MOBILE MICROAPPPS

The shortest path to enterprise mobility

Abstract

In the most general case, enterprise mobility is a tough problem requiring a big investment in the IT infrastructure, skills, and processes. The cost is prohibitive for many companies. The net result is a relatively small number of truly transformative enterprise-wide mobility deployments. Fortunately, the scope of the problem can be dramatically reduced by making small adjustments in some of the underlying assumptions.

MightyCloud Inc.
hello@mightycloud.com
Background

Enterprise Mobility Maturity Model

Several studies have pointed to a strong link between the business performance of a company and its maturity in the area of enterprise mobility. Companies with well-articulated enterprise-wide mobility strategies demonstrate higher revenue growth and better operating margins.

![Graph showing revenue growth and operating margin](image)

The nature of the cause and effect relationship between mobility and performance is open for discussion. Still, it may be informative to construct a model to show how a company might progress from a lower maturity to a higher maturity in enterprise mobility.

Based on what we see in the industry, such a maturity model could consist of three distinct phases: aware, reactive, and proactive.
The Aware Phase. At this point, with the 80 percent penetration of mobile phones in the United States, the majority of companies have been exposed to mobility and are aware of it. Some organizations have not yet had an opportunity to do anything about it. They are in the aware phase. Employees at these companies bring their own devices to work (BYOD) and install applications of their choosing (BYOA). They upload content to Dropbox and take meeting notes in Evernote. In spite of the lack of IT involvement, or maybe thanks to it, mobility at this stage has a positive effect on employee productivity. At the same time, due to the widespread use of third-party applications and cloud services, confidential information may be at risk.

The Reactive Phase. Realization of the risks associated with BYOD and BYOA puts IT in the reactive mode. They take action by deploying enterprise mobility management suites and issuing detailed BYOD policies. Use of third-party applications is limited to the ones approved by the IT department. There is little or no custom application development. Organizations issue “managed” mobile devices to employees. People begin carrying multiple devices, personal ones and the ones provided by the company. Companies in the reactive phase achieve a higher level of information protection and risk management. At the same time, their actions may actually have negative impact on the overall employee productivity.

The Proactive Phase. The pendulum swings in the opposite direction with employees demanding greater productivity, privacy, and control. The pressure often comes from the CXO office. Many CIOs are beginning to understand the limitations of mobile device management and transition to management of mobile content and applications. This is often achieved with the help of containerization of company-owned resources on devices that may be controlled by the employees. The new approach and mindset puts companies in the proactive mode. Many of them begin development of custom mobile applications that dramatically improve employee productivity and increase business velocity.

Nothing prevents a company from transitioning directly from the aware phase into the proactive phase in this maturity model. In fact, a growing number of enterprises are learning from the mistakes of their peers and doing just that.

Importance of Custom Application Development

While off-the-shelf productivity applications may become popular with employees during the aware phase of our maturity model, their effect on the overall company performance is limited. To achieve a noticeable impact in the proactive phase, companies must leverage mobility to optimize and transform their core business processes. Since these processes are usually unique to the company, their optimization and transformation requires custom application development.

The need for application modernization is yet another factor that is driving the development of custom applications. A great number of enterprise applications that are in use today were built in the nineties for the Windows platform. The emergence of the Internet did nothing to shake Microsoft’s dominance. Companies were under no pressure to replace their legacy applications. The smartphone revolution changed everything. Mobility introduced significant platform fragmentation, which is forcing enterprises to adapt their applications for the new environment.
State of the Art in Mobile Application Development

Application Architecture
At some point during the application development cycle, the development team must decide on the implementation approach. Three options are available: native, mobile web, and hybrid.

Native Applications
Native applications are developed using the toolset provided by the mobile platform vendor. They have direct access to the device hardware and the platform APIs. They are compiled into machine code before execution. This guarantees excellent performance. Developers optimize the end-user experience by following guidelines published by the platform vendor. Native applications developed for one platform cannot be launched on another platform. Native applications are distributed via app stores, which may lengthen the product release cycle.

Mobile Web Applications
Mobile web applications are developed in HTML and JavaScript. They are designed to support a broad range of mobile platforms. A web browser is used to execute the application and render its user interface. Consequently, many application parameters, such as its speed, responsiveness, and security, depend on the quality of the browser. The end-user experience is optimized for consistency across platforms and may not feel “native” on any particular platform. A mobile web application has no direct access to the device hardware and platform APIs. Web applications are distributed via URLs, instantly and with little overhead.

Hybrid Applications
Hybrid applications are mobile web applications that are wrapped in a layer of native code. The native code is used to provide access to the device hardware and platform APIs in a consistent manner across platforms. The native wrapper may also be used to address some of the browser shortcomings, such as lack of robust security or manageability. When wrapping a mobile web application, a separate hybrid application is created for each mobile platform. This increases the cost of the project, but not as much as in the case of native development. Hybrid applications are distributed via app stores.
Mobile Application Management

Once an application is developed, it must be launched in the enterprise. The process consists of several steps collectively known as mobile application management (MAM):

1. Add an app to the system of record (“app onboarding”)
2. Inspect the app to make sure it is safe (“app inspection”)
3. Secure the app with policies (“app protection”)
4. Add credentials to the app so that it can run properly (“app signing”)
5. Distribute the app to all the end users (“app deployment”)
6. Track the app usage and collect user feedback (“app analytics”)
7. Update the app on a regular basis (“app administration”)

Mobile Content Management

Many companies are developing mobile applications with the objective of improving their business processes. In a great number of cases, business processes are used to create, review, catalog, and publish digital content: forms, documents, presentations, images, videos, etc.

In order to provide secure access to content on mobile devices, a company must implement a mobile content management (MCM) solution. It consists of:

1. A content repository
2. A middleware layer that connects to external repositories
3. Mobile applications that provide access to content stored in these repositories
4. APIs and SDKs for accessing content in custom applications
5. Admin tools for managing content and defining access control policies

In the traditional mobility architecture, mobile content management is a discipline that is entirely separate from mobile application management.

Identity and Access Management

User identities and user roles form the foundation for security policies governing access to mobile applications and content. These artifacts are managed in the identity and access management (IAM) system, which includes:

1. A user directory
2. A directory of resources protected by the system
3. A library of access control policies
4. A tool for managing user identities, roles, resources, and security policies
5. Policy enforcement modules and SDKs
6. Extensions for supporting common protocols, such as SAML, Kerberos, etc.

Many IAM systems that are deployed today lack compatibility with cloud services and mobile applications. They must be either upgraded or replaced as part of the enterprise mobility effort.
Mobile Backend as a Service
A mobile application may require support from a number of cloud services:

1. Cloud storage
2. Persistence
3. Push notifications
4. Collaboration
5. Social networking

A number of vendors offer bundles of such services along with the corresponding SDKs for the various mobile platforms. The bundle is called a mobile backend as a service, or mBaaS.

Putting It All Together
In summary, before a company can begin mobilizing its business processes and modernizing its legacy applications, it must implement:

1. A mobile application management (MAM) solution
2. A mobile content management (MCM) solution
3. An identity and access management (IAM) system
4. A set of backend services supporting mobile apps (mBaaS)

The exact set of technologies depends on the business requirements—the number of mobile platforms that must be supported, the application architecture selected by the company, and other parameters.

This list of prerequisites may seem prohibitive to many IT managers. As a result, we are facing a crisis. On one hand, companies must build mobile applications in order to compete and grow. On the other hand, many of them lack the foundation required to complete this effort.
Microapps to the Rescue

Native vs Hybrid in the Enterprise

In the consumer space, where intense competition among many similar apps is forcing developers to reach for every trick in the bag, native apps are winning over mobile web and hybrid apps. Gaming is the top category. A game developed in Objective-C will always feel snappier and jazzier than the one written in JavaScript.

In addition, native vs. hybrid has always been a matter of developer productivity. Hybrid apps have little to offer the end users over native apps, and developer productivity matters little in the world of consumer apps, where well-funded start-ups can focus on gaining traction on a dominant platform.

The situation is radically different in the world of enterprise mobility. An IT manager deploying an employee-facing app in a BYOD environment must support multiple mobile platforms: iOS, Android, Windows, etc. Developer productivity has direct impact on the budget, and the ability to write code once and run it on all platforms is very important.

It is not surprising, therefore, that a recent survey of IT decision makers conducted by TechValidate discovered that native apps are trailing mobile web and hybrid apps in the enterprise by a large margin.

![Bar chart showing mobile architecture preferences for 2015]

What mobile architecture(s) do you intend on using in 2015?

- Mobile Web: 60%
- Hybrid: 85%
- Native: 26%
- Other: 4%

Note: This is a multiple-choice question – response percentages may not add up to 100.

Source: TechValidate survey of 178 IT Decision Makers

Preference for hybrid and mobile web apps may also be explained by the different nature of applications developed in the enterprise. The same survey reported that the list of the top five categories of apps in the enterprise included process automation (forms, routing), reporting and analytics, executive dashboards, CRM, and collaboration.
As far as forms and reporting are concerned, mobile web and hybrid applications offer perfectly adequate user experience and performance.

**Mobile Microapps**

By limiting the scope of the problem to mobile web and hybrid apps only, we can dramatically simplify the enterprise mobility solution.

The issue of platform fragmentation is automatically addressed by the cross-platform nature of web development. Code can be built, debugged, and tested on one platform and deployed on devices of all kinds across the enterprise.

Management and security of mobile applications is still a concern. The issue is poorly defined for mobile web apps. Management and security of hybrid apps presents the same set of challenges as management and security of native applications. There is no automatic gain.

There is no gain, that is, until we realize that instead of wrapping one mobile web app in native code to produce a single hybrid app, we can wrap several mobile web apps in one native envelope. For instance, we can wrap all apps required to support a certain workflow process.

We can significantly reduce the management and security overhead by managing and securing a set of related applications as a group.

We can take this line of thinking a step further and turn our native envelope into a generic container that can host all kinds of mobile web apps. We can transform a mobile web app into a hybrid app by simply throwing it into this container.
We will use the term **microapp** to refer to mobile web apps that are deployed in a container.

**The Microapp Container and Platform**

The microapp container is a native application that can host any number of mobile web applications. We can create one such container per mobile platform in our environment: iOS, Windows, Android, and others. The container isolates applications from idiosyncrasies of the various mobile platforms. It also provides access to a set of shared services. Implementation of these services may be local to the container. Services can also be hosted in the cloud. They form the microapp platform. In the following sections, we describe some of these services.

**Meta-Data Management**

Applications deployed in a container have a set of common attributes such as name, description, version number, and others. This information is maintained with help of the meta-data management service. It is the system of record for all microapps deployed in the enterprise. Each microapp is mapped to an asset managed by the service.

**Content Management**

The microapp container can host not only applications, but also content used by these applications: forms, documents, images, and videos. Each content item is mapped to an asset in the meta-data repository. From the perspective of the meta-data management service, **microapps and content are different types of assets that can be managed in a rather uniform manner**. In addition to meta-data management, the content management service provides a set of features that is typically associated with mobile content management: a content repository, a content integration layer, and others.
Identity and Access Management
Microapps and content assets are uniformly viewed as security resources that can be protected by the container. The container acts as an enforcement point for policies expressed in terms of user identities, user roles, device location, time of day, and other attributes. User identities and security policies are managed with the help of the identity and access management service provided by the microapp platform. This service can be integrated with other elements of the company’s security infrastructure, such as an instance of the Active Directory or a SAML provider.

Collaboration
The powerful mechanism for expressing security policies and the uniform approach to management of applications and content enable rich collaboration. Applications and content assets can be organized into collections, a special type of assets supported by the meta-data service. Collections can be nested. Access to collections is controlled with the help of security policies. Items can be shared by placing them in collections and granting access to these collections to users and groups.

Persistence
The meta-data management service can be used by the microapps to store and retrieve arbitrary JSON objects. Each application is automatically allocated an asset collection for this purpose. Sub-collections can be created if necessary. The same security mechanism that is used to protect apps and content can be used to control access to objects created by the application.

Event Management
The event management service provides a cross-platform publish-subscribe mechanism. Microapps can create event channels, subscribe for events, and publish events. Due to the intermittent network connectivity that is common for mobile devices, a microapp may not be reachable when an event is posted. The service takes care of storing the event and delivering it when the app comes back online.

Real-Time Communication
The real-time communication service supports video and audio communication between microapps running on different devices. The service implements the WebRTC standard. Communication can be performed either in peer-to-peer mode or via a cloud-based gateway.

The Platform Abstraction Layer
The platform abstraction layer provides transparent and secure access to the APIs of the underlying mobile platform. Microapps use this layer to access the device camera, list content in the picture gallery, read GPS coordinates, search for entries in the address book, and perform other platform-related tasks in a platform-independent manner.

Reporting and Analytics
Information about the app and content usage is automatically collected by the container. This information is aggregated and presented in the form of reports and diagrams. System administrators, business managers, and application developers can use them to derive valuable insights, tune the apps, improve the content, and optimize business processes.
Solutions
The microapp architecture is very versatile and can be used to support a number of common solution patterns such as collaboration, enterprise application integration, and business process management.

Mobile Collaboration
In the case of collaboration, we have a group of users, each with a mobile device that is running a microapp container. The container is populated with microapps that are used to share content, exchange messages, and communicate via audio and video.

The container requires user authentication. The user identity and profile information is available to the microapps and can be used to customize and enhance user experience. A messaging app, for example, can automatically add a user photo to all messages posted by the user.

When an instant message is posted on one device, the event management service can be used to propagate it to all other devices. The persistence service can be used to maintain a persistent stream of messages that users join and leave as needed. The real-time communication service can be used to establish a peer-to-peer link for streaming video and audio between devices.

Enterprise Application Integration
Enterprise applications can be integrated at the data level, application server level, API level, or user interface level. The user interface level integration is often the most effective and the easiest to implement. An enterprise portal is a good example of this approach.

An enterprise portal is a web application that consists of multiple portlets, each portlet displaying information from a single backend application. Users can get the bird’s eye view of all applications’ data on the portal home screen. They can interact with the portlets and get more information if necessary.
Microapps bring a similar approach to the world of mobility. A microapp can be used to provide access to a backend application. A single backend application can be represented with several microapps. Multiple microapps can be running in one container.

The concept of the portal home screen, the main screen displaying multiple portlets, cannot be easily represented on mobile devices. What works instead is a stream of notifications with snippets of information posted by different apps. A user can scroll through the stream, review individual entries, and launch the corresponding microapps.

**Business Process Management**

A business process is a sequence of steps that is performed by the various process participants in pursuit of a common goal. A business process can be used to review and publish a document, handle an insurance claim, or inspect a piece of equipment. Completing a step in the process may require interaction with a number of backend applications and content repositories.

The time it takes to complete a process influences a number of critical performance metrics such as employee productivity, company profitability, and customer satisfaction. This time can be substantially reduced by making it possible to perform some of the steps in the process on mobile devices.

Microapps are ideally suited for the purpose. We already discussed how microapps and services provided by the microapp platform support mobile collaboration and enterprise application integration. These are important components of business processes management.

To enable real-time execution of business processes, microapps can take advantage of the platform’s event management service. When a step in the process is complete, a message can be sent to the process participants responsible for the next step.
Conclusion
Companies with well-articulated enterprise-wide mobility strategies demonstrate higher revenue growth and better operating margins. A mature mobility strategy often implies development of custom mobile applications.

In the most general case, mobile application development requires a very significant investment in the IT infrastructure, skills, and processes. For many companies, the cost of such an investment could negate any benefits they would derive from enterprise mobility. The net result is a relatively small number of truly transformative enterprise-wide mobility deployments.

Fortunately, the scope of the problem can be dramatically reduced by making small adjustments in the underlying assumptions.

By analogy with the consumer space, it is widely assumed that enterprises must be developing native applications because they are delivering better user experience and guaranteeing higher rate of adoption. At the same time, numerous studies are pointing to the fact that typical enterprise use cases can be equally well served with mobile web and hybrid applications.

By limiting the scope of the problem to mobile web and hybrid apps, one can significantly simplify the overall architecture of an enterprise mobility solution.

Multiple related applications can be hosted in a single cross-platform container. They can be managed and secured as a group. Applications and content can be uniformly treated as IT assets. This simplifies the task of defining and enforcing security policies and enables non-trivial application and content sharing. The application container reduces the cost of application development by providing transparent access to the mobile platform APIs and cloud services.

Companies can use the proposed approach to implement sophisticated solutions for mobile collaboration, enterprise application integration, and business process management. They can do so while preserving their investments in the existing IT infrastructure, skills, and processes.

About MightyCloud
MightyCloud (www.mightycloud.com) is an enterprise mobility platform that dramatically simplifies mobile application and content management. MightyCloud pioneered the idea of mobile microapps and developed the world’s first secure and manageable microapp container. MightyCloud container and cloud services enable cost-effective development, deployment, and management of sophisticated solutions for mobile collaboration, enterprise application integration, and business process management.