Application Development Strategies for a Multi-device World

You can’t control the proliferation of devices and operating systems, but you can unify and streamline your development with HTML5.

The multi-device era is a Big Bang-style explosion, with devices and operating systems fragmenting and pulling away from each other—all on their own trajectories. Smartphones, tablets, laptops, PDAs, desktops, netbooks, e-readers… Android, iOS, Windows, Windows Phone, BB10, Mobile Linux, Tizen, Firefox OS… Every new permutation adds complexity to the creation and management of business applications.

At the same time, usage patterns and user expectations are changing rapidly. Employees are using multiple devices at work and mixing personal devices with company-issued devices—and users expect business applications to work seamlessly from one device to another, with the same ease of use consumer-oriented web apps deliver. The result—if you’re doing native development (writing code for only one platform)—is that everything is in an upward spiral: projects, platforms, toolsets, interfaces, code bases, debug cycles, regression tests; the list goes on.

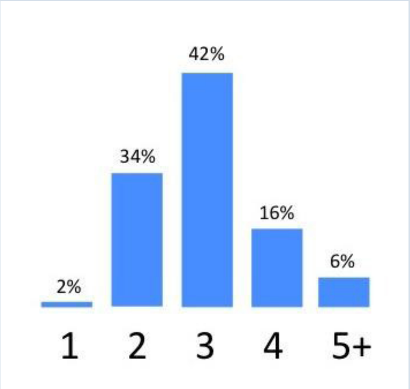
All of which leads to one simple question:
How much longer will native application development be sustainable?

There is a better option. This paper examines the capabilities of HTML5 for building universal apps that cut cost, complexity, and time-to-delivery in today’s multi-device world. It also considers the advantages of HTML5 development, common misconceptions about HTML5, and the emergence of powerful frameworks and tools that make HTML5 development simpler and more efficient.

Device proliferation: What can you still control?

The explosion in the sheer number of devices and operating systems isn't news to anyone in the development community, but evidence suggests that the speed and magnitude of this phenomenon—and the implications for application development teams—are still underestimated at many companies, even within the IT department.

For example, consider the BYOD (bring your own device) trend. According to a recent survey by IDC, 40 percent of devices used to access business applications are owned by employees, not employers, and that number is growing rapidly. Yet IT typically underestimates by 50 percent the proportion of employees using their own devices for company business,[1] and only 62% of IT managers even had a BYOD policy in place in 2012, according to Tech Target[2].



How many network devices do you use on a daily basis?

In a worldwide survey of corporate employees taken by Citrix Systems, most respondents said they were already using three different devices in their daily job routines; almost 25 percent were using four or more.[3] Yet a large percentage of internal business applications are still written for a small number of native clients—typically PCs and laptops—limiting accessibility and employee productivity.

Two things are clear:

- **The multi-device era is here to stay.**
  It's already big; it's growing fast; and device diversification will continue. Networking giant Cisco predicts that by 2020 there will be more than 50 billion network devices, or about seven per person, and that the categories of devices will expand well beyond today's array of smartphones, tablets, netbooks, etc.[4]
- **The IT department has lost control of hardware.**
  There is no longer any specific target hardware platform IT can count on, plan for, and write applications to. Five years ago BlackBerry was the dominant mobile platform and Windows was the dominant desktop platform. Today, there are multiple mobile and desktop platforms fighting for market share, and all of them play a role in the workplace. No one knows what types of devices will be used to access business applica-tions next year, let alone several years down the road. And the devices that do exist today will likely have much shorter lifespans than previous generations of devices, further complicating development strategies.

[1] Source: IDC, 2011, as reported in "BYOD Unleashed in the Age of IT Consumerization," by Bradford Networks.
[2] Source: "Businesses struggling to keep up with BYOD trend," by Simon Quicke, June 22, 2012.
[3] Source: Worldwide Survey of Corporate Employees, Citrix Systems, February 2011.
[4] Source: Cisco Visualization: "The Internet of Things": http://share.cisco.com/internet-of-things.html

What's left to control is the application itself: how it is designed, developed, deployed, provisioned, secured, and managed in the multi-device era. And that means IT will need a smarter application strategy—one that is application-centric rather than hardware-centric; one that focuses on optimizing the user experience across platforms; one that cuts the cost, complexity, and time-to-delivery for business applications that are accessible anytime, anywhere, on any device.

Specifically, the new breed of business applications must deliver:

- **Universality:** The application must provide a consistent user experience regardless of device, and all features and functions must work the same way across device types, in any context.
- **Great design:** Functionality alone is no longer enough; a business app needs to be visually engaging and work as smoothly and intuitively as consumer apps.
- **Compliance:** The application must adhere to internal policies for security, privacy, and confidentiality across devices.

For the IT department, this new reality is both a serious challenge and an enormous opportunity. The challenge is that change is required. IT policies and practices that have been honed over decades—and that have served the business well—need to be reimagined as application-centric rather than hardware-centric practices. That represents a significant change.

The opportunity, however, is compelling: By making the move to a more application-centric strategy and more flexible development practices, IT can reassert control over the quality and accessibility of business applications, while increasing its relevance and value to the business.

An application-centric development strategy is neither scalable nor sustainable with a native development approach. You would need multiple development teams with distinct, platform-specific skills, and the same application would have a different native code base for each platform.

By choosing HTML5, you can avoid these pitfalls and build universal apps that work across all devices and platforms—and you can scale them. The next section provides a quick recap of HTML5 basics, describes the growing momentum behind HTML5-based application development strategies, and summarizes the advantages of HTML5 development practices compared with native development.

HTML5 vs. Native: A Broader Perspective on the Debate

The dispute about whether HTML5 or native development is the best choice for the multi-device era rages on, with literally hundreds of articles, white papers, and blogs taking a side—often vehemently. The charter of this paper is not to settle the issue but rather to point out a few big-picture facts:

1 HTML5 vs. native is not an either/or choice.

There is a spectrum of options. For many companies, there are good reasons to use HTML5 for one project and native development for another. For others, a “hybrid” approach may make sense (hybrid can apply to an HTML5 app that is packaged in a native wrapper, or an app that uses native technology for some features and web technologies for others). Your company’s choice will depend on the number of applications involved, the number of platforms that must be supported, time and budget constraints, the development skills of your team, and many other factors.

The point is, there is no need to dedicate your application strategy to one approach or the other exclusively—and there is good reason to use HTML5 for many projects. Given the rising cost and complexity of native development, it makes sense to gain experience developing HTML5 applications. The cost is low and the potential payoff is high.

2 Momentum is building behind HTML5.

HTML5 adoption rates among developers and users are increasing rapidly and HTML5 feature support is expanding among operating systems and web browsers. HTML5 now has broad support from Apple, Mozilla, Microsoft, and Google, and new smartphone operating systems such as BlackBerry 10, Ubuntu, Sailfish, Tizen, and Windows Phone 8 are also betting on HTML5.

According to Forrester Research, nearly 75 percent of users in North America and 83 percent in Europe are running browsers that support a large segment of the HTML5 feature set. Forrester says that the penetration of HTML5-compatible browsers grew from 57 percent to 75 percent between the second quarter of 2011 and the second quarter of 2012.[5]

And nearly three-quarters (72 percent) of developers in a survey from researcher IDC said they were planning on integrating HTML5 in one form or another into their mobile apps. Only 21 percent planned to stick entirely to native code, according to the survey. This is key because smartphones and tablets represent more than 90 percent of the new net growth in device adoption for the coming four years, according to Gartner Research.[6]

[5] Source: “HTML5 – Maturing Desktop Browser Support Opens The Door To Enhanced Commerce Experiences,” by Peter Sheldon. Forrester Research, August 7, 2012.

[6] Source: Gartner Top Predictions for 2012 and Beyond, Jan. 2, 2012.

> “We are at an inflection point: HTML5 is no longer an emerging toolset for mobile and tablet development… it is fast becoming the de facto standard for web experience innovation across touchpoints.”
>
> – **Peter Sheldon**, Forrester Research

2 There are still many misconceptions about HTML5.

Many developers and development managers remain confused about what exactly HTML5 is, what it can do, where it fits within an application development strategy, and how it contrasts with native application development techniques.

Simply put, HTML5 is a family of technologies designed to transform the web browser into a rich application platform. HTML5 applications target a universal, cross-platform client: the web browser. And HTML5 applications are web-deployed, so they harness all the benefits of the web, including easy searchability and indexing (including deep linking), the ability to easily include third-party services and mashups, and hassle-free deployments and updates.

The core HTML5 specification works in concert with related technologies such as CSS3 (Cascading Style Sheets version 3) and JavaScript to enrich the capabilities of applications delivered via the web. Typically, HTML defines the structure of the web content, JavaScript manages behavior, and CSS determines the visual presentation of the content. For a more detailed overview of HTML5 technologies, visit http://pages.sencha.com/resources and read the white paper entitled “The Development Manager’s Quick Guide to HTML5.”

Why use HTML5 in business application development? Because it allows development teams to design and build universal applications that run on any web browser client—including mobile and desktop browsers—and because it enables teams to create richer, more responsive online application experiences that users can also take offline with them.

HTML5 also delivers instant gratification for development teams. HTML5, CSS, and JavaScript are all interpreted languages that require no compilation cycle. Developers can debug in-browser, make a quick edit, and refresh the browser to see the change live. That means applications can benefit from rapid-fire innovation: developers build, test, deploy, iterate, and repeat. HTML5 development is also easier to learn, and there are millions of excellent HTML5 developers in the workforce, so it is easier and less expensive to find talented professionals.

By contrast, native application development often requires separate tools and languages for each platform and limits user access to only those platforms. While native development may provide higher fidelity and tighter integration with the host OS, developers have to maintain separate code bases for each targeted device and master multiple languages and skills. If you develop an Android app in Java, for example, you'll be faced with writing a whole new app in Objective C to support iOS. An HTML5 app will run on both platforms—and all the others. And with a small investment of effort in "theming" (creating a common logic base for a specific platform, such as Blackberry 10), HTML5 apps will look and feel exactly like native apps to the end user.

In short, native development has inherent cost and timeframe disadvantages in the multi-device era. Finding, recruiting, hiring, and paying a high-quality Objective C or Java developer is expensive, and if you outsource your project it may cost even more—not to mention the fact that you may become locked in to specific vendors and their development cycles. Each platform you need to support adds more expense and requires additional time.

The bottom line: For most companies, native development practices are simply not sustainable in today's multi-device world.

The Value of HTML5 Frameworks

HTML5 is both a specification and a family of related technologies all dedicated to transforming the web browser into an application platform. But it is not a development environment. To help development teams fully realize the capabilities of HTML5, Sencha and other vendors have created powerful developer tools and rich JavaScript frameworks that provide cohesive, proven, documented components for HTML5 application development.

A framework is a reusable development platform that acts as a layer of abstraction on top of the basic family of web technologies: HTML5, JavaScript and CSS3. It vastly improves developer productivity and the potential sophistication of applications by providing libraries of components, architectural patterns, extensibility capabilities, UI themes, and other valuable development tools.

Choosing to use a framework rather than raw-metal HTML5 and JavaScript will save development teams an enormous amount of time and effort; but choosing the right framework is critical. There is a multitude of micro frameworks and feature libraries available, and at any given time one or another is particularly popular and fashionable. Many of these can be useful for very specific use cases, or can be cobbled together for larger more complex apps—though in both cases you typically need to maintain it all yourself and look to the open source community for technical support.

Sencha offers a more complete option: macro frameworks. These are designed to allow developers to build a wide variety of apps and serve the full range of application use cases. Sencha frameworks are used by more than 2 million developers worldwide, so there is a large talent pool available and an active community building components and extending the frameworks. And because these frameworks are backed by Sencha, enterprise-grade support is available, along with professional training and consulting services.

Sencha HTML5 Frameworks

Sencha frameworks enable developers to build cross-platform, cross-browser mobile and web apps. Our product portfolio consists of two classes of products: application frameworks and tools. The frameworks are the core systems that enable the rapid creation of rich multi-platform mobile applications and true, cross-browser web applications, from IE6 to Chrome 10 and everything in between. Our tools accelerate the developer's workflow even further by leveraging the frameworks to enable visual development of user interfaces and interactions. Our main framework offerings are Ext JS and Sencha Touch:

Ext JS is designed for the creation of desktop applications, enabling developers to build powerful functionality and rich user experiences with ease. Ext JS also includes a large library of UI components and default UI themes that enable developers to focus on building the applications they want.

Sencha Touch is a framework used to build mobile and touch applications. It's the world's first application framework built specifically to leverage HTML5, CSS3 and JavaScript in any touch-UI platform, and to provide the developer with the greatest power, flexibility and optimization.

Sencha products make specific use of HTML5 to deliver components such as audio and video, as well as a local Storage proxy for saving data offline. We have also made extensive use of CSS3 in our style-sheets to provide the most robust styling layer possible.

To complement the frameworks Sencha offers, we also deliver an array of tools and plugins, including:

Sencha Architect is an HTML5 visual app builder that empowers teams to design, develop, and deploy apps from a single, cohesive environment. Designers and developers can work in lock-step to create amazing applications.

Sencha Mobile Packager allows developers to take a web application and package it for deployment to the native stores on both Mac and Windows. Not only does it allow this hybridization of applications, it also provides deep hooks into the native APIs, enabling developers to build a more platform-integrated experience.

Sencha Animator is a desktop application used to create animations for mobile browsers and touchscreen mobile devices. Motion designers can use Animator to create rich experiences for today's most popular devices using just CSS3 and no plug-ins.

Sencha Desktop Packager is an easy-to-use, command-line tool that enables developers to package web applications — written with web technologies such as HTML5, CSS3 and JavaScript — as native desktop applications. Now developers can deploy web apps as high-performance native apps for Windows, Mac OS X and Linux systems without dependency on the browser.

Sencha Eclipse Plugin lets developers tap into the power of the Eclipse IDE to code faster and more efficiently when working with Ext JS and Sencha Touch. It provides efficient code editing features such as rich code completion, error correction, and Eclipse-based team collaboration — making it easier to build complex applications in large, distributed team environments.

Conclusion

For most companies, native development is no longer affordable or sustainable as an exclusive application development strategy in today's multi-device world. But with the maturation of HTML5 and the availability of powerful, proven HTML5 frameworks, there is a better option.

HTML5 frameworks provide a way to build universal, cross-platform business applications that deliver the levels of performance, reliability, accessibility, and security users need, with the efficiency development teams and business managers require. That is why momentum has been building behind HTML5 adoption, and why it makes sense for businesses of all types and sizes to begin experimenting with the capabilities of HTML5 frameworks.

Learn More

For additional information about how to take maximum advantage of HTML5 in your application development strategy, visit us online at www.sencha.com. You can also:

- Mingle with other web developers and Sencha experts on our developer forums (www.sencha.com/forum)

- View screencasts and code demos at our Learning Center (www.sencha.com/learn)

- Sign up for training sessions on our frameworks and tools to jump-start new developers as they use HTML5 (www.sencha.com/training)

5 Common Misconceptions about 5

MYTH HTML5 has inherent performance problems.

For the vast majority of business applications on modern platforms, HTML5 performance can equal or exceed that of native technology. The reality is that the performance of any specific app is more a function of development technique and coding skills than any inherent issue with HTML5. With the exception of a few types of apps that are very device-centric — such as graphics-intensive games — the application will perform well with HTML5 on today's platforms if it is well-written.

Recent comments by Facebook CEO Mark Zuckerberg added to the misconceptions about HTML5 performance. Zuckerberg attributed the slow performance of Facebook's mobile application to HTML5 and said HTML5 "just wasn't ready." But in that case and many others, the issues attributed to HTML5 — poor performance or choppy user experience — actually stemmed from taking a "website" development approach to building the app, using the wrong tools and page-centric architectures that are inappropriate for mobile applications. A few Sencha developers proved this point by building an HTML5 version of the Facebook mobile app that performed as fast or, in some cases, faster than the Facebook native mobile app. They shared the app, Sencha Fastbook, for anyone to try, and told the story of how they built it in a widely read blog post that you can find here: http://www.sencha.com/blog/the-making-of-fastbook-an-html5-love-story/.

MYTH HTML5 apps are not secure.

An application is a collection of many things: a client, middleware, and a data source. Enterprise applications built with best practices employ "defense-in-depth," meaning that each part of the application is designed to be secure and to never trust other parts of the application. HTML5 applications are no exception. HTML5 includes many security features such SSL/TLS to secure the data while in transit, cookie policies to prevent browsers from inadvertently storing data, XmlHttpRequest2 to allow secure and safe cross-domain access to data, content origin policies to prevent hijacking of data, and more. HTML5 applications have more restricted access to system resources than Flash. And HTML5 protocol updates are delivered through browser updates, so they're more likely to be applied. Moreover, the built-in security features of HTML5 can be supplemented as needed for specific requirements.

The result is that well-designed HTML5 applications can handle the most secure and sensitive transactions and data: healthcare, finance, insurance, etc. And HTML5 has proven to be secure in enterprise deployments. HTML5 applications and sites are used by the Internet's biggest brands, including Google, PayPal and Bing to deliver their applications to customers, and all major browser vendors work in conjunction with the HTML5 community to address and resolve identified security issues.

MYTH

HTML5 apps are useless when users are offline.

HTML5 developers have access to multiple resources that enable an HTML5 application to run when no Internet connection is available. For example, the browser can use AppCache to save resources locally, so they're accessible when a network connection is not available. Using the NETWORK and FALLBACK sections makes it possible to tailor the app's experience based on the network connection state. Local Storage and IndexedDB enable data to be stored for both online and offline use, and window.navigator.onLine and associated event listeners can even help you switch state at runtime and make the right decisions for your application.

MYTH

Source code transparency is a big problem with HTML5.

HTML5 applications are built on JavaScript, and by its nature as a scripting language, JavaScript is difficult to hide. After all, the browser needs to execute the scripts. Does that mean HTML5 apps are inherently insecure? Not at all. Modern apps are multi-tiered, and the security of the app does not need to reside entirely in any one tier. For most business applications, the security of the back-end business logic is far more vital to overall security than the client app. For example, is Google concerned about the source code transparency of Gmail? It's an HTML5 web app, but the business logic and data are protected by multiple layers of security. Developers need to address security holistically, and the transparency of the source code is simply one factor to consider.

MYTH

With HTML5 you can't put an app in the app store.

HTML5 apps actually have the widest set of distribution options of any app dev technology, and distributing via the app store is one of them. In fact, many apps in the Android and iOS stores are actually HTML5 applications that were packaged for store distribution using technologies such as Sencha Mobile Packager or Apache Cordova (PhoneGap). But look at the bigger picture. For most developers, the closed app market is not a great way to monetize or broaden the use of applications. The process of getting the application into the store can be time-consuming, frustrating, and expensive. Developers need to manually submit each app to each closed market and abide by their review and submission processes. Updating apps is a separate and equally complex process. HTML5 apps, on the other hand, are on the web and have a URL. They can be updated without requiring any action on the part of the user. This makes them ideal for business apps especially. And best of all, they're flexible enough, so they can be packaged as a native application for iOS or Android if there is a specific need.