UK (and beyond); and

(e) removing the revenue share would be an unwarranted intrusion on Apple's ability to monetise its platform and obtain proper compensation for the value that it brings to Google.[127]

196. In response to WP7 Google submitted the following points in relation to the effectiveness of potential remedy 4:

(a) If Google were prohibited from entering into a revenue share agreement with Apple in respect of Chrome on iOS, [✂]. This would result in Chrome's ability to compete being impaired [✂].[128]

(b) The ISA reflects the outcome of a complex commercial negotiation. [✂].[129]

197. Google, also submitted that the ISA creates rivalry-enhancing efficiencies: (i) greater browser choice and quality because Chrome is a stronger competitor on iOS; and (ii) there is greater incentive to invest in Chrome on iOS as a result of the ISA [✂].[130]

198. In response to the PDR, Google submitted that [✂].[131] [✂].[132] [✂].[133]

199. In response to the PDR, MOW submitted that it agreed with a prohibition on Apple and Google sharing revenue from Chrome.[134] MOW also submitted that given browsers' main funding model which has traditionally relied on revenue sharing from search advertising, it is important that any 'rivals to Safari or WebKit' are allowed to earn search revenues from search engine providers.[135]

200. MOW submitted that a remedy should ensure that Google does not shift its single payment to Apple to a multi-payment model.[136]

201. We address Apple and Google's points below by grouping them into two broad categories assessing whether: (i) prohibition of the Chrome Revenue Share could

---

[127] Apple's response to the CMA's provisional decision report dated 22 November 2024, paragraphs 193 – 195.
[128] Google's response to [✂].
[129] Google's response to [✂
[130] Google's response to [✂].
[131] Google's confidential response to the CMA's provisional decision report dated 22 November 2024, [✂].
[132] Google's confidential response to the CMA's provisional decision report dated 22 November 2024, [✂].
[133] Google's confidential response to the CMA's provisional decision report dated 22 November 2024, [✂].
[134] Movement for an Open Web (MOW)'s response to the CMA's provisional decision report dated 22 November 2024, pages 11-12.
[135] Movement for an Open Web (MOW)'s response to the CMA's provisional decision report dated 22 November 2024, page 12.
[136] Movement for an Open Web (MOW)'s response to the CMA's provisional decision report dated 22 November 2024, page 12. As a way to mitigate potential circumvention of potential remedy 4, MOW proposed that payments made by Google to any browser could to be capped as a percentage of revenue from Google to match what such browser vendors receive from rival search providers to Google.

have a negative impact on mobile browser competition on iOS; and (ii) the fact that the ISA is a commercial agreement covering a number of activities and relationships between Google and Apple means that intervention in relation to elements of the agreement could introduce distortions in markets which are outside the scope of this market investigation.

**Potential for prohibition of the Chrome Revenue Share to have a negative impact on mobile browser competition on iOS**

202. As set out in Section 10: Decisions on AEC(s) in the supply of mobile browsers, browser engines and in-app browsing, we have found that the ISA, individually or in combination with other features, gives rise to an AEC in the market for the supply of mobile browsers on iOS. This is because, in our view, the ISA significantly reduces Apple's and Google's financial incentives to compete in that market – a market in which Safari and Chrome account for around 99% of UK supply (with Safari accounting for 88% and Chrome for 11%).[137]

203. We note that Apple submitted that prohibition of the Chrome Revenue Share could lead to significant harms to mobile browser competition (as set out in sub-section The connection between the Chrome Revenue Share and revenue-sharing arrangements under the Safari Agreement above) and Google submitted that the ISA creates rivalry-enhancing efficiencies including [✂].

204. As regards Apple's submission that the Chrome Agreement [✂], and that the ISA secures Google Search as an input, we note that Apple [✂]. Further, it is unclear that Google would have the incentive to [✂]. We also note that it is unclear that it would be in Google's interests to [✂].

205. We also noted in Section 9: The Information Services Agreement that it is unclear that Google providing an identical search experience on Safari to the one it provides on Chrome on iOS enhances rivalry in the market for mobile browsers on iOS. On the contrary, better integration of Chrome with Google search could, in principle, competitively distinguish Chrome on iOS which could drive increased competition between Chrome and Safari on iOS.[138]

206. In relation to the Chrome Agreement providing Google [✂]. In light of this, we have considered whether there would be merit in an argument that the ISA is rivalry-enhancing in the supply of mobile browsers on iOS because it [✂].

[137] See Section 3: Market definition and market structure in the supply of mobile browsers browser engines and in-app browsing, Shares of supply sub-section.
[138] As noted in Section 9: The Information Services Agreement, [✂].

207. As noted in Section 9: The Information Services Agreement between Apple and Google, we do not consider that [✂]. Further, it is unclear that Apple would have incentive to [✂].

208. In any event, it would not be appropriate to assess claimed efficiencies of features which adversely impact competition against a benchmark that, [✂]. Rather, we assess the impact of the ISA against a benchmark of a well-functioning market in which we would not expect two close competitors, in a market where there are only a limited number of significant competitors, having revenue-sharing arrangements that significantly reduce their financial incentives to compete.

209. While we recognise that some contractual arrangements between the parties may be required in a well-functioning market to govern Apple's access to Google Search and [✂], the existing revenue-sharing arrangements – which have a significant impact on the parties' financial incentives, which are key drivers of competition – go far beyond what may be required to address the issues raised by Apple and Google. We do not consider these arrangements to be compatible with the concept of a well-functioning market in the supply of mobile browsers on iOS.

210. We do not agree with Apple's submission that this potential remedy would inhibit its ability to monetise its platform. The potential remedy would not preclude Apple from monetising its platform in return for the value it offers to app developers. The potential remedy would restrict the method by which such monetisation could occur by prohibiting Apple's main browser competitor on iOS, Google, from sharing a portion of revenue earned from qualifying searches on Chrome on iOS.

211. We consider that without removing the Chrome Revenue Share between the two largest mobile browser vendors on iOS, the effectiveness of any potential remedies package could be compromised. Potential access remedies (potential remedies 1 – 3) and choice architecture remedies (potential remedy 5) do not directly address Google's and Apple's financial incentives to compete vigorously with each other as browser vendors on iOS. Unless Google's and Apple's financial incentives to compete in the market are changed, other potential remedies may be less effective (as set out in Section 9: The Information Services Agreement), due to:

    (a) The significant impact of the ISA's revenue sharing arrangements on the parties' financial incentives to compete, as these arrangements mean that there is limited incremental value to be gained from Apple and Google winning customers from each other;

    (b) The significant magnitude of the revenue shares;

    (c) The parties' strong and stable position in the relevant market – where they are the only two players with a share over 1%; and

(d) The fact that any remaining incentives to compete in the supply of mobile browsers on iOS which may come from outside of such market are limited.

**Potential for intervention in relation to elements of the ISA introducing distortions in markets which are outside the scope of this market investigation**

212. We acknowledge that the ISA is a contractual arrangement between Apple and Google which has evolved over the years and has over time broadened from focusing on the terms of engagement in relation to Google being the default search engine on Safari to incorporating provisions relating to other search entry points, as well as Google's Chrome app.

213. This market investigation concerns the supply of mobile browsers, browser engines and cloud gaming. However, the ISA touches upon a number of the parties' activities, with mobile browsers being only one of those activities. We are conscious of the potential for intervention in relation to one element of the ISA – ie the prohibition of the Chrome Revenue Share – to introduce significant distortion risks in markets falling outside the scope of this market investigation. This is because such intervention could have implications for other activities covered by the ISA, such as the search market (including the terms for the default search engine on Safari, as well as the [✂]. Noting that search default agreements represent one of the main methods of monetising browsers,[139] we consider that any potential intervention in relation to the ISA would need to take account of the interactions between the search and browser markets. In this context, we note Apple's concerns in relation to the impact the remedy might have on Google's position in the search market.

214. Given the scope of this market investigation, our assessment of the potential remedy did not focus on the possible impact on the search market, but we recognise the close link between the mobile browser market and the search market, as noted above.

215. The new DMCC Act powers enable the CMA to carry out an SMS investigation (or multiple investigations) into one or more digital activities. As a result, the CMA is able to investigate and take account of the interplay between the markets that are the subject of this market investigation and Apple's and Google's wider mobile ecosystems in turn addressing potential distortion risks that may arise.

[139] As set out in Section 2: Nature of competition in the supply of mobile browsers, browser engines and in-app browsing.

**Prevention of Apple and Google entering into revenue-sharing agreements for future Apple and Google products that perform similar functions to a mobile browser**

216. Apple submitted that prohibiting the existing Chrome Revenue Share is unwarranted and disproportionate. In addition, Apple submitted that it would not be appropriate to attempt to 'future proof' the proposed remedy by applying it to products that do not exist.[140]

217. An important aspect of the effectiveness assessment would be to ensure that Apple and Google are not able to circumvent the objectives of the potential remedy. For example, an effective remedy would need to guard against the risk of circumvention where the parties could replace the Chrome Revenue Share with other arrangements sharing revenues in relation to a new or a renamed browser product which have similar effects in dampening the parties' incentives to compete.

218. A risk of circumvention would also arise if the parties were to shift payments previously made under the Chrome Revenue Share into the Safari Agreement – which has been considered above.

**Geographical scope of any prohibition of the Chrome Revenue Share**

219. Apple submitted that there is no basis to extend a remedy beyond the UK.[141]

220. As noted in Section 9: The Information Services Agreements between Apple and Google, the revenue-sharing provisions of the ISA significantly impact Apple's and Google's financial incentives to compete, as they mean that there is limited incremental value to be gained from Apple and Google winning customers from each other via their mobile browser on iOS. Therefore, the parties' financial incentives to compete, including via investing in Safari and Chrome respectively, are reduced compared to a situation in which the ISA is not in place.

221. To ensure effectiveness of this potential remedy, it would need to effectively address the impact of the ISA on Apple's and Google's financial incentives to compete in the supply of mobile browsers on iOS. For the reasons set out below, we consider that a prohibition of the Chrome Revenue Share limited to revenues derived from UK search advertising would be unlikely to be sufficient to achieve this aim. Limiting the prohibition to Europe may carry similar effectiveness concerns. Accordingly, we consider that it may be necessary to prohibit the Chrome Revenue Share on a wider, potentially global, basis.

[140] Apple's response to [✂].
[141] Apple's response to [✂].

222. By way of illustration, in 2023, Google paid Apple USD [✂][142] as part of the Chrome Revenue Share, which represented around [✂]% (see Table 2) of the total ISA payments paid to Apple in that year. UK search advertising will have only accounted for a small proportion of these payments. A remedy restricted to the UK would carry significant effectiveness risks considering that Apple would continue to receive very significant payments from its main competitor's mobile browser. As a result, such a remedy would be unlikely to have the intended impact on Apple's and Google's financial incentives to compete strongly.

**Table 2: Chrome Agreement and Safari Agreement global payments made by Google to Apple from [✂] on mobile devices only**

| | *[✂]* $m | *[✂]* $m | *[✂]* $m | *[✂]* $m | *[✂]* $m | *[✂]* $m | *[✂]* $m |
|---|---|---|---|---|---|---|---|
| Safari | [✂] | [✂] | [✂] | [✂] | [✂] | [✂] | [✂] |
| Chrome | [✂] | [✂] | [✂] | [✂] | [✂] | [✂] | [✂] |
| Total | [✂] | [✂] | [✂] | [✂] | [✂] | [✂] | [✂] |
| Safari/Total | [✂] | [✂] | [✂] | [✂] | [✂] | [✂] | [✂] |
| Chrome/Total | [✂] | [✂] | [✂] | [✂] | [✂] | [✂] | [✂] |

*[✂]*

223. A remedy that effectively addresses the financial benefits of the ISA, by way of prohibition of the Chrome Revenue Share, would increase Apple's incentive to improve and innovate its mobile browser Safari. This is because Apple would only earn revenues from traffic on Safari, rather than when either Safari or Chrome is used on iOS.

224. Absent the Chrome Revenue Share, Google would retain all its search advertising revenues derived through its Chrome browser on iOS (potentially driving Google to further improve and innovate its browser). We also consider that Apple would be more strongly incentivised to drive traffic to its own browser, Safari. If the Chrome Revenue Share were to be terminated, Apple's incentives to encourage its users to make and/or keep Safari as a default would likely increase.

225. Further, we consider that the implementation of potential remedy 1 (enabling alternative browser engines on iOS), potential remedy 2 (enabling equivalent access to features and functionalities for all WebKit-based browsers on iOS) and potential remedy 5 (a combination of choice architecture remedies) would be undermined should the Chrome Revenue Share continue to exist. This is because it would undermine Google's and Apple's financial incentives to strongly compete with Chrome and Safari respectively.

226. The prohibition of the Chrome Revenue Share would not preclude either party from continuing to fulfil the ISA obligations not affected by the potential remedy,

[142] Google response to the CMA's [✂].

including the arrangements relating to Google's search engine being the default on the Safari browser.

**Conclusions on potential remedy 4**

227. As noted above under the discussion of key remedy design considerations, there are a number of risks to the effectiveness of this potential remedy, if implemented through the remedy-making provisions of EA02.

228. These relate to:

(a) **Distortion**: a potential measure prohibiting one element of the ISA (ie the Chrome Revenue Share) could risk introducing significant distortions in markets which are closely connected with those which form the subject-matter of this market investigation but which are outside of its scope (eg the search market).

(b) **Circumvention**: the risk of Apple and Google circumventing the potential remedy by entering into revenue-sharing agreements for future Apple and Google products that perform similar functions to a mobile browser.

229. We conclude that, taken together, these risks mean that there is a significant risk to the effectiveness of potential remedy 4 in addressing AEC 2.

## Potential remedies 5 and 6 addressing AECs 2 and 4

230. Section 8: The role of choice architecture in mobile browsers, considered whether the use of choice architecture on iOS and Android devices reduces user awareness, engagement and choice, and encourages the use of Safari and Chrome for browsing, increasing barriers to entry and expansion for third-party browser vendors.

231. Potential remedies 5 and 6 relate to Apple and Google's control of choice architecture on iOS and Android devices, respectively:

(a) Potential remedy 5 would aim to address AEC 2 with respect to Apple's control over choice architecture on iOS devices and low user awareness in relation to browser choice.

(b) Potential remedy 6 would aim to address AEC 4 with respect to Google's control over choice architecture on Android and low user awareness in relation to browser choice.

232. In this section, we provide a description of potential remedies 5 and 6 and outline how they would seek to address AECs 2 and 4 and the resulting customer detriment, before outlining key considerations in relation to their design and implementation.

### Description of potential remedies 5 and 6

233. As set out in Section 10: Decisions on AEC(s) in the supply of mobile browsers, browser engines and in-app browsing, we have found that Apple and, to a lesser extent, Google's respective use of choice architecture practices constitute features, which individually or in combination with other features, give rise to AECs.

234. The two key stages in a consumer's engagement with mobile browsers on their smartphone are:

(a) The factory settings set on a device for first use, ie the pre-installation, prominent placement and default setting of mobile browsers.

(b) Practices used after initial device set-up, ie the chosen default browser, the user journey for changing a default browser via a device settings menu and prompts to change the default browser.

235. Potential remedies 5 and 6 would aim to address the control Apple and Google have over iOS and Android choice architecture as well as the contribution of current choice architecture to low user awareness of alternative mobile browsers.

## Potential remedy 5: Apple

236. Potential remedy 5 comprises a set of requirements that would seek to address Apple's control of choice architecture on iOS devices in device factory settings, Apple's control of choice architecture on iOS devices after the point of device set-up and users' low awareness and engagement with mobile browsers. These requirements, as detailed in Table 3, are set out below:

(a) Potential remedies 5a and 5b would address choice architecture at device set-up:

(i) Potential remedy 5a would require a browser choice screen for new users, allowing new device users to select a default mobile browser to install from several options.

(ii) Potential remedy 5b would ensure that the selected mobile browser is prominently placed and easily accessible by the user in the application dock/'hotseat', or elsewhere on the default home screen.

(b) Potential remedies 5c-e would address iOS choice architecture after device set-up:

(i) Similar to requirement 5a, potential remedy 5c would require a browser choice screen for existing iOS device users after the point of device set-up.

(ii) Potential remedy 5d would require Apple to provide an API allowing browser vendors to see when their mobile browser is set as default.

(iii) Potential remedy 5e would restrict the frequency with which all browser vendors could show a prompt linking users directly to the settings to switch default browser across multiple access points on iOS devices.

**Table 3: Description of potential remedy 5 addressing AEC 2**

| *Potential Remedy* | *Potential remedy description* | *Relevant features* |
|---|---|---|
| 5a | A requirement for Apple to ensure the use of a browser choice screen at device set-up. | • Apple's control over choice architecture in the factory settings for iOS devices on first use of browsers<br>• Users' low awareness and engagement with mobile browsers |
| 5b | A requirement for Apple to ensure the placement of a default browser selected by the user in the 'application dock'/'hotseat' or on the default home screen[143] at device set-up. | |

[143] The 'default home screen' refers to the initial screen that the user sees when unlocking their device.

| | | |
|---|---|---|
| 5c | A requirement for Apple to ensure the use of a browser choice screen after device set-up. | • Apple's use of choice architecture practices after the point of device set-up for mobile browsers<br>• Users' low awareness and engagement with mobile browsers |
| 5d | A requirement for Apple to share user data on default browsers settings with browser vendors. | |
| 5e | A requirement for Apple to ensure that the frequency of default browser prompts and notifications is limited across multiple access points. | |

*Source: CMA analysis*

## Potential remedy 6: Google

237. Potential remedy 6 would address Google's control over choice architecture on Android devices in device factory settings as well as users' low awareness and engagement with mobile browsers (see Table 4):

(a) Potential remedies (6a and 6b) would address Google's control of choice architecture in factory settings: a browser choice screen shown to new users at device set-up (6a) and the placement of the selected browser in the application dock/'hotseat' (or elsewhere on the default home screen) at device set-up (6b).

(b) Potential remedies (6c and 6d) would address users' low awareness and engagement with mobile browsers through a requirement to show a browser choice screen to existing Android users (6c) and a requirement for Google to ensure that a restriction is implemented in relation to the frequency with which all browser vendors can use a prompt to change default browser across multiple access points (6d).

**Table 4: Description of potential remedy 6**

| *Potential remedy* | *Potential remedy description* | *Relevant features* |
|---|---|---|
| 6a | A requirement for Google to ensure the use of a browser choice screen at device set-up. | • Google's control over choice architecture in the factory settings for device on first use of browsers<br>• Users' low awareness and engagement with mobile browsers |
| 6b | A requirement for Google to ensure the placement of a default browser selected by the user in the 'dock'/'hotseat' or on the default home screen at device set-up. | |
| 6c | A requirement for Google to ensure the use of a browser choice screen after device set-up. | • Users' low awareness and engagement with mobile browsers |
| 6d | A requirement for Google to ensure that the frequency of default browser prompts and notifications is limited across multiple access points. | |

*Source: CMA analysis*

## How potential remedies 5 and 6 would seek to address the AECs and customer detriment

238. As detailed in Section 8: The role of choice architecture in mobile browsers, the current choice architecture on iOS and Android contributes to low user awareness of other browsing options and encourages user inertia with respect to browser choice. Low user awareness and low user engagement reduces incentives for browser vendors to compete effectively on mobile browser quality, such that users may receive lower quality products in the long run.

239. Potential remedies 5 and 6 would seek to address the choice architecture features of AEC 2 in relation to Apple and AEC 4 in relation to Google. These potential remedies would aim to do so by raising user awareness of alternative mobile browsers, encouraging active browser choice and therefore engagement with the mobile browser market, and ensuring their browser choices are respected (including easy access to their preferred mobile browser on the application dock/'hotseat' or default home screen). Potential remedies 5 and 6 would also aim to provide browser vendors with the tools required to engage with users more effectively and therefore increase their ability and incentives to compete.

240. Requirements in the device factory settings on first use of mobile browsers (potential remedies 5a and 5b on iOS and potential remedies 6a and 6b on Android) would implement a browser choice screen which can increase user awareness of alternative mobile browsers and ensure that new users are able to make a choice about their preferred mobile browser. We would expect encouraging this active choice to reduce user inertia with respect to relying on pre-installed apps and pre-set defaults. Increased user engagement would, in turn, increase competitive pressure in the mobile browser markets.

241. In addition, potential remedies 5b and 6b would seek to ensure that the mobile browser selected by users was prominently placed and easily accessible. While default setting of the mobile browser is important in cases where users are directed to a browser via web links, users can also manually open a mobile browser on their device home screen. Placing the user's selected mobile browser prominently would reduce friction to use that browser and would reduce user inertia to rely on Safari and Chrome (which are placed on the default home screen in the device factory settings) for manual browsing.

242. These potential remedies would also include several requirements after the point of device set-up to ensure that users are able to engage with mobile browsers effectively and that they can easily switch to their preferred mobile browser.

243. For example, potential remedies 5c and 6c would require implementation of a choice screen for existing users. As with the choice screen at device set-up, a choice screen on existing devices would ensure that device users could engage

effectively with mobile browsers and enable them to make choices about their preferred mobile browser beyond the initial set-up of the device, providing a choice point that is not dependent on purchasing a new device.

244. Potential remedy 5d on iOS would require Apple to provide browser vendors with information about when their mobile browser is set as default. This requirement would allow browser vendors to target users more effectively, thereby increasing their ability and incentives to compete and removing the risk of showing a prompt to users when they have already set a mobile browser as default.

245. Potential remedies 5e and 6d would restrict the frequency with which all browser vendors could show a prompt linking users directly to the settings to switch default mobile browser across multiple access points. These requirements would ensure that browser vendors could engage users effectively through prompts, which could increase user awareness of alternative mobile browser options. This would also help to maintain a satisfactory user experience by minimising unnecessary friction, such that users are not prompted to switch browser across various access points, following a choice already having been made.

246. Potential remedy 6 would require similar options at factory settings on Android devices as potential remedy 5 would require for iOS devices – namely, a browser choice screen shown to new users at device set-up (potential remedy 6a) and the placement of the selected mobile browser in the application dock/hot seat (or on the default home screen) at device set-up (potential remedy 6b).

247. Potential remedy 6 would also include, after device set-up, a requirement to show a browser choice screen to existing Android users (potential remedy 6c) and a requirement for Google to ensure that a restriction is implemented in relation to the frequency with which browser vendors can use prompts to change default browser across multiple access points (potential remedy 6d).

248. However, we consider that there are a number of risks to the effectiveness of these potential remedies if implemented through the remedy-making provisions of the EA02. Taken together, they amount to a significant risk to the effectiveness of these potential remedies in addressing AEC 2 (for Apple) and AEC 4 (for Google) and resulting customer detriment. We set this out in further detail in the section below.

## Key remedy design considerations

249. We set out below an assessment of whether the potential remedies described above would be effective and the key remedy design considerations that would be relevant in this respect. In particular, effective remedies in relation to choice architecture would require:

(a) user-centred design principles which would need to be taken into account when designing any choice architecture remedies, including targeted, understandable and balanced principles;[144]

(b) a clear pathway to implementation of choice architecture remedies considering Apple's and Google's capabilities and infrastructure already in place as well as regulatory alignment with other jurisdictions; and

(c) testing and trialling before implementation to maximise the prospect that the remedies would be effective in achieving their intended aims.

### 1. User-centred principles for remedy design including targeted, understandable and balanced principles

250. The effectiveness of the choice architecture requirements under potential remedies 5 and 6 is likely to be dependent on adequate, user-centred design. Effective user-centred design would ensure that users are presented with choices at the right place, at the right time and with the right frequency to make active choices that they can understand and action. User-centred design aims to give users autonomy over their choices, rather than guiding their choices to a particular outcome, and ensures that unjustified friction is minimised where possible, so that user choice is actionable and practicable.

251. In WP7, we set out three design principles which would need to be taken into account in the design of any choice architecture remedies – namely, that the remedy should be targeted, understandable and balanced.[145] We note that Apple and Google would need to apply these principles in any implementation of potential choice architecture remedies.

**Choice architecture remedies in the factory settings set on a device for first use (potential remedies 5a-b and 6a-b)**

252. Certain design considerations such as timing, frequency and location (amongst others), are likely to substantially affect the effectiveness of choice screens.

253. A number of parties submitted representations regarding the effectiveness of choice screens in response to WP7:

(a) Apple submitted that introducing a choice screen in the device factory settings on first use of mobile browsers (potential remedy 5a) 'would create a jarring and confusing user experience' and stated that choice screens raise difficult design questions such as timing and the criteria for determining which browsers are included. Apple highlighted the potential for unintended harms

[144] CMA Working paper 7: Potential remedies, page 49.
[145] CMA Working paper 7: Potential remedies, page 49.

associated with choice screen and placement requirements (potential remedies 5a-c and 6a-c) such as the exclusion of smaller competing browsers and the reinforcement of the market position of larger competitors.[146]

(b) Google highlighted some parameters of choice screens, that if not properly designed and adjusted, could undermine their effectiveness. Relevant considerations noted by Google were the position in the user journey, the number of browsers shown on the choice screen, the amount of information shown about each browser and the frequency of presentation.[147]

(c) Google raised concerns relating to potential remedy 6b, stating that Android OEMs decide the logic of where an app is placed when it is downloaded.[148] Potential remedy 6b would require that a browser already placed in the hot seat (if one is positioned there at all), should be 'swapped out' for the one selected from the choice screen. As discussed in Section 8: The role of choice architecture in mobile browsers (see sub-section Choice architecture practices in the device factory settings at first use), we note that Android devices typically include a mobile browser in the application dock or on the default home screen in factory settings, and therefore potential remedy 6b is unlikely to significantly impact the freedom of OEMs to customise their devices.

(d) Several third parties expressed support for choice screen remedies (potential remedies 5a/c and 6a/c), but also highlighted design considerations that should be taken into account.[149] For example, Mozilla and Vivaldi submitted that showing the choice screen at initial device set-up and after major software updates would be most effective.[150] Vivaldi submitted that a choice screen that displays on first launch of a browser is more intrusive on users and gives incumbents an unfair advantage.[151] It also expressed concerns about which browsers should be included on the choice screen, suggesting that cross-platform browsers, browsers that compile their own code and browsers that update more frequently should all be given priority on choice screens.[152]

---

[146] Apple's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 79.
[147] Google's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 66.
[148] Google's response to Working Paper 7: Potential Remedies, 8 August 2024, paragraph 67.
[149] Mozilla's response to Working Paper 7: Potential Remedies, 8 August 2024; Vivaldi's response to Working Paper 7: Potential Remedies, 8 August 2024; DuckDuckGo's response to Working Paper 7: Potential Remedies, 8 August 2024; OWA's response to Working Paper 7: Potential Remedies, 8 August 2024.
[150] Mozilla's response to Working Paper 7: Potential Remedies, 8 August 2024; Vivaldi's response to Working Paper 7: Potential Remedies, 8 August 2024.
[151] Vivaldi's response to Working Paper 7: Potential Remedies, 8 August 2024.
[152] Vivaldi's response to Working Paper 7: Potential Remedies, 8 August 2024.

(e) DuckDuckGo submitted that the design of choice screens before and after device set-up should be as similar as possible.[153] DuckDuckGo also submitted that all users of new Android devices should see the choice screen and not just those whose default browser is set to Chrome. It noted that this is necessary due to Chrome's strong market position and the prevalence with which it is pre-installed on Android devices.[154]

(f) However, Samsung raised concerns over the impact that the combination of potential remedies 6a and 6b would have on browser usage on Android devices. Samsung noted that implementing potential remedies 6a and 6b would further promote Chrome's usage on Android and result in a strengthened market share for Chrome. Samsung therefore submitted that potential remedies 6a and 6b should only apply to devices where Chrome is currently set as the default browser, as is the case for the DMA mandated choice screen.[155]

(g) Samsung also questioned the degree to which potential remedies 6a and 6b can be readily implemented and enforced given that Google has no authority to implement a browser choice screen or adjust the default home screens of OEMs, unless the Android OEM agrees to do so.[156] The App Association (ACT) raised a concern that choice screens can solidify the dominance of already powerful players in the market and do little to benefit consumers or smaller players.[157]

254. In addition, while there was some support for a potential remedy requiring prominent placement of a default mobile browser (potential remedies 5b and 6b),[158] some parties have questioned why we do not consider placement requirements for the choice screen after device set-up as well (in addition to potential remedies 5c and 6c).[159] We did not consider it appropriate to explore any placement requirements for existing users after device set-up as this can interfere with, or potentially override, existing user app customisation on the device home screen (ie where existing users have after the initial device set up chosen to place a different apps in the 'hotseat'/application dock).

255. In response to the PDR, a number of parties made further representations on the effectiveness and potential impact of choice screens:

---

[153] DuckDuckGo's response to Working Paper 7: Potential Remedies, 8 August 2024.
[154] DuckDuckGo's response to Working Paper 7: Potential Remedies, 8 August 2024, page 2.
[155] Samsung's response to Working Paper 7: Potential Remedies, 8 August 2024.
[156] Samsung's response to Working Paper 7: Potential Remedies, 8 August 2024.
[157] App Association's response to Working Paper 7: Potential Remedies, 8 August 2024, page 2.
[158] Mozilla's response to Working Paper 7: Potential Remedies, 8 August 2024; Vivaldi's response to Working Paper 7: Potential Remedies, 8 August 2024; DuckDuckGo's response to Working Paper 7: Potential Remedies, 8 August 2024; OWA's response to Working Paper 7: Potential Remedies, 8 August 2024, Mozilla's response to the provisional decision report, 22 November 2024, page 7.
[159] DuckDuckGo's response to Working Paper 7: Potential Remedies, dated 8 August 2024; OWA's response to Working Paper 7: Potential Remedies, 8 August 2024.

(a) Apple submitted that user experience at device set-up may be degraded by the introduction of a browser choice screen. Apple raised the concern that users presented with unfamiliar mobile browser options 'may select a browser at random in order to get on with using their device' only to realise later that they want to return to their 'original browser'. Apple cited independent academic research that indicates when consumers are faced with options for which they are likely to make a welfare-reducing choice, and where the consumer perceives a significant cost to making that choice then they may be better off with a default option.[160, 161]

(b) Apple further submitted that independent studies into the browser choice screen that was implemented in the EU between 2010 and 2014 to address the pre-installation and setting as default of Internet Explorer on Microsoft's operating system demonstrate that the choice screen had limited impact on the market share of Internet Explorer.[162]

(c) Google argued that any choice screen requirement should apply symmetrically to cover all devices with a default browser, irrespective of whether the default is Chrome.[163]

(d) Google also submitted that, where appropriate any potential remedies should take account of the DMA compliance measures. Certain remedies are costly to design, test and roll-out and appropriately taken regulatory alignment can reduce these costs.[164]

(e) An OEM stated that potential remedies 6a-c go beyond addressing AEC4 as they also apply to choice architecture that OEMs control regardless of whether the OEM has a commercial arrangement with Google to set Chrome as the default and place in the 'hotseat'. The proposed remedies would also 'prevent an Android OEM from entering into an agreement with a third-party browser vendor' therefore, preventing those challenger browsers from gaining market share via such deals.[165]

(f) Mozilla suggested an alternative remedy which would require existing users to be offered a choice screen for browser placement that would be linked to remedy 5c and 6c. The user would be asked if they wish to replace the existing browser in their application dock with their new default browser selected via the choice screen.[166] We note that it is not clear how such a choice screen would work in scenarios where the user had either more than

[160] Goldin, J. & Reck, D. (2022), 'Optimal Defaults with Normative Ambiguity', The Review of Economics and Statistics, 104 (1), pp. 17–33.
[161] Apple's response to the provisional decision report, 22 November 2024, page 24
[162] Apple's response to the provisional decision report, 22 November 2024, page 24
[163] Google's response to the provisional decision report, 19 December 2024, page 29
[164] Google's response to the provisional decision report, 19 December 2024, page 30
[165] [✂] OEM response to the provisional decision report, 22 November 2024, page 5
[166] Mozilla's response to the provisional decision report, 21 December 2024, page 8

one browser or no browser placed in their application dock. Furthermore, it is our view that any potential remedy that addresses the placement of default browser for existing users does not interfere or override with the user's customisation of their default home screen.

**Choice architecture remedies after device set-up (potential remedies 5c-e and 6c-d)**

256. We consider that design across the choice screens shown at and after the device set-up (potential remedies 5a and 5c and 6a and 6c) would need to be largely consistent, apart from the time at which the choice screen would be displayed (eg for new users this would be shown at the initial device set-up, while for existing users it would be shown either the first time a user opens Chrome or Safari, or immediately after the release of a OS update).

257. In relation to use of prompts, we note that potential remedies 5d and 5e together aim to allow browser vendors to engage with users without sending unnecessary prompts across multiple access points – that is, limiting the frequency with which browser vendors can send prompts across browsers and other access points and helping vendors to target users who have not already set their mobile browser as default.

258. The effective design of these potential remedies would have to take into account user experience in relation to frequency of prompts, as well as the needs of browser vendors to effectively engage with users, ensuring that the prompts are shown at the right time, at the right place and at the right intervals.

259. In response to WP7, we received the following submissions from parties regarding potential remedies 5d-e and 6d:

(a) Apple highlighted the risks to user experience from browsers frequently prompting users to change default and stated that it would be more effective for the CMA to avoid requiring unnecessary prompts rather than encouraging them and then attempting to limit their usage.[167]

(b) Some browser vendors have expressed support for potential remedies 5e and 6d, stating that this requirement would ensure that neither Apple nor Google can leverage their control of their respective operating systems to self-preference in relation to browser prompts.[168]

[167] Apple's response to CMA's Working Paper 7 Potential Remedies, 8 August 2024, paragraph 85.
[168] Mozilla's response to CMA's Working Paper 7: Potential Remedies, 8 August 2024; Vivaldi's response to CMA's Working Paper 7: Potential Remedies, 8 August 2024.

(c) DuckDuckGo cautioned that the requirement should not put regulated firms on an equal footing with third parties, pointing to Google's use of browser switching prompts via other services (eg YouTube and Gmail).[169]

(d) OWA also stated that this requirement should explicitly set a frequency limit to the prompts that all browser vendors can use.[170]

260. In response to the PDR, we received the following submissions:

(a) Mozilla submitted that while restrictions on the frequency of prompting should apply equally to Apple and Google as to other browser vendors, it is important to recognise that Apple and Google are much less likely to benefit from such prompts given the respective positions of Safari and Chrome as pre-installed browsers set as default.[171]

(b) Mozilla further submitted that the drafting of remedies 5e and 6d should 'prohibit any prompts at the point of a user using one of Google's or Apple's other applications, such as Gmail, Google Maps, Mail or Apple Maps, following an active choice having already been made by the user to select an alternative default browser'.[172]

(c) OWA made a similar representation, advocating that Google should be banned from leveraging its other properties on Android to prompt users to switch default browser to Chrome.[173]

261. Overall, we acknowledge the need for a balanced approach when determining the design of remedies in relation to the use of prompts, to ensure that prompts shown by browser vendors are used effectively within the mobile browser and limit the use of prompts across other access points.

### 2. A clear pathway to implementation of choice architecture remedies considering Apple and Google's capabilities, infrastructure already in place and regulatory alignment with other jurisdictions

262. We consider that implementing choice screens would be technically feasible for both Apple and Google.

263. In the case of Apple, its control over both its operating system and mobile devices would ensure the feasibility of distributing choice screens before and after initial device set-up and of implementing the other choice architecture requirements

[169] DuckDuckGo's response to CMA's Working Paper 7: Potential Remedies, 8 August 2024.
[170] OWA's response to CMA's Working Paper 7: Potential Remedies, 8 August 2024.
[171] Mozilla's response to the provisional decision report, 21 December 2024, page 9
[172] Mozilla's response to the provisional decision report, 21 December 2024, page 9
[173] OWA's response to the provisional decision report, 19 December 2024, page 35

under potential remedy 5. We anticipate that this could be done at manufacture for new devices, and via OS updates for existing devices.

264. Google's ability to widely implement choice architecture updates across the Android operating system is dependent on cooperation of Android OEMs. Google has also raised concerns relating to potential remedy 6b, submitting that Android OEMs decide the logic of where an app is placed when it is downloaded. Potential remedy 6b would require that a browser already placed in the 'hotseat'/application dock (if one is positioned there at all), should be 'swapped out' for the one selected from the choice screen.[174]

265. However, Google has indicated that it has an existing framework to distribute the DMA browser choice screen on newly activated device models to Android OEMs as part of the proprietary suite of apps known as Google Mobile Services (GMS) – the Compatibility Test Suite that OEMs must comply with ensures that OEMs have properly implemented the apps and services that are part of the GMS.[175] We expect that distribution of the choice screen, placement requirements and other choice architecture requirements under potential remedy 6 would be able to follow a similar approach.

266. In response to the PDR, an OEM submitted that contrary to the suggestion in the PDR, it does not believe that the Google's compliance efforts to institute a choice screen in the EU provides 'a clear pathway to implementation' which achieves regulatory alignment between potential remedies 6a-6c and those imposed by the European Commission. This is because the potential choice screen remedy under consideration by the CMA could apply to all Android OEMs and risk the creation of two separate technical methods of compliance.[176]

267. [✂].[177] Several OEMs have confirmed that they have been working with Google to ensure the choice screen is implemented on their devices with recent or near-future software updates.[178]

268. Responses we have received from several third parties indicate that existing devices beyond a certain age (approximately three years old) do not receive updates and therefore users with older Android devices will not see any choice architecture changes implemented.[179]

[174] Google's response to CMA's Working paper 7: Potential Remedies, 8 August 2024, paragraph 67.
[175] Google's response to CMA's information request [✂].
[176] [✂] OEM response to the provisional decision report, 22 November 2024, page 7.
[177] [✂] response to the CMA's information request [✂].
[178] Responses to the CMA's information requests: [✂].
[179] Responses to the CMA's information requests [✂].

269. We are aware that both Apple and Google currently have browser choice screens implemented on iOS and Android devices in the EU, in compliance with the DMA, which includes placing the selected mobile browser in the application dock.[180]

270. In addition to the browser choice screen implemented in compliance with the DMA, Google has implemented choice screens in compliance with previous European Commission decisions:

(a) Since April 2019, Google has presented a dual choice screen to Android users in Europe with an option to install additional browsers and search engines from a list of five options.[181] The choice screen is presented to users the first time they open the Play Store following an update.

(b) Since 2019, Google has implemented a search engine choice screen for general search providers on all new Android phones and tablets shipped into the EU and the UK where the Google Search app is pre-installed, with an option to set the default.[182]

271. Both Apple and Google are also subject to compliance with Article 6(3) of the DMA, which ensures that users can easily switch default services, including modification of the user journey to switch default browser.

272. Therefore, we consider that both Apple and Google have considerable capability and infrastructure already in place to implement potential remedies 5a, 5b and 5c and 6a, 6b and 6c respectively. In addition, both Apple and Google also collect and provide data relating to browser choice screen performance to other parties, including browser vendors. Therefore, we would expect that effectiveness of the choice screen required under potential remedies 5a, 5c, 6a and 6c would be monitorable in a similar manner.

273. As noted above, we would expect that any roll-out of these potential remedies to non-Google Android devices, as well as distribution of the choice screen and other choice architecture requirements under potential remedy 6 could use the existing framework Google has in place to distribute the DMA choice screen.

274. However, we would also expect the timeline for potential remedy 6 to be longer, as further distribution of choice remedies on Android devices by OEMs will be dependent on their manufacturing and development resource. For example, some Android OEMs have submitted that the roll-out of the choice screen update on their devices is dependent on their manufacturing and release schedules, which did not immediately align with the release of the choice screen.[183]

---

[180] About the browser choice screen in the EU - Support - Apple Developer.
[181] Presenting search app and browser options to Android users in Europe (blog.google).
[182] Android Choice Screen.
[183] Responses to the CMA's information requests: [✂].

275. Apple submitted in relation to potential remedy 5a that prompting the user to select and install a browser at device set-up would be challenging, given that downloading a browser (or any other app) from the App Store requires an Apple ID as well as an internet connection.[184] However, we consider that these concerns could be mitigated via the sequencing of device set-up, such that setting up the App Store and internet connection would occur before display of the choice screen.

276. We consider that potential remedy 5d to provide data access to browser vendors to know when their mobile browser is set as default could be implemented as an API that browser vendors call to indicate whether their browser is currently set as default on a device. There is currently an API implementation on Android, which functions in this way.

277. The Information Commissioner's Office (ICO) submitted that any requirement to allow browser developers to store or gain access to information stored on a user's device (such as which browser is set as the default) would need to be compliant with Regulation 6 of the Privacy and Electronic Communications Regulations 2003 (PECR),[185] which states the following:

> (a) Subject to paragraph (4), a person shall not store or gain access to information stored, in the terminal equipment of a subscriber or user unless the requirements of paragraph (2) are met.
>
> (b) The requirements are that the subscriber or user of that terminal equipment –
>
> (i) is provided with clear and comprehensive information about the purposes of the storage of, or access to, that information; and
>
> (ii) has given his or her consent.

278. This consideration is relevant for potential remedy 5d. We note that consent to share this information could be obtained at the operating-system level as part of the potential remedy package – for example, when the user selects a mobile browser to download from the choice screen or App Store.[186]

279. In response to the PDR, Mozilla submitted that, if the consent for browser vendors to access a user's default browser status were to be obtained at the operating system level, then stakeholders such as Mozilla should be consulted on the design and implementation of such consent models to ensure that firms do not present

[184] Apple's response to the CMA's information request [✂].
[185] Note from meeting with the ICO, 20 September 2024, paragraph 5.
[186] Note from meeting with the ICO, 20 September 2024, paragraph 11-12.

the choice to users in such a way as to discourage them from changing their default browser.[187]

280. Potential remedies 5e and 6d, which would require Apple and Google to restrict the frequency of prompts browser vendors can show to users would rely on existing iOS and Android infrastructure. Browser vendors are currently able to show an operating system level prompt window that links users from the prompt in the app to the setting to change default browser – potential remedies 5 and 6 would enable Apple and Google to limit the number of times browser vendors can call the OS-level prompt within a specified time window as well as across other access points such as Google's or Apple's first-party apps, or prompts shown when accessing web content (eg via Google Search or Maps).

281. We consider that the choice architecture requirements proposed under potential remedies 5 and 6 could be implemented by Apple and Google within 12 months.

### 3. Testing and trialling of certain choice architecture remedies before implementation in order to maximise the remedies' effectiveness

282. As set out in Section 8: The role of choice architecture on competition in the supply of mobile browsers, users of mobile devices are presented with choice architecture which affects the presentation and placement of mobile browsers and the design of choices that a user may make between different browsers.[188]

283. Exactly how choices are presented to users can therefore have a substantial impact on the choices such users make. We consider that choice architecture remedies would, therefore, benefit from testing and trialling before being implemented to maximise the prospect that they would be effective in achieving their intended aims. We expect that testing and trialling would require an iterative process to determine effectiveness and reduce risk.

## Conclusions on potential remedies 5 and 6

284. As noted above under the discussion of key remedy design considerations, there are a number of risks to the effectiveness of these remedies if implemented through the remedy-making provisions of the EA02.

285. These relate to:

(a) **Specification**: this arises within the context of designing remedies that rely on user interaction. There are risks involved in designing effective user-based interventions without testing and trialling these with users in

[187] Mozilla's response to the provisional decision report, 22 November 2024, page 10.
[188] CMA's Working Paper 5: The role of choice architecture on competition in the supply of mobile browsers.

advance.[189] This would be compounded if it were not possible to iterate choice architecture requirements on firms in response to consumer behaviour and/or market changes; and

(b) **Ongoing monitoring:** this in turn would require subsequent ongoing monitoring and enforcement to ensure that any changes made to choice architecture remedies following test and trialling were adequately implemented by Apple and Google and to ensure that implementation continued to be compliant following iterations of the requirements.

286. We conclude that, taken together, these risks mean that there is a significant risk to the effectiveness of potential remedies 5 and 6 in addressing AECs 2 and 4.

[189] In this context we note for completeness that since 1 January 2025 the remedy-making provisions of EA02 include an ability to conduct implementation trials for remedies relating to the provision or publication of information to consumers (sections 161B to 161E EA02). The various test and trial functions under the digital markets competition regime are described in the sub-section Powers to test and trial potential interventions in section 11. They include the power to require SMS-designated firms to perform specified demonstrations or tests, to require such firms to vary their usual conduct, eg to assess the effect of different choice architecture practices, and in relation to PCIs, to test and trial different remedies or remedy design options to gain practical evidence on their effectiveness.