Digital Platform Services Inquiry Submission

John Allsopp, 30/07/2021

What value do browsers derive from being pre-installed on an operating system or device?

While Apple recently with iOS14 (well over a decade after the release of the iPhone) finally allowed users to choose a default browser other than their own Safari, this obscures the reality that there is in effect only a **single** browser on iOS and iPadOS devices–Safari.

How can this be when Chrome, Edge, Firefox, Brave, Puma and others browsers can be installed from Apple's AppStore?

Unlike on desktop and other phone and tablet operating systems, each of these browsers (and indeed any browser on iOS or iPadOS) is *required* by Apple to use Apple's browser engine WebKit.

On other platforms these browsers use their own engines, which enables them to innovate in performance, features and to provide their own privacy and security measures.

Apple in being not simply the default, but sole browser engine on iOS has a significant chilling impact on the feasibility of using open, standard web technologies to develop applications not just for iOS, but for any web enabled platform.

Where Apple refuses to implement a feature, because they control a key segment of the market, developers will often decide not to use that feature even if it's available across other platforms.

In short, it gives Apple an effective **veto** over the capability of the Web platform, including features that have been standardised through the W3C with the participation of many vendors.

To what extent does pre-installation and default settings in operating systems affect… competition and consumer choice in the supply of general search services and browsers?

Effect on Consumer Choice

The first order effect on consumer choice is that in essence there is no possible consumer choice in browsers, beyond the superficial user interface developers are able to wrap around the WebKit engine.

The second order effect on choice is a significant lack of innovation in browser capabilities on the predominant mobile operating system in many parts of the world, and lack of support for many key enabling mobile technologies such as NFC and Bluetooth in Web applications. Such technologies are supported in desktop browsers, and on Android devices.

This significantly curtails the use of Web technologies for developing many classes of mobile application, not just on iOS, but on other platforms, as it imposes a significant additional cost of development due to multiple code bases and increased testing.

In short, consumers are denied the choice of using web applications for many common use cases, since the capabilities required to build common mobile apps are restricted to Native Apps available only via Apple’s App Store.

Effect on Developer Choice

While possibly outside the scope of this issues paper, developers are denied the choice of using open, standard web technologies that otherwise enable the vast majority of the world’s devices to be targeted when developing for iOS.

Impacts on developer choice flow to consumer choice, as limiting the capability of developers to build apps for a platform limits the choice of consumers on that platform.

To what extent is it important for a browser to be able to offer a range of extensions and software for consumers?

In many respects the sole purpose of a browser is to offer software (web sites and applications) to consumers. Without this web content, there is no purpose for a Web browser.

Do developers typically develop extensions and/or software for more than one browser? What influences a developer's choice of browser?

The promise of the World Wide Web is the ability to reach a global audience regardless of the device they are using or the type of network (mobile, wireless, LAN, fixed) they are connected to. Developers who have chosen the Web as a platform to develop for do so in order to reach the largest possible audience. While code will typically need to be adapted to the bugs or other shortcomings in specific browsers Web developers will only in rare cases exclusively target a specific browser (or more pertinently browser engine).

Developers will generally continue to 'support' (ensure their code functions) browsers years after their release.

On iOS however, developers have effectively no choice as to which engines to target as there is a single engine, WebKit, mandated by Apple.

Outside iOS devices, the market share of Safari and Webkit is under 10% [https://gs.statcounter.com/browser-market-share/desktop/worldwide]

Is it difficult for developers to switch between developing extensions and/or software between multiple browsers?

In your response, please specify whether you are referring to mobile or desktop devices.

In theory, and in most cases, developing software (web sites and applications) that runs on multiple browsers is relatively straightforward, and typically much more so than developing for multiple distinct operating systems and devices.

This is one of the promises and attractions of the Web for development, and one of its guiding principles. As Tim Berners-Lee, the Web's inventor put it in 2010

> The principle of universality allows the Web to work no matter what hardware, software, network connection or language you use and to handle information of all types and qualities. This principle guides Web technology design [https://www.scientificamerican.com/article/long-live-the-web/].

The same core, standard, technologies (HTML, CSS, JavaScript, Browser Application Programming Interfaces) and code base can largely be used to target all modern browsers and browser engines.

The key areas of difficulty for developers are

1. Browser bugs (incorrect implementations of the standard technologies) for which “work arounds” are required. This was for many years the single biggest challenge to developers as browsers were only relatively rarely updated, and so bugs could take months or years to be fixed. The rise of ‘evergreen’ browsers, that auto-update every few weeks (all but Safari of the modern widely used web browsers are evergreen) has significantly reduced this issue, except in the case of Safari. Because of Apple’s ban on browser engines other than WebKit on iOS this is a significant source of difficulty for developers, especially when coupled with

2. Lack of ubiquitous support for standard web technologies. Because each browser decides which aspects of the core web technologies it implements, and when, developers need to have strategies for incomplete support for a specific technology they might wish to use. Sometimes, it is possible to “polyfill” the lack of support for a feature on a platform, at other times a developer may use a technique called “progrssive enhancement” to provide additional functionality for users of more capable browsers.

3. However, Safari’s lack of support for many of the essential capabilities available to developers of native apps for ios [https://webkit.org/tracking-prevention/#anti-fingerprinting] make entire classes of mobile app impossible on the predominant mobile platform using Web technologies

Technological Innovation

*The ACCC invites views on technological change or innovation that **may affect the supply of browsers** and/or search services in Australia in the future.*

Questions for market participants
15) Are there technological changes that will affect the supply of browsers, search services and/or the device ecosystem in:

a) Australia?
b) markets outside of Australia?

Web as the predominant computing platform

The Web is now the predominant way in which people use desktop computers, with 60% or more of the time users spend on a desktop being in the browser.

This outcome is a function of the maturity of web browser technologies, and of browser competition which over the last decade or so saw extraordinary innovation in browser engines, in terms of performance, and capabilities.

This innovation has enabled a new class of software businesses, “software as a service”, delivered via the browser rather than via an application, and driven competition and innovation in ecommerce (people buy from Amazon and Shopify via web sites, not native apps), and elsewhere.

There is no reason this trend of a move away from native apps toward browser based apps would not continue on mobile devices and tablet, *unless innovation is restrained by the owners of those platforms.*

There is the genuine risk that Apple’s complete control over browsers on iOS, the predominant mobile platform in Australia and many other countries and regions around the world, will undermine this long term trend.

Micropayments

The first wave of monetisation on the Web was advertising, with many attendant negative outcomes.

In recent years, some major media organisations have switched their business model largely to subscriptions, but beyond a small number of major publishers in major economies, and niche focused publishers, subscriptions appear unlikely to sustain publication businesses.

One promising emerging business model involves very low cost, frictionless microtransactions, such as Coil, enabled by technologies like the open Interledger Protocol. These work by “streaming” payments to a web site from the reader via their browser while they read content (or otherwise engage with the site).

The Puma web browser supports Coil natively, though not on iOS, as Apple’s requirement to use WebKit as the browser engine for any iOS browsers means this feature cannot be implemented on iOS.

Again, Apple’s ability to veto browser engine innovation on their platform has the potential risk of killing off potential business model innovation.