

David Chappell

July 2013

CHOOSING CLIENT PLATFORMS IN THE PC+ ERA

A PERSPECTIVE FOR ISVS

Sponsored by Microsoft Corporation

Copyright © 2013 Chappell & Associates



DavidChappell
& Associates

Whatever market you're in, it's crystal clear that desktop and laptop PCs are no longer the only clients that matter. Tablets and mobile phones are now critically important. But it's just as clear that PCs aren't going away. While tablets and phones are better for many scenarios, such as a sales professional visiting clients or a child watching a movie, they're not the right choice for many others. Anybody who creates or manipulates complex content for

This is not the post-PC era; it's the PC+ era.

significant periods of time is still likely to choose a PC. If you're writing a report, for example, or creating a presentation or working with spreadsheets or editing video or developing software, you probably want lots of screen real estate, the efficiency of a real keyboard, and the precision of a mouse.

The truth is evident. This is not the *post-PC era*; it's the *PC+ era*. Desktop and laptop PCs will matter for a long time to come, as will tablets and phones. The question for independent software vendors is clear: Which clients should you support in this new world?

In general, the answer is also clear. If you're shooting for the largest number of mobile devices today, you need to start by creating an app for iOS and/or Android. If you're looking a move or two ahead, aiming at where the installed base of Windows clients is headed, you'll also create apps for the Microsoft device ecosystem, especially if you're an enterprise-oriented company. To see why, it's helpful to take a broader look at application platforms today.

Application Platforms: The Big Picture

Every successful ISV leader knows how important it is to choose the right foundation for their firm's applications. Making this choice has sometimes been easy. The default platform in many markets, for example, was once Windows on both server and client.

But those days are gone. Today, most new enterprise ISV applications are built using the Software as a Service (SaaS) approach, often with mobile clients, with server logic running on a cloud platform. Many new consumer applications are either purely mobile apps, such as games, or mostly mobile apps with some functionality running in the cloud. In both cases, the platform technologies you should choose for a new application aren't necessarily obvious. Figure 1 summarizes the situation.



Figure 1: Modern ISV applications commonly spread their logic across cloud and clients.

One way to think about this is to look at how an application's functionality is divided between cloud and client. Figure 2 illustrates the options.

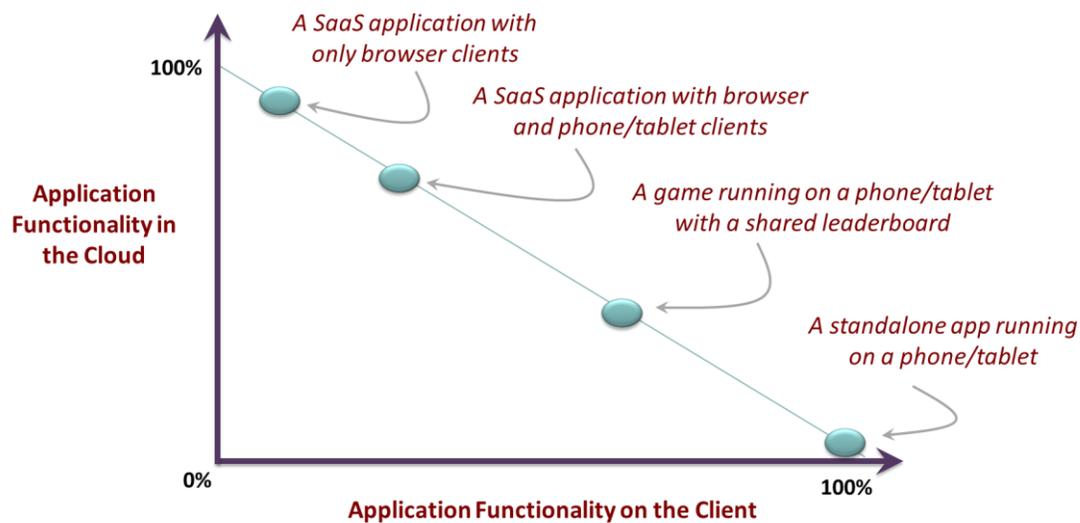


Figure 2: Every cloud/client application falls somewhere on this line.

At one extreme is a pure SaaS application with only a browser client. In this case, the great majority of the application’s functionality runs in the cloud, with perhaps a few things running as JavaScript inside the browser. At the other end of the scale is a standalone mobile app, such as a simple game. Going forward, however, relatively few ISV applications will be at either of these extremes—they’ll be somewhere in between. SaaS applications today commonly need to support tablets and phones, for example, while even mobile games rely on the cloud to store and share information, such as leaderboards.

Where an application falls on this continuum affects other things as well, including these:

- *Typical monetization strategies:* Cloud-centric apps, more often targeting business users, commonly use subscription pricing. Mobile-centric apps, those in the lower right of Figure 2, are more likely to be used by consumers, and they’re also more likely to make money through some combination of a one-time application sale, in-app purchases, and advertising.
- *Cloud platform technologies:* Mobile apps with relatively simple cloud functionality can use a technology designed explicitly for supporting mobile apps, such as Windows Azure Mobile Services. A cloud-centric SaaS application can’t get by with this, however. Instead, it needs to use a more general-purpose cloud technology such as Windows Azure Cloud Services.
- *Choice of client technology:* Applications in the lower right of Figure 2, those that are client-centric, are probably written as native apps for whatever client they run on. Since the bulk of their value comes from code on the client, that code needs to get the most out of the device. Cloud-centric applications, though, can likely get by with an HTML5 app that’s essentially the same across all clients. Since most of the application’s functionality is in the cloud, the client’s main purpose is often just to let users access data, something that might not require client-specific services. Building an HTML5 app is simpler and cheaper than building multiple client-specific apps—there’s less custom code to create and maintain—so it makes sense to do this whenever possible.

Choosing Clients: What Should Your Application Support?

To repeat: This isn't the post-PC era, but is certainly is the PC+ era. The client universe has expanded well beyond web browsers and Windows PCs. As an ISV leader, how can you make the best decisions for your applications today?

To think about this, it's useful to look at the choices made by the leading vendors of client operating systems. Figure 3 summarizes the situation.

| | <i>Apple</i> | <i>Google</i> | <i>Microsoft</i> |
|----------------------|--------------|---------------|------------------|
| Desktops/ Laptops | OS X | Chrome OS | Windows 8 |
| Tablets | iOS | Android | Windows 8 |
| Phones | iOS | Android | Windows Phone 8 |

Figure 3: Different vendors have taken different approaches to client operating systems for the PC+ era.

The PC+ era really began with Apple and the iPhone. The iPhone's iOS operating system brought a fresh approach to mobile devices, one that Apple (literally) expanded on with the introduction of the larger iPad. The iPad offered the same OS on a bigger screen, and it showed the world that there was a market for tablets. But Apple has chosen to use a different system, OS X, for its desktops and laptops.

Google's approach has been similar to Apple's. Android first appeared on phones, then moved to tablets, while Chrome OS has been Google's offering for desktops and laptops. Both Google and Apple appear to see phones and tablets as having a natural affinity, with PCs the odd man out.

But this apparently natural affinity can also be seen as a historical accident. Tablets and phones run the same operating system in no small part because of the order in which they were created and the organizations that created them. From another perspective, the natural affinity is between PCs, specifically laptops, and tablets. It's common to see tablets used with keyboards; some even come with covers that double as keyboards. At the same time, laptops are rapidly shrinking—many aren't much bigger than tablets—and new ones commonly have touch screens. The difference between the two is dissolving. And while desktop PCs don't look much like tablets, new models also come with touch screens today, allowing them to be used in similar ways.

From this perspective, the right approach is to use the same operating system on desktops, laptops, and tablets. This is what Microsoft has done with Windows 8, as Figure 3 shows. Doing this requires that Windows 8 support both traditional desktop users and newer tablet-style scenarios. Accordingly, the system combines both approaches, offering a traditional desktop and a touch-oriented user interface.

For phones, Microsoft currently uses Windows Phone 8, which is distinct from Windows 8. The company's stated strategy, however, is to converge the application environment of these two systems. Since they share a number of

things today, including a similar user interface, writing an application that runs on both Windows 8 and Windows Phone 8 isn't especially difficult. Over time, Microsoft tells us to expect these two worlds to become increasingly similar. This approach of providing a common platform across a range of environments is in Microsoft's strategy DNA. We've seen this movie before.

Providing a common platform across a range of environments is in Microsoft's strategy DNA.

Which approach is correct? Is it better to use the same OS on phones and tablets and something else on PCs? Or is running a common system on PCs and tablets more useful, with phones eventually supporting the same applications?

The most likely outcome is that both strategies will survive. Clients running iOS and Android are clearly valuable, as is shown by their many millions of customers.

But Windows desktops and laptops are also used today by many millions, and so an approach that marries this world with tablets makes sense. Especially in businesses, Windows 8 and its successors will be a dominant client system for years to come. The investment is too large to move away from easily, and users commonly hate every pixel of UI change.

These realities should shape your client platform choices. If you're shooting for the largest number of mobile devices today, you need to support iOS and/or Android. If you're targeting the future Windows world, you'll also create apps for the Microsoft device ecosystem, especially if you're an enterprise-oriented company or if you value significant investment from the platform vendor in your success.

Conclusion

It's an unsettled time for ISVs, with new platform choices for both server and client software. How can you best take advantage of this for your new applications?

One obvious answer is to rely on cloud platforms whenever possible. Why run your own hardware when somebody else will do it for you, probably for less money? For an ISV building a new application, whether it's mostly cloud, mostly mobile, or right in the middle, adopting Windows Azure or another cloud platform is usually an easy decision.

Choosing which clients to support is harder. If your application fits in the upper left of Figure 2—its functionality is almost all in the cloud—offering just a browser interface might be good enough. If this isn't true, however, you'll need to decide which client operating systems to bet on. For desktops and laptops, Windows 8 and its successors are likely to dominate for some time. While a true post-PC era will arrive at some point—no technology lasts forever—it's years away. This dominance is likely to let Microsoft extend Windows' popularity to tablets as they merge with laptops. And especially once Windows Phone can run the same applications, the ecosystem will get bigger still.

Still, any ISV looking exclusively at mobile clients today has to support iOS and/or Android. For enterprise-oriented ISVs, building client applications for the mobile Windows ecosystem is also important, largely because of the dominance of Windows today and tomorrow in businesses. One thing that's certain is that the PC+ era—our world today—won't be the same as what came before.

About the Author

David Chappell is Principal of Chappell & Associates (www.davidchappell.com) in San Francisco, California. Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better decisions about new technologies.