Open
Web Advocacy

OWA - DMA - Review of Apple's Compliance Proposal

VERSION 1.0

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

1. Table of Contents

2. Introduction

The Digital Markets Act (DMA) aims to restore contestability, interoperability, choice and fairness back to digital markets in the EU. These fundamental properties of an effectively functioning digital market have been eroded by the extreme power gatekeepers wield via their control of "core platform services".

The lack of competition on mobile ecosystems is, at its heart, a structural one. Gatekeepers wield vast power due to the security model that these devices are built on. Traditionally, on operating systems such as Windows, macOS and Linux, users can install any application they want, with no interaction from the operating system gatekeeper, either by the business or the end user. Users can then grant these programs the ability to do anything they desire.

Locking down what applications can do, such as restricting which APIs they can access behind user permissions, is not by itself anti-competitive and can bring legitimate security advantages. However, the manner in which it has been implemented on mobile devices is both self-serving and in its current form, significantly damages competition.

This damage surfaces in several forms:

First, the gatekeeper can control what is allowed to be installed on devices they have already sold to consumers, often for a significant profit. They utilize this device-level control to demand a 30% cut of all third-party software that the consumer installs, not on merit, but simply because they control the only mechanisms available to businesses to release that software, and can further block or hinder the consumer from using or acquiring services outside of their app store.

The second is more subtle. In order to deliver their “native” apps to consumers on Android or iOS, developers must create custom applications in specific programming languages for each individual platform. Typically, companies will require separate development teams for each OS. This not only multiplies development and maintenance cost, but puts in place an invisible barrier to interoperability. Even the built up expertise for creating software for a specific platform provides significant lock-in advantages to the platform’s gatekeeper.

Finally, even if a developer has no desire to interact with the gatekeeper, they are forced into a commercial and legally binding relationship with them. This is due to the fact that the gatekeeper inserts itself between the customer and these third party developers. With smartphones now 15 years old, this may seem normal to us now, but imagine if Microsoft demanded that every software provider signed an onerous contract with them or be barred from releasing a product on Windows. What would have been unacceptable, anti-competitive behavior to both consumers, businesses and regulators on desktop, has been tolerated on mobile simply because these computers were considered a “new” category.

Mobile devices are just small computers whose primary input is touch, there is no sacred or magical property that means they have to run on a proprietary app store model. Nothing is stopping mobile computers running on the open model that desktop computers run on, just as there is nothing stopping a desktop computer running the app store model. Inertia and great profits are however powerful forces. Between them, Apple and Google have created a powerful and entrenched duopoly.

Even with the DMA forcing Apple to open up the ecosystem to competition, Apple is still inserting themselves front and center between consumers and app developers. They insist all developers for iOS/iPadOS (including developers who have no intention of using their app store) pay them $100 per year, that they sign the full Apple developer program contract and submit themselves to what is effectively app store review (although nominally locked to security). Worse, they are attaching significant recurring penalty fees to developers who dare to make their software available outside of Apple’s app store. Apple is using every tool at their disposal to dissuade developers from leaving their app store and to undermine the goals of the DMA.

Apple's key excuse to impose this control is security. Apple's argument is, in essence, that only they can be trusted to vet what consumers are allowed to install on their devices. All third parties must submit to their review.

What is needed is a way to securely run interoperable and capable software across all operating systems. Luckily, such a solution already exists and is not only thriving on open desktop platforms but is dominating, and that dominance is growing every year. The solution is of course, the Web and more specifically Web Apps. Today, more than 60% of users' time on desktop is done using web technologies, and that looks set to only grow.

Web Apps have a number of properties that allow them to solve this critical problem. They are run in the security of the browser's sandbox, which even Apple admits is "orders of magnitude more stringent than the sandbox for native iOS apps.". They are truly interoperable between operating systems. They don't require developers to sign contracts with any of the OS gatekeepers. They are capable of incredible things and 90% of the apps on your phone could be written as one today.

So why aren't they thriving on mobile? The simple answer to this question is lack of browser competition on iOS and active hostility by Apple towards effective Web App support, both by their own browser and by their OS. Apple's own browser faces no competition on iOS, as they have effectively barred the other browsers from competing by prohibiting them from using or modifying their engines, the core part of what allows browser vendors to differentiate in stability, features, security and privacy.

The DMA explicitly sets out to right this wrong by mandating that gatekeepers can no longer enact such a ban:

> *"In particular, each browser is built on a web browser engine, which is responsible for key browser functionality such as speed, reliability and web compatibility. When gatekeepers operate and impose web browser engines, they are in a position to determine the functionality and standards that will apply not only to their own web browsers, but also to competing web browsers and, in turn, to web software applications. Gatekeepers should therefore not use their position to require their dependent business users to use any of the services provided together with, or in support of, core platform services by the gatekeeper itself as part of the provision of services or products by those business users."*
>
> Digital Markets Act - Recital 43

The intent stated for this in the DMA is to prevent gatekeepers from dictating the speed, stability, compatibility and feature set of "web software applications".

Apple has announced new rules that would, at first glance, allow browser vendors to port their real browsers to iOS. However, on closer inspection, this is a mirage. Instead Apple appears intent on making it "as painful as possible" for browser vendors to port their engines to iOS/iPadOS. As we will outline in our paper, Apple's current proposal falls far short of compliance. Apple is not only undermining browser competition on iOS, but appears to be actively attempting to prevent the growth of an entire open and interoperable ecosystem that could feasibly supplant and replace their app store model.

Apple have seen the Web as a threat to their app store as far back as 2011, when Philip Schiller internally sent an email to Eddie Cue titled “HTML5 poses a threat to both Flash and the App Store”.

> *"Food for thought: Do we think our 30/70% split will last forever? While I am a staunch supporter of the 30/70% split and keeping it simple and consistent across our stores, I don’t think 30/70 will last unchanged forever. I think someday we will see a challenge from another platform or a web based solution to want to adjust our model"*
>
> Phil Schiller - Apple Upper Management

This attitude appears not to have changed. Faced with the genuine possibility of third-party browsers effectively powering Web Apps, Apple's first instinct appears to have been to remove Web Apps support entirely with no notice to either businesses or consumers. Luckily, under significant pressure, Apple backed down from this particular stunt at the last moment.

Apple is very explicit in its public statement that they initially planned to remove the functionality as the DMA would force them to share it with third-party browsers. Even in their statement backing down, they make it clear they do not intend to allow third-party browsers that use their own engine to be able to install and manage Web Apps. In both statements, Apple cites "security" as the reason for their decisions.

Unfortunately for Apple, it has been unable to prove that Safari or WebKit are actually more secure than its competitors. When obligated by the UK's Competition and Markets Authority to provide evidence to back up its assertion that WebKit was more secure than Blink or Gecko, Apple failed to do so.

> *"... the evidence that we have seen to date does not suggest that there are material differences in the security performance of WebKit and alternative browser engines."*
>
> *"Overall, the evidence we have received to date does not suggest that Apple's WebKit restriction allows for quicker and more effective response to security threats for dedicated browser apps on iOS"*
>
> UK CMA - Interim Report into Mobile Ecosystems

Apple's actions not only hurt the Web ecosystem, third-party businesses (be they browser vendors or software developers), but also make their devices worse for their own consumers. By depriving their consumers of the choice and competition that fair and effective browser and Web App competition would bring, they are worsening the functionality, interoperability, stability, security, privacy, and price of services on their devices.

A reasonable person might argue *Why would Apple make their own devices worse, surely better devices means more hardware sales?* This behavior comes, however, with key advantages for Apple, even if it harms Apple's own consumers.

Critically, service revenue is of growing importance for Apple as their hardware sales have peaked and are declining. Apple has not had a "hit" new product for 14 years, namely the iPad, and, if you are being generous, 9 years for the Apple Watch. It does not currently seem likely that Apple's VR/AR headset will have any significant impact on Apple's overall hardware sales.

The UK regulator cites two incentives: protecting their app store revenue from competition from Web Apps, and protecting their Google search deal from competition from third-party browsers.

> ***Apple receives significant revenue from Google by setting Google Search as the default search engine on Safari,*** *and therefore benefits financially from high usage of Safari. [...]* ***The WebKit restriction may help to entrench this position*** *by limiting the scope for other browsers on iOS to differentiate themselves from Safari [...] As a result, it is less likely that users will choose other browsers over Safari, which in turn* ***secures Apple's revenues from Google****.*
>
> *[...]*
>
> *Apple generates revenue through its App Store, both by charging developers for access to the App Store and by taking a commission for payments made via Apple IAP. Apple therefore benefits from higher usage of native apps on iOS. By requiring all browsers on iOS to use the WebKit browser engine,* ***Apple is able to exert control over the maximum functionality of all browsers on iOS*** *and, as a consequence, hold up the development and use of web apps. This limits the* ***competitive constraint that web apps pose on native apps****, which in turn protects and benefits Apple's App Store revenues.*
>
> UK CMA - Interim Report into Mobile Ecosystems
>
> *(emphasis added)*

These two revenue streams are vast, even for a company of Apple's size. Apple collected $85 billion USD in App Store fees in 2022, of which it keeps approximately 30%. Apple reportedly receives $18-20 billion USD a year from their Google Search engine deal, accounting for 14-16 percent of Apple's annual operating profits.

A third and interesting incentive the CMA does not cite, but which the US's Department of Justice does, is that this behavior greatly weakens the interoperability of Apple's devices, making it harder for consumers to switch or multi-home. It also greatly raises the barriers of entry for new mobile operating system entrants by depriving them of a library of interoperable apps.

> *"****Apple has long understood how middleware can help promote competition*** *and its myriad benefits, including increased innovation and output,* ***by increasing scale and interoperability****.*
>
> *[...]*

> *In the context of smartphones, examples of* ***middleware include internet browsers****, internet or cloud-based apps, super apps, and smartwatches, among other products and services.*
> *[...]*
> ***Apple has limited the capabilities of third-party iOS web browsers, including by requiring that they use Apple's browser engine, WebKit.***
> *[...]*
> *Apple has sole discretion to review and approve all apps and app updates.* ***Apple selectively exercises that discretion to its own benefit****, deviating from or changing its guidelines when it suits Apple's interests and allowing Apple executives to control app reviews and decide whether to approve individual apps or updates. Apple often enforces its App Store rules arbitrarily.* ***And it frequently uses App Store rules and restrictions to penalize and restrict developers that take advantage of technologies that threaten to disrupt, disintermediate, compete with, or erode Apple's monopoly power."***
>
> DOJ Complaint against Apple

Interoperability via middleware would reduce lock-in for Apple's devices. Lock-in is a clear reason for Apple to block interoperability, as can be seen in this email exchange where Apple executives dismiss the idea of bringing iMessage to Android.

> *"The #1 most difficult [reason] to leave the Apple universe app is iMessage ... iMessage amounts to serious lock-in"*
>
> Unnamed Apple Employee

> *"iMessage on Android would simply serve to remove [an] obstacle to iPhone families giving their kids Android phones ... moving iMessage to Android will hurt us more than help us, this email illustrates why."*
>
> Craig Federighi - Apple's Senior Vice President of Software Engineering

The DMA has the power to fix all of these underlying issues and unleash a powerful, open, interoperable and secure competitor to not only Apple's app store but also Google's. Lack of contestability for mobile app stores and mobile operating systems is a key concern for the DMA that viable Web Apps solve.

This will also remove a heavy burden from new entrants into the operating system market; lack of apps. No longer will developers need to develop custom apps for each operating system, any operating system with good web app support and browser competition will support all web apps automatically. Web Apps support operating systems that developers have not even heard of. The impact of allowing them to compete fairly on mobile will be profound.

We request the Commission open a proceeding into Apple and investigate what we allege is severe and deliberate non-compliance. The number of ways that Apple is not complying is so myriad that we, recognising that the commision does not have infinite resources to pursue all of them simultaneously, have split them into three tranches of remedies.

Some remedies require time and/or pre-requisite remedies in order to be effective. Remedies in this document are ordered to ensure that either competitive benefits are delivered earlier or to unlock future remedies. In this way, we propose a program of continual improvement in the competitive landscape, delivering wins at every step along the path.

While we have attempted to be comprehensive, it is possible, and perhaps even likely, that there will be infringements that we have not included or are not yet aware of.

Our proposed remedies include:

- Restricting Apple's API contract for browsers down to strictly necessary, proportionate and justified security measures.
- Make clear what the security measures are for third party browsers using their own engine by publishing them in a single up-to-date document.
- Removing any App Store rule that would prevent third party browsers from competing fairly.
- Allow browser vendors to keep their existing EU consumers when switching to use their own engine.
- Removing the special placement of Safari.
- Making Safari uninstallable.
- Implementing Install Prompts in iOS Safari for Web Apps.
- Allowing Browser Vendors and Developers to be able to test their browsers and web software outside the EU.
- Allowing Browsers using their own engine to install and manage Web Apps.
- Make notarization a fast and automatic process, as on macOS.
- Allow direct browser installation independently from Apple’s app store.

- Allow users to switch to different distribution methods of a native app and allow developers to promote that option to the user.
- Don't break third party browsers for EU residents who are traveling.
- Opt-Into Rights contract should be removed.
- Core Technology Fee should be removed.
- Apple should publish a new more detailed compliance plan.

Apple is obligated under Articles 5(7), 6(3), 6(4) and 6(7) to fix each of the above issues. Apple has failed to achieve effective compliance with these obligations contrary to Article 13(3). Further Apple has taken numerous and significant steps that obstruct and undermine it in contravention of Article 13(4). Apple has introduced conditions and restrictions on DMA-conferred rights that have no legal basis in the DMA and gone far beyond the restrictions that the DMA does allow by introducing rules that have no basis in security or that are not justified, strictly necessary or proportionate.

Any intervention that the Commission makes will have global ramifications. Regulators around the world are carefully watching the implementation of the DMA as they plan their own regulatory regimes. Already Japan has become the second jurisdiction in the world to explicitly prohibit banning browser engines. Australia, India, Korea and Brazil are all planning on implementing their own versions of the DMA. The UK has just passed the Digital Markets, Competition and Consumers Bill that grants their regulator great power to enforce codes of conduct against tech giants.

Successful resolution of these issues have an exceptional chance of becoming de facto global, as no jurisdiction will want to miss out on clear benefits being enjoyed by EU consumers. However, if Apple manages to successfully avoid complying with the DMA, these problems could persist indefinitely.

We urge the Commission to enforce the DMA and obligate Apple to allow browsers and Web Apps to compete fairly and effectively on their mobile ecosystem. This will unlock contestability, fairness and interoperability. Companies will then have to compete for users on merit, not via lock-in or control over operating systems. Consumers will benefit from choice, better quality and cheaper software, interoperability, and the genuine ability to multihome across devices and operating systems offered by different companies.

The DMA can finally fix a mobile ecosystem that has been structurally broken, and artificially hindered, for more than a decade.

3. Review

3.1. Apple’s New Browser Engine Entitlement Contract

3.1.1. API Contract not Restricted to Only Security

As part of Apple’s compliance proposal, Apple published a new contract, namely the “Web Browser Engine Entitlement Addendum for Apps in the EU”.

This is a contract to be allowed to have the “Alternative Web Browser Engine App (EU)” entitlement profile. To understand what an entitlement is, we can look to Apple’s documentation:

> *“**An entitlement is a right or privilege that grants an executable particular capabilities**. For example, an app needs the HomeKit Entitlement — along with explicit user consent — to access a user’s home automation network. An app stores its entitlements as key-value pairs embedded in the code signature of its binary executable.”*
>
> Apple Developer Documentation
> (emphasis added)

An entitlement is a permission which grants applications the ability to access some specific hardware or software features of the operating system. For example Apple grants the Safari app the ability to access bluetooth via the “com.apple.bluetooth.internal” entitlement.

The contract further specifies:

> *“‘Alternative Web Browser Engine APIs’ means the restricted Application Programming Interfaces (‘APIs’) contained in the Apple Software, which are provided to You under this Addendum for using an Alternative Web Browser Engine.”*
>
> Web Browser Engine Entitlement Addendum for Apps in the EU

This therefore means that this contract is Apple’s terms and conditions for access to these APIs. This applies not only to browsers installed from Apple’s app store but also **applies to browsers that are either installed from a different app store or directly from a website**.

The DMA states:

> *Article 6(7)* ***The gatekeeper shall allow providers of services and providers of hardware, free of charge, effective interoperability with, and access for the purposes of interoperability to, the same hardware and software features accessed or controlled via the operating system or virtual assistant listed in the designation decision pursuant to Article 3(9) as are available to services or hardware provided by the gatekeeper.*** *Furthermore, the gatekeeper shall allow business users and alternative providers of services provided together with, or in support of, core platform services, free of charge, effective interoperability with, and access for the purposes of interoperability to, the same operating system, hardware or software features, regardless of whether those features are part of the operating system, as are available to, or used by, that gatekeeper when providing such services.*
>
> ***The gatekeeper shall not be prevented from taking strictly necessary and proportionate measures to ensure that interoperability does not compromise the integrity of the operating system, virtual assistant, hardware or software features provided by the gatekeeper, provided that such measures are duly justified by the gatekeeper.***
>
> Digital Markets Act - Article 6(7)
> (emphasis added)

This is clearly a strong restriction on which conditions Apple can place on general API access. That is, while 6(7) allows Apple to have a number of strictly necessary, proportionate and justified security conditions to protect the integrity of the operating system, it is not allowed to then add any other conditions it may want.

Apple's contract helpfully separates out which conditions Apple considers to be security requirements in two sections: "You must meet the following security requirements" and "3.1 Security".

Apple should ensure that **each and every condition** in this contract is:
1. To protect the **integrity** of the operating system
2. Strictly necessary
3. Proportionate
4. Duly justified by Apple

This immediately strikes out all conditions in the contract that are not security conditions - which make up the majority of the contract.

Apple must remove all non-security terms from its browser engine entitlement contract. The contract must only contain terms allowed by Article 6(7) of the DMA. OWA supports the contract spelling out the necessary, proportionate, and justified security rules to protect the integrity of the operating system.

3.1.1.1. Article 5(7) - No Security Exceptions

Under Article 5(7), Apple is obligated to allow browser vendors to port their existing browser engines to iOS. Article 5(7) contains no security exceptions.

This means Apple can not seek to prevent or hinder the features and functionality that browser engines can provide by claiming exemptions under security. Article 13(3) and 13(4) in conjunction with 5(7) means that Apple must enable browser vendors to port their browser engines in their entirety, and Apple must provide those engines the access and integration they require to operate.

We would propose that the Commission therefore looks at security rules through this lens and ensure that any security rule that would significantly hinder browser vendors from porting their existing engines should be struck out. Major browser vendors take security extremely seriously and will likely voluntarily accept a number of rules related to patching and promptly fixing vulnerabilities as they already do this for every other operating system.

Thus, any app store or API contract rule that would make it extremely difficult or impossible for a competent browser vendor to port their existing browser (with its engine) to iOS would be in violation of Article 5(7) and Article 13(4).

If we look at the intent of Article 5(7) with respect to browser engines it goes further. The intent is spelt out in this segment of Recital 43:

> *"In particular, **each browser is built on a web browser engine**, which is responsible for key browser functionality such as speed, reliability and web compatibility. **When gatekeepers operate and impose web browser engines, they are in a position to determine the functionality and standards that will apply not only to their own web browsers, but also to competing web browsers** and, in turn, to **web software applications**."*
>
> Digital Markets Act - Recital 43
> (emphasis added)

The intent is to allow third-party browser vendors to contest the functionality, speed and stability of the gatekeepers browser, including in the provision of the functionality of Web Apps.

This means that any app store or API contract rule that would block functionality from a third-party browser would be in violation of Article 5(7) and 13(4). The most that Apple can insist upon is that browser vendors take reasonable steps to mitigate any security issues to at least the baseline level of security for that API (or equivalent APIs) on iOS.

OWA expects that browser vendors should be subject to security requirements, but as previously stated, security should not be used to block the ability of browsers and Web Apps to compete with the gatekeepers native apps and services.

In summary, **security rules (and other app store rules) that violate Article 5(7) should be modified or struck out.**

3.1.2. Must not use the Browser Engine of the Operating System

Due to Apple’s 15 year ban of third-party browser engines, browser vendors will need to gradually phase in their own engines over time using phased roll-outs and multi-variant testing. Deploying an engine to a new operating system is a complex process and has to be done in a slow and methodical manner to identify bugs and performance issues.

As a result, it is essential that all browser vendors be allowed to ship dual engine browsers. That is, browsers that can use both the system provided WKWebView and their own engine within a single binary. Technically the binary would only contain the code for a single engine, the one the browser vendor provides, since the WKWebView is an operating system provided component.

However, Apple has added a rule in their contract to explicitly ban this:

> *"Be a separate binary from any Application that uses the system-provided web browser engine"*
>
> Web Browser Engine Entitlement Addendum for Apps in the EU

To our knowledge there is no reasonable or rational reason to impose this restriction.

It is not entirely clear what this rule means, but the most obvious interpretation is that the browser is not allowed to make use of the WKWebView. Potentially it could mean that the browser is also not able to make use of or call SFSafariViewController (which may have some niche use cases even for browsers that ship their own engine).

The WKWebView is an operating system component, and thus a software feature of iOS covered under Article 6(7). Since Apple has not provided any security justification for this rule, nor do we believe one plausibly exists, this term is in clear contravention of 6(7).

Without explanation we can only guess Apple’s reasons for adding this rule. The most obvious guess is that Apple is simply being malicious by ensuring browser vendors can not update their existing browser app to use their own engine, which will then ensure that those browser vendors will lose all their existing EU customers if they attempt to port their engine over.

It is also possible that this is simply as childish and anticompetitive as “if you want to not be forced to use it, then you're not allowed to use it at all”. Apple has behaved in a similar manner in a number of other instances with their app store and their payment handler.

A significant consequence of this rule is that it makes phasing in the engine over time with A/B testing impossible. A/B testing (also known as split testing or bucket testing) is a methodology for comparing two versions of an app in order to know which performs better. In the case of browsers, there are bugs which only occur rarely i.e for 1/10,000 or 1/100,000 users. In order to pick up these bugs before releasing them on millions of users, browser vendors turn on features for a percentage of users (i.e 1% or .1%) to catch issues. This is a critical part of browser development.

In this case, browser vendors will be toggling users between the WebKit version of their browser and their own engine version of their browser so as to collect sufficient bug data to make their own engine version stable. Such a rule would make this impossible.

This will make it significantly more difficult for browser vendors to port their engines, let alone have a successful product using their own engine which meets or exceeds the quality and adoption of their WebKit WebView-based browser. This is the only real plausible purpose of the rule. Coupled with the complete lack of security or any other justification, this rule would be in violation of Article 5(7), and Article 13(4).

Thus, under Article 5(7), Article 6(7), and Article 13(4) **Apple must allow browser vendors to ship "dual engine" browsers by removing** *""Be a separate binary from any Application that uses the system-provided web browser engine"* **from their browser engine entitlement contract and not include an equivalent rule in their app store rules.**

3.1.3. Must be New and Separate App

Apple has a number of rules making it clear that the Web Browser Engine Entitlement will only be provided in the EU.

Additionally, they will force browser vendors to submit a brand new application called an "Alternative Web Browser Engine App (EU)" in the contract.

> *"Your Application must:*
> *• Be distributed solely on iOS in the European Union;*
> ***• Be a separate binary from any Application that uses the system-provided web browser engine;***
> *...*
> *2.4 The Entitlement Profile is compatible and may only be used with Applications solely distributed within the EU on devices running iOS 17.4 or later*
> *...*
> *You are permitted to use the Entitlement Profile only in connection with Your Alternative Web Browser Engine App (EU) developed or distributed under this Addendum and with Apple-branded products"*
>
> Apple's Browser Engine Entitlement Contract

Apple has chosen to restrict browser competition on iOS to the EU. This means under Apple's current proposal all browsers that intend to bring and modify their own engine will need to create a brand new application. Existing users will need to switch to this "new" app or remain siloed on the existing WKWebView version of their browser.

The rule to ship a separate app in the EU does not affect Safari, as the WKWebView is under Apple's sole control, contains Safari's engine and is updated in lockstep with Safari. That is to say: the system provided browser engine is Safari's engine, hence they are automatically exempt from this rule.

For a third-party browser to make this transition, they would need to ship a new app, and then advertise to the existing users to switch to the new app. The ramifications of this is so severe it's unlikely any browser vendor would be willing to take this step as it causes serious issues:

- Browser vendors would likely lose a significant percentage of their existing users in the transition.
- Their "real" browser (i.e. the one they ship on every other operating system) will start with zero installs. This means that, even after a period of time, it may still not be eligible for the choice screen as it only displays the top 12 browsers for the country of the user. It would be worth investigating whether imposing the "new and separate app rule" would therefore breach 13(4)'s effective compliance obligation in relation to Apple's 6(3) choice screen obligations.
- This introduces considerable friction and complexity into porting users data, login cookies and settings to the new app. This is problematic as:
    - In some cases user's data will be lost in the transfer.
    - Significant development work will be required to sync it.
    - It is unclear what to do for users that become and cease to be EU residents,an artificial problem caused by Apple's geolock.
    - Users opting to keep both applications may find their information out of sync across the two.
    - Browsers will likely want to avoid any high-friction method such as requiring the user to sign-in to use browser cloud syncing functionality, if available.
- This is likely to lead to significant user confusion and frustration. Many users are completely unaware that they are not using the "real" versions of popular browsers. This is further complicated by Apple's insistence (previously indefinitely, now for the next 6 months) that browser vendors would not be able to port their browsers to iPad in the EU. Most users are not concerned with what engine their browser uses, rather they are concerned with the features, stability, speed, security and privacy, etc. that different browsers offer relative to each other.

  It is unlikely that a browser vendor would do a major advertising campaign for the EU for their updated browser that would inadvertently highlight the shortcomings of their WkWebView based browser to other large markets. While there is a high probability that Apple will be forced to allow browser competition on iOS in most (if not all) jurisdictions in the future, it is equally likely that there will be a multi-year lag. Browser vendors will not wish to lose market share with their WkWebView based browsers in those jurisdictions while they wait for change.

All of this is avoided if the browser vendor can simply update their existing application.

Hypothetically, if Apple were to update the technology powering the Apple TV app, they would not force all their users to download and sign into a new app. They would simply silently update the technology under the hood for all users, providing them a better product with no friction or user frustration. To force browser vendors to have to ship a new app simply to update the underlying technology is both unreasonable and anti-competitive. Particularly when allowing browser vendors to use their own engines for their browser is directly compelled via the DMA.

3.1.3.1. Potential Solutions

The issues above are of course entirely of Apple's own making. Apple's decision to restrict browser competition on iOS to the EU, while potentially entirely legal under the DMA, is the root cause of this harm.

There are a three potential solutions open to Apple including:

Solution A. Allow Browser Engines Globally

Solution B. Two Binaries for One Bundle ID

Solution C. Global Dual Engine Binary with Toggle

3.1.3.1.1. Solution A - Allow Browser Engines Globally

Apple could allow browsers to compete fairly with their own engines globally. This is the only option that would enable fair competition.

However we believe this is an unlikely option since:

1. The DMA has no ability to impose extra-jurisdictional remedies, and OWA is unaware of other legal mechanisms that could force this change.

2. Apple's revenue from Safari is reportedly $20B USD per year, an amount so significant that it's unlikely Apple would willingly enable competition unless compelled.

3. Enabling browser competition on iOS globally in the provision of features of Web Apps, will increase Web Apps ability to contest the Apple's app store which is a significant source of revenue for Apple.

The UK’s Competition and Markets Authority noted both of these potential motives for Apple’s rule locking browsers to the WKWebView in their report.

> ***Apple receives significant revenue from Google by setting Google Search as the default search engine on Safari,*** *and therefore benefits financially from high usage of Safari. Safari has a strong advantage on iOS over other browsers because it is pre-installed and set as the default browser. The WebKit restriction may help to entrench this position by limiting the scope for other browsers on iOS to differentiate themselves from Safari (for example being less able to accelerate the speed of page loading and not being able to display videos in formats not supported by WebKit). As a result, it is less likely that users will choose other browsers over Safari, which in turn secures Apple’s revenues from Google.*
>
> *Apple generates revenue through its App Store, both by charging developers for access to the App Store and by taking a commission for payments made via Apple IAP. Apple therefore benefits from higher usage of native apps on iOS. By requiring all browsers on iOS to use the WebKit browser engine,* ***Apple is able to exert control over the maximum functionality of all browsers on iOS*** *and, as a consequence, hold up the development and use of web apps. This limits the* ***competitive constraint that web apps pose on native apps****, which in turn protects and benefits Apple’s App Store revenues.*
>
> [UK CMA - Interim Report into Mobile Ecosystems]
>
> *(emphasis added)*

The reason we have included this as an option is that by providing it as the best and most reasonable option from a competition point of view. It gives us a frame of reference for Apple’s other options. The other options, while possibly legal, are still anti-competitive and place competing vendors under significant burden.

3.1.3.1.2. Solution B - Two Binaries for One Bundle ID

Allow browser vendors to provide two signed binaries under one application (one bundle id).

These two binaries would be:

One - A signed binary which contains the real browser with its own engine to ship to the EU and other jurisdictions that mandate Apple allow browsers be able to choose and modify their engine. This would need to include the ability to do A/B testing with WKWebView outlined above.

Two - A signed binary which contains the version of the browser which is forced to use Apple's WKWebView which can be shipped in other jurisdictions.

An update mechanism that can then toggle which binary to deliver based on if the end user is an EU resident. This would likely require some development work on Apple's part to update their distribution mechanism and to ensure that both binaries can continue to access user data.

One problem with this solution is the question of what happens when users become or cease to be EU residents. This would necessitate a complex swap over procedure by the browser vendors where local storage data is copied from one version of the browser to another. This could be a significant source of bugs.

3.1.3.1.3. Solution C - Global Dual Engine Binary with Toggle

Allow browser vendors to ship a single binary globally which contains both the browser vendor's own engine and the WKWebView version. For regulatory jurisdictions that haven't yet forced Apple to allow competition, the browser would use the WKWebView, and for those who have, the browser would use the browser's own engine.

Apple can indicate to the browser app whether they are allowed to use their own engine, and based on that the browser could then decide if they wish to use their own engine.

Likely only a small technical change would be required for this solution to work. That is for Apple to provide some mechanism to let browser vendors detect whether they can use their engine or not, as legally required. Aside from that this would likely require no technical changes on Apple's end, they would simply need to update their contracts to allow it. In the event Apple is unwilling to develop such an API, Apple can simply update their contracts and each browser vendor can detect whether a user is an EU resident to the best of their abilities using their own mechanisms such as IP address.

The downside to this solution for browser vendors is that they would then be forced to ship an additional 60 MB - 70 MB (OWA's estimate) to many millions of users who would not be able to use that engine due to Apple's browser engine restrictions. This would then have an effect on user acquisition and retention rates as well as those users' data usage.

We recommend that the Commission reach out to the vendors to ask their opinions, and whether this solution would be feasible, if required to distribute a single global binary.

3.1.3.2. Summary

It is not acceptable for Apple to use their mechanics of restricting browser engines to the EU to suppress and undermine browser competition in the EU. If Apple can not be compelled to allow browser competition globally on iOS, they should be compelled to make browser competition on iOS within the EU **at least** minimally viable. The aim should be to allow third party browser vendors to effectively contest Safari's features, performance, security and privacy on a fair playing field.

The commission does not need to force Apple to do anything outside of the EU. The commission simply needs to compel Apple to allow browser vendors to update their existing browser apps for EU customers to use their own engine, if Apple wishes to implement a complex solution to geolock that to the EU, that is Apple's choice, not the Commissions. Solutions A and Solution C are both straightforward ways in which Apple could achieve a fairer result with minimal technical work.

3.1.4. Review of Security Terms

3.1.4.1. Reasonable Security Clauses

OWA would recommend asking each of the browser vendors their opinions on whether they believe each security clause in Apple's contract is reasonable or if any modifications should be made. The following security clauses which Apple lists in their contract appear to be reasonable on our first review, however OWA would need additional time to comprehensively analyze each of these conditions.

1. **Secure Development, Supply Chain Monitoring & Best Practices**
   "*Commit to secure development processes, including monitoring Your Application's software supply chain for vulnerabilities, and following best practices around secure software development (such as performing threat modeling on new features under development);*

2. ***Vulnerability Submission Portal & Policy***
   Provide a URL to a published vulnerability disclosure policy that includes contact information for reporting of security vulnerabilities and issues to You by third parties (which may include Apple), what information to provide in a report, and when to expect status updates;

3. ***Timely Vulnerability Fixes***
   Commit to mitigate vulnerabilities that are being exploited within Your Application or the Alternative Web Browser Engine in a timely manner (e.g., 30 days for the simplest classes of vulnerabilities being actively exploited);

4. ***Publish Vulnerabilities***
   Provide a URL to a publicly available webpage (or pages) that provides information on which reported vulnerabilities have been resolved in specific versions of the browser engine and associated Application version if different;

5. ***Root Certificate Policies***
   If Your Alternative Web Browser Engine uses a root certificate store that is not accessed via the iOS SDK, You must make the root certificate policy publicly accessible and the owner of that policy must participate as a browser in the Certification Authority/Browser Forum;

6. ***Support Modern TLS***
   Demonstrate support for modern Transport Layer Security protocols to protect data-in-transit communications when the browser engine is in use

7. ***Memory Safe Languages OR Memory Safety Features***
   Use memory-safe programming languages, or features that improve memory safety within other languages, within the Alternative Web Browser Engine at a minimum for all code that processes web content;

8. ***Latest Security Mitigations***
   Adopt the latest security mitigations (for example, Pointer Authentication Codes) that remove classes of vulnerabilities or make it much harder to develop an exploit chain;

9. ***Best Security Practices***
   Follow secure design, and secure coding, best practices;

10. ***Process Separation and Validate IPC***
    Use process separation to limit the effects of exploitation and validate inter-process communication (IPC) within the Alternative Web Browser Engine;

11. ***Supply Chain Vulnerability Monitoring***
    Monitor for vulnerabilities in any third-party software dependencies and the Your Alternative Web Browser Engine App (EU)'s broader software supply chain, migrating to newer versions if a vulnerability impacts Your Alternative Web Browser Engine App (EU);

12. ***Don't use Orphaned Software Libraries***
    Not use frameworks or software libraries that are no longer receiving security updates in response to vulnerabilities;

These rules appear reasonable **on the provision that Safari must equally abide by them**. For these rules, the current level of security of Safari and Apple's native app ecosystem should be the benchmark. No browser should be penalized for doing a better job at security than Apple's accepted status quo on iOS.

Apple attempted to claim to the UK's CMA that WebKit's security was superior to that of Gecko and Blink, however the UK regulator found that Apple had no evidence to back up the assertion. In Apple's next response to the CMA they had dropped this claim.

> *"... in Apple's opinion, WebKit offers a better level of security protection than Blink and Gecko.*
>
> *... the evidence that we have seen to date **does not suggest** that **there are material differences in the security performance of WebKit and alternative browser engines.***
>
> *Overall, the evidence we have received to date **does not suggest that Apple's WebKit restriction allows for quicker and more effective response to security threats for dedicated browser apps on iOS**"*
>
> *UK CMA - Interim Report into Mobile Ecosystems*
>
> *(emphasis added)*

Both developers and users are already aware that installing a native app is more dangerous than visiting a website, including if that app has passed a brief human "app review".

> *"**The most dangerous feature that browsers have** are not the device API's; it **is the ability to link to and download native apps.**"*
>
> Niels Leenheer - HTML5test
>
> *(emphasis added)*

Apple strongly relies on app store review as a mechanism to protect users from iOSs weaker sandbox for Native Apps. Apple has placed great trust in "app review", despite the fact they reportedly only employ a mere 500 reviewers. These reviewers are expected to review 100,000+ Native Apps per week, without the benefit of access to source code. While they are assisted by automated tools, each reviewer gets less than 15 minutes to manually tap through apps, a process which is unlikely to be effective at picking up many types of malicious apps.

It doesn't even appear to be effective in picking up apps that violate Apple's payment rules (that grant it a 15-30% stake), something they are likely to want to enforce. In one particularly striking example of app review being ineffective at blocking updates that violate the current rules of the Apple's app store, Epic managed to slip in an entire third party payment solution into their app without it being picked up in review.

As Apple's head of fraud Eric Friedman said regarding review processes:

> *"please don't ever believe that they accomplish anything that would deter a sophisticated hacker".*
>
> *"I consider them a wetware rate limiting service and nothing more"*
>
> Eric Friedman - Apple's former head of fraud on App Store Review

This suggests that while Apple outwardly projects confidence in the value of their app store review some informed insiders view it as worthless.

Browsers use a far more reliable form of security than brief human review upon update. They lockdown the entire environment that the third party code, that is websites and Web Apps, runs in. This environment is called a sandbox and every action a website takes is tightly controlled. Even Apple acknowledges that browsers sandbox is massively superior to iOS's sandbox for native apps:

> *"WebKit's sandbox profile on iOS is orders of magnitude more stringent than the sandbox for native iOS apps."*
>
> Apple's Response to the CMA's Mobile Ecosystems Market Study Interim Report

Thus while we agree that the standard browsers should be held to a higher standard than that of a typical native app, it is critical that any security threat that browsers face be viewed holistically and not in isolation. That is, were the user forced to install a native app to use that functionality, instead of using a website or Web App what would their relative risk be. Would it be significantly better or worse.

For example, if a browser implements a relatively secure, but not perfect, implementation of WebBluetooth, then its level of security should be compared to the current accepted level of security for native iOS apps that iOS already meets in their own CoreBluetooth implementation, not the hypothetical perfect security that Safari's non-existent WebBluetooth implementation would like to obtain.

Security is a subtle art and many of these rules are by necessity subjective. For example "best practices" is no doubt a subject of fierce disagreement on certain points. As such, Apple should only be able to focus on points where there is broad agreement in the industry that certain practices are categorically and unambiguously bad.

Any complaint by Apple based on violation of these rules should be necessary, reasonable, proportionate and justified with extensive evidence by Apple.

In some cases there may be more than one method to mitigate a threat and browser engines' very design may dictate how a browser defends against a particular threat. It is important therefore that Apple's rules do not force browsers to mitigate each threat in specific ways but rather focus on the degree to which each browser has effectively mitigated a threat. In some cases there will be broad agreement between browsers on how to mitigate particular threats, and in those cases the rules can be more specific.

Apple should be very specific about which rule has been broken and its evidence for why it believes that is the case. Apple's current app review approach where apps are rejected without citing specific rules or clearly outlining why, is unacceptable here.

Browser vendors should be consulted on each individual rule to see if they jointly agree that all the conditions are reasonable before settling on the final list.

3.1.4.2. Problematic Security Clauses

The following security terms in the contract are problematic:

3.1.4.2.1. Vulnerabilities in Browser APIs

> *"Prioritize resolving reported vulnerabilities with expedience, over new feature development. For example, where the Alternative Web Browser Engine bridges capabilities between the platform's SDK and web content to enable Web APIs, upon request the developer must remove support for such a Web API if it is identified to present a vulnerability. Most vulnerabilities should be resolved in 30 days, but some may be more complex and may take longer."*
>
> Apple's Browser Entitlement Contract

What Apple is saying here, is that if an API has a vulnerability, no matter how minor and no matter how much effort the browser vendor has taken to mitigate the vulnerability, Apple can demand that they remove the browser API. This is not reasonable, proportionate, strictly necessary and Apple is not offering to justify its decisions. All APIs contain the potential for vulnerabilities. Further, Apple would never accept an equivalent clause for its own apps or browser.

This is particularly important as the primary reason that the DMA grants browsers the right to use their own browser engine is to prevent gatekeepers (such as Apple) from determining the functionality and standards for third-party browsers. Allowing Apple to ban third-party browsers from offering functionality on flimsy (or non-existent) security justifications would severely undermine this goal.

This appears to grant Apple broad powers to use unnecessary and disproportionate security measures to block rival browsers from providing functionality that is distinct from Safari, preventing one of the core aims of the act in allowing browser vendors to port their own engines. It is critical that browsers are allowed to bring functionality to enable browsers and Web Apps to compete with the software and services of the gatekeepers including their native app ecosystems. If Apple is allowed to block functionality then it will prevent this competition. Apple should not be able to impose any security rules that would outright preclude browser vendors from porting such functionality, especially when considering that 5(7) does not have any security carve-outs.

Additionally, in general, under 6(7), security terms which are not strictly necessary or proportionate must be removed or be modified to be in compliance with the act.

There is always a trade off between security and utility. The safest browser would be one that refuses to open. The key here for browser vendors is to offer as much security as reasonably possible for their chosen level of utility. No browser should be compelled to reduce utility in the name of security if they have taken firm efforts to mitigate risks.

3.1.4.2.2. Vulnerability Disclosure

Apple states that all browser vendors must publish a vulnerability disclosure policy and publicly state which reported vulnerabilities have been resolved in specific versions of the browser engine and associated Application version if different.

> *"Provide a URL to a published vulnerability disclosure policy that includes contact information for reporting of security vulnerabilities and issues to You by third parties (which may include Apple), what information to provide in a report, and when to expect status updates;*
> *...*
> *Provide a URL to a publicly available webpage (or pages) that provides information on which reported vulnerabilities have been resolved in specific versions of the browser engine and associated Application version if different"*
>
> Apple's Browser Entitlement Contract

We believe this is reasonable on the proviso that Apple also abides by it when publishing updates to Safari. Safari should not have any privileged status with regards to which security vulnerabilities it publishes or the manner in which it publishes them. An obligation for competitors to publish security vulnerabilities while exempting Apple from the same obligation would provide a significant unfair competitive advantage to Apple.

One example of this Apple failing to adequately warn users of a vulnerability or breach is the xCodeGhost iOS malware.

Documents revealed as part of the current trial of Fortnite vs Apple show that in fact 128 million users downloaded the more than 2,500 infected apps, about two thirds of these in China. Popular apps such as WeChat, Didi Chuxing, and Angry Birds 2, among others, were infected by XcodeGhost. These are some of the largest native Apps in the world, being the equivalents of Facebook and Uber in China. WeChat, for example, has 1.24 billion users.

Apple discussed contacting users and briefly made an announcement on their China website. Shortly after, it was removed. To our knowledge, Apple never contacted users to inform them of the breach.

> *"**this decision to not notify more than 100 million users** about potential security issues seems to **have more to do with protecting the platform's reputation** than helping users stay safe"*
>
> Kirk McElhearn - Intego

The security of iPhones is one of Apple's key marketing claims. According to two dozen security researchers, who spoke on the condition of anonymity, Facebook, Microsoft and Google all publicize their bug-bounty programs and take every opportunity to acknowledge and thank those that find bugs and vulnerabilities sharing valuable insights, weaknesses and solutions. In contrast, Apple conducts its programs in secrecy.

Apple reportedly has a considerable bug backlog:

> *"You have to have a healthy internal bug fixing mechanism before you can attempt to have a healthy bug vulnerability disclosure program. What do you expect is going to happen if they report a bug that you already knew about but haven't fixed? Or if they report something that takes you 500 days to fix it?"*
>
> Moussouris - Helped create Microsoft's Bug Bounty Program

> *"It seems like Apple thinks people reporting bugs are annoying and they want to discourage people from doing so"*
>
> Tian Zhang - an iOS software engineer

> *"Apple's closed-off approach hinders its security efforts"*
>
> Dave Aitel - co-author of "The Hacker's Handbook

> *"Apple is slow to fix reported bugs and does not always pay hackers what they believe they're owed. Ultimately, they say, Apple's insular culture has hurt the program and created a blind spot on security."*
>
> Reed Albergotti - Washington Post

> *"I want to share my frustrating experience participating in Apple Security Bounty program. I've reported four 0-day vulnerabilities this year between March 10 and May 4, as of now three of them are still present in the latest iOS version (15.0) and one was fixed in 14.7, but Apple decided to cover it up and not list it on the security content page. When I confronted them, they apologized, assured me it happened due to a processing issue and promised to list it on the security content page of the next update. There were three releases since then and they broke their promise each time."*
>
> Denis Tokarev

If Apple can avoid full disclosure while compelling other vendors to, Apple may use these disclosures to falsely claim superior security or create a pretense for delisting other browsers from its app store. We recommend keeping this policy with the caveat that it also applies equally to Apple's own browser.

We also believe that non-disclosure agreements, which entirely prohibit security researchers from publishing their findings (once a reasonable number of days has elapsed, for example Google Project Zero uses the shorter of 90 days or 30 days after patched) after reporting them to a particular browser vendor (such as Apple) should be prohibited, particularly if the browser vendor has declined to fix these exploits. The ultimate aim here is to improve security by removing the ability of parties to hide both fixed and unfixed exploits from the public and tech community. This applies great pressure to increase investment in security.

3.1.5. Lack of Justification of Security Terms (and Non-Security Terms)

> *"A gatekeeper can use different means to favour its own or third-party services or products on its operating system, virtual assistant or web browser, to the detriment of the same or similar services that end users could obtain through other third parties."*
>
> Digital Markets Act - Recital 49

One potential mechanism for favoring their own services is blocking competitors with unnecessary and disproportionate security measures. As alleged in the USA Department of Justice's (DOJ) complaint, Apple has a history of using security rules inconsistently to serve their own financial interests.

> *"In the end, Apple deploys privacy and security justifications as an elastic shield that can stretch or contract to serve Apple's financial and business interests."*
>
> DOJ Complaint against Apple

Although 5(7) does not contain carve-outs for security measures, we believe that it would be proportional under the act to require browsers to meet security standards providing it doesn't prevent the browsers from porting their engines, and prevent them from implementing functionality. The language under Article 6(7) of the DMA states that Apple must duly justify any security measures, which Apple should do for 5(7).

Apple has not attempted to provide any justification to any of the rules in their browser entitlement contract or rules found in documents or contracts it pulls in under "Apple Materials". We believe that Apple should be required to provide very detailed security justifications for each individual rule attached to obtaining this API access; Why is it strictly necessary and proportionate to prevent this access from compromising the integrity of the operating system, hardware or software features provided by Apple as per the wording and intent of the act.

It is important that Apple publicly publish these justifications so that they can be scrutinized by the browser development and developer community to make sure that Apple is not attempting to use security rules to undermine either the DMA or competition to their benefit, nor to EU consumers and business users' detriment. This will allow the browser development and developer community to provide evidence and feedback as to if each rule is reasonable and whether Apple's explanation as to its necessity and proportionality are credible.

We support Apple having the ability to remove browsers that are either malicious or consistently fail to meet a reasonable security standard clearly laid out in the rules (i.e patching regularly, fixing vulnerabilities in a timely fashion, etc). However, we believe Apple should need to prove any allegations with evidence and should face consequences for frivolous or unjustified removal of browsers.

3.1.6. Viral Terms

Apple appears to conflate other rules and terms into their contract for API access via the term “Apple Materials”.

It is defined in the contract as:

> *“‘Apple Materials’” means the Documentation, Entitlement Profile, and other materials provided by Apple to You, and which are incorporated by reference into the requirements of this Addendum.”*
>
> Apple’s Browser Entitlement Contract

This is a frustratingly vague term.

Apple also state in the contract that they can not guarantee the availability, completeness, or accuracy of Apple Materials. Given this term encompasses all potential rules the browser vendor must abide by it is critical that these are available, clear, complete and accurate. Browser vendors (and likely regulators) should be made aware of any plan to update them well in advance. Changes should be then subject to scrutiny by the Commission, browser vendors and third parties such as OWA.

From their definition, it appears to encompass all documentation related to third-party browser engines, any documentation those documents reference, the Apple Developer Program License Agreement and the Apple App Store Review Guidelines.

Further, the contract explicitly states on multiple occasions that signing and abiding by the terms in Apple Developer Program License Agreement is required to use these APIs, indeed this contract is structured as an addendum to the Apple Developer Program License Agreement.

These combined documents contain a vast number of non-security terms and are inappropriately referenced in the context of a browser that is not distributed via Apple’s app store.

As such these viral terms must be removed and Apple should clearly present all security rules that they wish third-party browser vendors to abide by in order to access APIs necessary for building iOS browsers.

3.1.7. Conflated Terms

Again “Apple Materials” is problematic as it conflates together various concepts that need to be separate.

It is defined in the contract as:

> *“‘Apple Materials’” means the Documentation, Entitlement Profile, and other materials provided by Apple to You, and which are incorporated by reference into the requirements of this Addendum.”*
>
> Apple’s Browser Entitlement Contract

Specifically this means Apple Materials includes:

- All documentation for the APIs
- The nature of how these APIs function and whether they exist
- All rules the browser vendor has to abide by
- Additional contractual terms
- Any other document referenced

Specifically out of these, the rules which browser vendors need to abide by (which the act mandates can only be security rules) need to be separated out into their own term and in a single document which is available, clear, complete, accurate and up to date.

3.1.8. Hard to Know if in Compliance

Even ignoring Apple's insertion of non-security rules into its API contract, the fact that rules could be hidden in any of a vast range of non-explicitly defined documents, that by Apple's own admission may be incomplete or inaccurate and which they can change at any time without notification, means that staying in compliance becomes a fraught and unpredictable task for even the most diligent and security conscious of browser vendors. Even browser vendors with security significantly superior to Safari in all aspects could not be confident of being in compliance when the existing rules are unclear and changes in compliance requirements are not telegraphed in advance

This is clearly not strictly necessary, reasonable or proportionate.

3.1.9. Severe and Unreasonable Penalties

Apple's Browser Engine API contract contains the following clause:

> *"2.9 While in no way limiting Apple's other rights under this Addendum or the Developer Agreement, or any other remedies at law or equity,* ***if Apple has reason to believe*** *You or Your Alternative Web Browser Engine App (EU) have* ***failed to comply with the requirements*** *of this* ***Addendum or the Developer Agreement,*** *Apple* ***reserves the right to revoke to Your access to any or all of the Alternative Web Browser Engine APIs*** *immediately upon notice to You; require You to remove Your Entitlement Profile from Your Alternative Web Browser Engine Application (EU);* ***terminate this Addendum; block updates of, hide, or remove Your Alternative Web Browser Engine App (EU) and/or other Applications from the App Store; block Your Applications from distribution on Apple platforms; and/or to suspend or remove You from the Apple Developer Program."***
>
> Apple's Browser Engine Entitlement Contract
> (emphasis added)

That is, if Apple believes (note, Apple is not placing any burden of proof on itself here) that any rule has been broken in either this contract or in the developer agreement then they grant themselves the right to:
- Revoke access to the APIs
- Remove the browser engine entitlement from the browser
- Block updates or remove the browsers
- Block other applications the browser vendor distributes on Apple platforms
- Suspend or remove the developer from the Apple developer program

Given the inherent vagueness of the rules that Apple is imposing here and Apple's long history of being a poor and self-serving judge of the implementation of those rules this is an extremely threatening clause even to browser vendors who are competently doing an appropriate job of *"protecting the integrity of the operating system, software and hardware of the gatekeeper".*

This is not strictly necessary, proportionate or reasonable.

While we support Apple's right to revoke access in cases of persistent negligence or significant and provable malice, this clause as it currently stands, will allow Apple to strategically and anti-competitively bully browser vendors over non-existant or minor edge cases.

For example, as this rule stands, if Mozilla were to violate a non-security rule related to scrolling logic buried in a recently updated document on Apple's components, then under

Apple's contract, they would be able to block and remove Firefox from all Apple platforms including macOS. While we think this exact scenario is unlikely due to the regulatory scrutiny it would bring, it does highlight the power Apple is granting itself though this contract and how it subverts the intent of the DMA that the rules for API access be only security rules which are strictly necessary, proportionate and duly justified for *"protecting the integrity of the operating system, software and hardware of the gatekeeper"*.

We believe this clause should be updated with language that impresses the necessity and proportionality of any response by Apple and place a burden on Apple to justify with evidence any punitive action it might take against any browser vendor.

3.1.10. Requirement to use Apple Components

In Apple's browser engine entitlement contract they appear to obligate browser vendors to use iOS user interface libraries, rather than their own already built into Gecko and Blink.

In their documentation for BrowserEngineKit they state:

> *"To correctly render CSS and manipulate the Javascript DOM, your browser app needs to integrate closely with UIKit and customize the way it handles many low-level user interface events. Use view classes in BrowserEngineKit to handle scrolling, drag interactions, and the context menu in your browser app."*
>
> Apple Documentation

This becomes a rule and not a suggestion due to this clause in the contract:

> *"**You agree to use**, only through the use of the Entitlement Profile, **an Alternative Web Browser Engine** in Your Alternative Web Browser Engine App (EU) **only as expressly permitted** in this Addendum and **in the Apple Materials**."*
>
> Apple Browser Engine Entitlement Contract
> (emphasis added)

As discussed before, "Apple Materials" is a disconcertingly vague and broad term that in effect means any imperative statement in any documentation referenced from the contract or referenced from those documents in a cascade are considered binding rules.

This is critical for two reasons. One, it means significant re-architecture of third-party browsers' engines. This is highlighted in this ticket from Google:

> *"It's not entirely clear, but this seems to imply that existing browser rendering engines must be re-architected to fit the UIKit rendering model. Chromium has a large and sophisticated rendering architecture which is designed to be platform-agnostic, rather than rely on operating-system-specific details such as the OS compositing architecture. As such, complex behaviour like scrolling is implemented nearly identically across all non-iOS Chrome platforms including Windows, MacOS, Android, Linux and ChromeOS. Using an operating-system provided scrolling / compositing architecture is infeasible for Chromium both because of the huge engineering expense and also because it would result in a worse product in some ways."*

Rick Byers - Web Platform Area Tech Lead, Chrome - Open Web Competition Platform

The second, is that it is in the implementation of these very user interface libraries that a not insignificant proportion of browser competition lies.

Scrolling is a good example. Apple's implementation for scroll on iOS Safari has been riddled with bugs for any Web App that is attempting more 'app-like' layouts, that is ones where part of the page scrolls vertically rather than the entire page scrolling vertically.

> *"For more than 6 years, one of the biggest issues Web Developers have had to deal with in Safari and other WebKit-based browsers on iOS (and in a lesser extent on macOS), is the inability to reliably block body scroll, or, said in other words, to remove the ability to scroll the current page in certain situations."*

OWA - Proposed Interop Issue

Apple appears unable or unwilling to fix these bugs, despite identical functionality working in almost every other browser for years. This has likely been prolonged by the complete lack of browser competition on iOS to apply pressure to fix these bugs or risk losing market share.

Finally, Apple is only allowed to apply security rules in their API contract under Article 6(7), but even if they were to move these rules to their app store guidelines, they should still be struck out under Article 5(7), and Article 13(4) for preventing browsers from porting significant and key parts of their engines.

It is essential that Apple does not constrain browsers from competing and bringing their full rendering stack (the rendering stack is the collection of libraries responsible for converting html, css and JavaScript into the displaying and updating the actual pixels on the screen inside the area controlled by the browser). Browser vendors should have the right, but not the obligation, to use these iOS provided user interface libraries and components as appropriate to their needs.

This does not require Apple to make any technical changes. They simply have to update their contracts and associated documentation. Apple may state they do not intend to enforce these rules in their contract, but this is a misleading stance. Apple should not assert it has rights that are prohibited by the DMA.

3.1.11. Privacy

Apple is only allowed to apply strictly necessary, proportionate and duly justified for *"protecting the integrity of the operating system, software and hardware of the gatekeeper"* on API access.

While improving privacy is a laudable goal, we have three concerns with Apple's privacy section in their browser entitlement contract:

1. The browser entitlement contract is the wrong place to include these clauses as the DMA requires this to be limited to strictly necessary, proportionate and justified security related rules. All of these rules, at a minimum, will need to be moved to the app store guidelines.

2. We are concerned that Apple may have weak or unenforced privacy rules for both its own apps and apps distributed by its app store - But extremely strong rules for browsers and Web Apps. This will create pressure for apps that are based on advertising to be distributed via Apple's app store instead of the open web, as the former offers expanded access and weaker privacy protection for consumers.

3. Any privacy rule in the app store guidelines must not, in effect, ban any of the existing major browsers from being ported, and thus contravene Article 5(7). Currently this would likely only impact Chrome. Chrome is in the process of phasing out third-party cookies but has halted this while the UK's Competition and Market Authority reviews this change. The CMA is concerned that the phase out of third-party cookies in Chrome and replacement with privacy sandbox will reduce competition in advertising. Apple must not apply any rule, that combined with this fact, would de facto ban Google from porting its browser to iOS. While as an organization we can see the privacy benefits of the phase out of third-party cookies and support it, it can not be used as a tool to block competitors from iOS. We would also be concerned at any attempt to fragment the functionality of browsers from jurisdiction to jurisdiction.

"The CMA clarified that raising these concerns does not imply a belief that the Privacy Sandbox changes cannot proceed. However, it signals the necessity for further resolutions before moving forward."
UK's CMA Raises Concerns, May Delay Google's Third-Party Cookie Phase-Out

"The CMA's preliminary view is that Google is likely to have been aware that these announcements, including the setting of a two-year deadline for deprecating TPCs, would adversely affect market participants and reduce competition. For example, studies cited by Google in the announcement of 22 August 2019 suggested that when advertising is made less relevant by removing TPCs, funding for publishers falls by 52% on average."
CMA on Privacy Sandbox

We also note that Apple intends to apply privacy rules to apps that are not distributed by their app store but are signed/notarized by them. OWA considers Apple's proposed notarization system for third party apps to be App Store Review, in all but name. Again while improving privacy is a laudable goal, we are concerned at the power this allows Apple to indiscriminately wield over apps that are not delivered though their app store. This is strictly prohibited by the DMA as the security carve outs in both 6(4) and 6(7) are both fairly narrowly scooped.

The appropriate way for this issue to be dealt with is not to place this power in Apple's hands. Apple has a long and proven history of bending its own rules to act in its own financial interests even when that weakens or harms user privacy. Apple has also been shown to have dubious policies when it comes to their own employees' own privacy. Apple is not a trusted arbiter of privacy.

"In the end, Apple deploys privacy and security justifications as an elastic shield that can stretch or contract to serve Apple's financial and business interests."
DOJ Complaint against Apple

"Apple has a tactical commitment to your privacy, not a moral one. When it comes down to guarding your privacy or losing access to Chinese markets and manufacturing, your privacy is jettisoned without a second thought."
Cory Doctorow - Former European director of the Electronic Frontier Foundation

"Tim Cook, Apple's chief executive, has said the data is safe. But at the data center in Guiyang, which Apple hoped would be completed by next month, and another in the Inner Mongolia region, ***Apple has largely ceded control to the Chinese government.***

Chinese state employees physically manage the computers. Apple abandoned the encryption technology it used elsewhere after China would not allow it. And the digital keys that unlock information on those computers are stored in the data centers they're meant to secure.

[...]

Apple has tried to isolate the Chinese servers from the rest of its iCloud network, according to the documents. The Chinese network would be "established, managed, and monitored separately from all other networks, with no means of traversing to other networks out of country." ***Two Apple engineers said the measure was to prevent security breaches in China from spreading to the rest of Apple's data centers****.*

Apple said that it sequestered the Chinese data centers because they are, in effect, owned by the Chinese government*, and Apple keeps all third parties disconnected from its internal network."*

New York Times - Censorship, Surveillance and Profits: A Hard Bargain for Apple in China
(emphasis added)

"Apple's marketing insists that when it comes to privacy the company is ethical and different, but when it comes to the people actually tasked with building the products that adhere to those standards — these values suddenly and mysteriously disappear."

Karl Bode - Techdirt

*"****Privacy is a fundamental human right.*** *It's also one of our core values. Which is why we design our products and services to protect it. That's the kind of innovation we believe in."*

Apple on Privacy
(emphasis added)

To solve the privacy issues, we would propose three solutions:

1. Stronger privacy laws for the EU that would equally apply to native apps, Web Apps, websites and Apple's own apps.

2. Apple taking steps to actually enforce a number of their privacy rules and abiding by them for their own apps. They could then advertise the privacy advantages of their app store over competitors or direct downloads.

3. Apple should not use any word games to exclude themselves from their own privacy rules. For example Apple often refers to itself as a first party and displays different softer warnings or no warnings for similar behavior when Apple is the one collecting users data for the purposes of advertising. While a small part of its overall business, advertising is a significant and growing chunk of Apple's revenue coming in at $3.7 billion directly in 2021 and an additional $18 billion in 2021 via its search deal with Google.

For example in this document they state that for their app store, they use users' personal and identifiable data for the purposes of showing adverts. Users can opt-out via a deeply nested menu option via "Settings > [your name] > Media & Purchases > View Account, and then tap to turn off Personalized Recommendations":

> *"**We collect your personal data** so that we can provide you the content you purchase, download, or want to update in the App Store and other Apple online stores, including iTunes Store and Apple Books.*
>
> ***We also use information about your account, purchases, and downloads in the stores to offer advertising** to ensure that ads on the App Store, Apple News, and Stocks, where available, are relevant to you. You have choices with respect to this advertising, as described below."*

Apple App Store Data Policy

This appears to be potentially in breach of the DMA's data rules that all gatekeepers must abide by. Specifically this Article 5(2) (b):

> *2. The gatekeeper shall not do any of the following:*
> *...*
> *(b)* ***combine personal data from the relevant core platform service with personal data from*** *any further core platform services or* ***from any other services provided by the gatekeeper*** *or with personal data from third-party services;*
> *...*
> ***unless the end user has been presented with the specific choice and has given consent*** *within the meaning of Article 4, point (11), and Article 7 of Regulation (EU) 2016/679.*
>
> Digital Markets Act - Article 5(2)
> (emphasis added)

In this case Apple is combining data from its app store (a core platform service of Apple) with other services of Apple including Apple News and Apple Stocks in order to perform targeted advertising. Whether this is in breach would likely depend on whether users were prompted and the nature of that consent prompt. Prior to iOS 15 users were not prompted and were automatically opted into targeted advertising for Apple's own apps.

Apple's behavior related to targeted advertising on iOS is already under investigation by the French Competition Authority. They are investigating whether Apple is self preferencing by categorizing its behavior as it relates to advertising as special and thus under different rules to other third parties on iOS.

Apple's carefully chosen definition of "tracking" excludes its own behaviors from its privacy rules as they relate to targeted advertising. As Apple is both the operating system and the publisher of multiple Apple default apps, the app store is able to directly show targeted adverts without providing the applied data to a third-party. We believe this is an intentional redefinition of "tracking" to allow Apple to craft rules to allow its own use of targeted advertising whilst disallowing the same behavior in third parties.

> *"**Apple does not consider the processing activities it undertakes in terms of personalised advertising (ie use of first-party data from different Apple apps and services) as 'tracking'**, particularly as it does not link information collected by apps from different companies, and therefore its apps are not required to show the ATT prompt. This is factually correct, given that, as detailed above, Apple uses data it collects from its own services which it operates under a single corporate ownership for personalised advertising purposes.*
>
> *6.177 However, as further discussed in Appendix J, based on our consideration of*

> *the ICO's definition of online tracking, which does not distinguish between first-party and third-party data, we consider* ***Apple's own use of its users' personal data no less consistent with this description of 'tracking' than that of third-party developers.****"*
>
> CMA Mobile Ecosystems Report
> (emphasis added)

3.1.11.1. Sharing of State

> *"Not sync cookies and state between the browser and any other apps, even other apps of the developer"*
>
> Web Browser Engine Entitlement Addendum for Apps in the EU

This line is problematic as it prohibits functionality that browsers already provide to users. For example, many browsers (including Safari) sync state between a user's mobile browser and their desktop browser. A lack of sync may prohibit other user-facing functionality, such as password autofill and tab syncing.

As discussed in the previous section, Apple may only have strictly necessary, proportionate and justified security rules in their browser engine API contract, thus this rule would need to be moved to the app store rules. If Apple chooses to introduce this as an app store rule they would need to be more specific and be cautious so as not to block browser vendors from providing useful functionality to users, otherwise they risk violating Article 5(7).

3.1.12. Confidentiality Clause

Apple has long had a culture of secrecy. This includes inserting confidentiality clauses into both its agreements with other companies and with its staff.

Its initial developer contract contained clauses banning companies publicly complaining about being rejected from the app store. Rejection letters include a clause that read:

> *"THE INFORMATION CONTAINED IN THIS MESSAGE IS UNDER NON-DISCLOSURE."*
>
> Apple Rejection Email

They still have a similar clause in their current developer contract:

> *"You may not issue any press releases or make any other public statements regarding this Agreement, its terms and conditions, or the relationship of the parties without Apple's express prior written approval, which may be withheld at Apple's discretion."*
>
> Apple Developer Program License Agreement

Astonishingly, in Apple's developer contract they state that anything the developer submits to Apple is not confidential including if Apple wishes to publish it, use it to develop competing products or to share this information with Apple's partners and the app developers competitors. Given app store review is essentially a condition of being on iOS, currently this seems a prime example of the *"serious imbalances in bargaining power and, consequently, to unfair practices and conditions for business users"* that recital 4 of the DMA discusses:

> *"Apple works with many application and software developers and some of their products may be similar to or compete with Your Applications.* ***Apple may also be developing its own similar or competing applications and products or may decide to do so in the future. To avoid potential misunderstandings*** *and except as otherwise expressly set forth herein, Apple cannot agree, and expressly disclaims, any confidentiality obligations or use restrictions, express or implied, with respect to any information that You may provide in connection with this Agreement or the Program, including but not limited to information about Your Application, Licensed Application Information, and metadata (such disclosures will be referred to as "Licensee Disclosures").* ***You agree that any such Licensee Disclosures will be non-confidential. Except as otherwise expressly set forth herein, Apple will be free to use and disclose any Licensee Disclosures on an unrestricted basis without notifying or compensating You. You release Apple from all liability and obligations that may arise from the receipt, review, use, or disclosure of any portion of any Licensee Disclosures. Any physical materials You submit to Apple will become***

> ***Apple property and Apple will have no obligation to return those materials to You or to certify their destruction.***"
>
> [Apple Developer Program License Agreement]

Apple has inserted the following clause into its browser engine entitlement contract:

> *Confidentiality* ***You agree that any non-public information regarding the Alternative Web Browser Engine APIs or Alternative Web Browser Engine (EU) Entitlement Profile shall be considered and treated as "Apple Confidential Information"*** *in accordance with the terms of Section 9 (Confidentiality) of the Developer Agreement. You agree to use such Apple Confidential Information solely for the purpose of exercising Your rights and performing Your obligations under this Addendum and agree not to use such Apple Confidential Information for any other purpose, for Your own or any third party's benefit, without Apple's prior written consent.* ***You further agree not to disclose or disseminate Apple Confidential Information to anyone other than those of Your employees or contractors who have a need to know and who are bound by a written agreement that prohibited unauthorized use or disclose of the Apple Confidential Information.***
>
> [Apple Browser Engine Entitlement Document]
> (emphasis added)

Besides clearly not being a security measure, this clause directly violates the DMA in another way, In Recital 42 the DMA states:

> *"****Any practice that would in any way inhibit or hinder those users in raising their concerns*** *or in seeking available redress,* ***for instance by means of confidentiality clauses in agreements*** *or other written terms, should therefore be prohibited. This prohibition should be without prejudice to the right of business users and gatekeepers to lay down in their agreements the terms of use including the use of lawful complaints-handling mechanisms, including any use of alternative dispute resolution mechanisms or of the jurisdiction of specific courts in compliance with respective Union and national law. This should also be without prejudice to the role gatekeepers play in the fight against illegal content online."*
>
> [Digital Markets Act - Recital 42]

The EU courts, the Commission and other EU national authorities do not work for the browser vendors, nor are they required to sign any such non-disclosure agreement (NDA). This contract explicitly states that such sharing of any non-public interactions with courts and regulators in the EU would be a violation of the contract.

Apple should not attempt to grant itself confidentiality rights that it is already explicitly prohibited from granting by the DMA.

This combination of an unreasonable and blanket ban on criticizing Apple's app review process and a clause that Apple can use confidential information submitted to them for the purposes of app store review in any way they see fit, including to compete against the very same app, again points to an imbalance in power between the Apple the gatekeeper and the business users.

This clause will, at a minimum, need to be updated with a sentence that explicitly explains that browser vendors are permitted to share such information about their interactions with Apple, to regulators and other government enforcement bodies in the EU.

We would go further and argue it is unreasonable of Apple to hide malicious compliance of the DMA from the broader public via such confidentiality clauses and to obligate business users to release confidential information to Apple nominally for app store review, that Apple can then use for any purpose.

Should browser vendors wish to talk to the press about Apple (for example) rejecting their browser for frivolous reasons, then they should be able to do so. Blocking public pressure, which facilitates the proper functioning of the DMA via NDAs or confidentiality agreements, should be prohibited.

Preventing individuals and companies from talking to regulators and courts via such illegal NDAs is not hypothetical. In this example Apple subcontractor Thomas Le Bonniec was frightened by his non-disclosure agreement and initially did not come forward with allegations about Siri's privacy practices.

> *"It was scary. I thought I may end up broke for the rest of my life,"*
>
> Apple subcontractor Thomas Le Bonniec

> *"His contract threatened unspecified 'monetary damages' if he shared confidential information 'outside of work' — with no exceptions, according to his complaint.*
>
> *As a dedicated Siri data analyst, Le Bonniec said he heard conversations he believed violated users' privacy, including details about sexual preferences, bank account numbers and health problems. He said that the importance of exposing the existence of the recordings outweighed the risks of breaking his agreement, so he went public. After an outcry by consumers, Apple made changes to Siri to address privacy concerns."*
>
> Sabrina Willmer, Austin Weinstein and Bloomberg -
> Tech firms are using NDAs to illegally muzzle whistleblowers

This clause needs to be removed from the browser engine entitlement contract. Any similar clause in the app store rules should include language making it explicit that complaints to the Commission, EU courts and other EU legal bodies are permitted. In our opinion, it is not enough to simply state elsewhere that they will not enforce such action. The contracts themselves need to be compliant with the DMA. Further, if possible under any section of the DMA or any other applicable EU law, Apple's confidentiality clauses should be scrutinized to make sure they are necessary, proportionate, fair and reasonable.

3.1.13. Apple can Change or Break any API with No Notice

In their browser engine entitlement contract Apple states that they can change or break any API with no notice.

> *"**Apple may at any time, and from time to time, with or without prior notice to You, modify, remove**, or reissue the Apple Materials or the **Alternative Web Browser Engine APIs, or any part thereof. You understand that any such modifications may require You to change or update Your Alternative Web Browser Engine App (EU) at Your own cost and that features and functionality of such App may cease to function.** Except as required by applicable law, Apple has no express or implied obligation to provide, or continue to provide, the Apple Materials or Alternative Web Browser Engine APIs, and may suspend or discontinue all or any portion of Your access to them at any time"*
>
> Apple Browser Engine Entitlement Document
> (emphasis added)

This is clearly unreasonable, as the browser vendors need time to update their browser in order to make sure it continues to work despite the API change or removal.

Next, Apple is obligated to make sure any software or hardware feature that is used, or available on the system, is provided to browser vendors free of charge subject to narrow scope security measures under Article 6(7).

Apple is also not allowed under Article 5(7) or Article 13(4) to remove or alter any API in such a way that would undermine the effectiveness of Article 5(7) in allowing browser vendors to port their engines to iOS.

Finally, under Article 13(6) of the DMA, Apple is not allowed to degrade the conditions or quality of any of the core platform services provided to business users or end users who avail themselves of the rights or choices laid down in Articles 5, 6 and 7 in particular Article 5(7) and Article 6(7).

3.1.14. Apple can Withhold any API from any Browser Vendor

In the same paragraph in Apple's browser engine entitlement contract Apple states:

> *"Except as required by applicable law, Apple has no express or implied obligation to provide, or continue to provide, the Apple Materials or Alternative Web Browser Engine APIs, and may suspend or discontinue all or any portion of Your access to them at any time"*
>
> Apple Browser Engine Entitlement Document

That is, Apple can suspend any API from any individual browser vendor. Apple does not state they need to provide any evidence or rationale for such a suspension.

Given that this contract is in essence exclusively for the EU, and Apple are even calling the app "Alternative Web Browser Engine (EU)", and the DMA causes Apple to have an express obligation to provide and continue to provide these APIs to browser vendors subject to narrow scope security measures, Apple should not be attempting to override the DMA by contract.

As such this paragraph is explicitly inconsistent with the DMA and should be removed.

3.1.15. Apple can Remove any Browser from its App Store at its Sole Discretion

> *"Nothing herein shall imply that Apple will accept Your Alternative Web Browser Engine App (EU) for distribution on the App Store, and You acknowledge and agree that Apple may, in its sole discretion, reject, or cease distributing Your Alternative Web Browser Engine App (EU) for any reason. For clarity, once Your Alternative Web Browser Engine App (EU) has been selected for distribution via the App Store it will be considered a "Licensed Application" under the Developer Agreement"*
>
> Apple Browser Engine Entitlement Document

Apple in this clause states that, at its sole discretion, it can reject or remove a browser vendor's app from the app store for any reason.

Again this is clearly not permissible under the DMA. Apple is required to operate its app store with FRAND (Fair, reasonable and non-discriminatory) conditions under Article 6(12). Further, it is not allowed to restrict browser vendors from using their own engines under Article 5(7).

Any attempt to construct app store rules to prevent browsers from using their own engine in order to provide improved features, stability, security or privacy relative to Safari would likely be in violation of Article 13(4) for undermining Article 5(7).

That is, rejecting a browser from Apple's app store would not be at its sole discretion and it could not do it for any reason. Apple should not try to override the DMA via its contract and this paragraph should be modified to be compliant with the text of the DMA.

3.1.16. Apple Developer Agreement

While, in the short term, all browser vendors will likely want to be on Apple's app store, it is not clear if this will remain the case. We would expect that over the long term many browser vendors will switch to direct distribution and no longer provide their apps via the Apple's app store.

Many developers do not distribute on Apple's app store for macOS, as a point of comparison.

> *"Neither is on the store because they don't have to be. They can be on Mac and distribute to users without sharing the revenue with us"*
>
> Philip Schiller - Apple Upper Management - On the Mac App Store

Mere control of the app store review process, while of dubious and likely negative value in improving browser security, does allow Apple to exert great control by holding up or rejecting app store updates. This exact concern is highlighted in the DOJs case against Apple. This means that the prospect of having users download signed apps directly from the browser vendors website using an automatic notarization system, similar to macOS, is likely quite appealing for browser vendors.

> *"Apple often enforces its App Store rules arbitrarily. And it frequently uses App Store rules and restrictions to penalize and restrict developers that take advantage of technologies that threaten to disrupt, disintermediate, compete with, or erode Apple's monopoly power."*
>
> DOJ - Case 2:24-cv-04055

For this reason it is quite concerning that Apple has chosen to make the contract for browser engine APIs an amendment to the "Apple Developer Program License Agreement".

In the opening paragraphs it states that:

> *"Applications developed under this Agreement for iOS, iPadOS, macOS, tvOS, visionOS, and watchOS can be distributed: (1) through the App Store, if selected by Apple, (2) on a limited basis for use on Registered Devices (as defined below), and (3) for beta testing through TestFlight. Applications developed for iOS, iPadOS,*

> *macOS, and tvOS can additionally be distributed through Custom App Distribution, if selected by Apple. Applications developed for macOS can additionally be separately distributed as described in this Agreement."*
>
> Apple Developer Program License Agreement

Importantly, the only apps that can be distributed using this agreement outside Apple's app store or Apple's app store guidelines are applications for macOS:

> *"Applications developed for macOS can additionally be separately distributed as described in this Agreement."*

The agreement then has an entire section describing the rules for having your app notarized for macOS.

> *"Apple may revoke Tickets at any time in its sole discretion in the event that Apple has reason to believe, or has reasonable suspicions, that Your Application contains malware or malicious, suspicious or harmful code or components or that Your developer identity signature has been compromised."*
>
> Apple - Notarization for macOS

It is worth taking note that even in the instance where Apple has found that your app contains malware, this clause has lighter implications and is less broad than the one in Apple's contract for browser engine API access for potentially far more minor or dubious infractions.

While this clause does have the phrase *"in its sole discretion"* it does at least make clear the nature of the infringement: That Apple would have reason to believe that app contains malicious code and that the penalty would simply be having that specific app unnotarized.

Apple is required by the act to allow browser vendors to distribute directly and will only be subject to the strictly necessary, proportionate and justified security measures in 6(4) and 6(7). Thus, either the browser engine entitlement contract should be separate or the Apple Developer Program License Agreement should be updated with language making it clear that apps can be distributed directly and that any rules in the Apple Developer Program License Agreement or associated documents do not apply to them. Instead, only a clearly listed set of security rules would apply.

3.2. Choice Screens

> *"**A gatekeeper can use different means to favour its own or third-party services or products on its operating system**, virtual assistant or web browser, **to the detriment of the same or similar services that end users could obtain through other third parties**. This can for instance happen where certain software applications or services are pre-installed by a gatekeeper. To enable end user choice, gatekeepers should not prevent end users from un-installing any software applications on their operating system. It should be possible for the gatekeeper to restrict such un-installation only when such software applications are essential to the functioning of the operating system or the device. Gatekeepers should also allow end users to easily change the default settings on the operating system, virtual assistant and web browser **when those default settings favour their own software applications and services. This includes prompting a choice screen**, at the moment of the users' first use of an online search engine, virtual assistant or web browser of the gatekeeper listed in the designation decision, allowing end users to select an alternative default service when the operating system of the gatekeeper directs end users to those online search engine, virtual assistant or web browser and when the virtual assistant or the web browser of the gatekeeper direct the user to the online search engine listed in the designation decision."*
>
> Digital Markets Act - Recital 49
> (emphasis added)

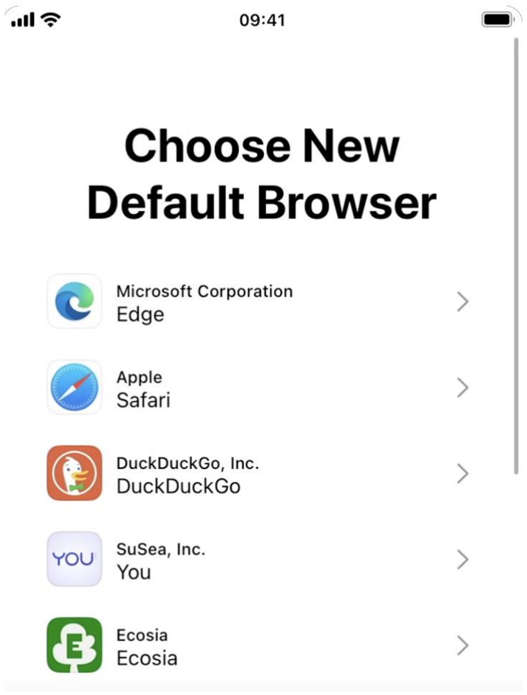
The act explains that gatekeepers of operating systems can favor their own products by pre-installing them and setting them as the system default. One of the methods used by the act to help combat this self-preferencing is a choice screen for browsers on operating systems where the browser of the gatekeeper has been set as the default.

No choice screen will ever completely nullify the advantage granted to the gatekeeper by setting the default browser. In order to have maximum impact in minimizing the effect of this self-preferencing, to increase user autonomy and to not introduce new self-preferencing via the choice screen itself, the choice screen must be carefully designed. Apple's choice screen, while much improved from initial proposals, still has significant issues.

While this document focuses only on Apple, in most cases we recommend equivalent remedies for Google's Android.

3.2.1. Browsers should Be Allowed To Show More Information on Choice Screen

Currently on iOS the amount of information that browser vendors can show on the choice screen is extremely limited. It is limited to the company's name, browser name and browser logo.



Apple's Browser Choice Screen

One of the aims of the act with the choice screen is to allow users to make an informed choice and to limit the degree to which operating system gatekeepers can self-preference their own browser by setting it as the default.

With this in mind, it seems entirely appropriate that Apple allow browser vendors to display a sentence explaining why users should choose their browser. The layout should be updated to allow the full sentence to be displayed without clicking on any buttons, a more reasonable character limit might be 150 characters (about 30 words).

To do otherwise concentrates power in the hands of major browsers and suppresses smaller entrants from the mobile browser market.

3.2.2. Shouldn't be Locked to the Gatekeepers App Store

> *"**To prevent further reinforcing their dependence on the core platform services of gatekeepers, and in order to promote multi-homing, the business users of those gatekeepers should be free to promote and choose the distribution channel that they consider most appropriate** for the purpose of interacting with any end users that those business users have already acquired through core platform services provided by the gatekeeper or through other channels."*
>
> Digital Markets Act - Recital 49
> (emphasis added)

Multi-homing is where an end user or business user uses multiple core-platform services. For example owning an Android phone and an iOS iPad. It could also be having multiple apps on a phone from different distributors such as different app stores or directly from the business user (i.e. direct download / distribution via the web).

The act specifically highlights direct distribution, in this case downloading and updating directly from the business users own website, as something business users should have the option of choosing for their users. Should a browser vendor wish to distribute to their users directly via their own website (as opposed to Apple's app store), they should be able to do so.

Choice screens aim to nullify the way that operating system gatekeepers self-preference themselves via setting defaults on the operating system. This remedy should not then reinforce the dependence on the core platform services of gatekeepers, in particular their app store.

There are a number of concerns here:

1. Apple is only counting browsers downloaded from their app stores in the calculation of the 12 most popular browsers. Browsers that are downloaded directly or from another app store need to be included in this count.

2. Browsers that are not available on the gatekeepers app store at all still need to be included.

3. Browser vendors should not be locked to the gatekeepers app store and should be able to opt into distributing directly to users. Given that all of Apple's security checks are built into their browser entitlement contract and notarization process, there is absolutely no plausible reason to introduce scare screens, and doing so would, in our opinion, be a violation of the act.

To allow Apple to impose their app store as the sole gateway for downloading browsers via the choice screen or even the default is against the intent of the act.

These changes are critical to browser competition on mobile operating systems. In the same manner that it would have been ridiculous for Microsoft to tie their choice screen to only browsers on a Microsoft store from which they set the rules and get a 30% cut, it is unreasonable for Apple to tie their choice screen to their app stores along with their rules and their up to 30% cut.

Without allowing browser vendors chosen via the choice screen to be directly downloaded from the browser vendor themselves, and instead forcing them to use the Apple's app store, will continue to damage browser competition including:

- Forcing browser vendors to share 30% of all revenue made via in-app purchases, which would effectively kill various business models or significantly reduce browser revenue (i.e. for purchases of VPNs, or subscriptions to AI assistants).
- Forcing all browser vendors to accept the gatekeeper's app store rules and conditions, which are well documented to be unfair and hamper competition.
- Introduces delays and complexity into planning and releases.

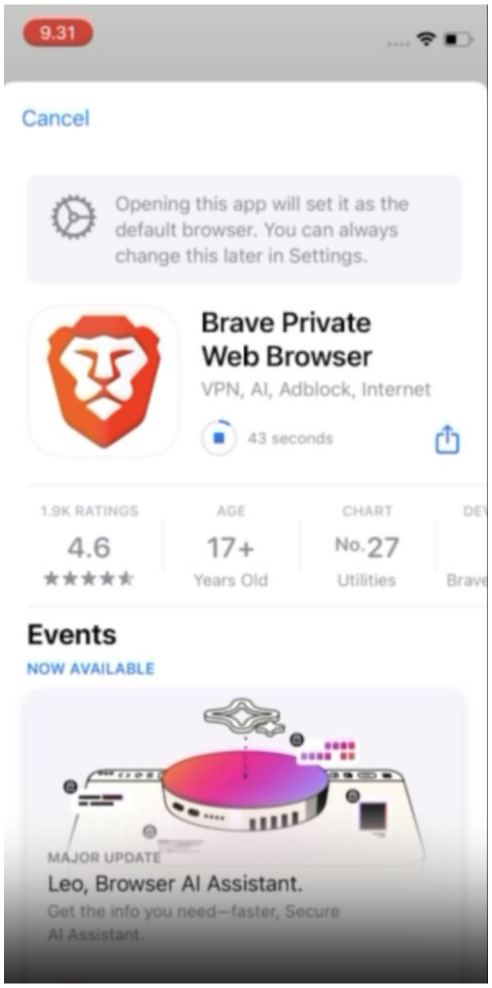
3.2.3. Should be a Background and Direct Download

As discussed in the previous section, taking the user to the app store is an inappropriate flow, it also introduces friction relative to the gatekeepers browser which is preinstalled.

Instead, the user should simply pick their new default browser from the choice screen and the operating system should download and install the browser in the background. The choice screen should allow the browser vendor to show relevant information about their browser, including pictures and videos that the user can see by clicking on a "see more" arrow icon.

This design works far better when different browsers are coming from different distribution channels such as directly from the browser vendors website. Both Google and Apple's design currently conflict with this requirement and will need to be reworked.

Next, Apple appears to have recently added a time to download countdown to the Apple app store page.


9.31
Cancel
Opening this app will set it as the default browser. You can always change this later in Settings.
Brave Private Web Browser
VPN, AI, Adblock, Internet
43 seconds
1.9K RATINGS
4.6
AGE
17+
Years Old
CHART
No.27
Utilities
Events
NOW AVAILABLE
MAJOR UPDATE
Leo, Browser AI Assistant.
Get the info you need—faster, Secure AI Assistant.

The Choice Screen when Brave has been selected and the download symbol has been pressed. Note the "cancel" button at the top and the install time widget that Apple has added.

We believe that this, in combination with the prominent cancel button, is an attempt by Apple to use the fact they are preinstalled to undermine user autonomy and push users towards Safari.

Instead of having a countdown, we believe that Apple should simply set that browser as the default browser immediately, continue to the homescreen, and replace Safari on the hotseat with the user selected browser. It will be clear to the user that the browser is downloading, as it will have the standard downloading circle on the hotseat. We believe after this process has been started, it should continue until it has finished. Providing an additional cancel option undermines the choice screen.

3.2.4. Browser Vendors Need Data on the Choice Screen to Measure Performance

Third party browser vendors need access to the number of times their browser was shown and the number of times their browser was picked on the choice screen by end users. This data can be aggregated per country per day. Additionally it would be useful to include additional information, such as the position in the choice screen the browser was shown.

For example one piece of aggregated data a browser vendor might receive could be:

> Country: Germany
> Date: 2024-07-01
> Browser: Firefox
> Times Shown: 1451
> Times Selected: 435

i.e. 1451 people were shown Firefox across all positions in Germany on 2024-07-01, out of them 435 chose to make Firefox the default browser.

For the more detailed one that includes position an example would be:

> Country: Germany
> Date: 2024-07-01
> Browser: Firefox
> Times Shown: 77
> Times Selected: 5
> Position: 12

i.e. 77 people were shown Firefox in the 12th position in Germany on 2024-07-01, out of them 5 chose to make Firefox the default browser.

This is important to give browser vendors the chance to work out what works in advertising and adding features to their browser, as well as providing data that the remedy is effective.

Given the data is aggregated and boils down to two numbers per country per day (or 24 for the more detailed one), there is no plausible privacy or security reason not to collect or share this data. The only party that benefits from this data not being available is the operating system gatekeeper.

The operating system gatekeeper benefits in two ways. First, third party browser vendors can not work out what blurb, advertising or features work in encouraging users to switch browsers. Second, the relative effectiveness of the choice screen is obscured.

Given this data is aggregated and non-confidential in nature, gatekeepers should not be able to prevent browser vendors from publishing it if they choose to.

Finally, whether or not a remedy is effective is clearly important to the DMA. For example in the event a gatekeeper is believed to have frustrated the objectives of an article, a proceeding can be opened. The gatekeeper can then be obligated to fix the behavior via commitments or be fined. In the event those commitments are not effective the proceeding can be reopened.

> *2. The Commission may, upon request or on its own initiative, reopen by decision the relevant proceedings, where:*
> *(a) there has been a material change in any of the facts on which the decision was based;*
> *(b) the gatekeeper concerned acts contrary to its commitments;*
> *(c) the decision was based on incomplete, incorrect or misleading information provided by the parties;*
> ***(d) the commitments are not effective.***
>
> Digital Markets Act - Article 25(2)
> (emphasis added)

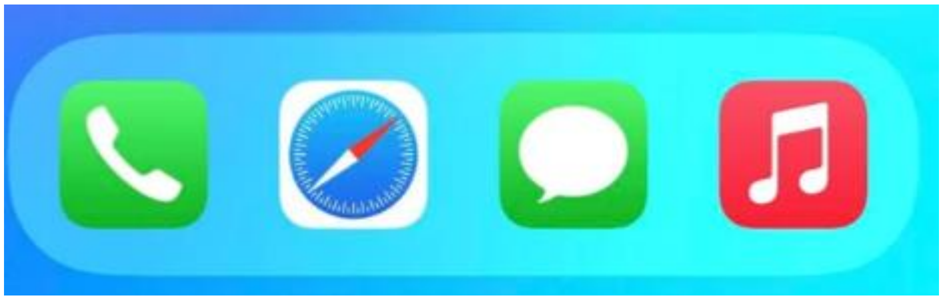
The objective of Article 6(3) as outlined in Recital 49 is to nullify the self-prefencing advantage gatekeepers grant their own apps via setting them as the default by default.

> ***A gatekeeper can use different means to favour its own or third-party services or products on its operating system, virtual assistant or web browser, to the detriment of the same or similar services that end users could obtain through other third parties.*** *This can for instance happen where certain software applications or services are pre-installed by a gatekeeper. To enable end user choice, gatekeepers should not prevent end users from un-installing any software applications on their operating system. It should be possible for the gatekeeper to restrict such un-installation only when such software applications are essential to the functioning of the operating system or the device.* ***Gatekeepers should also allow end users to easily change the default settings on the operating system, virtual assistant and web browser when those default settings favour their own software applications and services. This includes prompting a choice screen****, at the moment of the users' first use of an online search engine, virtual assistant or web browser of the gatekeeper listed in the designation decision, allowing end users to select an alternative default service when the operating system of the gatekeeper*

> *directs end users to those online search engine, virtual assistant or web browser and when the virtual assistant or the web browser of the gatekeeper direct the user to the online search engine listed in the designation decision.*
>
> Digital Markets Act - Recital 49
> (emphasis added)

As such it is of critical importance that data to measure the effectiveness of Apple's compliance with the remedy is both collected and shared with relevant parties.

3.2.5. Apple's Dark Pattern Exacerbated by Keeping Hotseat



The hotseat on iOS's homescreen

The **hotseat** is any of that collection of app locations on the dock on the base of iOS homescreen. They have the advantage of always being shown regardless of which homescreen the user is on. Apple has set Safari to be in the hotseat by default (along with phone, messages and Apple Music), while users can change this setting manually by dragging the item out and another in, Apple has added significant friction by not making this automatic.

> *"Only about half (52%) of people understand that their default browser is opened when they, for example, click on a link in an email or document.*
> *[..]*
> *over half (53%) also erroneously believed that their default browser would automatically be pinned to their task-bar."*
>
> Mozilla - "Can browser choice screens be effective?" paper

Being selected as the default browser does not grant the hotseat. This means it is entirely possible, and in fact likely that many existing users who have set a third party browser as their default browser will still have Safari or Chrome on certain versions of Android in the hotseat.

While this is of concern on both Android and iOS, due to the undermining of the meaning of default browser, an additional concern is introduced on iOS due to how Apple has implemented their choice screen.

The DMA states this on choice screens:

> *"This includes* ***prompting a choice screen****,* ***at the moment of the users' first use of an*** *online search engine, virtual assistant or* ***web browser of the gatekeeper****"*
>
> Digital Markets Act - Recital 49
>
> (emphasis added)

Apple has interpreted the DMA very directly here and will only show the choice screen upon Safari being clicked, unlike Google which does it immediately upon system software update.

In Apple's design, Safari's interface actually loads and then the choice screen is a full screen overlay over Safari. This has the psychological effect of making the choice screen seem more like an interruption rather than part of the setup of the device. While this was more problematic in Apple's initial design which included a giant skip button (labeled "not now") at the bottom (which would essentially choose Safari), this still seems to be a nudge to users to quickly choose a browser and get back to their initial task.

> *"People are significantly more likely to choose a pre-installed browser as their default when the choice screen is shown at first use of the browser [...] from 11% to 19%"*
>
> Mozilla - "Can browser choice screens be effective?" paper

Apple's design then creates the problematic scenario where the choice screen will be shown even if Safari is not the default browser. Apple is clearly aware of this based on the careful phrasing of their speech at the workshop:

> *"In terms of what's being required here by the DMA our focus is on* ***presenting this choice*** *even more clearly to consumers* ***the first time you use Safari****. Obviously there are not choices being presented* ***when you use other browsers on iOS****. So we are complying* ***from our perspective with the spirit here.****"*
>
> Apple representative at DMA Workshop

Showing the choice screen when Safari is not the default is clearly against the intent of the act and Apple needs to update the choice screen to only show if Safari is the default browser. The default browser not automatically granting the hotseat, significantly compounds and re-enforces this problem. While it is possible for users to manually change this, this style of friction significantly undermines browser competition. Operating system gatekeepers know this and this is precisely why they are so resistant to such a change.

It should be assumed that because Apple has displayed the choice screen when third party browsers were already set as default, that they would have already gained users for Safari that they would otherwise not have.

A reasonable remedy here would be to re-run the choice screen for all users that picked Safari upon such an inappropriate choice screen being shown. If Apple does not store that information, it can simply re-run the choice screen for all Safari users.

3.2.6. Periodic Choice Screens

The act aims to nullify the power operating system gatekeepers grant themselves by setting the defaults on the operating system. In particular, the "default browser" to their own and placing their browser in a prominent location by default.

A one-off choice screen may have an impact, perhaps even a significant one, but that impact will fade and will not achieve this goal of the act. It therefore seems appropriate that the choice screen is re-shown to users at an appropriate interval.

We believe then, that the choice screen should be shown whenever users encounter a new reinstallation event, such as purchasing a new phone, restoring from an old phone or a factory reset.

This allows the choice screen to have a lasting impact. While it will not remove the immense self privilege of the defaults the operating system gatekeeper sets, it will at least make a significant dent. Critical mass is important for browsers as it strongly affects website developers’ interest in supporting them and dictates their funding, which dictates their development budget.

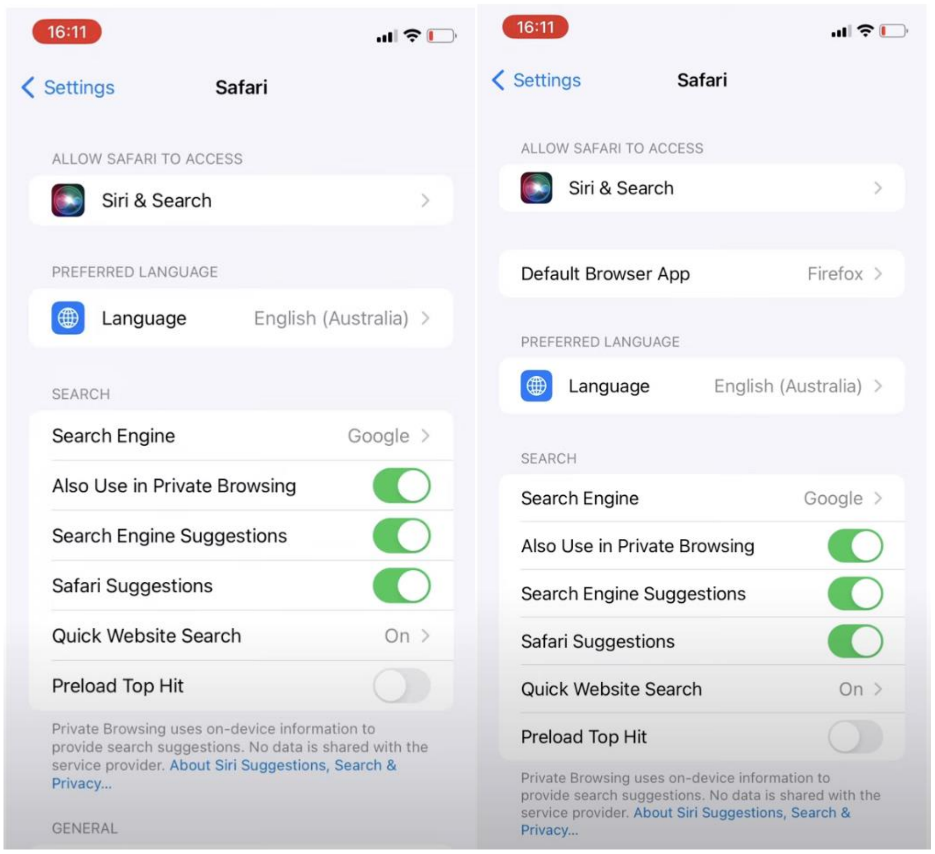
We believe the act would support the display of choice screen on devices in which the user is either transferring or restoring their settings to that device. As of the time of writing this document (iOS 17.5), this is not implemented by Apple.

3.3. Resolved Issues

3.3.1. Option to Change Default Browser Hidden if Gatekeepers Browser is the Default Browser

For the first 13 years of iOS it was impossible to change the default browser, it was locked to Safari. Then in iOS 14 (released on 2020-09-06), Apple added the ability to change it. However as part of this update, Apple specifically coded the astonishing brazen dark pattern of hiding the option to change the default browser in Safari settings, *if Safari was already default*. If Safari was *not the default browser*, the option to change the default browser became prominently shown.

In our own testing, we found that, after significant publicity (generated by OWA), Apple has quietly fixed this issue globally in iOS 17.5, which was released on 2024-05-13, long past Apple's cutoff compliance date for the DMA of 2024-03-07. While we are pleased they have taken the step to fix this without being directly compelled to do so, we felt given the clear intent, brazenness and late compliance, that it was important context worth including. You can see this below.



16:11

Settings — Safari

ALLOW SAFARI TO ACCESS
Siri & Search

PREFERRED LANGUAGE
Language — English (Australia)

SEARCH
Search Engine — Google
Also Use in Private Browsing — On
Search Engine Suggestions — On
Safari Suggestions — On
Quick Website Search — On
Preload Top Hit — Off

Private Browsing uses on-device information to provide search suggestions. No data is shared with the service provider. About Siri Suggestions, Search & Privacy...

GENERAL

16:11

Settings — Safari

ALLOW SAFARI TO ACCESS
Siri & Search

Default Browser App — Firefox

PREFERRED LANGUAGE
Language — English (Australia)

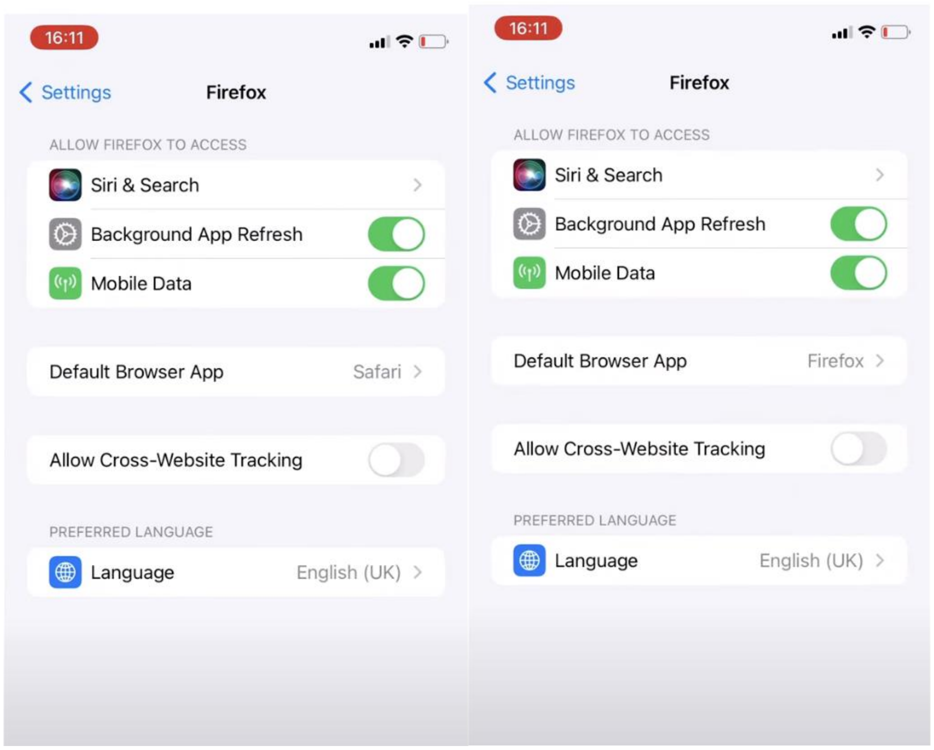
SEARCH
Search Engine — Google
Also Use in Private Browsing — On
Search Engine Suggestions — On
Safari Suggestions — On
Quick Website Search — On
Preload Top Hit — Off

Private Browsing uses on-device information to provide search suggestions. No data is shared with the service provider. About Siri Suggestions, Search & Privacy...

Left: When Safari is the default
Right: When Firefox is the default (Note the option to change default browser has appeared)

We asked Apple directly about this at the Apple DMA Workshop. You can see our question, their answer, and our analysis of their answer in this article, which was later covered on ArsTechnica.

In other browsers on iOS the option is always shown regardless of whether they are the default browser or not.


16:11
Settings — Firefox
ALLOW FIREFOX TO ACCESS
Siri & Search
Background App Refresh: on
Mobile Data: on
Default Browser App: Safari
Allow Cross-Website Tracking: off
PREFERRED LANGUAGE
Language: English (UK)

16:11
Settings — Firefox
ALLOW FIREFOX TO ACCESS
Siri & Search
Background App Refresh: on
Mobile Data: on
Default Browser App: Firefox
Allow Cross-Website Tracking: off
PREFERRED LANGUAGE
Language: English (UK)

Left: When Safari is the default
Right: When Firefox is the default (always shown on other browsers settings pages)

Hiding the option to change the default browser if the gatekeeper's browser is the default was a very clear violation of Article 13(4) as it clearly sought to undermine the users autonomy of choice.

4. Remedies

Here we propose three tranches of remedies which we believe are both justified and proportionate. Some remedies require time and/or pre-requisite remedies in order to be effective. Remedies in this document are ordered to ensure that either competitive benefits are delivered earlier or to unlock future remedies. In this way, we propose a program of continual improvement in the competitive landscape, delivering wins at every step along the path.

4.1. Tranche 1

4.1.1. Browser Engine Entitlement Contract

4.1.1.1. Remove Non-Security Terms

Apple's browser engine entitlement contract defines the conditions with which browser vendors must comply with to be granted access by Apple to hardware and software features on iOS. Article 6(7) authorizes the gatekeeper to take such measures, but exclusively in order to preserve the security of the operating system.

However, as discussed in section 3.1, Apple has incorporated into this contract a vast number of rules that do not relate to security. These rules are therefore non-compliant with the DMA, and we recommend that Apple should be compelled to remove them.

Apple should remove all non-security-related rules from this contract.

4.1.1.2. Dual Engine Browser

Apple will need to remove the following rule from its browser engine entitlement contract, and not add an equivalent rule elsewhere, such as in its app store guidelines:

> *2.2 (page 1): "Be a separate binary from any Application that uses the system-provided web browser engine"*
>
> Apple Browser Engine Entitlement Contract

This rule is a significant barrier to browser vendors porting their real browsers to iOS and defacto causes them to lose all their EU users on their existing browser app on iOS.

We cover this in greater depth in Section 3.1.2.

4.1.1.3. Allow Browser Vendors to keep their existing EU customers

Apple has chosen to restrict browser competition on iOS to the EU. Apple’s actions seek to force third-party browser vendors to build, develop, and maintain two versions of their apps for iOS while Apple’s own Safari bears no such costs.

Apple’s rules appear to force browser vendors to ship a brand new version of their app in the EU, rather than update existing apps to use their own engines. This will cause browser vendors to lose existing customers, as consumers will need to manually download the new browser.

This complexity and friction is a direct result of Apple’s own anti-competitive actions over the past 15 years. Apple’s insistence that competitors will only be allowed to compete on a level playing field (with their own engines) in the EU is already a high burden, and this rollout plan would levy additional transition costs on competitors. It is reasonable and proportionate that Apple takes steps to mitigate the damage they appear intent on causing.

Forcing browser vendors to ship two distinct products in Europe will lead to end user confusion and significant harm to these browser vendors. We do not believe that such an approach would be either fair or compliant with the DMA.

Apple should not be forcing browser vendors to reacquire existing users. Apple will need to implement a solution where browser vendors can use their own engines and keep their existing EU users.

Specifically, we are seeking the Commission to compel Apple to allow browser vendors to update their apps for EU consumers to their own engines. With this remedy, we are not asking that the Commission compel Apple to make any changes outside the EU. It is between Apple and other global regulators if they add any complex code or logic to restrict browser competition on iOS to the EU.

Apple has a number of options to comply with this remedy and have the opportunity to suggest other alternatives that may solve the underlying issue. They are:

Solution A. Allow Browser Engines Globally

Solution B. Two Binaries for One Bundle ID

Solution C. Global Dual Engine Binary with Toggle

Solution A and C would only require contract changes and the most minimal technical changes.

We cover this in greater depth in Section 3.1.3.

4.1.1.4. Requirement to use Apple Components

Apple will need to remove any rule or suggestion that browser vendors are obligated to replace parts of their browser engine with particular Apple layout, rendering or user interface components.

Importantly, browser vendors need the right, but not the obligation, to use these libraries and components.

We cover this in greater depth in Section 3.1.10.

4.1.1.5. Remove Viral Terms

“Apple Materials” is an inappropriately vague term that expands Apple’s Browser Engine Entitlement Contract to include a great many non-security rules. The way it is phrased is that any imperative statement in any document referenced from the contract or referenced from those documents in cascade is a binding rule.

As this is purely a contract for API access, Apple is required to only have strictly necessary and proportionate rules to protect the integrity of the operating system. Apple is also required to justify all of these rules as being so.

As such, these viral terms must be removed and Apple should clearly present all security rules that they wish third-party browser vendors to abide by in order to access APIs necessary for building iOS browsers, preferably in a single document.

Any update to this document should be public and all relevant parties should be alerted to the new proposed changes. Barring exceptional circumstances, changes should be known by browser vendors and outside groups months in advance.

We cover this in greater depth in Section 3.1.6.

4.1.1.6. Security Rules must be Clear

All security rules for browser vendors must be clear and upfront.

It is important that browser vendors can be confident they are abiding by all security rules. In the event that Apple wishes to introduce rules that are not strictly necessary, proportionate, justified or that violate Article 5(7) then browser vendors will need time to respond and, if appropriate, notify the Commission.

Specifically, Apple must publish all security rules for browsers in a single document. Apple must commit to these rules being available, complete and accurate. In the event Apple wishes to update this document, it must notify all parties and give significant advanced warning of the exact changes it is making, and the justification of the necessity and proportionality of said changes. Apple will need to attest to the fact that the updated or new rules do not conflict with Article 5(7).

4.1.1.7. Justify All Security Rules

We believe that Apple should be required to provide a security justification for each individual rule attached to obtaining the API access required by browsers using their own engine: Why is it strictly necessary and proportionate to prevent this access from compromising the integrity of the operating system, hardware or software features provided by Apple as per the wording and intent of the act.

Any rule that cannot be justified, or where the justification is inappropriate or insufficient, should be removed or modified.

We cover this in greater depth in Section 3.1.5.

4.1.1.8. Penalties for Security Rule Violation must be Proportionate and Justified

Apple's current penalties, as described, do not take into account the need for proportionality.

The phrase "in Apple's sole discretion" is not consistent with the fact that Apple will need to justify each of its security measures.

The language of the contract should be updated to make it clear that the penalties will be proportional and that Apple will provide detailed evidence and reasoning to support any penalty it might choose to impose.

We cover this in greater depth in Section 3.1.9.

4.1.1.9. Justify and Update Non-Proportionate Security Terms

Apple has granted itself the ability to force any browser vendor to remove any browser API that contains a vulnerability, no matter how minor and no matter how much effort the browser vendor has taken to mitigate the vulnerability.

This is particularly important, as the primary reason that the DMA grants browsers the right to use their own browser engine is to prevent gatekeepers (such as Apple) from determining the functionality and standards for third-party browsers. Allowing Apple to ban third-party browsers from offering functionality on unjustified or non-existent security justifications would severely undermine this goal.

This appears to grant Apple broad powers to use unnecessary and disproportionate security measures to block rival browsers from providing functionality that is distinct from Safari, preventing one of the core aims of the act in allowing browser vendors to port their own engines.

This clause must be removed or modified to be strictly necessary and proportionate. Apple must be required to provide clear and convincing evidence of necessity and proportionality of any individual execution of such a rule.

We cover this in greater depth in Section 3.1.4.2.2.

4.1.1.10. Vulnerability Disclosure

Apple has proposed a security disclosure requirement in their browser engine API contract. We welcome this security disclosure policy, provided it applies equally to Safari. Our concern is that if Apple can avoid full disclosure while, at the same time, compelling other vendors to disclose vulnerabilities, Apple may use these disclosures to falsely claim superior security or create a pretense for delisting other browsers from their app store. Our recommendation would be for Apple to keep this policy but to also abide by it for their own browser.

We cover this in greater depth in Section 3.1.4.2.2.

4.1.2. Safari and other Apple Services get Special Placement

On the settings page for iOS, pre-installed Apple apps are not placed with the other apps. Instead they are given a special, far more prominent location, in the settings. Other third-party apps are shown in a separate location further down the settings page. This divide suggests to users that these are *official* apps they *should* be using (which come pre-installed) and other apps are “alternative apps”.

These apps include:

- Siri
- Apple App Store
- Wallet and Apple Pay
- Mail
- Contacts
- Calendar
- Notes
- Reminders
- Freeform
- Voice Memos
- Phone
- Messages
- FaceTime
- Safari
- Stocks
- Weather
- Translate
- Apple Maps
- Compass
- Measure
- Health
- Journal
- Apple TV
- Photos
- Camera
- Books
- Podcasts
- Game Center

It indicates to the user that these apps are special and that, while it may be mechanically possible for some of these apps to be replaced by third party apps, the non-neutral nature of the interface arguably seeks to dissuade users from changing the default.

In particular the fact that Safari has a separate and elevated position from other browsers is, we believe, in contravention of Article 6(3), Article 13(6) and Recital 70 which state:

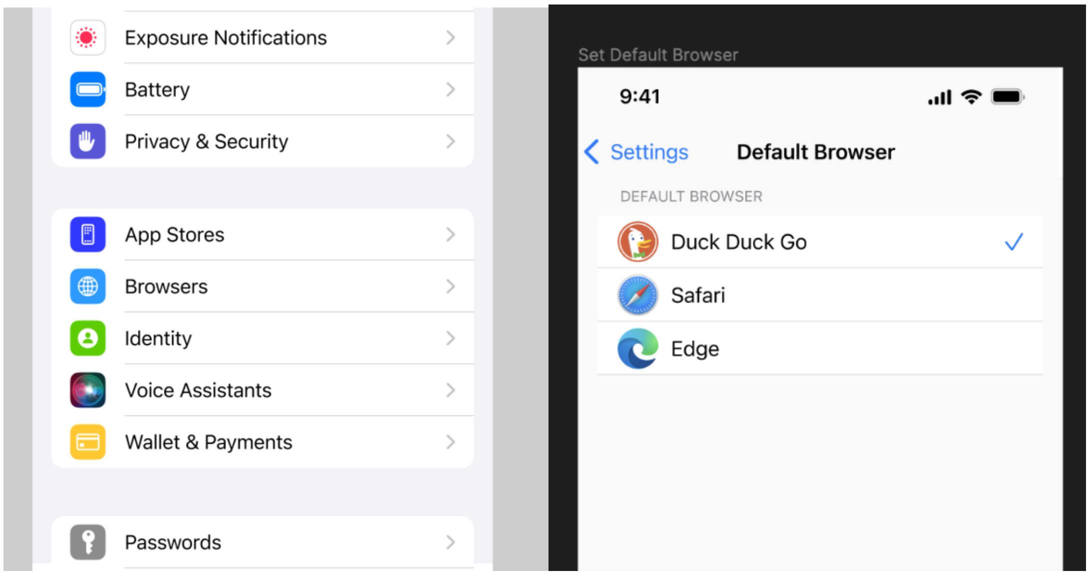
> *"**The gatekeeper shall allow and technically enable end users to easily change default settings on the operating system**, virtual assistant and web browser of the gatekeeper that direct or steer end users to products or services provided by the gatekeeper. "*
>
> Digital Markets Act - Article 6(3)
> (emphasis added)

> *"The gatekeeper shall not degrade the conditions or quality of any of the core platform services provided to business users or end users who avail themselves of the rights or choices laid down in Articles 5, 6 and 7, or make the exercise of those rights or choices unduly difficult, **including by offering choices to the end-user in a non-neutral manner, or by subverting end users' or business users' autonomy, decision-making, or free choice via the structure, design, function or manner of operation of a user interface or a part thereof.**"*
>
> Digital Markets Act - Article 13(6)
> (emphasis added)

> *"**Gatekeepers should not engage in behaviour that would undermine the effectiveness of the prohibitions and obligations** laid down in this Regulation. Such behaviour **includes the design used by the gatekeeper, the presentation of end-user choices in a non-neutral manner**, or using the structure, function or manner of operation of a user interface or a part thereof to **subvert or impair user autonomy, decision-making, or choice.**"*
>
> Digital Markets Act - Recital 70
> (emphasis added)

Apple should move all of its apps and third party apps into the same location in settings. In the event particular types of apps are of elevated importance such as app store, browser, photos, or backup provider, Apple could additionally place the current default in a special location in a neutral manner.

This is an example mockup of what that might look like:



Possible Default Location and Default selection page for browsers

The goal of this remedy is to indicate to the user that while these apps are pre-installed by Apple, they are replaceable and have no elevated privileges over other third party apps. This will encourage contestability of these services as per the intent of the act.

4.1.3. Safari is Not Uninstallable

Apple is obligated under the act to make Safari uninstallable. This is psychologically important as it indicates to users that Safari is just another app on their phone that can be uninstalled and replaced, just like any other non-Apple app.

> *"**The gatekeeper shall allow and technically enable end users to easily un-install any software applications on the operating system of the gatekeeper**, without prejudice to the possibility for that gatekeeper to restrict such un-installation in relation to software applications that are essential for the functioning of the operating system or of the device and which cannot technically be offered on a standalone basis by third parties."*
>
> Digital Markets Act - Article 6(3)
> (emphasis added)

Importantly, the WKWebView and SFSafariViewController should be treated as system components, and it should not be possible to uninstall them. Both of these components are used in a wide variety of native apps. Many native apps use the WKWebView as a convenient way to render first/second party content and any remedy which would break these apps would be unreasonable, disproportionate and counter-productive. Other apps use SFSafariViewController when the user clicks on an external http/https link and should be allowed to continue to do so, provided Apple commits to making SFSafariViewController respect the user's choice of default browser.

We would also support it not being possible to uninstall the default browser until a new default browser has been selected. An appropriate and neutral error message should be displayed if the user attempts to do so.

We are uncertain if SFSafariViewController is built on top of Safari or the WKWebView. In the event it is currently built on the WKWebView it is not a blocker to Safari being fully uninstallable.

In the event it is currently mechanically bound to Safari it will need to be updated to instead not impose a browser engine on end users and business users but rather interoperate with the users chosen default browser as discussed in 4.2.2. In the (likely rare) instance that the user's chosen default browser does not support interoperating with SFSafariViewController then links can simply open in the user's default browser directly.

4.1.4. Implement Web App Install Prompts for iOS Safari and WKWebView browsers

According to Article 6(6), Gatekeepers, such as Apple, shall not "*restrict technically or otherwise the ability of end users to switch between, and subscribe to, different software applications and services that are accessed using the core platform services of the gatekeeper.*"

iOS and Safari are both implicated in delivering Web Apps, and both are designated core platform services. Via its control of both iOS and Safari, Apple has long denied Web Apps the ability to prompt the user to be installed, adding great friction to the process of installing them. Instead, the option to install a Web App is hidden away in a hard to find "share" menu option. Competing browser vendors were only provided with access to this share menu function last year, and are still denied access to the necessary APIs within WKWebView to implement the ability for websites to enable such user-driven prompts.

Apple has implemented many pieces of equivalent functionality to install native apps from their own app store via Safari, including but not limited to App Clip Associated Domains and Smart Banners.

Due to the manner in which WKWebView-based browsers on iOS are implemented, only Apple can effectively implement install prompts for them, and has steadfastly refused to do so despite consistent developer requests.

We believe that Apple should be obligated to implement install prompts, including both automatic prompts (akin to Smart Banners) and the ability to programmatically display prompts on a button click from within websites that meet the minimum criteria for installation (e.g., providing a Web App Manifest). Apple should also implement APIs already in use across the web to facilitate control over this behavior (“onbeforeinstallprompt”). onbeforeinstallprompt is the technical term for the current implementation, it is an event that tells developers that it is possible to install the website as a web app, this means they can add a button to the page that will trigger showing the choice to the user. This allows developers to create their own install buttons.

> *“The beforeinstallprompt event fires when the browser has detected that a website can be installed as a Progressive Web App.*
>
> *There's no guaranteed time this event is fired, but it usually happens on page load.”*
>
> Mozilla Documentation

Given that Apple has withheld this functionality for almost a decade, we believe that Apple should not be able to delay discussions of the specifications as this could take months or even years. Rather, they should implement the current specification as is currently available in other browsers which support the feature. After implementing the functionality, Apple can then propose any updates or upgrades.

Apple should not be allowed to add additional scare screens or other friction as it is not necessary from a security perspective. The ability to install Web Apps via iOS Safari has been available for almost 15 years without such measures, and Apple has not felt it was necessary to implement such measures. They surely cannot be justified now, particularly after Apple’s strong claims regarding the security of its iOS Web App container.

Web Apps are protected by the sandbox of the browser, which Apple states is "*orders of magnitude more stringent than the sandbox for Native Apps*". This is important as it means that Apple does not have any significant unmitigatable security objections to such a change.

4.1.5. App Store Rules for Browsers Must Be Fair, Reasonable, and Non-Discriminatory (FRAND)

Apple may be tempted to move proposed non-security rules regarding browsers from its proposed browser engine entitlement contract to the broader app store guidelines, which it includes by reference within the proposed Browser Engine Entitlement Addendum. By phrasing the browser engine contract as an addendum, rather than a stand-alone contract, the terms of the entire “Apple Developer Program License Agreement” is included.

A number of the rules within the engine entitlement addendum do not appear to be fair or reasonable, including:

- That the browser must be solely distributed in the EU
- The browser must be a separate binary from one that uses the systems browser engine
- Apple may change the rules at any time with no notice
- Apple makes no guarantees as to the accuracy of the rules
- Apple can reject browsers for any reason at its sole discretion
- Apple can remove, suspend, break any API with no notice
- Any breach of rules, no matter how minor, will allow Apple to block or remove your application from all Apple platforms including macOS

Apple should not be allowed to add rules for browsers to the app store guidelines or apple program developer license that are unfair, unreasonable or discriminatory.

While we recognise that the DMA may not have the ability to compel Apple to allow browsers to compete fairly on iOS globally, the unfairness of keeping this privilege exclusive to Safari (out of browsers which ship their own engine) should be noted. The EU should consider any remedies available to lessen this unfairness.

4.1.6. App Store Rules for Browsers Must Not Violate Article 5(7) and Recital 43

Under Article 5(7), Apple can not impose a browser engine on competing browsers. The aim here outlined in Recital 43 is to prevent gatekeepers (such as Apple) from determining the functionality and standards of rival browsers by imposing a browser engine.

That is, browser vendors must be free to compete in features, stability, security and privacy. Under Article 13(4) Apple can not construct alternative rules or OS design choices that would undermine and render Article 5(7) ineffective.

So any rule, including rules moved from Apple’s browser engine entitlement contract into Apple’s app store rules, Apple developer program license contract or rules in documents included by reference must be compliant with Article 5(7).

4.1.7. Testing for Browser Vendors and Developers Outside the EU

Under Apple's current proposal, browser vendors and companies that develop websites for the EU, whose development teams are outside the EU, will be unable to test their websites or Web Apps in browsers on iOS that use their own engine - except for Safari. Apple's Safari will be the only browser which uses its own engine that will be available globally.

Apple's browser engine entitlement contract makes no exception for test devices in its rule that browsers using their own engine (other than Safari) must only be available (on iOS) in the EU. Apple's technical restrictions, launched as part of iOS 17.4, further indicate hostility towards reasonable exceptions for development and testing, with the OS using multiple mechanisms to geofence access to EU-specific behavior to devices physically within the EU.

For Browser Vendors, the contract is less clear. It does reference "Authorized Test Units" but offers no clarification. In practice, Apple appears to have a policy that all test units for browser vendors must be *physically located* in the EU.

> *"The Register has learned from those involved in the browser trade that **Apple has limited the development and testing of third-party browser engines to devices physically located in the EU**. That requirement adds an additional barrier to anyone planning to develop and support a browser with an alternative engine in the EU.*
>
> *It effectively geofences the development team. Browser-makers whose dev teams are located in the US will only be able to work on simulators. While some testing can be done in a simulator, there's no substitute for testing on device – which means developers will have to work within Apple's prescribed geographical boundary.*
>
> *...*
>
> *'The contract terms are bonkers and almost no vendor I'm aware of will agree to them,' lamented one industry veteran familiar with the making of browsers in response to an inquiry from The Register.*
>
> *'Even folks that may have signed something to be able to prototype can't ship under the constraints Apple's trying to impose. They're so broad and sweeping as to try to duck most of the DMA by contract ... which is certainly bold."*

Thomas Claburn - The Register
(emphasis added)

> *“Great story showing how it's clear Apple isn't serious about browser engine competition in Europe.*
>
> *We've got a blink build running well on the simulator but still have no idea if it works on device 🤷.”*
>
> Rick Byers - Web Platform Area Tech Lead, Chrome

This places browser vendors and web developers attempting to reach EU customers via the web at a significant disadvantage that Apple and Safari are not subject too. This greatly harms the goals of interoperability and competition that the DMA sets out to allow.

This will harm EU end users by making these browsers less secure due to the inability for any engineering team, geographically outside of the EU, being able to develop and test their products accurately. Some, perhaps even most, browser vendors hire security experts from all around the world and some security bugs (like JIT bugs) will need a real device to test and debug on. Not only would this put EU users at risk, but it would also unfairly disadvantage the competing browser vendors.

Is it really reasonable for Apple to in effect force the entire development teams of these mobile browsers to relocate to the EU? Would Apple accept a similar requirement for their own products in another jurisdiction?

This also prohibits automated on-device tests for browser vendors outside the EU. Again, the end result of this will be poorer quality products as a result of Apple’s unreasonable restriction.

Finally, of the millions of web developers and businesses outside the EU who serve EU customers, but do not live in the EU, should Apple really be able to make it impossible for them to effectively test their software on competing browsers?

We had hoped that Apple might reasonably foresee these complications and voluntarily cease this continued anti-competitive behavior and make it possible for developers to test their products outside of the EU. Unfortunately, Apple seems content with making the web less interoperable and more difficult to develop for.

As the DMA is (likely) unable to compel Apple to allow browser vendors to compete on iOS with their own engines globally, we propose two potential solutions:

1. Relax the requirements on test devices so that browser vendors can at least test their own browsers on test devices outside the EU. This seems an extremely reasonable request and Apple has no reason beyond malice and a belief they can’t legally be forced to do so to not comply with it. If Apple believes it does have

legitimate reasons it should publicly publish them so they can be scrutinized. This part of the remedy solves the problem for allowing browser vendors to develop and test their products.

In the case that this scenario is actually a mistake on Apple's part, and is just a lack of foresight on the needs of developers in regards to testing, Apple should issue updated guidance that browser vendor test devices are exempt from this policy.

2. Any developer with an Apple developer account should be able to download the EU versions of browsers globally onto their own iOS devices for the explicit purpose of testing. This will allow them to test their software in these browsers, which is critical in allowing Web App/website developers to compete. Apple should not be allowed to add undue friction, charge a fee, or restrict these browsers in any way that might significantly undermine the extensive manual and automated testing that major web based products undergo. This will allow the millions of web developers that service the EU to test their products.

Apple may point to TestFlight as a possible solution, but this is insufficient. There are hard usage caps of 10,000. Given there are 1.1 billion websites worldwide, we would anticipate that the number of developers that need to test for mobile browsers in the EU would be in the millions and likely comparable to the number of developers who download beta versions of Firefox/Chrome/Edge.

Apple may also point at the xCode simulator as a possible solution, however this is not acceptable, as many bugs will only appear on real devices, and development of user interfaces will often require the developer to be able to interact with the device to test the responsiveness and feel of the interactions. This sort of testing is not possible in the simulator.

4.1.8. Allow Dev/Beta Versions of Browsers on Non-Beta Versions of iOS

On other operating systems such as macOS, Windows, Linux, and Android, developers can download multiple versions of the same browser including Stable, Beta, Dev, and Canary “channels”. This allows developers of websites and Web Apps to test their products for compatibility and regressions prior to new versions of browsers being made available to consumers. This increases the quality and safety of both Web Apps and browsers.

Apple’s app store rules contains the following provision:

> *“2.2 Beta Testing*
> *Demos, betas, and trial versions of your app don’t belong on the App Store – use TestFlight instead. Any app submitted for beta distribution via TestFlight should be intended for public distribution and should comply with the App Review Guidelines. Note, however, that apps using TestFlight cannot be distributed to testers in exchange for compensation of any kind, including as a reward for crowd-sourced funding. Significant updates to your beta build should be submitted to TestFlight App Review before being distributed to your testers. To learn more, visit the TestFlight Beta Testing page.”*
>
> Apple’s app store rules

Due to this and limits imposed by Apple’s “Test Flight” program it is practically ensured that Beta, Dev, and Canary versions of competing browsers are not widely available. The number of test users required for popular browsers, which are platforms in their own right, is extraordinary among apps.

> *“TestFlight has a 10,000 user limit, which is a tiny fraction of the beta population for web browsers with large user bases.The limitation becomes a maintenance burden for browser developers: when reaching the limit, old users must be manually removed from the list. The user experience for prerelease testers is more complicated than on other platforms: users must download the separate TestFlight application, sign up for an access code, wait to receive it, and then paste it into the application. Distributing with a public link is not feasible because of the user limit restriction.”*
>
> Mozilla - Platform Tilt

Tying of browsers to OSes used to be commonplace, for example with Internet Explorer and Microsoft Windows, but in the past 15 years this practice has been abandoned by most browser vendors. Apple remains a notable exception, meaning iOS Safari can only be updated when the whole OS updates, and there can only be a single version of the

WebView or Safari at any time. This raises costs on developers who may need to procure multiple iOS devices to adequately test.

Given that the purpose of the DMA is to allow rivals to compete to provide better functionality, stability, and security, Apple should not be able to restrict browser vendors via app store rules from providing Beta, Dev and Canary versions of their products provided they are clearly labeled and marketed as such. Specifically, it should amend the rule to make an exemption for browsers.

Such competition will improve browser testing on iOS and may push Apple to make long-needed improvements to provide equivalent functionality for Safari.

This change will not require any technical changes on Apple’s end nor will it require the EU to make any extra-territorial requests. This simply requires Apple to update its contracts and app store rules to allow this critical use case.

4.1.9. Browser Choice Screen on iOS

In order to comply with both the letter and intent of the Digital Markets Acts Article 6(3) and to make that compliance effective we believe that Apple should make the following changes:

- Allow browsers to show a single line of text explaining why users should pick their browser on the choice screen directly. This line should be allowed to be at least up to 150 characters long.
- The browser choice screen should not be locked to or entrench Apple app store. Both the browsers available and the calculation of the top 12 should include browsers downloaded either directly or from other app stores. Browser vendors should be able to opt to provide their browsers directly to users.
- Once the user has chosen their browser, the selection flow should continue to the home screen and download the chosen browser in the background, allowing the user to continue their task. Making the user wait with a prominent cancel button undermines the autonomy of users.
- Apple should share aggregate statistics with browsers vendors about the performance of their browser on the choice screen. This should be daily and automatic. Importantly this needs to include enough detail to allow browser vendors to calculate the percentage of the times it is chosen, per country, per day. Browser vendors should be free to publicly publish this data if they so choose.
- Being set as the default browser should grant the chosen browser the *hotseat*. Under no circumstances should the choice screen show if Safari is not already the default browser. Apple should re-show the choice screen to all users that currently have Safari as their default, and that the choice screen was inappropriately shown too, if they do not have this data it should be reshown to all users with Safari as the default.
- The choice screen should be shown whenever users encounter a new reinstallation event, such as purchasing a new phone, restoring from an old phone or a factory reset

We cover this in greater depth in Section 3.2 .

4.2. Tranche 2

Tranche 2 interventions are not less important than those proposed for Tranche 1 (or for Tranche 3), but are instead proposed in a staged order to ensure that progress is made consistently.

4.2.1. Web App Installation and Management for third-party Browsers

> *"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

First, Apple claimed that they had to entirely remove Web App functionality to avoid sharing it with third-party browsers using their own engine.

> *"**Addressing the complex security and privacy concerns associated with web apps using alternative browser engines would require building an entirely new integration architecture that does not currently exist in iOS** and was not practical to undertake given the other demands of the DMA and the very low user adoption of Home Screen web apps. **And so, to comply with the DMA's requirements, we had to remove the Home Screen web apps feature in the EU.**"*
>
> Apple's statement
>
> (emphasis added)

They then, under significant pressure, reversed that decision and stated that they will keep the current status-quo, implying that access to the underlying APIs required to run Web Apps would be locked to the WebKit implementation and that third-party browsers using their own engine would not be provided sufficient API access to contest it. That is, third-party browsers would continue to be unable to install and manage Web Apps using their own engine.

> *"This support means Home Screen web apps continue to be built directly on WebKit and its security architecture, and align with the security and privacy model for native apps on iOS."*
>
> Apple's statement

This stance is in clear contravention of Article 5(7), 6(7) and 13(4).

It also undermines the Act's core purpose in prohibiting imposing browser engines outlined in Recital 43, which is to prevent gatekeepers from blocking third-party browsers for implementing features for "***web software applications.***"

Browser vendors require the ability to install, manage, and control Web Apps using their own engines. This is the only way that browser vendors can compete in the provision of functionality, stability, security and privacy for Web Apps.

As such, Apple must build, update, or provide access to all relevant APIs to install Web Apps to browser vendors with the browser engine entitlement. OWA believes the Commission, and competitors, are owed a full and timely explanation of the implementation plan for these features.

4.2.2. SFSafariViewController Must Respect Browser Choice

SFSafariViewController, Apple's Remote-Tab In-App Browser (IAB) system continues to self-preference Safari, denying users access to their default browsers when web pages are loaded within third-party apps.

In order to respect the users choice of default browser as mandated in Article 6(3), and to conform with Article 5(7) which prohibits the imposition of a browser engine on end users and business, Apple must extend SFSafariViewController and provide an operating system API which other browsers can register and interoperate with it to be invoked when users click on links in non-browser apps. This can work in a manner similar to how Android Custom Tabs works. Apple must provide an implementation plan to the Commission and should share its documentation and timeline prior to release, as soon as practicable. OWA suggests this should take less than 6 months, even with Tranche 1 interventions underway as the engineering work is not large.

Apple may claim that the fact they allow Native Apps to implement their own in-app browser means they are not imposing their browser engine on business users, however, this is an extremely high friction barrier: In order not to use Safari's browser engine and instead use the default browser, these businesses must invest to build their own WebView browser as opposed to simply calling the appropriate OS API.

Further this also undermines the users choice of default browser by making all native apps that call the system default remote tab browser use Safari instead of the users default browser. This, in turn, will cause friction as Safari will not share user preferences and history with the user's default browser, resulting in an experience that is dubbed The Forgetful Web.

We cover this in detail in our document "*OWA - DMA Interventions - In-App Browsers*".

4.2.3. Allow Third-Party Browsers to Ship their Own Extensions

> *"Browser extensions are a key part of the web ecosystem and most popular browsers support them. Browser extensions allow developers to add functionality to the browser providing increased utility, usability, and interoperability with applications installed on the system.*
>
> *For distribution, popular browsers have established extension catalogs that are available on the open web and curated by the browser vendors. Because extensions have elevated privileges, developers submit them to be approved to ensure safety and compatibility. Browser vendors make their own decisions about the APIs available to extensions. Extensions for each browser are installed and managed within the browser resulting in a common user experience across platforms.*
>
> *Safari supports extensions distributed on the iOS App Store. However, third-party browsers are prevented from offering their own established extension functionality because it would violate section 2.5.2 of the App Store Review Guidelines. To allow third-party browsers to offer the same functionality and be competitive with respect to browser extensibility, the App Store software requirements should be relaxed to permit third-party browsers to use their own extension catalogs. Third-party browsers could then use the existing web-based distribution model (where users can browse and install extensions directly from the browser) allowing for browser extensions to be used on iOS, similarly to other mobile and desktop platforms."*
>
> Mozilla - Platform Tilt

Currently Apple does not allow, and has no plans to allow, third-party browsers to ship their own extensions. Currently this functionality is exclusive to browsers that rely on the WKWebView and rely on extensions shipped via Apple's app store.

In order to compete fairly, browsers will need the ability to distribute extensions directly. This is explicitly prohibited by Apple's current app store rules.

Apple will need to update their browser entitlement contract and app store rules to allow browser vendors to compete in the provision of such functionality.

Under 5(7), browsers must be allowed to port their extension architecture.

4.2.4. In-App Browsers Must Respect Browser Choice

Apple has encouraged and overseen an ecosystem of native apps downloaded via Apple’s app store which ignore user choice of default browser on a massive scale.

In our document "*OWA - DMA Interventions - In-App Browsers*" we proposed the following remedies to fix the issue:

1. Designated Core Platform Services should respect the users choice of default browser.

2. App store rules must mandate non-browser apps use the user's chosen default browser for http/https links to third-party websites/Web Apps.

3. Apple must update SFSafariViewController to respect the user's choice of default browser.

4. Third-party businesses must be provided an explicit and effective technical opt-out from non-default in-app browsers.

5. OSes must provide a global user opt-out. Apps must also request explicit permission from users.

6. Google must remove the ability to override the users choice of default browser via Android Custom Tabs.

Users' choice of default browser and the consequent competition is only effective if that choice is respected.

4.2.5. Default Browser Dark Patterns and Prompt API

Apple currently introduces friction that impairs user choice of default browser. This is explicitly forbidden in Recital 70, Article 6(3), and Article 13(4).

> *"Given the substantial economic power of gatekeepers, it is important that the obligations are applied effectively and are not circumvented. To that end, the rules in question should apply to any practice by a gatekeeper, irrespective of its form and irrespective of whether it is of a contractual, commercial, technical or any other nature, insofar as the practice corresponds to the type of practice that is the subject of one of the obligations laid down by this Regulation. Gatekeepers should not engage in behaviour that would undermine the effectiveness of the prohibitions and obligations laid down in this Regulation.* ***Such behaviour includes the design used by the gatekeeper, the presentation of end-user choices in a non-neutral manner, or using the structure, function or manner of operation of a user interface or a part thereof to subvert or impair user autonomy, decision-making, or choice."***
>
> Digital Markets Act - Recital 70
> (emphasis added)

Apple introduces this friction in five ways:

1. Third-party browsers cannot prompt users to change the default via a one-click system prompt.
2. Third-party browsers cannot detect whether they are the default.
3. Safari's settings are prominently positioned on the iOS settings page compared to third-party browsers.
4. Searching for "default" or "default browser" in settings yields no results.
5. There is no centralized location for changing default apps (including browsers) on iOS.

Combined, these factors significantly restrict users' free choice in selecting and switching between their choice of browser.

In order to be compliant with this specific aspect of the DMA, we believe Apple should make the following changes:

1. Move the option to change default browser out of the browser settings and into a centralized location.

2. Have this option visible, even if Safari is the only currently installed browser.

3. Have this option show up in search if the user searches for "default", "browser" or "default browser".

4. Allow browsers to know if they are the current default browser.

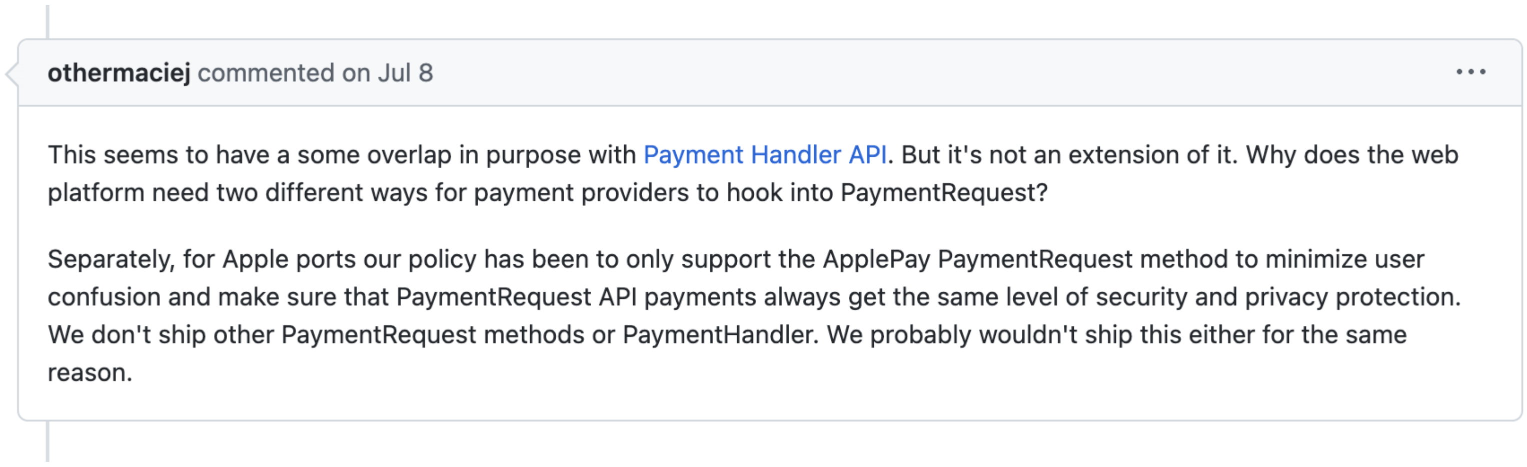
5. Provide a system prompt to browsers (with an option to never ask again, as is usual for all permission prompts) that allows browsers to prompt the user to one-click set it as the new default browser. This is standard on most operating systems.

4.2.6. Safari is Locked to Apple Pay

Currently iOS Safari payments are locked to Apple Pay. As Safari is a core platform service it is not allowed to impose a payment service on either end users or business users under Article 5(7). Therefore, Safari must be upgraded to support third-party payment services including allowing them to reach feature parity with Apple Pay.

This upgrade must also extend to all WKWebView-based browsers on iOS.

We would like to note that the specific standardized "PaymentRequest" Web API that Apple is using is actually designed to support multiple payment services and to our knowledge Safari is the only major browser (possibly only browser) that has locked it to a particular payment provider.


othermaciej commented on Jul 8

This seems to have a some overlap in purpose with Payment Handler API. But it's not an extension of it. Why does the web platform need two different ways for payment providers to hook into PaymentRequest?

Separately, for Apple ports our policy has been to only support the ApplePay PaymentRequest method to minimize user confusion and make sure that PaymentRequest API payments always get the same level of security and privacy protection. We don't ship other PaymentRequest methods or PaymentHandler. We probably wouldn't ship this either for the same reason.

Maciej Stachowiak - Senior Director of Software Engineering at Apple has publicly stated that Apple will *"only support the ApplePay PaymentRequest method to minimize user confusion and make sure that PaymentRequest API payments always get the same level of security and privacy protection."*.

This will need to be fixed in order for Apple to be in compliance with the payment service aspect of Article 5(7).

4.3. Tranche 3

Tranche 3 interventions include some of the most impactful, long-term changes, but OWA recognises that they may not be possible within a quarter or two. As such, we suggest the DMA team provide notice of intent to require them early, but work with stakeholders to bring them about in a phased way.

4.3.1. Direct Browser Installation

Apple is obligated under the act to allow native apps to be downloaded from a web page. They have allowed, and interpreted it this way, for third-party app stores. The act states that Apple is only allowed to have strictly necessary, proportionate and justified security measures to protect the integrity of the operating system. This means any security measures, such as warnings or scare screens, need to be heavily justified by Apple as strictly necessary.

There is great fear in the industry that Apple wields app store review as a weapon to punish competitors. This fear appears to be well justified.

On March 5th, Spotify submitted an update to Apple that puts links to Spotify's website, along with pricing information for different subscription options, directly in the EU version of its app, without using Apple's payment system. Despite being fined 1.8 billion by the EU on this very topic, Apple refused to let the update pass app review even though more than 2 weeks had passed since the update was submitted.

> *"Moreover, Apple has demonstrated its ability to use its smartphone monopoly to impose fee structures and **manipulate app review to inhibit aggrieved parties from taking advantage of regulatory and judicial solutions imposed on Apple** that attempt to narrowly remedy harm from its conduct."*
>
> DOJ - Case 2:24-cv-04055
> (emphasis added)

> *"Specifically, Apple sets the conditions for apps it allows on the Apple App Store through its App Store Review Guidelines. Under these guidelines, Apple has sole discretion to review and approve all apps and app updates. Apple selectively exercises that discretion to its own benefit, deviating from or changing its guidelines when it suits Apple's interests and allowing Apple executives to control app reviews and decide whether to approve individual apps or updates. **Apple often enforces its App Store rules arbitrarily. And it frequently uses App Store rules and restrictions to penalize and restrict developers that take advantage of technologies that threaten to disrupt, disintermediate, compete with, or erode Apple's monopoly power.**"*
>
> DOJ - Case 2:24-cv-04055

(emphasis added)

> *"there are endless horror stories around curation of the store. Apps are rejected in arbitrary, capricious, irrational and inconsistent ways, often for breaking completely unwritten rules."*
>
> Benedict Evans - Technology Writer

> *"There's a lot of talk about the 30% tax that Apple takes from every app on the App Store. The time tax on their developers to deal with this unfriendly behemoth of a system is just as bad if not worse"*
>
> Samantha John - CEO Hopscotch

Given that third-party browsers using their own engine already need to comply with extensive security rules in order to be allowed access to the relevant APIs, there is no security justification for any additional scare screens or warnings.

Currently, only a few browser vendors offer their browser through the macOS app store. It seems plausible that browsers will want to abandon the iOS app store at some point in the future to avoid having to deal with any unreasonable app store guidelines that Apple might impose.

A key idea in the DMA is that gatekeeper services, such as the Apple's app store, offered on top of operating system core platform services such as iOS are optional, both for users and businesses. Businesses should be free to distribute directly and users should be free to get software directly from third-party developers. If Apple wishes to retain users and business on their app store they must compete to make it more attractive. As mentioned before, the only leeway they are given by the act to intervene in these direct interactions is via strictly necessary, proportionate and justified security measures to protect the integrity of the operating system.

This will be beneficial to competition as it will remove a significant source of power Apple has over these browser vendors. However, this future is only possible if the process for installing a browser directly is made frictionless and painless.

We believe that in order to comply with the DMA, Apple should:

- Allow browsers to be installed directly from their own websites.
- Not place any friction or scare screens in the installation process.
- Not place any rules that aim to inhibit or make more expensive installations that are not via Apple's app store.

4.3.2. Allow Users to Switch the Distribution Method of Native Apps

The DMA obligates gatekeepers to allow business users to promote and choose the distribution channel that they consider most appropriate for the purpose of interacting with any end users; those business users have already acquired core platform services provided by the gatekeeper.

This means that business users need to be able to switch the distribution channel of individual native apps downloaded by Apple's app store to either direct download or to be managed by another app store.

> *In certain cases, for instance through the imposition of contractual terms and conditions, gatekeepers can restrict the ability of business users of their online intermediation services to offer products or services to end users under more favourable conditions, including price, through other online intermediation services or through direct online sales channels. Where such restrictions relate to third-party online intermediation services, they limit inter-platform contestability, which in turn limits choice of alternative online intermediation services for end users. Where such restrictions relate to direct online sales channels, they unfairly limit the freedom of business users to use such channels. To ensure that business users of online intermediation services of gatekeepers can freely choose alternative online intermediation services or direct online sales channels and differentiate the conditions under which they offer their products or services to end users, it should not be accepted that gatekeepers limit business users from choosing to differentiate commercial conditions, including price. Such a restriction should apply to any measure with equivalent effect, such as increased commission rates or de-listing of the offers of business users.*
>
> Digital Markets Act - Recital 39

> *"To prevent further reinforcing their dependence on the core platform services of gatekeepers, and in order to promote multi-homing, **the business users of those gatekeepers should be free to promote and choose the distribution channel that they consider most appropriate for the purpose of interacting with any end users that those business users have already acquired through core platform services provided by the gatekeeper** or through other channels."*
>
> Digital Markets Act - Recital 40
> (emphasis added)

> *"**The gatekeeper shall allow business users**, free of charge, **to communicate and promote offers, including under different conditions, to end users acquired via its core platform service** or through other channels, and to conclude contracts with those end users, **regardless of whether, for that purpose, they use the core platform services of the gatekeeper**."*

Digital Markets Act - Article 5(4)
(emphasis added)

Currently the only way of doing this is the extremely awkward process of entirely uninstalling an app (and thus deleting) all its data, then reinstalling it via either the third party app store or directly from the developers website.

> *"**The gatekeeper shall not restrict technically or otherwise the ability of end users to switch between**, and subscribe to, **different software applications and services that are accessed using the core platform services of the gatekeeper**, including as regards the choice of Internet access services for end users."*

Digital Markets Act - Article 6(6)
(emphasis added)

In this case the core platform service would be the operating system iOS which is restricting the ability of users to switch apps between app stores, or being managed directly by the developers.

Switching which store or developer is controlling a particular app should be straightforward. To not do so is by design of the operating system and its interfaces undermine effective compliance with Recital 39, Recital 40, Article 5(4) and Article 6(6). Apple should build appropriate interfaces to facilitate such a switch.

This also comes with the advantage that should any individual app store become defunct, there would now be a mechanism to not orphan apps on that app store. It would also apply great pressure on app stores to be reasonable otherwise risk businesses and users switching their existing apps en masse to an alternative. Enabling this competitive pressure would benefit both business and consumers.

Finally, Apple must allow businesses the right to be able to promote this switch, and its advantages such as price decreases or extra features, to users which have downloaded the app via Apple's app store. There should also be no prohibition via app store rules or contracts on promoting direct download or third party app stores in adverts on other third party apps in Apple's app store. This should also include the case where a business wishes to discontinue offering their app via a particular app store and wishes to promote alternative options for users to switch too.

Apps will need the ability to detect which source is managing their distribution, i.e which app store or direct download. An API will need to exist which provides this to the app. This is to allow the app to appropriately alter its features, interfaces or displayed prices.

This change will bring significant advantages for consumers and businesses. It will allow app stores to be more easily contested by third parties and reduce the entrenchment of the existing body of installed apps. This will lead to better prices, more competition and greater choice for consumers.

4.3.3. Direct Install Browsers Should Be Included In Choice Screens

There is no plausible reason for Apple to exclude browsers from choice screens if they are not listed within its app store. Choice screens attempt to remedy the harm from Apple's historic self-preferencing by setting its browser as the default. The article seeks to prevent gatekeepers that might "*favour its own or third-party services or products on its operating system*", and so it does not make sense that participation in systems designed to redress harms caused by favoring of their own services and products must then rely on forced participation in them unduly.

Entrants that choose to entirely avoid Apple's app store should not be penalized for this choice. Thus, browsers that are available for direct download should be eligible to appear on the choice screen.

> *"**To prevent further reinforcing their dependence on the core platform services of gatekeepers, and in order to promote multi-homing, the business users of those gatekeepers should be free to promote and choose the distribution channel that they consider most appropriate** for the purpose of interacting with any end users that those business users have already acquired through core platform services provided by the gatekeeper or through other channels."*
>
> Digital Markets Act - Recital 49
> (emphasis added)

It is important that Apple does not entrench an existing core platform service in the process of satisfying other obligations of the DMA. In the context of both the intent and letter of the DMA, locking all browsers to its app store both mechanically and via its rules does not appear to be compliant.

Apple must publish the mechanisms for including a "direct install" version of a browser on the choice screen. Browser vendors should be allowed to distribute both via the AppStore and directly, and be able to opt to have the choice screen connected to their direct install version.

4.3.4. Apple Should Make Notarization for Directly Downloaded Browsers Automatic

Apple has announced that it intends to notarize apps that are not downloaded by its app store but that it will keep this review strictly limited to security.

> *"Notarization for iOS apps is a baseline review that applies to all apps, regardless of their distribution channel, focused on platform policies for security and privacy and to maintain device integrity. Through a combination of automated checks and human review, Notarization will help ensure apps are free of known malware, viruses, or other security threats, function as promised, and don't expose users to egregious fraud."*
>
> Apple - On Notarization

It is our opinion that this policy for third-party browsers which are downloaded directly is unproportionate, unnecessary and will in fact worsen security.

It is important to note that what Apple is proposing is not equivalent to the system Apple has for notarization of macOS software which is a fast and automated process.

> *"Notarize your macOS software to give users more confidence that the Developer ID-signed software you distribute has been checked by Apple for malicious components.* ***Notarization of macOS software is not App Review****. The Apple notary service is an automated system that scans your software for malicious content, checks for code-signing issues, and returns the results to you quickly. If there are no issues, the notary service generates a ticket for you to staple to your software; the notary service also publishes that ticket online where Gatekeeper can find it."*
>
> Apple - on macOS Notarization
> (emphasis added)

Were Apple's proposal simply to apply automatic checks including for code signatures that verify the developer is the same developer with the browser entitlement, that would be perfectly acceptable.

Apple's language however makes it clear that this is "app store review" in disguise, although nominally locked to security issues.

Our concern is that Apple will introduce considerable delay and friction into the process of updating browsers. We are also concerned, given Apple's known abuse of their Apple app store review process, that if Apple is in a dispute with a third-party browser, they may attempt to use blocking of updates on security ground as a weapon to gain concessions from browser vendors or otherwise wield the fear of such a rejection as a tool to limit third-party browsers to compete.

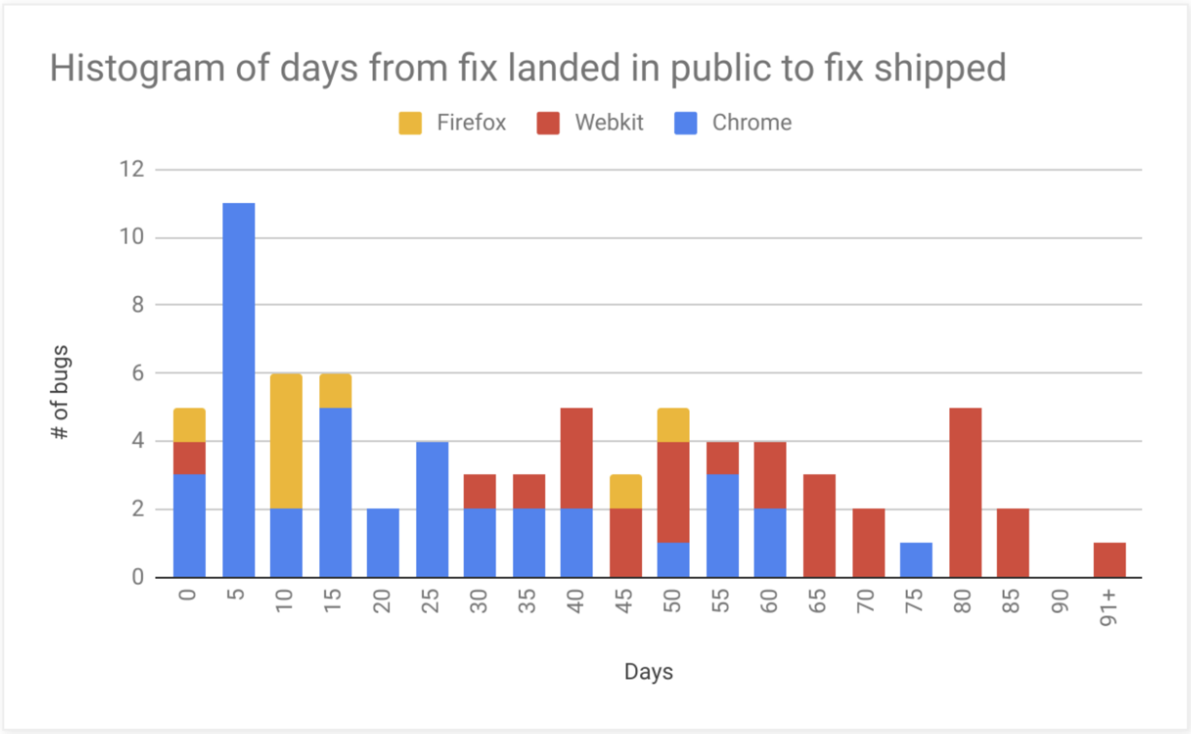
"Patch gap" is the amount of time between a vulnerability being discovered and it being patched on consumers devices. It is a critical aspect of security that this gap is as small as possible.

There is a strong argument that behavior mentioned in the previous two remedies, such as Apple preventing an update to Spotify for two weeks, and the pattern of behavior as outlined in the DOJ complaint, means that delays can often be arbitrary and significant.

Even the 90% in 24 hours for app review that Apple claims it achieves with its app store will significantly worsen patch gap. For reference the macOS notarization process is automated and takes less than 1 hour.

> *"Apple suppresses such innovation through a web of contractual restrictions that it selectively enforces through its control of app distribution and its 'app review' process*
>
> *[...]*
>
> *Apple often claims these rules and restrictions are necessary to protect user privacy or security, but Apple's documents tell a different story. In reality, Apple imposes certain restrictions to benefit its bottom line by thwarting direct and disruptive competition for its iPhone platform fees and/or for the importance of the iPhone platform itself. "*
>
> DOJ Complaint against Apple

Browser vendors have their own dedicated security teams which have worked for decades to secure their own browsers. Apple has a worse track record than Firefox and Chrome when it comes to patch gap.



Google Project Zero - Statistics on Patch Gap

iOS users remain vulnerable to known bugs in Safari longer than users of alternative browsers on every other OS. This picture is made even clearer by OS update rates. Since Safari requires a full operating system update, further steps (and attendant delay) is introduced in getting patches into user's hands, than if the browser updated like a "normal app" separate from OS updates (the standard on all other modern OSes). Safari requires the user to update their entire operating system, a process that makes the device unusable for up-to 30 minutes.

Apple is presumably claiming this right to review under the security provision in 6(4) which states:

> *"The gatekeeper shall not be prevented from taking, to the extent that they are strictly necessary and proportionate, measures to ensure that third-party software applications or software application stores do not endanger the integrity of the hardware or operating system provided by the gatekeeper, provided that such measures are duly justified by the gatekeeper."*
>
> Digital Markets Act - Article 6(4)

However, this does mean the security measure, in this case the need to notarize the third-party browsers, needs to:

- Be strictly necessary
- Be proportionate
- Be to prevent browsers from endangering the integrity of the hardware or operating system provided by the gatekeeper
- Be justified by Apple

Given there is a strong argument that Apple will in fact worsen security via this policy, the onus is on Apple to state a convincing argument on how they intend to improve security with this policy, including what additional staffing specifically for third-party browser vendors they are intending on hiring to implement it.

Apple will need to convincingly show why it is strictly necessary and proportionate for them to enforce this when all the major browser vendors have their own dedicated security teams.

They will also need to show why simple rules such as patching at a regular cadence or committing to fixing vulnerabilities within a particular time period are not sufficient. For all these rules, Safari should be the benchmark; No browser should be penalized for doing a better job at security than Safari.

Apple should guarantee that notarization for third-party browsers, with the browser entitlement which are downloaded outside Apple’s app store, is a fast and automatic process.

Our recommendation is that Apple’s only recourse against third-party browsers with serious security issues, which they refuse to fix in a timely manner or that are compromising the operating system against the interests of the user, is the option to revoke their browser engine entitlement.

This needs to be paired with strong financial penalties for Apple, in the event it takes such an action without sufficient evidence.

This setup would remove Apple’s ability to bully browser vendors via threatening to delay updates on bogus security grounds, while still granting the ability to outright remove malicious or incompetent browsers where strictly necessary, reasonable and justified.

Apple should be subject to a SLA (Service Level Agreement) in return for the ability to impose automated notarization on third parties. It would be our recommendation that the SLA for notarization should be less than an average of 15 minutes with a maximum of 1 hour. It is important that Apple not be left free to abuse the ability to delay app updates. This delay would be solely within Apple's control and no third party software provider would have the ability to improve upon the speed of Apple's notarization.

A key metric directly downloaded browsers and third party app stores will compete on is the ability to provide my timely updates. Allowing Apple the ability to selectively block or make arbitrarily slow (potentially weeks) particular updates with no explanation significantly undermines competition with Apple's app store and grants Apple great power to bully developers who have opted to distribute outside of Apple's app store.

4.3.5. Apple Should Not Break Updates for EU Residents Traveling Outside The EU

Apple has stated that it will prevent all updates for apps downloaded from third-party app stores that are outside the EU for more than 30 days.

Apple has not released any explicit statement on what will happen to browsers using their own engine downloaded from Apple's app store if the user (an EU resident) leaves the EU for greater than 30 days. Given Apple is attempting to block even browser vendors from testing on their own test devices outside the EU, it seems likely they will attempt to extend a similar policy to third party browsers which use their own engine even if they are downloaded from Apple's app store.

We would like a statement from Apple clarifying that this does not apply to browsers with the browser engine entitlement.

> *"If you leave the European Union, you can continue to open and use apps that you previously installed from alternative app marketplaces. Alternative app marketplaces can continue updating those apps for up to 30 days after you leave the European Union, and you can continue using alternative app marketplaces to manage previously installed apps. However, you must be in the European Union to install alternative app marketplaces and new apps from alternative app marketplaces."*
>
> Apple's statement on breaking updates for EU residents outside the EU for greater than 30 days

Importantly:

- This has no exemption for EU residents.
- This rule will block security updates.
- Safari is immune to this rule (as it is only available on Apple's app store)
- No other gatekeeper has taken such extreme measures to limit the competition the DMA allows to the EU.

Apple has not included any arguments in their statement on why EU residents traveling for longer than 30 days cease to be EU residents, nor any discussion on how they will ensure that this policy does not apply to any EU residents.

Under Schengen agreements, EU residents have the legal right to freedom of movement between all 29 Schengen countries, and the legal right to stay in each of the 29 countries up to 90 days without the need to apply for a travel visa or permit, as there are no borders between these countries. Importantly, among these 29, there are countries like Switzerland, Norway and Iceland, which are not members of the EU. Meaning that, for example, a German citizen can legally visit and stay in their summer cottage in Switzerland for up to 90 days.

In the same spirit, the EU and the US have mutual visa-free travel agreements, allowing an EU citizen to travel to the US with an ESTA visa for up to 90 days without losing their status as an EU resident.

Meaning that, a Swedish citizen can legally visit their parents in Norway for up to 90 days, then fly to Iceland and spend an additional 90 days with their siblings, then fly to the US to visit their friends for another 90 days, then spend their summer in Switzerland for up to 90 days, and so can legally stay away from the EU for even a year or more if they wish, whilst continuing to be an EU resident throughout.

In addition, according to the EU / EEA VAT and taxation law, which includes non-EU countries such as Norway and Iceland, EU residents are only considered tax residents of another member state after 6 months. A rule which is more or less globally harmonized to simplify global taxation of individuals. Meaning that the residents of EU member states are only legally considered tax-residents of another country after having spent 6 months in the destination country.

This means that for a very large number of EU residents traveling outside the EU, or working outside the EU, Apple's arbitrary 30-days rule to prevent app updates actually puts EU residents' cyber-safety and security at great risk. Apps subjected to this policy, in particular browsers, will cease to receive security updates.

This 30-day policy greatly contradicts Apple's justification claiming that their biggest priority is the safety of their users, as they could simply remove the arbitrary 30-day rule, or increase it to something more reasonable, such as 6 months to a year or more to accommodate and match the legal rights of EU residents.

Even if this policy is scoped by Apple to not apply to browsers shipped on their app store, should browsers choose to abandon Apple's app store in the future they will be subject to Apple's decision. Allowing browsers the option to ship outside of Apple's app store is an important option to weaken Apple's ability to dictate to browsers their form and functionality, will allow browsers to reduce the time to deploy security patches and will improve their ability to compete fairly and effectively. The right to choose how to distribute to users is codified in the DMA under Recital 40, Article 5(4) and Article 6(6).

> *"To prevent further reinforcing their dependence on the core platform services of gatekeepers, and in order to promote multi-homing,* ***the business users of those gatekeepers should be free to promote and choose the distribution channel that they consider most appropriate for the purpose of interacting with any end users that those business users have already acquired through core platform services provided by the gatekeeper*** *or through other channels."*
>
> Digital Markets Act - Recital 40
> (emphasis added)

> *"****The gatekeeper shall allow business users****, free of charge,* ***to communicate and promote offers, including under different conditions, to end users acquired via its core platform service*** *or through other channels, and to conclude contracts with those end users,* ***regardless of whether, for that purpose, they use the core platform services of the gatekeeper****."*
>
> Digital Markets Act - Article 5(4)
> (emphasis added)

> *"****The gatekeeper shall not restrict technically or otherwise the ability of end users to switch between****, and subscribe to,* ***different software applications and services that are accessed using the core platform services of the gatekeeper****, including as regards the choice of Internet access services for end users."*
>
> Digital Markets Act - Article 6(6)
> (emphasis added)

Apple should not be able to break important updates (including security updates) for EU residents who are traveling outside the EU. Apple will need to remove or update this policy to be compliant with the DMA.

4.3.6. Opt-In Rights Contract Should Be Removed

All businesses serving EU customers should automatically have all of the rights granted to them under the DMA. They should not have to opt-in to those rights under different rules or punitive pricing changes.

Business users in the act are defined as:

> *"'business user' means any natural or legal person acting in a commercial or professional capacity using core platform services for the purpose of or in the course of providing goods or services to end users"*
>
> Digital Markets Act

This is a broad definition that currently covers almost every business supplying an app on the iOS app store.

Apple should not be able to either contract businesses out of their rights under the DMA, or attach a price to those rights.

This is not how any law functions. For example, Apple can not sell iPhones to consumers at two different prices, one which abides EU's laws on deceptive advertising, and one that does not; The law applies to all of Apple's services and hardware.

Similarly, Apple should not be able to segment business users into ones for which Apple has to abide by the DMA and another for which they do not have to. Rather, Apple should have to automatically abide by the DMA for all business users operating within the EU, that is every business that supplies applications to EU residents.

Apple should remove its new "alternative contract" and update its existing contract such that all business users within the EU are automatically moved onto terms compliant with the DMA. All of Apple's pre-existing contracts which apply to the business interactions between end users and business users in the EU must be updated to comply with the DMA. Business users should not have the choice of being on non-DMA compliant terms. This false choice allows gatekeepers a method of circumventing their responsibilities.

4.3.7. Core Technology Fee Should Be Removed

Apple's proposal also includes plans for how they would allow third-party native app stores to compete with their own on iOS. This is directly compelled by the act.

Apple's proposal seems designed to prevent any free app from ever even considering moving to a third-party native app store on iOS, or being available for direct download from its own website.

In order to list their app on a third-party app store or on their own website, the app developer must sign an alternative contract allowing them the rights granted under the DMA. However, this alternative contract comes with significantly different pricing.

If you choose the second contract, Apple will waive some app store fees but will add a new fee called "Core Technology Fee", summarized as a 50 cent charge per-user per-year, past the first million downloads.

Critically, and to OWA's astonishment, **this also includes downloads from the Apple app store.**

In Apple's own words the fee is:

> *"The Core Technology Fee (CTF) is an element of the new business terms in the European Union (EU) that reflects the value Apple provides developers through ongoing investments in the tools, technologies, and services that enable them to build and share innovative apps with users around the world. Developers can choose to remain on the App Store's current business terms or adopt the new business terms for iOS apps in the EU."*

Apple Documentation

They also explain:

> *"The Core Technology Fee (CTF) reflects Apple's investment in the tools, technology, and services that enable developers to build and share their apps with Apple users. **That includes more than 250,000 APIs**, TestFlight, Xcode, and so much more. These tools create a lot of value for developers, whether or not they share their apps on the App Store.*
>
> ***The CTF only applies to developers who adopt the new terms for alternative distribution and payment processing"***

Apple Documentation
(emphasis added)

That is, Apple explicitly states that the fee covers access to iOS's APIs.

The DMA specifically precludes any fee for software or hardware API access:

> *Article 6(7) The gatekeeper shall allow providers of services and providers of hardware, **free of charge**, effective interoperability with, and access for the purposes of interoperability to, the same hardware and software features accessed or controlled via the operating system or virtual assistant listed in the designation decision pursuant to Article 3(9) as are available to services or hardware provided by the gatekeeper. Furthermore, the gatekeeper shall allow business users and alternative providers of services provided together with, or in support of, core platform services, free of charge, effective interoperability with, and access for the purposes of interoperability to, the same operating system, hardware or software features, regardless of whether those features are part of the operating system, as are available to, or used by, that gatekeeper when providing such services.*
>
> Digital Markets Act
> (emphasis added)

"Free of charge" being the key and unambiguous phrase.

For the remaining items, given that it is not possible (or at least very difficult) to distribute to iOS without Xcode (by Apple's design), it is not clear how developers could avoid using these tools. Many developers would love the ability to be able to develop for iOS without being forced to use Xcode or TestFlight, as this would allow developing for iOS without requiring an Apple Mac computer. It seems unreasonable to charge for usage of software libraries when you have, by contract and technical setup, prevented developers from using competing libraries.

Astonishingly, Apple has added wording that you are **unable to switch back to the standard contract** if you are unhappy with the new contract, while still granting themselves the right to change either contract at any time.

> *"Developers who adopt the new business terms at any time will not be able to switch back to Apple's existing business terms for their EU apps. Apple will continue to give developers advance notice of changes to our terms, so they can make informed choices about their businesses moving forward."*
>
> Apple Documentation

Next, it is worth considering how much it would cost for a browser to list on one of these third-party app stores. iOS has 1.46 billion users. So that is 14.6 million users for each 1% browser market share. If a browser vendor dares to list their app on a third-party store, even once, and **even if no one downloads it** from that third-party store, Apple will bill them **$7.3 million per year per 1% market share every year with no recourse** to change back to the previous contract.

It seems clear this is designed to make it as difficult as possible for free apps to list on third-party app stores, depriving alternative app stores of these apps, and preventing their ability to succeed.

Apple should remove the Core Technology Fee entirely, and listing on a third-party app store should not be able punitively change additional fees to a business's apps on the Apple app store in **retaliation** for listing on other app stores on iOS.

Under pressure Apple has added some exceptions to Core Technology Fee including:

- Nonprofit organizations
- Developers who's apps collectively have less than 1 million users.
- A phase-in period of 3 years for developers that hit $10 million+ revenue
- Students, hobbyists, and other non-commercial developers with a no-revenue business.

The last one is important as it is designed to prevent businesses from listing a free app either directly via their website or third party app stores if they have **any** apps with revenue.

None of these concessions solve the underlying problem. Apple is attempting to charge fees on interactions between customers and businesses of which they have no part and provide no service beyond that which comes with their operating system by default. These are apps not installed via Apple's app store on devices that Apple has already sold at incredibly high margins to consumers. This type of behavior is clearly against both the intent and letter of the DMA.

If Apple wishes to make additional revenue from consumers who purchase their devices in the EU, they need to do it by making their app store and their optional services (Safari, iCloud, Apple TV, Apple Music etc) more attractive. They should not attempt to siphon revenue from consumers and businesses which are opting to not use either for particular apps.

Further they should not apply punitive pricing to these same apps on their own app store for daring to list outside of it.

4.3.7.1. The Fair Price for API Access

Even though the DMA explicitly sets the cost of access to these APIs at zero, it is worthwhile discussing what the “fair” price is.

4.3.7.1.1. Value of IP in APIs

Clearly Apple does not believe that they have IP rights over the general category of each of these APIs, otherwise with their over billion USD a year legal budget they would have sued other operating systems for violating their IP (eg Windows, Android, Linux etc).

So the question arises does the IP in Apple's iOS API's actually provide value or is the real value being able to have sufficient access to the operating system hardware and software to provide needed functionality to consumers. It is important to remember that only Apple has control over what APIs are available on iOS and so all of them are by definition made by Apple and likely contain Apple's IP even if they are stock standard and available on every operating system. In a sense it is like Apple is simply charging for the ability to have certain functionality on iOS rather than charging for the value of the IP in those APIs.

One thought experiment to illustrate this is, to imagine for a particular API that Apple also implemented a replica of Androids equivalent for that API for iOS. Developers are then asked whether and how much more they would be willing to pay for Apple's version of the API containing Apple's IP instead of using the Android APIs. If the answer is zero or low then we believe that the amount Apple should be allowed to charge under should be equally low as this would indicate the the value of the IP in this case was low or zero.

4.3.7.1.2. Comparable Licenses

Another avenue for deciding reasonable prices is to look at the comparable licenses charged to other developers. Other operating systems such as Windows, macOS, Linux, ChromeOS and Android all do not charge for access to system APIs. That is the comparable cost of licenses charged to other developers on other platforms is zero.

It is worth noting that iOS also does not currently charge for API access. Nowhere in Apple’s iOS guidelines or in their app store features list do they mention that they are waiving API access fees in return for particular conditions or the App Store 15-30% revenue sharing arrangement.

4.3.7.1.3. Apple Developer Fee

Apple also charges all developers $100 USD per a year to be part of the developer program. With 34 million registered developers, this is a business worth at least $100m a year to Apple, quite plausibly significantly higher than the budget Apple has for API development for iOS. Apple currently plans to charge this developer fee even to developers that do not distribute via their app store but rather distribute directly or via third party app stores.

The fact that Apple already derives considerable income directly from developers wishing to distribute an app on their operating system is a significant factor.

> *"I get it – saying "Spotify pays Apple nothing' is a much stronger lobbying position than 'Spotify pays Apple just $99 a year, the same as every other developer'. But only if it is, you know, true. And Apple makes hundreds of millions of dollars a year from charging that fee to developers as standard, which complicates the narrative that the App Store is only funded by commission on sales.*
>
> *I tried asking Apple how they squared this, and a spokesperson repeated the claim that Spotify paid $0 to Apple. When I asked if I could explicitly write that 'Apple claimed that Spotify is not charged the developer fee', though, the company stopped replying to my emails. Spotify had no such bashfulness, and confirmed that they pay the fee like all major developers."*
>
> Alex Hern - The Guardian

Apple policy personnel are clearly aware of this and have previously directly lied in order to obscure this point.

4.3.7.1.4. Why Apple Should Not be Allowed to Charge for API Access

We believe that charging for access to APIs is deeply problematic for a number of reasons, and may end up being used as a tool to significantly hinder competition by potentially unscrupulous gatekeepers.

In mandating that such API access must be free, we believe that the authors of the DMA had tremendous foresight into the games that large gatekeepers would play were they granted such a weapon to use against those who would contest their hardware and services.

They should not be able to charge for the following reasons:

- Could be a mechanism to hinder competition
- Gatekeepers effectively can't charge themselves
- Face no competitive pressure to lower prices for API access
- Will lock out smaller players
- Consumers not the gatekeepers own their devices
- Difficult to enforce reasonable pricing
- No benefit to consumers but potential for considerable harm
- Is not and will not be the gatekeepers' primary business model

Allowing such fees opens the door to Apple and other gatekeepers to charge unreasonable prices and lock out competition and is in our view a strictly worse option than mandating free access to these APIs.

4.3.7.1.5. The Fair Price is Zero

Given these APIs have no significant value above what is offered on other operating systems, the fact developers are in fact forced to use them via virtue of it being the only choice on iOS, that developers are already paying Apple $100 USD per year and that allowing Apple to charge for this access will lead to considerable harm for consumers by suppressing the contestability of third party services, the fair price for these APIs is zero.

4.3.8. Apple Should Publish a New More Detailed Compliance Plan

The act states in Recital 68:

> *"In addition, a clear and comprehensible non-confidential summary of such information should be made publicly available while taking into account the legitimate interest of gatekeepers in the protection of their business secrets and other confidential information. This non-confidential publication should enable third parties to assess whether the gatekeepers comply with the obligations laid down in this Regulation."*
>
> Digital Markets Act - Recital 68

Apple's compliance report is woefully lacking in detail relative to the reports created by the other gatekeepers. Key parts of Apple's compliance plan are not mentioned even in passing, for example phrases such as "Core Technology Fee" or "CTF" do not appear once in the document.

The lack of detail in Apple's compliance report makes it impossible for *"third parties to assess whether the gatekeepers comply with the obligations laid down in this Regulation"*.

By comparison:

Microsoft	421 pages
Google	271 pages
Meta	57 pages
Bytedance	52 pages
Amazon	32 pages
Apple	**12 pages**

Without a full and detailed compliance report from Apple, it makes it significantly more difficult for us, or any other party, to analyze their compliance.

The commission should consider requiring Apple to release a more detailed compliance report where redactions from their existing confidential submission to the Commission are strictly limited to "business secrets and other confidential information". In the event that their confidential submission is lacking sufficient detail they should be forced to rectify it or be fined.

5. 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.

By standing firm against Apple's malicious compliance and forcing them to comply with their obligations under the act as intended, the Commission can improve interoperability, contestability, and fairness leading to lower priced and higher quality apps— not only for the EU but for the entire world.

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

6. References

The Digital Markets Act
https://eur-lex.europa.eu/legal-content/EN/TXT/?toc=OJ%3AL%3A2022%3A265%3ATOC&uri=uriserv%3AOJ.L_.2022.265.01.0001.01.ENG

MOBILE ECOSYSTEMS MARKET STUDY - APPLE RESPONSE TO INTERIM REPORT
https://assets.publishing.service.gov.uk/media/62277271d3bf7f158779fe39/Apple_11.3.22.pdf

Mozilla says Apple's new browser rules are 'as painful as possible' for Firefox
https://www.theverge.com/2024/1/26/24052067/mozilla-apple-ios-browser-rules-firefox

HTML 5 poses a threat to Flash and the App Store
https://www.patentlyapple.com/2021/05/in-the-epic-vs-apple-trial-today-epic-revealed-apple-memos-discussing-whether-the-70-30-split-with-developers-would-stand.html

OWA - It's Official, Apple kills Web Apps in the EU
https://open-web-advocacy.org/blog/its-official-apple-kills-web-apps-in-the-eu/

OWA - Apple Backs Off Killing Web Apps
https://open-web-advocacy.org/blog/apple-backs-off-killing-web-apps/

Mobile ecosystems market study interim report
https://www.gov.uk/government/publications/mobile-ecosystems-market-study-interim-report

Apple sales fall in nearly all countries
https://www.bbc.com/news/articles/c99zxzjqw2ko

No one's talking about the Apple Vision Pro anymore
https://mashable.com/article/apple-vision-pro-sales-drop

Apple's App Store grossed more than $85 billion in 2022
https://www.cnbc.com/2023/01/10/apple-app-store-revenue-update-shows-slowing-growth-.html#:~:text=If%20all%20developers%20paid%20a%2030%25%20cut%20to,billion%20in%202022%2C%20based%20on%20a%20CNBC%20analysis.

$18B-20B in annual payments from Google to Apple

https://www.theregister.com/2023/10/10/google_pays_apple_18_20_claims_bernstein/#:~:text=%22We%20estimate%20that%20the%20ISA,of%20Apple's%20annual%20operating%20profits.%22

DOJ Complaint against Apple
https://www.justice.gov/opa/media/1344546/dl?inline

Apple says iMessage on Android 'will hurt us more than help us
https://www.theverge.com/2021/4/9/22375128/apple-imessage-android-ecosystem-lock-in-epic-games-filings-app-store-dispute

OWA - Japan ends the #AppleBrowserBan
https://open-web-advocacy.org/blog/japan-ends-the-apple-browser-ban/

OWA - New Digital Competition Laws for Australia
https://open-web-advocacy.org/blog/new-digital-competition-laws-for-australia/

OWA - Open Web Advocacy 2023 in Review
https://open-web-advocacy.org/blog/owa-2023-review/#korea%2C-brazil%2C-india

OWA - UK passes Digital Markets, Competition and Consumers Bill
https://open-web-advocacy.org/blog/uk-passes-dmcc/

Web Browser Engine Entitlement Addendum for Apps in the EU
https://developer.apple.com/contact/request/download/web_browser_engine.pdf

Apple Developer Documentation - Entitlements
https://developer.apple.com/documentation/bundleresources/entitlements

OS X/iOS Entitlement Database
https://newosxbook.com/ent.jl?osVer=iOS17&exec=/Applications/MobileSafari.app/MobileSafari#:~:text=com.apple.bluetooth.internal%3A%20true

Hardware and the web: the balance between usefulness, security and privacy
https://nielsleenheer.com/articles/2021/hardware-and-the-web-the-balance-between-usefulness-security-and-privacy/

Eight things Apple could do to prove it actually cares about App Store users
https://www.theverge.com/2021/9/3/22654197/apple-app-store-editorial-scams-refunds

Epic skirts Apple's 30% commission fee by implementing 'direct' payments
https://appleinsider.com/articles/20/08/13/epic-skirts-apples-30-commission-fee-by-implementing-direct-payments

Apple’s head of fraud on “App Store Review”
https://embed.documentcloud.org/documents/21043943-2016-january-friedman-how-many-apps-through-pipe/#document/p1

XcodeGhost Malware Infected 100+ Million iOS Users and Apple Said Nothing
https://www.intego.com/mac-security-blog/xcodeghost-malware-infected-100-million-ios-users-and-apple-said-nothing/

Apple pays hackers six figures to find bugs in its software. Then it sits on their findings.
https://www.washingtonpost.com/technology/2021/09/09/apple-bug-bounty/

Project Zero team at Google - Vulnerability Disclosure FAQ
https://googleprojectzero.blogspot.com/p/vulnerability-disclosure-faq.html#:~:text=%22This%20bug%20is%20subject%20to%20a%2090%20day%20disclosure%20deadline.%20If%20a%20fix%20for%20this%20issue%20is%20made%20available%20to%20users%20before%20the%20end%20of%20the%2090%2Dday%20deadline%2C%20this%20bug%20report%20will%20become%20public%2030%20days%20after%20the%20fix%20was%20made%20available.%20Otherwise%2C%20this%20bug%20report%20will%20become%20public%20at%20the%20deadline.%22

Apple Developer Program License Agreement
https://developer.apple.com/support/terms/apple-developer-program-license-agreement/

Apple App Review Guidelines
https://developer.apple.com/app-store/review/guidelines/

Apple - BrowserEngineKit
https://developer.apple.com/documentation/BrowserEngineKit

Open Web Competition Platform - Alternate browser engines on iOS must be able to bring their own rendering architecture
https://github.com/OpenWebAdvocacy/OpenWebCompetitionPlatform/issues/11

Interop - Extend Scrolling to cover Body Scroll Issues
https://github.com/web-platform-tests/interop/issues/84

Apple’s app tracking triggers statement of objections from French competition authority
https://techcrunch.com/2023/07/25/apple-att-antitrust-france/

When you ‘Ask app not to track,’ some iPhone apps keep snooping anyway
https://www.washingtonpost.com/technology/2021/09/23/iphone-tracking/

UK’s CMA Raises Concerns, May Delay Google’s Third-Party Cookie Phase-Out
https://www.pymnts.com/cpi-posts/uks-cma-raises-concerns-may-delay-googles-third-party-cookie-phase-out/

UK’s CMA - Decision to accept commitments offered by Google in relation to its Privacy Sandbox Proposals
https://assets.publishing.service.gov.uk/media/62052c52e90e077f7881c975/Google_Sandbox_.pdf

Apple's Dedication To Privacy Is Missing When It Comes To Its Workforce, Employees Say
https://www.techdirt.com/2021/08/31/apples-dedication-to-privacy-is-missing-when-it-comes-to-workforce-employees-say/
Cory Doctorow - Apple has a tactical commitment to your privacy
https://twitter.com/doctorow/status/1459914164152016905

Censorship, Surveillance and Profits: A Hard Bargain for Apple in China
https://www.nytimes.com/2021/05/17/technology/apple-china-censorship-data.html

Apple - Privacy
https://www.apple.com/au/privacy/

Apple: The Story Behind Its New Ad Offerings To Retailers, Restaurants, Hotels, Other Location-Based Businesses
https://www.forbes.com/sites/daviddoty/2023/02/08/apple-the-story-behind-its-new-ad-offerings-to-retailers-restaurants-hotels-other-location-based-businesses/#:~:text=Apple%20seems%20to%20be%20growing,coming%20up%20fast%20if%20quietly.)

Apple - App Store & Privacy
https://www.apple.com/legal/privacy/data/en/app-store/

iOS 15 now prompts users if they want to enable Apple personalized ads, after it was previously on by default

https://9to5mac.com/2021/09/02/apple-personalized-ads-targeting-ios-15/#:~:text=For%20iOS%2015%20users%2C%20Apple%20has%20begun%20prompting,well%20as%20for%20targeting%20App%20Store%20Search%20Ads.

UK CMA's - Mobile ecosystems - Market study final report
https://assets.publishing.service.gov.uk/media/63f61bc0d3bf7f62e8c34a02/Mobile_Ecosystems_Final_Report_amended_2.pdf

Apple's Culture of Secrecy
https://www.nytimes.com/2008/07/26/business/26nocera.html

Apple Imposes NDA For App Store Rejections
https://www.wired.com/2008/09/apple-imposes-n/

'It's concerning': Tech firms are using NDAs to illegally muzzle whistleblowers by threatening to sue them for talking, SEC says
https://fortune.com/2023/04/18/tech-nda-illegally-muzzle-whistleblowers-sec/

Apple execs discuss why the Mac App Store has not been successful in internal email
https://applescoop.org/story/apple-execs-discuss-why-the-mac-app-store-has-not-been-successful-in-internal-email

Apple - Notarized Applications for macOS
https://developer.apple.com/support/terms/apple-developer-program-license-agreement/#ADPLA5.3

Mozilla - "Can browser choice screens be effective?" paper
https://research.mozilla.org/files/2023/09/Can-browser-choice-screens-be-effective_-Mozilla-experiment-report.pdf

OWA - Apple's one weird trick to stop you changing your default browser
https://open-web-advocacy.org/blog/apples-one-weird-trick-to-stop-you-changing-your-default-browser/

Report: People are bailing on Safari after DMA makes changing defaults easier
https://arstechnica.com/tech-policy/2024/04/report-people-are-bailing-on-safari-after-dma-makes-changing-defaults-easier/?comments=1

OWA - App Clips
https://open-web-advocacy.org/walled-gardens-report/#app-clips

OWA - Smart App Banners

https://open-web-advocacy.org/walled-gardens-report/#smart-app-banners

Mozilla Documentation beforeinstallprompt
https://developer.mozilla.org/en-US/docs/Web/API/Window/beforeinstallprompt_event

Apple says if you want to ship your own iOS browser engine in EU, you need to be there
https://www.theregister.com/2024/05/17/apple_browser_eu/

Rick Byers - On how Chrome can't test their own browser on iOS
https://twitter.com/RickByers/status/1791609238000705824

Platform Tilt - Beta testing on iOS
https://github.com/mozilla/platform-tilt/issues/16

CMA plans market investigation into mobile browsers and cloud gaming
https://www.gov.uk/government/news/cma-plans-market-investigation-into-mobile-browsers-and-cloud-gaming#:~:text=We%20all%20rely,chance%20to%20compete.

Apple - On removing Web App Support
https://developer.apple.com/support/dma-and-apps-in-the-eu#8

Apple Is Not Defending Browser Engine Choice
https://infrequently.org/2022/06/apple-is-not-defending-browser-engine-choice/#:~:text=This%20split%20experience%20causes%20a%20sort%20of%20pervasive%20forgetfulness%2C%20making%20the%20web%20less%20useful.

Digital Markets Act - Interventions In-App Browsers
https://open-web-advocacy.org/files/OWA%20-%20DMA%20Interventions%20-%20In-App%20Browsers%20v1.2.pdf

Platform Tilt - Browser extension support on iOS
https://github.com/mozilla/platform-tilt/issues/15

W3C - Payment Request API
https://www.w3.org/TR/payment-request/

WebKit - standards-positions - Safari will be locked to Apple Pay
https://github.com/WebKit/standards-positions/issues/30

Spotify says its iPhone app updates in the EU are getting held up by Apple
https://www.theverge.com/2024/3/14/24100944/spotify-ios-app-update-eu-apple-dma

EU fines Apple €1.8bn over App Store restrictions on music streaming
https://www.theguardian.com/business/2024/mar/04/eu-fines-apple-18bn-over-app-store-restrictions-on-music-streaming#:~:text=EU%20fines%20Apple%20%E2%82%AC1.8bn%20over%20App%20Store%20restrictions%20on%20music%20streaming,-This%20article%20is&text=Apple%20has%20been%20fined%20%E2%82%AC,streaming%20services%20such%20as%20Spotify

App stores, trust and antitrust
https://www.ben-evans.com/benedictevans/2020/8/18/app-stores

Samantha John - CEO Hopscotch on App Store Review
https://twitter.com/samj0hn/status/1431001795904561160

Apple - Notarizing macOS software before distribution
https://developer.apple.com/documentation/security/notarizing_macos_software_before_distribution

Apple - App Review 90% in less than 24 hours
https://developer.apple.com/distribute/app-review/#:~:text=On%20average%2C%2090%25%20of%20submissions%20are%20reviewed%20in%20less%20than%2024%C2%A0hours
.

Apple - Notarizing macOS software typically less than 1 hour
https://developer.apple.com/documentation/security/notarizing_macos_software_before_distribution/customizing_the_notarization_workflow#:~:text=After%20you%20upload%20your%20app%2C%20the%20notarization%20process%20typically%20takes%20less%20than%20an%20hour.

Google Project Zero - Patch Gap
https://googleprojectzero.blogspot.com/2022/02/a-walk-through-project-zero-metrics.html

How long does an iOS Update Take - 30 mins
https://www.ubackup.com/phone-backup/how-long-does-an-ios-update-take.html#:~:text=Updating%20to%20the%20latest%20iOS%20takes%20about%2030%20minutes%20on%20average.

Apple - About alternative app distribution in the European Union

https://support.apple.com/en-us/118110#:~:text=If%20you%20leave%20the%20European,apps%20from%20alternative%20app%20marketplaces.

The Schengen area
https://home-affairs.ec.europa.eu/system/files_en?file=2020-09/schengen_brochure_dr3111126_en.pdf

ESTA - Electronic System for Travel Authorization
https://esta.cbp.dhs.gov/esta

You will usually be considered tax-resident in the country where you spend more than 6 months a year
https://europa.eu/youreurope/citizens/work/taxes/income-taxes-abroad/index_en.htm#:~:text=you%20will%20usually%20be%20considered%20tax%2Dresident%20in%20the%20country%20where%20you%20spend%20more%20than%206%20months%20a%20year

Apple - Core Technology Fee
https://developer.apple.com/support/core-technology-fee/

Apple - Update on apps distributed in the European Union
https://developer.apple.com/support/dma-and-apps-in-the-eu/

Apple - App Store Features
https://developer.apple.com/app-store/features/

How much does Spotify really pay Apple?
https://www.theguardian.com/technology/2024/mar/05/techscape-open-source-software-cryptocurrency

Apple’s Non-Confidential Summary of DMA Compliance Report
https://www.apple.com/legal/dma/dma-ncs.pdf

7. 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