# Table of Contents

1 Executive Summary 3
   1.1 MEAP Background 4

2 Key Findings 5

3 Mobile Middleware 7
   3.1 The Role of Mobile Middleware 7

4 The Mobile Middleware Ecosystem: Tools and Technologies 9
   4.1 Overview of the Mobile Enterprise Middleware Ecosystem 9
   4.2 Application Development Tools 10

5 Implementation Models 14
   5.1 Model #1 16
   5.2 Model #2 18
   5.3 Model #3 20
   5.4 Model #4 22

6 Recommendations 23
1 Executive Summary

Today, enterprise mobility is a high priority in most large organizations. In order to meet the demand for mobile solutions by their own internal staff, organizations must understand, implement, and maintain mobile middleware solutions that will enable them to efficiently integrate with back office systems and support many different mobile devices, mobile applications, and mobile operating systems. The mobile middleware solution category that has emerged to meet this demand is commonly referred to as mobile enterprise application platforms (MEAPs), an acronym coined by Gartner.

As mobility has evolved into a core strategic imperative, the role of the MEAP has become an issue of vital importance. Companies that are considering the purchase of a MEAP are faced with a diverse array of products and services that vary dramatically in their solution architecture, technical capabilities, and core functionality. This wide selection of products and the differences among them complicates the selection process and makes it difficult to decide which MEAP to support.

Enterprises will find it valuable to standardize on one MEAP when possible, and when not possible, standardize on as few as possible to help reduce costs, complexity, and the requirements for training and support.

The MEAP is most often used as a mobile middleware layer between back office systems (ERPs, business applications, and databases), and mobile applications. Different mobility vendors have different definitions for MEAPs, but commonly MEAPs are recognized for having the following capabilities:

- Mobile application development tools: integrated development environments (IDEs), templates, forms builders, and libraries
- Support for developing once and deploying to many different tablet and smartphone operating systems such as iOS, Windows Phone, Blackberry, and Android
- Connection management
- Support for online or offline mobile apps (mobile client data storage)
• Data synchronization
• ERP, database, and business application integration between mobile apps and backend systems
• Mobile app deployment

The MEAP plays a valuable role today and as mobile technologies continue to evolve and expand it will become an even more important component of the IT infrastructure, whether on-premise or in the cloud. This report provides a description of MEAPs and their general components. It then considers the different types of mobile application development tools available, the value they deliver, and the primary questions and variables a company must consider when selecting development solutions. The second component of this analysis examines the leading MEAP providers, lending insight into their core product offerings as well as their business strategies.

1.1 MEAP Background

In the past, MEAPs were often invisible and embedded in tactical mobile enterprise solutions such as service management, proof-of-delivery, and mobile enterprise asset management solutions. Each mobile vendor would include their own proprietary MEAP with their solution, so the more mobile apps used inside the enterprise, the more MEAPs were added to the inventory of supported products. This was not a desired, efficient, cost effective, or sustainable model. In fact, multiple mobile clients from different MEAP vendors running on the same mobile device often conflicted with each other, resulting in performance issues.

Enterprises today are seeking ways to expand their support for mobility, while at the same time limiting the numbers of different MEAPs that need to be supported. They want to standardize on one MEAP, when possible, so they can simplify the design, development, deployment, security, and support for enterprise mobility solutions. That said, many companies will require more than one MEAP in order to satisfy the needs of its various constituents.
2 Key Findings

• Enterprise mobility is a key strategic imperative for most large companies today. Choosing the right technologies, architectures, middleware, and frameworks to support the plethora of requirements and demands from all the various business units is often a daunting and high-pressure process that can be very confusing for those just starting down this path.

• Understanding the major components of an enterprise-class mobile solution will help you decide which are most important in helping you meet your specific objectives.

• The core enterprise mobility technology components include:
  - MEAP
  - Mobile device management (MDM)
  - Mobile application client runtime
  - Mobile virtual private network (VPN)
  - Mobile database solutions for offline data storage
  - Private app store
  - Hosting

• The core capabilities of MEAPs include:
  - Rapid mobile application development tools, templates, and libraries
  - Support for developing once and deploying to many different smartphone, tablet, handheld computer, and laptop operating systems such as iOS, Windows Phone, Blackberry, and Android.
  - Connection management
  - Support for online or offline mobile apps (mobile client data storage)
  - Data synchronization
  - ERP, database, and business application integration between mobile apps and backend systems
  - Mobile app deployment
• The role that the MEAP plays is an issue of critical importance to most enterprises and its strategic value includes:
  - Rapid design, development, and deployment of cross-platform apps
  - Efficient support of multiple mobile devices, mobile applications, and mobile operating systems
  - Publishing and deployment capabilities across mobile operating systems
  - Enterprise-grade security
  - Data integration capabilities with backend ERPs and other business systems

• There is a diverse array of products and services that vary dramatically in their solution architecture, technical capabilities, and core functionality. This wide selection of products and the differences among them complicates the selection process, requiring companies to conduct a considerable amount of research and analysis prior to investing in a MEAP.

• The right mobile solution for an enterprise depends on the specific requirements and implementation model. We will cover four (of many possible) implementation models in this report, each identifying critical considerations.

• The decision to take an HTML5, HTML5 hybrid, or native application approach to development should be made on a case-by-case basis. There are distinct advantages and disadvantages to each.

• Formulating a mobile strategy is critical and companies should begin by asking a series of questions to define the purpose of their mobile initiative and their core requirements.

• ROI should be evident and well defined for any mobile initiative.
3 Mobile Middleware

3.1 The Role of Mobile Middleware

According to recent survey results from Netcentric Strategies, over 90 percent of companies responding to the “Enterprise Mobility Survey 2011,” plan to implement mobile enterprise applications in 2012. 41.6 percent of those companies expect to implement six or more applications and 23.4 percent expect to implement 11 or more. Supporting and managing that number of applications is very difficult without the proper tools and standardized processes in place.

As a rule of thumb, when connecting to multiple backend data sources, building multiple applications, or deploying to multiple device types, mobile middleware offers many advantages. The value associated with these solutions is centered on their ability to efficiently perform design, development, deployment, and management functions across multiple operating systems and devices thus simplifying the effort. For example, many MEAPs allow a developer to build one application and deploy it to multiple operating systems such as BlackBerry, iOS, Android, and Windows Phone.

<table>
<thead>
<tr>
<th>What is mobile middleware?</th>
<th>How is it being used?</th>
<th>What are the benefits?</th>
</tr>
</thead>
<tbody>
<tr>
<td>For the purposes of this report, mobile middleware refers to software that acts as an intermediary between enterprise applications or any back office or cloud-based data sources and mobile devices. This report focuses on MEAPs that support both native and web-based mobile applications.</td>
<td>Middleware solutions are used for a variety of purposes, including: • Developing and managing web-based and native mobile applications • Publishing mobile applications • Integrating users with backend enterprise systems, web services, and cloud-based data sources • Securing and managing mobile devices • Securing data connectivity, data transmission, and on-device application data</td>
<td>Mobile middleware solutions enable organizations to more efficiently develop, secure, and manage mobile applications and devices. Compared with native SDKs and conventional tools, developers and IT professionals can leverage middleware solutions to develop, secure, manage, support, and maintain mobile applications with fewer IT and developer resources and less specialized training, such as the ability to program in languages specific to various mobile devices.</td>
</tr>
</tbody>
</table>

Application development tools offer the ability to build mobile applications faster than conventional approaches. Cross-platform MEAPs, often referred to as “write once, deploy to many,” reduce the amount of coding required. Other solutions providers reduce the development burden even more dramatically by utilizing
web-based clients to reproduce backend functionality on the mobile device with little-to-no resources required by a company's IT staff. Our discussion of mobile middleware tools, technologies, and implementation models will explore these uses and benefits in more depth.
4 The Mobile Middleware Ecosystem: Tools and Technologies

The mobile middleware ecosystem encompasses several key technologies that work in conjunction with MEAPs. The following section provides a description of the core technologies that play a major role in enterprise mobility as well as an overview of application development tools and strategies.

4.1 Overview of the Mobile Enterprise Middleware Ecosystem

The mobile enterprise middleware ecosystem is comprised of several core technologies that work closely together. Along with MEAPs, these technologies include MDM, mobile VPN, and mobile application publishing (private app stores) solutions. The following chart describes the baseline capabilities of MEAP, MDM, VPN, and app publishing tools:

<table>
<thead>
<tr>
<th>MEAP</th>
<th>MDM</th>
<th>VPN</th>
<th>Private app stores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build apps with integrated development environments or SDKs</td>
<td>Deploy, update, install, and manage apps</td>
<td>Authenticate and manage data connections</td>
<td>Publish and manage apps</td>
</tr>
<tr>
<td>Integrate with backend systems</td>
<td>Remotely wipe devices</td>
<td>Manage roaming and persistence</td>
<td>Control access by user group</td>
</tr>
<tr>
<td>Deploy application runtime</td>
<td>Configure and provision devices</td>
<td>Encrypt data over-the-air (OTA)</td>
<td>Remotely wipe apps and data</td>
</tr>
<tr>
<td>Implement security protections</td>
<td>Implement permission and security policies</td>
<td>Data compression and performance optimization</td>
<td>Monitor downloads and usage</td>
</tr>
<tr>
<td>Test and debug apps</td>
<td>Backup and restore devices</td>
<td>Manage security policies</td>
<td>Distribute apps for multiple mobile platforms</td>
</tr>
<tr>
<td>Perform reporting and analytics</td>
<td>Perform troubleshooting and diagnostics</td>
<td>Receive network notifications</td>
<td></td>
</tr>
<tr>
<td>Log and report on usage activity</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mobile Middleware Strategies
4.2 Application Development Tools

MEAPs vary dramatically in their features and functionality. Some are open source while others are proprietary; some allow unlimited customization of code whereas others restrict developers to building apps with only the features and functionality available in the development environment. Beyond cost, perhaps the most important variable to be considered when selecting an application development solution is the amount of work required by developers to build and maintain applications versus the amount that’s shouldered by the development tools and/or company that provides the tools. Some solutions require extensive knowledge of the programming languages specific to each device platform (Java, Objective C, etc.) while others enable cross-device development in WYSIWYG environments with little to no coding skills required.

There are four primary approaches to mobile app development: web-based (non-HTML5), HTML5, hybrid, and native. Web-based (both HTML4 and HTML5) applications run through the device’s browser. Hybrids utilize a combination of web-based (HTML5) and proprietary native “containers”, while pure native apps are written in the native language of each device operating system. The following chart articulates some of the key issues concerning each approach.

<table>
<thead>
<tr>
<th>Application type</th>
<th>Benefits</th>
<th>Concerns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native applications</td>
<td>Superior UI functionality and user experience</td>
<td>A separate application must be developed and maintained for each mobile operating system</td>
</tr>
<tr>
<td></td>
<td>Integration with native OS optimizes use of native phone features and hardware</td>
<td>Updating applications can be a complex process, especially when dealing with multiple operating systems</td>
</tr>
<tr>
<td></td>
<td>Ability to store data locally enables offline use</td>
<td>Development costs are generally higher, particularly with cross-platform deployments</td>
</tr>
<tr>
<td></td>
<td>More security and authentication options available than with web-based apps</td>
<td></td>
</tr>
</tbody>
</table>
Web-based applications (non-HTML5)

<table>
<thead>
<tr>
<th>Benefit</th>
<th>disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leveraging the browser simplifies cross-platform deployment</td>
<td>Applications require an active network connection at all times in order to run</td>
</tr>
<tr>
<td>Upgrading and version updating is easy and changes are immediate</td>
<td>Since data is not stored locally, offline data storage and functionality is not possible</td>
</tr>
<tr>
<td>Development costs are generally lower than native</td>
<td>Since the application is not integrated with the OS, the UI capabilities are less robust than native apps</td>
</tr>
<tr>
<td>Lost, stolen, or compromised devices pose less of a security threat because data cannot be stored offline</td>
<td>Fewer security options are available than with native applications</td>
</tr>
</tbody>
</table>

Hybrid

<table>
<thead>
<tr>
<th>Benefit</th>
<th>disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combines the benefits of HTML5 with the power of a native application</td>
<td>Requires additional development, support, and maintenance costs and usually requires vendor software licenses and maintenance fees</td>
</tr>
</tbody>
</table>

HTML5 applications

<table>
<thead>
<tr>
<th>Benefit</th>
<th>disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplified cross-platform development, it takes about half the time of a native application to develop</td>
<td>Limited UI capabilities compared with native applications</td>
</tr>
<tr>
<td>Easier upgrading and version updating in real time</td>
<td>Slower performance than applications programmed in native code</td>
</tr>
<tr>
<td>Development costs are generally lower than native</td>
<td>Fewer security options are available than native applications</td>
</tr>
<tr>
<td>Offline functionality and data storage</td>
<td>Not as robust data storage capabilities when compared to other specialized mobile database options</td>
</tr>
</tbody>
</table>

Since it leverages a device’s native browser, the web-based approach yields a product that runs across multiple device operating systems but has more limitations in terms of functionality than a native app. For example, non-HTML5 mobile web sites and browser apps cannot cache data locally or run offline, so without an active connection the app will not perform. Although HTML5 resolves many of these limitations, it does not eliminate all of them.
The native approach results in an app that is tightly integrated with the OS and capable of a more robust user experience than web-based apps can provide. Native applications can run offline and data can be stored locally, which is essential for employees who travel in and out of network coverage. The native approach, however, can be far more expensive since a different application must be developed, deployed, supported, and maintained for each mobile operating system.

The average lifecycle of a mobile consumer application is fewer than 30 days. While mobile enterprise applications generally last longer, their lifespan is significantly curtailed by the rapid evolution of devices and their operating systems. Updating applications is a key component of mobile application development that is often overlooked, and selecting a development model with a short release cycle can provide several strategic advantages. Building and deploying small but targeted high-value apps in a relatively short timeframe allows an organization to recognize ROI early on in the development process. Following an initial release with more complex functionality enables that organization to be more responsive to the needs of their users and to build apps that continuously add value.

In general, web-based applications are easier to update than native apps, particularly if you are dealing with an application that is meant to run across multiple device types and operating systems. From a developer’s perspective, making a single change to a browser-based app or mobile-optimized website will immediately result in that change taking effect across all compatible device types. Making that same change to a native app entails updating all relevant code bases, redeploying, and reinstalling those applications.

Choosing an approach to app development involves weighing a series of pros and cons relevant to your particular circumstances. Many companies today are developing for multiple groups of users, including both B2C (business to consumer) and B2E (business to employee). Depending on the in-house expertise, budget, and required functionality, it often makes sense for organizations to utilize a combination of web-based, hybrid, and native development approaches. For example, the native approach may work best for a B2E app that locally stores large image-based product catalogues, product specs, and troubleshooting information for field service technicians to use when working off-site and out of range of cellular
networks. That same company, however, may choose a web-based approach to build a mobile commerce application for reaching customers and promoting new products and services across multiple devices and operating systems.

Key considerations for selecting the right tools:

1. What skill sets are required? How advanced do your IT employees have to be in order to use the technology? (i.e. do we need computer engineers, programmers, analysts, web developers, etc.?)
2. How capable is the IDE? Can it be used for all of your development and support needs or just a percentage?
3. What security options does it provide? Do you need to purchase a separate MDM solution?
4. How does it connect to data sources and does it provide out of the box connectivity to the sources I need?
5. Does it provide MDM functionality?
6. What are the UI, graphics, and customization capabilities?
7. Is the tool supported in your region and language?
5 Implementation Models

In order to identify vital issues related to mobile implementations, this section lays out several implementation models that take into account a variety of common needs, including the number of apps being developed, who they are being developed for, and the devices they will run on. The following four implementation models are:

- **Model #1**: Developing one app for employees to run on one kind of company-issued device
- **Model #2**: Developing multiple apps for employees to run on one kind of company-issued device
- **Model #3**: Developing one cross-platform app for employees to run on a wide variety of devices
- **Model #4**: Developing multiple cross-platform apps for employees or customers to run on a wide variety of devices

Each model is described in detail and primary issues that a company must consider are articulated for each case.

It is important to note that many vendors offer pre-packaged, or “off-the-shelf” applications, which can simplify development efforts and lead to a faster time to implementation. Off-the-shelf applications provide several advantages over custom built applications, including:

- Low (or no) development cost
- Rapid implementation
- Streamlined data integration (often apps come pre-integrated)
- Ongoing support and updates provided by the vendor

While off-the-shelf applications offer these important advantages, they have their limitations as well. Most notably, off-the-shelf solutions are typically designed to mobilize a specific vendor’s system, so they are limited in their ability to be customized. Also, since vendors are responsible for the app design, it is not usually possible to incorporate outside data sources or functionality into the app. In other words, what you see is what you get.
Given that off-the-shelf applications are most often pre-built and pre-integrated, it is important to consider them as viable alternatives to custom built applications in virtually any implementation model, particularly when a company only requires one application or deployment to small numbers of users to fulfill its mobility needs.
5.1 Model #1
Developing one app for employees to run on one kind of device

**Development options:**
Companies can develop the app in-house if resources are available. They can utilize SDK’s or free development tools. However, companies must consider if it is worth training internal developers for only one application. In many cases it is simply not cost effective to develop in-house competency for just one application.

If development resources are not available, outsourcing the project to a development firm is a good alternative.

Many companies make it a policy to contract with mobility experts to develop their initial mobile application using an IDE that can be turned over to the IT department to edit and support the app after the initial development effort is complete.

**MDM:**
If the current system utilizes a BES or Exchange server, the company’s MDM requirements will likely be fulfilled by those tools. Proper authentication parameters should be built into the app in order to ensure that only authorized users are accessing the application even in the case of a lost or stolen device, and that IT has the ability to control access by centrally managing employee credentials to the app and/or the backend systems it ties into.

**Hosting:**
Most likely, for one application being accessed only by employees, contracted hosting services are not necessary.

**Choices that need to be made:**
- Do we have the internal resources to build and then maintain and update this application?
- Can we host the application internally?
- What security parameters are necessary?
- How do we protect company data when devices are lost or stolen?
- Can BES and/or Exchange Server provide the MDM functionality necessary?
• How are we going to provide access to backend systems?
• Is there an existing packaged application available that is offered by a software provider (e.g. Salesforce.com Mobile, Oracle Sales Assistant, SharePoint Mobile, etc.) and if so, does it provide the necessary functionality and security?
• Ongoing support – which business budget will support future upgrades and maintenance?
5.2 Model #2
Developing multiple apps for employees to run on one kind of device

*Development options:*
If in-house development resources are available, then custom coding with available IDEs and SDKs using internal staff augmented by external mobility experts is a reasonable option. It will enable you to build up your in-house development competency for future projects.

If development resources are not available, outsourcing development to third party mobility experts is a good alternative. However, if companies anticipate needing many different mobile applications on a permanent basis, developing in-house skills using a standardized MEAP with an IDE is a useful strategy.

*MDM:*
If the current system utilizes a BES or Exchange server, the company's MDM requirements will likely be fulfilled by those tools. Proper authentication parameters should be built into the app in order to ensure that only authorized users are accessing the application even in the case of a lost or stolen device, and that IT has the ability to control access by centrally managing employee credentials to the app and/or the backend systems it ties into.

*Hosting:*
Most likely, for one application being accessed only by employees, contracted hosting services are not necessary.

*Choices that need to be made:*
- Do we have the internal resources to build and then maintain and update multiple applications?
- Can we host the applications internally?
- What security parameters are necessary for each app?
- How do we protect company data in the event of a lost or stolen device?
- Can our BES and/or Exchange Server provide the MDM functionality necessary?
- How are we going to provide access to backend systems?
• Is there one or more packaged application(s) that we can use to meet some of our needs?
• Ongoing support – which business budget will support future upgrades and maintenance?
5.3 Model #3
Developing one cross-platform app for employees to run on a wide variety of devices

Development options:
If in-house development resources are available, custom coding web based or native applications with available SDK’s or free development tools is an option, but companies should be aware there will be a significant amount of future work associated with maintaining the code base(s). The availability of quality development resources and on-going budgets are critical factors that will determine project success.

If in-house development resources are not available, outsourcing development to a third party mobility expert is a good option. However, companies should be aware there are significant long term costs to maintaining multiple code bases and should only contract with third party developers that are sure to be available in the future to update the apps for new operating systems and device types. In addition, it is highly recommended that you required third parties to align with your development standards and MEAP strategies to ensure a smooth transition to in-house maintenance and support.

MDM:
Although multiple device types and operating systems will need to be accounted for, it is likely that a BES or Exchange server can fulfill the MDM requirements. Again, proper authentication parameters should be built into the app in order to ensure only authorized users are accessing the application even in the case of a lost or stolen device, and that IT has the ability to control access by centrally managing employee credentials to the app and/or the backend systems it ties into.

Hosting:
Most likely, for one application being accessed only by employees, contracted hosting services are not necessary.
Choices that need to be made:

- Do we have the internal resources to build and then maintain multiple code bases?
- Will we be able to ensure compatibility with new operating systems, upgrades, and device types as they are released onto the market and adopted by end users?
- Can we host the application internally?
- What security parameters are necessary?
- How do we protect company data when devices are lost or stolen?
- Can a BES and/or Exchange Server provide the MDM functionality necessary?
- How are we going to provide access to backend systems?
- Is there an existing mobile application provided by a software vendor (e.g. Salesforce.com Mobile, Oracle Sales Assistant, SharePoint Mobile, etc.) and if so, does it provide the necessary functionality and security?
- Ongoing support – which business budget will support future upgrades and maintenance?
5.4 Model #4
Developing multiple cross-platform apps for a wide variety of devices.

Development options:
If in-house development resources are available, a cross-platform development tool (MEAP) should be used, and in-house competencies should be developed.

If development resources are not available, finding a long term and strategic development partner is essential. It is important to require your third party development partner to use a MEAP and IDE of your choice so you can edit, update and control future development efforts.

MDM:
A full-featured MDM solution is likely the best option.

Hosting:
Hosting services should be considered if the strain on IT is significant.

Choices that need to be made:
- If we have the internal resources, how are we going to build and then maintain multiple code bases?
- What tools are necessary to do this effectively?
- How do we ensure compatibility with new operating systems, upgrades, and device types as they are released into the market and adopted by end users?
- Can we host the applications internally?
- What security parameters are necessary?
- How do we protect company data in the event of a lost or stolen device?
- What MDM solutions are available that fit our needs and provide the required functionality?
- What is the cost/benefit of purchasing an MDM solution?
- How are we going to provide access to backend systems?
- Is there one or more packaged application(s) that we can use to meet some of our needs?
- Ongoing support – which business budget will support future upgrades and maintenance?
6 Recommendations

In a recent survey conducted by *InsiderResearch* titled “Mobile Outlook 2012,” survey participants were asked to describe their mobile strategy. Only 33 percent report having an enterprise-wide mobile strategy, and 52 percent report little to no oversight of mobile strategy. This is a problem given that nearly all analysts list enterprise mobility as a top three strategic initiative. In another survey published in October of 2011 titled “Enterprise Mobility Survey 2011,” 80 percent of respondents said enterprise mobility was very important or critical to the future success of their companies.

These survey results reflect a large gap between the recognized importance of enterprise mobility and the actual progress that companies are making towards developing a comprehensive mobile strategy.

**Mobile Strategy in the Enterprise: Three Year View**

![Bar chart showing mobile strategy progression]

- **Currently**: 28% have no mobile strategy, 6% have an experimental approach, 16% have a distributed approach, and 32% have an enterprise-wide strategy.
- **Three years from now**: The distribution is 6% no mobile strategy, 18% experimental, 67% distributed, and 6% enterprise-wide.

(Source: InsiderResearch Mobile Outlook 2012)
One explanation for this disconnect is that the IT organization cannot develop a mobile strategy until the business first articulates how they want to use mobile solutions. Without a comprehensive mobile strategy, however, enterprises will find it difficult to know what mobile technologies, frameworks, architectures, or platforms will be most appropriate and beneficial for their enterprise.

Mobility vendors and services companies would be well served this year to focus on educating clients and helping them develop both a business and a mobile strategy to advance purchasing cycles.

Once the business has documented how they want to use mobile technologies, the IT organization needs to develop their own enterprise mobility strategy that demonstrates how they will achieve their business objectives. There are many challenges involved here as the following chart demonstrates.

**Top Challenges to Building Mobile Strategies**

(Source: InsiderResearch Mobile Outlook 2012)
According to this survey, once the mobile strategy and business cases are documented and prioritized, the next biggest challenges include choosing the right platforms/technologies and finding a budget. Why are platforms and technologies a challenge? There is a broad array of mobile solutions that play a role in the enterprise strategy. Many solutions, such as MEAPs, are complex and require extensive analysis in order to ensure that they contain the features and functionality necessary for fulfilling the stated business and IT requirements. In the Executive Summary of this paper we listed the common components of a MEAP as:

- Rapid mobile application development tools, templates, and libraries
- Support for developing once and deploying to many different tablet and smartphone operating systems such as iOS, Windows Phone, BlackBerry, and Android
- Connection management
- Support for online or offline mobile apps (mobile client data storage)
- Data synchronization
- ERP, database and business application integration between mobile apps and backend systems.
- Mobile app deployment

How important are each of these components? Enterprises were asked to rank a number of considerations according to their importance to mobile strategies. Five of the seven considerations are related to features found in most MEAPs. The other two (application maintenance and management), can be found in either MEAPs or MDMs.
Considerations for Selecting a MEAP

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>3.02</td>
</tr>
<tr>
<td>Integration with back-end data</td>
<td>3.28</td>
</tr>
<tr>
<td>Cost of development/deployment</td>
<td>3.76</td>
</tr>
<tr>
<td>Integration with infrastructure</td>
<td>4.07</td>
</tr>
<tr>
<td>Application maintenance and management</td>
<td>4.25</td>
</tr>
<tr>
<td>Device management</td>
<td>4.74</td>
</tr>
<tr>
<td>Need to support multiple devices</td>
<td>4.89</td>
</tr>
</tbody>
</table>

(Source: InsiderResearch Mobile Outlook 2012)

Survey participants were also asked to describe their current and future (three year) plans for mobile application development. Were they going to support a single mobile device or multiple devices? Were they going to buy off-the-shelf mobile applications or develop apps internally? Their answers, reported in the chart below, indicate that enterprises are planning to support multiple devices, and they are about evenly split between purchasing mobile solutions from vendors, and developing their own internally.

In summary, enterprises will need to support many mobile solutions, many device types, many mobile operating systems, and many custom-developed mobile solutions.
Development Strategies: Three Year View

All of the survey results and reports we have analyzed point to the need for enterprises to standardize quickly on mobile integrated development environments, mobile enterprise application platforms, and mobile device management solutions.

We recommend that enterprises:

1. Identify mobile application demand
2. Inventory the types of mobile solutions required to meet this demand
3. Document the anticipated ROI for each mobile solutions
4. Identify which mobile solutions are candidates for buying off-the-shelf, and which will likely need to be developed internally
5. Identify technologies required and their costs
6. Document estimated implementation efforts including design, development, and deployment efforts in time and costs
7. Differentiate short term and long term costs and identify the funding sources that will provide future maintenance and support
8. Prioritize mobile application projects
9. Identify budgets and resources
10. Seek ways to standardize on design, development, integration, deployment, and support
11. Seek ways to standardize on mobile platforms, IDEs, and mobile security solutions

Once these items have been completed, enterprises must immediately seek ways to leverage MEAPs, MDMs, and IDEs across as many of these mobile projects as possible with the goal of standardizing on platforms whenever possible. This approach will keep the mobile technology landscape as simple and uniform as possible, which will minimize the total cost of ownership and ensure a manageable support environment.
Analyst Contact

This report has been prepared by:
Kevin Benedict and André Guillemin

If you have questions arising from the content of this report or wish to explore the subject matter in greater detail, please contact:

Kevin.Benedict@netcentric-strategies.com
andre.guillemin@gmail.com