Harnessing Mobile Middleware

White Paper Developed by

December 19, 2007

TABLE OF CONTENTS

Figures

Tables

1. Introduction

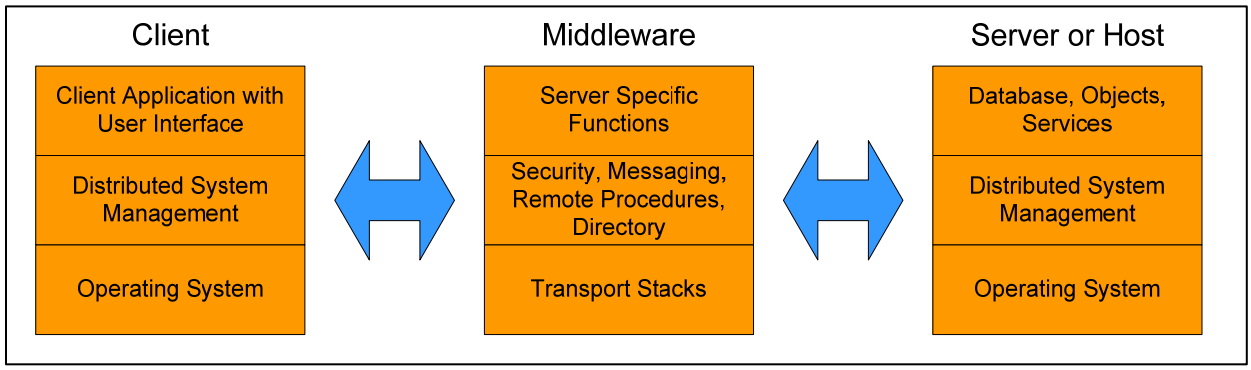
Mobile applications can significantly increase the productivity of workers, especially if the workers can access enterprise applications in a timely and efficient manner using their mobile wireless devices. Organizations have embraced wireless e-mail thanks to effective options that are available today for small form-factor devices. Now many organizations are working to make other data equally available. Mobile middleware platforms provide one means of achieving this objective, offering capabilities and functions that can simplify overall application deployment. Because there are many different types of mobile middleware, this paper seeks to explain how these platforms function, what their characteristics are, and how organizations can select the most appropriate solutions. In the appendix, the paper summarizes the capabilities of leading offerings.

This white paper targets IT developers, system architects, and managers looking at how best to develop and deploy mobile applications.

2. Middleware Background

Before embarking on a discussion of mobile middleware, it is useful to understand the general concept of middleware. The definition of middleware is computer software that connects software components or applications, enabling services to interact over a network. The purpose of middleware is to facilitate client/server operations, access to host applications, and to enable complex distributed systems, as shown in the following figure.

Figure 1: Middleware Facilitating Operations Over a Network



There are many types of middleware solutions in the market today based on a variety of technologies including, among others: message-oriented systems, object request brokers, database access, enterprise service buses, and remote procedure calls. Although the distinction between middleware, operating systems and applications is not always clear-cut, the role that middleware plays in today's heterogeneous computing environments is crucial. Similarly, mobile middleware plays an important role in extending enterprise computing to mobile platforms.

3. Mobile Applications and Middleware

Vendors have developed mobile middleware to address a variety of challenges in mobile application deployment. Mobile applications, especially when designed for handheld form factors, are not just ports of desktop applications. They must be able to deliver precisely the right information quickly and easily, and need to function in fundamentally different ways due to factors such as user interface constraints and wireless networks that function differently than wired networks.

Another challenge is that a mobile application may need to operate on multiple device operating systems and form factors as few organizations have standardized on just one device, users have device preferences, and it may be necessary to easily add new device types in the future. Given the different screen sizes and different underlying operating systems, supporting multiple device types flexibly can be challenging.

Companies regularly list security as one of their top concerns with mobile and wireless computing, so any mobile deployment needs to not only secure data communications, but also data on the device, and must properly authenticate users, as well as provide only authorized access to select information.

There is also the question of device management, which for larger deployments is essential. This includes items such as provisioning and updating software, and neutralizing lost or stolen devices. All of these functions can potentially be addressed by mobile middleware systems.

3.1. Evolution of Wireless Data Systems

Historically, wireless data usage has been driven by vertical-market applications tailored for very specific job functions. The first horizontal market application that saw large success was wireless e-mail. Today, however, users are transitioning from single-point solutions like e-mail to mobility being an integral part of their life and work and spanning multiple applications. IT is designing mobility into business practices, and computing trends such as service-oriented architectures that benefit server/application integration are extending to mobile systems. Meanwhile, an increasing number of mobile applications are becoming feasible due to:

- More powerful mobile platforms
- More capable wireless networks
- Greater enterprise experience with wireless technologies
- Better application integration platforms

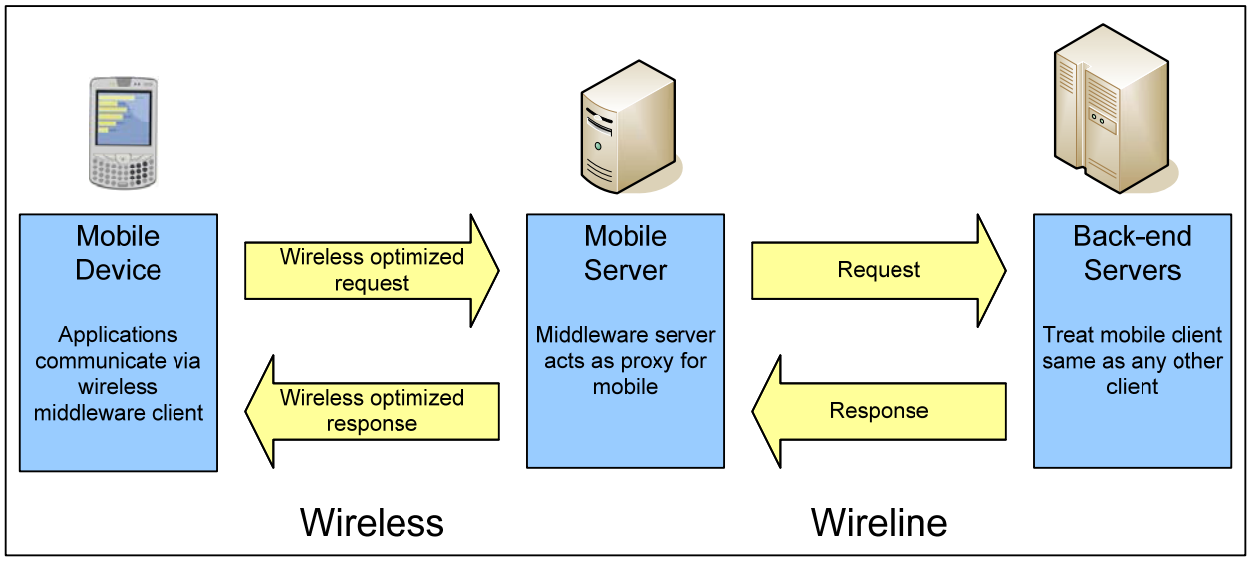
The result of all these trends is much more demanding application environments for mobile systems with multiple demands. It is this complexity that mobile middleware seeks to address.

3.2. Mobile Middleware Characteristics

There is no formal definition of mobile middleware, but a practical view is that it is software that operates between mobile systems and end services and applications. What vendors call their products can vary as well. Other names include wireless middleware, wireless infrastructure software, multichannel access gateways, and mobility platforms. Mobile VPNs and wireless e-mail gateways are also forms of mobile middleware. Actual functions and capabilities can vary tremendously across vendors.

Mobile middleware often includes client software that operates on the mobile device and server components. The mobile sever typically functions as a proxy on behalf of the mobile device, employing transport and application protocols that are optimized for wireless networks. Meanwhile, communication to back-end services employs standard application protocols designed for wireline connectivity.

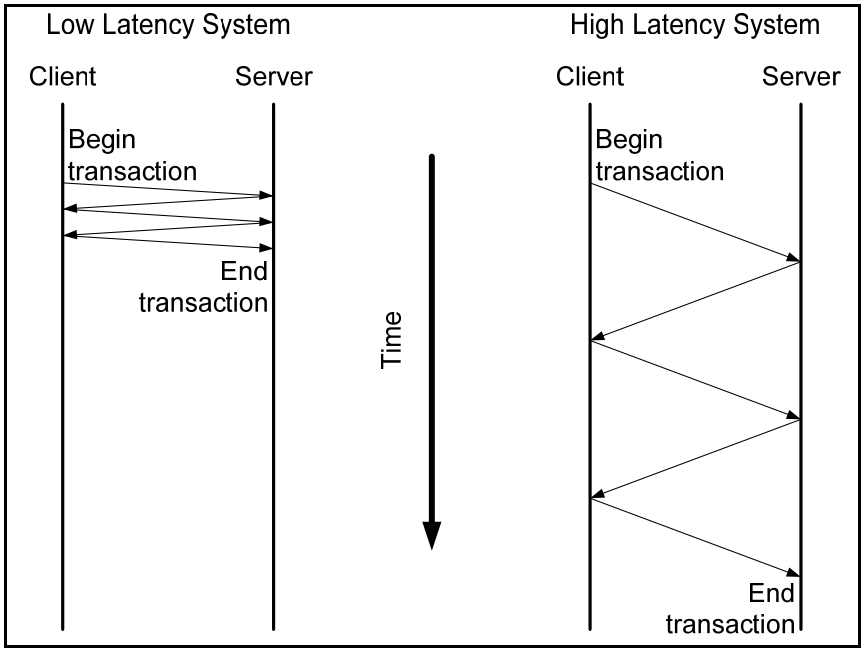
Figure 2: Simplified View of Mobile Middleware



Most middleware platforms support multiple types of mobile platforms, including mobile phones with browsers, smartphones and PDAs from a variety of vendors, as well as laptops. The middleware facilitates transport, presentation or application-level functions. The emphasis on different layers varies by vendors.

An example of the form of optimization that middleware systems can perform is to reduce the number of messages or other data that are sent to and from the mobile device during a mobile transaction, as well as to minimize the size of the data transfer. Since latency is typically higher in wireless networks than wireline networks, “chatty” applications can make the user experience quite sluggish. By reducing the number and size of actual messages communicated over the wireless connection, the middleware can significantly improve the user experience.

Figure 3: Managing Higher Latency



Not only do mobile middleware platforms improve communications efficiency, but they can enable a single development effort to support multiple mobile scenarios, including multiple types of wireless networks, as well as multiple types of devices.

3.3. Mobile Middleware Necessity

Many applications function well over wireless connections without middleware. They are aided by fast IP-based networks that have both high throughput and low latency, as well as by flat-rate pricing plans where application efficiency is not as much of a concern.

For some applications, especially ones not used that frequently, middleware may not be necessary. If a mobile client is available for the target device, IT managers should test the application to see how well it performs.

Typically, IT managers should consider middleware when no mobile client exists for the back-end application or when they need:

- Support for multiple device types with the same application
- Completely robust operation even with network variability
- Simultaneous access to multiple back-end applications
- Management of large numbers of devices and users
- Highly secure transactions and storage
- Close alignment of job functions and the mobile application process
- Highly intuitive operation optimized for the mobile device

Mobile middleware can provide significant benefits, but it also comes at a cost, including licensing fees, possible service fees and learning curves. IT managers must balance this cost against the benefits the middleware confers.

4. Mobile Middleware Categorization

One way of categorizing mobile middleware is as follows.

1. Wireless e-mail and synchronization products emphasize e-mail functions, along with the ability to perform synchronization tasks with databases. These products are simpler to configure than cross-platform mobile middleware, but do not offer the same degree of customization.
2. Cross-platform mobile middleware is designed to support a wide range of back-end host applications. This middleware typically offers flexible customization and development options, and generally supports a broad range of mobile devices.
3. Enterprise ISV mobility extensions are means of extending ISV applications to mobile devices. These products emphasize the vendor's own applications.
4. Mobile VPNs are designed to optimize and secure networking operations.

Table 1: Categories of Mobile Middleware

Category	Examples of Companies Providing Products in this Category
Wireless E-Mail and Synchronization	Motorola Good, Nokia Intellisync, RIM, Visto
Cross-Platform Mobile Middleware	Antenna Software, Dexterra, MobileAware, Sybase, Syclo, Vettro
Enterprise ISV Mobility Extension	IBM, Oracle, Salesforce.com, SAP
Mobile VPNs	IBM Lotus Mobile Connect, NetMotion Wireless

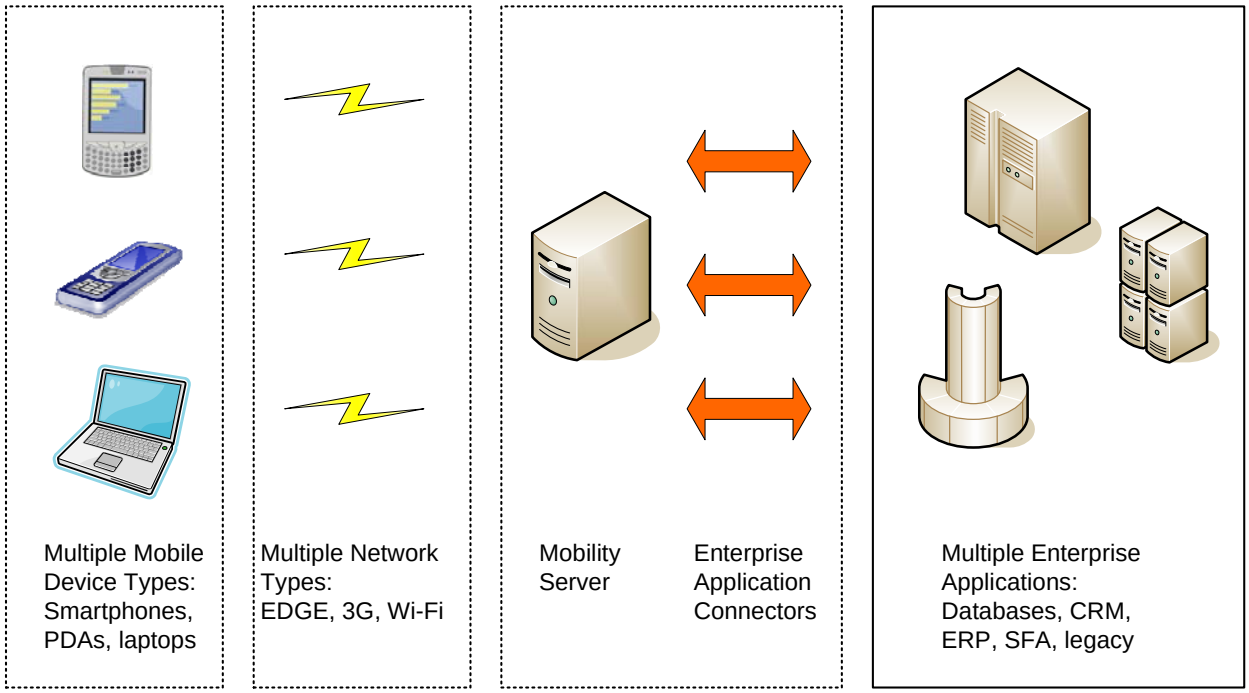
There are additional types of mobile applications solutions, some of which could be considered mobile middleware. These include thin-client mobile application servers; terminal servers (terminal emulation) and mobile device management systems. Note that this management functionality may also be included within the functionality provided by the above vendors.

This paper concentrates on cross-platform mobile middleware, as this category has the potential for mobilizing the greatest number of applications. However, it should be noted that enterprise ISV mobility extensions, while not typically as rich in cross-platform support or functionality, may provide a deeper level of data integration. For instance, if the mobile application needs to execute transactions that cannot be repudiated, this may be more readily achieved with the ISV's mobility extension than with a third-party solution. On the other hand, if the mobile application needs to access multiple back-end systems, then the cross-platform middleware may be best.

5. Typical Mobile Middleware Architecture

The following figure shows the architecture that is typically found in cross-platform mobile middleware.

Figure 4: Mobile Middleware Architecture

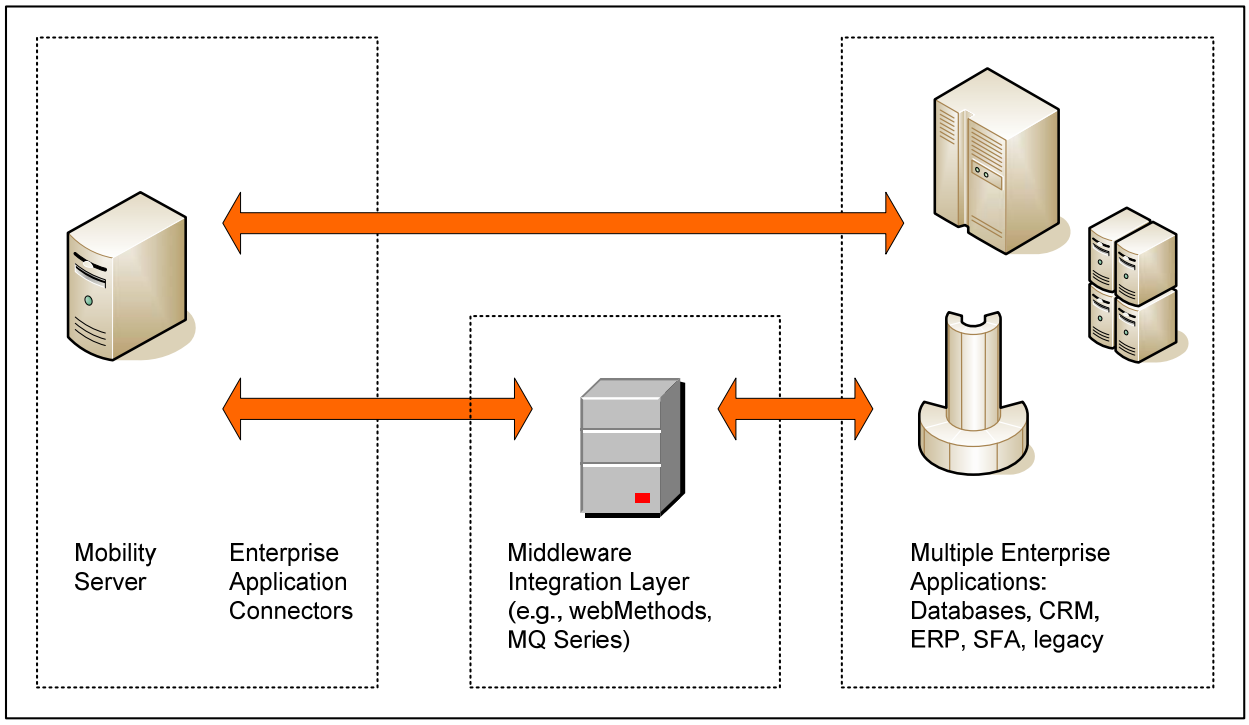


The salient points and elements of this architecture are as follows:

- **Multiple Mobile Device Types.** The middleware can support a range of mobile operating systems and form factors.
- **Multiple Network Types.** The middleware can operate over multiple types of mobile networks, such as 2G, 2.5G and 3G wide-area networks and/or Wi-Fi local area networks.
- **Mobility Server.** There is typically a server component that resides in the enterprise behind the firewall, or alternatively, is hosted by the middleware vendor in a software-as-a-service model.
- **Enterprise Application Integration.** This is increasingly accomplished using standardized methods, but nevertheless can involve custom programming.

In some cases, enterprises may have already deployed a middleware integration layer independent of the mobile middleware, as shown in the following figure.

Figure 5: Middleware Integration Layer



In this case, the mobile middleware must interface with the middleware integration layer.

Examples of integration vendors include:

- IBM WebSphere MQ Series
- webMethods
- BEA WebLogic
- Tibco
- SeeBeyond

Some of these integration middleware vendors themselves may offer mobility functions that developers can leverage.

6. Mobile Middleware Variables

There are considerable variations in the breadth and depth of functions implemented in mobile middleware products. The following table summarizes some of the most important ones.

Table 2: Middleware Variables

Variable	Possibilities
Types of Functions	Functions that facilitate the transport of enterprise application data Network communication optimization Security Device management
Programmability	Pre-defined functions APIs and scripting for customized applications Extensible application templates
Mobile Platforms Supported	Most commonly supported: • Phone browser (note that there are multiple types of phone browsers available) • Java Platform Micro Edition • RIM Blackberry • Symbian • Palm Garnet OS • Windows Mobile • Windows (for notebook platforms)
Location of Middleware Server	Behind the firewall at the enterprise Software as a service (hosted)
Mobile Application Model	Always connected (e.g., browser) Offline capability (i.e., local client able to work in connected or disconnected mode)
Applications Supported	Single back-end applications Composite/hybrid applications
Server Integration Protocols	Multiple available protocols for server integration, e.g., Web services, COM objects, etc. (Standards-based approaches such as Web services facilitate integration.)

Variable	Possibilities
Vendors Supported	Mobility platform and end application from same vendor Platform supports multiple enterprise application vendors

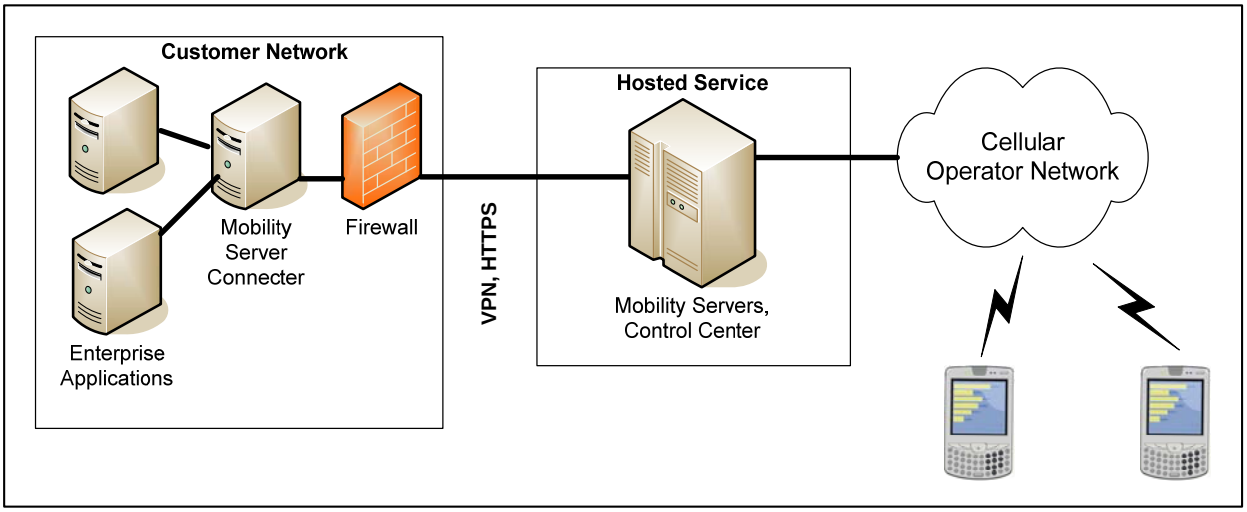
IT managers need to assess which variables are important to them. Based on the vendor information in Appendix A, they can determine which mobile middleware platforms are most suitable in addressing their needs.

The following sections elaborate on some of the variables presented in this table.

7. Hosted Mobile Middleware Model

One option that some mobile middleware vendors provide is to host the middleware servers in what people sometimes call software as a service. See the following figure.

Figure 6: Hosted Mobile Middleware Model



The hosted mobility server communicates with mobile devices in one direction and with enterprise systems over secure connections using transports such as VPNs or HTTPS in the other direction. The middleware server may communicate directly with enterprise applications on the customer's network, or via an additional component labeled the mobility server gateway in Figure 6. The middleware vendor typically offers means for customers to configure and manage their applications.

This approach may be of interest to organizations that want to minimize the amount of equipment they need to install and maintain. It also simplifies firewall considerations at the enterprise since only a connection to the hosted service is required, as opposed to multiple connections to multiple mobile devices. However, organizations that want to manage their

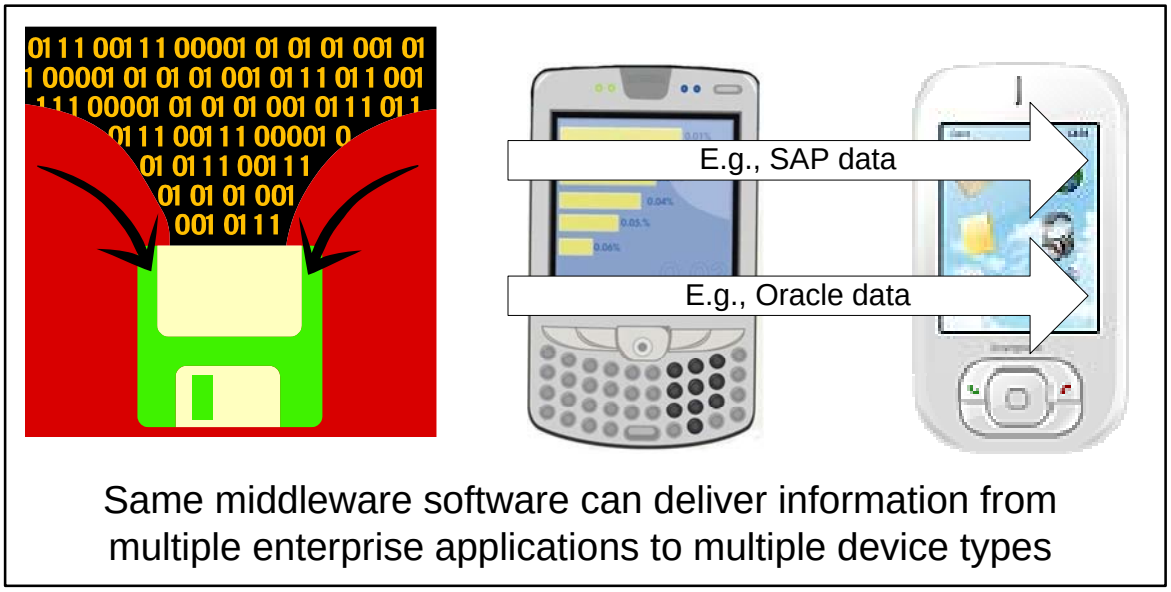
own infrastructure, or maintain full control over their own data may prefer to host their own solution.

8. Cross-Platform Middleware Solutions

There are a number of situations where a cross-platform middleware solution will be of interest. They are as follows:

1. **Device Flexibility.** Cross-platform mobile middleware products can support a large number of mobile device platforms. Often, they are structured so the mobile application will function on new devices and platforms as they become available. Vendors can achieve this through approaches such as mobile scripting engines.
2. **Enterprise Application Flexibility.** Cross-platform offerings are often flexible in how many enterprise applications and data stores they can access. In many cases, the same mobile application can access data from multiple back-end applications.

Figure 7: Flexibility of Cross-Platform Middleware



Note that some wireless e-mail gateways and synchronization products are also quite flexible with respect to the number of mobile devices that they support.

9. Network Level Functions

While many new middleware products emphasize application-level functions, some also provide network and transport layer functionality that can improve performance in wireless environments.

Though higher-performing 3G wireless technologies lessen the need for network optimization, either the application or middleware must address the following considerations and approaches:

- **Network Speed.** Depending upon network speed in a given coverage area, optimized network communications can benefit users with faster response time.
- **Compression.** Mobile middleware can compress data to improve network performance. The benefit of the compression depends on the type of data, but can significantly benefit some applications.
- **UDP-Based Transport Protocols.** Since TCP protocols are not fully optimized for wireless communications, the middleware may use UDP instead, and implement wireless-optimized transport-layer protocols.
- **Dropped Connections.** When mobile, devices can sometimes lose their connection. Middleware can queue messages and send information when connections are available. If done in the background, users do not necessarily even need to know when they are in or out of coverage.
- **IP Address Changes.** When reconnecting to the network, perhaps after a coverage gap, the IP address assigned to the device may be different than during the previous session. Middleware can isolate the application from these changes by assigning virtual IP addresses so that applications experience a consistent connection.
- **Mobility Across Networks.** Some middleware can allow seamless roaming across multiple network types. For example, the same application could continue operating even as the device connection changes from cellular to Wi-Fi.
- **Unwanted Applications Accessing the Network.** Some middleware solutions can implement policies that control which applications have access to network connections under what circumstances, and can even prioritize network access. For example, with a cellular connection, the middleware might block operating system updates, but allow them with Wi-Fi.

Mobile VPNs are instances of mobile middleware that implement many of the features discussed in this section. However, other types of mobile middleware, depending on vendor, may also implement some or all of the features listed above.

10. Other Types of Middleware

This section briefly discusses two specific types of mobile middleware: mobile VPNs and wireless e-mail gateways.

10.1. Mobile VPNs

Mobile VPNs are mobile middleware products that emphasize network and transport optimization and security. Examples are NetMotion Wireless and IBM Lotus Mobile.

Some of the specific functions they offer include:

- **Network Optimization.** Mobile VPNs implement many of the features discussed in the previous section, “Network Level Functions,” including optimized transport protocols, protection against dropped connections, managing IP address changes, mobility across network and policy management.

- **End-to-end Security.** Mobile VPNs encrypt communication between the mobile VPN client and the mobile VPN server using algorithms, such as the Advanced Encryption Standard (AES). They also typically implement multi-factor and mutual authentication methods.

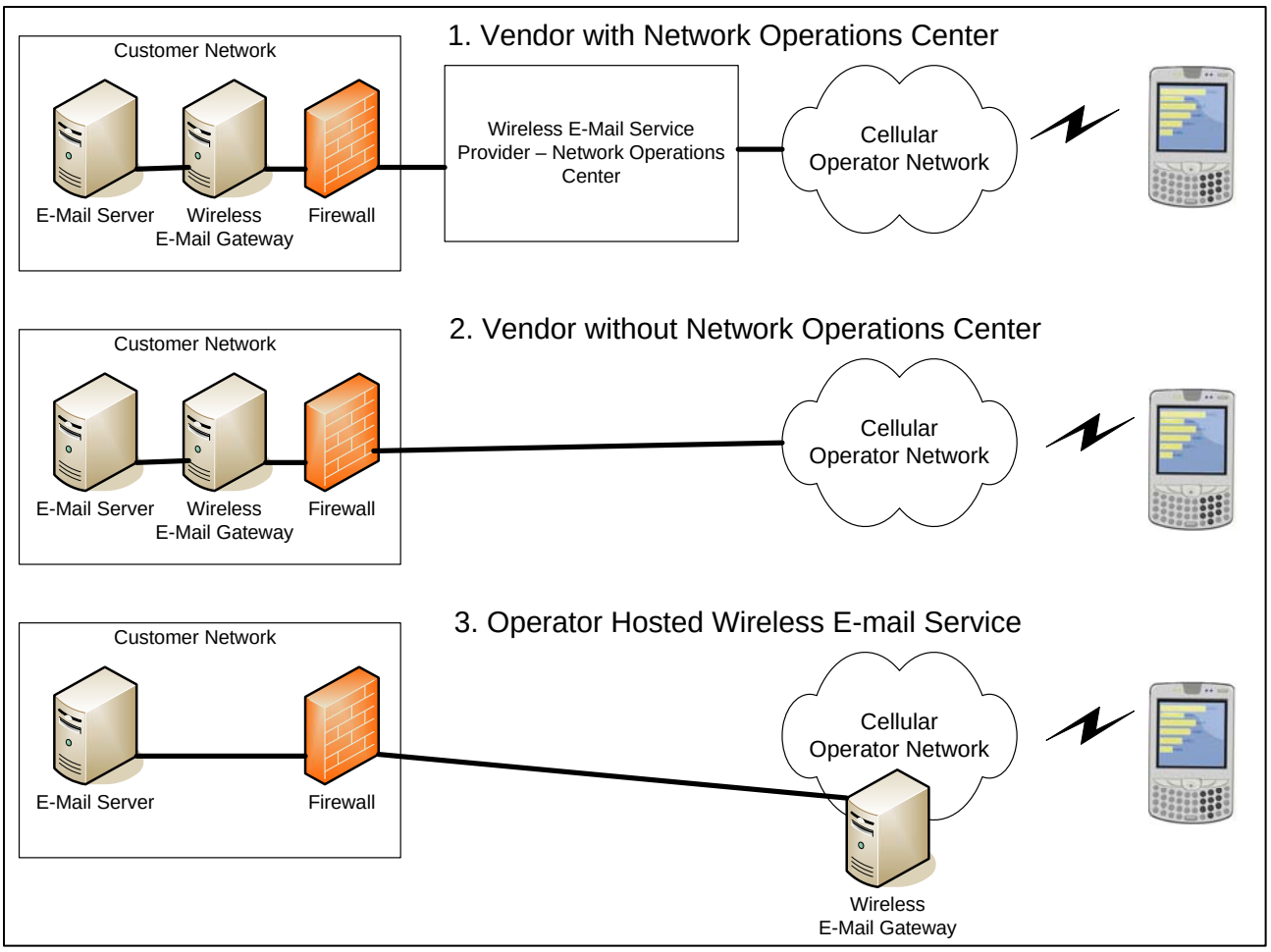
If an enterprise already has a mobile application that performs well over wireless connections, but needs to bolster security or networking robustness, then a mobile VPN can be a good solution.

10.2. Wireless E-Mail

A highly popular mobile middleware category is that of wireless e-mail, which typically consists of an e-mail client and a wireless e-mail gateway. The gateway (corresponding to the mobile middleware server in the discussion above) acts as a proxy for the mobile client, delivering e-mail with optimizations that include push in which the gateway sends new messages automatically to the device; efficient attachment viewing and handling; efficient network utilization; and synchronization of other data, such as contacts and appointments.

There are a number of models for deploying wireless e-mail, as shown in the following figure.

Figure 8: Wireless E-Mail Middleware



In scenario 1, the wireless e-mail gateway resides behind the firewall at the enterprise and channels e-mail via a network operations center or "NOC." RIM and Motorola Good fall into this category.

In scenario 2, the wireless e-mail gateway behind the firewall communicates directly with wireless devices. Nokia Intellisync in an example of this type.

In scenario 3, the wireless carrier hosts the gateway. Visto is an example from this category.

Note that how well the middleware vendors integrate with email and PIM is a key consideration when choosing a vendor – especially for CRM.

11. Mobile Middleware Clients

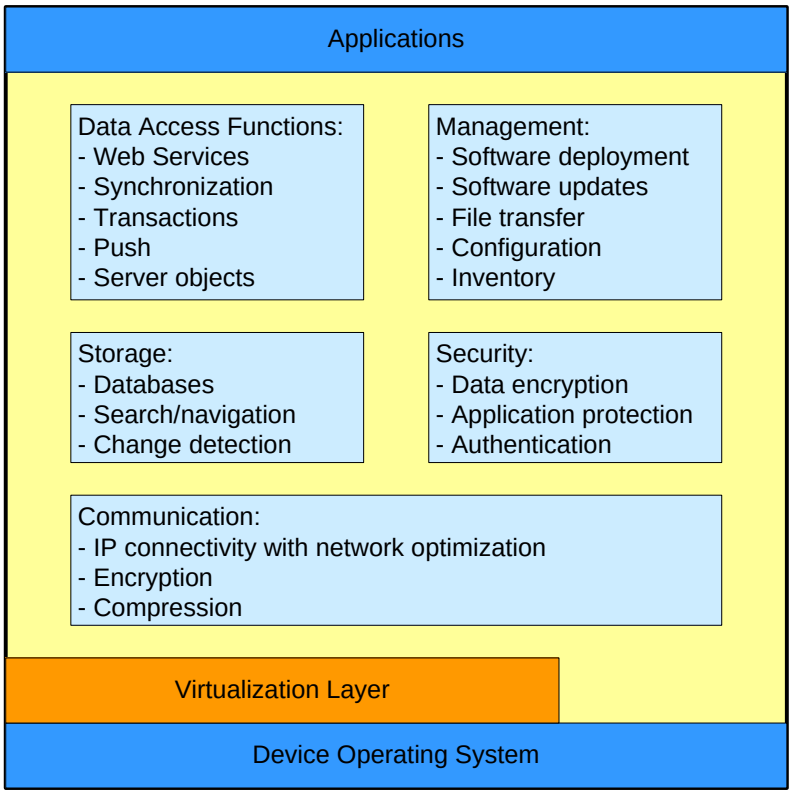
There are a number of ways that mobile middleware systems deploy the client on the mobile device. In some cases, the client provides dedicated functions, such as in some wireless e-mail or mobile VPN solutions. However, others offer programmable clients, which allow implementation of interactive user interfaces, customer-specific business rules, validations and event-based interactivity. Other application functions might include sending and receiving documents, and capturing customer signatures.

For programmable solutions, the middleware typically provides a virtual environment with consistent APIs, regardless of the target device. This client may be designed as a Java client or as a native client for the particular platform (e.g., on Symbian or Windows Mobile). Programming the client may consist of scripts developed in an integrated development environment, making programming significantly simpler than in the native platform environment. However, developers need to realize that the simpler environment can limit functionality as compared to a native application. For instance, the application will not have as direct a control over the hardware environment.

Some middleware solutions are based on a browser approach, which offers some advantages in readily supporting a large number of mobile devices and simplifying mobile code management, but will have a slower user interface and will usually require network connectivity for operations. However, some products (e.g., RIM BlackBerry) do allow for off-line forms completion even in a browser mode.

The following figure shows a typical mobile middleware client implementation.

Figure 9: Mobile Middleware Client Architecture



12. Middleware Integrated Development Environments

Initial middleware development environments were proprietary and limited to a small number of device types. Over time, device support improved, and many systems today emphasize open-source technologies, such as Eclipse Plug-Ins and WSDL introspection.

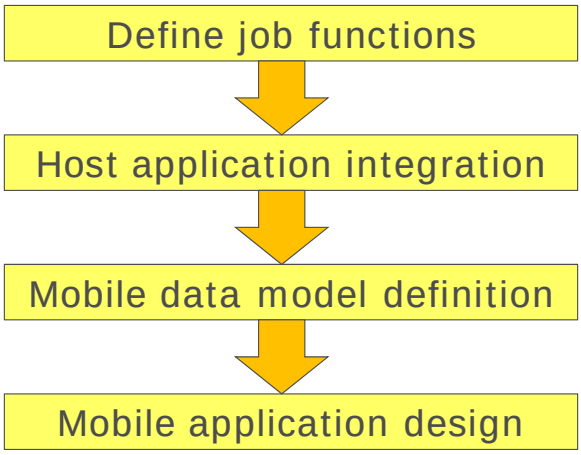
Rather than needing to write actual software, mobile middleware environments can provide a modeling environment in which developers can create the application through an interactive process of picking the elements of the user interface and the objects they will manipulate. These middleware tools then use the application model to create the software that runs on the actual device. The IDEs can include local emulation of mobile devices so developers can see exactly how the application will run on the mobile device without having to actually install the software on the device.

Some of the innovations of current IDEs include the ability to embed high-level objects, such as synchronization, communication and remote object access. Synchronization refers to automatic mobile-to-server data synchronization that can simplify off-line versus connected operation. Web services and remote methods allow services to be wrapped and exposed as local modules for mobile application environments, such as .NET Compact Framework and Java environments. This facilitates the delivery of server-side information to mobile devices.

Other application functions include access to local environment data stores; e-mail and phone integration in which the mobile application can make calls, send e-mails and log such activity; peripheral support for printers, credit-card readers and barcode scanners; and signature capture.

The typical steps involved in developing a mobile application using today's mobile middleware platforms are shown in the following figure.

Figure 10: Typical Mobile Application Development Process.



The following sections elaborate on the items shown in this figure.

13. Host Application Integration

On the host application side, the middleware must interconnect with end services or host applications. Here, service-oriented architecture provides a significant benefit as it facilitates a loose coupling, but tight integration of composite applications. The most common implementation of service-oriented architecture is through Web services, based on standards such as XML, SOAP (Simple Object Access Protocol) and REST (Representational State Transfer.)

Alternative interfaces include Java Messaging Service (JMS) and Internet Inter-ORB Protocol (IIOP), as well as ActiveX Data Objects (ADO) and native support for server-side integration vendors such as IBM, webMethods, BEA, Tibco and SeeBeyond. Developers are likely to find standards-based approaches to be the best supported and most effective.

Developers may need to work with the host application to enable the mobile application model. The following is a typical process if using a Web services approach.

1. Decide on objects to be made available (contacts, sales opportunities, etc.)
2. Pick fields for the objects (name, phone number, etc.)
3. Generate integration objects, which are combinations of fields that allow creation of XML data structures
4. Define allowed operations (insert, get, update, etc.)
5. Finalize data structures and operations to allow definition of Web Services

6. Generate WSDL (Web Services Definition Language)
7. Import WSDL into mobile middleware application environment

14. Mobile Application Design

The next step is the design of the mobile application. Developing an application using mobile middleware requires the creation of a mobile application model based on the enterprise data model. The vendor may provide some predefined applications or templates that the developer can leverage. Only in the most simplistic cases would these precisely address a customer's needs. Modern tools emphasize a model-driven development process that reduces complexity, provides user interface controls, platform independence and comprehensive tools.

The typical steps involved in developing the application include:

1. Importing the application specification (e.g., WSDL) (see previous section)
2. Designing the mobile application model (objects, related objects, object transactions, etc.)
3. Designing the mobile application details (user interfaces, lists, fields, searches, etc.)
4. Having the IDE automatically generate the application
5. Testing the application with simulators
6. Publishing the application for further testing on mobile devices
7. Refining as needed

Developers should:

1. Consider carefully exactly what data the user needs and how he or she will interact with it. This requires a very intimate understanding of the job function, and the realization that how the job is performed may actually change due to the mobile technology.
2. Realize that a single mobile-middleware code base may support multiple devices, but each device type will have to be tested and adjustments may be necessary.
3. Understand the specification of the mobile device operating system such as memory and computational power. In addition, security policies may prevent the use of external cards for storage, thus limiting how much local data is available for the application.
4. Understand how local data stores on mobile devices operate. For example, developers may need to know whether local databases can be encrypted and whether the middleware provides access to the encryption functions.
5. Consider any compatibility issues that may exist when deploying multiple applications on a single mobile device.

15. Security Considerations

In deploying mobile applications, developers and IT managers must carefully implement security policies including application security, device security and, if using a hosted service, hosted server policies.

15.1. Application and Device Security

Many middleware solutions provide security functions that can assist in security both for the device and the application. Common security functions include protection of data on the mobile device (e.g., encryption), protection of data at servers and control centers, encryption of data communications between devices, servers and control centers, user authentication, and neutralization of lost or stolen devices. Less common functions are firewalls and antivirus.

For some deployments, the security functions that the middleware provides will be sufficient. For others, developers may choose to implement separate third-party security and/or management systems.

15.2. Hosted Server Policies

If using a hosted middleware server system, developers will need to evaluate the extent of protection at the data center. Ideally, the vendor stores each customer's data in a separate and distinct database. Other recommended data protection methods include replication of data for high availability, regular backups of data, off-site storage, and servers monitored for reliable and healthy operation.

Other important security aspects for hosted servers include policies for items such as passwords and telnet access, physical security, power management and backup, and network redundancy. Developers may also look to see whether the site has Systrust certification as set by the American Institute of Certified Public Accountants (AICPA).

15.3. Firewall Considerations

Developers and integrators will need to examine the connections between the enterprise and the mobile devices. If using a hosted service, this will consist of a connection to the hosted service. If using a behind-the-firewall approach, this will require multiple mobile devices being able to access the mobile server.

16. Management

Enterprises may need to manage hundreds if not thousands of mobile devices. A mobile management system can greatly facilitate tracking and maintaining these devices. Many middleware solutions, though not all, provide management capabilities. Developers need to determine whether to use the vendor-provided capabilities, whether to employ mobile device capabilities of existing internal network management systems if available, or whether to deploy a third-party system.

Typical device management functions include:

- Over-the-Air (OTA) Provisioning
- OTA application or features downloads and updates

- Provisioning, installation and updates via wired sync
- Mobile system policy control, such as allowed applications and required software versions
- Software and hardware inventory management
- Backup and restore
- Power-on password enforcement
- Server management consoles for overall system functions.

17. Conclusion

In conclusion, mobile middleware can provide significant benefits in the deployment of mobile applications. It can address the variability of diverse mobile devices, and it can provide a consistent programming environment across these with high level modeling approaches. This paper has focused on cross-platform mobile middleware solutions that provide the greatest programming flexibility. However, there are also other types of systems such as mobile VPNs, wireless e-mail and synchronization gateways, and enterprise application mobility extensions that organizations also need to consider.

Organizations also need to consider the scenarios wherein mobile middleware provides the greatest benefit. For example, many networking applications work very well on laptops over 3G connections without the need for middleware. On the other hand, an application for a handheld device used for sales force automation that needs to simultaneously access multiple back-end databases may be much easier to implement with mobile middleware. How much actual programming and development the middleware requires depends on the type of middleware and the functions it implements. Beyond application-level functions, many middleware systems also provide rich security and management features.

18. Table of Acronyms

The following table lists the acronyms used in this paper.

Table 3: Table of Acronyms

Term or Acronym	Definition
2G	Second Generation
3G	Third Generation
3GPP	Third Generation Partnership Project
ADO	ActiveX Data Objects
AES	Advanced Encryption Standard
AICPA	American Institute of Certified Public Accountants
CRM	Customer Relationship Management
ERM	Enterprise Resource Management
FFA	Field Force Automation
HR	Human Resources
IIOP	Inter-ORB Protocol
JMS	Java Messaging Service
OTA	Over the Air
PIM	Personal Information Manager
REST	Representational State Transfer
SFA	Sales Force Automation
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
WSDL	Web Services Definition Language

19. Vendor Summary Information

The following includes summary information about mobile middleware products received from different vendors. More detailed information on these products is available at AT&T devCentral at http://developer.att.com/middleware. This information is compiled from responses to an RFI conducted by Rysavy Research and AT&T in 2007. Readers interested in any particular solution should obtain up-to-date information directly from the vendor.

Table 4: Selected Vendor Products

Vendor	Product	Type
Antenna Software	Antenna Mobility Platform	Cross-platform
Dexterra	Concert	Cross-platform
MobileAware	Express Q, ExpressSync, Smart IP, Mobile Interaction Server	Cross-platform
Nokia Intellisync	Nokia Intellisync Mobile Suite	E-mail and synchronization
Oracle	Oracle Database Lite	Enterprise ISV mobility extension*
SAP	SAP Netweaver Mobile	Enterprise ISV mobility extension
Sybase	Information Anywhere Suite	Cross-platform
Syclo	Agentry	Cross-platform
Vettro	Vettro 360	Cross-platform

* Note: Oracle response to RFI was limited to Oracle database mobility extension.

Table 5: Vendor Application Emphasis

Vendor	FSA/FFA	SFA/CRM	ERP/ERM	Other and Comments
Antenna Software	X	X	X	Inventory management, scheduling and dispatch, knowledge and content management, time reporting, IT service management and support desk applications, integrated voice response (IVR) applications, merchandising, custom and composite applications
Dexterra	X	X	X	Facilities management, asset management, inspections, data collection, merchandising
MobileAware	X	X		
Nokia Intellisync	X	X	X	Wireless Email
Oracle Database Lite	X	X		Machine-to-machine, fleet logistics and telematics
SAP Netweaver Mobile	X	X	X	HR, logistics
Sybase	X	X		Enterprise application mobilization, custom application mobilization, wireless Email & PIM, enterprise messaging, Web-content mobilization
Syclo	X	X	X	Inventory management, auditing, inspections, rounds, scheduling, task tracking, asset management, sales/CRM, delivery management, facilities maintenance, fleet management, safety and security measurement, IT asset management
Vettro	X	X		IT service management, pickup & delivery logistics, facilities management, custom solutions

Table 6: Application Vendor Support

The following table lists which applications are supported by the mobile middleware, as cited by each of the vendors.

Vendor	Applications/Vendors Cited
Antenna Software	Amdocs Clarify, BMC Remedy, NetSuite, SAP R/3, Siebel CRM, Siebel CRM On Demand, Oracle E-Business, Oracle PeopleSoft CRM, Oracle JD Edwards, Servigistics, MRO Maximo, ATG Knowledge (Primus), TeleNav, Astea, QAD MFG/PRO, Navision, custom built systems
Dexterra	SAP, Siebel, Maximo, Salesforce.com, JD Edwards, Oracle, Remedy, Clarify, FAMIS, Mincom
MobileAware	Microsoft SQL Server, Oracle databases
Nokia Intellisync	Microsoft, IBM, Oracle, Sybase
Oracle Database Lite	Oracle, including Oracle's Enterprise Business Suite's Mobile Field Force Automation application
SAP Netweaver Mobile	SAP
Sybase	SFDC, SAP, Remedy, Siebel, Saratoga, Business Objects, Siebel, Microsoft Exchange, IBM Lotus Domino
Syclo	BMC Remedy, Crossform, Documentum, Fourth Shift, Indus IAS, PassPort, InSite, EMPAC, INFOR, Datastream, JD Edwards Enterprise Software, Mainsaver, Maximo, Maximus, Micromain, Sentact, Oracle, Peoplesoft, Primavera, SAP, Shaker Software, COINS, Siebel, Tririga 7x, 8i, Momentum
Vettro	Oracle/Seibel, Informix, Salesforce.com, HP, BMC Software, HEAT, Amdocs/Clarify, Tabware, Datastream, MFGPro, Solomon, Odessey

Table 7: Host Integration

Vendor	Host Integration
Antenna Software	Web Services, XML/HTTP, SOAP, JMS, RMI, text files, XML file drop, XML over HTTP. Integration with message bus systems such as webMethods, Vitria and WebSphere MQ.
Dexterra	SOA, XML Web Services, Java, .NET
Mobile Aware	Proprietary SDK. ExpressQ provides generic connectors for JMS and Email. Web Services being developed. ExpressSync provides database sync against Microsoft SQL Server and Oracle databases.
Nokia Intellisync	XML connectors; database connectors for Microsoft, IBM, Oracle, Sybase; vendor APIs
Oracle Database Lite	JDBC connection pool
SAP Netweaver Mobile	BAPI/RFC and Web Services
Sybase	Native data access to Oracle, Microsoft SQL Server, IBM DB2, SQL Anywhere, and Sybase Adaptive Server Enterprise, through ADO.NET, OLE DB, ODBC 3.5/level 2, JDBC 3.0, Embedded SQL. Synchronizing with Application Servers, ERP systems such as SAP etc, using Connectors, Web services, XML files, SOA-based back-end access, or other third party relational databases. Email Systems (Microsoft Exchange and Lotus Domino).
Syclo	Connectors based on SQL, Java, XML, ODBC, integration bus.
Vettro	Web services, stored procedures, JDBC/ODBC, JCA, JMS, flat file, SOAP, XML

Table 8: Mobile Client Approach

Vendor	Browser	Java Micro	Native Client	Scripting Client
Antenna Software		X	X	X
Dexterra		X	X	
MobileAware Mobile Interaction Server	X			
MobileAware Express Q		X	X	X
MobileAware ExpressSync	X			X
MobileAware Smart IP	X			
Nokia Intellisync		X	X	
Oracle Database Lite		X	X	X
Oracle Application Server	X			
SAP Netweaver Mobile	X	X		
Sybase	X	X	X	X
Syclo	X		X	
Vettro		X	X	

Table 9: Device Support

Vendor	Palm Garnet OS	RIM BlackBerry	Windows Mobile	Symbian	Mobile Linux	Windows Notebook	Comments
Antenna Software	X	X	X			X	Also XP Tablet, voice-based Devices
Dexterra		X	X	X		X	Also XP Tablet
MobileAware Interaction Server	X	X	X	X	X	X	All browser
MobileAware Express Q		X	X			X	
MobileAware ExpressSync			X			X	
MobileAware Smart IP			X			X	
Nokia Intellisync	X	X	X	X		X	
Oracle Database Lite			X	X	X	X	
SAP Netweaver Mobile		X	X	X	X	X	BlackBerry (browser, thick client 2008), Symbian (thin client)
Sybase	X	X	X	X	X	X	Also XP Tablet
Syclo		X	X			X	Blackberry by browser
Vettro		X	X				Also Java phones

Table 10: Hosting Capabilities

Vendor	Hosting Capabilities
Antenna Software	Yes (single and multi-tenant)
Dexterra	Yes
MobileAware	No
Nokia Intellisync	Yes
Oracle Database Lite	Yes
SAP Netweaver Mobile	No
Sybase	Yes (by some carriers and SI partners)
Syclo	Yes
Vettro	Yes

Table 11: Vendor Developer Environments

Vendor	Developer Environment
Antenna Software	AMP™ Studio is an Eclipse plug-in, model-driven and component based development environment that leverages SOA oriented infrastructure. AMP™ Studio enables development of one application that can run simultaneously across BlackBerry, Windows Mobile, and Palm OS devices. Provides integrated testing and publishing. Portfolio of Enterprise System Adapters and Baseline SmartClient Applications.
Dexterra	Visual Studio 2005 and Eclipse with Dexterra Composer.
MobileAware	ExpressQ: SDK for C++, .NET (C#/VB) and Java. Visual Studio and most Java IDEs (such as JBuilder, Eclipse, IntelliJ, NetBeans), as well as SOA/BPM tools that support JMS and Web Services. ExpressSync: SDK for database integration and drag-and-drop IDE for database-oriented mobile applications. Smart IP provides optional SDK to optimize the performance of socket-based networking applications. Mobile Interaction Sever: Plug-ins for Eclipse, DreamWeaver, BEA WebLogic Workshop/Studio.
Nokia Intellisync	Web-based application for development for use with Mobile Suite Application Sync module.
Oracle Database Lite	Most IDEs including .Net Visual Studio, Jbuilder, Jdeveloper, Eclipse, IntelliJ or any that support ODBC, JDBC and ADO.Net.
SAP Netweaver Mobile	Eclipse IDE, model-driven development.
Sybase	Microsoft Visual Studio, Eclipse, Sybase PowerBuilder, others. Languages include C/C++/C#, VB.NET, Java, Perl, PHP, and others. Application models supported include synchronized client/server, mobile messaging, mobile offline Web, mobile SOA (including Web Services), and on and offline thin-client models.

Vendor	Developer Environment
Syclo	Agentry editor employs service-oriented development. Object model used to put applications together using a library of building blocks that includes connectors, peripheral controls, transaction templates, synchronization protocols, screen sets and other elements.
Vettro	Generally Eclipse or Visual Studio. Vettro V360 client application descriptions run within a Vettro Mobile Virtual Machine (VMVM) and are comprised of Internet standard development technologies including XML, xPath, xQuery and scripts.

Table 12: Security Functions

Vendor	Local Encryption	Communications Encryption	User Authentication	Remote Data Wiping	Remote Device Kill	Comments
Antenna Software	X	X	X	X	X	SysTrust compliant operations platform (On-Demand and On-Premise deployments)
Dexterra	X	X	X	X	X	Local encryption, firewall and antivirus by BlueFire
MobileAware Express Q, Express Sync, Smart IP		X	X			
Nokia Intellisync	X		X	X	X	Local encryption via partner products, enforcement of security policies on device, antivirus through partner products
Oracle Database Lite	X	X	X	X	X	
SAP Netweaver Mobile	X	X	X	X		
Sybase	X	X	X	X	X	
Syclo	X	X	X	X	X	Local encryption using Microsoft Enhanced Cryptographic Provider on device
Vettro	X	X	X	X	X	Uses RIM's and Microsoft's device encryption services

Table 13: Management Functions

Note that some functions, such as OTA provisioning, when not available from the middleware vendor itself, may be supplied by third parties.

Vendor	OTA Provisioning	OTA Updates	Wire Sync	Mobile System Policy Control	Comments
Antenna Software	X	X	X	X	Web-based portal for the management of users, devices and OTA deployments NOC-grade monitoring & troubleshooting tools Real time usage & activity reports tracking users, devices & apps
Dexterra	X	X	X	X	
MobileAware				X	Support for remote configuration.
Nokia Intellisync	X	X	X	X	Remote control of Windows Mobile devices for helpdesk troubleshooting
Oracle Database Lite	X	X	X	X	
SAP Netweaver Mobile	X	X	X	X	
Sybase	X	X	X	X	
Syclo		X	X	X	
Vettro	X	X	X		