Oracle9*i*AS Wireless: Creating a Mobilized Business

An Oracle White Paper
March 2002

ORACLE

EXECUTIVE OVERVIEW

The Internet has transformed the way we conduct business by changing the way we communicate with each other, changing how we buy and sell products, and changing how we attract and retain customers. The Internet has made businesses more efficient by expanding the markets in which they compete, streamlining corporate processes, and extending the ways customers are reached.

Up until now, traditional applications have not accounted for the fact that employees leave their desks. The time spent walking to and from meetings or traveling to see customers is wasted time without access to key information. Effective mobile computing aims to solve this problem by delivering the information users want – on the device of their choice – thereby making them more productive. Like enterprises embraced the Internet to improve business processes, gain competitive advantages and reach new customers – mobile technology further extends this opportunity. For enterprises looking to cut operational costs and drive new revenue, mobile technology offers untapped potential. One independent market research company, iGillottResearch, created benchmarks for benefits associated with some key mobile applications:

- Mobile e-mail and PIM saves 5-6 hours per mobile employee per week
- Mobile CRM – 10% average reduction in calls to customer service center
- Mobile Field Force – 15% average reduction in cost per service call
- Mobile Sales Force – 15-20% increase in sales revenue
- Average payback period for mobile applications is 4-6 months

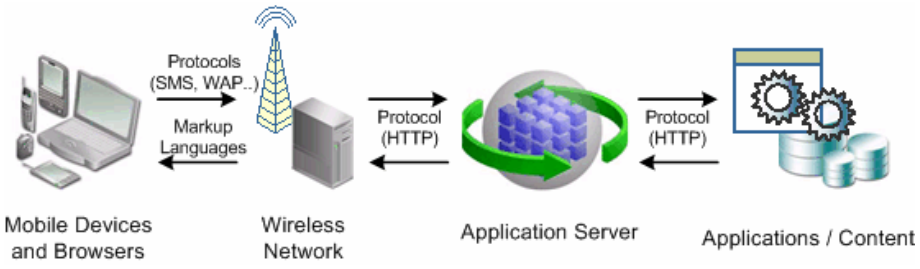
Oracle identified the potential of mobile technology early on and has been developing mobile products since 1995. The centerpiece of Oracle's mobile platform is Oracle9*i*AS Wireless, an integrated set of features of the Oracle9*i* Application Server, which provides the most flexible, scalable and reliable mobile infrastructure on the market. It offers true multi-channel access to any application with virtually any device, including voice, and is the fastest, most cost-efficient way to start streamlining mobile business processes.

WIRELESS OVERVIEW

Developing mobile applications can be challenging for three important reasons. First, wireless devices typically have small screens and limited data entry capability making them best suited to carry out specific inquiries and transactions rather than for complicated processes. Additionally, while wireless standards are emerging, there is still tremendous heterogeneity in the standards supported by wireless devices. Finally, devices communicate with different wireless protocols and support a variety of different markup languages – these different standards preclude a developer from writing every application to individually support all the available devices. As a result, as companies plan to exploit mobile application opportunities, they need to choose a software platform that will radically simplify how they develop, deploy and manage mobile applications and portals.

Introduction to Wireless Infrastructure

There are many infrastructure components that interoperate in the wireless value chain solution. The components of a typical wireless network are described below.



Wireless infrastructure Components

Mobile Devices and Browsers

- Wireless devices and browsers are used to access the mobile Internet. Each wireless device generally runs a browser to display received information. This is analogous in the fixed Internet world to a personal computer running a standard Internet browser. Wireless devices also include in-car systems and traditional phones using voice technology.
- Wireless markup language is the language that a browser speaks - the markup language specifies how information should be presented on the device. Common Markup Languages include VoiceXML, WML, and HDML. These standards are always evolving.

Wireless Network

- Networks are the underlying infrastructure that is used by the wireless carriers. An important characteristic of networks is the bandwidth and connection type. For example, 2.5G and 3rd generation networks will provide high-speed access and always-on capabilities.

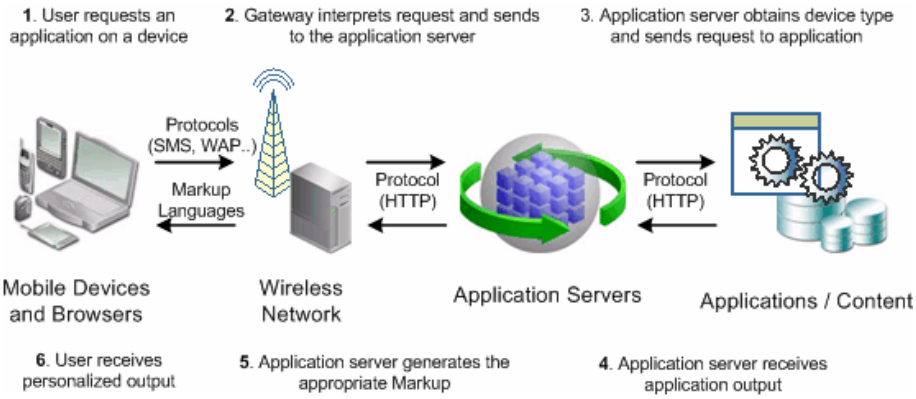
- Protocols are used to deliver the content to devices. Examples of protocols are WAP (Wireless Access Protocol), voice and SMS (Short Messaging Service). Note that wireless protocols are more efficient over the wireless networks than the standard HTTP protocol. This is one of the primary reasons that many wireless Internet clients do not speak HTTP directly.
- Wireless Gateways translate the wireless protocol request to the standard HTTP protocol. Gateways speak a variety of protocols such as WAP, SMS, voice and others.

Application Servers, Applications and Content

- Application servers have come into play to increase the efficiency of application development, deployment and management. A wireless application server connects the wireless content source over the wireless network to the wireless gateway/device. To do so, it retrieves the content from the content source, personalizes it for individual users, and transforms it to the specific markup language spoken by the wireless device being used. This relieves developers of the responsibility to design for all protocols and markups during development. In addition, application servers include tools (testing, device control…) and frameworks (messaging, offline, provisioning…) that simplify the development process.
- Applications/Content have a wide variety of forms including database information, personalization content, alerts, e-mail, location services etc. The large amount of content sources increases the complexity of having a manageable way to deliver each application to every type of device in the most optimized fashion.

The Mobile Request Process

This section describes the process when a wireless device requests a mobile application with the use of a wireless application server in the network. This example uses the WAP standard, however there are many other protocols, like SMS, that could have been used.



Wireless request process

Sending a Wireless Request

A user invokes a wireless service with a web-enabled device (or a telephone for voice). In order to do so, the user dials the telephone number for the appropriate service provider - 2.5G and 3G networks provide always-on access where it is unnecessary to dial a service provider to start a session. The microbrowser on the wireless device sends a request to the wireless network base station. The request can be sent over a variety of different protocols, such as SMS or WAP, depending on the kind of device being used. Note that all of these protocols are packet-based protocols that have been optimized to function over a wireless network with limited bandwidth and intermittent connectivity. Packet-based protocols are more efficient over the existing wireless network than the standard Internet HTTP protocol.

Recognizing and Authenticating the Wireless Device

When the wireless network's base station receives the request, it requests the mobile device to identify itself in order to proceed with authentication. Once the WAP gateway has established a session, it passes information about the specific web request to the wireless application server. The message header encodes information such as the user's identity, the device the user is accessing the Internet with, the geographical location of the user, and the specific web address or service that the user is accessing. This information will be used by the wireless application server to personalize the interaction with the user.

Establishing the Wireless Session

Once authentication is successful, the service provider accepts the call and establishes a connection with the mobile device. The request is sent from the base station over the wireless network using the Wireless Transport Protocol (WTP). The wireless operator's gateway receives the request.

Translating the request over the Internet

Note that since the protocol over the cellular network is not the standard Internet protocol, it needs to be converted into the standard Internet HTTP protocol before

the request is passed from the wireless network to the traditional Internet - a gateway provides this function. For WAP enabled devices a WAP gateway is necessary to bridge WTP to HTTP. The gateway not only maps the message from one protocol to another but also knows how to pass the message from the wireless network to the traditional Internet infrastructure. Other gateways include voice gateways and SMS gateways.

Connecting to the Application Server

The gateway converts the wireless request to a URL for a specific web site. Once converted, the message is sent as a standard Internet request to the wireless application server that sits at the specific URL or web address being accessed. The application server and gateway then go through a standard authentication process each authenticating itself to the other and establish a session.

Recognizing the User's Information

The application server interprets the user's identity, the device, the location and other identification parameters that are passed to it.

Processing the Wireless Request

When the wireless application server receives the request, it processes it in three steps – a) retrieves the content from the wireless application being accessed, b) customizes it for the user, and c) transforms it to the specific device being used. Content adaptation essentially involves aggregating the content from the application being accessed in an XML format.

Customizing the Content for Every User

Oracle9*i*AS Wireless also recognizes the user's session context and customizes the services being rendered to the individual user. Oracle9*i*AS Wireless allows users to configure their own customized portal choosing which services they would like to see, setting up notification services, and personalizing services based on the device they are accessing the Internet from and their geographical location (Location-based Services).

Adapting the Content to the Appropriate Device/Network

Finally, since each user has the ability to use one or more different devices to access the Internet and each device speaks a different markup language, Oracle9*i*AS Wireless transforms the content rendering it to the markup language appropriate to the device being used. Note that most wireless application servers can only adapt a very limited range of content and they can only render them to a small number of devices (typically only those that speak WAP/WML and HDML content). Oracle9*i*AS Wireless can certainly be a WAP server but more importantly it uses XML to translate the content to any format for any device.

Challenges in Mobile Application Development

This section discusses the major challenges in developing mobile applications and clarifies the requirements they place on a mobile development platform. Following chapters will discuss how Oracle9*i*AS Wireless addresses these challenges.

Wireless devices present some unique challenges to accessing information:

Cumbersome User input – It is difficult to type on a cellular phone. Even devices with more user-friendly interfaces, like a Palm and iPaq, can be much less easy to use than a PC with mouse and keyboard. These limitations modify how the Internet can be accessed from a wireless device.

Limited Device Display Form Factor – The screen size and display capability of devices vary tremendously - since applications are likely to be accessed from a variety of different devices, it is not practical to optimize every application for every single device available. The requirements that these two limitations place on a Wireless Platform are twofold: first, the platform must be able to exploit device-specific functionality such as voice browsing which make it easier to navigate through applications from a wireless device; and second, the platform must provide ways to find and execute applications quickly and effectively by personalizing services and content to make them relevant to individual users. The user experience is far richer and more effective when the wireless platform supports a variety of personalization facilities such as allowing users to personalize which services they see, to see different information based on the device they are using, and to see different information based on the geographic location they are accessing data from.

Heterogeneous Wireless Device Standards – Since wireless devices speak different wireless protocols and support a variety of different wireless markup languages - these different standards make it difficult for developers to write every application to individually support every single device available. It is therefore critical that a wireless Internet software platform meet two requirements:

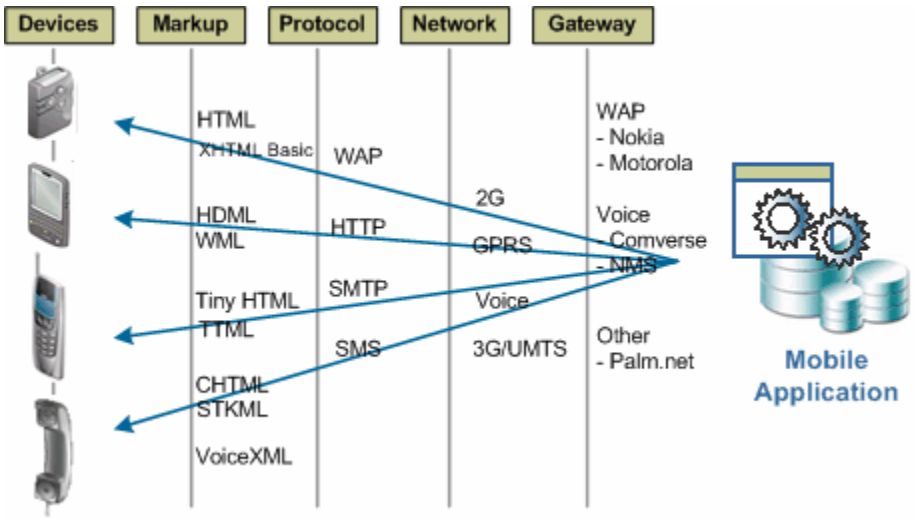
- Support a Broad Variety of Devices and Protocols – First, the wireless software platform should be able to support a broad variety of wireless devices each of which supports a different markup language, a different microbrowser, and a different communication protocol. Most wireless platforms support only the WAP protocol and as a result applications built to such a platform cannot be accessed from an SMS, i-Mode, Voice or a Blackberry device, for instance. With the rapid growth in the number and heterogeneity of wireless devices, the need for a single software platform that supports all wireless devices becomes critical.
- Write Applications Once, Deliver Anywhere – Second, several software vendors claim to take an approach of optimizing how a wireless web site looks on a particular wireless device. As a result, they require a web site developer to develop an application once for each wireless device - once for a Palm, once for a Nokia phone, once for voice, once for a Blackberry

device etc. Recognizing the growing heterogeneity of wireless devices, such an application development paradigm does not scale - a developer must be able to develop a web site once and have the software platform deliver it to any wireless device.

In addition to wireless devices, other wireless complexities make development, deployment and management arduous. This includes:

Heterogeneous Sources of Content – The mobile Internet does not require content and applications that are specific to it. Rather, customers are leveraging existing Internet content and e-Business applications and bringing them to the wireless environment. This drives two requirements on a wireless software platform:

- Development new wireless portals and applications – The wireless software platform must be able to support applications that are being developed specifically for the mobility. It must provide a seamless set of facilities to develop such applications using open standards such as Java and XML. The wireless server must also provide a framework to offer mobile related functions to the applications – location-awareness, customization and messaging.
- For quick time to market, it may be necessary to re-purpose existing content and applications to the mobile Internet – First, the wireless software platform must be able to easily re-purpose any Internet content or application, no matter how it was originally built to a wireless device. This requires the platform to be able to support content from a variety of repositories whether it comes from an Internet web site, from an e-mail server, or from a database. Additionally, the wireless platform must also be able to support any Internet content whether the application has been built using Java, Visual Basic, PERL, PL/SQL, PHP, server-side scripting, or any other web site development language.



Complications of mobile application development

Application Performance and Scalability Requirements – A Wireless Internet Platform must also be able to meet scalability requirements in three ways:

- Support a Large Numbers of Users – Since wireless carriers and portals support millions of subscribers, the wireless platform must provide facilities to centrally manage/provision these users, their security and access control privileges, and their ability to personalize services.
- Support a Large Number of Concurrent Sessions – Additionally, users who access applications from wireless devices often have relatively long lived conversational interactions with a number of different web services. In addition, users desire always-on capabilities for fast notification of messages. Due to the limited bandwidth on the wireless network and the lack of support for "cookies" in most wireless gateways, it is not possible to push the user's session state from the server to the client. As a result, a wireless Internet platform must be able to manage user sessions and maintain session state in a highly scalable fashion.
- Support a Large Volumes of Content – Finally, since wireless users access large volumes of content which needs to be delivered very efficiently to their client, the wireless platform must be able to use caching and share data facilities to serve large volumes of content efficiently.

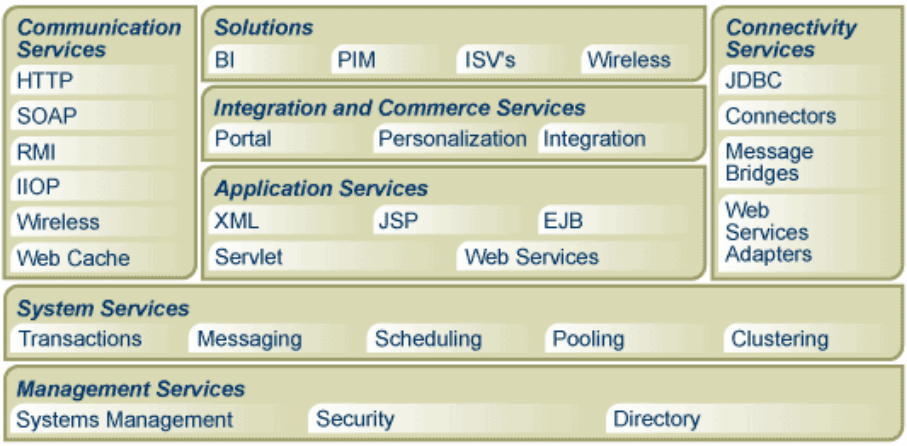
Evolving Wireless Internet Market Requirements – Wireless users want to send messages, browse information and services, carry out wireless commerce transactions, and access business applications. Most wireless software platforms only address a small set of requirements requiring users to choose one wireless software infrastructure for messaging, another to browse content, and yet another for mobile commerce. It is critical that a wireless platform must seamlessly integrate facilities for wireless messaging, content browsing, voice access, mobile commerce

and business applications to allow developers to combine these facilities in building state-of-the-art applications and portals.

Evolving Wireless Standards – Finally, wireless standards are evolving rapidly - at the network level standards such as CDMA, GSM, TDMA, iDEN, SMS, i-Mode, GPRS, and UMTS are all evolving; at the device level standards such as VoiceXML promise to change how the wireless Internet is used. As a result of these differences, a wireless Internet platform must meet two requirements: first, it must be current with wireless standards such as i- Mode, WAP, SMS, GPRS, 3G and others; and second, it must support open industry standards such as XML, XHTML, Java Servlets, Java Server Pages for application development. The Oracle9*i*AS Wireless platform and services address these issues making a complete wireless solution for businesses.

THE ORACLE9*i*AS WIRELESS SOLUTION

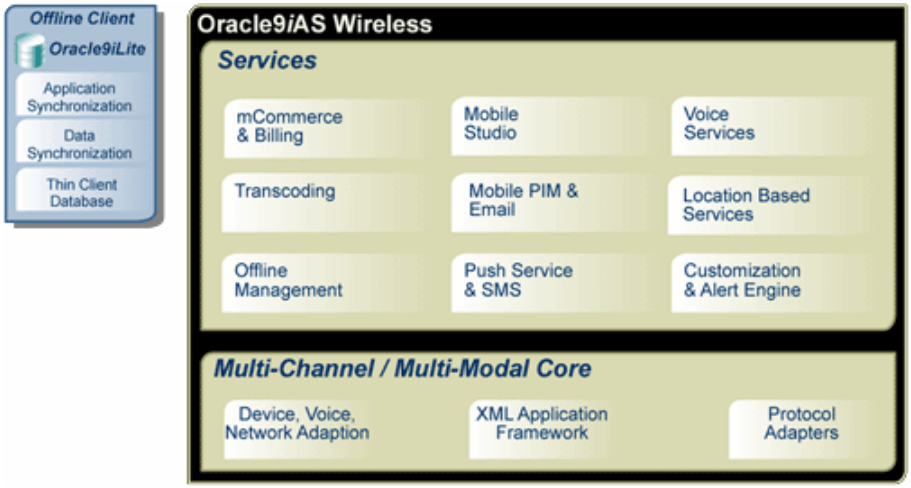
Oracle9*i*AS Wireless is the mobile component of the Oracle9*i* Application Server. The Oracle9*i* Application Server is an integrated suite for internet-enabling your applications and portals. The Oracle9*i* Application Server runs the Oracle e-Business suite as well as partner applications.



Oracle9*i* Application Server Architecture

Oracle9*i*AS makes your Web site and all your enterprise or consumer applications accessible from any browser or mobile device. You can deliver tailored 1:1 customer experiences via real-time personalization and satisfy demands for up-to-the minute business information using Oracle9*i*AS integrated business intelligence services. Oracle9*i*AS simplifies management tasks by using a single management console.

For enterprises and service providers looking to increase efficiency and drive revenue with mobile technology, Oracle9*i*AS Wireless is a complete platform for mobile development and deployment.

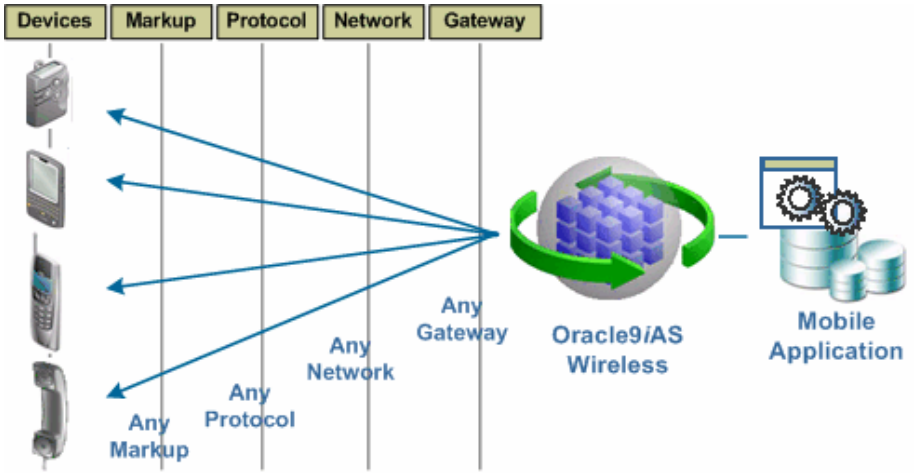


Oracle9*i*AS Wireless Architecture

Oracle9*i*AS Wireless includes mobile services such as PIM and e-mail, location-based, and messaging services (via SMS, WAP-push, fax, e-mail and voice). These services can be run out-of-the-box or be extended with custom functions. Oracle9*i*AS Wireless is an integrated solution employing open standards for mobile-enabling any e-business.

Multi-Channel, Multi-Modal Development

The Oracle9*i*AS Wireless multi-modal core is the framework that gives application developers independence from the underlying networks, protocols, devices, gateways and other wireless complexities. The Oracle9*i*AS Wireless core normalizes the wireless complexities to one protocol and one language, HTTP and XML. The core acts as a multi-channel proxy so developers can create an application once for multiple channels such as voice, push, and SMS.



Oracle9*i*AS Wireless simplifies application development

To utilize the core, a developer creates an application generating XML and points the Oracle9*i*AS Wireless core to the application with a URL. The Oracle9*i*AS Wireless core will detect the device and render the application appropriately, at the same time taking advantage of individual device's features.

The HTTP Adapter is used to retrieve mobile content from any HTTP/XML server. The HTTP Adapter securely retrieves the application content and delivers it to the core for processing. The HTTP adapter supports HTTP 1.1 and accompanying features such as redirects and cookies.

The multi-channel XML Application Framework is the key to developing only once for multiple channels. XML provides simplicity and power to application developers. The XML framework supports service linking (web services), location awareness, and context information to give developers the ability to quickly develop mobile applications with maximum efficiency.

Device and network adaptation transforms and optimizes the application content to any device and network. Supported devices include 2-way pagers for asynchronous services (SMTP/SMS), all WAP devices, voice access through regular phone lines, PDA devices etc. Over 100 mobile devices are supported. Oracle has a proven product patch system to maintain the most current device list.

The Oracle9*i*AS Wireless runtime uses the Oracle9*i* Database as the repository for storing persistent application objects. Runtime API's provide the functionality to manipulate the platform's persistent data objects. The Oracle9*i*AS Wireless APIs can be used to customize the runtime behavior of the server, for example, providing a different authentication scheme or providing a customized device identification mechanism. Oracle9*i*AS Wireless also provides an extension framework, which allows for plug-in of additional logic, like logging or system monitoring.

Oracle9*i*AS Wireless provides a complete web-based tool to manage your wireless business. The Service Designer is used by developers to manage the applications and devices, the Content Manager is used to manage the end user's view, the User Manager controls the users, groups and access control and the System Manager administers processes and performance.

Delivering Applications via Voice

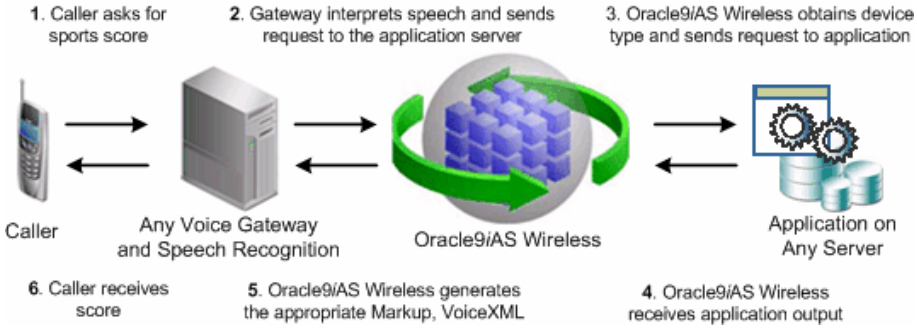
Accessing applications via voice entails an end user calling a server on a traditional phone line and interacting with an audio interface. The end user has two possible input methods: either through speech or the number pad.

Prior to speaker-independent speech recognition, users had to train the software to recognize their utterances. Today, speaker-independent speech recognition and VoiceXML are sufficiently mature for large-scale deployment enabling enterprise-grade voice applications to be developed and deployed for telephone access.

There are three components in such a deployment, the voice gateway, the application server, and the content source.

The application server is responsible for generating VoiceXML pages from the Oracle9*i*AS Wireless XML. These VoiceXML pages typically enable a caller to accomplish a task of some kind using their voice, and may be customized to the caller's personal profile as stored in a database. During the course of a phone call, the application server may generate dozens of VoiceXML pages, however, the generation of VoiceXML pages is transparent to the caller.

Each channelized voice T1/E1 line terminated into a voice gateway supports approximately 24 simultaneous phone calls. Large scale deployments may support 112 T1s per rack of equipment. To handle a specific incoming call, the voice gateway fetches VoiceXML pages from the application server and then interprets the VoiceXML page. The VoiceXML page usually contains a set of audio recordings called "prompts" which may ask the caller to speak a voice command. The VoiceXML interpreter uses speech recognition software, such as that available from Nuance, to understand which voice command the caller has uttered. The interpreter can then take action dependent on this voice command. The action might be to play an audio file, ask another question, or request another VoiceXML page. Oracle9*i*AS Wireless does not ship with a voice gateway; however it functions with any VoiceXML compliant gateway including NMS, Verascape, Comverse, IBM, and Voice Genie.



Voice request process

In this simple, non-personalized example, a caller dials the phone number of the voice gateway. The phone number is associated with an initial VoiceXML page, either static, or dynamically generated. The voice gateway answers the phone and (1) the user asks for a sports score The voice gateway uses speech recognition to determine what the caller uttered and then (2) makes an HTTP request to Oracle9*i*AS Wireless for a VoiceXML page that will contain the score in audio. (3)

Oracle9*i*AS Wireless relays the request for sports scores to the Application. (4) Oracle9*i*AS receives a response in XML from the Application, and (5) transforms it into the markup language appropriate for the user's device, in this case VoiceXML. (6) The voice gateway interprets the VoiceXML and plays the sports score over the phone line for the caller.

Adding 2-Way Messaging Capabilities

The Asynchronous Server allows you to "internet-enable" any phone or device that supports asynchronous messaging like a mobile phone with SMS or a two-way pager. You simply ask for the service you are interested in by sending a short message, and get a reply back in the same manner. What's even better is that you can now interactively run the same applications through asynchronous devices that you ran from synchronous devices (like WAP phones, PDAs or browsers).

Oracle9*i*AS Wireless Async Server allows customers to access wireless services using asynchronous messaging like SMS, e-mail or 2-way pagers. This means that your applications can be used by the vast majority of mobile devices that are capable to send SMS messages but not internet-enabled. Moreover, these applications can be used interactively, as Oracle 9*i*AS Wireless will maintain the session for you.

Using Oracle9*i*AS Wireless Async Server you can run services that may include forms or menus. In order to reduce the number of messages sent back and forth, power users may user "command chaining" to send several commands that get called sequentially.

Oracle9*i*AS Wireless Asynchronous Server uses the Oracle9*i*AS Wireless Transport interface to send and receive messages. Currently, Oracle9*i*AS Wireless ships with a number of drivers on to deliver messages. These include drivers for SMS Centers (SMSC) and e-mail servers. In addition, writing a new driver to support additional devices is supported. There is a well-documented API in Java that can be used to write drivers.

Oracle9*i*AS Wireless adapts the messages based on your device. Hence, if a message is larger than the maximum allowed size, it is broken into several smaller messages for delivery. You can customize the behavior of the Asynchronous Server and hence re-brand it by defining your own command set for invoking services.

Creating Powerful Push and SMS Applications

The Oracle9*i*AS Wireless Push Service provides a highly scalable mechanism to deliver messages to mobile devices. These messages can be delivered in numerous ways such as SMS, e-mail or voice.

The Push Service is built on a scalable message delivery architecture that supports large volumes of messages to many different types of devices. It also provides several ways to manage and track your messages, including status of message delivered. The Push Service allows the addition of business logic, to provide billing and routing control. The Push Service integrates into the user and device preferences of the Oracle9*i*AS Wireless platform where distribution lists of recipients of push messages may be created. These recipients will receive these

messages on the device of their choice, without the application required to do anything special.

The architecture is extensible and adding support for other types of devices and mechanisms can be done by writing a driver against the Oracle9*i*AS Wireless Transport API.

Push services can be used straight away to build rich applications or to make existing applications richer by adding push messaging. At the top level, the application developer can use XML, the Web Service interface or the Push Java Beans to write applications. The Push Service is available as a SOAP-enabled Web Service and comes with a WSDL that can be used to create applications that can then use the Push Service.

Oracle9*i*AS Wireless ships with a number of drivers used to deliver messages. Currently these include drivers for popular SMS Centers (SMSC), e-mail servers, Fax servers and Voice Gateways.

Customizing the End User Experience

Mobile applications are more efficient when applications are tailored to the end users' needs. The Oracle9*i*AS Wireless alert engine and customization framework provide a way to build applications resulting in a one-to-one interaction between the user and the application.

The Oracle9*i*AS Wireless alert engine uses a publish/subscribe model to manage alerts. The two roles in the publish/subscribe model are the Content Manager and the End User. A Content Manager has the ability to publish alert categories to end users and end user groups. The Content Manager creates alert subscriptions using either the Oracle9*i*AS Wireless Content Manager or the Oracle9*i* Wireless alert subscription APIs. End users are able to subscribe and unsubscribe to the published alerts made available to them. The publish/subscribe concept provides a flexible method to categorize alerts by topic. This gives end users an intuitive way to subscribe to the alerts that are relevant to them.

Oracle9*i*AS Wireless alert model

There are three main components involved in delivering an alert to a subscriber group. The Datafeeder retrieves alert content from a content source, the Alert Engine manages the alert subscriptions and trigger conditions, and a transport layer is used as the delivery mechanism to send the alert to any device. These three components are integrated to offer high scalability, however each of the

components can be used independently for other applications.

The Oracle9*i*AS Wireless customization framework is used to streamline mobile applications by understanding the user's roles and preferences — for example, it is beneficial to present information in different ways to customers, suppliers, and employees by understanding the individuals' needs. The result is to turn a series of single transactions into a series of interactions that leads to an enduring, mutually profitable relationship.

Oracle9*i*AS Wireless gives the option to save the input values that a user has entered as a preset value for future invocations – these are called Presets. Furthermore, Oracle9*i*AS Wireless gives options to enter a symbolic name to represent the preset groups. These symbolic names allow easy selection if there is more than one group of preset values. In addition, the user can manage their presets and preset groups with the Personalization Portal from any device or PC.

Additional customization features include User and Group Management, Service Management and Multiple User Profiles.

Reusing Existing Web Applications

The Oracle9*i*AS Wireless Transcoding Service allows applications developed for a particular device or markup to be reformatted for other devices, including voice. Oracle9*i*AS Wireless supports a content adaptation service and a translator service.

The web content adaptation service allows quickly extending existing legacy web application to wireless devices. Oracle9*i*AS Wireless can connect web resources, like an HTML page, and acquire the content for reformatting. The content is transformed into the Oracle9*i*AS Wireless XML format and then rendered to the requesting device' markup language. Web Integration Beans provides an abstraction and masks the complex nature of input and output elements involved in Web Service transactions.

The WML translator delivers existing WML (WAP) applications to non-WML devices. The goal of the WML transcoding service is to provide a simple way for companies with existing WAP services to break the barrier of device-specific applications. The most commonly used wireless language is WML. WML has different syntax and behavior from other device specific languages such as HDML. Oracle9*i*AS Wireless translates the WML into XML as a common language for wireless devices that hides device-dependent complexity. The Oracle9*i*AS Wireless XML schema defines the basic structures that exist in WML. The structures are then rendered into any mobile device, such as Palm, Pocket PC and even voice.

Offline Management

Offline Management is used in cases where mobile connectivity is nonexistent or low. This gives your users the ability to use applications without any network access. When Internet connection is available again, the device user can synchronize to update the server with the new information. Oracle9*i* Lite provides this ability.

Oracle9*i* Lite is an integrated set of technologies that provide critical infrastructure for developing, deploying, and managing offline mobile applications. Oracle9*i* Lite

provides necessary framework businesses need to extend the enterprise applications to all of today's popular mobile platforms: Palm OS, Symbian EPOC, Microsoft Windows CE, and Microsoft Windows 95/98/NT/2000.

Oracle9*i* Lite is strongly grounded on the Oracle9*i* database and the Oracle9*i* Application Server. Oracle9*i* Lite includes two main components:

Oracle9*i* Lite Mobile Server is an offline mobile application server, which runs on Oracle9*i*AS Wireless and allows administrators to centrally provision, deploy, manage, and synchronize offline mobile applications to a wide range of devices. With Oracle9*i* Lite Mobile Server's centralized Application Deployment, IT Departments no longer need to worry about how to roll out the new or updated mobile applications to thousands of mobile devices. Oracle9*i* Lite Mobile Server offers robust, highly scalable, bi-directional, asynchronous data synchronization between Oracle9*i* database and thousands of mobile devices over any wired or wireless network. This enables Mobile Users to have access to latest enterprise data and applications at all times. Oracle9*i* Lite also offers centralized provisioning and management of mobile applications allowing IT Departments to control access to mobile applications from anywhere using the powerful web interface.

Oracle9*i* Lite Mobile Development Kit offers the fastest and convenient way to develop and package enterprise class offline mobile applications for Palm Computing Platform, Microsoft Windows CE, Symbian EPOC, and Windows 95/98/NT/2000. The Mobile Development Kit is a set of tools, API, and sample code that accelerates the development of mobile applications that access the Oracle9*i* Lite database on mobile devices.

At the heart of Oracle9*i* Lite is Oracle9*i* Lite database, a powerful lightweight relational database that offers complete transactional support, data integrity constraints, and the ability to construct queries on the fly. This feature enables business applications that can reliably and effectively operate on small memory constrained mobile devices.

Oracle9*i* Lite supports popular development tools like Microsoft Visual Studio and Oracle JDeveloper, letting developers rapidly develop mobile applications using the tools and knowledge they already possess. Oracle9*i* Lite offers support for ODBC and JDBC across the platforms, which lets enterprises invest in mobile applications that leverage open standard APIs and are easily extendable to new platforms as the business need arises.

Giving Applications Location Awareness

Location-aware applications make decisions based on the end user's geographical location. Location-based services greatly improve mobile applications by making them easier to use and providing quick access to timely and critical information. Companies that take advantage of location-based technologies can greatly enhance the value of their applications. The Oracle9*i*AS Wireless Location-Based Service

not only reduces the number of inputs and lowers the time required to obtain information, but also derives improved efficiencies, enabling access to information that is immediately relevant to users, such as maps, driving directions, traffic reports, or nearby businesses and services.

The performance and capability requirements expected for wireless location-based service can easily approach that of heavily used applications —that is, millions of queries on a daily basis, hundreds of concurrent transactions, and millisecond query-response times. When you build on Oracle9*i*, Oracle Spatial, and Oracle9*i*AS Wireless, you have the assurance that your location-based services solution will be scalable, reliable, and secure. In particular, it will be able to handle the unique storage and CPU-intensive processing inherent in location queries (e.g. street routing, proximity searches, and map rendering).

The Oracle9*i*AS Wireless Location Service includes two components. First is a location framework that offers a full set of APIs to give location abilities to applications. Some of the commonly used location abilities include mapping, geocoding, positioning and routing. Second, Oracle9*i*AS Wireless includes a set of out-of-the-box location applications. The applications are built on the Oracle9*i*AS Wireless framework and focus on reusability allowing developers to extend and integrate them seamlessly into their applications. These applications include a location picker, a map viewer, driving directions and Yellow Pages interfaces.

Location Services give users the ability to choose a location or have their location automatically detected. In order to automatically detect the location of a mobile device, Oracle9*i*AS Wireless easily integrates with vendors such as CellPoint, Ericsson, Nokia, Signalsoft, etc.

Privacy and the security of information are important concerns in a location acquisition system. The location services provide a privacy management component that allows users to view and edit their privacy settings, to enable and disable the positioning operation on themselves, and to authorize one or more people (a mobile community) to obtain positioning information on them within certain time frames. All capabilities are accessible through public APIs.

Extending your Desktop and E-mail

The Personal Information Management Service modules are based on standard protocols, allowing a simple integration into existing environments. The Mobile e-mail Client gives access, from any mobile device, to any IMAP4 or POP3 server. This includes such servers as Microsoft Exchange and Lotus Domino. The Mobile Directory Client connects to any LDAP directory server. Finally, the Mobile Calendar Client integrates natively with Exchange and Lotus Servers, and through published interfaces it enables easy customization to support any Calendar Server.

Service	Description
Mobile Email	Provides full management features to IMAP and POP3 servers from any mobile device, including voice.
Mobile Calendar	Enables users and applications to schedule and manage appointments. Integrates with Lotus and MS Exchange servers.
Mobile Directory	Enables access to LDAP directory servers.
Mobile Address Book	Allows users to manage their own address book and contacts.
Instant Messaging	Enables users to exchange instant messages from mobile devices. Integrates with popular IM networks (e.g. Yahoo, MSN, ICQ…).
Document Management	Integration with Oracle IFS provides file attachment services.

The Oracle9*i*AS Wireless PIM solution has a single "Universal UI" to present a common interface regardless of the back-end. The idea is to have PIM business objects between the UI and the backend implementation, so that the same UI can be used for different backends. The same "Universal UI" can be reused for any new backend that may hit the market.

Creating Advanced mCommerce and Billing Applications

The Oracle9*i*AS Wireless mCommerce Service is a set of modules that securely stores user profiles, supplies information authorized by users for third party applications, and interfaces with on-line payment mechanisms to complete transactions. It also uses FormFiller to map forms and spare the users from the frustration of typing in mobile devices.

Service	Description
m-Wallet	• Aggregates all profiles under one interface • Secure store with 3-part key decrypt/encrypt • User history tracking
Form Filler	• Dynamically save/maintain mappings among applications • History of recently visited sites
Payment Processing	• Integration with online payment solutions • iPayment solution

Mobile Wallet module (m-Wallet) provides a convenient single-click commerce payment mechanism. It is a server side, encrypted entity that contains payment instrument, identification and address information for registered users. m-Wallet enables users to store all the information required to fill out commerce-related forms from any application. That information is used to complete transactions, and through the m-Wallet APIs can be made available to partners and e-merchants authorized by the user.

It processes requests (via proxies) for personal and payment instrument information issued (through HTML or WML forms) by third-parties, and presents

them to users, who decide explicitly what information gets sent back to the requesting application. The power of the wallet is that it securely stores this information for users, providing them an easy, secure shopping experience, and freeing them from repeatedly entering information.

m-Wallet also encrypts/ decrypts all information stored in the repository using a three part key made up from a combination of:

- A system key (specific to each deployment of the product)
- A user-specific key (uniquely identifying users within the system, and retrieved when a function is applied to specific user information)
- The user's trading password

Each portion of the 3-layer key can be changed independently, but each of them is required in order to decrypt wallet-stored information.

Test and Deploy with the Mobile Studio

Enterprises and services providers with mobile development efforts are continually striving for the most cost-effective and efficient methods for developing and deploying mobile applications. Development efforts are enhanced when the separate development groups or communities can develop on the same, easy to use mobile platform. These development groups can span beyond the corporations internal development team to include partners, external developers, contractors, and content providers. The establishment of a development community amongst development groups promotes self-support, developer interest, and a central source of compelling applications.

Oracle9iAS Wireless offers the Mobile Studio as a unique, next-generation development environment allowing companies to benefit from faster time to market, increased productivity, and a dramatically simplified testing cycle, while providing access to the latest mobile applications and tools. The Oracle9iAS Wireless platform leverages a
multi-channel XML to give developers the ability to develop once for multiple channels including voice, wireless browsers (e.g. WAP, PDAs, ...), and messaging (e.g. SMS, e-mail). Developers can also leverage or create reusable applications, called modules, that can be "intelligently" linked to an existing application. Oracle9iAS Wireless enables companies to focus on their business logic - your core competency - while the platform focuses on the device complexity, its core competency.

As the Mobile Studio is easily customizable supporting multiple communities, it is an excellent platform for creating vibrant developer communities that focus on self-sufficiency and reusability to streamline the mobile development process. The Oracle9iAS Mobile Studio is a single platform offering a:

- Tool to successfully build and maintain a vibrant mobile developer community
- Path to increase collaborative development
- Clear and simple mobile application development process for developers
- Clear process for administrators to manage developers and applications
- Tool to efficiently prototype, test and deploy applications to Oracle9iAS Wireless

Mobile Studio creates self-sufficient developer communities

Enterprises and service providers can brand the Mobile Studio and customize its look-and-feel and content in order to integrate it with existing portals. The Mobile Studio is built on open standards; the J2EE/XML architecture to allow for complete customization using well known methodologies. The Mobile Studio's intuitive administrator interface allows web masters to rapidly create a compelling developer portal. This makes it easy for service providers' administrators to support their developer community and attract new development. Please see http://studio.oraclemobile.com for a running instance of the Oracle9iAS Mobile Studio.

A SECURE MOBILE SOLUTION

Secure wireless access to banks, enterprises, mCommerce applications, or any other source of sensitive data is a primary concern for enterprises, carriers and application developers. Security-related issues may be generally classified into standard well-

known categories. The matrix in figure below outlines these issues and the standard ways to avoid them.

Security Issue	Security Solution
Privacy – ensure that only the sender and the intended recipient can read the contents of a message (such as credit card numbers, account numbers, etc)	**Encryption/decryption –** allows two communicating parties to scramble/unscramble information they send to each other via special keys only they possess. In transit, this information is scrambled and unintelligible to any eavesdropper.
Integrity – ensure that information is not tampered with in transit to the recipient	**Digital Signatures –** using an encrypted one-way hash algorithm, it is possible to detect at the receiving end even if a single character has been changed. The values of the hash are unique for the hashed message, and the hash values will not expose the message since the hash is one way only.
Authentication – ensure that all parties are who they claim to be such that there is no *spoofing* (pretending to be someone else, a legitimate entity) and *misrepresentation* (misleading purpose)	**Digital Certificates –** the process of confidently confirming the identity of one party by another party. Typically, a client communicates with a server and both client and server can be authenticated via: passwords (name/password pairs) or certificates (proof of ID from an authorized source)
Non-repudiation – ensure that a party to a genuine transaction cannot falsely deny their participation	**Digital Certificates/Signatures –** Again these are either password based or certificate based and act as proof that a designated party commissioned the transaction.

In order to understand the overall security of Oracle9*i*AS Wireless, let us examine how end-to-end security is enforced using a WAP network as an example. There are three issues to be considered:

1. Wireless Network Security: From the wireless device to the WAP Gateway, a WAP 1.2 compliant network speaks the WTLS (Wireless Transport Layer Security) protocol. WTLS is a close relative of SSL and uses two kinds of certificates to manage encryption and authentication - WTLS server certificates (defined as part of WAP 1.1) are used to authenticate a WTLS server to a WTLS client and to provide a basis for establishing a key to encrypt (a handset); and WTLS client certificates (defined as part of WAP 1.2) are used to authenticate a WTLS client to a WTLS server. Both types of certificates are like standard SSL certificates except that two different certificate formats are defined - X.509 certificates (as in SSL) and WTLS mini-certificates which are functionally similar but are smaller and simpler than X.509 to facilitate processing in a resource constrained handset environment. Additionally, the mini-certificates also implement certification revocation methods that are more efficient over the wireless network than the traditional OCSP protocol.
2. Gateway to Wireless Application Server Security: From a security point of view, a wireless gateway typically performs a security intermediary function

such as bridging a WAP/WTLS protection environment on the wireless side with a HTTP/SSL protection environment on the wired side.

3. Encryption and User Authentication: When a wireless request is sent over the Wireless Network, the following steps occur:

    - The Carrier authenticates that the user is a valid wireless network user before completing the call and letting the user access the network
    - If the user is a valid user, the call is completed and the WAP Gateway receives the WAP request. The gateway and the client then perform a standard WTLS handshake that both encrypts the communication and authenticates the gateway to the handset and vice versa.
    - The Gateway opens a HTTP session to the Oracle9*i*AS Wireless and conducts a standard SSL handshake with it - this authenticates the Gateway to the Oracle9*i*AS Wireless server and vice versa.
    - The user then accesses his or her personal portal and carries out a standard username/password based login; note that if both communication over the wireless network and between the wireless gateway and Oracle9*i*AS Wireless are secure (i.e. if the wireless network supports WTLS) then the username/password combination is not passed in the clear.
    - The user then accesses a web service - the wireless service either accepts the user's identity passed to it through the Oracle9*i*AS Wireless adapter as a bind variable or can ask the user to re-authenticate them again using a username and password.

With an evolving labyrinth of wireless infrastructure - mobile devices (WAP phones, PDAs, pagers, voice, etc), protocols, carriers/providers, and accompanying hardware – maintaining security can be difficult. Depending on your application, Oracle9*i*AS Wireless supports techniques to satisfy your end-to-end security requirements. Oracle9*i*AS Wireless is built on Open Standards that support integration with standard security technology and 3rd party systems.

Oracle builds security models designed to meet the sophisticated security needs for applications such as banking, e-commerce, self-service, CRM and enterprise office applications extended to a mobile workforce. Oracle9*i*AS Wireless utilizes leading edge encryption technology such as Wireless Transport Layer Security (WTLS), Secure Sockets Layer (SSL), Virtual Private Networks (VPN), and Public Key Infrastructure (PKI) to deliver solid end-to-end security across the Internet and the wireless network.

In addition to network security, Oracle supports application security to ensure that the wireless applications protect the integrity of the user's information and the data center's information. A sophisticated ACL (Access Control List) is used to ensure that the appropriate user is mapped to the desired information. All information

such as mobile commerce data and user profile data is encrypted and stored in the secure Oracle9*i* Database.

CONCLUSION

The wireless Internet promises two fundamental opportunities - you can run your company more efficiently using the wireless Internet, and you can exploit new business opportunities that arise because all of your customers, partners, suppliers, and employees happen to be accessing the Internet from a variety of wireless devices. This paper explained the technical details of Oracle9*i*AS Wireless, the world's leading wireless application server and addressed three basic issues:

• Developing wireless Internet applications is challenging for four reasons: (i) Wireless devices have very different form factors and input capability than desktop personal computers; (ii) Wireless device support a variety of different network protocols and markup languages; (iii) Wireless network standards are evolving differently in different markets; and (iv) Wireless Internet market continues to drive new requirements for an integrated platform.

• Oracle9*i*AS Wireless is a complete wireless solution: Oracle9*i*AS Wireless is the only integrated platform in the industry to make all of your applications and data available, any where, any time, and on any device. Oracle9*i*AS Wireless adapts content from any source; transforms content to any device; and personalizes services and content for each user:

• Oracle can help you become a wireless e-Business faster than anyone else: Oracle today is the world's leading Wireless Internet software provider – Oracle powers most of the world's leading Wireless Internet carriers and portals. Oracle9*i*AS Wireless provides an infrastructure platform to build your own wireless Applications; Oracle's e-Business Suite provides an integrated suite of wireless-enabled Applications to run your e-Business; and Oracle offers expertise in designing, implementing and hosting your Wireless Applications.

March 2002
Author: Kalle Radage

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com