MOBILE BROWSERS AND CLOUD GAMING MARKET INVESTIGATION

WP7: Potential Remedies

8 August 2024

Competition & Markets Authority

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

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

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

Website: www.gov.uk/cma

The Competition and Markets Authority has excluded from this published version of the working paper information which the inquiry group considers should be excluded having regard to the three considerations set out in section 244 of the Enterprise Act 2002 (specified information: considerations relevant to disclosure). The omissions are indicated by [✂].

Contents

Tables

Figures

1. Introduction

1.1 This working paper provides our emerging views on the type of remedies that may be appropriate if we were to find an adverse effect on competition (AEC), or AECs, resulting from one or more of the features identified in our market investigation into the supply of mobile browsers and mobile browser engines, and the distribution of cloud gaming services through app stores on mobile devices (and the supply of related ancillary goods and services) in the United Kingdom.

1.2 We are at an early stage of considering potential remedies and as our understanding of the market(s) and the potential issues develops, we expect our consideration of potential remedies to evolve. As set out in the CMA's guidance,[1] we will consider and discuss potential remedies alongside working on understanding what features may give rise to adverse effects.

1.3 Where relevant, we have set out the evidence that we have gathered to date in our investigation across a series of working papers which consider competition in mobile browsers, mobile browser engines and cloud gaming services. This paper should be read alongside these other working papers:

(a) 'WP1: The nature of competition in the supply of mobile browsers and browser engines';

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

(c) 'WP3: Access to browser functionalities within the iOS and Android mobile ecosystems';

(d) 'WP4: In-app browsing within the iOS and Android mobile ecosystems';

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

(f) 'WP6: Cloud gaming services - Nature of competition and requirements for native apps on mobile devices'.

1.4 The purpose of this working paper is to:

(a) set out our framework for the assessment and selection of remedies;

(b) discuss cross-cutting considerations for potential remedies, including those that have the potential to address more than one of the potential issues;

[1] CMA3 (Revised), Market Studies and Market Investigations: Supplemental guidance on the CMA's approach, paragraph 3.50.

(c) set out the types of remedies that may be available and issues relating to the composition of any package of remedies; and

(d) summarise the potential remedies currently under consideration linked to each potential issue identified.

1.5 We invite interested parties to make submissions on any element of this working paper, including our emerging thinking on the wider, cross-cutting considerations and the different types of potential remedies discussed.

1.6 This paper does not include consideration of any consultation responses received relating to the earlier working papers. We will consider all responses received as we refine and update our thinking on potential remedies.

2. Framework for our assessment of potential remedies

2.1 If we find any AEC(s), we are required to decide the following questions:

(a) whether we should take action for the purpose of remedying, mitigating or preventing the AEC(s) or any detrimental effect on customers so far as it has resulted from, or may be expected to result from, the AEC(s);

(b) whether we should recommend the taking of action by others for those purposes; and

(c) in either case, if action should be taken, what action should be taken and what is to be remedied mitigated or prevented.[2]

2.2 A detrimental effect on customers is defined as one taking the form of:

(a) higher prices, lower quality or less choice of goods and services in any market in the UK (whether or not the market to which the feature or features concerned related); or

(b) less innovation in relation to such goods or services.[3]

2.3 When deciding whether any remedial action should be taken and, if, so, what action should be taken, the Enterprise Act 2002 requires the CMA 'in particular to have regard to the need to achieve as comprehensive a solution as is reasonable and practicable' to the AEC(s) and any detrimental effects on customers so far as resulting from the AEC(s).[4]

2.4 As part of the requirement for the CMA to achieve 'as comprehensive a solution as is reasonable and practicable' it will identify and consider potential remedies that may be effective and proportionate in addressing potential AEC(s) and any detrimental effects on customers.[5] The CMA may also take into account, in accordance with the Enterprise Act 2002, any relevant customer benefits (RCBs) of the market feature or features giving rise to the AEC(s).[6]

2.5 It is the CMA's preference to comprehensively deal with the cause or causes of AECs wherever possible, and by this means significantly increase competitive pressures in a market within a reasonable period of time. Remedies that are effective in generating competition are likely to deliver substantial benefits, by

[2] Enterprise Act 2002, section 134(4).
[3] Enterprise Act 2002, section 134(5).
[4] Enterprise Act 2002, section 134(6).
[5] CC3 (Revised), Guidelines for market investigations, paragraph 329.
[6] Enterprise Act 2002, section 134(7); CC3 (Revised), Guidelines for market investigations, paragraph 329.

driving down prices and costs and increasing innovation and productivity, thereby facilitating economic growth and increasing the choice available to customers.[7]

2.6 In assessing potential remedies, we consider their effectiveness and proportionality. With respect to effectiveness, we highlight that:

(a) we consider the risks associated with different potential remedies and will tend to favour remedies that have a higher likelihood of achieving their intended effect;

(b) a remedy should be capable of effective implementation, monitoring and enforcement. To facilitate this, the operation and implications of the remedy need to be clear to the persons to whom it is directed and also to other interested persons, such as customers, other businesses that may be affected by the remedy, sectoral regulators and/or any other body which has responsibility for monitoring compliance;

(c) we will generally look for remedies that prevent an AEC by extinguishing its causes, or that can otherwise be sustained for as long as the AEC is expected to endure. We also tend to favour potential remedies that are expected to show results within a relatively short time;[8]

(d) where more than one measure is being introduced as part of a package of remedies, we will consider the way in which the measures are expected to interact with each other.[9]

2.7 Beneficial effects might include lower prices, higher quality products/services and/or greater innovation, while the potential negative effects of a remedy may arise in various forms, for example:

(a) unintended distortions to market outcomes, which may reduce economic efficiency (including dynamic incentives to invest and innovate) and adversely affect the economic interests of customers over the longer term;

(b) implementation costs, ongoing compliance costs, and monitoring costs (for example, the costs to the CMA or other agencies in monitoring compliance); and

(c) if remedies extinguish RCBs, the amount of RCBs foregone may be considered to be a relevant cost of the remedy.

---

[7] CC3 (Revised), Guidelines for market investigations, paragraph 331.
[8] The CMA may also consider including a sunset clause in a remedy where the AEC is expected to be time limited, or may alternatively specify the circumstances in which it may be appropriate to review the functioning of or requirement for a given remedy.
[9] CMA3 (Revised), Market Studies and Market Investigations: Supplemental guidance on the CMA's approach, paragraphs 4.15 to 4.24.

2.8 In deciding what action to take at the end of a market investigation, the CMA will also typically consider whether to tackle all of the features it has identified, or whether tackling a subset of features would be sufficient to generate effective competition and thereby remedy the AEC(s).[10]

[10] CC3 (Revised), Guidelines for market investigations, paragraph 332.

3. Cross-cutting considerations

3.1 We have identified a number of cross-cutting considerations relevant to potential remedy design and implementation which may apply to more than one potential remedy set out in this working paper. In this section we discuss:

(a) measures taken in other jurisdictions – in particular, the Digital Markets Act 2022 (the DMA) entering into force in the European Union;[11]

(b) the geographic scope of potential remedies;

(c) links between different remedy options as part of a wider mobile 'ecosystem';

(d) risks relating to the level of specification of certain proposed remedies;

(e) the need for testing and trialling of certain user-choice based remedies; and

(f) the Digital Markets, Competition and Consumers Act 2024 (the DMCC Act).[12]

Digital Markets Act 2022

3.2 To date, the key international legislation (or decisional practice) that we are aware of that applies specifically to the supply of mobile browsers and browser engines is the European Union's DMA, which came into force in 2022.[13]

3.3 The DMA establishes a set of criteria to identify 'gatekeeper' firms. Gatekeepers are large digital platforms providing so-called core platform services which must comply with the obligations and prohibitions listed in the DMA.

3.4 Apple and Google are both designated gatekeeper firms for the purposes of the DMA.[14]

3.5 The DMA includes a number of requirements related to matters within the scope of this market investigation:[15]

(a) Article 5(7) of the DMA sets out that: 'The gatekeeper shall not require end users to use, or business users to use, to offer, or to interoperate with, an identification service, a **web browser engine** or a payment service, or technical services that support the provision of payment services, such as payment systems for in-app purchases, of that gatekeeper in the context of

[11] Regulation (EU) 2022/1925 of the European Parliament and of the Council of 14 September 2022 on contestable and fair markets in the digital sector and amending Directives (EU) 2019/1937 and (EU) 2020/1828 (Digital Markets Act).
[12] Digital Markets, Competition and Consumers Act 2024.
[13] The DMA entered into force on 1 November 2022 and became applicable on 2 May 2023.
[14] European Commission, Digital Markets Act - Gatekeepers.
[15] Emphasis added.

services provided by the business users using that gatekeeper's core platform services.'

(b) Article 6(3) of the DMA sets out that 'The gatekeeper shall allow and **technically enable end users to easily un-install any software applications** on the operating system of the gatekeeper, without prejudice to the possibility for that gatekeeper to restrict such un-installation in relation to software applications that are essential for the functioning of the operating system or of the device and which cannot technically be offered on a standalone basis by third parties.' And Recital 49 notes 'Gatekeepers should also allow end users to **easily change the default settings** on the operating system, virtual assistant and **web browser** when those default settings favour their own software applications and services…'

(c) Article 6(4) of the DMA set out that 'The gatekeeper shall allow and technically **enable the installation** and effective use of **third-party software applications** or software application stores using, or **interoperating with**, its **operating system** and **allow** those software applications or software application stores to be **accessed** by means other than the relevant core platform services of that gatekeeper. The gatekeeper shall, where applicable, **not prevent** the downloaded third-party software applications or software application stores **from prompting end users to decide whether they want to set that downloaded software application or software application store as their default**.' And Recital 50 notes '…the gatekeeper should furthermore allow the third-party software applications or software application stores to **prompt the end user** to decide whether that service should **become the default** and enable that change to be carried out easily.'

(d) Finally, Article 6(6) of the DMA sets out that 'The gatekeeper shall allow **providers of services** and providers of hardware, free of charge, **effective interoperability** with, and **access for the purposes of interoperability** to, the same hardware and software features accessed or controlled via the operating system or virtual assistant… **as are available to services or hardware provided by the gatekeeper**.'

3.6 We have considered below the policies announced by Apple and Google to comply with these provisions of the DMA, where relevant to the design and/or implementation of the potential remedies set out in this paper.

Geographic scope of potential remedies

3.7 We also need to consider the appropriate geographic scope of any remedies to ensure that they are both effective and proportionate.

3.8 As set out in 'WP1 – The nature of competition in the supply of mobile browsers and browser engines', browsers and browser engines are typically made available on a global basis.[16,17] Accordingly, there is a particular question in this market investigation due to the nature of mobile browsers, as to whether a remedy that is limited in geographic scope (eg to the UK) would be effective and proportionate.

3.9 As set out in 'WP6 – Cloud gaming services: nature of competition and requirements for native apps on mobile devices', Cloud Gaming Service Providers (**CGSPs**) are multi-national, and CGSPs generally deliver a consistent service to consumers across the countries where their service is available.[18] Therefore, a remedy enabling UK-only cloud gaming services may be an insufficiently attractive commercial proposition, which could impact its effectiveness.

3.10 We welcome views from stakeholders on the impact the geographic scope of remedies being considered in this paper could have on the effectiveness and proportionality of potential remedies.

The supply of browsers and cloud gaming services within a wider mobile ecosystem

3.11 Third-party services and apps like browsers and cloud gaming services require access to interoperability features and functionality in order to be able to work effectively on Apple's and Google's mobile platforms or ecosystems.[19]

3.12 For example, mobile browsers and browser engines are a type of software that need to interoperate with the operating system on a mobile device. The level of access and integration provided to a browser engine and operating system is an important aspect of many of the potential remedies set out in this paper.

3.13 Apple and Google both control the development of application programming interfaces (APIs) [20] and functionality that allow third-party software to access and use operating system features of a mobile ecosystem. As noted in 'WP3: Access to browser functionalities within the iOS and Android mobile ecosystems', it is not always clear whether a particular functionality resides within the operating system or browser engine layer;[21] and this could mean that functionality that is currently

---

[16] 'WP1 - The nature of competition in the supply of mobile browsers and browser engines' paragraph 3.67.
[17] 'WP1 - The nature of competition in the supply of mobile browsers and browser engines' also notes that sometimes versions of browsers can be released for particular territories, for example Firefox Lite (an Android browser).
[18] 'WP6 - Cloud gaming services: nature of competition and requirements for native apps on mobile devices', paragraph 3.29.
[19] Apple is a vertically integrated company with Apple selling the mobile device and owning and operating, the operating system, the app store and the mobile browser and browser engine used on its devices. Google sells Pixel devices, but also has multiple agreements with Android OEMs through which a certain degree of control may be exerted. Google is the main contributor to the Android operating system and is the steward of the Blink browser engine and owns the Chrome browser.
[20] APIs – Application programming interface is a way for two or more computer programs to communicate with each other. A software interface, offering a service to other pieces of software. Often, a written specification describes how to implement the connection exposed by the API. In this way, one set of software is said to *expose* its API to another.
[21] 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems' paragraphs 3.55 – 3.56.

offered by one component (a browser or browser engine) could be moved across layers of the mobile ecosystem (for example, to the operating system layer). Further, functionality relating to browsers (or browser engines)[22] could be technically modified, creating additional difficulties to effective enforcement and monitoring.

3.14 Mobile browsers are also native apps which can be accessed through native app stores (browser app may also be pre-installed on the device at the point of set up). Cloud gaming services, considered in 'WP6: Cloud gaming services: nature of competition and requirements for native apps on mobile devices' may operate as web apps accessed through a mobile browser but also as a native app accessed through an app store.[23]

3.15 Providers of such apps are therefore affected by any changes or developments in app stores, whether that relates to guidelines, technical requirements and/or changes in the fee structures applied to the app stores.

3.16 The above points mean that the potential issues being considered in this market investigation relate to mobile operating systems, browsers and app stores. The design, implementation and monitoring of the potential remedy options set out below would involve consideration of how these aspects of mobile ecosystems interrelate.

Risks relating to the required level of specification of proposed remedies.

3.17 Many of the potential remedies set out below relate to access for third parties to technical functionalities within the mobile ecosystem.

3.18 Any high-level requirement to provide access to technical functionality may provide more flexibility but could also be difficult to monitor and enforce if there are information asymmetries. On the other hand, if a requirement to provide access to technical functionality is too prescriptive, this could result in the requirement becoming out of date; or being circumvented if the functionality being offered can be adapted.

3.19 Many of the remedy options under consideration in this working paper may therefore require dedicated and extensive monitoring.

Testing and trialling user choice remedies

3.20 As set out in 'WP5: The role of choice architecture on competition in the supply of mobile browsers', users of mobile devices are presented with choice architecture

[22] Mobile browsers and browser engines are pieces of software which serve themselves to be modified through changes to code and APIs.
[23] 'WP6 - Cloud gaming services: nature of competition and requirements for native apps on mobile devices'.

which affects the presentation and placement of browsers and the design of choices that a user may make between different browsers.[24]

3.21 Some of the potential remedies outlined in this paper relate to the choice architecture that is presented to users of mobile browsers (see section 7).

3.22 We consider that choice architecture remedies may benefit from some form of testing and trialling before being implemented to maximise the prospect that they will be effective in achieving the intended aim.

Digital Markets, Competition and Consumers Act 2024 (the DMCC Act)

3.23 The DMCC Act gives the CMA new powers to intervene in digital markets by establishing a new, targeted regime for digital markets.

3.24 The DMCC Act has been introduced in recognition of the fact that there are specific features of fast-moving digital markets that can lead to a small number of firms establishing substantial and entrenched market power. The new regime will strengthen the existing UK competition rules and allow the CMA to take faster, more targeted and effective action where required, and also to monitor, enforce and iterate ongoing requirements.

3.25 The digital markets competition regime will apply to firms designated by the CMA as having Strategic Market Status (SMS) in relation to one or more digital activities. The DMCC Act sets out that a digital activity is the provision of a service by means of the internet, the provision of digital content (which includes software), or any activity which is being carried out for the purposes of providing an internet service or digital content.

3.26 Under the DMCC Act, for a firm to have SMS, it must have:

(a) substantial and entrenched market power in a digital activity which is linked to the United Kingdom;

(b) a position of strategic significance; and

(c) global turnover of more than £25 billion or UK turnover of more than £1 billion.[25]

3.27 Decisions in respect of the new digital markets regime are the responsibility of the CMA Board. Certain decisions under the new regime must be made by the CMA

[24] 'WP5 - The role of choice architecture on competition in the supply of mobile browsers'.
[25] DMCC Act, section 2.

Board (eg whether to begin an SMS investigation) although some decisions may be delegated by the CMA Board.[26]

3.28 The DMCC Act and the new responsibilities that will be conferred on the CMA are relevant to our consideration of possible remedial action in this market investigation. For example, the Inquiry Group could make a recommendation to the CMA Board to consider whether to make an SMS designation in relation to mobile ecosystems and whether to impose certain remedies, should we consider that use of the CMA's new powers would be an effective remedy to any AEC(s) identified.

3.29 We are giving active consideration to whether making a recommendation to the CMA Board to use the powers available under the new digital markets regime would be an effective way of implementing the potential remedies set out in this working paper, particularly in light of the cross-cutting considerations set out in this section above, which suggest that many or all of the potential remedy options would be highly technical and may require ongoing monitoring and possible iteration across different parts of Apple's and/or Google's mobile ecosystems.

3.30 We therefore invite views from stakeholders on the possibility of the Group making a recommendation to the CMA at the conclusion of the market investigation (if the Group were to find the existence of one or more adverse effects on competition) to consider using its powers under the DMCC Act and, where the relevant tests are met, to formulate appropriate remedies, taking account of the findings of this market investigation.

[26] DMCC Act, sections 2 and 19. See further the CMA's draft digital markets competition regime guidance, paragraphs 9.28-9.29.

4. Types of remedies available

4.1 We usually classify remedies as either structural or behavioural. Structural remedies in market investigations are generally one-off measures that seek to increase competition by altering the competitive structure of the market. Behavioural remedies are generally ongoing measures that are designed to regulate or constrain the behaviour of parties in a market and/or empower customers to make effective choices. Remedies can also include recommendations for others to take action.[27]

4.2 At this stage we have not identified structural remedies that we consider likely to effectively address any potential concerns.

4.3 Behavioural remedies can generally be classified as taking the form of either enabling measures or controlling outcomes.

4.4 We have focused our assessment of remedies on enabling measures, which aim to facilitate increased competition by removing obstacles to competition or stimulating actual or potential competition.

4.5 Enabling measures can be split into three categories:

(a) Market-opening measures, which are intended to open up a market to new sources of competition by removing or reducing barriers to entry, expansion or switching;

(b) Informational remedies, which are aimed at giving customers information to help them make choices and thereby increase competitive pressure on firms in the market; and

(c) Remedies that restrict the adverse effects of vertical relationships. Such measures may include obligations to provide access to facilities on fair, reasonable and non-discriminatory (FRND) terms.[28]

4.6 Behavioural remedies seek to change aspects of business conduct from what may be expected based on businesses' incentives and resources. The design of behavioural remedies should seek to avoid four particular forms of risks to enable these measures to be as effective as possible:

(a) Specification risks – these risks arise if the form of conduct required to address the AEC or its detrimental effects cannot be specified with sufficient clarity to provide an effective basis for monitoring and compliance.

[27] CC3 (Revised), Guidelines for market investigations, paragraph 371.
[28] CC3 (Revised), Guidelines for market investigations, paragraph 376.

(b) Circumvention risks – it is possible that other adverse forms of behaviour may arise if particular forms of behaviour are restricted.

(c) Distortion risks – these are risks that behavioural remedies may create market distortions that reduce the effectiveness of these measures and/or increase their effective costs.

(d) Monitoring and enforcement risks – even clearly specified remedies may be subject to significant risks of ineffective monitoring and enforcement.[29]

4.7 For behavioural remedies to have the desired impact it is important that there are effective and adequately resourced arrangements in place for monitoring and enforcement so that there is a powerful threat that non-compliance will be detected and that action will be taken to enforce compliance where it is necessary.[30]

Selection of remedies

4.8 The CMA's selection of remedies is an iterative process in which a potentially wide range of remedy options are progressively narrowed down until a solution has been found that enables the CMA to meet its statutory duties.[31]

Package of remedies

4.9 The CMA's experience to date suggests that remedies in market investigations may take the form of a 'package' of measures, rather than the implementation of a single measure.

4.10 This may be because there could be several features giving rise to an AEC, and consequently an individual measure may be incapable of addressing such an AEC as comprehensively as is reasonable and practicable.[32] For example, to deal with problems associated with a lack of customer switching it might be necessary to both remove contractual barriers to switching and also to put in place informational remedies that raise customer awareness of the potential benefits of switching.

4.11 Alternatively, an individual intervention may contribute to addressing multiple features either alone or in combination with other measures. For example, requirements around technical barriers could seek to both improve customers' ability to switch and reduce the barriers facing a new entrant.

---

[29] CC3 (Revised), Guidelines for market investigations, Annex B, paragraph 40.
[30] CC3 (Revised), Guidelines for market investigations, Annex B, paragraph 41.
[31] CC3 (Revised), Guidelines for market investigations, paragraph 381.
[32] Section 134(6) of the Enterprise Act 2002 , provides that, in making a decision about remedies for an AEC, 'the CMA shall, in particular, have regard to the need to achieve as comprehensive a solution as is reasonable and practicable to the adverse effect on competition concerned and any detrimental effects on customers so far as resulting from the adverse effect on competition.'

4.12 Where more than one measure is introduced, the CMA will consider the way in which the measures are expected to interact with each other.[33] We would consider both the effectiveness of individual measures in the context of an overall package, and the potential package of remedies as a whole.

Summary of potential issues set out in Working Papers 2 - 6 published in this market investigation

4.13 Working papers 2-6 set out the following issues that are being considered in this market investigation:

(a) 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine' - whether Apple is using its position in the supply of mobile operating systems to require that all browsers on iOS and iPadOS use Apple's WebKit browser engine (**Issue 1**). This requirement is referred to in the remainder of this paper as the 'WebKit restriction'.[34]

(b) 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems' - whether Apple and Google are using their positions in the supply of browser engines to restrict rival browsers' access to functionality available in the WebKit and Blink browser engines respectively (**Issue 2**).[35]

(c) 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' - sets out four issues:

(i) Whether Apple is using its position in the supply of mobile operating systems to prevent all rival browser vendors from offering remote tab In-App Browsers (IABs) on iOS (**Issue 3**).

(ii) Whether Apple prevents all rival browsers from offering webviews and bundled engine IABs as part of the wider ban on alternative browser engines (**Issue 4**).

(iii) Whether through default settings and pre-installation of Android WebView on Android devices, Google makes it difficult for app developers to use IABs based on alternative browser engines (**Issue 5**).

(iv) Whether Apple's and Google's policies result in users having limited choice and control in relation to which browser is used for IAB implementations in native apps that they use and IAB in general (**Issue 6**).

---

[33] CC3 (Revised), Guidelines for market investigations, paragraph 393.
[34] 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine'
[35] 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems'

(d) 'WP5 - The role of choice architecture on competition in the supply of mobile browsers' sets out two broad issues:

(i) Apple's and Google's control of choice architecture in factory settings (**Issue 7**).

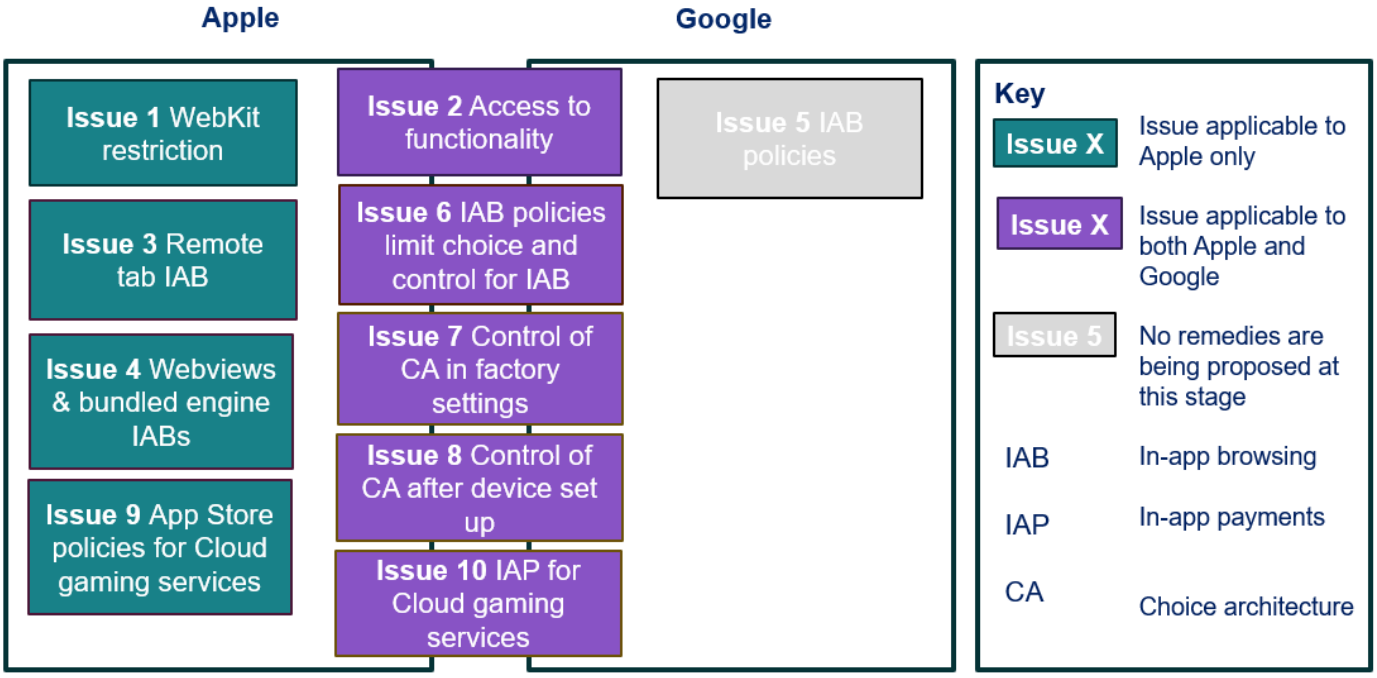
(ii) Apple's and Google's use of certain choice architecture practices after a device is set up (**Issue 8**).

4.14 Working paper 6 sets out emerging thinking on the extent to which access to Cloud Gaming Services on mobile devices is being impeded and the impact that this has on competition in the supply of Cloud Gaming Services. In particular, the working paper explores whether:

(a) Apple's app store policies in relation to cloud gaming apps could preclude or limit the extent to which Cloud Gaming Services are available on mobile devices (**Issue 9**)

(b) Apple's or Google's rules relating to in-app payment systems for in-game transactions, acts as a barrier to the development of native cloud gaming apps (**Issue 10**).

Figure 4.1 Summary of potential issues for which remedies are being considered



Source: CMA

Summary of potential remedies under consideration

4.15 Table 4.1 shows a summary of the potential remedies considered further in this working paper in relation to the issues set out above. In some cases the options in respect of each issue are presented as alternatives; in other cases, a combination of options may be required to deal comprehensively with the relevant issues.

Table 4.1: Summary of browser remedies under consideration

Potential issue	*Remedy Option*	*Remedy description*
Issue 1 – Apple's WebKit restriction	Apple - **Option A1**	Requirement for Apple to **grant access** to alternative browser engines to iOS.
	Apple - **Option A2**	Requirement for Apple to **grant equivalent access to iOS** to browsers using alternative browser engines.
	Apple - **Option A3**	Requirement for Apple to grant **equivalent access** to **APIs used by WebKit and Safari** to browsers using alternative browser engines.
Issue 2 – Apple's and Google's control over supply of browser engines to restrict access to functionalities	Apple - No standalone remedy needed	Could be addressed by Options A2-3 above, ie the granting to rival browsers of equivalent functionality available to WebKit and Safari.
	Google - **Option A4**	Requirement for Google to **grant equivalent access** to **APIs used by Chrome.**
Issue 3 – Apple preventing all rival browser vendors from offering remote tab IABs on iOS	Apple – **Option B1**	A requirement for Apple to **enable remote tab IABs** for **WebKit-based browsers.**
	Apple – **Option B2**	A requirement for Apple to **enable remote tab IABs** for browsers wishing to use **alternative browser engines.**
Issue 4 – Apple preventing rival browser engines from offering non-WebKit based webview IABs, including bundled engine IABs to app developers on iOS	Apple - **Option B3**	A requirement for Apple to **allow alternative webviews** to Apple's iOS WKWebView.
Issue 5 – on Android, default settings and preinstallation of Android WebView make it difficult for app developers to use IABs based on alternative webviews	No remedy is put forward at this stage	
Issue 6 – Apple's and Google's IAB policies offer users limited choice and control in relation to which browser is used for IAB implementation in native apps	Apple/ Google - **Option B4**	A requirement for Apple and Google to **implement remote tab IABs** using the **default browser**.
	Apple/ Google - **Option B5**	A requirement for Apple and Google to **make users aware of being in an IAB** by implementing changes to the interface or implement disclosures.
	Apple/ Google - **Option B6**	A requirement for Apple and Google to **implement opt-out settings** for in-app browsing.
Issue 7 - Apple's and Google's control of choice architecture in factory settings	Apple/Google - **Option C1**	A requirement for Apple and Google to ensure that multiple browsers are **pre-installed**, using defined criteria.
	Apple/Google - **Option C2**	A requirement for Apple and Google to ensure the use of browser **choice screens** at device set-up.
	Apple/Google - **Option C3**	A requirement for Apple and Google to ensure the placement of a default browser selected by the user in the '**dock' / 'hot seat'** or **on the default home screen** at device set-up.
	Apple/Google - **Option C4**	A requirement for Apple and Google to ensure that a user's choice of default browser is always **followed** across **all** browser **access points**.
Issue 8 - Apple's and Google's use of certain choice architecture practices after device set-up	Apple/Google - **Option C5**	A requirement for Apple and Google to ensure the **use of browser choice screen(s)** after device set-up.
	Apple/Google - **Option C6**	A requirement for Apple and Google to **make adaptations to the user journey** for changing their default browser
	Apple/Google - **Option C7**	A requirement for Apple and Google **to share user data on default browsers settings** with browser vendors.
	Apple/Google - **Option C8**	A requirement for Apple and Google to ensure that **the frequency of default browser prompts and notifications is limited.**
	Apple/Google - **Option C9**	**A requirement for Apple and Google to allow users to** uninstall Safari **browser app on iOS and** Chrome **on Android devices.**
Issue 9 – Apple's App Store policies in relation to cloud gaming services	Apple – **Option D1**	A requirement for **Apple** to **review** and **amend its Guidelines** to remove the specific restriction identified as restrictive and a **prohibition** on **Apple** introducing new **restrictions** with equivalent effect.

Issue 10 – app store rules in relation to in-app payment systems for in-game transactions	Apple – **Option D2**	A requirement for **Apple to enable cloud gaming native apps to operate on a 'read-only' basis** (i.e. with no in-game purchases or subscriptions) so that games do not need to be re-coded and no commission would therefore be payable to Apple).
	Apple/Google – **Option D3**	A requirement for **Apple and Google to allow CGSPs to incorporate their own or third party in-app payment systems for in-game transactions**.

Source: CMA

5. Potential remedies addressing Apple's WebKit restriction (Issue 1) and Apple and Google using their position in the supply of browser engines to restrict rival browsers' access to functionality available in the WebKit and Blink browser engines respectively (Issue 2).

5.1 Our emerging analysis set out in 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine' is that Apple's restriction placed on browser vendors to use WebKit on iOS and iPadOS (WebKit restriction) limits rival mobile browser vendors from innovating and improving their browsers on iOS, including on security, privacy, and performance. This prevents browser vendors from achieving potentially higher levels of security or performance than would be possible absent the restriction, or from further differentiating on privacy protections for users, as all browsers are largely restricted to the levels determined by WebKit.[36]

5.2 Further, the emerging analysis set out in 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems' is that Apple's Safari makes use of several features and functionalities on iOS to which third-party browsers do not have the same access. These features are likely to be important to the ability of browser vendors to attract users by offering new or innovative browser features.[37] This is particularly important for smaller browsers which need to provide users with strong reasons to switch away from more established browsers like Safari on iOS.

5.3 Additionally, 'WP3 – Access to browser functionalities within the iOS and Android mobile ecosystems' sets out concerns that the visibility and documentation of the functionalities (or 'APIs') that can be accessed by third-party browsers on iOS by Apple can be poor. This may increase the cost or difficulty of implementing a feature for third-party browsers or result in third-party browsers not being aware that a given functionality is available.[38]

5.4 A key concern highlighted by third parties in 'WP3 – Access to browser functionalities within the iOS and Android mobile ecosystems' in relation to Android was a possible lack of access to WebAPKs, which is essential for installing progressive web apps (PWAs[39]).[40] Google [✂].[41] Any diminished ability to add features relative to Chrome, whether through a complete lack of access, or

---

[36] 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine' paragraph 5.65.
[37] 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems' paragraphs 3.60 and 3.64.
[38] 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems' paragraph 3.65.
[39] PWAs – progressive web apps, which are particular versions of web apps which aim to create an experience more comparable to a native app than a normal web app.
[40] 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems' paragraph 4.6
[41] 'Note of meeting with Google [✂]

time delay or additional costs, may adversely impact the ability of third-party browsers to attract users. This is particularly important for smaller browsers which need to provide users with strong reasons to switch away from more established browsers like Chrome.[42]

Aim of remedies seeking to address Issue 1 and Issue 2

5.5 Remedy options A1-A4 are considered below in the context of mobile devices which include both mobile phones and tablets. In the case of iOS, we include both iOS and iPadOS.

5.6 'WP2 – The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine' sets out various issues which are collectively referred to as the 'WebKit restriction' (Issue 1). This includes Apple's App Store Review Guideline 2.5.6 for browser apps to use WebKit framework and WebKit Javascript.[43] The WebKit restriction means that:

(a) browser vendors are prevented from using alternative browser engines or modified versions of the WebKit browser engine, which would provide a mechanism for browser improvements and differentiation;[44]

(b) browser vendors must develop and maintain an additional version of their browser based on WebKit to serve iOS users which results in increased costs;[45]

(c) because WebKit is the only permitted browser engine on iOS, browser vendors must engage with Apple (which controls the version of WebKit available on iOS) regarding issues with WebKit or requests for new features to be implemented in WebKit. Evidence suggests Apple is slow to engage and often does not respond to such requests, leading to delays in the implementation of new features or fixes.[46]

5.7 'WP3 – Access to browser functionalities within the iOS and Android mobile ecosystems' identified several features and functionalities that Safari had access to that third-party browsers (using WebKit) did not.[47] In addition, Apple noted instances where it made functionalities available to Safari first before extending these to third-party browsers. In some cases, the delay in extending the

[42] 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems' paragraph 4.19.
[43] 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine', paragraphs 2.1 – 2.2.
[44] 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine', paragraph 2.4.
[45] 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine', paragraph 3.28.
[46] 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine', paragraph 3.29.
[47] 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems' paragraph 3.60.

functionalities was significant[48] which is important because even a small-time advantage for Safari can have an impact on competitiveness of third-party browsers.[49]

5.8 'WP3 – Access to browser functionalities within the iOS and Android mobile ecosystems' also noted concerns that the visibility and documentation of APIs that can be accessed by third-party browsers on iOS by Apple is poor. This may increase the cost or difficulty of implementing a feature to third-party browsers or result in third-party browsers not being aware that a given functionality is available.[50]

5.9 In relation to Android, 'WP3 – Access to browser functionalities within the iOS and Android mobile ecosystems' notes that the concern in respect of access to functionalities by third-party browser vendors on Android to be less pronounced than that on iOS, with lack of access to WebAPKs being the main issue highlighted by third parties.[51]

5.10 Examples of delaying availability of features and functionalities to third-party browsers also exist on Android, with the paper noting that third-party browsers' diminished ability to add features relative to Chrome, whether through a complete lack of access, time delay or additional costs, can adversely impact the ability of third-party browsers to attract users, particularly for smaller browsers.[52] The point around when features and functionalities are made available to third-party browsers compared to Chrome and Safari is taken into account for both Apple and Google in the remedy design.

5.11 The remedy options set out below seek to address Issues 1 (WebKit restriction) and 2 (access to browser functionalities).

5.12 The concerns around third-party browser vendors' ability to access features and functionalities available to Safari are, in our initial view, greater than the concerns observed on Android. Our remedy options take this into account under key design considerations.

5.13 Remedy options A1-3 aim to:

(a) enable browsers operating on iOS to use a browser engine other than WebKit, should they wish to do so, and to access the necessary functionality to do so (addressing Issue 1).

---

[48] Intelligent Tracking Protection was extended to third party browsers three years after it was made available to Safari. 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems', paragraph 3.64.
[49] 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems', paragraph 3.64.
[50] 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems', paragraph 3.65.
[51] 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems', paragraph 4.18.
[52] 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems', paragraph 4.19.

(b) provide equivalent access to key features and functionalities that Safari has access to, including the ability to configure and customise these features (addressing Issue 2).

5.14 Option A4 aims to provide equivalent access to key features and functionalities that Chrome has access to, including ability to configure and customise these features (addressing Issue 2).

Remedy Options A1 – A4 overview

5.15 Options A1-3 relate to access to APIs, which currently enable WebKit and Safari to connect with the operating system and device. They present alternative approaches to specifying how access to iOS could be enabled for third-party browsers wishing to use alternative browser engines.

5.16 Under these options, Apple would be required to grant access to any current *and* new functionalities which it makes available to its browsers and browser engines. The access to new features and functionalities would also be required to be made available to third-party browsers irrespective of the browser engine being used at the same time as access is being made to Apple's own browser engines and browsers.

5.17 Third-party browser vendors already have access to several categories of APIs on iOS:

(a) public APIs;

(b) APIs extended under public entitlements; and

(c) APIs extended under managed entitlements.

5.18 Remedy options A1-4 in summary are as follows:

(a) **Option A1** '**Requirement for Apple to grant access to alternative browser engines to iOS**'. This option would stipulate that alternative browser engines should be permitted to operate on iOS. However, this option would not specify the level or type of access to functionality that should be granted by Apple in order to comply with the remedy. It not does address Issue 2 (Apple using its position in the supply of browser engines to restrict rival browsers' access to functionality available in WebKit) because it does not specify the extent of the access to functionality to be provided.

(b) **Option A2 'Requirement for Apple to grant equivalent access to iOS to browsers using alternative browser engines'**. This option would require Apple to give equivalent access to alternative browser engines to iOS to enable browsers to compete with WebKit (and Safari) on an equal footing.

(c) **Option A3 'Requirement for Apple to grant equivalent access to APIs used by WebKit and Safari to browsers using alternative browser engines'**. Under this option, Apple would be required to make public existing (and future) private APIs used by both WebKit and Safari.

(d) **Option A4 'Requirement for Google to grant equivalent access to APIs used by Chrome'**. This option would require Google to provide equivalent access to key features and functionalities that Chrome has access to, including the ability to configure and customise these features.

Option A1 'Requirement for Apple to grant access to alternative browser engines to iOS'

5.19 Option A1 is a high-level, principles-based requirement for Apple to enable access to alternative browser engines on iOS. Apple would need to consider and set out the steps it would take in relation to existing APIs (public or private) to enable access to alternative browser engines.

5.20 Compared to options A2 and A3, this remedy option gives more flexibility to Apple in deciding how to achieve the objective of the remedy, and so may be considered less intrusive and costly to Apple. This option may lead to effectiveness risks, if there is lack of transparency about what APIs and interoperability features exist within Apple that would be relevant for third-party access.

5.21 We note that the DMA contains a high-level requirement for Apple to enable third parties to use alternative browser engines (see paragraph 3a above). To date, there are no live browser apps in the EEA using alternative browser engines on iOS.

5.22 [✂].[53]

5.23 Furthermore, several browser vendors expressed their concerns with Apple's approach to complying with the DMA requirement to enable use of alternative browser engines on iOS which related to two aspects: [54]

(a) application of the WBEE only in the EEA and only to iOS, not iPadOS and

(b) the terms attached to the WBEE make the use of alternative browser engines commercially and technically unattractive, with users in the EEA being required to un-install and then re-install a browser app that chooses to use a browser engine other than WebKit.

[53] [✂] submission to the CMA [✂].
[54] Responses to the CMA's information request [✂].

5.24 We note there may be some benefits in aligning requirements across jurisdictions, such as ease of compliance for Apple and browser vendors and the speed of resolving compliance issues across jurisdictions due to having consistent conditions.

Option A2 'Requirement for Apple to grant equivalent access to iOS to browsers using alternative browser engines'

5.25 Option A2 would create a requirement for Apple to ensure that alternative browser engines are granted access to iOS. However, this option goes further, requiring that third-party browsers using alternative browser engines are granted the access on **equivalent** terms to WebKit and Safari.

5.26 This remedy option would also, for Apple, address Issue 2 (Apple using its position in the supply of browser engines to restrict rival browsers' access to functionality available in WebKit) in that Apple would be required to grant equivalent access to third-party browsers to functionality made available to Safari.

5.27 Under this option, Apple would also have flexibility about how to implement the requirement. In order to provide equivalent access, Apple could choose to create new APIs replicating the functionalities and features made available to WebKit and Safari or it could choose to give access to some of the existing private APIs that exist as internal interfaces within iOS.

5.28 Creating new APIs which replicate the access granted to WebKit (and Safari) may provide a better option than granting access to existing APIs for third-party browser engines. This is because the significant integration that exists at present between WebKit, Safari and iOS could make extending existing APIs to third-party browsers insufficient to achieve equivalent access.

Option A3 'Requirement for Apple to grant equivalent access to APIs used by WebKit and Safari to browsers using alternative browser engines by offering the same APIs'

5.29 Option A3 would require Apple to remove the WebKit restriction by making public all private APIs used by WebKit[55] and Safari[56] (ie by documenting all APIs). This would result in browser vendors wishing to use alternative browser engines on iOS having access to those APIs that were previously only known to and used by

[55] For WebKit this includes iOS APIs made available to WebKit.
[56] For Safari this includes iOS APIs and WebKit APIs.

Apple.[57] It would also result in third-party browser vendors using WebKit having access to features and functionalities that were previously only available to Safari.

5.30 This remedy option would mandate Apple to also make public[58] APIs which Apple has acknowledged it does not extend to third parties (and which are not covered by the categories of APIs set out in paragraph 5.17) by documenting these APIs and making them available to third parties.[59]

5.31 This remedy option would prescribe the way Apple would be expected to provide access to iOS to alternative browser engines and offer equivalent access to third-party browsers choosing to continue to use WebKit which could make it potentially more intrusive compared to remedy options A1 and A2.

5.32 Apple would be required to eliminate its use of private APIs for WebKit and Safari without degrading currently available functionality made available for WebKit and Safari.

5.33 Apple previously noted that it does not have a means of estimating the extent of investment required to ensure that third-party developers have access to the same WebKit and iOS functionality available to Safari, [✂].[60,61]

5.34 As noted above, it is unclear at this stage whether extending access to currently private APIs to third-party browsers would be the best option of enabling access to third party browser engines.

Design considerations applicable across Options A1-3

5.35 Certain design considerations are relevant to the effectiveness of Options A1-3. These are:

(a) the quality of documentation maintained by Apple;

(b) the level of service support available for third-party browsers in relation to making the extended access to functionalities and features effective;

(c) the clarity of any requirements imposed by Apple on third-party browsers wishing to use alternative browser engines on iOS; and

---

[57] Although WebKit is described as open-source software, meaning that its source code can be taken and used by anyone to build software and it can benefit from contributions from a range of stakeholders, Apple employees comprise the majority of code reviewers on the project, the consensus of which decides which changes WebKit incorporates into the WebKit Open Source project . Additionally, as owner of the iOS and macOS operating systems Apple also retains control over the features and functionalities included in the versions of WebKit offered on macOS and iOS.
[58] We consider that 'make public' in this context includes making these APIs available through public and managed entitlements.
[59] 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems' paragraph 3.15 – 3.20, 3.29 and 3.60.
[60] Apple response to CMA's information request [✂].
[61] Apple response to CMA's information request [✂].

(d) the commercial terms attached to ability to use alternative browser engines.

Quality of documentation maintained by Apple

5.36 As set out in section 3 of 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems', clear guidance and documentation is required by browser vendors to make proper use of APIs and add new features into their browsers. As part of the remedy design for options A1-3, Apple would be required to produce clear and complete documentation that is kept up to date.

5.37 Apple has submitted that it does not maintain a [✂] list [✂] relevant to third party browsers nor a [✂] but are made available to third-party browsers on iOS. However, Apple noted that in the vast majority of instances, third parties have access to the same functionality or are able to achieve the same level of functionality without necessarily having the same type of access as Apple apps.[62,63]

5.38 Apple submitted that many of the features that are not yet granted to third-party browsers could in theory be made available to third-party apps, but that it had to account for many considerations before doing so and has to guard against any risks to security, user privacy, and device performance. Apple also submitted that it would have to [✂] necessary to make features available to third-party apps and to consider whether such features would yield better results than if its resources were allocated to features which Apple believes bring meaningful benefit for developers and users.[64]

5.39 Apple subsequently submitted that it would also need to consider that it does not inadvertently compromise user experience or cause practical issues for developers.[65]

Service support extended by Apple to third-party browsers in relation to making the extended access to functionalities and features effective

5.40 Any new APIs created by Apple or existing APIs that are made public under Options A1-3 to provide access to iOS should be kept up to date and maintained to a similar level and standard to APIs used by WebKit and Safari at no additional cost to browser vendors.

5.41 The service offered by Apple would need to be timely and not deter browser vendors from choosing alternative browser engines. The support provided should

[62] Apple's response to the CMA's information request [✂].
[63] Apple also submitted that, given the number of APIs available on Safari, it is not feasible to provide a list of functionalities available to Safari at this time. Apple's response to CMA's information request [✂].
[64] Apple's response to the CMA's information request [✂]. Apple's response to the CMA's information request [✂].
[65] Apple's response to the CMA's information request [✂]. Apple's response to the CMA's information request [✂].

be performed at a similar speed (in relation to resolving security and other significant bugs) as the service received by WebKit and Safari.

5.42 As part of the support offered, we envisage that Apple would need to extend access to a full range of metrics to allow all browser vendors on iOS to measure the performance of their respective browsers.

5.43 Apple submitted that some features (APIs) made available via WebKit are not initially extended to third parties and are developed on Safari before being made available to other WebKit-based mobile browsers. Apple submitted that this allows Apple to efficiently design, test, revise, and ship features, and to ensure that new features will not compromise user privacy and security before making them available to third parties.[66]

Clarity of requirements placed on third-party browsers

5.44 Under Options A1-3, Apple may seek to impose certain security and privacy requirements on browser vendors wishing to use alternative browser engines on iOS.

Security requirements

5.45 We envisage that any security requirements should be balanced and consider the needs of users, operating system providers and browser vendors seeking to compete with Safari.

(a) If security requirements are too specific this may limit browser vendors' ability to differentiate, sometimes not being able to provide further security for users above that which is offered by other browser vendors, including Apple.

(b) If requirements are vague, this may mean that browser vendors are unclear about whether their interpretation of the requirement would be acceptable to Apple. This could create uncertainty over whether the browser app would be accepted by Apple through the App Store review process and may deter browser vendors from considering using alternative browser engines.

5.46 Apple's[67] proposed general security requirements for browser vendors applying for the WBEE in the EEA, includes general security requirements and programme security requirements.

[66] Apple's response to the CMA's information request [✂].
[67] Apple's proposal under the DMA for use of alternative browser engines in the EU.

5.47 We are currently considering the level of specification for Apple's security requirements under this proposal,[68,69] and how this should inform the specification of this remedy option.

Privacy requirements

5.48 In 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine', we set out evidence that the WebKit restriction has resulted in browser vendors being less able to add features and improvements on key parameters of competition, including privacy.[70]

5.49 There may be a case for permitting Apple to set out an expected privacy baseline, to preserve any privacy benefits offered to iOS users currently either through expected privacy outcomes or privacy requirements, and to help to ensure that the quality of users' browser experience is not degraded.

5.50 We envisage that any privacy requirements should not significantly impair browser vendors' ability to differentiate their products or make implementation of those features more burdensome.

5.51 Imposing security requirements on browser vendors may also address privacy risks by serving as an 'exclusionary' mechanism to remove browsers that are unlikely to maintain baseline levels of privacy expected by users.

5.52 Similar to security requirements, privacy requirements should not be overly prescriptive or vague.[71,72]

5.53 Privacy outcomes[73] could be less prescriptive than requirements and could leave more room for browser vendors to innovate. However, attention would be needed

---

[68] For example, programme security requirement 2 requires the use of latest exploit mitigation technologies, including specifically Pointer Authentication Codes (PAC). One party noted that Apple had not made available the components necessary to use PAC [Pointer Authentication Codes], and that alternative mitigations (like memory safe languages) should be considered. [✂] submission to the CMA [✂].
[69] For example, programme security requirement 3 requires to 'follow secure design, and secure coding, best practices'. Apple's proposal under the DMA for use of alternative browser engines in the EU.
[70] An extensive list of examples of privacy features third-party browser vendors are not able to implement on WebKit are described in 'WP2 - The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine' paragraph 3.15.
[71] For example, one browser vendor noted that Apple's privacy requirement 2 (see description below) under the DMA would preclude the browser from being granted the WBEE despite it employing a different but adequate method of tracker blocking. [✂] response to CMA information request [✂].
Programme privacy requirement 2 under Apple's DMA compliance proposal is 'Partition any storage or state observable by websites per top level website or block such storage or state from cross-site usage and observability.'
[72] Any privacy requirements that are too vague and are open to interpretation could impact browser vendors' ability to differentiate. Privacy requirements should be clear and unambiguous giving browser vendors clarity over what the privacy requirements mean and how they can be met.
[73] Privacy outcomes may be expressed as guidance issued to third-party browser vendors by Apple on the types of outcomes that browser vendors should adhere to when considering features and functionalities enabled on their browser. For example, this could be presented such as 'user data is not to be shared without their informed consent'.

to clearly set out the desired outcomes to ensure these are objective and sufficiently specific so they can be monitored and enforced.

Restricting the remedy to browser apps

5.54 We envisage that the access to iOS functionalities required under remedy options A1-3 is relevant to browser apps main purpose of which is to enable users of devices to access the web, view web pages and navigate by hyperlinks and is not expected to be needed by other types of native apps.

5.55 The category of browser apps that may access these functionalities could be prescribed through a defined limited category.

5.56 Changes facing the mobile browser market from new technologies and potential in changing consumer habits as to how online content is consumed (eg search apps or in-app browsing within native apps) would need to be taken into account when defining the category of apps that this potential remedy would apply to.

Commercial/App Store/technical implementation terms

5.57 The design of the relevant remedy would need to take account of any commercial terms imposed by Apple to ensure that these do not introduce frictions or barriers which may undermine the effectiveness of the proposed remedy. For example:

(a) commercial or terms of business precluding browser vendors from choosing the geographic location of where the testing of browser apps using alternative browser engines is to be performed;

(b) geographic restrictions on where alternative browser engines can be used by users;

(c) commercial and business terms which are highly restrictive, which do not similarly apply to Apple's own Safari browser, making launching a competitive browser using an alternative browser engines significantly more difficult;

(d) prohibitive or excessively burdensome App Store Guidelines for browser vendors using alternative browser engines, compared to browser vendors choosing to use Apple's WebKit browser engine or guidelines which mandate use of a specific browser engine;

(e) technical implementation obligations requiring browser vendors to maintain multiple versions of the browser app across Apple's multiple platforms (iOS, macOS and iPadOS) which result in:

(i) maintaining multiple versions of the same browser app for iOS devices;

(ii) users being required to uninstall and re-install a browser app as a result of the change in the browser engine being used; and/or

(iii) users being presented with prompts or disclosure in relation to the change in the browser engine being used by the browser app in an alarming or unbalanced way.

Application of Options A1-3 to address Issue 2: enable access to functionalities and features only made available to Safari to other WebKit-based browsers

5.58 Option A1 would not require a certain level or type of functionality to be provided by Apple to third parties; and as such, may not capture the concerns related to Safari's use of private APIs compared to other WebKit-based browsers (Issue 2).

5.59 By contrast, Option A2, which could involve Apple creating new APIs to replicate the equivalence of access WebKit and Safari use on iOS for third party browsers would also address both issues through inclusion of features and functionalities of both WebKit and Safari.

5.60 Option A3 would also address Issues 1 (WebKit restriction) and 2 (restriction of rival browsers' access to functionality available in WebKit browser engines) because Apple would be required to make public all APIs that are used by both WebKit and Safari. Therefore, a separate remedy addressing this issue would not be required as the APIs used by Safari would be included when implementing the remedy.

Unintended consequences

5.61 The evidence we have seen to date suggests a potential unintended consequence of remedy options A1-A3 could include the impact on security and privacy standards within iOS arising from the removal of WebKit as the sole browser engine on iOS.

5.62 Apple submitted that following its creation of the WBEE to comply with the DMA, EEA users will be worse off in terms of security.[74] Apple submitted that security and privacy requirements (which it introduced via managed entitlements), together with monitoring for their enforcement, will not prevent incremental vulnerabilities from harming users in the EU.[75]

5.63 Apple also submitted that some system-wide security protections will no longer apply across different browser engines.[76] Apple submitted that 'the tight

[74] Update on apps distributed in the European Union.
[75] Update on apps distributed in the European Union.
[76] Note of meeting with Apple [✂].

coordination between different engineering functions plays a critical role in preventing, mitigating, and detecting security vulnerabilities'.[77]

5.64 The design considerations section above identified features of remedy design that we think could go some way towards addressing the circumvention risks inherent to the remedy design but we are still considering the monitoring and enforcement risks which these options may carry, and how they could be mitigated.

Option A4 'Requirement for Google to grant equivalent access to APIs used by Chrome'.

Aim of remedy Option A4 for Issue 2

5.65 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems' sets out the concerns that any diminished ability to add features relative to Chrome, whether through a complete lack of access or time delay or additional costs can adversely impact third-party browsers' ability to compete with Chrome.[78]

5.66 As noted in 'WP3 – Access to browser functionalities within the iOS and Android mobile ecosystems' on Android browser vendors can choose which browser engine to use (although Google's Blink engine is used widely).[79] Therefore, as noted earlier in paragraphs 5.9 and 5.10, the concerns for Android are limited to Issue 2 (Google using its position in the supply of browser engines to restrict rival browsers' access to functionality available in the Blink browser engine) and the extent of the concern appears to be limited (compared to iOS).

5.67 The ability of browser vendors to choose the browser engine of their choice provides a means for browser vendors to overcome any potential shortcomings in the features and functionality that is offered by the browser engine offered by the operating system (in Android's case, Blink). This is because mobile browser engines play an important role in the user experience of mobile browsing, as they can impact factors such as speed, stability, and compatibility with different types of web content and websites.[80]

5.68 However, our emerging view is that it is important that browser vendors seeking to differentiate are able to do so at the browser engine level as well as the browser level because it is not always clear which features are built at which level.[81] This is taken into account when considering remedy design.

---

[77] Apple's response to CMA's information request [✂].
[78] 'WP3 – Access to browser functionalities within the iOS and Android mobile ecosystems' paragraph 4.19.
[79] 'WP3 – Access to browser functionalities within the iOS and Android mobile ecosystems' paragraph 1.2.
[80] 'WP1 – The nature of competition in the supply of mobile browsers and browser engines' paragraph 2.8.
[81] 'WP1 – The nature of competition in the supply of mobile browsers and browser engines' paragraph 2.10.

5.69 The aim of the remedy for Google would be to provide **equivalent** access and functionality to third party browsers including the ability to configure and customise these features. This is similar in nature to the aim of remedy Option A2 for Apple which aims to achieve equivalence of access to WebKit and Safari but does not prescribe the means by which Apple (or Google in the case of Option A4) would achieve this. As Google already makes most APIs public, there is no equivalent to remedy Option A3.

5.70 The remedy would facilitate greater competition between browser vendors, which may be able to offer functionalities and features they previously could not offer on Android or functionalities which were previously only reserved for Chrome or to offer functionalities and features at a similar timeline to when these features become available on Chrome.

Design considerations

5.71 Design considerations set out for Options A1-3 (set out in detail in paragraphs 5.35 to 5.57) also apply to the design of Option A4.

5.72 Based on the evidence to date, it may be sufficient that Google enables access to the WebAPK minting functionality, which is essential for implementing PWAs, for third-party browsers to address the issue set out in 'WP3 - Access to browser functionalities within the iOS and Android mobile ecosystems' paper.

5.73 Google submitted [✂].[82]

Remedy options currently not being considered further

5.74 We identified two possible remedies which we consider would not address Issues 1 and 2 effectively. These are:

(a) Divestment of WebKit by Apple.

(b) Prohibiting Apple from owning a browser engine.

Divestment of WebKit by Apple

5.75 We do not consider that a divestment intervention would be effective in addressing the specific issues considered in this market investigation. For instance, although Apple is WebKit's 'steward',[83] WebKit is open source, meaning that its source code can be taken and used by anyone to build software.

[82] Note of meeting with Google [✂].
[83] This means Apple's employees make a significant portion of WebKit contributions, provide most of the equipment and infrastructure for WebKit's public-facing interfaces and retain control over which course code contributions are accepted and included in the WebKit source code shipped with iOS devices.

5.76 There are practical difficulties with divesting WebKit. Because it is deeply integrated into iOS, Apple would either need to carve out WebKit from iOS, or divest its whole operating system. In our view, a carve-out would give rise to substantial practical implementation risks such that it would be very unlikely to be effective. Requiring Apple to divest iOS would be highly intrusive, have implications far beyond mobile browsers and would not directly enable alternative browser engines on Apple mobile devices.

5.77 Even if WebKit could be effectively divested, Apple could choose to create an alternative browser engine, a version of WebKit (soft or hard fork)[84] or create another version of another browser engine which it could continue to mandate for third-party browser vendors to use. This would undermine the effectiveness of any divestiture.

Prohibiting Apple from owning a browser engine

5.78 Rather than restricting Apple from using a particular browser engine on iOS, an alternative intervention could be to prohibit Apple from developing any browser engines and therefore force Apple to use existing third-party versions of browser engines.

5.79 This type of remedy would not be effective in enabling competition between browser engines and browsers on iOS due to Apple's business model of vertical integration.

5.80 Apple is an OEM, a mobile operating system provider, a native app store operator (of the only app store currently available on iOS UK mobile devices) and a browser vendor. By not allowing Apple to own a browser engine, we would be forcing a browser engine supplier out of the market.

Invitation to comment on Options A1 – A4

5.81 We welcome comments on the emerging views set out above and would particularly welcome views on the following questions:

(a) Are there any alternative remedy options that we have not considered in this paper that could address Issues 1 and 2 as effectively as those set out above?

[84] Browser engines are run on an 'open source' basis. This means developers can use their existing code as a starting point from which to develop their own browser engine (so-called 'forking'). The soft and hard description of the 'forking' signifies the extent to which changes to the original code base are made by the developer. With minimal changes signifying a 'soft fork' and with more significant number of changes being referred to as a 'hard fork' which often would mean a new type of browser engine has been created. Blink is an example of a hard fork because it was created by using WebKit's as the source code.

(b) Do you agree with our emerging assessment that Options A2 and A3, as described, could address both Issue 1 and Issue 2? Please explain why or why not.

(c) As part of remedy design of Options A1-3, are there significant parameters that browser engine providers and browsers would require to be made available to ensure equivalence of access to iOS, in addition to those set out in paragraphs 5.25 to 5.57 above?

(d) Which security and privacy requirements, if any, are reasonable for access to additional iOS functionalities necessary for browsers?

(a) Are there any other commercial or other terms that we have not considered that could undermine the effectiveness of the remedy options set out above?

(e) What are the main monitoring and enforcement risks, and how could they be mitigated?

(f) What are the potential costs or lost relevant customer benefits (RCBs) of remedy Options A1 to A3 that we should consider?

(g) What is the appropriate geographic scope of Options A1-3?

(h) Under Option A4, would enabling the WebAPK minting feature alone be sufficient to level the playing field relative to Chrome for all third-party browsers on Android?

6. Potential remedies addressing Apple's and Google's in-app browsing policies

Summary of emerging in-app browsing issues

6.1 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' sets our emerging analysis in relation to:

(a) Apple preventing all rival browser vendors from offering remote tab IABs on iOS. This means native apps cannot call on any other functionality other than SFSafariViewController[85] for a remote tab implementation of in-app browsing, which is likely to limit rival browsers' ability to compete because they lack the functionality of displaying web content within an app. We understand that browser vendors would be interested in offering alternative remote tab IABs on iOS to improve the quality of their product (**Issue 3**).[86]

(b) Apple preventing rival browser engines from offering non-WebKit based webview IABs, including bundled engine IABs, to app developers on iOS. This policy directly prevents browser engine providers from competing against WKWebView with their own webview on iOS. This may impact app developers' ability to innovate and improve their apps. However, we do acknowledge that allowing bundled engine IABs may introduce security risks such that only app developers with significant resources could offer these securely (**Issue 4**).[87]

(c) Apple's and Google's in-app browsing policies offer users limited choice and control in relation to which browser is used for in-app browsing implementations in native apps. This means that users are likely to provide a limited constraint in relation to competition between browsers for in-app browsing implementations (**Issue 6**).[88]

6.2 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' sets out our emerging view that Google's policy on remote tab IABs does not appear to have a clear impact on competition between browsers on Android because it is not preventing rivals from offering competing products. Further, the impact on competition to date on Google's webview IABs policy is currently unclear. Therefore, at this stage we do not propose any remedies for Google to address these points.

---

[85] An SFSafariViewController object presents a self-contained web interface inside a native app. It allows users to view websites without leaving the native app. SFSafariViewController | Apple Developer Documentation.
[86] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraph 4.14.
[87] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraph 4.31 – 4.32.
[88] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraph 4.54-4.55, 5.34 – 5.35.

6.3 A summary of all remedies under consideration for in-app browsing are presented in Table 6.1 below.

Table 6.1 Summary of IAB remedies under consideration

Potential Issue	*Remedy Option*	*Remedy description*
Issue 3 – Apple preventing all rival browser vendors from offering remote tab IABs on iOS	Apple – **Option B1**	A requirement for Apple to **enable remote tab IABs** for **WebKit-based browsers.**
	Apple – **Option B2**	A requirement for Apple to **enable remote tab IABs** for browsers wishing to use **alternative browser engines.**
Issue 4 – Apple preventing rival browser engines from offering non-WebKit based webview IABs, including bundled engine IABs, to app developers on iOS	Apple - **Option B3**	A requirement for Apple to **allow alternative webviews** to Apple's iOS WKWebView.
Issue 5 – on Android, default settings and preinstallation of Android WebView make it difficult for app developers to use IABs based on alternative webviews	No remedy is put forward at this stage	
Issue 6 – Apple's and Google's IAB policies offer users limited choice and control in relation to which IAB is used in native apps	Apple/ Google - **Option B4**	A requirement for Apple and Google to **implement remote tab IABs** using the **default browser**.
	Apple/ Google - **Option B5**	A requirement for Apple and Google to **make users aware of being in an IAB** by implementing changes to the interface or implement disclosures.
	Apple/ Google **- Option B6**	**A requirement for Apple and Google to** implement opt-out settings for in-app browsing.

Source: CMA

Potential remedies addressing Apple's remote tab IAB policies (Issue 3)

Aim and description of remedy options B1-B2

6.4 App developers can currently only implement a remote tab IAB with SFSafariViewController.

6.5 Requiring Apple to enable third-party browser vendors, irrespective of the browser engine they use, to offer remote tab IABs on iOS for app developers would enhance browser vendors' ability to compete with SFSafariViewController on iOS.

6.6 Enabling third-party browser vendors to offer remote tab in-app browsing would improve the quality of their product and would allow browser vendors to better support their users while preserving essential security, performance and privacy parameters that are important to users.[89]

6.7 Compared to other in-app browsing implementations, remote tab IABs largely rely on dedicated browsers installed on the device thereby improving the security (and privacy) of the in-app browsing experience.

[89] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraphs 2.36.

6.8 As set out in further detail in 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paper, browser vendors noted two main benefits of offering remote tab IABs on Android:[90]

(a) browser vendors want to be able to display web content and extend their differentiating features (eg tracker blockers) to users for in-app browsing; and

(b) offering remote tab IABs increases the time users spend on their browser, which indirectly benefits the browser vendor.

6.9 Two potential remedy options are considered below as a way to address the issue that third-party browser vendors are unable to offer remote tab IABs to app developers on iOS. We consider that these two options (Options B1-2) could be implemented in combination or be mutually exclusive:

(a) a requirement for Apple to enable remote tab IABs for WebKit-based browsers (**Option B1**); and/or

(b) a requirement for Apple to enable remote tab IABs for browsers wishing to use alternative browser engines (**Option B2**); and

Options B1 and B2: A requirement for Apple to enable remote tab IABs for WebKit-based browsers and for browsers wishing to use alternative browser engines

6.10 Options B1 and B2 would require Apple to enable remote tab functionality for all third-party browsers on iOS, irrespective of the browser engine being used. This is likely to involve making the functionality available through APIs which third-party browsers and native app developers could use to implement remote tab IABs.

6.11 Extending the remote tab in-app browsing functionality to third-party browsers could form part of the remedy design considered in Options A1- A3 (discussed in section 5 above).

6.12 Since remote tab IABs are not currently offered to any third-party browsers on iOS, Apple[91] submitted that iOS [✂]. Apple has also submitted that [✂].[92]

6.13 Facilitating cross-app functionality to enable remote tab IABs for third-party browsers may carry some security risks as this functionality can enable security exploits which would need to be mitigated.[93]

---

[90] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraph 2.36.
[91] Apple's response to the CMA's information request issued [✂].
[92] Apple's response to the CMA's information request issued [✂].
[93] Holmberg, A. (2022) iOS vs Android: Security of Inter-App Communication.

6.14 Remote tab IABs are available on Android devices and we consider should be technically feasible on iOS, but we have limited evidence to suggest the types of mitigations that can be put in place to address any security concerns in extending this functionality.

6.15 We also have only limited indication at this stage of the likely costs that Apple, browser vendors and native app developers would be likely to incur to implement remote tab IABs functions on iOS.

Potential remedy addressing Apple's webview and bundled engine IAB policies (Issue 4)

6.16 This remedy option addresses the concern that third-party browser vendors are not able to use alternative browser engines on iOS or offer webview and bundled engine IABs to app developers.

Aim and description of remedy Option B3

6.17 Remedy Option B3 would require Apple to enable browser engine providers[94] to offer webview and bundled engine IABs on iOS.

6.18 The remedy would allow native app developers to choose which webview IAB implementation (including bundled engine) they can rely on to implement in-app browsing in their native app.

Option B3: A requirement for Apple to enable webviews and bundled engine IABs using alternative browser engines

6.19 The remedy would require Apple to allow alternative webviews to Apple's iOS WKWebView that is currently the only webview IAB available on iOS.[95]

6.20 Because webviews are based on a browser engine, Options A1 to A3 (discussed in section 5 above) would remove Apple's requirement to use a specific browser engine on iOS and therefore could indirectly allow alternative browser engines to be the basis for webviews used by native app developers.

6.21 As part of Apple's proposed compliance with the DMA, Apple has enabled native app developers to use bundled engine IABs using alternative browser engines through an entitlement mechanism called the Embedded Browser Engine

[94] Browser engine providers might choose to provide a version of their browser engine for native apps to incorporate within an in-app browser. Please refer to WP4: In-app browsing within the iOS and Android mobile ecosystems paragraphs 2.42 to 2.45 for further discussion on this.
[95] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraph 2.14(a).

Entitlement (EBEE).[96] However, Apple currently does not allow alternative webviews on iOS as part of its compliance with DMA in the EEA.

6.22 Based on the current evidence, we think it is unlikely that alternative webviews would emerge on iOS as a result of this remedy. On Android, where app developers can choose which webview they can implement, there is little competition between alternative webview IABs.[97]

6.23 Interest in providing alternative webview IABs by browser engine providers also appears to be limited.[98]

6.24 Some app developers expressed interest in developing bundled engine IABs as they see benefits to their businesses but are unable to do so on iOS due to Apple's restrictions on alternative webviews. However, the level of interest in this option appears low, and it might be taken up by very few app developers as a result of such a remedy.[99]

6.25 We have heard evidence from app developers that webview IABs offer a seamless and customised user experience which is valued by their users. As explained in 'WP4 – In-app browsing within the iOS and Android mobile ecosystems', we have also seen evidence from several parties indicating that webview IABs could have weaker security and privacy protections relative to remote tab IABs and dedicated browsers.[100]

6.26 Due to greater security and privacy risks inherent to webviews and bundled engine IABs as compared to remote tabs IABs, it may be necessary to introduce mitigations to address these risks, especially if webviews and/or bundled engine IABs become more prominent. For example, relevant bodies engaged in web standards development or in privacy standards regulation and oversight may be well placed to develop guidance or a proposed set of minimum guidelines on privacy, security and performance[101] for webview IABs.

6.27 We will continue to consider the significance of this potential issue as our market investigation continues.

---

[96] As part of Apple's compliance with DMA, Apple introduced the Embedded Browser Engine Entitlement (EBEE) which allows app developers to embed an alternative browser engine within the native app to provide in-app browsing.
[97] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraph 2.14(b).
[98] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraph 4.32.
[99] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraph 4.32.
[100] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraph 2.13.
[101] Webview IABs can lack certain functionalities that are offered by dedicated browsers and which users value, for example, accessibility settings.

Potential remedies addressing IAB and their impact on users' limited choice and control (Issue 6)

6.28 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paper sets out our emerging thinking that users have limited choice and control in relation to which browser is used for in-app browsing in native apps and IABs in general across both Android and iOS.[102]

Aim and description of remedy options B4 - B6

6.29 The potential remedy options set out in this section are designed to address users' limited choice and control in relation to which browser is used for in-app browsing in native apps and IABs in general.

6.30 We are considering three options which could address this potential issue:

(a) a requirement for Apple and Google to implement remote tab IABs using the default browser (**Option B4**);

(b) a requirement for Apple and Google to make users aware of being in an IAB by implementing changes to the interface or implement disclosures (**Option B5**); and

(c) a requirement for Apple and Google to implement opt-out settings for IAB (at device settings level) (**Option B6**).

Option B4: A requirement for Apple and Google to implement remote tab IABs using the default browser

6.31 Option B4 remedy would require that where the app developer utilises remote tab IAB, then the app would call on whichever browser a user has set as their default dedicated browser – although would still allow app developers to change which browser they use for in-app browsing if needed.

6.32 Currently, SFSafariViewController is the only remote tab IAB functionality available on iOS.[103] However, on Android, remote tab IAB (Custom Tabs IAB) already relies on the user's dedicated browser by default, but app developers are free to change this and choose a specific browser.[104]

[102] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraphs 4.54 and 5.34.
[103] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraph 4.34(a).
[104] 'WP4 - In-app browsing within the iOS and Android mobile ecosystems' paragraph 5.20(b).

6.33 An alternative would be to stipulate that operating systems providers always rely on the users default browser. This would override any setting at the operating system level or app level which integrates a different browser for in-app browsing.

6.34 This remedy could be complementary to and contingent on, Options B1 and B2 discussed above (see paragraphs 6.10 to 6.15) enabling remote tab IABs for third-party browser vendors.

Option B5: A requirement for Apple and Google to make users aware of being in an IAB by implementing changes to the interface or use disclosures

6.35 'WP4 – In-app browsing within the iOS and Android mobile ecosystems' paper describes how the in-app browsing interface looks across different in-app browsing implementations on iOS and Android devices.

6.36 Broadly, in-app browsing interfaces mimic browser apps' visual interface meaning that users may not realise they are browsing within an in-app browser. This may contribute to very low levels of user awareness and control of in-app browsing.[105]

6.37 Option B5 would aim to increase user awareness of the in-app browsing experience to facilitate more informed user choice when browsing in native apps.

6.38 A potential remedy option would be to require an 'information screen/ disclosure' to be presented to a user when opening a link to third-party content in an IAB that would provide an alert about the presence of an in-app browsing feature (for example, this may read '*This browsing experience is provided within the app and is not your default browser*').

6.39 An alternative remedy would be to require operating system providers to visually change the design of the in-app browsing interface to differentiate in-app browsing from the dedicated browsing experience.

Option B6: A requirement for Apple and Google to implement user settings allowing users to opt-out of in-app browsing

6.40 iOS and Android usually do not have the option to disable in-app browsing completely in their device settings. Option B7 would aim to provide users with an option to opt-out of in-app browsing on their mobile devices at the device settings level.

[105] The CMA has commissioned qualitative and quantitative research by Verian on users' awareness, understanding and behaviour in relation to mobile browsers and in-app browsers. Qualitative interviews with users showed that overall users have very low levels of awareness of in-app browsing, with users not interested in what is happening operationally 'behind the scenes' when they are in an app. Quantitative survey revealed that almost half of respondents correctly understood that different apps on their smartphone may use different web browsers, 13% of respondents incorrectly believed that different apps did not use different web browsers, while the remaining 40% responded that they did not know.

6.41 A requirement to allow users to opt out of using IABs would rely on operating system providers providing settings to users to enable and disable in-app browsing in their device settings and to periodically confirm those preferences.

Unintended consequences in relation to potential remedy options B4-6

6.42 Ensuring that remote tab IABs invoke the user's default browser has implications for app developers' control over in-app browsing in their apps. For example, some native apps rely on advertising revenue to monetise their apps and therefore app developers may choose specific browsers that enable collection or tracking of certain user metrics. In cases where the users' default browser is more privacy-focused and may not allow for the collection of certain data, native apps which rely on this advertising revenue may need to develop other revenue models (eg charging for the app).

6.43 Further, if users' default browser was always to be used for remote tab IABs, this may create uncertainty for app developers in relation to which features can be implemented through the IAB in their app (eg specific website capability would not be supported or could break). This uncertainty might lead to a greater uptake of webview IABs by app developers so they can customise the IAB experience to their needs (see paragraph 6.25).

6.44 Research carried out in this market investigation indicates that user awareness and understanding of browsers is low and awareness and understanding of IAB is even lower. This may be a result of a lack of competition or of existing choice architecture for browsers, or it may be because users find it technically complex to understand how dedicated browsers and in-app browsing work on mobile devices. This means that increasing the visibility of the IAB experience (Options B5 and B6) might not lead to expected or immediate benefits for users, and it might potentially worsen the user experience overall.

Invitation to comment on Options B1 to B6

6.45 We welcome comments on the emerging views set out above and would particularly welcome views on the following questions:

(a) What technical considerations would need to be considered when extending remote tab in-app browsing to third-party browsers on iOS?

(b) What are the likely costs that would be incurred by Apple, app developers and third-party browser vendors to enable remote tab IABs on iOS?

(c) What are the benefits and drawbacks in extending users' default browser choice to remote tab IABs (ie always implementing remote tab IAB using users' dedicated browser)?

(d) What are possible remedy options, if any, to address Google's webview IAB policy (Issue 5)?

(e) In relation to Option B6, should user-based awareness and consent for in-app browsing be increased and if so:

(i) Which design considerations should be taken into account?

(ii) Should the user be prompted to consent to in-app browsing at a:

(1) System-level (phone settings)

(2) App-level (each app's settings)

(3) At both the system and app levels?

(iii) Should the default setting be set as opt-in or opt-out in each of the cases above, and why?

7. Potential remedies addressing Apple's and Google's control of choice architecture in factory settings (Issue 7) and Apple's and Google's use of certain choice architecture practices after the device set up (Issue 8)

7.1 'WP5 - The role of choice architecture on competition in the supply of 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.

7.2 The six choice architecture practices referred to in 'WP5 – The role of choice architecture on competition in the supply of mobile browsers' are:

(a) pre-installations of browsers in the factory settings and installations of alternative browsers;[106]

(b) placement of browsers on a mobile device home screen in the factory settings;[107]

(c) default browsers settings in factory settings ('system default browser')[108] and at or after device set up ('chosen default browser');[109]

(d) friction in the user journey to change default browsers;[110]

(e) prompts and push notifications to switch or change default browser settings;[111] and

(f) the ability of users to uninstall a pre-installed browser.[112]

7.3 These practices relate to two key stages in the use of mobile devices:

(a) Practices (a) to (c) relate to the factory settings on a device for first use (**Issue 7**), eg the initial pre-installation placement and pre-set default setting for browsers included in the factory settings for devices on first use.

---

[106] 'Pre-installation' refers to browsers that have been installed on a mobile device at point of purchase, such that they are available for users 'out-of-the-box'.
[107] 'Placement of browsers' refers to the positioning of a browser on the mobile device, typically on the 'default home screen' of the device, and in many cases in the 'hotseat' on the home screen (centrally in the row of apps placed at the bottom of the home screen).
[108] System default browser: a default chosen by the OS provider or device manufacturer.
[109] Chosen default browser: a default chosen by the user.
[110] Friction in the user journey for changing the default browser refers to the number and/or complexity of steps involved for changing the default browser app unprompted.
[111] Prompts and push notifications refer to pop-ups or screens encountered by users (for example, on launching a browser app) which encourage the users to either download a new browser app or set a particular browser as the default.
[112] 'Uninstallation' refers to removing or deleting an app from a device.

(b) Practices (c) to (f) relate to the use of choice architecture practices after initial set up (**Issue 8**), eg chosen default browser, friction in the user journey to change defaults, prompts and uninstallations.

7.4 These practices may mean that users may make fewer effective choices about which browser to use on their mobile device, or experience difficulty or friction in exercising choice or switching between the use of different browsers. Overall, this may mean that fewer consumers are likely to switch between browsers, and therefore, drive browser competition.

Remedy design principles for all choice architecture remedies

7.5 As discussed more extensively in 'WP5 - The role of choice architecture on competition in the supply of mobile browsers', the way choice architecture remedies are designed and implemented can significantly impact the effectiveness and uptake of a particular remedy.

7.6 We have considered the following three broad principles:

(a) **Targeted** - presenting choices to users at the right place, at the right time, with the right frequency. Firms should identify suitable points for the choice to be presented and ensure that users' preferences are consistently applied. Firms should also prompt when users are most likely to engage with the choice. Users can learn from previous experience (including mistakes) and their preferences can also change over time. It is important that firms give users the opportunity to make choices more than once, which can help them learn but also let them change without encountering difficulty whenever they choose. However, asking users too often can overwhelm them and lead to poor decisions.

(b) **Understandable** – giving choices that users understand. Firms should think about the layout or presentation of choices to ensure that users adequately understand the choice and can make decisions in their best interests.

(c) **Balanced** – giving users autonomy and minimising unjustified friction where possible. Users should have the ability and facility to make important choices with their preferences implemented and respected. Also, firms should focus on finding the right amount of friction for users – minimising unjustified friction and understanding where friction can be positive (eg confirming an important action) or protecting users from potential self-harm.

7.7 These principles draw on the CMA's existing work and thinking in the realm of online user choices, serving as a summary of current good practices for choice architecture remedy design.[113]

7.8 In seeking to design choice architecture remedies that are effective and proportionate, it may be desirable, or even necessary, that design choices are tested with users before they are implemented.

Summary of proposed remedies

7.9 As noted in 'WP5 - The role of choice architecture on competition in the supply of mobile browsers', a range of choice architecture is presented to users in relation to the use of a mobile browser on mobile devices.

7.10 Table 7.1 below lists all the remedy options that we are considering in relation to the six choice architecture practices set out above. Some of these remedy options below are described as alternatives where relevant.

Table 7.1: Choice architecture remedies under consideration

Potential Issue	*Remedy Option*	*Remedy description*
Issue 7 - Apple's and Google's control of choice architecture in factory settings	Apple/Google - **Option C1**	A requirement for Apple and Google to ensure that multiple browsers are **pre-installed**, using defined criteria.
	Apple/Google - **Option C2**	A requirement for Apple and Google to ensure the use of browser **choice screens** at device set-up.
	Apple/Google - **Option C3**	A requirement for Apple and Google to ensure the placement of a default browser selected by the user in the '**dock' / 'hot seat'** or **on the default home screen** at device set-up.
	Apple/Google - **Option C4**	A requirement for Apple and Google to ensure that a user's choice of default browser is always **followed** across **all** browser **access points**.
Issue 8 - Apple's and Google's use of certain choice architecture practices after device set-up	Apple/Google - **Option C5**	A requirement for Apple and Google to ensure the **use of browser choice screen(s)** after device set-up.
	Apple/Google - **Option C6**	A requirement for Apple and Google to **make adaptations to the user journey** for changing their default browser
	Apple/Google - **Option C7**	A requirement for Apple and Google **to share user data on default browsers settings** with browser vendors.
	Apple/Google - **Option C8**	A requirement for Apple and Google to ensure that **the frequency of default browser prompts and notifications is limited.**
	Apple/Google **- Option C9**	**A requirement for Apple and Google to allow users to uninstall Safari browser app on iOS and Chrome on Android devices.**

Source: CMA

Aim and description of potential remedy options C1-4 (relating to device set up)

7.11 The potential remedy options set out in this section are designed to level the playing field for third-party browsers in relation to initial device settings (ie factory settings), increasing traffic to these browsers. C1-4 below all aim to address user

[113] For example this includes the CMA's (2022) publications on Online Choice Architecture (OCA), CMA's (2023) joint position paper with the Information Commissioner's Office (ICO) on the impact of harmful digital design on user choices and control over personal information and the CMA's (2020) Online platforms and digital advertising market study final report.

inertia in relation to a first-party browser where such browsers come pre-installed and prominently placed on a device:

(a) Option C1 relates to pre-installations of Safari on iOS and Chrome on Android and would require adaptations to the design of choice architecture in the factory settings. This remedy would aim to limit Apple and Google's control over which browsers are pre-installed on devices.

(b) Option C2 relates to the default setting for browsers. This would aim to ensure that the browser used on a mobile device is actively chosen by the user at device set-up.

(c) Option C3 would require adaptations to current choice architecture which places Safari on iOS and Chrome on Android on the 'dock'/'hot seat'[114] and/or default home screen[115] at the point of device set up. Option C3 would aim to ensure the browser which has the most salient placement is the same browser which has been selected as a default by the user.

(d) Option C4 relates to the default setting for browsers. This would aim to ensure that the user default is respected as such across all other access points.

Design considerations for remedy options C1-4

Option C1: A requirement for Apple and Google to ensure that multiple browsers are pre-installed, using defined criteria

7.12 Option C1 would require Apple on iOS and Google on Android to pre-install a certain number of browsers in factory settings based on pre-determined criteria, though we note that the criteria would need to be carefully considered.

7.13 Both Google and Apple submitted that having a browser app pre-installed and pre-set as a default creates a positive experience for users, who expect to be able to immediately access the internet and use the device with minimal set-up when they first power on their device.[116,117]

7.14 Increasing the number of browsers that users have pre-installed on their device may increase both user awareness of alternative browsers and influence users' choice of which browser to use. Mandating Google and Apple to increase the

[114] The 'hotseat' or 'dock' position refers to the positioning centrally in the row of apps placed at the bottom of the home screen. Apps located in the 'hotseat' remain visible even when the user moves away from their default home screen to another screen on their device. This is explained at p15 of 'WP5 - The role of choice architecture on competition in the supply of mobile browsers'.
[115] The 'default home screen' refers to the initial screen that the user sees when unlocking their device.
[116] Apple's response to CMA's information request [✂].
[117] Google's response to CMA's information request [✂].

number of pre-installed browsers on their respective operating systems would increase the number of users reached with this remedy.

7.15 Remedy Option C1 would need to consider potential interactions in relation to which browser is set as the default and any other choice architecture remedies that increase user choice in this regard. These considerations include the criteria for selecting the list of pre-installed browsers, and where the pre-installed browsers will be placed on the home screen. In addition, this remedy might have an impact on users' device storage and users' ability to uninstall a pre-installed browser (see remedy Option C9 for more details).

Option C2: A requirement for Apple and Google to ensure the use of browser choice screens at device set-up

7.16 A requirement for both Apple for iOS and Google for Android to implement choice screens at the point of device set up would require users to make an active choice and increase the visibility of third-party browser alternatives. This remedy would focus on new smartphone users.

7.17 By having a choice screen at the point of device set up, the user's choice of a default browser could also be automatically installed as part of the device set up process (addressing issues with pre-installations covered in C1) and automatically placed in the 'dock/ hotseat' position on the device home screen (addressing the issues with placement covered in C3). This means that once the user has completed their device set up, their chosen default browser would be ready for use on a device home screen without the need to download the browser app in the app store or move the app manually to the default home screen.

7.18 The selection of a choice screen has been considered on a number of occasions internationally, particularly by the European Commission. This applies to both mobile browsers and to search engines. In particular we note:

(a) In April 2019, Google announced that it would start presenting a new dual choice screen to Android users in Europe with an option to install additional browsers and search engines from a list of five options.[118] The choice screen is presented to users the first time they open the Play Store following an update.

(b) In August 2019, Google announced that it would implement a choice screen for general search providers on all new Android phones and tablets shipped into the EEA and the UK where the Google Search app is pre-installed, with an option to set the default.[119]

[118] Presenting search app and browser options to Android users in Europe (blog.google).
[119] Android Choice Screen.

(c) More recently, the DMA has mandated both Apple on iOS[120] and Google on Android[121] to implement a choice screen for browser default selection.

7.19 These developments suggest that it is possible for the choice architecture remedy options to be specified within a particular geographical location and that potential remedies in the UK relating to the use of a choice screen could differ from the choice screen that is currently being used in the EEA by Apple and Google.

7.20 Initial evidence from third-party browser vendors in the EEA is that the choice screens for browsers are having a positive impact, though their longer-term impact is yet unknown.[122,123,124] For example, some browser vendors (eg Brave, Opera, Ecosia have publicly stated that users have been increasingly downloading their browser since the choice screen remedy was implemented.

7.21 We note that this remedy, if pursued, would need careful design and testing with users in order to consider a range of issues, including but not limited to:

(a) when the choice screen should be shown (eg at which point in the user journey);

(b) where the choice screen should be introduced (eg access point);

(c) how frequently the choice screen should be displayed (eg one-off vs repeatedly);

(d) how information should be framed; and

(e) how the options should be displayed.

7.22 The remedy would need to determine a framework of criteria for browsers to be included in the choice screen. Two possibilities we are considering are a randomised list or a set number at the point of device set-up, by market share or other selection criteria.

7.23 For users that have existing smartphones, a 'choice screen' could be displayed upon the next major iOS/ Android OS update (remedy Option C5 to mandate choice screen after the device set-up). However, we recognise that Google's ability to control operating system updates with non-Pixel OEMs might be limited which might impact this option's effectiveness.

---

[120] About the browser choice screen in iOS 17 - Support - Apple Developer.
[121] DMA Android Choice Screen.
[122] Opera saw 164% growth in the inflow of new EU users on iOS after the DMA-enforced ballot screen - Opera Newsroom.
[123] Brave Software on X: "Why did Apple and Google make it hard to switch default browsers for so many years? Because it's a powerful way to block competitors. Just look at what happened to Brave installs on iPhone in the EU when Apple rolled out a new default browser choice screen on March 6th: https://t.co/Wefz4mCHGi" / X (twitter.com).
[124] Ecosia published a blog: "It should be easy to switch search engines: 10 principles for fair choice screens" on May 7th

Option C3: A requirement for Apple and Google to ensure the placement of a selected default browser by the user in the 'dock'/'hot seat' or on the default home screen at device set-up

7.24 Option C3 would mandate that Apple on iOS and Google on Android always place the browser selected by the user in the 'dock' or 'hot seat' at device set-up. Given the importance of the placement of a browser, particularly the 'dock'/'hot seat' position, this would entail requirement for Apple and Google to ensure that the default browser that is selected by users at device set-up is automatically given the 'dock'/'hot seat' position. This remedy option would complement Option C2 – a choice screen - that would allow users to select their default browser at device set-up. However, this requirement will not be maintained after the device set up (eg if the user subsequently changes their default browser, or if the user decides to move their default browser icon out of the 'dock' / 'hot seat'). By placing the default browser in the 'dock'/'hot seat', users are likely to find it quicker and easier to use the default browser that they have selected.[125]

7.25 We understand from Apple that their objective in placing apps on the default home screen and the dock is to provide the best user experience. Apple has submitted that, when considering the placement of apps out-of-the-box on iPad and iPhone, it takes into account a number of factors, including [✂].[126]

7.26 While the placement of Chrome can vary depending on the device manufacturer, we consider the fact that Google has sought specific placement agreements emphasises the importance of placement and pre-installation of Chrome, as do revenue sharing agreements that allow Chrome to be set as a default in factory settings.[127]

7.27 However, we understand from some third-party browser vendors that, historically, some agreements have been pursued or agreed with non-Pixel OEMs for pre-installation, placement and default settings on devices on Android, but this was not a commercially viable approach for them.[128]

7.28 On iOS devices at factory settings, due to Apple's restrictions, third-party browser vendors have never had an option to be pre-installed, placed prominently or pre-set as a default.

[125] We note that users are able to choose to move and change which apps show in the hotseat or dock position after device set-up.
[126] Apple's response to CMA's information request [✂].
[127] For more details see 'WP5 - The role of choice architecture on competition in the supply of mobile browsers', paragraphs 4.25 – 4.28
[128] Notes from meetings with [✂]. [✂] response to CMA's request for information [✂].

7.29 In addition, we have also heard from some browser vendors that the placement of a browser in the 'dock'/'hot seat' is valuable for steering users to access and adopt their browser.[129]

7.30 This is supported by findings in the Verian survey that visibility of a browser app impacts usage and engagement. For the majority of survey respondents (63%) their most used browser was on their home page, with only 8% reporting that it was on a page other than their home page.[130]

7.31 Our preliminary view is that Option C3 might be particularly effective as a means of facilitating competition between browsers, given the importance of the placement of a browser, particularly the 'dock'/'hotseat' position, alongside Option C2. However, we recognise that this would be less impactful after device set-up, especially if users subsequently change their default or decide not to have a browser app in the 'dock'/'hot seat' at all.

Option C4: A requirement for Apple and Google to ensure that a user's choice of default browser is always followed across all browser access points

7.32 A further remedy (in addition to Options C3 and C5) would require that a user's choice of default browser should be carried across all relevant access points where users may access web content on their device rather than rely on the pre-installed browser.[131] Our preliminary view is that Option C4 would ensure that effectiveness of users choice of default browser (via 'choice screen' in relations to Options C2 and C5) is not restricted only to a direct use of browser. This remedy could also incorporate Option B4 (see paragraph 6.31 above) that mandates that remote tab IABs use default browser.

Aim and description of remedy options C5-9 (relating to choice architecture practices used after device set up)

7.33 The remedy options set out in this section are designed to enhance consumer decision-making by providing meaningful choices, presented in a way that facilitates effective and informed decisions by users about which browser to use on their mobile device, both in relation to which browsers are available and switching between different browsers. This in turn could contribute to greater competition between browsers.

(a) Option C5 relates to the default browser on existing devices, as opposed to Option C2 which relates to the choice of default browser at set up on new

[129] Notes from meetings with [✂]. [✂] response to CMA's request for information [✂].
[130] Verian Group UK (2024) Mobile Browsers Quantitative Research Data Tables, BROWLOC1.
[131] For example, if a user gave Siri a command to search the web, the results would open up in the user's choice of browser.

devices. It would require the presentation of a browser choice screen to users that would not have been presented with a choice screen at set-up of their device. This would seek to ensure that users of existing devices have the opportunity to actively set a default browser.

(b) Option C6 seeks to address frictions in the user journey for changing default browser settings by mandating that Apple and Google reduce the number of steps to switch default browser on their devices. This option would seek to ensure that the journey for users who, unprompted, want to change their default browser, is not unnecessarily burdensome.

(c) Option C7 seeks to facilitate access to user data in relation to downloads and default browser settings, in order to increase the effectiveness of third-party browser vendors.

(d) Option C8 seeks to ensure that third-party browser vendors use the same volume and frequency of prompts as Safari or Chrome currently do, suggesting that users change their default browser.

(e) Option C9 seeks to ensure that users could uninstall any pre-installed browsers on their device, in order to provide users with greater autonomy over their usage of apps on their devices.

Remedy description

Option C5: A requirement for Apple and Google to ensure the use of browser choice screen(s) after device set-up

7.34 This remedy can be viewed as similar to Option C2 in respect to the design and framing, with the key difference being that this remedy would be shown to existing smartphone owners, as opposed to Option C2, which concerns a choice screen at set-up for new devices. In this case, a choice screen of browser options would prompt users to select their default browser.

7.35 As with Option C2, the timing, frequency and the design of a choice screen (eg one-off choice screen prompt for existing users only to avoid duplication with Option C2) would need to be considered carefully; together with whether this could be implemented as a part of an operating system update.

Option C6: A requirement for Apple and Google to make adaptations to the user journey for changing the default browser

7.36 This remedy would specify the maximum number of steps that must be taken, starting from the device home page, through to the successful completion of the change default browser action, for a browser that has been installed on the device.

7.37 It would also tackle another feature of the difficulty that users experience when navigating default browser settings - the lack of visibility of the switch of default browser setting. This can occur either because the font for the relevant option is difficult to read or is positioned where a user might not expect to find it. We envisage that both of the adaptations mentioned above would be applied regardless of which browser is currently set as default.

7.38 This remedy would also seek to ensure good visibility for the relevant option at each step of the user journey by specifying some minimum standards for visibility such as font size, colour and positioning in the device settings. Such visibility standards may be framed in terms of a requirement that the relevant options be no less visible than other options in the settings of devices.

7.39 This remedy may consider either a single centralised location in the settings for changing default browser, regardless of what browser is currently set as default and/or that the user journey for changing default browser should be identical regardless of which browser is set as default.

7.40 Apple's view is that the current user journey on iOS for changing the default browser, is 'incredibly easy and intuitive. Users need only open Settings and find the relevant browser app, tap the app, then tap 'Default Browser App' and select the desired default'.[132]

7.41 Google submitted that the 'effectiveness of a user journey to switch defaults cannot be judged effectively solely by the 'number of steps' involved. A choice's effectiveness must be determined by a combination of relevant considerations, such as:'

(a) Examining the journey in the context of other choice options that are already provided.

(b) Examining the user journey in the context of the user's task.

(c) How simple or complex the user journey to switch defaults is at present.[133]

7.42 Google also submitted that it is OEMs rather than Google, that configure the user journey for changing the default browser on Android devices. Therefore, the effectiveness of this remedy might be reduced.[134]

7.43 Option C6 has also been part of the DMA compliance since March 2024. Therefore, we may be able to observe trends in browser behaviour that follow from

[132] Apple's response to CMA's information request [✂].
[133] Google's response to CMA's information request [✂].
[134] Google's response to CMA's information request [✂].

these changes in the future. Furthermore, this should make it easier for the relevant parties to extend these remedies to the UK.

Option C7: A requirement for Apple and Google to share user data on default browsers settings with browser vendors

7.44 We understand that browser vendors have limited visibility over whether a user has set their browser as a default after it is downloaded on iOS and Android.[135]

7.45 Option C7 would allow browser vendors to query whether or not their downloaded browser is set as the default browser on the device. This increased transparency over default browser settings would allow browser vendors to more efficiently and effectively engage with their users (eg eliminating the risk of redundant prompts to users that have already set their default browser). This option would be subject to existing data protection rights and user control.

7.46 Option C7, although closely aligned with Options A1 to A3, does differ from them in so far as it would require the creation of a new API for tracking user default status should one not exist.

Option C8: A requirement for Apple and Google to ensure that the frequency of default browser prompts and notifications is limited

7.47 Option C8 would be to regulate the volume, frequency and design of the prompts that browser vendors use to compete for default browser status. For example, it would limit Google's current practice of using prompts to users who have a non-Chrome browser set as their default browser to switch their default back to Chrome.

7.48 Limiting the frequency of prompts would seek to level the playing field for other providers, ensuring that neither Apple nor Google can leverage their control of their respective operating systems to self-preference in relation to prompts and notifications regarding default browser status. Users might benefit from the limit on the use of such prompts as it would remove the potentially 'nagging' nature of prompts that users are exposed to and ameliorate the fatigue that an overload of prompts and notifications can produce. However, using prompts is an important tool for third-party vendors as it is one of the main mechanisms through which they can obtain a foothold in the market.

7.49 For example, DuckDuckGo submitted that it displays prompts to users on iOS and Android upon download and first opening of the DuckDuckGo browser. It

[135] Responses to CMA requests for information [✂].

submitted that these prompts help users to set DuckDuckGo as the default browser.[136]

7.50 A competing browser vendor submitted that it would like to see the number of clicks required in the iOS user journey for changing the default browser to be reduced, ideally with a direct link via the prompt window to the change browser setting.[137]

7.51 Furthermore, the quantitative survey conducted by Verian indicates that, of respondents that have encountered such prompts, 30% report finding them 'usually helpful' or 'occasionally helpful' (41%), as opposed to 'rarely helpful' (20%) or 'never helpful' (8%).

7.52 This remedy option would require careful design, and consideration of a range of issues such as remedy Options C2 and C5 on choice screens.

Option C9: A requirement for Apple and Google to allow users to uninstall Safari browser app on iOS and Chrome on Android devices

7.53 Option C9 would require Apple and Google to allow users to uninstall pre-installed Safari on an iOS device, or to uninstall pre-installed Chrome on an Android device. If pre-installed browsers cannot be uninstalled, this might mean that a user who has deleted the app from their home screen in their respective operating system may still be using that browser if chosen by app developers such as when clicking a web link in a third-party app.

7.54 An important consideration for this remedy option would be whether pre-installed browser apps (eg Safari and Chrome) are critical to the functioning of the operating system, so that for technical reasons they could not be uninstalled.

7.55 We note that there is a provision within the DMA which prohibits gatekeepers from preventing users from un-installing pre-installed software or apps if they wish to do so.[138]

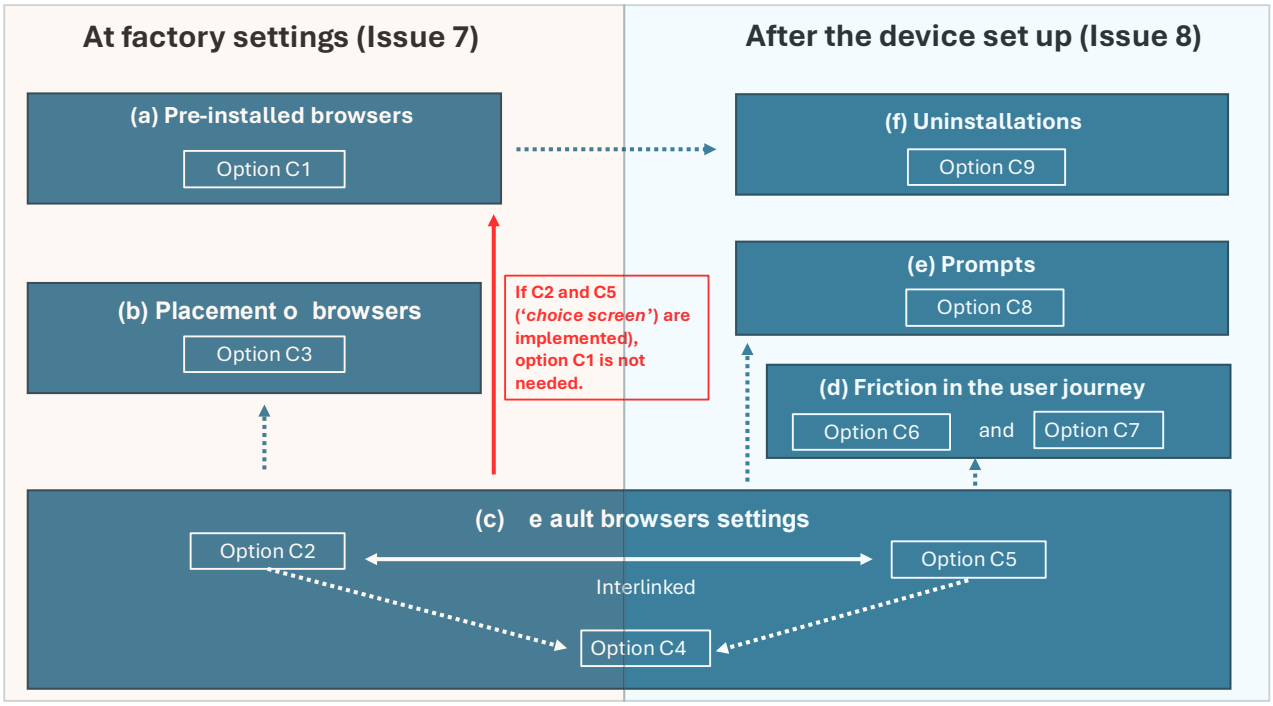
Design considerations applicable across options C1-9

7.56 In Figure 1, we show an overview and some interlinkages between potential choice architecture remedy options at factory settings (Options C1 to C4) and after device set-up (Options C5 to C9). All choice architecture remedies would require careful design, including as mentioned above, when, where and how frequently the choice architecture practices will be shown, as well as how the information is framed and displayed. Therefore, some choice architecture remedies (eg choice

[136] DuckDuckGo's response to CMA's information request [✂].
[137] Note of meeting with [✂].
[138] Article 6(3) of the DMA.

screens and/ or prompts) would benefit from some form of testing and trialling to ensure that these remedies are effective and minimise unintended consequences.

Figure 7.1 Overview of potential choice architecture remedy options at factory settings (Issue 7) and after device set-up (Issue 8)



Source: CMA

7.57 Another point relevant to effectiveness of choice architecture remedies is that user awareness and engagement is already low.[139] Some of this may be a result of a lack of competition or due to the current choice architecture shown to users of mobile browsers, but may also be due to the technical nature of the product and the low level of consumer awareness of browsers and the choices which are available.

Potential unintended consequences

7.58 Some potential unintended consequences of remedy Options C1-C9 are:

(a) Increasing the friction in the user journey at or after device set-up by introducing a choice screen or prompts which may impact user experience. This relates to Options C2, C5 and C8.

(b) A potential impact on OEMs of Android devices by precluding the payment currently made by Google in relation to pre-installations, placement and

[139] For more details see 'WP5 - The role of choice architecture on competition in the supply of mobile browsers (publishing.service.gov.uk)'

defaults in relation to Chrome. This would potentially reduce OEMs' revenue. This is related to Options C1, C2 and C3.

(c) Apple and Google may seek payment from third party browsers in relation to Options C1, C2, C3 and C5 above, and this may marginalise smaller third-party vendors.

7.59 The design considerations section above identified features of remedy design that we think could go some way towards addressing the circumvention risks inherent to the remedy design, but we are still considering the monitoring and enforcement risks which these options may carry, and how they could be mitigated.

Remedy options currently not being considered further

7.60 We have identified several remedies which we consider would not address Issues 7 and 8 effectively. These are:

(a) Prohibiting Safari on iOS and Chrome on Android from being pre-installed at factory set-up.

(b) Prohibiting Safari on iOS and Chrome on Android from being placed in the 'hot seat' or 'dock', or on the default home screen.

(c) Prohibiting browser vendors from displaying default browser prompts and notifications.

Prohibiting Safari on iOS and Chrome on Android from being pre-installed at factory set-up

7.61 We have considered a potential remedy that would prohibit Apple pre-installing Safari on iOS and Google and other OEMs pre-installing Chrome on Android devices.

7.62 We do not consider that this would be effective as having a pre-installed browser may be critical to the functioning of the operating system and might decrease overall user experience.

Prohibiting Safari on iOS and Chrome on Android from being placed in the 'hot seat' or 'dock', or on the default home screen

7.63 We have considered a potential remedy option preventing Safari on iOS or Chrome on Android from being placed on the ''dock'/'hotseat'' or on the default

home screen, minus one or minus two,[140] in the initial configuration of a device at in factory settings.

7.64 This type of remedy is unlikely to be effective in enabling competition between browser vendors on iOS due to Apple's business model of vertical integration as well as Google's Pixel devices on Android. This also might prevent OEMs from placing browsers they want to configure on their devices.

Prohibiting browser vendors from displaying default browser prompts and notifications

7.65 We have also considered a potential remedy option prohibiting all browser vendors from displaying prompts to users about changing their default browser. This might prevent certain larger browser vendors with data advantages leveraging that advantage with targeted prompts and notifications. However, we do not consider that a prohibition of all prompts would be effective as using prompts is an important tool for third-party vendors, particularly those with small market shares.

Invitation to comment on Issues 7-8

7.66 We welcome comments on the emerging views set out above and would particularly welcome views on the following questions:

(a) What are your views on the three proposed choice architecture principles for remedy design (see paragraph 7.6 above)?

(b) Which, if any, of the remedy proposals described above do you think will be most effective and proportionate should an AEC be found?

(c) Which remedies are likely to be effective? Please explain your answer.

(d) Which of the remedies listed above is least intrusive for users? Please explain your answer.

(e) Which, if any, of the remedy proposals described above would offer opportunities for increasing user awareness and engagement?

(f) How important is regulatory alignment and cohesion with existing regulation (eg DMA) when considering choice architecture practices?

[140] The 'minus one' or 'minus two' screen refers to the screens that are respectively one or two swipes from the home screen. Browser apps that are placed further away require more effort from users to access when opening the browser manually.

8. Potential remedies addressing Issues 9 and 10 in cloud gaming services

8.1 As discussed in 'WP6 – Cloud gaming services: nature of competition and requirements for native apps on mobile devices', Apple made various changes to its App Store Guidelines in January 2024. In this section, we focus on remedies that we consider are most likely to address the concerns expressed to us since we recommenced the market investigation in January 2024, as set out in the working paper.[141]

8.2 In this section, we discuss:

(a) A potential remedy concerning Apple's App Store Guidelines in relation to cloud gaming apps (**Option D1**); and

(b) potential remedies for Apple and Google policies on in-app transactions (**Options D2 and D3**).

Potential remedy concerning Apple's App Store Guidelines

8.3 As detailed in the 'WP6 – Cloud gaming services: nature of competition and requirements for native apps on mobile devices', Apple announced that its revised App Review Guidelines (updated on 25 January 2024) will enable Cloud Gaming Service Providers (**CGSP**s) to offer iOS native apps.[142] However, we have received submissions from CGSPs that there remains uncertainty in the App Store Guidelines, and CGSPs identified various Apple Guidelines that could potentially restrict their ability to distribute and operate cloud gaming iOS native apps (see paragraphs 4.25 to 4.35 of 'WP6 – Cloud gaming services: nature of competition and requirements for native apps on mobile devices' working paper).

8.4 Should we conclude that one or more of Apple's Guidelines restrict or prevent the ability of CGSPs from distributing and operating cloud gaming iOS native apps, a potential remedy Option D1 could be:

(a) a requirement for Apple to review and amend its Guidelines to remove specific guidelines that may contain technical or other forms of restrictions on cloud gaming apps;

(b) a prohibition on Apple introducing new restrictions with equivalent effect; and

[141] The issues statement included a broader set of possible remedies, which were focused on Apple App Store Guildelines before the changes were made in January 2024.
[142] 'WP6 – Cloud gaming services: nature of competition and requirements for native apps on mobile devices' paragraph 4.4.

(c) a requirement for Apple to submit regular reports to the CMA setting out the circumstances of any rejection of cloud gaming apps in the UK App Store.

Potential remedy for Apple and Google policies on in-app transactions

8.5 As noted in the 'WP6 - Cloud gaming services: nature of competition and requirements for native apps on mobile devices', a further concern raised by CGSPs is that the requirement that app developers must use Apple's and Google's in-app payment systems for in-game transactions acts as a major barrier to the development of native cloud gaming apps – both in terms of technical feasibility and economic viability.[143]

8.6 CGSPs have told us that enabling in-game transactions is important as the CGSPs use a range of monetisation strategies, including the purchase of virtual goods, upgrades and additional content.

8.7 We are considering two possible remedy options:

(a) Requiring Apple to enable cloud gaming native apps to operate on a 'read-only' basis (ie with no in-game purchases or subscriptions) so that games do not need to be re-coded (and no commission is payable to Apple) (**Option D2**), or

(b) Requiring Apple and Google to allow CGSPs to incorporate their own or third party in-app payment systems for in-game transactions (**Option D3**).

8.8 On Option D2 above, we note that Microsoft currently operates a cloud gaming app on the Google Play Store as 'consumption only' by disabling the relevant in-game purchase code from the standard version of the Microsoft app.[144] However, as noted in 'WP6 - Cloud gaming services: nature of competition and requirements for native apps on mobile devices', the potential disadvantages of operating as a 'read-only' app may include not being able to monetise directly in the app, less discoverability and/or an inferior user experience.

8.9 On Option D3 above, Google has recently offered commitments to the CMA in relation to its in-app payment rules for the Play Store which would permit app developers of 'Digital Content Apps' available on the UK Play Store to offer alternatives to Google Play's billing system (**GPB**) for processing in-app payments for access to digital content or services. As noted in the Notice of Intention to Accept Commitments in the CA98 investigation, a number of considerations would

[143] 'WP6 – Cloud gaming services: nature of competition and requirements for native apps on mobile devices', paragraphs 4.28(b), 4.32 and 4.42.
[144] WP6 – Cloud gaming services: nature of competition and requirements for native apps on mobile devices', paragraphs 4.30 and 4.31.

be relevant to whether this option for app developers is an effective one in practice:[145]

(a) The existence of technical measures that could degrade the user experience of in-app purchases on a third-party system, and/or the use of information screens.

(b) The possibility of circumvention of such a remedy, such as through introduction of new commercial terms, commercial requirements or fees that in effect prevent or restrict CGSPs from implementing their own or third party in-app payment systems.

8.10 We consider this list of examples non-exhaustive and will continue to consider how these and other risks might be mitigated in the design of any potential remedy.

Invitation to comment on cloud gaming remedies

8.11 Do you consider that the remedy options above and/or any other remedies are likely to be effective? Please explain your answer.

[145] Notice of intention to accept commitments (publishing.service.gov.uk).