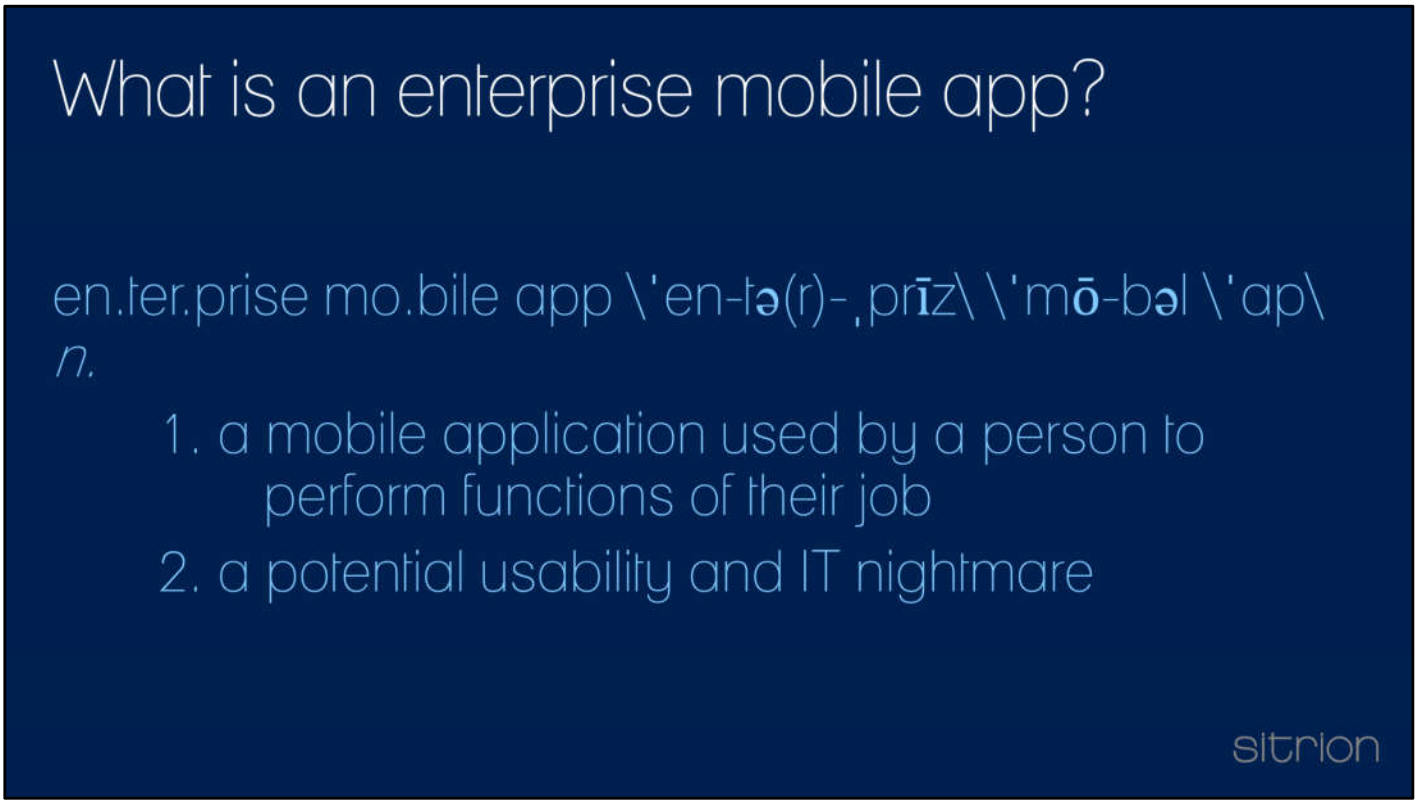
My name is Jenny Blumberg and this is a session devoted to enterprise mobile apps.

I am a product manager for a small software company here in Denver called Sitrion and we build mobile apps for businesses. We were formerly focused on enterprise social software and now we're devoting most of our time to productivity apps.



To kick things off, what is an enterprise mobile app?

Quite simply, a mobile application used by a person to perform functions of their job. Simple enough. There are many apps I can think of that I use on a regular basis to work like Evernote and Dropbox. But a true enterprise mobile app is one that would come with a blessing from IT. So the focus of our discussion today is how to arrive at "enterprise ready."

Now we know the "what" for enterprise mobility, how about why enterprise mobile?

A couple of metrics really stand out to me that show just how sought after enterprise mobile apps really are.

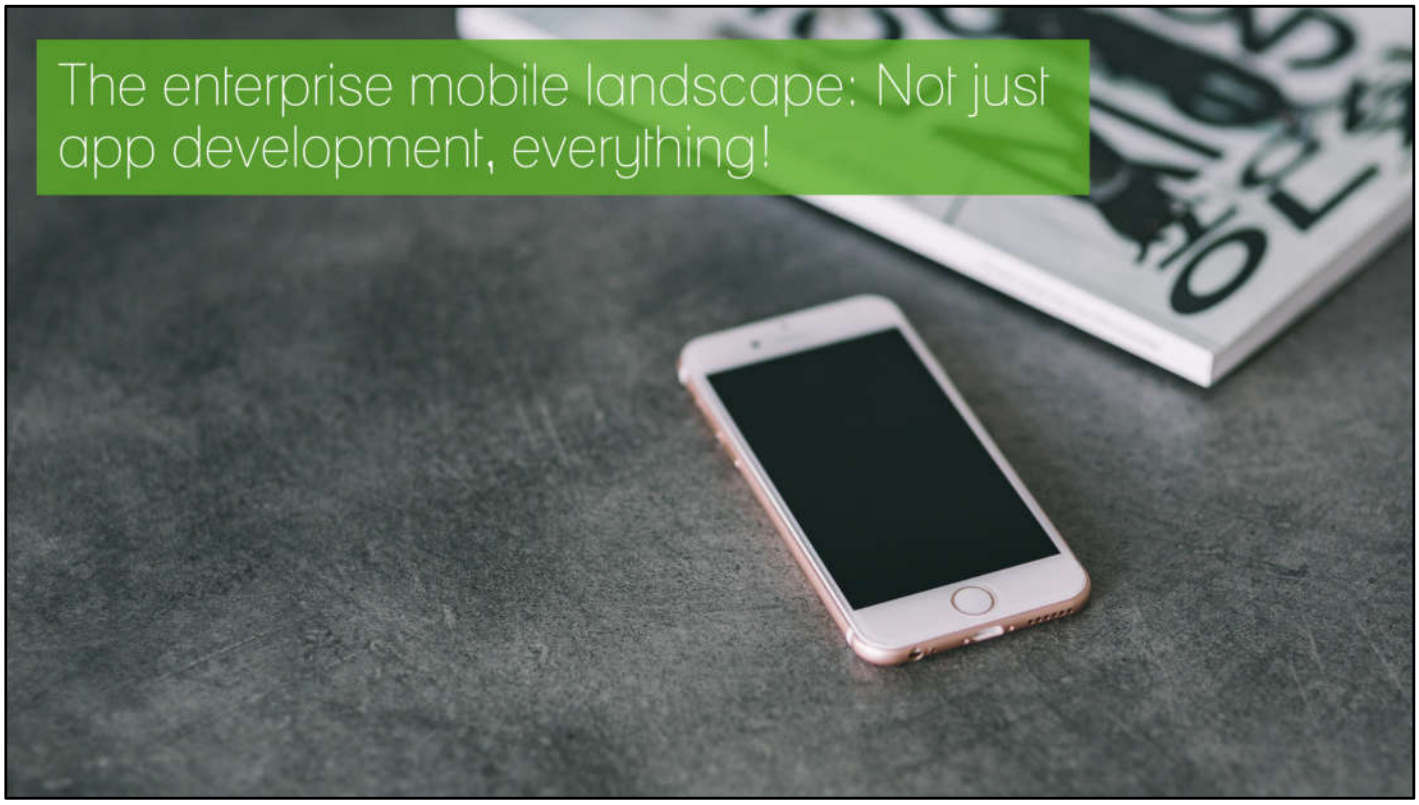
Enterprise mobile as an industry is growing at a rapid pace.

91% of companies plan to invest in mobile, and 86% plan to spend more than $100,000 on the initiative. Across all different industries, companies are interested in starting their mobile journey and oftentimes they don't yet have a mature mobile solution in place.

You might have heard, but iOS is dominating the enterprise in comparison to Android and Windows Phone. As of last year Apple represented 66% of device activations for business use. Things like the iPad Pro and Apple's key partnerships with the likes of IBM and Cisco keep them at the forefront of mobile for business.

Finally, demand for enterprise mobile app development is way beyond what is available right now as far as resources. According to Gartner this is to the extent that the demand will grow five times faster than what internal IT organizations are able to deliver.

So what we have is this gap where there is a lot of demand for enterprise mobile apps but not enough developers to meet the goals of business right now.



This has given rise to rapid mobile app development, or RMADs, that give non-developers and developers alike tools to create mobile app experiences. You might recognize the names Kony or Xamarin or maybe even have had the chance to try out an RMAD yourself. The focus of this session is more geared towards native iOS development. RMADs certainly have their place in the realm of enterprise mobility for some situations, it all depends on the needs of the business.

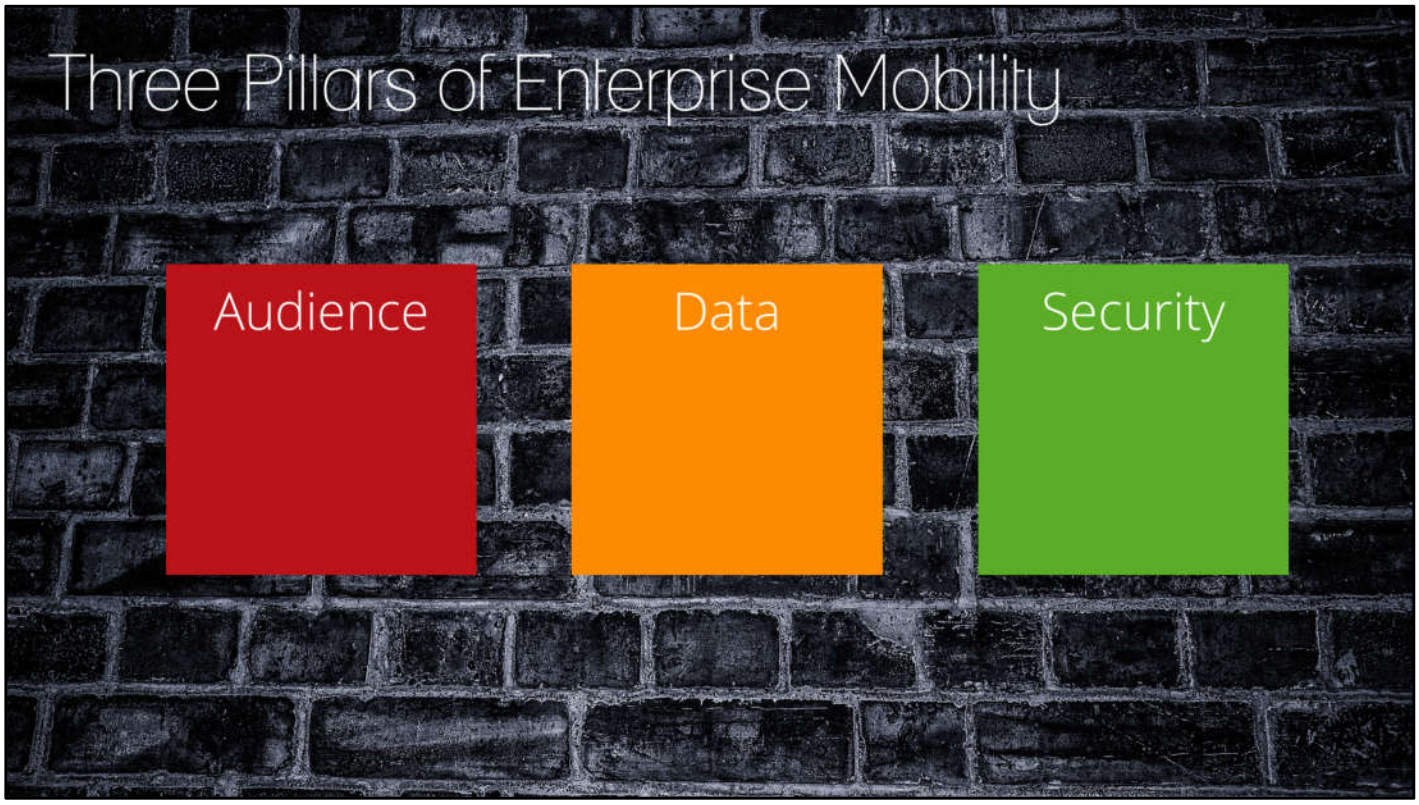
Understanding what options you have out there, not just for development, but also distribution, providing connectivity, device and app management, etc., you begin to see that the enterprise mobile landscape isn't just comprised of app development- there's a lot more to it!

With a lot of these aforementioned features of the enterprise mobile landscape, you'll find they are pretty standard within EMM vendors, EMM standing for Enterprise Mobile Management. You may be familiar with AirWatch/Vmware, MobileIron, Citrix, Blackberry, etc. If you're working with or for a big company, they likely have one of these systems in place to assist with features and security that entail a good mobile solution, most notably device and app management and tunneling for access to internal resources.

However! For small to mid-size companies, purchasing an EMM may be out of the question. There is also sort of this pendulum swing we're in where EMM solutions can be seen as over-reaching, they can overtly control a user's device, and companies are responding by looking into built-in OS features that can do things like tunneling and app management without having to integrate an SDK from an EMM vendor. If you want to learn more about that, take a look at the AppConfig Community.

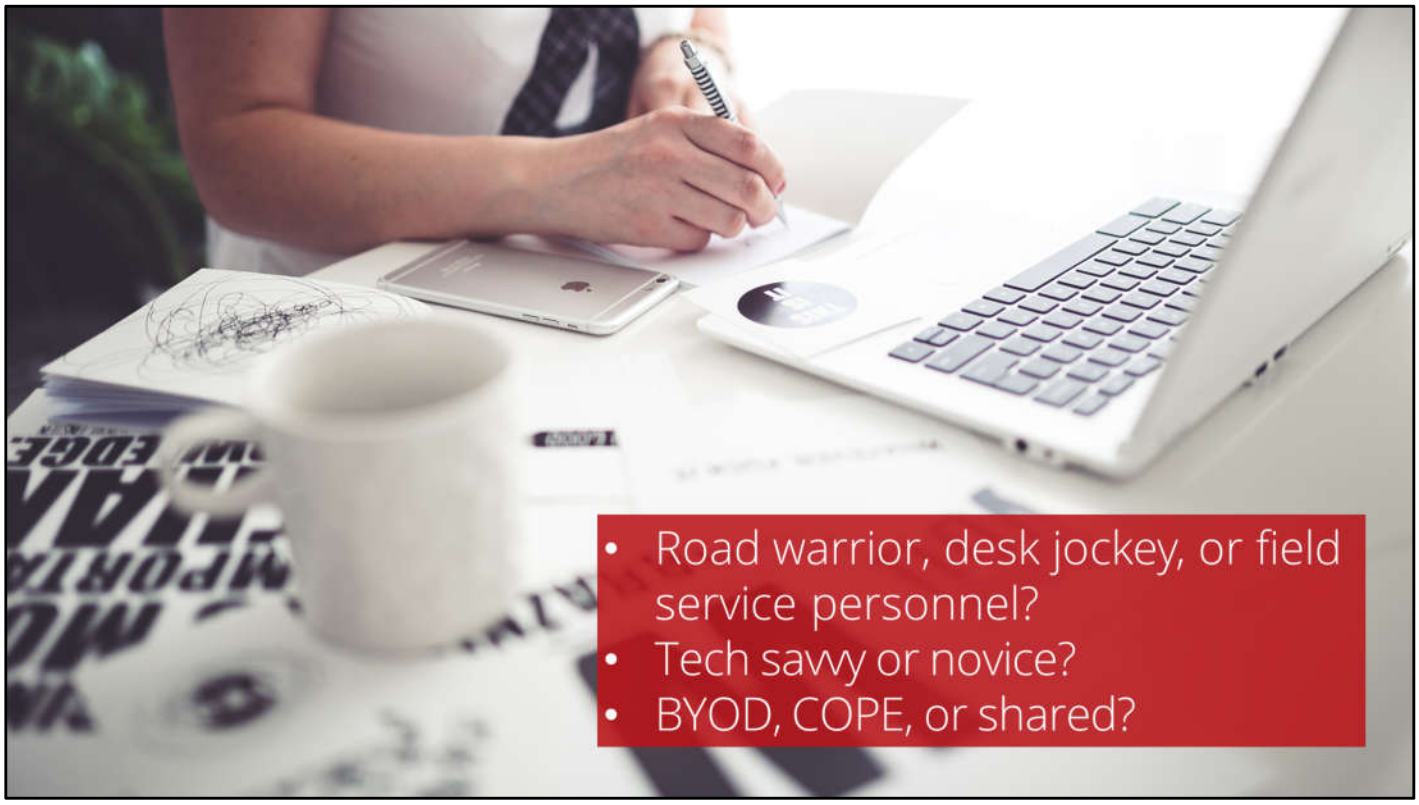
Regardless of whether we're doing an integration with an EMM, or if we're doing a home grown solution, there are still critical tenants that comprise a mobile strategy that app developers should be concerned with.

We are finally ready to get to the nuts and bolts of enterprise mobility, so let's now walk through now how we can make an app enterprise ready!



Let's divide up all those enterprise mobility topics into what we'll call the pillars of enterprise mobility: audience, data, and security.

Audience



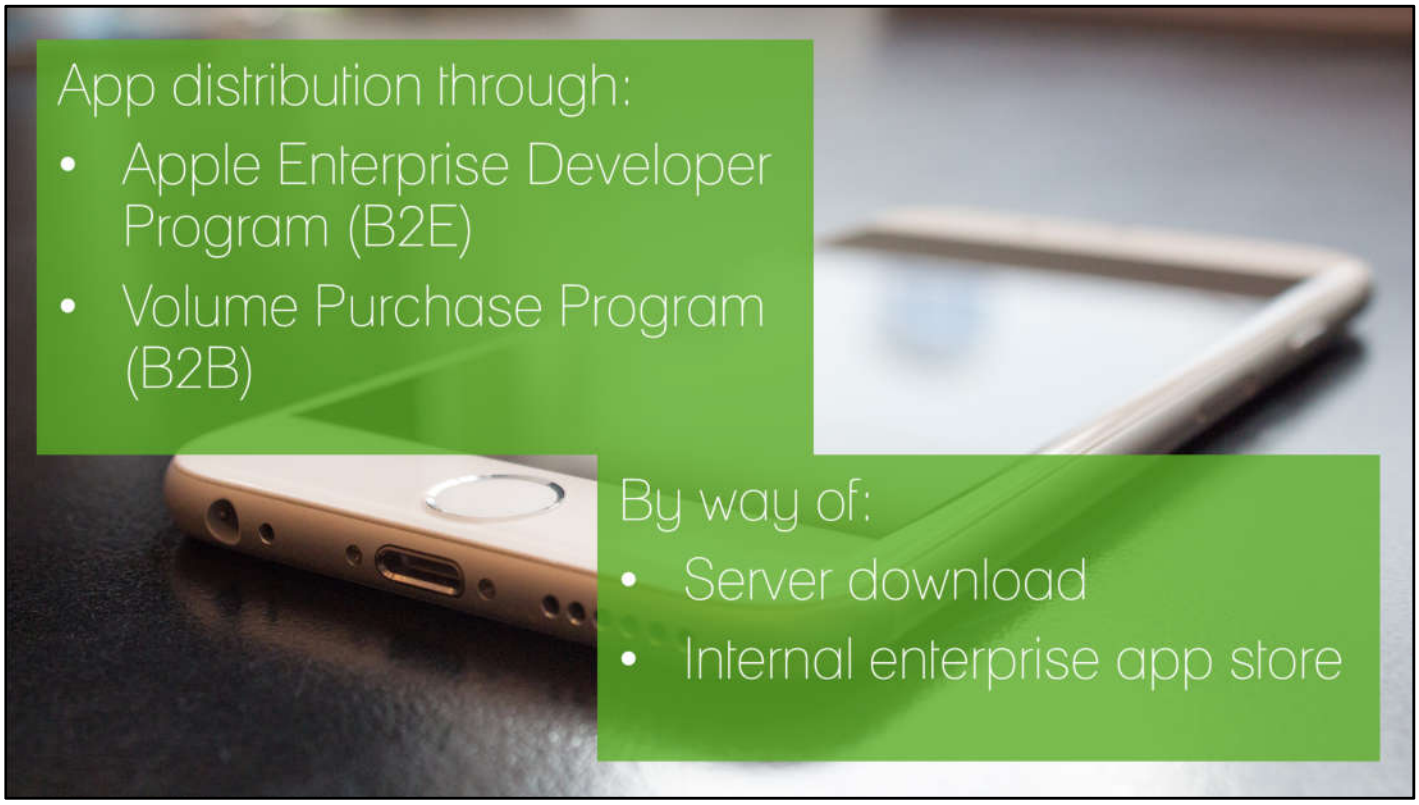
When you think of the individuals who'll use your enterprise app, it can influence decisions down the line that then can impact security and permissions and the like. Some qualities you might consider are when they might be using their mobile device. Are they a sales person on the go? Are they always sitting at a desk? Or are they in an industry where they're mostly moving around and may have unreliable internet access? In other words, are they always on mobile, or only every so often? Do they even have an email address? How do we identify them in order to first allow access and then decide what we can show them?

--Are they tech savvy or more of a novice? How simple does the app need to be? Do they understand native OS controls and behaviors or they're an Android user and someone just handed them an iPad one day to keep track of timesheets?

--Finally, do they own the own device or using corporate owned? BYOD meaning bring your own device and COPE stands for corporate owned, personally enabled. Is our app going on a device that is shared amongst workers?

This last factor especially can have a great impact on your app design and security considerations. While it can be quite nice to have your own phone that you use for everything, it can also be an IT nightmare.

With these considerations we can start to understand how to manage what comes a bit later, namely access to enterprise data and how we secure it. But as part of reaching our audience, let's talk about app delivery.



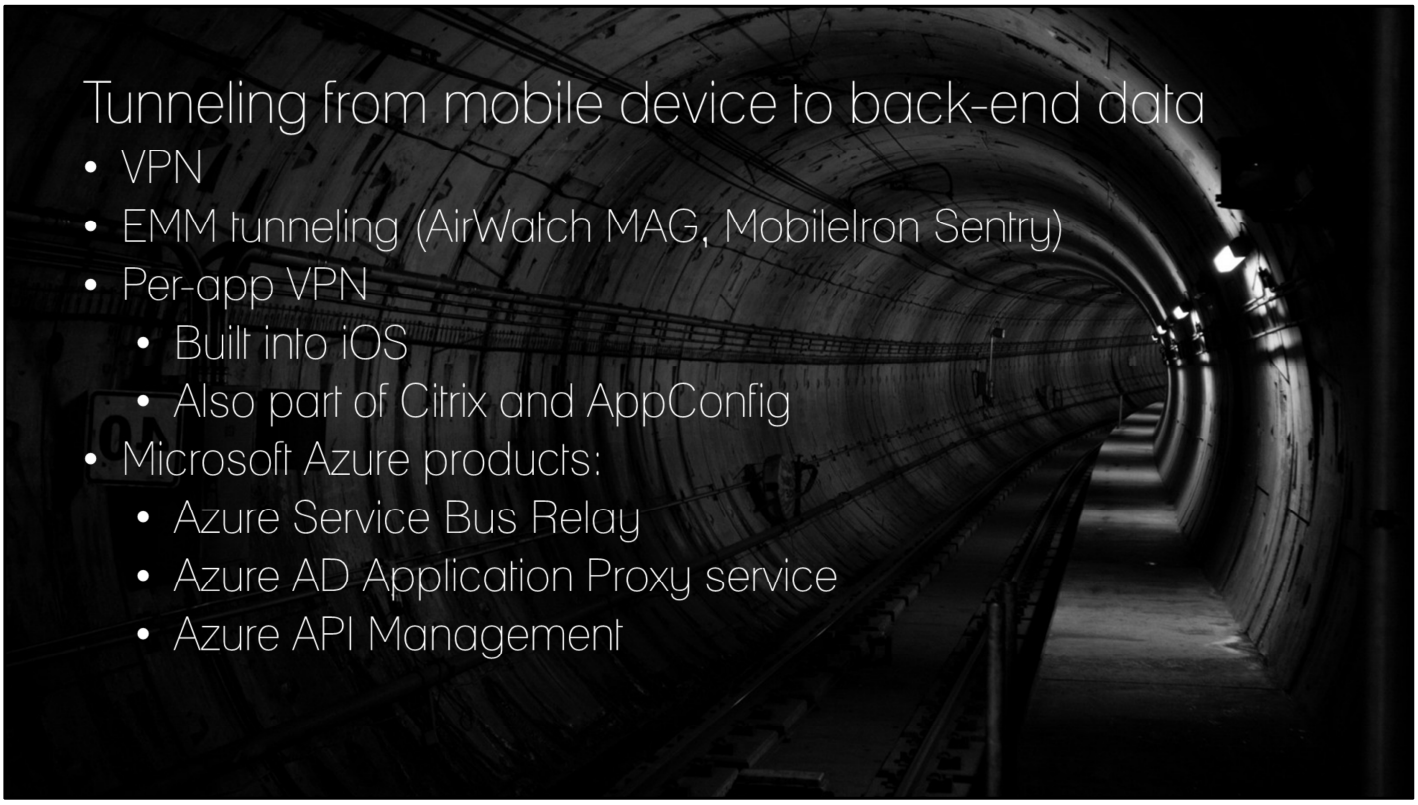
We've got an app that is an internal resource, whether it's B2B or B2E, and it does not belong in the public App Store. Luckily Apple has pretty well streamlined the distribution process for the enterprise. Depending on what audience we're targeting, if we're providing an app to fellow employees it's best to distribute through the Apple Developer Enterprise program. And if we're providing the app to an external entity we'll use the Apple Volume Purchase Program.

Now at the very least for distribution we can simply put the app up on a server somewhere and link people to it. Oftentimes though a company will have their own enterprise app store, especially in the case where they're using an EMM vendor. Most EMMs can connect to VPP to streamline the download and distribution process. In fact, Apple even recommends an EMM solution for enterprise app distribution.

For our purposes on the mobile development side, as long as we have a good understanding of our audience and how they're using mobile, we can move on to how that'll impact the more technical aspects of enterprise mobility .

Data

If you have an enterprise mobile app you're most definitely interacting with enterprise data. In the case of SAP, Oracle, SharePoint, etc., the qualities of this enterprise data fly in the face of the spirit of mobile: stored on-premise, behind a firewall, locked down, and formatted in a way that isn't designed for mobile consumption because it was implemented 20 years ago. The good news is that companies are embracing cloud technology which interface a lot better with mobile technologies. The bad news is that you're still likely to encounter situations where your enterprise mobile app is going to have to somehow get through to that data stored behind the firewall.



Unfortunately for IT, mobile devices have a tendency to move around outside of reliable internal corporate networks. What people were stuck with was one of two options: only allow access when connected to corporate wifi, which sort of defeats the purpose of being on mobile, or using a separate VPN client on the phone which was cumbersome to use and could significantly drain the battery. So what happened was EMM providers came up with proprietary tunneling technologies that would streamline mobile access through the firewall, provided the requests were coming from a managed device; or, a device where you enroll with the EMM even though it installs a creepy profile on your phone that lets you know your IT admin is watching your every move.

But what if you're not using EMM? Well, you have a couple of options.

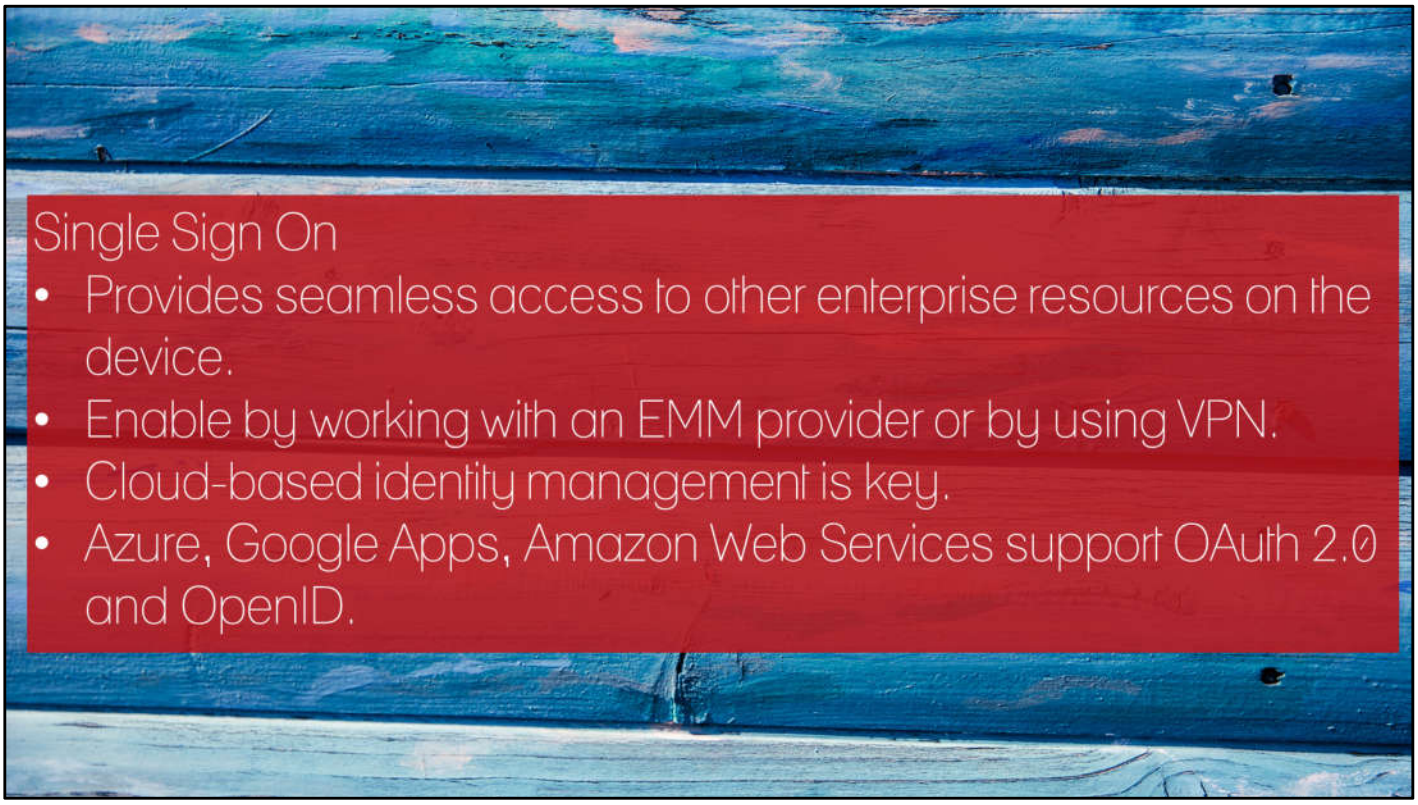
There's something called per-app VPN which is actually built into iOS. Obviously you would use this in conjunction with a VPN provider. What it provides is more granular access than device level VPN and so innately personal data and company data is separated. Citrix as an EMM vendor also provides it and the AppConfig community supports this approach as well. Microsoft is also getting into the game with a number of options as part of Azure. These are sensible in that they use your normal web protocols to gain access so your IT team doesn't have to open any new ports in the firewall.

Azure Service Bus Relay is a utility hosted in the cloud (obviously) and it acts as a gatekeeper handling connections between a mobile client and internal corporate resources.

Azure AD Application Proxy service isn't yet very widely available but it can be advantageous to use in that you can manage application, device, and user controls and it easily supports multi-factor authentication.

And finally Azure API Management which also acts as a proxying service where you're able to expose your internal web services securely to the outside world to enable access. There's a little more work involved is supporting this one as it requires changes to the on-premise web server.

In my personal experience more often than not, a company has an EMM vendor in place to handle tunneling and they aren't usually open to making changes to their infrastructure just to support a mobile app. I had mentioned earlier though you do see companies moving away from the all-reaching EMM and so it's a good idea to familiarize yourself with these technologies.



In addition to your typical need to access internal company resources, it's common for an enterprise app to support some kind of single sign on scenario (SSO). What does that mean? It gets thrown out there when you talk to companies about mobile and what do they really want? Well, once access has been granted to a user by way of an enterprise mobile app, what we would want for them then is to have a seamless access experience to other apps too. This concept works pretty well when you're on a desktop computer and you're using SAML to handle authentication and identity, but mobile presents its own problems.

SSO capabilities, like much of what we've discussed so far are leveraged by EMM providers. One alternative to support SSO would likely be to use VPN, which we already know to be cumbersome, and so keeping in mind those Microsoft Azure tunneling products from the last slide, where they not only manage secure connectivity but they also provide cloud-based identity management. What you want to find is a system that supports OAuth 2.0 or OpenID, and this includes Azure, Google Apps, and Amazon Web Services to name a few. These technologies are all able to issue tokens that provide secure access to your protected data.

Security

Why should we care about balancing user needs like ease of use and privacy with what IT demands for security? On the one hand you have all your company data secured somewhere behind the firewall or in the cloud, and on the other side you have users who are bringing personal devices to work, and wanting to access that data freely and autonomously.

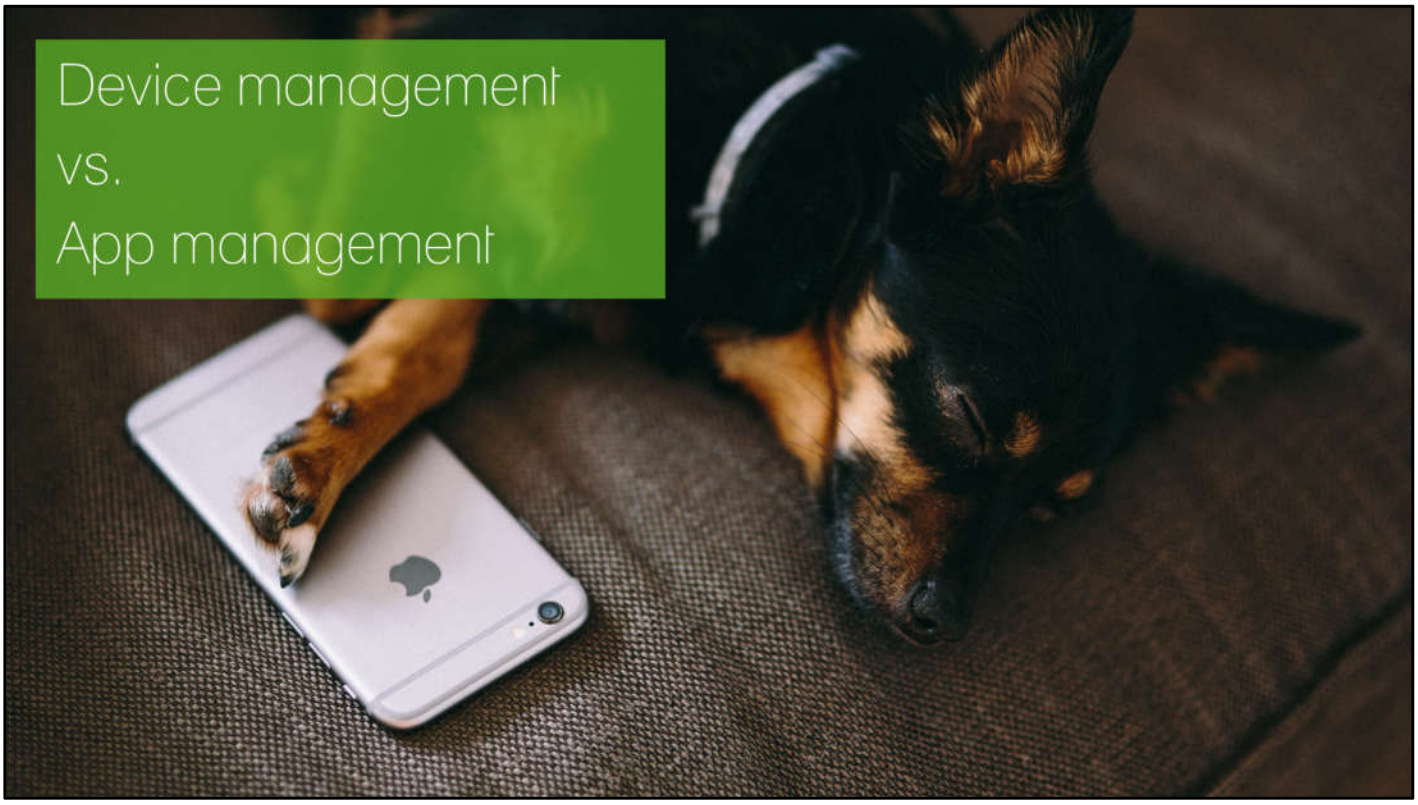
The way things should be decided and then acted upon for enterprise mobile security play out something like this:

1. We define a security policy. Remember what type of users we're looking to support? Are they BYOD? Who gets access to what?

2. We then decide how to honor that security policy from a technical standpoint. Are we using EMM or are we handpicking different technologies to create our own solution?

3. And finally you'll want to educate users on security practices.

So considering how we can best shape our security policy and then act upon it with our enterprise mobile app, let's go back to what we discussed earlier about whether people are bringing their own devices to work or using corporate owned.



We can categorize these security considerations somewhat by looking at device level security, app level security, and then you can get even more granular through content level permissions. Before we dive into MDM's, I want to quickly mention there is one very super simple device management player and that would be Exchange Active Sync. Worth knowing if you're not going the EMM route.

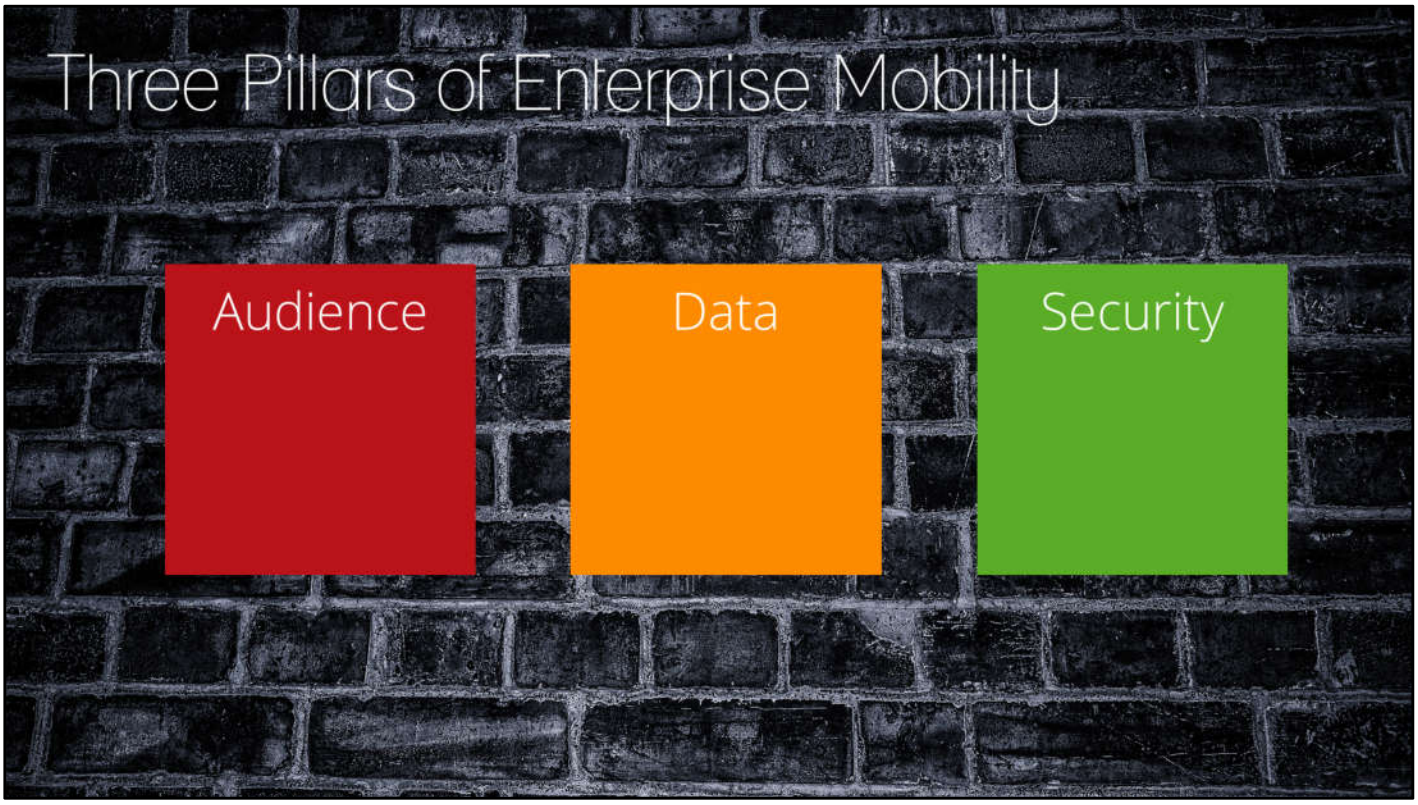
For device management and what that would mean for an app developer, there isn't much you can *do* about it, but you should be aware that this is a common factor you'll come across in secure organizations. Earlier we mentioned some EMM vendors like AirWatch and MobileIron, and companies will use these vendors to exert some control over mobile devices. This may mean for you then that you need to "support" AirWatch by doing an integration with their SDK, or at the very minimum testing your app out with their app-wrapping technology.

If we go the less intrusive route and simply do app management, this is another security layer that can be dealt with by an EMM. But it would be advantageous to understand what features companies look to restrict in the case that you build in app management yourself. Examples would be giving companies the ability to turn off Open In, disable the ability to call or email from the app, and encryption and containerization options at the content level. If you're building one app for use within your own organization, app management would simply be security measures you build into the app. If you're on the B2B path, you may find yourself in situations where certain app management components should be optional for your users.

We've saved the best for last, and that would be authentication. With authentication for enterprise mobile, there are lots of business decisions around 1. what data can be accessed by the mobile device (does it differ from say, laptops), 2. what data (if any) can be stored on the device (and encrypted), and 3. who can access what data.

When we start to consider what type of authentication works best when we again keep in mind usability vs. security, for mobile the username/password approach may not necessarily be the right answer. For secure environments the password rules we normally apply when people are on a computer can be a serious hindrance to someone typing on a phone screen. Things like length, complexity, and expiration. You may consider non-text authentication by way of drawing patterns or bio-metric solutions like using fingerprint or voice, but in the real world what you commonly see in the realm of enterprise mobile is that certificate-based auth plus requiring a device lock is a solid multi-factor solution. Don't forget also that if we're having to support SSO then we'll want to use a token-based approach.

So with the cert-based approach the company uses a certificate authority that issues certificates to their users by way of mobile technologies we know and love like MMS and SMS. You might be surprised to hear again, finally, that many people do this by way of using an EMM.



To bring us back around, in conclusion:

There are three pillars of enterprise mobile app development. As long as you have an understanding of each you can reach success in implementing apps for business.

-Our audience for a B2B/B2E app differs from a B2C app in a number of ways, notably the way in which people use their mobile device and how the app gets distributed to them.

-If you have an enterprise app you're most assuredly connecting and making use of some kind of enterprise backend system. If you're lucky, it's in the cloud with mobile-ready endpoints. Otherwise, you'll need to be adept at working with more legacy options.

-Security considerations are important not just because you need to protect data, but you must protect your users and shield them from the frustrations that go into supporting secure solutions.


Questions?

Useful links

- AppConfig community: http://appconfig.org/
- Enterprise mobile access: Considerations for two-factor mobile authentication: http://searchsecurity.techtarget.com/tip/Enterprise-mobile-access-Considerations-for-two-factor-mobile-authentication
- Three considerations for planning your mobile device authentication: http://asmarterplanet.com/mobile-enterprise/blog/2013/08/three-considerations-for-planning-your-mobile-device-authentication.html
- Top 4 ways to add single sign-on to enterprise mobile apps: http://techbeacon.com/needs-revision-4-best-practices-adding-single-sign-enterprise-mobile-apps
- #AzureAD: Certificate based authentication for iOS and Android now in preview!: https://blogs.technet.microsoft.com/enterprisemobility/2016/07/18/azuread-certificate-based-authentication-for-ios-and-android-now-in-preview/
- Azure Insider - BYOD Challenge: Connect Mobile Devices to the Enterprise: https://msdn.microsoft.com/en-us/magazine/dn890373.aspx?f=255&MSPPError=-2147217396
- The Best Mobile Device Management (MDM) Solutions of 2016: http://www.pcmag.com/article/342695/the-best-mobile-device-management-mdm-software-of-2016
- 50 Top Rated Websites for Royalty Free Stock Images: http://www.smallbusinesswebdesigns.net.au/royalty-free-images.html
- An Essential Overview of the Mobile Application Enterprise Environment: https://onehundred15.files.wordpress.com/2015/05/anessentialoverviewofthemobileapplicationenterpriseenvironment.pdf