

EXECUTIVE SUMMARY

It was only five years ago that we were debating which technology - HTML5, Silverlight or Flex, should be the preferred option for developing enterprise web-based applications¹. At the time HTML5 showed real promise, but the more mature plugins were the preferred option for building complex applications. Few could have predicted that just a few years later HTML5 would win the battle, with plugins universally viewed as a 'dead' technology.

In the years in between, we've seen another heated technology debate emerge, this time between native and HTML5 mobile technologies. While HTML5 has clearly 'won' the web, the mobile technology debate still rages.

The continued debate hasn't halted the rise of HTML5 and JavaScript; they are now considered viable technologies for both server-side development, via Node.js, and also desktop application development; in both cases pushing out the incumbent technologies, which have been growing stale.

This paper looks at the many opportunities HTML5 offers for desktop development, including innovative UI technologies, accelerated application development, evergreen deployment and the prospect of a single technology for app development, regardless of the platform (be it form-factor or operating system). It also looks at how the collaborative nature of HTML5 is pushing the technology forward in a way that makes it hard, if not impossible, for others to compete.

If you haven't previously considered HTML5 as a desktop technology, it's time to rethink your strategy...



"We are fully committed to HTML5 across both our web and desktop applications at Saxo Bank. We believe it's not just the best option; it's the only option."

Arne Henningsen, Head of IT FX & B2B, Saxo Bank.

¹ <http://www.scottlogic.com/knowledge/white-papers/>

A HISTORY OF 'APP' DEVELOPMENT

DESPITE THE TERM 'APP' BEING RELATIVELY NEW, HAVING BEEN POPULARISED BY SMARTPHONES, WITHIN THE ENTERPRISE WORLD WE'VE BEEN DEVELOPING 'APPS' FOR DECADES. WHILE THE WEB IS A GENERIC PLATFORM THAT DISPLAYS A DIVERSE RANGE OF CONTENT, APPS ARE FOCUSED ON SPECIFIC AND OFTEN COMPLEX USER TASKS.



The technologies used to build apps has transformed considerably over the past few years:

Circa 2010, Microsoft technologies dominated, with one solid product, Windows Forms, being replaced by another, Windows Presentation Foundation (WPF). With WPF, Microsoft hoped to engage both developers with its elegant core, and designers with Expression Blend. Meanwhile Java Swing, while significantly better than its predecessor, had lost much of its market share to Microsoft's superior products, or as Forrester put it, 'Java bungled the presentation layer'². At this point in time, the use of web technologies for serious business applications was gaining momentum under the umbrella term Rich Internet Applications (RIA), with the Flex plugin dominating.

Fast-forward five years and WPF has successfully replaced Windows Forms and become the de-facto technology for enterprise app development. However, many feel that Microsoft has been neglecting this technology³ as the company's attention turned to Windows Store, Azure and Windows Mobile. WPF is a mature and stable

technology, but the performance expectations were never fully met. Meanwhile, Oracle intends to use JavaFX (the failed plugin technology) to replace Java Swing (the failing desktop technology), further cementing their irrelevance! On the web, plugins, together with the term RIA have been consigned to the past, as HTML5 reigns supreme.

Desktop technologies are still widely used for enterprise applications. However, their continued use presents a number of challenges:

- Deployment and delivering updates using desktop technologies within regulated environments is, and always has been, an issue
- There is a lack of developer engagement among desktop teams due to the stagnation of desktop technologies
- Desktop technologies do not have the UI richness of HTML5
- Desktop technologies lack reach, whilst internal users are increasingly mobile. Their expectations are also changing; cloud-based, device-agnostic applications are becoming the norm.

Finally, a split in technologies, no matter which technologies, results in silos. This impedes movement between teams and creates an inflexible workforce.

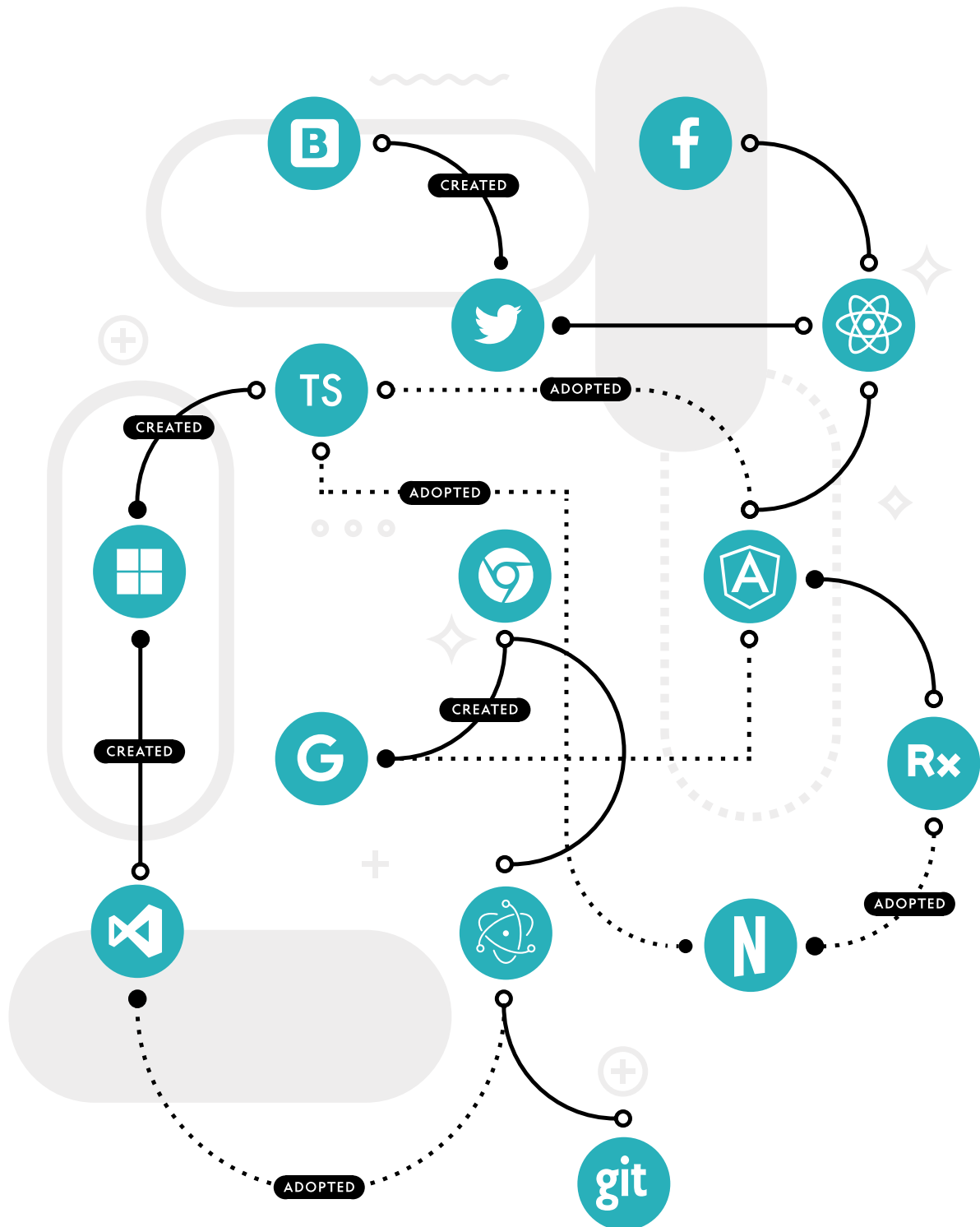


² http://blogs.forrester.com/mike_gualtieri/10-11-23-java_is_a_dead_end_for_enterprise_app_development

³ <http://pragmateek.com/is-wpf-dead-the-present-and-future-of-wpf/>

COLLABORATION DRIVES INNOVATION

DESKTOP TECHNOLOGIES ARE ENTIRELY VENDOR OWNED AND MANAGED. WPF IS MICROSOFT'S TECHNOLOGY AND SWING IS ORACLE'S. IN CONTRAST, HTML5 IS A COLLABORATIVE EFFORT WITH THE TECHNOLOGY BEING PUSHED FORWARD BY A DIVERSE RANGE OF COMPANIES AND INDIVIDUALS.



Large technology companies are putting a considerable amount of resource behind HTML5, resulting in significant contributions, many of which can be used in conjunction. The following are just a few recent innovations:

- Microsoft - developed TypeScript, a superset of JavaScript with static typing, that lends itself well to large-scale app development. TypeScript has recently been adopted by Google's Angular 2.0 team and has deep support for React's JSX syntax.
- Google - created the Chrome browser that has set the benchmark that all other browsers aspire to. Google have also been the leading force behind many web technologies, including Angular.
- Facebook - developed React that has revolutionised the way developers write their UI code. Its functional programming concepts have influenced a great many other frameworks, notably Google's Angular 2.0.
- Netflix - adopted TypeScript and rewrote the popular RxJS library which is itself a core component of Angular 2.
- Mozilla - continues to push web technologies both with its browser and quality documentation in the form of the Mozilla Developer Network, widely recognised as the most complete and detailed source of information on a wide range of HTML5 topics. Mozilla has also adopted React, whilst rebuilding their browser's developer tools.
- Twitter - developed Bootstrap, which brought the concept of CSS reuse to the mainstream. Now on its fourth version it is very widely adopted. Twitter also uses React for its mobile website.

The level of investment in HTML5 is considerable, but just as notable is the level of collaboration. Examples include Google adopting Microsoft technology (Angular 2 uses TypeScript), and Microsoft using GitHub's technology (Microsoft VSCode is built on Electron), with engineering teams often working side-by-side.

The combined efforts of the companies listed above, together with the thousands of smaller companies and individuals, has led to the considerable duplication of ideas, healthy competition and ultimately, innovation. Ideas, concepts and techniques rapidly evolve as they percolate through the open source world of HTML5. As a result, web development is improving exponentially.

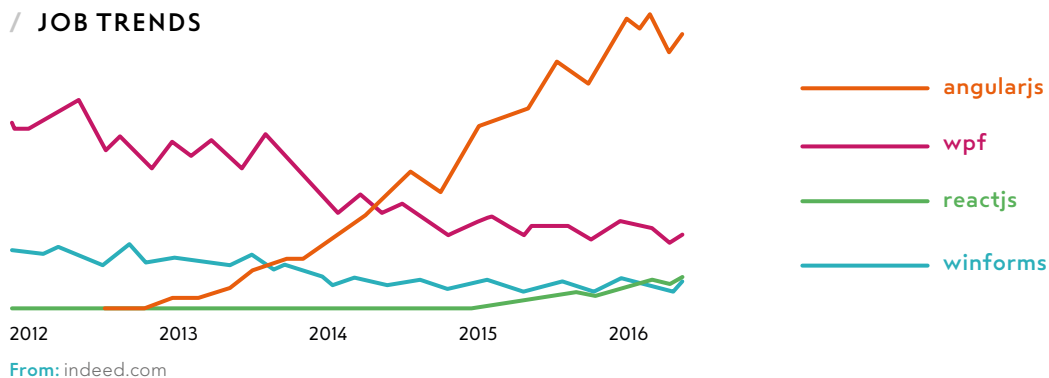
The tangible benefits of this collective innovation are manifold:

- Newer and better frameworks and libraries that are continually evolving
- Better browser technology - notably Internet Explorer is no longer lagging behind
- Better development tooling, with novel concepts such as hot reload⁴ and time travel⁵
- Innovation attracts developers and as a result, there is an ever growing number of skilled developers using these technologies.

The points above add up to better productivity and better products. The collaborative effort that is behind the HTML5 ecosystem is something that single vendor solutions, such as WPF, have a hard time competing with.

As a direct result of the above, there has been a considerable rise in demand for skilled HTML5 developers, while traditional desktop technologies are in demise.

/ JOB TRENDS



⁴ <http://blog.scottlogic.com/2016/01/27/a-case-for-hot-reloading.html>

⁵ <http://blog.scottlogic.com/2016/01/25/angular2-time-travel-with-redux.html>

HTML5 TAKES ON THE DESKTOP

The use of HTML5 as a desktop technology is a relatively recent concept, but it is rapidly gaining momentum and maturity.

The basic mechanics of running an HTML5 application on the desktop are quite straightforward. The app runs within a container which provides a JavaScript runtime and an HTML rendering engine. In many ways it is just a stripped down browser.

As well as providing a JavaScript runtime environment, the container provides APIs that allow the application to integrate with the desktop environment:

- Desktop notifications
- Interacting with low-level hardware
- System level menus
- Multi-monitor support
- Directly interact with the file system

These APIs are often operating system agnostic, for example by allowing the developer to create notifications for both Windows and Mac from the same codebase. This architecture is very similar to Cordova / PhoneGap, which allows developers to write cross platform HTML5 applications for mobile.

Desktop container technology has enjoyed the same collaboration as other HTML5 technologies, with the third generation wrappers representing the combined efforts of many companies and individuals.

As the technology has matured, its use has become mainstream. A number of high profile desktop applications are now written using HTML5:

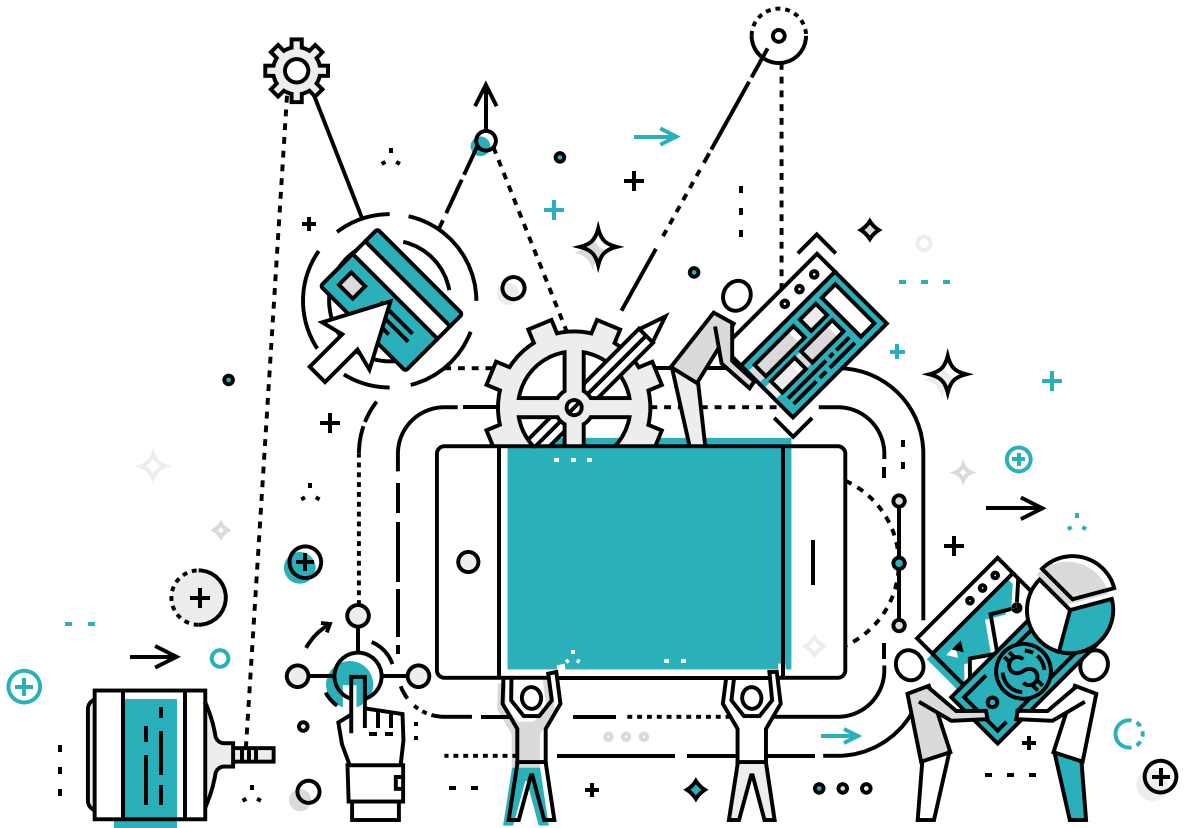
- GitHub Atom
A very popular open source editor
- Visual Studio Code
Microsoft's open source editor
- Brave
A privacy-focused web browser
- Slack
A popular team-based messaging service

As well as these, there are numerous financial-services applications being built using this technology by companies including ICAP, Tullett Prebon, Trading Technologies, REDI and Green Key - in these cases using the OpenFin container⁶. Furthermore, the Symphony Foundation, which is seeking to drive open source software within finance, has a dedicated Desktop Wrapper Working Group⁷.

The applications mentioned above require both security and performance, both of which can be delivered using HTML5. With the third generation of wrapper technology, creating HTML5 applications for the desktop is now just as easy as creating them for the web.

⁶ <https://openfin.co/>

⁷ <https://symphony.foundation/>



/ EVOLUTION

The first generation of HTML5 desktop applications were technically complex, built directly on top of Chromium Embedded Framework (or one of its language wrappers such as CEFSharp). Building a desktop application with these technologies required the use of multiple languages (C++, C#, JavaScript) and the desktop integration layer had to be handcrafted.

As the use of HTML5 on the desktop started to become more mainstream with the release of both Adobe's Brackets and GitHub's Atom in 2014, development of this class of application became much easier. Open source frameworks such as node-webkit, brackets-shell and atom-shell provided a level of abstraction around CEF, and added desktop integration.

More recently a third generation of wrapper technology has emerged, with Atom Shell evolving to become Electron. This framework provides developers with access to the Node.js APIs and ecosystem, automated updates, windows installers, and much more. OpenFin, an HTML5 container focused on the needs of the financial community, has also moved to an electron-based stack⁸. OpenFin provides additional integration for .NET, Flash and Java, an inter-process communication bus, and fixes some of the significant security issues present in Electron⁹.

⁸ <https://openfin.co/2015/12/07/why-were-moving-to-electron/>

⁹ <http://blog.scottlogic.com/2016/06/01/An-update-on-Electron-Security.html>



CONCLUSIONS

A move towards using HTML5 for desktop development gives access to a great many innovative technologies, frameworks and techniques.

The ultimate, and altogether achievable goal, is for your development team to be able to quickly and easily move between projects, technologies and platforms, making use of modern tooling and productivity. This can be succinctly expressed by a subtle twist on Sun's original (and failed) mantra, write once, run anywhere:

LEARN ONCE, USE EVERYWHERE

It looks like traditional desktop technologies are destined to follow the same fate as plugins, ultimately being replaced by HTML5. How can a technology created by a single company compete with the collaborative efforts of hundreds of enterprises, thousands of startups and tens of thousands of individuals?

If you haven't previously considered HTML5 as a desktop technology, it's time to rethink your strategy...

AND WHAT OF MOBILE DEVELOPMENT?

HTML5 has some new tricks up its sleeve, with Progressive Web Apps able to deliver app-like functionality with browser technologies. It's not only native technologies that are at risk, but also the app stores used to distribute them!



HTML5 is ever-changing, with the coming year promising to bring new technologies, tools and approaches. If you want to know more about what this year holds for HTML5, please get in touch for insights from our technology experts who have their fingers on the HTML5 pulse!

Scott Logic is a leading software consultancy with enterprise clients in finance, energy, healthcare and the public sector. Our services include software development & delivery, user experience design and an independent consultancy advising on strategic software selection and deployment. Our consultants, recruited on their exceptional qualifications, skill sets and knowledge, are central to continued project success and complete client satisfaction.

If you would like to discuss your HTML5 strategy, please get in touch:

enquiries@scottlogic.com

SCOTT LOGIC / ALTOGETHER SMARTER

3rd Floor, 1 St James' Gate
Newcastle upon Tyne
NE1 4AD

+44 333 101 0020

www.scottlogic.com