Open
Web Advocacy

OWA - Mobile Browsers and Cloud Gaming - Response to Provisional Decision Report

VERSION 1.0

Open Web Advocacy
contactus@open-web-advocacy.org

1. Table of Contents

2. Introduction

We want to express our heartfelt gratitude to both the CMA and the MIR team for their dedicated efforts. We support the vast majority of the proposed remedies, as we believe they are essential for addressing the issues plaguing the mobile browser and mobile application markets, which have been severely impacted by anti-competitive practices for over 15 years.

These practices were accurately highlighted by the CMA in their December 2021 mobile ecosystem report. The study revealed that Apple and Google maintain an effective duopoly over mobile ecosystems, enabling them to dominate key areas such as operating systems, app stores, and web browsers on mobile devices.

The MIR was then launched to address this:

> *"This follows a year-long study of the companies' mobile ecosystems, the final report of which has been published today.* ***The study found that Apple and Google have an effective duopoly on mobile ecosystems*** *that allows them to exercise a stranglehold over these markets, which include operating systems, app stores and web browsers on mobile devices.*
>
> ***Without interventions, both companies are likely to maintain, and even strengthen, their grip over the sector, further restricting competition and limiting incentives for innovators.***
>
> *While the report identified a range of potential interventions across these ecosystems, the CMA has looked at where it can take* ***immediate targeted action*** *to* ***tackle these problems using its current powers****."*
>
> CMA plans market investigation into mobile browsers and cloud gaming
>
> (emphasis added)

Browsers and Web Apps were identified as a critical aspect requiring intervention:

> *"We all rely on browsers to use the internet on our phones, and the engines that make them work have a huge bearing on what we can see and do. Right now, choice in this space is severely limited and that has real impacts –* ***preventing innovation and reducing competition from web apps****. We need to give innovative tech firms, many of which are ambitious start-ups, a fair chance to compete."*
>
> Andrea Coscelli - Chief Executive of the UK's Competition and Markets Authority
>
> (emphasis added)

Web Apps and the browsers that power them, are a key component to address the mobile app duopoly controlled by Apple and Google. When Web Apps, which rely on interoperable, open-source, free, and untaxed technologies, are allowed to compete fairly with native app ecosystems, these ecosystems would face significant competitive pressure from both consumers and developers. By eliminating the anti-competitive practices that hinder Web Apps and the browsers that support them, the MIR team can unlock significant new opportunities for UK businesses and higher quality, cheaper and more interoperable software for consumers. This will also greatly reduce the lock-in of these mobile ecosystems as a significant percentage of software would now be interoperable.

In working paper 7, the MIR outlined the following remedies:

- A1 - Requirement for Apple to grant access to alternative browser engines to iOS.
- A2 - Requirement for Apple to grant equivalent access to iOS to browsers using alternative browser engines.
- A3 - Requirement for Apple to grant equivalent access to APIs used by WebKit and Safari to browsers using alternative browser engines.
- A4 - Requirement for Google to grant equivalent access to APIs used by Chrome.
- B1 - A requirement for Apple to enable remote tab IABs for WebKit-based browsers.
- B2 - A requirement for Apple to enable remote tab IABs for browsers wishing to use alternative browser engines.
- B3 - A requirement for Apple to allow alternative webviews to Apple's iOS WKWebView.
- B4 - A requirement for Apple and Google to implement remote tab IABs using the default browser.
- 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.
- B6 - A requirement for Apple and Google to implement opt-out settings for in-app browsing.

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

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

- 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.

- 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.

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

- C6 - A requirement for Apple and Google to make adaptations to the user journey for changing their default browser.

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

- 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.

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

We strongly support the majority of the proposed remedies and detailed our perspectives extensively in our remedies response paper. Overall, we found these remedies to be necessary, practical, and proportionate.

However, we were previously, particularly concerned that the MIR's initial emphasis on Web Apps, specifically how they had been undermined by the lack of effective browser competition, and the appropriate remedies to address this issue, had been inadvertently left out. At the time, we published our reasoning in detail to highlight this concern.

In particular, we advocated for the MIR team to introduce a new remedy:

> *"Mobile OSes shall allow third-party browsers to install and manage Web Apps using their own browser engine."*

We are more than delighted that the MIR team has taken our feedback to heart, along with that of numerous developers and businesses who shared similar concerns. Notably, we believe that the inclusion of the following provision, when enforced alongside other proposed remedies, would represent a critical step toward restoring competition in the provision of browser functionality for mobile Web Apps. This change would enable Web Apps to achieve feature parity with native apps, creating significant competitive pressure on the mobile app store duopoly maintained by Apple and Google.

> *"For example, 'equivalence of access' would need to include* ***enabling third-party browsers using alternative browser engines to install and manage PWAs*** *(rather than relying on WebKit to support parts of this process), including enabling mobile browsers using alternative browser engines to implement installation prompts for PWAs."*
>
> MIR - Provisional Decision Report
> (emphasis added)

2.1. Enforcement

We applaud the MIR team for their excellent work and analysis. Even though we have disagreements on some of the many issues that they have covered, on the whole they have succinctly and accurately covered an immensely complex set of issues and created appropriate remedies in the vast majority of cases.

A significant concern we do have, however, is that in the provisional report, the MIR has essentially deferred enforcement to the CMA / DMU under the Digital Markets, Competition and Consumers Act 2024 (DMCC). With the DMCC powers set to take effect in January 2025, firms like Apple and Google will still then need to be designated under its provisions. Even if the CMA acts with maximum efficiency, our understanding is that it will likely take until mid to late 2025 to formally designate these companies. At that point, further investigations and enforcement measures, despite leveraging the MIR's research and findings, are likely to be lengthy and resource-intensive. This timeline risks delaying meaningful remedies to these anti-competitive issues by at least an additional two more years.

We recognize that regulatory processes are inherently slow and acknowledge that the DMCC and the DMU (Digital Markets Unit) was created specifically to address these challenges with tech giants. However, the stark reality is that by the time the DMCC can compel Apple and Google to address these issues, browser and web app competition will have been undermined for over 17 years. This will also mark five years since we first raised these concerns with the CMA and since they were first identified in the CMA's mobile ecosystems study.

We urge the MIR team to reconsider its approach and immediately implement at least a minimum of a core set of the most critical remedies (such as removal of the WebKit restriction). Once the DMCC is in effect, the DMU can take over the responsibility for ongoing enforcement, addressing any remedies that have been bypassed or whose objectives remain unfulfilled. Fully deferring the task of identifying and enforcing remedies to the DMCC will allow Apple and Google to continue the anti-competitive practices that the MIR's own findings have clearly identified as harmful to competition and the broader market. Moreover, from the CMA's perspective, accelerating enforcement by two years represents a strong return on the investment made into this market investigation reference, ensuring that the identified harms are addressed sooner, enabling a more competitive and innovative digital ecosystem.

For any remedies or anti-competitive behaviors the MIR team decides to delegate entirely to the CMA under the DMCC, it is essential that the CMA/DMU is prepared to act swiftly and decisively once the DMCC is operational. If the MIR team does in its final decision

decide to push this to the DMU, delaying enforcement by up-to another two years, we believe they should provide comprehensive explanation of why implementing remedies now, with DMU oversight and further enforcement to follow, would not be a viable approach. We felt that the rationale for deferring every intervention to the DMCC rather than utilizing existing powers to address some of the anti-competitive conduct was not adequately explained in the report.

2.2. In-App Browsers, WebAPKs & Install Prompts

OWA identified three main areas of concern in relation to the provisional decision report, which we strongly urge the MIR team to deeply review and consider:

1. **In-App Browsers**: The lack of respect for users' default browser choices and the silent overriding of preferences by In-App browsers, such as Apple's SFSafariViewController, undermines competition and user autonomy.

2. **WebAPKs**: Google's restriction of WebAPK minting to Chrome damages both browser and web app competition, as third-party browsers are deprived of key functionality needed for Web Apps to compete effectively with native apps.

3. **Install Prompts**: The absence of clear and user-friendly install prompts for Web Apps in Safari limits the competitive potential of Web Apps against native apps and perpetuates both Apple and Google's dominance in mobile apps.

2.3. Unlocking the Potential of a Competitive Mobile Ecosystem

We sincerely thank the MIR team for their extensive efforts in producing this groundbreaking report, which includes numerous first of its kind insights into browsers and web apps, alongside significant recommendations which will lead to a more competitive mobile ecosystem

The web, which has flourished on desktop, has been prevented from realizing its full potential on mobile devices due to the actions of a small number of gatekeepers. The remedies proposed collectively have the power to change the course of mobile browsing and software development, not just in the UK, but globally. These changes promise greater competition, improved quality, lower costs, enhanced privacy, and more secure software for all consumers and businesses in the UK.

3. Review of Provisional Findings

3.1. Apple's Mobile Browser Policies Hurt Competition

"Apple's control over iOS gives it market power at the operating system level. In turn, this enables Apple to set the rules and parameters relevant to how mobile browsers are allowed to work on iOS.

We have heard widespread, detailed and compelling evidence that the rules Apple sets due to its control of the iOS operating system limit the ability of mobile browsers other than Apple's Safari to provide more innovative, differentiated features.

Fundamental to this is Apple's rule on iOS which bans the use of different underlying browser engines, which are crucial for determining browser performance, security, privacy, and new features.

We note that there is no such rule on Apple's desktop operating system macOS, where other browser engines are allowed, nor on other mobile platforms beyond iOS.

We have considered submissions from Apple that insisting browsers only use WebKit is necessary because allowing alternative browser engines could raise security, privacy and performance risks.

We accept that the current restriction does reduce the risk of third-party browsers on iOS using outdated, vulnerable engines or implementing insecure new features. However, our provisional view is that the risks could be managed in other ways, e.g. by Apple imposing minimum security standards on mobile browsers using browser engines other than WebKit. We also note that alternative browser engines have strong records on security outcomes, and more widely, that Apple's current restriction actually prevents mobile browsers competing and innovating on security and privacy features, for example by implementing security updates more frequently than Apple's architecture currently allows.

We have provisionally concluded that this limits the ability of mobile browsers competing with Safari on iOS to attract users by offering high-quality products, and as a result, reduces competition and the resulting benefits for consumers."

MIR - Provisional Decision Report

We fully support this finding, as it addresses the fundamental issue undermining browser competition.

We also agree that the two proposed remedies, namely, removing the "**WebKit Restriction**" and "**Allowing equivalent access to software and hardware APIs**" are both proportionate and effective solutions.

However, we did have some concerns regarding the precise implementation of these remedies, which we detailed in our response to Working Paper 7.

3.2. Apple-Google Revenue Sharing Reduces Competition

> *"We have provisionally found that competition between mobile browsers on iOS is likely further weakened by an agreement between Apple and Google, pursuant to which Google pays Apple a significant share of the search advertising revenue earned from traffic on Safari and Chrome on iOS.*
>
> *This means Apple and Google earn significant revenue when their key rival's mobile browser is used on iOS, reducing their financial incentives to compete. In fact, the extent of this revenue-sharing is so large that the revenue share they earn from their competitor's product is lower but similarly significant to the revenue share they earn from their own, so that the incremental revenue from winning a customer is limited. We have provisionally found that this is likely to reduce competition between the two main mobile browsers on iOS devices."*
>
> MIR - Provisional Decision Report

We wholeheartedly agree with this finding.

Google's revenue sharing agreements for default search engine placement on smaller browsers both funds them and provides them an incentive to gain greater market share.

However, Google's revenue sharing agreements for default search engine placement with Apple not only undermines competition in the search engine market but also undermines browser competition on iOS. These agreements, which are substantial in scale (estimated at USD $20 billion annually), ensure that Google shares revenue not only from searches conducted via Safari but also from searches made through Chrome on iOS. This removes the main incentive for Google to expand Chrome's market share on iOS.

Additionally, unlike other browser vendors, where a significant portion of such revenue is reinvested to improve the browser or its underlying engine, Apple retains the vast majority (we estimate well over 95%) of this revenue rather than reinvesting it into Safari. This practice further weakens the competitive dynamic in the browser market on iOS.

3.3. SFSafariViewController and Remote-tab In-App Browsers on iOS

"7.138 Our provisional views on the impact of Apple's ban on remote tab IABs on competition in the markets for mobile browsers, in-app browsing technology and browser engines on iOS are as follows:

(a) First, Apple's ban on remote tab IABs (together with its ban on bundled engine IABs) means that Apple does not face any competition in the supply of in-app browsing technology on iOS. We consider that remote tab IABs would be similar to SFSafariViewController in that they are low cost and easy to implement for the app developer. Remote tab IABs would, therefore, represent an avenue via which alternative in-app browsing technology providers such as browser vendors could exert competitive pressure on Apple's own in-app browsing offering.

(b) Second, the inability of browser vendors to offer remote tab IABs also harms their ability to compete in the market for mobile browsers on iOS as it prevents them accessing a sizeable and likely growing proportion of webtraffic. Indeed, offering remote tab IABs would allow browser vendors to benefit from getting additional traffic (including via improved web compatibility) and therefore grow, as well as to support their existing customers better by providing a more 'consistent' web browsing experience on the device. This would increase competitive pressure on browsers on iOS, including Safari.

(c) Third, the ban on remote tab IABs on iOS also reduces the ability of alternative browser engine providers to compete on iOS. Currently, WebKit is the only available browser engine for in-app browsing on iOS, and SFSafariViewController is based on WebKit. If alternative browser engines were permitted on iOS, additional traffic via remote tab IABs may contribute to increased web compatibility for them and therefore allow them to compete more effectively. Indeed, web compatibility affects the ability of alternative browser engines such as Gecko to compete (see Section 2: Nature of competition in the supply of mobile browsers, browser engines and in-app browsing).

(d) Fourth, while we acknowledge that app developers largely appear satisfied with their existing in-app browsing options on iOS, we note their general lack of concern may depend on Apple's outright ban on remote tab IABs, which may contribute to them not being fully aware of the potential benefits of using this in-app browsing technology. Their lack of concern might also relate to the fact that those who use SFSafariViewController are looking for a relatively low-cost, easy-to-implement solution, so they may be less engaged in this area in general. Enabling remote tab

> *IABs on iOS would give app developers greater choice around how they present web content within their apps. Further, we note that Custom Tabs on Android has achieved widespread use by many app developers, which suggests that app developers may also take up this option on iOS.*
>
> *[...]*
>
> *Finally, we further note that enabling remote tab IABs on iOS would also provide the option for app developers to call upon a user's default browser for in-app browsing if they wish. We place particular emphasis on the fact that in-app browsing technology is provided first and foremost to app developers to incorporate within their apps. It is therefore important app developers are given sufficient choice of in-app browsing implementations to best meet their requirements – eg with respect to factors such as cost, customisability and visibility over user activity (eg see sub-section titled 'How browsing works when accessed within an app' for more detail on use cases of IABs for app developers)."*
>
> MIR - Provisional Decision Report

The current implementation of SFSafariViewController functions as a stripped-down version of Safari. Apple's documentation explicitly positions **SFSafariViewController** as providing a Safari-like experience, utilizing the same underlying **WKWebView** framework and incorporating many of Safari's settings and features. Essentially, users are presented with a cut-down version of the same features and functionality they would experience in Safari.

Apple further emphasizes this in their own documentation, stating:

> *With it, people can enjoy **the same web browsing experience they get in Safari** — including features like Password Autofill, Reader, and Secure Browsing — without ever having to leave your app.*
>
> Apple
> (emphasis added)

This design stands in contrast to Android Custom Tabs, which is not tied to Chrome. On Android, if a user sets Firefox as their default browser, Android Custom Tabs will seamlessly invoke Firefox to handle web pages. SFSafariViewController, on the other hand, is exclusively tied to Safari, disregarding the user's selected default browser entirely.

This setup enables Apple to override users default browser, effectively funneling a significant portion of in-app browsing traffic (and significant proportion of total iOS web

traffic) through Safari. Even in regions where Safari can be uninstalled, such as the EU, SFSafariViewController remains the de-facto browser for most in-app browsing activities, bypassing any default browser the user may have selected.

In earlier discussions, we have argued that SFSafariViewController should be treated as a system component, particularly when assessing whether Safari should remain uninstallable. However, this perspective was predicated on Apple being required to update SFSafariViewController to function like Android Custom Tabs, which respects the user's default browser preference. As it stands, Apple effectively circumvents user choice and undermines browser competition by funneling in-app web traffic back to its own browser, which ties into the broader issue of web apps and the open web struggling to compete with Apple's native app ecosystem and the broader Apple/Google duopoly.

> *"the inability of browser vendors to offer remote tab IABs also harms their ability to compete in the market for mobile browsers on iOS as it prevents them accessing a sizeable and likely growing proportion of webtraffic. [...] This would increase competitive pressure on browsers on iOS, including Safari."*
>
> MIR - Provisional Decision Report

The MIR team, in its Provisional Decision Report, acknowledges that Apple's restrictions are harmful to browser competition on iOS, particularly noting that browser vendors will be denied significant and growing web traffic from native apps outbound links. This lack of access hampers their ability to compete and weakens competitive pressure on Safari. The MIR team is correct in identifying the problem, but their proposed solution is misguided. Instead of obligating Apple to upgrade SFSafariViewController to respect the user's browser choice, the team suggests app developers, rather than users, should determine how outbound links are rendered and which browser handles them.

This approach effectively strips users of their agency, as app developers would then be not only free to disregard the default browser setting and thus the user's choice of default browsers but also unable to honor it. The MIR team further proposes that third-party browser vendors could create their own remote tab IAB components for iOS and convince developers to integrate these into their apps. Yet, even the MIR team appears unconvinced of the effectiveness of this approach, and rightly so. Most app developers are not looking for complex or burdensome solutions; they need a simple and efficient mechanism for opening web links within their applications.

> *"Their lack of concern might also relate to the fact that those who use SFSafariViewController are looking for a relatively low-cost, easy-to-implement solution, so they may be less engaged in this area in general. "*
>
> MIR - Provisional Decision Report

The MIR team's position is perplexing, particularly given their thorough examination of browser choice architecture and the risks of gatekeeper suppression. By supporting a solution that allows app developers to bypass user choice and creates barriers to honoring it, the team undermines its own recommendations on improving competition and user engagement.

A choice is only meaningful if it cannot be easily ignored by the system or the applications that run on it.

> *"We have provisionally found the pre-installation and prominent placement of Safari and default settings on iOS devices reduce user awareness, engagement and choice, increases barriers to entry and expansion for other browser vendors and further reinforces Safari's very strong position on iOS."*
>
> MIR - Provisional Decision Report

It also appears the MIR team has been influenced by arguments from companies like Meta, which claim that the ability to implement in-app browsers on iOS would foster new competition and innovation. According to this view, enabling app developers to select in-app browsers would lead to fierce competition as browser vendors compete to be bundled into apps. While superficially plausible, this argument collapses under scrutiny.

There is nothing preventing Apple from providing a straightforward system-level solution that respects the user's default browser while still allowing companies like Meta to compete. Developers who wish to implement custom in-app browsers could do so but should have to obtain explicit permission from the user via an operating system prompt.

This ensures transparency and allows users to make an informed choice about which browser they wish to use. Such a system would incentivize app developers to offer compelling reasons for users to opt into their native app's custom browser while also maintaining respect for the default browser setting. This would then incentivize app developers who wish to produce their own in-app browser to both add compelling reasons to use it and display those reasons to end users.

Furthermore, the MIR team's proposal that developers bundle third-party remote tab components is entirely impractical. Few developers are likely to integrate such solutions, given the additional app size and the technical challenges involved. It is also unclear whether a reliable solution could be built, as third-party code would not have system-level access to detect the user's default browser or to invoke the required default browser. The alternative, requiring developers to bundle entire browsers into their apps, presents further challenges. Not only would this significantly increase app sizes, but it would also

default to ignoring the user's browser preference. Browser vendors, for their part, would have little incentive to produce standalone remote-tab browser components. It is also not clear how this would be remotely beneficial to users.

Additionally, when app developers ship an embedded browser, they inherently take on the responsibility of acting as a browser provider, which comes with significant challenges, including the need for frequent updates and addressing critical security risks to ensure a safe and reliable user experience. Given the browsing functionality is almost entirely secondary in all of these apps, the likelihood that all of these apps will be responsible with updates is low.

The most probable outcome is that the majority of developers will stick with using **SFSafariViewController**, which essentially functions as Safari. In short, this solution is infeasible and will not achieve any of the goals of allowing more user choice or greater competition between browsers, in-app or otherwise.

This misguided approach risks creating broader harm. By endorsing such logic, the MIR team may inadvertently encourage companies like Google to restrict Android Custom Tabs to Chrome alone. Google could then justify this behavior by using the MIR team's own arguments for SFSafariViewController, stating that if app developers on Android wished to provide a different in-app browser they could bundle it into their app. The result would be a further erosion of competition on Android and a removal of the user choice that drives browser competition, effectively blessing Google's ability to dominate in-app web traffic on Android.

The correct solution is clear and simple, Apple must be required to upgrade SFSafariViewController to respect the user's default browser choice. This change would bring SFSafariViewController in line with Android Custom Tabs and ensure all apps currently relying on SFSafariViewController automatically respect the user's browser preference. Such an upgrade is not only feasible but would immediately enhance competition by making user choice meaningful. Browsers would need to compete on merit to become the user's preferred browser, fostering genuine competition.

Companies like Meta that wish to implement their own in-app browsers would still have the opportunity to do so, but they would need to obtain the user's consent to override the default browser. This approach respects user choice, applies competitive pressure on developers to improve their offerings, and ensures transparency for users.

By upgrading SFSafariViewController to respect the user's default browser, Apple would provide app developers with a simple, system-level solution that requires no additional effort or complexity. Updating this component would also make all app developers that

currently use SFSafariViewController support the user's chosen default browser with no code changes or additional effort.

In short, Apple must be obligated to update SFSafariViewController to function like Android Custom Tabs. Apple should also be prohibited from offering a remote-tab browser solution that is locked to Safari or that does not respect the user's choice of default browser. Apple must not be allowed to lock in-app browsing to Safari, as doing so undermines user agency and distorts competition. Requiring this upgrade is the most effective and straightforward solution to ensure that user choice remains meaningful.

Among all the issues associated with in-app browsing, this one is the most critical and the most important remedy for the MIR team to implement.

3.3.1. Apple's misleading statements and self-preferencing

Summary:

- Apple misled the CMA by stating SFSafariViewController does not share state with Safari
- Apple's own apps such as Apple Maps are given special preference by Apple over other third party apps to share state
- Third Party Apps such as Google Maps are unable to share state with Safari

Apple has made the following statements to the CMA which are demonstrably false:

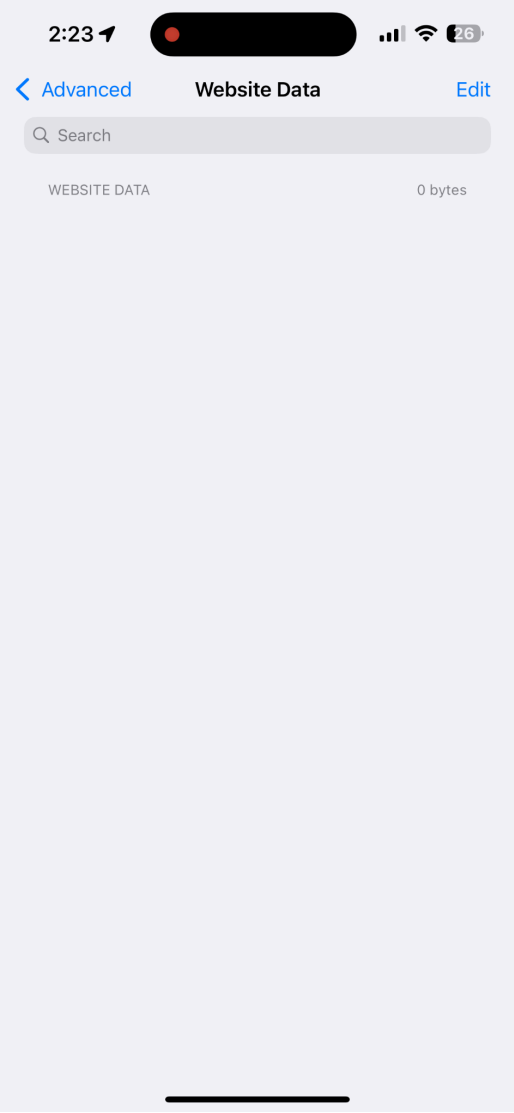
> "**SFSafariViewController does not share state with Safari**, a user's default browser, **or any native app** – according to evidence from Apple. Instead, SFSafariViewController uses the iOS webview technology, WKWebView, to build a view controller which is sandboxed, meaning **it is deliberately isolated and does not share resources with other apps or processes**."
>
> *MIR - Provisional Decision Report*
> *(Page 279, from a meeting with Apple)*

> *"In contrast, SFSafariViewController does not rely on an underlying browser. Apple submitted that SFSafariViewController* ***used to share state with Safari*** *to stop users having to re-authenticate credentials when leaving Safari,* ***but this was changed*** *[✂].*
>
> ***SFSafariViewController currently stores user data, cookies, and browsing activity in a private container****, which neither third parties* ***nor Safari*** *can See."*
>
> *MIR - Provisional Decision Report*
> *(Page 279, from Apple's response to the CMA's information request)*

Apple claims that SFSafariViewController operates in an isolated state and does not share data with Safari. However, by following the outlined steps, the CMA can independently verify that this is not the case and reveal evidence of bi-directional state sharing between Safari and SFSafariViewController within Apple's own apps.

3.3.1.1. Steps to Replicate

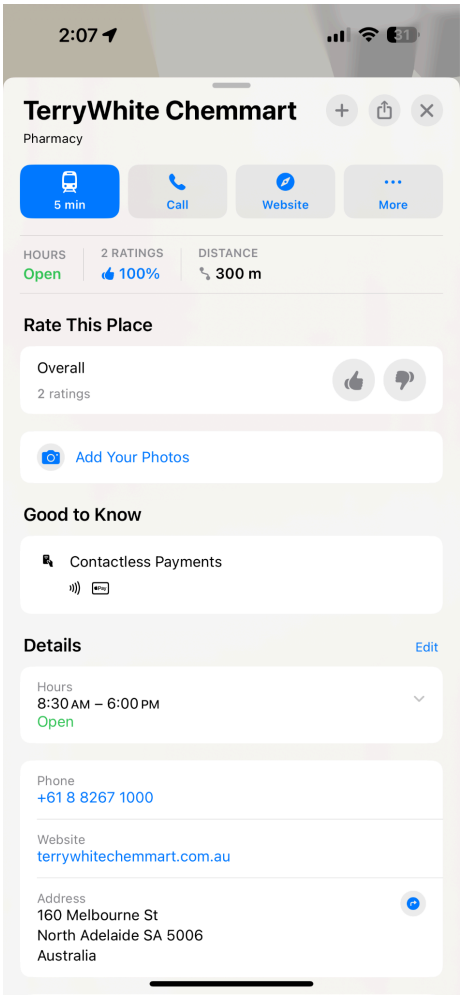
1. Open **Settings > Apps > Safari**
2. Tap the "**Clear History and Website Data**" button
3. Tap **All History**, and **Clear**
4. Navigate to **Settings > Apps > Safari > Advanced > Website Data**. Note that it is currently empty. Which means Safari is currently not storing any data for any website.



5. Without opening Safari, navigate to Apple Maps

6. Identify a website that is listed within a business Apple maps and which has login functionality or a shopping cart

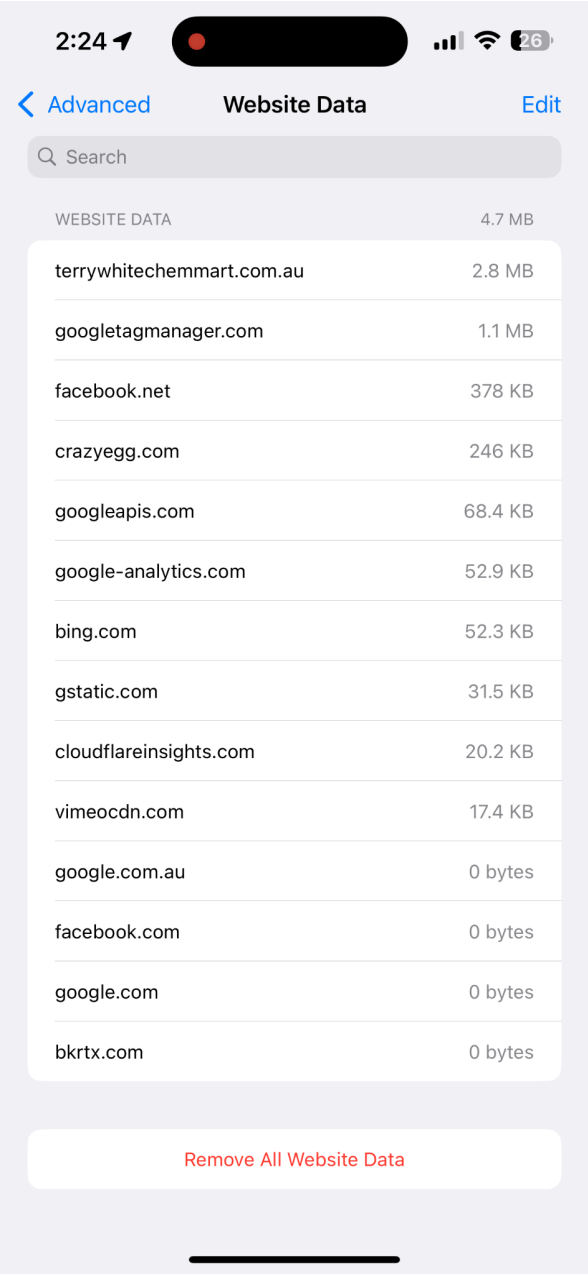
7. Find the business in Apple Maps and open the website under Details → Website



8. This will the open the website in SFSafariViewController inside Apple Maps

9. Navigate to **Settings > Apps > Safari > Advanced > Website Data**

   You'll notice that even without ever opening Safari, all the assets from the site we just visited are now listed. This directly challenges Apple's assertions about always isolating **SFSafariViewController** and further highlights the lack of a clear distinction between **SFSafariViewController** and **Safari** itself.



10. Next navigate back to **Apple Maps**, and **login** to the website within SFSafariViewController.

    You can note in our example we are logged in by the initials AM in the top right hand corner.

11. Now navigate to the same website inside **Safari.** You will now see that you are also logged in, showing that there is no site isolation.

12. If you repeat the same steps with Google Maps, the behaviour is different and the state is not shared with Safari.

3.3.1.2. Does Apple use SFSafariViewController?

One plausible response Apple may make is that they don't use SFSafariViewController, and instead they use a separate component. Based on the user visible features between what we can see in Apple's apps and third party apps they appear to be identical. Additionally both third party apps and Apple's apps trigger "com.apple.SafariViewService" which we believe is the underlying process for service used by **SFSafariViewController.** This strongly suggests that it is essentially the same.

3.3.2. Are Safari and SFSafariViewController Different?

SFSafariViewController functions as a cut-down version of Safari, sharing many of its core features and settings. It is built entirely on WKWebView, the engine that powers Safari, and uses key Safari functionalities, such as zoom settings, translations, and native app install banners, in an identical manner.

State data, including cookies and login sessions, is shared between SFSafariViewController and Safari when used in Apple apps, with this data stored and managed within Safari's settings.

Apple's documentation highlights SFSafariViewController as providing a "Safari-like experience," incorporating Safari-specific features like Password Autofill and Secure Browsing. This deep integration underscores that SFSafariViewController is not a separate browser but a trimmed-down interface built directly on Safari's infrastructure.

Given this level of overlap, it is fair for the CMA to consider SFSafariViewController as part of the Safari browser, reflecting its role as an extension of Safari's ecosystem rather than a distinct or standalone browsing tool.

3.3.3. Respecting User Choice in Default Browsers Across iOS

The core concern is that within native apps, Apple effectively takes control of the user's entire web experience, regardless of the browser the user has selected as their preference. Considering Apple's significant stake in the native app ecosystem, it's reasonable to question whether their decisions regarding the web truly align with the best interests of users, rather than serving their own strategic goals.

The solution is straightforward: when a user selects a default browser, that choice should be honored consistently across the entire operating system, including within SFSafariViewController.

3.4. In-App Browsing Rules Limit Competition and Choice

> *"In our provisional view, banning the use of alternative browser engines for in-app browsing limits the development of the user experience within apps, and of new innovative products. It also limits the possibility that apps with in-app browsers might introduce new features that could be adopted or introduced more widely and therefore improve competition between standalone browser engines and between mobile browsers. One such example is the experience of Meta, a firm with millions of users through popular apps such as Facebook and Instagram. This is set out in box 5, below.*
>
> ***Box 5: Case study on missed innovations: Meta's desire to build its own in-app browser on iOS***
> *• Meta told us that it wants to build an in-app browser using its own browser engine on iOS that it could customise completely to create in-app browsing experiences.*
> *• According to Meta, this would allow it to develop new features that could improve user experience, security and performance, for example, by being able to more quickly load web pages and also to make the in-app browser more stable.*
> *• While Meta has been able to do this on Android, it cannot develop these features on iOS currently because Apple's rules require apps to use Apple's own technology – including its WebKit browser engine – for in-app browsing within apps like Facebook.*
>
> *Second, apps are prevented from using mobile browsers in place of a technical solution currently offered by Apple for in-app browsing, and this limits traffic to alternative browsers and browser engines, and reduces competitive pressure on Apple's offering of in-app browsing and Safari. We provisionally consider that it may be limiting the growth of alternative browsers and preventing innovation that could benefit apps and consumers."*
>
> MIR - Provisional Decision Report

While the MIR team demonstrates a strong understanding of the technical aspects of in-app browsers (IABs) and their functionality, we fundamentally disagree on the core issue that needs to be addressed.

Our primary concern lies in the silent overriding of users' default browser choices. This occurs in many places such as Apple's SFSafariViewController, Google's Android Google Search App, ByteDance's TikTok, and Meta's apps like Instagram and Facebook Messenger.

We firmly believe that users' choice of default browser, the one that opens HTTP/HTTPS links from non-browser apps, should be meaningful and respected. This is based on a combination of what is best for businesses and consumers. Companies seeking to have users adopt their browsers should persuade them to make an active choice, not bypass this choice through technical overrides.

We have explored this issue extensively in our paper.

The MIR team however, seems focused on granting app developers greater freedom to create their own in-app browsers, citing Meta as a prominent example of a developer interested in such flexibility and allowing app developers more freedom to choose which in-app browser users of its app use.

3.4.1. Why this undermines Browser Competition

> *"App developers can address these concerns directly, by taking steps to promote their users' "sense of place," i.e., their awareness of the app that they are using.*
>
> *For example, to promote its users' sense of place, Meta recently implemented changes to its apps' user interfaces, displaying "Facebook" or "Instagram" in the IAB header, as shown in Figure 1 below.*

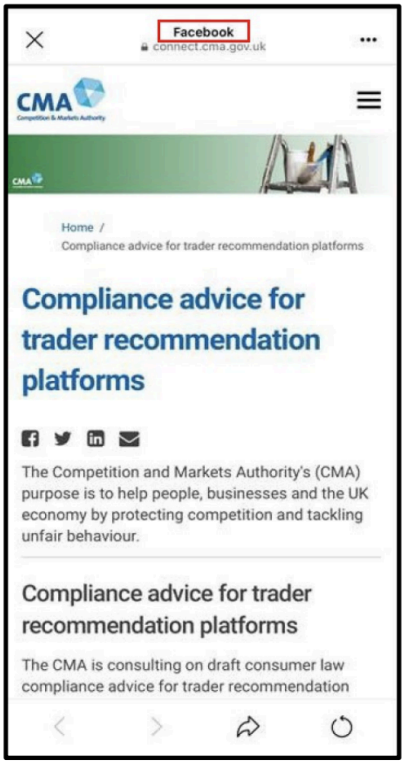


Figure 1: Example of a web page displayed in Meta's IAB (UK)

> *While **Meta supports measures that enhance users' awareness of the apps they are using**, Meta believes 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. These actions would diminish both the substantial benefits that IABs deliver to consumers, and competition and innovation among browsers and browser engines."*
>
> Meta - Response to MIR Working Paper 4

Allowing Meta to silently replace a user's default browser with its own in-app browser significantly undermines competition and user autonomy. Meta's own response does not deny that most users are unaware they are using Meta's browser.

This lack of awareness appears to be a deliberate choice by Meta. The in-app browser lacks distinctive branding or logos and does not actively promote its benefits over the user's default browser. By keeping users in the dark, Meta sidesteps any need to compete on functionality or user experience, as users cannot identify the new browser as the source of any issues or limitations they encounter.

While it is a good step that Meta have added the word Facebook or Instagram on their in-app browsers (likely in direct response to the MIR), we believe that this is insufficient and may be unclear to users (i.e. they may believe that that is the app that is hosting the in-app browser rather than that Meta's in-app browser has replaced their default browser). **We believe at a minimum, the simplest, cleanest, best for users and best for competition solution would be to ask for explicit permission.**

This lack of active choice strikes at the core of browser competition. Without clear and active user choice, there is no accountability or incentive for the in-app browser to improve.

The MIR team has invested significant effort in developing remedies to enhance choice architecture on both iOS and Android. However, the user's choice of a default browser is meaningful only if it is respected. If a user genuinely prefers to open links in an app's in-app browser, that option should exist, but it must be the result of an active informed choice. Allowing apps to silently override the default browser subverts user intent and erodes trust in the digital ecosystem.

3.4.2. Motivation’s of Native Apps

As the CMA's mandate is to foster competition, it is crucial to recognize that the core focus of the MIR investigation should be resolving the barriers preventing the web from competing effectively with Native Apps. The web’s greatest strength lies in enabling direct business-to-customer relationships, eliminating the need for intermediaries and gatekeepers such as those who run app stores or operating systems.

Native apps lack strong motivation to provide a good for the user in-app browser experience because their primary goal is to keep users engaged within the app itself. By design, native apps can discourage users from fully exploring external content, installing third party apps and effectively trap them within the app’s ecosystem. This strategy helps prevent users from leaving the app, increases retention, and ensures the app retains control over the user’s attention and data, at the expense of the third party and even at the expense of providing a quality browsing experience.

Browsers lack the same motivation as native apps to restrict users (sans Apple) because their primary value proposition is providing seamless, open access to the web. Unlike native apps, which aim to retain users within their ecosystem, browsers benefit from encouraging user exploration and interaction with diverse web content. Their success depends on offering a superior browsing experience with regards to performance, trust, usability, and functionality, as users can easily switch to a competing browser if their expectations aren't met. This open, user-centric approach aligns browsers more closely with supporting innovation and competition across the broader web ecosystem and is what will allow web apps to compete with native apps.

Mini-apps in China serve as a compelling example of how apps themselves can evolve into gatekeepers, further fragmenting the digital ecosystem. Platforms like WeChat or Alipay host these mini-apps within their ecosystems, allowing users to access services without leaving the primary app. While this model creates convenience, it also consolidates control within the host app, dictating the terms of access, user data, and monetization, in a very similar manner to the app stores.

Our primary concerns with in-app browsers stem from:

- Behaviors designed to trap users within apps and prevent them from fully engaging with external web content.
- Poor quality implementations that fail to support important web features, limiting user experience and functionality.
- The inability to install web apps directly, stifling competition with native apps.
- Significant fragmentation caused by modifications introduced by native apps, which complicates testing and development for web standards.

3.4.3. Unaddressed Privacy Concerns

We are also particularly troubled by the MIR team's reliance on Meta as an example. In our analysis, we highlight serious privacy concerns raised by Meta's injection of JavaScript that effectively acted as a keylogger on third-party websites. This was done without the end user's knowledge or a business relationship with the affected websites. While we have no evidence of how the data collected was used, we contend that such data should not have been collected in the first place, and the mere presence of this code raises significant concerns.

> *"Apple and other parties such as Open Web Advocacy (OWA) also raise concerns over IABs' use of JavaScript injection. These concerns are largely speculative (resting on technical possibilities, rather than evidence of actual misuse) and apply equally to dedicated browsers.*
>
> *Many of the allegations concerning potential misuse of JavaScript injection by IABs cited in WP4 appear to derive from arguments advanced by OWA, which are in turn largely based on commentary from Felix Krause. However, Krause's allegations concerning Meta's use of Javascript injection are entirely speculative; Krause himself acknowledged that his statements addressed only 'what is possible on a technical level,' not that Meta's IABs "actually steal . . . passwords, address and credit card numbers" or otherwise misuse data. Tellingly, private plaintiffs that filed lawsuits against Meta in the aftermath of Krause's allegations later agreed voluntarily to dismiss their claims."*
>
> Working Paper 4 - Meta Response

Rather than outright denying the allegations, providing a thorough explanation for the code's purpose, or confirming that the code is no longer active, Meta instead focuses on

discrediting the reliance on a single security expert. The second paragraph's careful phrasing avoids denying the collection of data via the injected JavaScript, but simply argues that any misuse of the data is speculative.

This response does little to inspire confidence and fails to adequately address the underlying privacy concerns.

3.4.4. How to fix it?

In our paper, we proposed 6 remedies. Of the remedies relevant to companies such as Meta and Bytedance who would like to produce their own in-app browsers, the lightest touch remedy would be obligating them to ask permission to replace the user's default browser for handing outbound links from their app.

Users should be allowed to opt-out of using the apps own in-app browser in two ways.

First is a per-app permission for using the app's own in-app browser to manage http/https links to non-cooperating third-party websites.

A possible message could be:

> *Allow "App Name" to use its own in-app browser instead of your default browser:*
> *Allow Once*
> *Allow*
> *Don't Allow*

Second, a global user opt-out blocking apps from asking for this permission to stop every new app bothering the user if they have already decided they wish to always use their default browser for http/https links to non-cooperating third-party websites.

A possible settings page could be:

> ***Allow Apps to request to use its own in-app browser [Toggle Button]***
> *Allow Apps to request to use their own in-app browser instead of your default browser.*
>
> *When this is off, all new requests by apps to use their own in-app browser instead of your default browser will be denied.*
>
> *Apps that have asked for permission to use their own in-app browser will appear here.*

This per-app opt-out paired with a global opt-out has a precedent on iOS with the app tracking transparency settings that iOS introduced in iOS 14.0 on September 16, 2020.

Operating system gatekeepers would then need to enforce respecting these settings via app store rules and to abide by it for their own non-browser apps.

This would allow companies such as Meta the opportunity to create and compete in the provision of in-app browsing while removing their ability to silently replace the user's default browser without consent. This would also place the choice back where it belongs, in the hands of the user. App developers should not be able prevent users from leaving their app to the wider web, nor should they have the ability to track what those users do once they click on those links. We ask that the MIR team reconsider their position on this matter.

3.5. Apple's Design Choices Hinder Consumer Browser Choice

> *"We have provisionally found the pre-installation and prominent placement of Safari and default settings on iOS devices reduce user awareness, engagement and choice, increases barriers to entry and expansion for other browser vendors and further reinforces Safari's very strong position on iOS.*
>
> *In addition, Apple's design choices for how users need to navigate through device settings make it harder for users to change their default browsers away from Safari after the device set up."*
>
> MIR - Provisional Decision Report

We wholeheartedly agree with this assessment.

Recent developments have shown some progress, notably Apple's decision to roll out its default settings page, originally created to comply with the EU's Digital Markets Act, on a global scale. Apple has also addressed a significant dark pattern that we previously highlighted and that is referenced in the provisional decision report.

However, several barriers remain that continue to reinforce Safari's stronghold on iOS. These challenges must be addressed to ensure fair competition and user choice in the browser market on iOS.

3.6. Google's Design Choices Limit Browser Choice

> *"Google's control of the Android operating system means it is able to determine key design decisions such as which products are placed prominently on a user's screen and which apps are treated as the 'default' option. We have seen evidence that this is happening in relation to how browser options are presented when users first get their device, and again later, while they are using it.*
>
> *Google uses factory setting agreements with device manufacturers who use Google's Android operating system, with Chrome being pre-installed, prominently placed,[12] and often set as the default browser on many devices. This can be seen in the three diagrams in Figure 1.4 below.*
>
> *We recognise that it can be helpful for consumers to have phones which are ready to use 'straight-out-of-the-box', but we have provisionally found that the use of factory settings which see Google's mobile browser app frequently pre-installed, given prominent placement, and in some cases set as default can limit competition, particularly given low levels of user engagement with these types of products.*
>
> *We have provisionally found that this raises barriers to entry and expansion for other browser vendors and maintains low levels of consumer awareness and engagement in relation to choice of mobile browsers, reinforcing Chrome's very strong position on Android.*
>
> *Furthermore, after device set-up, Google allows its own apps, such as Gmail and Google Maps, to send 'prompts' encouraging users who have set a different browser as their default to switch back to Chrome.*
>
> *We have provisionally found that Google's use of prompts across multiple access point makes it harder for browser vendors to retain newly switched users and therefore, compete with Google, limiting competition between mobile browsers on Android."*
>
> MIR - Provisional Decision Report

We wholeheartedly agree with this assessment.

We support remedies that would prohibit Google from bundling Chrome's placement and default status with Google Play, whether directly, through fees, revenue-sharing agreements or other such means.

Additionally, we would advocate for banning Google from leveraging its other properties on Android, such as Gmail, as mentioned in the report, to prompt users to switch their default browser to Chrome.

Browsers should compete based on their merits, not on the ability of their vendors to exploit other properties (be it operating systems, operating system in-app browsers, apps, search engines, or app stores) to pressure, manipulate, or coerce OEMs or consumers into adopting their browser.

3.7. Use of New Digital Markets Powers

> *"The Group has considered a number of potential measures which could, in principle, address the competition issues identified above; and concluded that there would be significant risks to the effectiveness of these measures if implemented through the remedy-making powers available to us at the end of this market investigation.*
>
> *During the course of this market investigation, the CMA has been granted powers under the Digital Markets, Competition and Consumers Act which establishes a new pro-competition regime for digital markets. These powers enable the CMA to designate firms as having 'strategic market status' (SMS) in relation to one or more digital activities; and impose forward-looking requirements to guide the conduct of firms designated with SMS.*
>
> *We have provisionally concluded that an effective and comprehensive means of addressing the competition concerns we have provisionally identified is to recommend to the CMA Board that, using these new powers:*
>
> *(a) it prioritises commencing SMS investigations to assess whether it would be appropriate to designate Apple and/or Google for their respective digital activities in mobile ecosystems; and it is recommended that the scope of such SMS investigations includes the supply of mobile browsers, browser engines and in-app browsing technology; and*
>
> *(b) if such designation(s) are made, it considers imposing appropriate interventions, such as those we have considered in this report."*
>
> MIR - Provisional Decision Report

Given the extended duration of both the mobile ecosystems study and the browsers and cloud gaming MIR, we are disappointed that the MIR has opted not to utilize its powers to directly address these issues. This decision risks delaying resolution by several more years.

That said, we recognize that the MIR's powers, while significant, are primarily designed for one-time, decisive interventions. In the absence of feasible and proportionate structural remedies that do not risk unintended harm, any resolution would likely need to rely on behavioral measures.

In our view, Apple has demonstrated notable reluctance to comply with the DMA and, at times, has appeared deliberately obstructive. This is underscored by the European

Commission's preliminary finding of non-compliance and its ongoing investigations into three separate alleged instances of Apple's non-compliance.

This reinforces the critical importance of the CMA's ability, under the DMCC, to act swiftly and adaptively to ensure compliance. Such agility will be both necessary and invaluable in addressing these challenges effectively.

We believe the MIR team should reconsider and immediately implement, at a minimum, a select set of the most important remedies. Once the DMCC comes into effect, the DMU can then assume responsibility for ongoing enforcement, updating any remedies that have been circumvented or whose goals remain unmet. Completely deferring the responsibility for identifying and enforcing remedies to the DMCC risks enabling Apple and Google to continue the anti-competitive behaviors that the MIR's own findings have clearly shown to harm competition and the market.

Delaying action in this manner would undermine the significant effort and public funds already invested in the mobile ecosystems study and the browsers and cloud gaming MIR. Immediate intervention should be seriously considered to prevent further harm and ensure meaningful and immediate progress in restoring competition.

4. Missing Remedies

In our response to the MIR's working paper 7, we proposed that 5 additional remedies were needed in order to solve the core issues that the MIR was created to solve.

These were:

1. **Allow browsers to install and manage web apps using their own browser engine.**
   "Mobile OSes shall allow third-party browsers to install and manage Web Apps using their own browser engine."

2. **A requirement for Apple to implement Install Prompts for iOS Safari.**
   "A requirement for Apple to implement Install Prompts for iOS Safari."

3. **Allow feature parity between Web Apps and native apps.**
   "Where feature parity between Web Apps and native apps is possible, Apple must technically enable it and it should not be artificially prevented either by OS rules or OS design. Apple must not self-preference their own apps, apps sold via their app store or their own services over Web Apps."

4. **Only allow strictly necessary, proportionate, and justified security measures to browsers using their own engine.**
   "Apple and Google can only apply strictly necessary, proportionate, and justified security measures to browsers using their own engine. All security rules and their justifications must be publicly published. All security rules for browser vendors should be available in a single public up-to-date document. Changes to these rules should be subject to regulatory scrutiny"

5. **Allow direct install for browsers**
   "Apple and Google shall allow browsers meeting strictly necessary, proportionate and justified security conditions the ability to be installed directly from their own websites. Taking advantage of this should not impose penalties on the browser vendor nor block them from offering their apps on Apple and Google's app stores or financially penalize them (e.g. 'core technology fee').".

We are delighted that the MIR team has adopted the first and most important of these remedies.

In our previous response we have detailed reasoning for why each of the other remedies are proportionate and necessary. We ask that the MIR team consider including them in their final report.

5. Implement Install Prompts for Safari

> *Many respondents told us they were concerned with the existing installation process for PWAs on iOS. These respondents explained that better install prompts would allow users to understand how to download web apps more easily and feel more secure in doing so. They told us this would increase the popularity of web apps which would in turn make them more economically viable for developers.*
>
> *Summary of Individual Responses to the CMA*

It is crucial for the CMA to intervene and mandate the implementation of install prompts for Web Apps in iOS Safari to ensure fair competition between Web Apps and native apps within Apple's ecosystem.

Currently, the installation process for Web Apps in Safari is obscure, requiring users to navigate a cumbersome multi-step process through the “share” menu, which most users are unaware of.

This lack of visibility effectively sidelines Web Apps, preventing them from competing on equal footing with native apps available through Apple’s App Store. Despite years of developer advocacy, Apple has refused to implement an equivalent feature to native app install prompts for Web Apps, a deliberate design choice that undermines the potential of Web Apps to offer businesses advantages such as improved security, interoperability, and cost efficiency. Concurrently, Apple has progressively added more features to iOS Safari to push users towards native apps on their app store.

Without regulatory intervention, Apple is unlikely to address this imbalance, which stifles innovation and limits consumer choice. Mandating install prompts would rectify this disparity, fostering a more competitive and equitable digital ecosystem.

Install Prompts are the essential missing feature for Web Apps, without which Web Apps will not be able to compete with native apps.

As such, we believe the following remedy is appropriate, justified and proportionate to allow Web Apps to compete with the OS’s apps, app store, services and apps delivered via their app store. We ask that the MIR team consider including it:
"A requirement for Apple to implement Install Prompts for iOS Safari.”

We cover this in more detail in section “3.3.1. Install Prompts” in our earlier submission “OWA - Mobile Browsers and Cloud Gaming - Response to Working Papers 1-6”.

This remedy was absent from the provisional decision report. While ensuring competition between browser engines is critical, if Safari maintains its dominant market share following the interventions, the absence of this feature will continue to prevent the adoption and growth of Web Apps and undermine the success of the market investigation in achieving its intended goals.

6. WebAPK Minting

> *"Overall, the evidence available to date indicates that Google engages in self-preferencing less, in respect of access to functionalities on Android compared toApple's approach on iOS. Lack of access to WebAPKs, which is essential for installing PWAs, is the main issue highlighted by third parties (see paragraphs 4.6 to 4.7). Whilst Google has acknowledged this restriction, its latest submission to the CMA indicates that it is working to resolve it. Google has in some cases provided justifications for lack of access to functionality being provided or noted that it is working towards providing equal access."*
>
> MIR - Working Paper 3

> *"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."*
>
> MIR - Working Paper 7

At the time of publishing its remedies paper, the MIR team had correctly identified Google's self-preferencing by withholding WebAPK minting from third-party browsers. This recognition led to Google acknowledging the issue and beginning work toward equal access.

Unfortunately, between then and the publication of the Provisional Decision Report, the MIR team has completely reversed its stance, unjustifiably absolving Google of responsibility.

> *"The above evidence* does not show that Chrome has greater access *to functionality required to implement user-facing features relative to third-party mobile browsers, with the exception of WebAPK minting. However, given the availability of alternative methods for installing web apps on Android,* ***our provisional view is that this does not impact competition****. For the other functionalities considered, the evidence suggests that third-party mobile browsers have equivalent access."*
>
> MIR - Provisional Decision Report
> (emphasis added)

This shift is deeply concerning, as the MIR team's earlier position had successfully compelled Google to begin addressing its exclusivity over WebAPK minting functionality. Without sustained regulatory pressure, however, there is a high probability that Google will abandon these efforts or otherwise deprioritize them, leaving third-party browsers unable to fairly compete in the provision of web app functionality.

Specifically the MIR team appear to have been swayed by the following arguments from Google:

> *"Google submitted PWAs installed through third-party mobile browsers have the 'competitively significant functions needed to compete'. It stated that any PWA (whether installed on Chrome or a third-party mobile browser) appears on the home screen as a bookmark, can send notifications, and can be updated after installation. Further, Google submitted that other app store services are able to offer similar 'minting' functionalities that they can make available to mobile browsers. As an example of this, Google submitted that Samsung's Galaxy Store makes similar functionality available to Samsung Internet and that third-party mobile browsers could work with Samsung if they viewed this functionality as sufficiently important. For these reasons, Google stated that that access to WebAPK minting does not have a significant impact on a mobile browser's ability to compete."*
>
> MIR - Provisional Decision Report

Google has two main arguments.

First, it pointed to Samsung's Galaxy Store, which offers similar WebAPK minting functionality for Samsung Internet, suggesting that third-party browsers could partner with Samsung if they view this capability as essential.

Second, Google argued that Web Apps installed through third-party browsers already possess all the "competitively significant functions" required, such as appearing on the home screen as bookmarks, sending notifications, and supporting updates after installation. Based on this, Google claimed that WebAPK minting is not essential for maintaining competition.

Both arguments are clearly flawed.

While Samsung has restricted its implementation of WebAPK minting to its own devices and browser, this does not justify Google's behavior. Samsung's exclusivity mirrors Google's, but Samsung's ability to implement this functionality is limited to its own devices. Only Google has the capability to provide universal access to WebAPK minting across all Android devices with Play Services, 99.6% of the Android market covered by Google Play Service (Play Services goes back to Android 5.0 - 2014). Compelling Samsung to share its implementation would not resolve the issue for non-Samsung devices, nor would it be necessary if Google were to share its own implementation. Only Google is in a position to share this functionality across all Android devices with Google Play Services; no other party, including Samsung, would even approach that percentage.

Regarding functionality parity, Google's claim that WebAPK minting is unnecessary contradicts its own actions. The shortcut functionality already previously existed, and Google then invested in WebAPK minting precisely because it addressed key limitations. WebAPK minting allows Web Apps to function as standard Android apps, enabling critical features like badging and availability across all surfaces where native apps appear. This works because as far as the system is concerned (due to WebAPK minting), it is a standard Android app.

Bookmarks, by contrast, function as mere shortcuts and cannot replicate this functionality. They lack critical features required to compete with native apps. For example, shortcuts cannot update long-press menus, limiting their utility. Additionally, the primary issue with shortcuts lies in the fact that the majority of the Android ecosystem historically did not support programmatic additions of shortcuts, making them unreliable. Furthermore, Android launches for shortcuts have not been consistently reliable for third parties.

We have covered this in extensive detail in our paper.

The core issue is how Google's exclusivity in WebAPK minting affects competition between browsers in the provision of web apps. No developer creating a native chat app or similar tool would willingly sacrifice key features like badging. Allowing only Web Apps installed via Chrome to access such features inherently disadvantages third-party browsers.

Is the MIR team genuinely asserting that only Web Apps installed via Chrome on Android should have access to critical features like badging, and that this exclusivity has no effect on competition between browsers in delivering Web App functionality, the very issue the MIR was established to address?

By permitting Google to persist in this self-preferencing behavior and provisionally finding that it has no impact on competition, the MIR team is effectively endorsing Google's actions and enabling it to prevent all third-party browsers on Android from meaningfully competing in the delivery of installed web apps.

We respectfully urge the MIR team to reconsider its provisional findings and reaffirm its earlier conclusion that Google is engaging in self-preferencing by withholding WebAPK minting functionality from third-party browsers.

To address these concerns, we encourage the MIR team to take immediate targeted action within its current powers by compelling Google to provide equal access to WebAPK minting. This would address a clear example of self-preferencing and ensure a more level playing field for third-party browsers on Android.

7. Toward A Brighter Future

OWA believes that the Web's unmatched track record of safely providing frictionless access to information and services has demonstrated that it can enable a more vibrant digital ecosystem. The web's open, interoperable, standards-based nature creates an inclusive environment that fosters competition, delivering the benefits of technology to users more effectively and reliably than any closed ecosystem.

OWA's goal is to ensure that browser competition is carried out under fair terms, that user choice in browsers matters, and that web applications are provided equal access and rights necessary to safely contest the market for digital services.

The MIR team has a critical opportunity to fix key issues that have undermined both browser and Web App competition for over a decade to the benefit of both UK consumers and UK businesses. This will improve interoperability, contestability, and fairness leading to lower priced and higher quality apps — not only for the UK but for the entire world.

OWA believes competition, not walled gardens, leads to the brightest future for consumers, businesses, and the digital ecosystem.

8. Open Web Advocacy

Open Web Advocacy is a not-for-profit organization made up of a loose group of software engineers from all over the world, who work for many different companies and have come together to fight for the future of the open web by providing regulators, legislators and policy makers the intricate technical details that they need to understand the major anti-competitive issues in our industry and potential ways to solve them.

It should be noted that all the authors and reviewers of this document are software engineers and not economists, lawyers or regulatory experts. The aim is to explain the current situation, outline the specific problems, how this affects consumers and suggest potential regulatory remedies.

This is a grassroots effort by software engineers as individuals and not on behalf of their employers or any of the browser vendors.

We are available to regulators, legislators and policy makers for presentations/Q&A and we can provide expert technical analysis on topics in this area.

For those who would like to help or join us in fighting for a free and open future for the web, please contact us at:

Email	contactus@open-web-advocacy.org
Web / Web	https://open-web-advocacy.org
Mastodon	@owa@mastodon.social
Twitter / X	@OpenWebAdvocacy
LinkedIn	https://www.linkedin.com/company/open-web-advocacy