




Google's Response to Working Paper #7: Potential Remedies

6 September 2024

Introduction

1. Google welcomes the opportunity to comment on the CMA's working paper on potential remedies (**WP7**).
2. As set out in our responses to Working Papers 1-6,¹ the CMA has not demonstrated to the necessary legal standard that there are any adverse effects on competition (**AECs**) on Android. Google therefore disagrees that any of the remedies discussed in WP7 are necessary on Android. This submission responds to the remedies the CMA is considering should it nonetheless find that any AEC exists on Android.
3. []. In this submission, Google also provides its views on the remedies the CMA is considering in respect of iOS.
4. Our views on the proposed remedies focus on: (i) whether they are necessary and well-targeted to address any potential AEC that the CMA has identified on a preliminary basis; (ii) whether they are proportionate (for example, given the implementation costs involved) and the least onerous where there is a choice of remedy; (iii) whether any relevant consumer benefits (**RCBs**) would be lost by implementing them; and (iv) whether they would result in any unintended consequences.
5. This submission is structured as follows:
 - [Section I](#) addresses WP7's proposed remedies on the WebKit restriction and access to functionality (Options A1-A4).
 - [Section II](#) addresses WP7's proposed remedies on in-app browsing (Options B1-B6).
 - [Section III](#) addresses WP7's proposed remedies on choice architecture (Options C1-C9).
 - [Section IV](#) addresses WP7's proposed remedies on cloud gaming (Options D1-D3).
 - [Section V](#) addresses the cross-cutting themes described in WP7.

¹ Available on the [CMA's website](#).

I. WebKit Restriction and Access to Functionality

A. Google's Comments on Options A1-A3


6. Apple does not allow third-party browsers to compete at the browser engine-level on iOS by modifying or forking WebKit or using an alternative browser engine (like Blink or Gecko) (the **WebKit Restriction**). Allowing browser engine choice on iOS would allow browsers to compete more effectively on security, privacy, and performance, and promote compatibility in the web ecosystem, without compromising security. We expand on this in our response to Working Paper 2 (**WP2**), and welcome the CMA's emerging thinking on this issue.
7. WP7 sets out three alternative options for requiring Apple to allow third-party browser engines on iOS. We set out our views on these options below. In short:
 - Google supports Option A2, which would enable third-party browsers to use alternative browser engines on iOS with equivalent access to APIs used by WebKit and Safari (and, where appropriate, third-party apps).
 - Option A1 would be ineffective in resolving the WebKit Restriction, as Apple would be under no obligation to provide access to equivalent features and functionalities that WebKit and Safari have access to (as WP7 acknowledges²).
 - Option A3, by contrast, may be disproportionately specific and risk unintended consequences for performance and security. This is because exposing the same process architecture to third-party browsers and browser engines that Safari uses may cause performance and security issues.
8. We expand on this feedback below.
9. **Option A2 is most appropriate for facilitating effective browser engine competition on iOS.** Option A2 would allow third-party browser engines on iOS and require Apple to grant browsers using alternative browser engines equivalent access to APIs that WebKit and Safari have on iOS. This would place Webkit and non-Webkit based browsers on "*an equal footing*"³ on iOS. Apple could do this by building "*new APIs replicating the functionalities and features made available to WebKit and Safari*".⁴ Alternatively, Apple may choose to make existing private APIs available if it can do so in a safe and secure manner.

² WP5, ¶5.18(a).

³ WP5, ¶5.18(b).

⁴ WP7, ¶5.27.

10. In principle, this remedy option could be an effective means of remedying the WebKit Restriction, subject to the following six principles:

- ***First, it may be necessary to extend access to functionality to features and functionalities used by iOS apps other than Safari.*** Any remedy should prevent policies or technical limits that restrict browsers from accessing APIs available to other non-browser apps (even if not used by Safari), if browsers routinely access and use such APIs on other platforms. .
- ***Second, effective browser engine choice must be accompanied by adequate support and documentation.*** We agree with WP7’s proposal that any remedy to address the WebKit restriction would need to provide for “clear and complete documentation that is kept up to date”.⁵ Public documentation and support channels would also enable third-party developers to undertake a monitoring role, minimising the need for ongoing monitoring by the CMA.
- ***Third, equivalent features and functionalities should be granted to third-party browsers at no cost.*** WP7 recognises that APIs “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” (emphasis added).⁶ We agree. Apple has already built many of the necessary features and functionalities to lift the WebKit restriction in the EU under the Digital Markets Act (**DMA**) (which it is required to make available to third parties for free).
- ***Fourth, equivalency should be validated through scrutiny by other stakeholders.*** Uncertainty over what constitutes “equivalent” features and functionalities risks undermining Option A2’s effectiveness. This risk can be mitigated, however, through public bug trackers where experts can scrutinise and comment on various stakeholders’ positions on specific features. This is the approach taken already by Chromium and Android’s open-source development forums. Openness and transparency as a requirement in Option A2’s implementation would reduce the need for ongoing monitoring.
- ***Fifth, equivalency must extend to performance metrics.*** We agree with the emerging thinking in WP7 that equivalent treatment of third-party browser engines on iOS would require “access to a full range of metrics to

⁵ WP7, ¶5.36.

⁶ WP7, ¶5.40.

allow all browser vendors on iOS to measure the performance of their respective browsers.”⁷ [✂].

- **Sixth, effective browser engine choice must not involve arbitrary policy requirements that undermine effectiveness.** WP7 observes that despite the requirements under the DMA, there are “no live browser apps in the EEA using alternative browser engines on iOS.”⁸ [✂]. Any WebKit remedy must therefore prohibit arbitrary policy requirements that undermine effectiveness. [✂]

11. If Option A2 is implemented in a way that takes account of the above considerations, we think the remedy would be effective and there would be minimal ongoing monitoring and enforcement costs.

12. **Option A1 would be ineffective as it is insufficiently specific to enable effective browser engine competition.** Option A1 would consist of a “high-level, principles based requirement for Apple to enable access to alternative browser engines on iOS”.⁹ It would not, however, require Apple to provide access to equivalent features and functionalities that WebKit and Safari have access to.¹⁰ This is a critical gap that would prevent Option A1 from having any useful effect. To launch competitive browsers based on alternative browser engines, third-party browsers would at least need access to equivalent features and functionalities that Apple makes available to Safari and WebKit. Accordingly, as WP7 acknowledges, addressing the WebKit Restriction under Option A1 would pose a significant “effectiveness risk”.¹¹

13. [✂].

14. **Option A3 is disproportionately specific and risks unintended consequences.** Option A3 would require Apple to facilitate access to the same APIs used by both Safari and WebKit, both now and in the future.¹² While this approach may be an appropriate option in circumstances where Apple cannot (yet) make equivalent new functionality available, it may risk unintended consequences that undermine its effectiveness. In particular, because not all web browsers share Safari’s architecture, exposing only the exact APIs to third-party browsers might limit either the functionality or security that could be achieved by a Blink-based browser.

⁷ WP7, ¶5.42.

⁸ WP7, ¶5.21.

⁹ WP7, ¶5.19.

¹⁰ WP7, ¶5.18(a).

¹¹ WP7, ¶5.20.

¹² WP5, ¶5.18(c).

15. In addition, WebKit or Safari may currently access lower-level iOS features that could create security risks if exposed in the same form to third-party apps. Accordingly, Apple may need to facilitate equivalent or indirect access to such features by exposing alternative APIs in a safe and secure manner.

B. Google’s Comments on Option A4

16. WP7 suggests a requirement for Google to provide “*equivalent access and functionality*” to third-party browsers, to address any potential AEC resulting from the alleged inability of third-party browsers on Android to access certain functionality.¹³ We make three observations in this connection:

- First, there is no AEC resulting from lack of access to functionality on Android.
- Second, a general interoperability remedy would be unreasonable and disproportionate, even if the CMA were to find an AEC.
- Third, any remedy concerning WebAPK minting should be designed in terms of equivalent access.

17. We expand on these points below.

18. **There is no AEC resulting from lack of access to functionality on Android.** All browsers on Android have access to the functionality they need to compete effectively. This is borne out by the range of differentiated browsers on Android—over 20 in total—that UK consumers regularly use. Of the six issues identified in Working Paper 3 (**WP3**), five have been resolved or are not features that are restricted to third parties. The one exception, WebAPK minting, is not competitively significant, nor is it necessary for the web app ecosystem to flourish on Android. Therefore, access to functionality on Android cannot plausibly give rise to an AEC and no remedy is necessary. We expand on this in our response to WP3.

19. **A general interoperability remedy would be unreasonable and disproportionate.** Should the CMA nevertheless identify an AEC in respect of the narrow issue of WebAPK minting and propose a remedy, a general requirement for Google to grant equivalent access to APIs used by Chrome (*i.e.*, Option A4) would be unnecessary, unreasonable, and disproportionate, for the following reasons:

- ***A broader equivalent access requirement is unnecessary on Android.*** WP7 acknowledges that the concern in respect of access to functionality by third-party browsers on Android is “*less pronounced*” than on iOS, as “*Google already makes most APIs public*”, with lack of access to WebAPK

¹³ WP7, ¶5.9.

minting being the main issue highlighted.¹⁴ It considers that the evidence to date shows “*it may be sufficient that Google enables access to the WebAPK minting functionality*”.¹⁵

We agree that if the CMA concludes that third-party browsers’ lack of access to WebAPK minting on Android gives rise to an AEC (which it does not),¹⁶ it would not be appropriate to remedy this issue through a general interoperability obligation. The CMA’s investigation has shown that third-party browsers either already have access to the APIs they need, or have no interest in receiving access to additional features and functionalities. A broad interoperability remedy is therefore not necessary or well-targeted to resolve any potential AEC on Android.

- **A broader equivalent access requirement would create unreasonable and disproportionate implementation and ongoing monitoring costs.** A general equivalent access requirement would necessitate a wholesale review of *all* Google’s first-party features and functionalities to determine if and how Google could safely and securely facilitate access to third parties. This would include features and functionalities that third parties would not find useful or relevant and that no third party has expressed any interest in having access to. Such an exercise would be significant, time-consuming, costly, and unreasonable. A broader access requirement would also require ongoing monitoring in circumstances where Google already provides browser vendors with access to the vast majority of features and functionalities they want, on an equal footing with Chrome.

20. We therefore agree with WP7’s suggestion that, if the CMA finds an AEC and proposes a remedy, it would suffice that Google enables access to web app installation.¹⁷ Among the menu of potential options, this would be the least unreasonable and disproportionate remedy to resolve the CMA’s concerns if it finds an AEC.

21. **Any remedy concerning WebAPK minting should be designed in terms of equivalent access.** If the CMA proposes a remedy concerning WebAPK minting, it should be designed as a requirement for Google to provide equivalent access to relevant features and functionalities (akin to Option A2) rather than to provide access to the very same API that Chrome accesses today (akin to Option A3). Enabling access to the same (rather than equivalent) APIs could risk serious unintended consequences in the form of:

¹⁴ WP7, ¶¶5.69, 5.72.

¹⁵ WP7, ¶¶5.69.

¹⁶ See Response to WP3, ¶¶13.

¹⁷ WP7, ¶¶5.72.

- Exposing users to malware, as there are security risks of apps automatically installing other apps that a user has not expressly requested or—in the case of sideloaded apps—expressly consented to.
 - Inhibiting Google from evolving the relevant APIs in ways that mitigate any security or privacy risks (e.g., if the remedy is framed in a way that leaves Google with no freedom to alter APIs if new security issues emerge).
 - Resource exhaustion, as the same API being called on by multiple actors could risk it being flooded with so much traffic that it becomes inaccessible to users.
22. If the CMA imposes this remedy, Google should be free to implement it in the safest and most secure manner, including by building new APIs that enable third-party app developers to install web apps on Android devices on an equal footing with Chrome. This would minimise the risk of unintended consequences such as exposing users to unacceptable security risks and compromising the integrity of Android devices.

C. Response to the CMA’s Invitation to Comment

(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?*

23. As explained above, Google supports Option A2 if it is properly implemented. For completeness, though, Google notes that:
- If a browser based on a non-WebKit browser engine adds a web app to the user’s home screen, that web app should be powered by the same browser (and its underlying browser engine) that added the web app to the home screen.
 - Google supports WP7’s position that any remedy applied to iOS should also apply to iPadOS.

(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.*

24. As explained above, in our view Option A2 is a more appropriate and effective remedy option than Option A3 to resolve the WebKit Restriction (Issue 1).
25. Issue 2 (access to functionality) on iOS could be addressed more appropriately by targeted interventions designed to address the issues identified in Working Paper 3 which the CMA may find give rise to an AEC, rather than a general interoperability obligation.

(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?**

26. [✂].

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

27. Security and privacy standards must be reasonable, objective, and clearly defined. They must be proportionate to the risk and should not lead to arbitrary and excessive requirements. [✂].

(e) **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?**

28. As described above, any circumvention risks resulting from Options A2 can be effectively managed [✂].

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

29. Option A2 may increase the monitoring and enforcement risk as it allows a choice between “new APIs replicating the functionalities and features made available to WebKit and Safari” and “access to some of the existing private APIs that exist as internal interfaces within iOS”.¹⁸ But as described above, these risks can be effectively managed and mitigated by public documentation of relevant APIs and public scrutiny of and debate about the equivalency of access that third-party browsers and browser engines are granted.

30. While Option A3 may require a lower level of monitoring and enforcement, it could lead to unintended consequences because not all web browsers share Safari's process architecture. As explained above, access to the very same APIs that Safari and WebKit use could therefore pose inherent limitations to non-WebKit based browsers on iOS as regards their performance, functionality, and security. Option A1 would create the most need for ongoing monitoring, as Apple would have more flexibility over how it facilitates third-party browser engines on iOS.

¹⁸ WP7, ¶5.27.

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

31. We have not identified any RCBs that Option A2 (which we support) would sacrifice. As explained above, privacy and security risks inherent to facilitating browser engine choice on a mobile platform can be managed through security and privacy measures, including necessary and proportionate minimum standards. In any event, allowing effective browser engine choice on iOS would bring net benefits to security and privacy by allowing browsers to compete and innovate effectively on these parameters on iOS.

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

32. [✂].
33. [✂].
34. [✂].

(i) 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?

35. As explained above and in response to WP3, there is no AEC on Android resulting from browsers' lack of access to functionalities that Chrome has access to.
36. Lack of access to WebAPK minting in particular does not restrict third-party browsers' ability to compete on Android. But if the CMA were to find an AEC and a remedy were deemed necessary, providing equivalent access to WebAPK minting would be the only proportionate remedy, as it would be the least unreasonable and disproportionate of a choice of effective remedies, and would give rise to the lowest possible risk of unintended consequences.

II. In-App Browsing

37. As explained in our response to the CMA's Working Paper 4 (**WP4**), Android's in-app browsing (**IAB**) policies cannot plausibly give rise to an AEC. Google's IAB policies facilitate developer freedom to design the IAB experiences that suit their users' needs and their business models. And at the same time, Google's IAB policies promote user choice and control in relation to the browsers used for IAB.
38. As WP4 recognises, our practices are unlikely to limit competition among mobile browsers on Android as they do not prevent rivals from offering competing products. Accordingly, Options B4 to B6 in relation to Android are not necessary.

39. [🔗].

A. Google's Comments on Options B1-B3

40. Google supports the three options set out in WP7 for requiring Apple to allow user and developer choice in respect of remote tab and webview IABs on iOS.¹⁹ We explain our views on these options below. In short, Google supports Options B1 to B3, which support choice for developers and customisation of their apps to meet their users' needs and suit their business models. Specifically:

- Google supports Option B1, which would require Apple to enable remote tab IABs for WebKit-based browsers. But Option B2, which would require Apple to enable remote tab IABs for browsers wishing to use *alternative browser engines*, must be implemented in parallel. This is also required to ensure the effective implementation of Option A2 (discussed in **Section I** above). Otherwise, non-WebKit based browsers would be prohibited from competing in remote tab IABs, restricting their ability to access and support users and likely disincentivising browser vendors from launching non-WebKit based browsers on iOS.
- Google also supports Option B3, which would allow third-party browser engine providers to offer webview and bundled engine IABs on iOS. As above, this remedy is required to ensure the effective implementation of Option A2.

41. We expand on these points below.

42. **Options B1 to B3 would promote competition, differentiation, multi-homing, and innovation in native apps' IAB implementations.** WP4 recognises the benefits that developer choice over IAB implementations can bring,²⁰ allowing further room for differentiation and innovation in native apps. Enabling app developers to choose from alternative browsers for remote tab IABs and browser engines for webview IABs would increase their ability to innovate and tailor their IABs to their app interface, ultimately benefiting users. This would match the current approach on Android today.

43. **Google supports Option B1, which is necessary due to the importance of remote tab IABs.** As explained in Google's response to WP4, offering remote tab IABs is an important way for browser vendors to compete by supporting their users more effectively and increasing the time users spend in their browser.²¹ Remote tab

¹⁹ WP7, ¶¶6.4-6.27.

²⁰ WP4, ¶5.8. See also Meta's WP4 response, which describes the "*real benefits*" that enabling IABs to have custom browser engines has brought.

²¹ See Google's response to WP4; and WP4, ¶¶2.36; 4.8.

IABs based on other browsers are technically feasible on iOS and any justifications for a restriction based on security are unfounded.²²

44. **Option B1 alone would not effectively remedy Apple’s restriction on third-party remote tab browsers.** WP7 considers that Options B1-2 “*could be implemented in combination or be mutually exclusive*”.²³ But Option B1 would be insufficient to remedy the AEC in relation to Apple’s remote tab policy, as it would prohibit non-WebKit based browsers from reaching users in remote tab IABs, hindering the ability of non-WebKit based browsers to compete. This would undermine the aim of a remedy in relation to the WebKit restriction, as discussed above. Any implementation of Option B1 must therefore be combined with Option B2 to be effective.
45. **Option B3 would promote competition in browser engines and allow app developers to customise their apps.** Apps offering in-app browsing on iOS are limited in their ability to change the way data is handled and content rendered by the underlying WebKit engine. Option B3 would provide app developers with choice between browser engines, increasing the opportunities available for third-party browser engines to compete and for native app developers to customise their apps. [🔗].

B. Google’s Comments on Options B4-B6

46. WP7 identifies three possible remedies to address the issue identified by the CMA’s emerging thinking that Apple’s and Google’s IAB policies offer users limited choice and control in relation to which browser is used for IABs in native apps:
- A requirement for Apple and Google to implement remote tab IABs using the user’s default browser (Option B4).
 - A requirement for Apple and Google to make users aware they are in IABs by implementing changes to the interface or disclosures (Option B5).
 - A requirement for Apple and Google to implement opt-out settings for IABs (Option B6).
47. For the reasons given in our response to WP 4, there is no AEC on Android resulting from lack of user choice or control over IABs. Custom Tabs, which is used by most app developers for IABs, implement the user’s underlying default browser by default. We provide users with options to exit IABs both in our apps and in Chrome’s implementation of Custom Tabs. And in two of our apps commonly used for in-app browsing—Google Search and Gmail—we allow users to switch off IABs altogether.

²² See further Google’s responses to question (a) of WP7’s invitation to comment below.

²³ WP7, ¶6.9.

48. But even if the CMA disagrees and concludes that our approach to IABs on Android gives rise to an AEC, the remedy options described above would be unnecessary, disproportionately costly (for both app developers and Google) to implement, and/or risk various unintended consequences including the loss of RCBs. No remedy should therefore be imposed on Google even if the CMA finds an AEC. We expand on these points in more detail below.
49. **Option B4 (remote tab IABs using the user’s default browser) would be unnecessary on Android, disproportionately costly to implement and enforce on app developers, and risk unintended consequences.** Option B4 would require Apple and Google to implement remote tab IABs using the user’s default browser. WP7 provides two alternatives for this remedy:²⁴
- Allowing app developers to change the browser they use for remote tab IABs, as is currently the case on Android (**Option B4(i)**).
 - Using the user’s default browser for remote tab IABs without app developers being able to override the user’s default browser and specify a particular browser (**Option B4(ii)**).
50. We support Option B4(i), which reflects the *status quo* on Android. As explained in our response to WP4—and recognised in WP4 itself²⁵—native app developers should have the freedom to specify a particular browser to power their remote tab IABs so they can facilitate access to features in specific remote tab IAB implementations that not all browsers support.
51. Option B4(ii) would be unnecessary, disproportionate, and risk unintended consequences. In particular:
- **Option B4(ii) is unnecessary on Android.** Option B4(ii) would have limited impact and be poorly targeted on Android. This is because Android app developers generally use the default Custom Tabs APIs, which will use the user’s default browser (in line with our Custom Tabs documentation, which generally does not provide guidance on hard coding to a specific browser in the first place).²⁶ This remedy is therefore not necessary on Android and would have limited impact compared to its costs and potential unintended consequences.
 - **Option B4(ii) would be disproportionately costly to implement and enforce on app developers.** Option B4(ii) would require significant engineering work by Google, app developers, and/or browser vendors to

²⁴ WP7, ¶¶6.35 -39.

²⁵ WP4, ¶¶5.9.

²⁶ WP4, ¶¶2.10(b) (“In most cases, apps invoke the user’s default dedicated browser in Custom Tabs mode”).

implement. Possible methods for implementing this option on Android would involve disproportionate costs for those involved, and Google encourages the CMA to gather feedback from app developers on this topic. In particular:

- First, Google could implement Option B4(ii) at OS-level by [redacted].²⁷ This would involve significant time, costs, and engineering resources [redacted]. [redacted]. This would impose disproportionate costs and disruption on app developers.
- Second, Google could [redacted] require app developers to respect the user default in all cases through its Play Store policies. [redacted].

In addition, app developers that currently hardcode to a specific browser would need to re-code their apps, involving significant engineering time and cost. For example, [redacted]. For app developers with fewer resources, this timeframe and the relative cost is likely to be much higher.

Option B4(ii) would therefore result in significant monitoring difficulties and costs. In any event, any requirements enforced on app developers through Google Play policies would not extend to sideloaded apps or apps downloaded through third-party app stores (e.g., the Galaxy Store).

- Third, Option B4(ii) could be implemented via browser vendors making changes to their Custom Tabs implementations. This would, however, be challenging to implement correctly and still require Google to monitor browser vendors manually, which would incur significant engineering costs for browser developers.²⁸

- **Option B4(ii) would risk unintended consequences.** In particular:

- First, it may result in app developers switching to webview IABs. WP7 notes that such a remedy “*may create uncertainty for app developers*” in relation to whether the user’s default browser’s Custom Tabs version can support the developer’s features.²⁹ This may lead to greater uptake of webview IABs by app developers who wish for greater certainty over specific features being present in their IABs. This in turn could result in app developers having to take

²⁷ [redacted].

²⁸ [redacted].

²⁹ WP7, ¶6.43.

responsibility for the security and privacy capabilities of their IABs when they may not want to³⁰ and/or result in lower quality browsing in terms of safety and privacy for users. Google encourages the CMA to gather feedback from app developers on this topic.

- Second, it would sacrifice important RCBs. As WP4 notes, “*there are benefits in allowing app developers ... to have some degree of choice over the IABs in their app*”.³¹ Developers have the choice on Android to specify a particular browser to power their Custom Tabs, allowing room for app differentiation as they may wish their IABs to offer, or not offer, certain features available through particular browsers’ Custom Tabs. Ultimately, this choice allows developers to innovate their apps and improve the user’s mobile experience. A remedy prohibiting this developer freedom would sacrifice these RCBs.

52. Accordingly, only Option B4(i) would be reasonable and proportionate.

53. **Option B5 (greater user awareness of IABs) is unnecessary on Android, would be technically challenging and costly to implement, and would risk unintended consequences.** Option B5 would require Apple and Google to make users aware they are in IABs by implementing changes to the interface or implement disclosures. For the following reasons, however, such a remedy would be unnecessary on Android, be disproportionately difficult for Google to implement beyond Custom Tabs, and risk unintended consequences (by incentivising developers to use alternative IAB implementations to Custom Tabs and creating unnecessary friction for users):

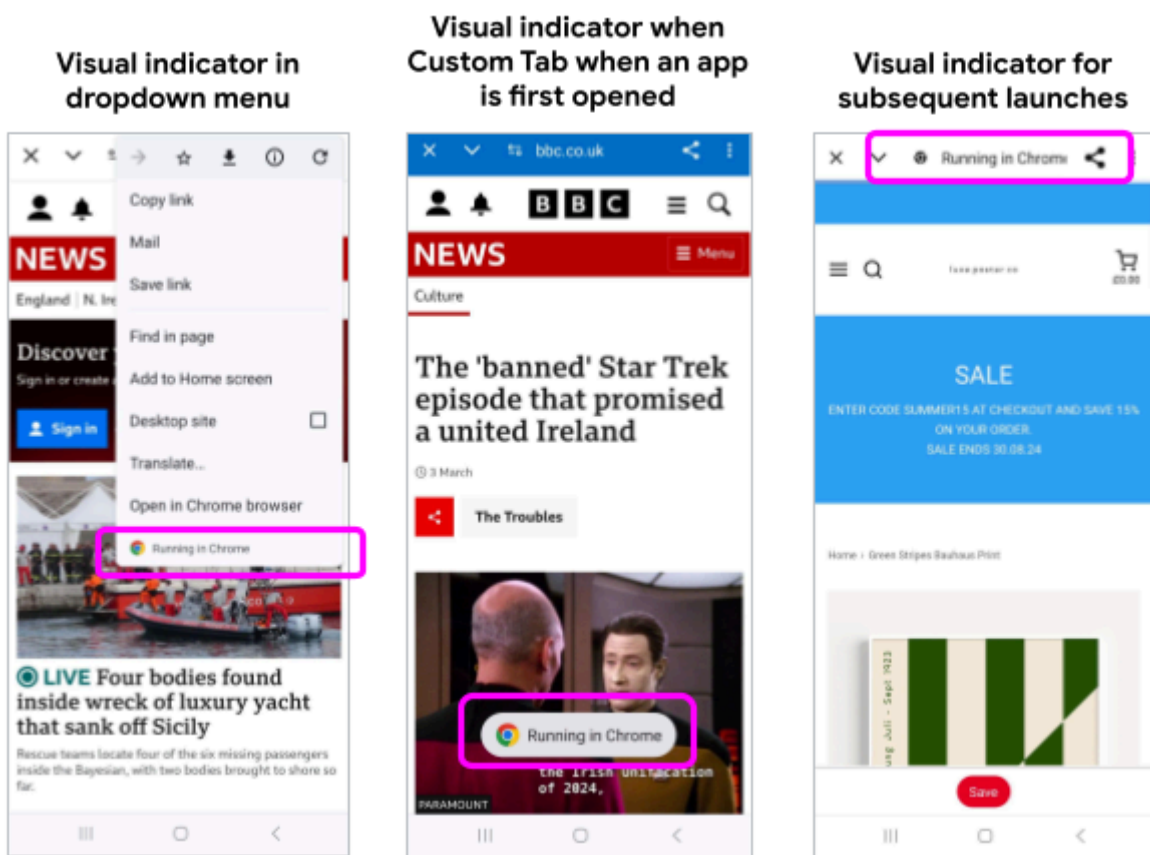
- ***Requiring interface changes or information disclosures are unnecessary on Android.*** This is confirmed by two main points:
 - First, Verian’s qualitative consumer research demonstrates that users are generally aware that they are not in their dedicated browser app and had remained within the same app after clicking a link. Users responded that they could distinguish between leaving the app “*for a browser*” versus remaining within the app.³²
 - Second, Chrome Custom Tabs—which is the only IAB implementation that Google is responsible for—already clearly informs users that they are in Chrome via visual disclosures. In particular:

³⁰ Developers may not be able to replicate features not available in their preferred Custom Tabs implementation themselves, such as certain web platform features that the browser does not support or Custom Tabs features that the remote tab IAB makes available in the surrounding user interface, such as Partial Custom Tabs.

³¹ WP4, ¶5.9.

³² Verian consumer research presentation of key qualitative findings, p. 31.

- Chrome Custom Tabs shows an indicator in the dropdown menu at all times.
- The first time an app ever launches a Chrome Custom Tab, Chrome shows a pop-up that says “Running in Chrome” with a Chrome logo.
- For subsequent launches, if it has been more than an hour since branding was shown for a Chrome Custom Tab launched from the given app, Chrome shows “Running in Chrome” text and a monochromatic Chrome logo in the top toolbar.



Third-party browsers are able to implement similar visual indicators in their IAB implementations. For example, Meta has implemented visual indicators that users are in Facebook’s and Instagram’s IABs. We agree with Meta’s argument that “it would be a profound error to respond to certain users’ lack of awareness either by curbing developers’ freedom to utilize IABs or by failing to address OS restrictions on that freedom.”³³

³³ Meta’s WP4 response, ¶13.8.

- **Requiring interface changes or information disclosures could not be launched uniformly on Android.** It is only technically possible for Android to show a disclosure screen *in a Chrome Custom Tab IAB*. It is not possible for Android to do so in non-Chrome Custom Tabs, webview IABs, or own-bundled IABs for the following reasons:
 - First, with webview IABs and own-bundled IABs, developers build their own custom IAB experiences from scratch. [✂️].
 - Second, third-party browsers control the visual interfaces of their Custom Tabs implementations.

Google is therefore unable to implement additional visual indicators for users of non-Chrome Custom Tabs, webview, or own-bundled IABs. Such a remedy would need to be enforced directly on app developers, giving rise to significant monitoring challenges and implementation costs for the relevant app developers. Google encourages the CMA to gather feedback from app developers on this topic. In addition, this would not be effective as enforcement will need to happen through a policy enforced by the Play Store which would result in sideloaded apps or apps coming from a third-party app store necessarily escaping these requirements.

- **Requiring interface changes or information disclosures would risk unintended consequences.** Option B5 could harm both users and developers:
 - It risks reducing the attractiveness of Custom Tabs as an IAB solution and may therefore result in more developers choosing to create a bespoke solution based on a webview or own-bundled engine, which would result in the adverse consequences described at ¶51 above for Option B4.
 - It risks creating user frustration given that additional disclosures are not warranted every time an IAB opens, as not every IAB involves content that requires high user awareness (e.g., content that prompts the provision of financial information and thus the possibility for scams or fraud).

54. **Option B6 (IAB opt outs at device level) is unnecessary on Android, would be disproportionately costly to implement and monitor, and would risk unintended consequences.** Option B6 would require Apple and Google to implement opt-out settings for IABs. On Android, this remedy is unnecessary, would be disproportionately costly for app developers and burdensome to monitor, and risk unintended consequences by removing user choice, resulting in a loss of RCBs.

- **A device-level opt out is unnecessary on Android.** On Android, users can easily exit most IABs—including Chrome Custom Tabs—to their default

browsers, or opt out of in-app browsing altogether in some apps' settings. The availability of these choices is set by (i) browser vendors (in the case of options available in Custom Tabs implementations); and (ii) app developers, in the case of options available in apps themselves. We facilitate this choice for users in both respects:

- In Chrome Custom Tabs, we provide a permanent option for users to open the relevant content in their default browser.
- In popular Google apps like Google Search and Gmail, we provide users with options to turn off in-app browsing altogether via a setting in the app.

Other app developers facilitate similar choices for their users including, for example, Facebook, LinkedIn, Reddit, and Instagram. A device-level opt out is therefore unnecessary, as Android users already have choice and control.

- **A device-level opt out would impose disproportionate costs on developers and require ongoing monitoring.** Option B6 gives rise to similar concerns as Option B4(ii) around developer costs and monitoring.³⁴ A device-level setting would only be necessary for IABs based on a webview or own-bundled engine, as Custom Tabs-based IABs are already powered by a browser (usually the user's default browser) and there would be little additional user benefit from an opt out in these circumstances. But implementing this remedy vis-à-vis apps using a webview or own-bundled engines would be extremely difficult, if not impossible, for Google to monitor or enforce. Android would have no way of "inserting itself" into the app's code to open links in the app's custom IAB only when the user's setting permits it.

The costs involved for all app developers in such a scenario would likely be prohibitively high. Google encourages the CMA to gather feedback from app developers using webview or own-bundled IABs on this topic.

- **A device-level opt out would risk unintended consequences and result in a loss of RCBs.** Currently, users can opt out of or exit IABs with respect to specific apps. A device-level opt out is neither necessary nor beneficial for the user experience as: (i) users can already opt out of or exit IABs when they want to in the majority of apps; and (ii) users may prefer IABs for some apps but not others. As explained in our response to WP4, Google's research has found that [🔗]. A system-wide setting could sacrifice these fine-grained controls users currently benefit from, in favour of a binary choice (as it is unclear if app developers would be able to "override" the user's system setting by offering an app-specific choice).

³⁴

See above at ¶51.

C. Response to the CMA's Invitation to Comment

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

55. [🗂️].

56. [🗂️].

(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?

57. See the response to (a) above. [🗂️].

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

58. See Google's comments on Option B4(ii) above.

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

59. As explained in our response to WP4, there is no plausible risk of an AEC in relation to Google's webview IAB policy. There are therefore no appropriate remedy options.

(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?

60. As explained above, it is not necessary to increase user awareness and consent for in-app browsing on Android. The CMA's research confirms that users are aware they are in IABs, and Chrome Custom Tabs—which is the only IAB implementation

that Google is responsible for—already clearly informs users that they are in Chrome via multiple visual disclosures. There is therefore no AEC on Android relating to users’ lack of awareness of IABs.

61. We have also explained above how information disclosure or device-level opt outs are unnecessary, disproportionately costly to implement and monitor compliance with, and risk unintended consequences including the loss of RCBs. Even if the CMA identifies an AEC on Android, it is not clear that there is a viable remedy option that could resolve any AEC in a proportionate manner.

III. Choice Architecture

A. Google’s Comments on Options C1-9

62. WP7 proposes nine potential remedies to address its emerging concerns in relation to choice architecture on iOS and Android. As explained in our response to the CMA’s Working Paper 5 on Choice Architecture (**WP5**), Google’s practices and choice architecture do not give rise to any AEC. In fact, they result in RCBs and enable Android OEMs to compete more effectively through ensuring a high-quality “out-of-the-box” experience. Even if the CMA were to conclude that Google’s choice architecture results in an AEC (which it does not), no remedy should be applied on Android for the reasons given below.

63. **Choice architecture remedies are better suited to enforcement and monitoring by the Digital Markets Unit (DMU) under the new UK digital regulatory regime.** The CMA has recognised that *“the effectiveness of active choice as a remedy is hugely dependent on the nature of the choice and its design.”*³⁵ Presenting users with an active choice *“does not in itself make it meaningful or one that actually gives consumers what they might want, had they not been constrained by what is on offer.”*³⁶ We agree that choice architecture is a particularly complex area of regulatory intervention:

- Choice architecture remedies risk unintended consequences by introducing too many choices or options, unduly complicating the choices users have to make, or prompting users to make choices too frequently.
- Choice architecture remedies involve complex design decisions. For example, choice screen remedies (Options C2 and C5) require careful consideration of (among other things) which devices should be subject to

³⁵ CMA, [Online Choice Architecture - How digital design can harm competition and consumers](#) (April 2022), ¶4.33.

³⁶ Ibid.

any choice screen,³⁷ when and how any choice screen should be displayed (i.e., the appropriate moment in the user's journey), the number of options shown, the criteria for selecting browsers to appear in the choice screens, the order of the browsers, and other information to be displayed on any choice screen.

- By nature, choice architecture remedies deprive device manufacturers of the ability to make design decisions that are responsive to users' preferences and to offer the device configurations they believe will most likely succeed in a competitive environment.
64. Choice architecture interventions may therefore require iterative design, development, and testing. The new UK digital regime is designed to provide for this sort of flexibility, ongoing dialogue and monitoring, and testing. Implementation of potential choice architecture remedies—including deciding on the right mix of remedies to impose—may therefore be better suited to the new regime and the DMU.
65. **Option C1 (preinstalling multiple browsers) is not necessary on Android, may create unintended consequences, and may result in the loss of RCBs.** Option C1 would require Apple and Google to ensure that multiple browsers are preinstalled, using defined criteria.³⁸ Preinstalling more browsers on Android devices would, however, be unnecessary, ineffective at increasing user choice, and risk unintended consequences and a loss of important RCBs. In more detail:
- ***Preinstalling more browsers would be neither necessary nor effective at increasing user choice or resolving the CMA's emerging concerns.*** This is borne out by the following two considerations:
 - First, Android OEMs choose which browsers to preinstall, and are free to preinstall as many as they want. They are strongly incentivised, in a competitive OEM environment, to preinstall high-quality apps and create an attractive out-of-the-box experience for their users. It can therefore be assumed that OEMs are already preinstalling the optimal number of browsers to suit their users needs (which, data show, include 1-2 browsers on most Android devices).
 - Second, mandating preinstallation of multiple browsers is unnecessary because users have high confidence in downloading

³⁷ For example, WP7 does not discuss whether choice screens would apply to Android devices where Chrome is not set as default (which represent over half of UK Android device shipments). If the CMA finds an AEC on Android resulting from a browser's status as the device's initial default, any choice screen remedy should apply symmetrically to cover all devices with a default browser (irrespective of whether the default browser is Chrome).

³⁸ WP7, ¶¶7.12-7.15.

alternatives. Verian’s research found that 85% of UK users feel confident in their ability to download a new web browser without assistance.³⁹ This can be done easily through Play, third-party app stores, or sideloading via an existing browser. WP5 therefore finds that, among users who had *not* switched browsers, there was “no concern” about having a preinstalled browser because “if [the user] cared about the browser they used they could open a website in whichever browser they preferred.”⁴⁰

- **Preinstalling more browsers on Android devices may result in unintended consequences in the form of a low-quality out-of-the-box experience.** OEMs can and do preinstall multiple apps in the same category if, in their view, users would benefit from a better out-of-the-box experience. The Android agreements at issue allow OEMs this freedom. In these circumstances, forcing OEMs to preinstall more browsers than the ones that (in their view) are highest quality risks degrading the out-of-the-box experience for users by preinstalling browsers that users have no interest in using.
- **Preinstalling more browsers on Android devices may sacrifice important RCBs.** As explained in Google’s response to WP5, Google’s Android agreements create an additional revenue stream for Android OEMs that choose to enter into them (see further **Section III.B** below). This allows OEMs to invest in device quality and innovation and pass on the benefit from the money to end users in the form of lower device prices.⁴¹ Requiring multiple browsers to be preinstalled would undermine the value of the preinstallation opportunities browser vendors pay for and would risk reducing the flow of money which helps fund cheaper and higher-quality devices.

66. **Options C2 and C5 (choice screens) are unnecessary on Android, would risk unintended consequences, would be technically complex and costly to implement, and may result in the loss of RCBs.** Options C2 and C5 would require Apple and Google to implement choice screens on iOS and Android devices (i) when users first set up their devices (Option C2); and (ii) after users set up their devices (Option C5). However, choice screens on Android are not necessary to ensure user choice of browsers. Even if they were, there are several reasons why choice screens would not be appropriate, including a high risk of unintended

³⁹ Verian Quantitative Consumer Research Report, slide 27 (comprising “Probably” (28%) and “Definitely” (57%)).

⁴⁰ WP5, ¶4.19.

⁴¹ See Google’s Response to WP5, ¶61.

consequences due to various design considerations, significant cost and complexity of choice screen roll-outs, and the risk of sacrificing important RCBs. In more detail:

- **Choice screen remedies on Android are not necessary to ensure user choice of browsers.** On Android, users have clear paths to set their preferred browser as default via settings and default prompts by downloaded browsers. The CMA's research found that 8 in 10 users could change their default browsers if they wanted to and approximately 90% of those who *had* changed their default browser found the process easy.⁴² Showing a choice screen to users who prefer their initial default browser causes unnecessary friction, especially when users can easily switch at any time if they want to.
- **Choice screen remedies risk unintended consequences.** Choice architecture remedies like choice screens risk unintended consequences that undermine their effectiveness if they are not properly designed and tested. For example:
 - Showing choice screens at a sub-optimal point in the user's journey may undermine the effectiveness of their choices. For example, a user may be better placed to make a choice of default browser when setting up their device for the first time (when they are used to making decisions on various settings), as opposed to when they use their browser for the first time (when they are more likely to want to complete a task).⁴³
 - Showing too many options in a choice screen may cause choice overload. The CMA has recognised that introducing an active choice remedy "*between many items*" could "*increase the burden on consumers and might cause them to disengage.*"⁴⁴
 - Showing too much information on a choice screen may overwhelm users and cause them to disengage with their choices. This is especially true in circumstances where word of mouth, searching, and browsing app stores and the internet are users' typical means of seeking out and installing new apps.⁴⁵

⁴² WP5, ¶4.45 and 2.45.

⁴³ See, for example, Annex S174(8)Q13.4 and Annex S174(8)Q13.10.

⁴⁴ See CMA, [Online Choice Architecture - How digital design can harm competition and consumers](#) (April 2022), ¶4.33.

⁴⁵ See [How People Discover, Use, and Stay Engaged With Apps](#).

- In the case of Option C5, showing choice screens more than once per device may cause decision fatigue and users swatting away subsequent choices. This could reduce user engagement with browser choice, rather than enhance it. Option C5 should therefore only be contemplated on devices that do not show choice screens under Option C2.

As explained above, these considerations indicate that implementation and enforcement of any choice screen remedy may be better suited to the new digital regime.

- **Choice screen remedies must take account of implementation costs and complexities.** Choice screen remedies can be costly to design, test, and roll-out to relevant devices. For example, to comply with its choice screen obligation under the DMA, we:

- [✂].

In addition, rolling out choice screens to existing Android OEM devices (as Option C5 would require) is complex because Google does not control the updates to these devices and relies on OEMs to ship the relevant updates. WP7 acknowledges that this may “*impact this option’s effectiveness.*”⁴⁶ Regulatory alignment (*i.e.*, with the DMA) can reduce these costs, taking advantage of work already undertaken, so should be a relevant—and important—consideration in any potential remedy design, testing, and implementation, should the CMA disagree with us that there is no AEC on Android.

- **Choice screen remedies may sacrifice important RCBs.** Requiring choice screens could undermine the value of default opportunities that browsers pay for (or, in the case of OEMs, use to promote their own browsers) (see further **Section III.B** below). In addition, it could also affect the user experience by requiring additional effort from users to make an active browser choice when most users prefer their out-of-the-box default.

67. **Option C3 (placement of default in the “hotseat”⁴⁷ or on default home screen) is unnecessary on Android, would be disproportionately costly to implement, and would risk a loss of RCBs.** Option C3 would require that Apple on iOS and Google on Android always place the browser selected by the user in any choice screen shown at device set up in the “hotseat”. This remedy is not necessary on

⁴⁶ WP7, ¶7.23.

⁴⁷ The “hotseat” refers to the bottom row of apps on a device’s home screen. It is also referred to as the “Application Dock.”

Android, would not materially improve users' experience, would be costly to implement, and would sacrifice important RCBs. In more detail:

- **Option C3 would be of limited value to Android users, who already rearrange apps on their home screens.** Option C3 would not materially increase user choice on Android or be well-targeted to the potential AEC identified in WP5. It is easy for Android users to change their default browser (see further below) and to move this to the hotseat if they prefer. According to the CMA's research, of those users who reported downloading their main browser, 8 in 10 recalled actively repositioning its placement.⁴⁸ This demonstrates that rearranging a newly downloaded browser's placement on a phone does not constitute a barrier to switching on Android. Option C3 is therefore not necessary if the CMA decides to implement a choice screen remedy, because users routinely rearrange apps on their home screens.
 - **Option C3 would be costly to implement.** Android OEMs currently decide the logic that applies to the placement of apps when they are downloaded onto a device. There is not a dedicated "browser" slot on the home screen and no means, as a technical matter, of ensuring that the browser initially placed in the hotseat (if one is placed there by the OEM at all) is "swapped out" for one selected from a choice screen. This would likely necessitate changes that Google would have to put into a new Android release, which would not be available for at least one year. There would also be significant costs for OEMs, which would all have to update their "launcher" software individually.
 - **Option C3 would risk a loss of important RCBs.** On Pixel devices (and some Android OEM devices), the hotseat out-of-the-box is populated automatically by apps that Android predicts the user is likely to want to use (e.g., if they plug in headphones, the hotseat may suggest a music app). On devices that include this feature, Option C3 would require the system to ensure that a browser is placed in the hotseat and therefore prevent Android from showing apps that the user may be more likely to want to use. Option C3 may therefore result in a loss of RCBs.
68. **Option C4 (access points) is unnecessary on Android as browser default choice already propagates to all relevant access points.** Option C4 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 preinstalled browser."⁴⁹ However, on Android, when a user switches default

⁴⁸ Verian Quantitative Consumer Research, slide 48.

⁴⁹ WP7, ¶7.32.

browser, their choice propagates to all relevant default access points using the Android “intents” system. This includes apps’ in-app browsers when they use Custom Tabs (unless the app developer has decided to open web content in a specific browser).⁵⁰ Option C4 is therefore not necessary on Android if the CMA decides to impose a choice screen under Options C2 or C5.

69. **Option C6 (changes to default switching journey) is unnecessary on Android, may not achieve its intended purpose, and may make it more difficult for users to switch.** Option C6 would consist of a combination of: (i) specifying the maximum number of steps that must be taken to switch default browsers; and (ii) improve the visibility of the relevant settings.⁵¹ This remedy would not, however, improve Android users’ ability to switch defaults. It would also disrupt OEMs’ flexibility to customise their devices’ default switching journeys. This is for the following reasons:

- **Android users already find it easy to switch default browsers.** As explained in our WP5 response, users on Android can already easily switch default browsers, and the CMA’s research confirms that users are confident in doing so.⁵² Not only does this indicate that there is no AEC on Android, it also means that the situation cannot be meaningfully improved through a remedy.
- **Specifying the maximum number of steps would not achieve Option C6’s intended purpose.** As explained in our response to WP5, the number of steps the user has to take is an arbitrary metric and not necessarily determinative of users’ ability or tendency to carry out this action in practice. For example, hiding a default switching option under an obscure, poorly worded, or less visible menu may reduce the number of steps involved in switching, but could drastically increase the actual cost of switching because users will not be able to find it.⁵³

Setting the maximum number of steps a user has to take could therefore result in the default browser setting being housed in an illogical and confusing place in the menu. In any event, Android provides an alternative

⁵⁰ See **Section II** above for Google’s comments on whether Custom Tabs on Android should be required always to use the user’s default browser.

⁵¹ WP7, ¶¶7.36-7.37.

⁵² WP5, ¶4.45; Verian Quantitative Consumer Research, slide 62.

⁵³ Settings menus must balance the number of settings choices a system can offer and the distribution of those choices across the menu. They contain many different choices and so they are often categorised into a tree of menus and submenus. The alternative (showing all settings choices in a single list) would be confusing and ineffective.

(easier) means of switching default browsers by letting browsers prompt users to switch defaults without leaving the browser app.⁵⁴

- **Changing the default switching journey on Android may make it harder for users to switch.** The user journey for switching default browsers on Android is intuitive, well-signposted, and housed with other similar default settings. Introducing a new user journey for default browser switching may therefore cause user confusion—and make default switching harder—by interfering with established design patterns and users’ existing mental models.⁵⁵ This risk is especially pronounced because, according to the CMA’s research, 8 in 10 users reported already being confident in their ability to switch defaults.⁵⁶

70. **Option C7 (sharing data on default browsers) is unnecessary on Android because it reflects the status quo.** Remedy C7 proposes “allow[ing] browser vendors to query whether or not their downloaded browser is set as the default browser on the device”.⁵⁷ On devices running Android 10 or later, a browser vendor can determine whether or not their app is the default browser on a device by calling an API.⁵⁸ If the browser is not the current default, the browser can prompt the user to make it the default. Accordingly, browser vendors already have visibility over whether a user has set their browser as a default on Android. This remedy is therefore not necessary on Android and would not address in a targeted manner a potential AEC the CMA finds on Android.⁵⁹

71. **Option C8 (limits on default prompting) is unnecessary on Android, would not achieve its intended purpose, risks depriving users of a useful means of switching default browsers, and gives rise to monitoring and enforcement risks vis-à-vis browser developers.** Option C8 would “regulate the volume, frequency and design of the prompts that browser vendors use to compete for default browser status.” For the following reasons, however, this remedy would be ineffective on Android, risk depriving users of a helpful route to switching default browsers, and be difficult to monitor and enforce on browser vendors:

⁵⁴ As WP5 notes (at ¶¶3.64-3.65), there is no equivalent API available on iOS to enable users to switch defaults in a single tap.


⁵⁵ See generally Ben Shneiderman and Catherine Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (4th ed., 2010) (discussing the importance of consistency in UI design and the impact of breaking established patterns).

⁵⁶ Verian Quantitative Consumer Research, slide 62.

⁵⁷ WP7, ¶7.45.

⁵⁸ `RoleManager.isRoleHeld(BROWSER)`.

⁵⁹ By contrast, on iOS there is no API that browsers can use to determine whether they are set as the user’s current default.

- Google already designs its default browser prompts carefully to limit their frequency and intrusiveness.** We agree that seeing a high volume of prompts and notifications can irritate users and lead to them becoming habituated to the point that they no longer engage, particularly if prompts appear at a time when the user is focused on a separate task (e.g., visiting a website and consuming content). Accordingly, as explained in response to WP5 ([]⁶⁰), we already apply limits to the frequency and intrusiveness of Chrome’s default prompts on Android and iOS, so they are proportionate and targeted and do not frustrate users’ choices. This remedy would therefore not be well-targeted or effective to resolve a potential AEC on Android.
- There is no evidence that Option C8 would “level the playing field for other providers.”** WP5 did not find (or allege) that Google uses default prompts in a more frequent or intrusive manner than other browser vendors. On Android, Chrome uses the same default prompt API that is open to other browser vendors.⁶¹ Placing limits on vendors’ use of prompts would therefore not “level the playing field”,⁶² as this is already the case on Android.
- Option C8 risks depriving users of a helpful route to switching browsers.** The CMA’s research confirms that users find default prompts useful as a means of switching default browsers.⁶³ As WP7 recognises, “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.”⁶⁴ Capping browsers’ ability to use these tools—which they currently use to improve users’ experiences—risks undermining their ability to compete in this respect, and risks sacrificing the RCBs that users benefit from. In other words, Google’s design of browser prompts already fosters the CMA’s goal of increasing user awareness of and engagement with browsers.
- Option C8 would need to be enforced directly on browser vendors, which raises enforcement and monitoring risks.** Google currently does not know how often a third-party browser invokes the default switching prompt on Android. They can, in theory, prompt users as many times as they want (and risk the user frustration that would result). It is not clear how limits on the use of these prompts could be monitored or enforced in these circumstances.

⁶⁰ [].

⁶¹ See further WP5 response, ¶54.

⁶² WP7, ¶7.48.

⁶³ WP5, ¶4.70.

⁶⁴ WP7, ¶7.48.

72. **Option C9 (uninstallation) is unnecessary on Android and risks resulting in a loss of RCBs.** Option C9 would require Apple and Google to allow users to uninstall Safari on iOS and Chrome on Android. For the following reasons, however, this remedy would be ineffective to address any potential AEC on Android because Chrome is already uninstalleable. Chrome is not fully *deletable* from an Android device as this is necessary for the integrity of the device and changing this would sacrifice important RCBs. In particular:

- **Option C9 is unnecessary on Android because Chrome is already effectively uninstalleable.** As explained in Google’s response to WP5, disabling stops Chrome from running in the background, collecting any data, or updating automatically.⁶⁵ In addition, Chrome can no longer be used once disabled “such as when clicking a web link in a third-party app”.⁶⁶ Chrome is therefore already effectively uninstalleable by Android users and Option C9 is therefore unnecessary.
- **A remedy that would require Chrome to be fully deletable from Android devices would result in a loss of important RCBs.** If the CMA disagrees that disabling Chrome constitutes uninstallation, and implements a remedy that would require Google to make Chrome fully deletable, Option C9 would be inappropriate to apply to Android because it would result in a loss of important RCBs. As explained in Google’s response to WP5, preventing full deletion of system partition apps such as Chrome serves two important purposes that safeguard the integrity of Android devices:⁶⁷
 - First, it ensures that a device can always be returned to a known, functioning state through an action known as a “factory reset” (e.g., for reselling the phone, returning it to a safe state, or protecting it from malicious apps or attacks).

Preventing deletion of software in the system partition enables users to perform such a reset while retaining a functional phone. If the reset were to remove all software, including software in the system partition, the phone would no longer be usable. The device would have become what is known as “bricked”. Disabling maintains the device’s out-of-the-box state which, for example, makes the device re-sellable on the secondary market.

- Second, the system partition is hermetically sealed from the rest of the device as an anti-tampering mechanism for the device. If it were possible for users to modify the system partition by deleting apps

⁶⁵ See Google’s Response to WP5, ¶¶45-48.

⁶⁶ WP7, ¶7.53.

⁶⁷ See Google’s Response to WP5, ¶48.

from it, users (or apps downloaded by users) could modify the system partition in other ways, such as granting untrustworthy apps dangerous permissions. Such an ability would greatly reduce the overall security of the device.

B. Google's Android Agreements Create Relevant Customer Benefits

73. As explained in response to WP5, if the CMA is minded to find that Google's Android agreements result in an AEC, its remedies analysis must take account of the RCBs that they create and which would be lost by any remedy the CMA imposes. The Android agreements create RCBs—for both OEMs and consumers—in the following ways:

- Lower device prices and/or higher quality devices;
- High-quality out-of-the-box experiences where users' preferred apps are preinstalled and set as default; and
- Safeguarding Android devices' resale value and device integrity, insofar as they require Chrome to be preinstalled on devices' "system" rather than "user" partition.

74. The CMA's remedies would likely result in the loss of these RCBs. In more detail:

75. **The Android agreements create RCBs in the form of lower device prices and/or higher quality devices.** Google's Android agreements provide an important revenue stream to Android OEMs that choose to enter into them. In the UK, the Android agreements on average are worth [X] per device.⁶⁸ This represents [X] of the average Android device price.⁶⁹ As explained in Google's response to WP5,⁷⁰ this creates benefits for two types of customer:

- First, OEMs, which can use the additional money (in aggregate over [X] per year) to invest in device quality and innovation.
- Second, end consumers, who benefit from this flow of money because it is passed on in the form of lower device prices and/or higher quality devices.

76. WP7's proposed remedies (specifically Options C1 to C3 and C5) risk upsetting this flow of money to OEMs [X]. They would therefore result in the loss of RCBs. We encourage the CMA to gather evidence directly from OEMs to confirm this understanding.

⁶⁸ [X].

⁶⁹ [X].

⁷⁰ Response to WP5, ¶61.

77. **The Android agreements create RCBs in the form of high-quality out-of-the-box experiences where users' preferred apps are preinstalled and set as default.** Chrome is preinstalled on Android devices alongside a range of other high-quality apps, including Google's popular apps like Maps, Google Search, YouTube, and Gmail. As explained in Google's response to WP5,⁷¹ this benefits OEMs (by facilitating device sales) and users (by saving them time and effort to reach their preferred apps). Option C1 (the proposal to require preinstallation of more than the usual 1-2 browsers usually preinstalled on Android devices) risks overwhelming users with preinstalled apps and reducing their out-of-the-box experience. This would result in the loss of these RCBs.
78. **Android's disabling of Chrome creates RCBs.** As explained above, disabling acts as an important safeguard to Android devices and therefore constitutes a RCB. If the CMA were to implement Option C9 that would require Chrome to be fully deleted from a device as opposed to disabled when a user uninstalled it, these RCBs would likely be lost.

C. Response to the CMA's Invitation to Comment

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

79. As explained above and in response to WP5, a remedy is not necessary to enable effective browser choice on Android. If the CMA disagrees and imposes a remedy, Google agrees with the principles outlined at WP7, ¶7.6 that any eventual choice interventions should be targeted, understandable, and balanced.

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

80. We have explained above and in response to WP5 that Android facilitates effective user choice of browsers. No remedy is therefore required on Android that would be effective or proportionate to the risks of unintended consequences it would create. If the CMA disagrees and decides there is an AEC that needs remedying, the most proportionate remedy could involve aligning with Google's DMA compliance measures (see further Google's response to (f) below).

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

81. As explained above and in response to WP5, a remedy is not necessary to enable effective browser choice on Android. Even if the CMA finds an AEC, it is difficult to opine on which remedy would be "most effective" because the effectiveness of a

⁷¹ Response to WP5, ¶61.

user's choice depends on various factors like the precise modalities of the choice design, where it is shown in the user journey, and how often it is shown.

82. However, the success and effectiveness of a choice architecture remedy should not be judged solely by their outcome (e.g., changes in market share). The CMA has previously acknowledged, for example, that the effectiveness of choice screens can “*be difficult to measure since its purpose is not to shift users away towards*” a particular service, but to “*ensure that users are free to exercise choice in an informed manner.*”⁷² We agree.
83. The concerns set out in WP5 relate to alleged distortions in users' choices resulting from certain choice architectures. An effective remedy to these concerns, if made out, would be to introduce undistorted, informed, and neutral user choice. Judging the success of a remedy by the outcome of users' engagement with such a remedy would itself distort market outcomes.

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

84. See Google's response to (c) above.

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

85. See Google's response to (c) above. Some remedies under consideration risk unintended consequences that could decrease user awareness and engagement (e.g., certain remedies risking choice overload and decision fatigue, and limits on browsers' use of prompts to encourage default switching).

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

86. Choice architecture remedies can be costly to design, test, and roll-out to relevant devices (see above at ¶66 for work undertaken to implement Google's DMA choice screens). Regulatory alignment and cohesion can reduce these costs, taking advantage of work already undertaken, so should be a relevant—and important—consideration in any potential remedy design, testing, and implementation. See further **Section V.A** below.

⁷² CMA, Online advertising and digital advertising market study final report, Appendix V (1 July 2020), ¶60.

IV. Cloud Gaming Services

A. Google's Comments on Options D1-D3

87. WP7 proposes three remedy options regarding cloud gaming services: Options D1, D2, and D3:
- **Option D1** would require Apple to review and amend its Guidelines to remove the specific restriction affecting cloud gaming apps, and prohibit Apple from introducing new restrictions with equivalent effect. Google no longer offers a cloud gaming service on iOS and is therefore not in a position to provide comments on this proposal.
 - **Option D2** would require 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 is payable to Apple).
 - **Option D3** would require Apple and Google to allow cloud gaming services providers to incorporate their own or third-party in-app payment systems for in-game transactions.
88. As explained in our response to WP6, there is no AEC in relation to cloud gaming services on Android, as shown by the fact that there are several successful cloud gaming apps on Google Play, including those offering a ‘consumption-only’ model. Accordingly, there is no need for a remedy as regards Android or Google Play. Option D3 is therefore unnecessary and disproportionate, for the reasons we set out below.
89. **Option D2 would in any event be sufficient.** Offering a “read-only”/“consumption only” model is a viable and attractive option for cloud gaming apps. There are several, highly-successful cloud gaming apps available on Google Play that operate a ‘consumption-only’ model. For example, Microsoft’s Xbox Game Pass operates on this basis and has been downloaded more than 10 million times on Google Play.⁷³ There are also several other examples of streaming apps—such as Netflix—that use this model. Allowing cloud gaming apps to operate on the same basis on iOS, as proposed by Option D2, would therefore be sufficient to address any potential AEC.
90. **Option D3 is unnecessary and disproportionate.** As any potential AEC would be addressed by Option D2, there is no need for Option D3. Although Microsoft makes certain allegations about perceived “disadvantages” of offering a “consumption-only” model on Google Play, these are meritless, for the reasons we set out in our response to WP6. There is a well-functioning market for cloud gaming apps on Android, and accordingly no need for the CMA to implement Option D3.

⁷³ See Google Play, [Xbox Game Pass](#).

91. The CMA has closed its investigation under the Competition Act 1998 into Google Play's billing system in light of the forthcoming availability of new powers to the CMA under the DMCC Act, noting that these new powers may enable the issues to be afforded a "more holistic consideration" which "is likely to be most effective in achieving meaningful [...] impact where appropriate".⁷⁴ The same is true for the CMA's consideration of Google Play's rules for cloud gaming services on Android. An additional remedy for Android via Option D3 is therefore unnecessary and would be disproportionate.

B. Response to the CMA's Invitation to Comment

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

92. See above.

V. Cross-Cutting Considerations

93. WP7 outlines several factors applicable across the potential remedies the CMA is considering. We address each of them below.

A. Measures Taken in Other Jurisdictions

94. WP7 states that it is considering Apple's and Google's measures announced to comply with the DMA where relevant to the design or implementation of its potential remedies.⁷⁵
95. None of the issues under investigation give rise to an AEC on Android. If the CMA disagrees with this position, however, we agree that consideration of DMA compliance measures is relevant to the CMA's remedies assessment, given the overlap between certain DMA articles (summarised in WP7⁷⁶) and the issues and potential remedies the CMA is considering.
96. For example, we do not agree with the CMA's emerging thinking that a choice screen may be a necessary means of facilitating user choice on Android, as Android already provides users with effective choice. If, however, the CMA disagrees and decides to require a remedy, it is relevant to the CMA's proportionality assessment that we have launched DMA-compliant browser choice screens on Android in the EU under Article 6(3) DMA, following extensive research into different options and

⁷⁴ Case 51183 - Google - Google Play Billing, Statement regarding the CMA's decision to close an investigation on grounds of administrative priority.

⁷⁵ WP7, ¶3.6.

⁷⁶ WP7, ¶3.5.

engagement with the European Commission and choice screen participants.⁷⁷ Regulatory alignment may also have benefits for monitoring and resolving compliance issues across jurisdictions due to having consistent conditions, as WP7 recognises.

B. Geographic Scope of Potential Remedies

97. WP7 welcomes views on the impact that the geographic scope of remedies being considered “*could have on the effectiveness and proportionality of potential remedies.*”⁷⁸ The CMA’s focus should be on protecting customers and businesses from any harms resulting from any AEC(s) identified in the UK. As a general principle, when considering the geographic scope of potential remedies, we therefore agree that it is important to scope any remedy in such a way that the risk of circumvention is mitigated (while at the same time ensuring that the scope of the remedy goes no further than is necessary to address any potential AEC).
98. How this should be achieved will depend on the specific AEC and associated remedy concerned. We would be happy to engage in more detail on this question if the CMA reaches an AEC finding.

C. Links Between Different Remedy Options as Part of a Wider Mobile Ecosystem

99. WP7 states that “*potential issues being considered in this market investigation relate to mobile operating systems, browsers and app stores,*” and that remedy considerations should take account of how these aspects of mobile ecosystems are interrelated.⁷⁹ These considerations seem relevant to potential remedies that are not relevant to Android, including those discussed in WP2 (WebKit) and WP6 (Cloud Gaming).

D. Risks Relating to The Level of Specification of Certain Proposed Remedies

100. WP7 acknowledges that higher-level requirements to “*provide access to technical functionality may provide more flexibility but could also be difficult to monitor and enforce if there are information symmetries.*”⁸⁰ We address this trade-off in our comments on WP7’s potential remedies on WebKit and access to functionality in **Section I** above.


⁷⁷ [].

⁷⁸ WP7, ¶3.10.

⁷⁹ WP7, ¶3.16.

⁸⁰ WP7, ¶3.18.

E. The Digital Markets, Competition, and Consumers Act 2024

101. WP7 states that the CMA is “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 potential remedies”.⁸¹
102. We agree that certain remedies (if implemented) may be better suited to design, implementation, and enforcement by the DMU, such as choice architecture remedies that may involve longer-term oversight, iteration, and testing. In particular, we note the following:
- First, the digital markets regulatory regime is designed to be applied flexibly. Ongoing monitoring of digital markets is a central pillar of the regime, with over 20 pages of the CMA’s Draft Guidance devoted to it.⁸² The market investigations regime is, by contrast, designed around one-off interventions that do not require extensive ongoing monitoring.⁸³
 - Second, the CMA’s Draft Guidance describes in detail how it plans to identify potential concerns to address under conduct requirements, test potential solutions, monitor compliance, and make adjustments as required.⁸⁴ The potential design, testing, implementation and monitoring of choice architecture remedies is therefore arguably better suited to the new regime, where these structures can be applied with certainty for firms and businesses.
103. .

⁸¹ WP7, ¶3.30.

⁸² CMA, Digital markets competition regime guidance CMA194con DRAFT (24 May 2024), Section 6 (**Draft Guidance**).

⁸³ See Martin Coleman speech at the Swedish Competition Authority’s Pros and Cons Conference, [Market investigations: 75 years of UK experience](#) (20 May 2024) (“[T]he challenges presented by digital markets cannot be solely addressed by an investigatory system designed around one-off interventions even where [...] one can build in longer term oversight of remedies. Market investigations are also not always optimal in fast moving markets where more rapid interventions may be required.”).

⁸⁴ This can include information notices requiring a person to vary their usual conduct or perform a specified demonstration or test, which the CMA’s Draft Guidance (at ¶5.13) states “may be necessary to assess the effect of different choice architecture and assess compliance with particular [conduct requirements].” The Draft Guidance (at ¶5.14) provides a framework for when this information-gathering tool will be used, including the value, feasibility, and proportionality of doing so. For pro-competitive orders, the Draft Guidance (at ¶4.67) anticipates that testing and trialling may “provide valuable additional evidence [...] for interventions which require the SMS firm or third parties to develop new features or functionalities or remedies involving design choices which require behavioural insight.”

104. [redacted].

F. Testing and Trialling Certain User Choice-Based Remedies

105. WP7 considers 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.*”⁸⁵ As above, we agree that research, testing, and engagement with the CMA and third parties may have a role to play in any potential choice architecture measures. It is for this reason that these potential remedies may be better suited to design, monitoring, amendment, and enforcement under the new UK digital regime.

G. Effectiveness and Proportionality of a “Package” of Remedies

106. WP7 states that remedies in market investigations are likely to take the form of a package of measures, rather than the implementation of a single measure. Google reserves its position on the effectiveness and proportionality of any proposed package of measures until it has seen the CMA’s decision on any AECs, and the remedies package it proposes to address them, following publication of the Provisional Decision Report.

H. Inappropriateness of Structural Remedies

107. WP7 reports that the CMA has “*not identified structural remedies that we consider likely to effectively address any potential concerns.*”⁸⁶ We agree. As a general principle, we do not consider structural or operational separate remedies to be a necessary or proportionate means of resolving competition issues. Structural remedies in this investigation would be disproportionately intrusive and costly, sacrifice significant relevant customer benefits, and would not necessarily be an effective means of resolving the CMA’s emerging concerns.

Conclusion

108. We acknowledge the CMA’s extensive and detailed investigation into mobile browsing and cloud gaming. [redacted]. On Android, there is no AEC for the reasons given in Google’s responses to Working Papers 1-6. Accordingly, no remedy is required on Android.

109. Even if the CMA disagrees with this assessment, though, it should carefully consider whether any remedies would improve outcomes for UK consumers. This response has shown that the CMA’s potential remedies that would apply to Android and Chrome:

⁸⁵ WP7, ¶3.22.

⁸⁶ WP7, ¶4.2.

- Would not necessarily enhance user awareness of or engagement with browser choice, which is already ensured on Android.
- Would risk unintended consequences in the form of choice overload, decision fatigue, and other disbenefits for UK consumers.
- Would impose disproportionate costs on Google and other ecosystem participants such as app developers and OEMs.
- Would risk a loss of important RCBs.

110. In light of these considerations, no remedy should be imposed on Android or Chrome. Google looks forward to engaging with the CMA further on these issues in the remainder of its investigation.

* * *