

# Browser Choice and the Future of Open App Development

Matthew Thomas

# Table of Contents

<b>Browser Choice and the Future of Open App Development</b>	<b>1</b>
<b>Table of Contents</b>	<b>2</b>
<b>Browser Choice and Competition</b>	<b>4</b>
iOS and Competing Browsers	5
The Financial Harm Argument	7
Impacts of the 30% AppStore Fee	7
Costs of Native App Lock-in	7
Building Native Apps vs Universal Web-Apps	8
Summary	8
<b>Suggested Solutions</b>	<b>8</b>

**The open web is under threat** and its great future potential is being heavily limited by abuse of Apple's dominant market position.

As a set of free and open standards, the open-web has allowed people to create and distribute content and has changed the way we communicate and organize information. The web is the ideal platform for free market competition, offering a level-playing field which lowers costs for consumers and provides a direct link from businesses to users unencumbered by fees and anti-competitive restrictions. It has underpinned the creation of some of the world's largest companies and been the driver of huge economic growth. Extraordinary progress has been made in just 25 years from basic humble websites to powerful systems that manage, store and organize enormous amounts of data.

However the emergence of gated aftermarkets, their negative influence and incentives has stalled the once rapid progress towards this free, universal development and distribution platform. Apple hides behind claims of extra security and privacy when in fact their restrictions deprive the consumer of choice and serve primarily to lock in users with their data and purchases, prevent them from moving to competing platforms and hamper interoperability all while applying a heavy tax.

Apple has in effect **banned**<sup>1</sup> competing Web-Browsers from their mobile devices and **limited** functionality essential for building and distributing applications written with free and open web technology. In the absence of competition Apple has restricted the feature set and integration of their own browser to push iOS developers and users towards the gated ecosystem of the AppStore, protecting their captive market from competition and increasing friction.

Without the heavy restrictions applied by Apple, the open web offers an alternative future where the control from corporate intermediaries along with their fees are replaced with consumer choice and the freedom to easily shift from one platform to another. Instead of having to write multiple separate applications for every device, it allows developers to build their application once and have it work on all consumer devices be it desktop, laptop, tablet or phone.

The reduction in development cost alone will result in new applications, greater competition and ultimately lower costs for consumers. It frees consumers and businesses from the gatekeepers who control the platform.

Had the abuse of dominant market position never occurred, namely the ban on browsers and stagnation of key and important functionality and integration, this platform would be available today. We propose **two key remedies**, reverse Apple's ban on competing browsers and compel them to provide full integration and functionality for apps built with open web technology.

The importance of the open-web for competition can not be understated and the entire software development market depends on the decisions made by government bodies and legislators in the coming months.

---

<sup>1</sup> The ban is on browser engines is discussed in detail in "iOS and Competing Browsers"

# Browser Choice and Competition

On iOS, the **only** browser that is allowed is Safari<sup>2</sup> and all other browsers are **banned**<sup>3</sup>. The majority of iOS users are unaware that when they are installing an alternative browser such as Firefox, Edge or Chrome that they are in fact just installing Safari but with a different “skin” and branding. iOS appears to the end user to allow multiple browsers, but Apple specifically prohibits the use of any engine except for Safari’s webkit and JavascriptCore engine and since the engines are what provides most features and functionality, a ban on third-party engines can be considered a ban on third party-browsers.

None of the other major operating systems impose a ban on third-party browsers including Microsoft Windows, Android and Apple’s own MacOS but somehow Apple has thus far managed to evade regulatory oversight. There is some technical nuance to this ban that will be explored later in this section.

The restriction to only a single browser means that there is no meaningful competition between browsers, developers are limited by the functionality Apple decides to build and users are deprived of choice.

Browser choice is what drives the technology forward which ultimately results in better, faster, more reliable software for users. Microsoft’s IE6 was once the dominant browser with a 95% market share<sup>4</sup> due to its pre-installation on Windows. Without competition on the Windows platform, browser development remained stagnant for years until Firefox’s market share triggered Microsoft to start investing in browsers once again. At no point did Microsoft ban competing browsers as Apple has done.

Banning other browsers is a significant issue since:

1. Removes all competition within Apple’s aftermarket ecosystem
2. Removes user choice
3. User’s are unaware of this lack of choice
4. Increases Cost of Applications.
  - a. Lack of Browser choice leads to
  - b. Reduced Functionality and Stability
  - c. Forces developers to use the AppStore instead
  - d. Increases cost of development is at least double + the AppStore tax leads to a minimum 42% increase in cost to the end user
5. No incentive for Safari to add features because there is no competition
6. Apple is the only body that decides which features and functionality are included

---

<sup>2</sup> All browsers are required to use Safari’s internal framework Webkit and Safari’s internal javascript engine webkit as is provided by Apple.

<sup>3</sup> Apple’s AppStore review “guidelines” states that Apps must use use Safari’s internal framework Webkit and Safari’s internal Javascript Engine

<sup>4</sup> "Usage share of web browsers - Wikipedia." [https://en.wikipedia.org/wiki/Usage\\_share\\_of\\_web\\_browsers](https://en.wikipedia.org/wiki/Usage_share_of_web_browsers). Accessed 23 Jun. 2021.

7. Disincentive for Safari to add features as it would compete with the AppStore of which they take a 30% cut
8. Other Browsers can't differentiate themselves from the competition, thereby eliminating normal market competition.
9. Apple through Safari can extract up-to 12 billion USD<sup>5</sup> per year from search-engine licensing agreements, whereas competing browsers like Firefox see their revenues decreased.

## iOS and Competing Browsers

The two main components of a browser are the "Browser Engine" and the "Javascript Engine". The combination of the two make up the majority of the browser. The User Interface in terms of functionality both to users and developers is a very minor part.

The internal engines of each browser are:

Browser	Browser Engine	Javascript Engine
Safari (Apple)	Webkit	JavaScriptCore
Firefox (Mozilla)	Gecko	Spidermonkey
Chrome (Google)	Blink	V8
Edge (Microsoft)	Blink	V8
Internet Explorer	Trident	IEJavascript (IE4 - IE9) Chakra (IE9 - IE11)

Each browser, if they are available use the engines listed above across Windows, MacOS, Linux and Android, however iOS has banned other engines through the following clause:

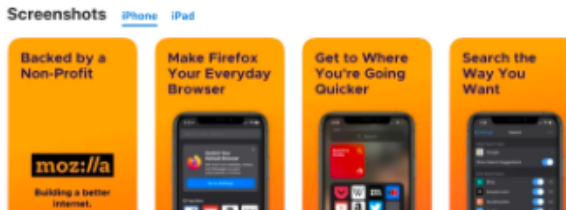
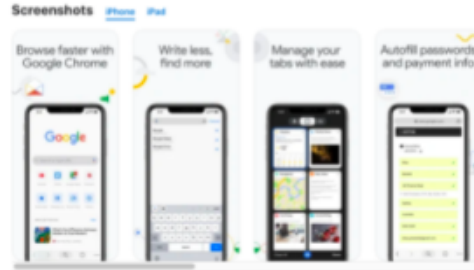
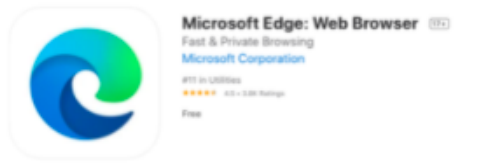
### **2.5.6 Apps that browse the web must use the appropriate WebKit framework and WebKit Javascript.**<sup>6</sup>

This ban forces all of the other browser makers to get Safari's internals (Webkit and Core Javascript), and apply a new skin and branding for their apps on iOS. Every browser installed on iOS therefore is restricted to the functionality and features of Safari. Neither Windows, MacOS, Linux or Android applies such a restriction.

This gives an illusion of choice, users and even a significant number of experienced web developers are unaware that there is in essence only one browser.

<sup>5</sup> "CNBC – Apple Steps Up Effort to Build Google Search Alternative." 28 Oct. 2020, <https://www.cnbc.com/2020/10/28/apple-steps-up-effort-to-build-google-search-alternative.html>. Accessed 23 Jun. 2021.

<sup>6</sup> "App Store Review Guidelines - Apple Developer." <https://developer.apple.com/app-store/review/guidelines/>. Accessed 22 Jun. 2021.



## Browsers shown in the Apple AppStore

To truly allow browser competition Apple would have to open up a wide set of low-level APIs that are currently blocked for use from all other apps and allow other browsers to use their own javascript engines and own rendering engines. The lack of competition on the iOS system.

# AppStore Revenue

It is not possible to know exactly why Apple has underfunded Safari and banned alternatives web-browsers but the flow-on effects are clear.

Many have speculated that it is to protect AppStore revenue. Despite Apple's marketing claiming a thriving AppStore marketplace it recently came to light in the Epic vs Apple trial that 72% of all AppStore revenue comes from free to play games.

To quote <https://www.ben-evans.com/benedictevans/2021/7/8/app-store>

*80% of that was from games, mostly in the US and north-east Asia, and mostly on iPhone. There's no clear reason to think the proportions have changed much since then, except that China is probably bigger (Apple had only just added support for Alipay in 2016). So, this is mostly games, and, from other disclosures, over 90% free-to-play.*

This would indicate that the majority of revenue comes from mobile gaming whales, which has parallels with problematic gambling.

*A mobile gaming whale is someone who spends a lot of microtransactions. So-called "whales" are the main target for microtransactions in free-to-play games, for example; they're the ones who buy booster packs, cosmetics, etc. Tons of them.*

7

Whether or not the motivation is to protect this revenue source, Safari is having profound negative effects. Apple hides behind security and privacy but you can see from the following article that even features with no possible security or privacy concerns are not getting developed. (<https://httptoolkit.tech/blog/safari-is-killing-the-web/>)

All of this comes back to competition. Because Apple has effectively banned off all of the competition they are under no pressure to produce a competitive browser. Despite Apples claim that the AppStore provides security and privacy others have found the review process to be ineffective <sup>8 9</sup>.

---

<sup>7</sup> <https://www.blog.udonis.co/mobile-marketing/mobile-games/mobile-games-whales>

<sup>8</sup> <https://www.theverge.com/2021/4/21/22385859/apple-app-store-scams-fraud-review-enforcement-top-grassing-kosta-elftheriou>

<sup>9</sup>

<https://www.forbes.com/sites/gordonkelly/2021/04/07/apple-iphone-ipad-app-store-scam-warning-new-iph-one-problem/?sh=7231e8a960aa>

# The Financial Harm Argument

## Impacts of the 30% AppStore Fee

For a start, imagine you require \$10 per user to cover the costs of developing, publishing and maintaining an Application. The Application Store that you are selling your App in decides to add a 30% fee. In order to still receive \$10, you now need to charge \$14.2, which is a 42% price increase for the end user. However the actual price increase will be higher as when you increase the price by 42% you will lose a percentage of users as you move to a higher position on the demand curve, causing the equilibrium price to be even higher.

## Costs of Native App Lock-in

Native iOS Apps have to be written in an Apple created language called Swift and use APIs that are specific to iOS. You can not run a Swift app on an Android device. Web-Apps however only have to be written once, in one language and then can run on any device.

Currently it is possible to write the majority of apps as a Web-App for Android devices. It is not possible on iOS devices because Apple has:

- a) banned all browsers other than their own
- b) has not allowed or implemented critical features and APIs for Web-Apps,
- c) not invested in Safari enough to make it stable and reliable

For a developer wishing to target all platforms they are there for constrained to the following choices:

Choice A. iOS (Swift) + Android(Java) + Web/Desktop (Web App)

Choice B. iOS (Swift) + Android(Web App) + Web/Desktop (Web App)

Without the browser ban and proper application integration we would have:

Choice C. Web App for all for platforms

Swift (iOS) and Android (Java) are sufficiently different that the company building the app essentially has to build it twice and spend almost twice as much building the application.

## Building Native Apps vs Universal Web-Apps

Currently we would have to hire at least one Java developer and one Swift developer. However if iOS did not ban other browsers and had proper integration with Web-Apps, we would be able to build a single universal App that could be deployed to both mobile and desktop devices.



Universal Application means less bugs, and easier to keep the application consistent across multiple platforms. Managing a project written in duplicate or triplicate more than doubles/triples the cost due to all the cross compatibility issues.

Think of the example where as the developer you have an Android App and iOS App and a Desktop App and you wish to deploy a new feature. You need 3 separate developers implementing the feature on all 3 platforms, you then have 3 different deployments. With Universal Web-Apps which don't have to be distributed through an App Store you simply need to update one application that could be done by a single developer.

## Summary

Apple's aftermarket monopoly combined with an iOS browser ban and lack of Web-App integration prevents developers from distributing apps outside of Apple's AppStore significantly increases costs to businesses and significantly increases cost to the end user because of at least a doubling in development cost and a greater than 42% increase in fees to the end user to get the same revenue for the developer because of AppStore Charges.

## Suggested Solutions

### 1. **Prevent Operating Systems from banning Web Browsers**

Platform owners should not be allowed to ban or restrict the functionality of any Web Browser in any consumer facing operating system that includes a web-browser except for heavily justified narrow scope protections for the security and privacy of users.

### 2. **Regulate that Web-Apps must be given the same status and functionality as Native Apps across all operating systems**

Platform owners need to develop whatever is necessary for web-apps to compete on an even playing field as native apps. Web-Apps must be easy to install, manage and receive the same privileges and functionality as native Apps. Platforms should not be allowed to prioritize apps distributed by their proprietary platforms. Web-Apps should be free to function as App Stores for other Web-Apps.