MOBILIZING ORACLE APPLICATIONS ERP
An Approach for Building Scalable Mobility Solutions

A RapidValue Solutions Whitepaper
# TABLE OF CONTENTS

**Executive Overview** .......................................................................................................................... 3

**Typical Architecture for Mobilizing Oracle Applications ERP Modules** ................................. 4
  - Integrated SOA Gateway .................................................................................................................. 5
  - Server Middle Layer ....................................................................................................................... 5
  - Mobile Client Application .............................................................................................................. 5

**Key Benefits of a Middle Layer** ..................................................................................................... 6

**Enabling Integrated SOA Gateway and Publishing Web Services** ................................................. 6

**Options for Building Mobile Client Applications** ............................................................................. 7
  - Native Apps .................................................................................................................................... 7
  - Cross-Platform Apps ....................................................................................................................... 7
  - Mobile Web Apps ............................................................................................................................ 8

**Oracle ADF Mobile** .......................................................................................................................... 11

**Key Steps: Designing Mobile Applications Integrated with Oracle E-Business Suite** .......... 12
  - Requirements and Strategy ............................................................................................................ 12
  - Oracle SOA Design and Build ........................................................................................................ 13
  - Mobile Client Application Design and Build ................................................................................ 13

**Case Study: Building a Field Service App on Oracle Applications ERP** .................................... 14
  - Requirements .................................................................................................................................. 14
  - Solution .......................................................................................................................................... 14

**Appendix: Generating Web Services in an Integrated SOA Gateway** ............................................ 15
  - Custom Interface Package for Creating WSDL .............................................................................. 15
  - Generating Web Services in Oracle ............................................................................................... 16
  - Generated WSDL Example ............................................................................................................ 16

**About RapidValue** ............................................................................................................................ 17
EXECUTIVE OVERVIEW

“Enterprise mobility has transformed the way businesses engage with their customers, partners and staff, using innovative applications that enhance and accelerate the exchange of critical information. However, mobile applications are only part of the overall solution – integration with existing systems in a scalable manner is equally important and challenging.

Enterprises are recognizing the importance of extending their ERP (Enterprise Resource Planning) systems to mobile devices and providing employees with access to internal systems while on the move. This paper examines the approaches for extending Oracle E-Business Suite ERP modules to mobile. It describes the overall architecture and solution, and includes an example of extending the Oracle Applications Field Service module to mobile. In the example, a mobile application is built on the Android platform to leverage the capabilities of the mobile device.

Oracle Integrated SOA module is leveraged to expose web services to outside the firewall. Web services can be generated using pre-built packages or custom packages so as to enable required behavior of the mobile application. A middle layer is introduced to ensure security and mobile device management capabilities. The middle layer ensures that the ERP system is not exposed to outside the firewall to protect the system and its data.

Once web services for requisite functions are exposed, mobile client application can be developed using multiple options: native, cross-platform and mobile web. The technology used depends on business requirements for usability, functionality and the long-term technology roadmap. Oracle ADF Mobile is an excellent option for extending Oracle Applications modules, as it fits well into the overall architecture stack. To select the right approach for developing the mobile client platform, please refer to our white paper: “How to Choose the Right Architecture for your Mobile Application” at http://www.rapidvaluesolutions.com/whitepaper/
TYPICAL ARCHITECTURE FOR MOBILIZING ORACLE APPLICATIONS ERP MODULES

The diagram below illustrates key architectural components involved in mobilizing Oracle Applications. The main components include:

- **Integrated Service-Oriented Architecture (SOA) Gateway** – a core component of Oracle Applications, allows service enabling of PL/SQL integration interfaces and deploys them as web services.

- **Middle Layer** – a custom integration layer that provides security and some mobile device management (MDM) capabilities. The middle layer can also be integrated with a standard MDM package that provides security features. This is an optional layer in the architecture; however this layer increases scalability and security considerably.

- **Mobile Client Application** – can be built as a native application, hybrid application or a mobile web application. The mobile client interacts with the ERP modules leveraging web services exposed through the Integrated SOA Gateway module.
Each component is explained more in detail in the following sections:

**INTEGRATED SOA GATEWAY**
Oracle Integration Repository module within Oracle Integrated SOA Gateway is the centralized repository that contains a number of interface endpoints exposed by applications within the Oracle E-Business Suite. These interfaces are typically exposed as PL/SQL packages that provide certain functionality. The Integrated SOA Gateway provides a comprehensive, consistent browsing view of the interface mechanism which lets users easily discover and search on the business interface from the catalog. In addition, web services can be generated by transforming interface definitions to a machine-processable format that complies with web standards using WSDL. These web services can then be deployed to an application server and used for building mobile applications and other external applications.

**SERVER MIDDLE LAYER**
A middle layer is an external layer built on the server that facilitates communication of the Mobile Client Application with the enterprise ERP System. This layer resides on a separate server on the DMZ and provides a bi-directional channel between the Oracle SOA and the mobile client application. The middle layer ensures that the server running the main Oracle applications architecture is not exposed to outside the firewall. Only a specific port from DMZ will be exposed to outside world, thus increasing security considerably.

This can be a custom-built layer or one of the MEAP solutions. (Please refer to our white-paper “Point-to-point vs. MEAP – The Right Approach for an Integrated Mobility Solution” for more details on architecting the middleware.) The middleware layer will maintain a secure channel or connection with the ERP system within the Enterprise for transferring desired data between the ERP system and the mobile client application. The middleware layer handles all system integration, security, communication, scalability and cross-platform support. Typically, no data is stored in the middleware; it only manages data from the ERP system to the mobile client application and vice versa.

**MOBILE CLIENT APPLICATION**
The end users of the system interact with the ERP system using the Mobile Client Application. Users will be able to see required data from the ERP system on the mobile client application. They will also be able to modify and add new data via the application.

There are multiple options for building mobile client applications, from native applications that provide the best usability to mobile web that provides maximum device coverage. The right approach depends on the user requirements. More information can be found in the section “Options for Building Mobile Client Applications”.

©RapidValue Solutions
KEY BENEFITS OF A MIDDLE LAYER

Without a middle layer, the main Oracle architecture would be exposed directly to outside the firewall. The middle layer maintained in the DMZ acts as a bi-directional channel between the Oracle SOA gateway and the mobile device applications, providing secure communication by opening only a specific port on the DMZ to the outside world, thus reducing the impact of external threats.

This layer can be custom built or can be a MEAP solution. Some of the functionality for which the middle layer can be leveraged includes:

- Mobile device provisioning and management
- Mobile device functions such as remote-wipe of data
- Analytics and reporting of app usage, GPS mapping, etc.
- Integration with other systems. This layer can be used as a common integration layer which enables information from multiple sources to be integrated into a mobile application.
- Static data cache. Data configurations that do not change frequently can be stored in the middle layer to reduce traffic between the mobile application and the server
- Optional usage of the application while the Oracle system is down. The middle layer can cache data and synchronize with the Oracle system when the server is in back-up mode. (Which data elements can be synchronized needs to be evaluated carefully to avoid any compromise of data.)

Some of these capabilities can be achieved using a standard MDM solution. Therefore, detailed evaluation needs to be done to understand which functionality needs to be included in the middle layer.

ENABLING INTEGRATED SOA GATEWAY AND PUBLISHING WEB SERVICES

Oracle Integrated SOA Gateway enables publishing of PL/SQL package interfaces and business events from multiple modules and making these packages available as web services for external applications.

Integration Repository comes with a set of native packaged integration interfaces. These interfaces can be transformed into web services through a service generator. Web services are generated in a machine-processable format called WSDL (Web Services Description Language). External systems including mobile clients interact with the web services using SOAP (Simple Object Access Protocol) messages.

In addition to the pre-built integration packages, custom interface packages can be created and web-services generated out of these.
A typical web services generation and deployment process flow is shown below.

1. After the interface integration definitions have been identified, custom packages can be written for them. An integration administrator transforms the integration interface definitions residing in Oracle Integration Repository into web services in WSDL URLs.

2. An integration repository administrator then deploys the web services.

3. Web services are used to build mobile applications.

Please refer to the appendix for an example of WSDL generation.

OPTIONS FOR BUILDING MOBILE CLIENT APPLICATIONS

While many technology options are available for mobile applications development, it’s possible to simplify the lay of the land. Mobile applications can be developed in three fundamental ways:

NATIVE APPS
Build the application for individual platforms using its native development environment. The application is downloaded from an app store and resides on the device.

CROSS-PLATFORM APPS
Build the application once using a cross-platform framework and run it on multiple mobile platforms (after fine-tuning for each platform). The application is downloaded from an app store and resides on the device.
**MOBILE WEB APP**

Build the application to run over the internet from a central server, from where it can be accessed by any device with a web browser.

Due to differences in underlying technology, each approach has inherent advantages and drawbacks, development frameworks and appropriate use cases. Careful analysis is required to ensure that an application is built using the right technology for the functionality required. The following table summarizes the different approaches:

<table>
<thead>
<tr>
<th>Development Approach</th>
<th>Native</th>
<th>Cross-Mobile Platforms</th>
<th>Mobile Web</th>
</tr>
</thead>
</table>
| **Definition and Tools** | Build the app using native frameworks:  
- iPhone SDK  
- Android SDK  
- Windows Phone SDK | Build once, deploy on multiple platforms as native apps:  
- Oracle ADF Mobile  
- PhoneGap  
- Worklight | Build using web technologies:  
- HTML5  
- Sencha  
- JQuery Mobile  
- Etc. |
| **Underlying Technology** | iPhone: Objective C  
- Android: Java  
- Windows Phone: .NET | Oracle ADF Mobile: HTML5,  
JavaScript and CSS  
- Appcelerator: JavaScript, HTML  
- PhoneGap: JavaScript, HTML | JavaScript, HTML |
| **Deployment** | App stores | App stores | Over the web |
| **Key Use Cases** | Apps requiring high-end user experience, more transactional in nature  
- Large user base on one device (e.g. physicians with iPad)  
- Offline usage  
- Apps requiring extensive device and/or OS functions | Simpler apps, more informational in nature  
- Apps with content updated on a frequent/live basis—perfect for news-type applications  
- Multiple device types distributed across key users | Generic user experience, performance depends on Internet connection  
- Distributed user base across smart phone platforms  
- Need to maintain single code base  
- Moderate amount of device functions |

For mobile applications integrating with Oracle Applications ERP modules, Oracle ADF Mobile would be an excellent solution since it fits into the overall technology architecture stack.
Key questions that need to be considered when identifying the appropriate mobile client technology include:

- Who will use the app?
- What type of experience do users expect?
- What functionality is required?
- How important is multi-platform compatibility?
- Does the app need to run offline or when the connection is bad?
- Are there industry-specific requirements?
- How much time do you have to build the app?
- Does your staff have the necessary skill sets?
- What’s the total cost of ownership?
- How soon do you need new device or OS features?

The table below compares the different approaches:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Fully Native</th>
<th>Cross-Platform Frameworks</th>
<th>Mobile Web Apps</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost of Development</td>
<td></td>
<td></td>
<td></td>
<td>- Cross-platform frameworks and mobile web apps reduce development costs, as the application can be built for multiple platforms.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Native apps require a separate development effort on each platform, with minimal reuse.</td>
</tr>
<tr>
<td>Multi-Platform Compatibility</td>
<td></td>
<td></td>
<td></td>
<td>- Mobile web apps provide maximum reach, running on any phone with a web browser.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Cross-platform frameworks reach all major platforms (but not always the same ones). Native features not included can be coded as plug-ins.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Native apps run only on the operating system they are developed for.</td>
</tr>
</tbody>
</table>
### User Experience

- Native apps provide the best user experience.
- Cross-platform frameworks may not give the best experience on all phones, with web-app look & feel in some cases. Scrolling and touch response are slower than native apps.
- Web apps provide a generic experience that may look rather clumsy on some browsers (although this is changing with HTML5).

### Features & Performance

- Native apps provide access to all device features and provide the best performance.
- Apps developed using cross-platform frameworks can provide features comparable to native apps, but somewhat slower performance.
- Mobile web apps have more limited capabilities and performance. For example, Push Notification alerts are supported in native/cross-platform apps but not mobile web apps.

Please refer to our white paper “How to Choose the Right Architecture for Your Mobile Application” for a detailed discussion on this topic.
For mobile applications built on Oracle Applications, ADF Mobile is a great choice for building applications. The ADF Mobile Architecture is illustrated below.

Oracle ADF Mobile is a hybrid platform that allows building applications in Java and HTML5. The applications can run on both iOS and Android Operating System, allowing developers to build applications in the technology with which they are familiar.

Some key advantages of Oracle ADF Mobile are:

- **Multi-platform**: Ability to develop once and deploy on multiple platforms like iOS and Android
- **Device Access Features**: Ability to access device features like GPS, Camera, Email, etc.
- **Ease of development**: ADF mobile leverages Java and HTML5 frameworks, making development easier
- **Security**: ADF Mobile provides security that works with the server back end using a consistent login mechanism and SQL Lite encryption
- **Scalability**: ADF Mobile can be extended to other touch-friendly platforms like jQuery, making the platform highly scalable.
The following image provides a framework for building mobile applications integrated with Oracle Applications ERP modules.

**KEY STEPS: DESIGNING MOBILE APPLICATIONS INTEGRATED WITH ORACLE E-BUSINESS SUITE**

The key steps are:

- Identify main functionality
  
  For the Field Service mobile application described below, the key requirements would be login, view tasks, view service requests and update debrief.
Identify key data elements/objects such as tasks, service request and installed base

- Identify local data storage/cache requirements

Understanding data synchronization requirements is critical to build a scalable system. For example, in a mobile Field Service application, the requirement may be to show only the active tasks for a particular user which will be cached for that user. Also, some of the static data elements like part numbers and sub-inventories that do not change on a regular basis can be stored locally and synchronized only when a new data element is added.

An optimal definition of data elements that are stored locally vs. accessed live will require the right balance between performance, robustness and functionality. Multi-user support is also an important factor to consider while caching data locally.

**ORACLE SOA DESIGN AND BUILD**

This phase involves identifying the web services that are to be created for the mobile application, mapping with existing web services along with interface packages and creating new services as required.

The key steps are:

- Identify web services

  For example, in the case study below, some of the web services would be login, view all tasks or a specific task, and update task status.

- Map web services to existing native packaged integration interfaces

  Oracle Integration Repository in SOA Gateway comes equipped with a number of pre-defined integration interfaces (PL/SQL packages). Identify all interfaces that can be used.

- Identify and create new integration interfaces required

  Depending on your business requirements, custom web services may be required. Identify the new integration interfaces (PL/SQL packages) that need to be created to enable this.

- Generate WSDLs from the integration packages

  Create and deploy a web service WSDL file for a selected interface using Oracle SOA Gateway. This needs to be done for both pre-defined (seeded) integration packages as well as custom integration packages. Custom packages will need to be loaded to the integration repository using the FNDLOAD utility provided by Oracle.

**MOBILE CLIENT APPLICATION DESIGN AND BUILD**

Identify the right technology framework to develop the mobile client application after considering all factors discussed in the previous section “Options for Building Mobile Client Applications”.

©RapidValue Solutions -13-
CASE STUDY: BUILDING A FIELD SERVICE APP ON ORACLE APPLICATIONS ERP

REQUIREMENTS
A large consumer product firm wanted to extend their Oracle Field Service Application to Android tablets. Some of the key requirements were:

- Login and view tasks
- View service requests for a task
- View customer, location and product details for a task
- View installed base related to a task
- Create debrief information for a task
- Update status of a task

SOLUTION
Integrated SOA Gateway was leveraged to publish web services for interface packages. Custom packages were identified and web services were created. Web services were used to integrate with the mobile client. The mobile client in this case was built on Android native platform, as most of field service personnel were provided with Android tablets.

A few screens of the app.
APPENDIX: GENERATING WEB SERVICES IN AN INTEGRATED SOA GATEWAY

CUSTOM INTERFACE PACKAGE FOR CREATING WSDL

CREATE OR REPLACE PACKAGE BODY XX_FS_MOB_LOGIN
IS

/**********************
Package Name: XX_FS_MOB_LOGIN
Purpose : This package outputs assignee details

**********************************************************************

History:
Version | Date       | Created / Modified By | Comments
---------|------------|-----------------------|----------
1.0      | 09/03/12   | Nithin                | Initial Creation.

*****************************************************************************/

--get_login

    procedure get_login(
        p_username IN varchar2,
        x_assigneeid OUT number,
        x_assigneedisplayname OUT varchar2,
        x_return_msg OUT varchar2
    ) IS
        assigneeid number;
        assigneedisplayname varchar2(50);
    begin
        SELECT res.resource_id assignee_id
            ,res.source_name assigneedisplay_name
        into assigneeid,assigneedisplayname
        FROM per_all_assignments_f paaaf
            ,pay_people_groups g
            ,per_all_people_f papf
            ,jtf_rs_resource_extns res
            .fnd user fu
GENERATING WEB SERVICES IN ORACLE

```xml
 xmlns=http://schemas.xmlsoap.org/soap/
 xmlns:s= http://xmlns.oracle.com/apps/csf/soapprovider/plsql/xx_fsmob_login/get_1c
>
  <complexType>
    <sequence>
      <element name="Response" minOccurs="0" type="string/>
      <element name="AppName" minOccurs="0" type="string/>
      <element name="SecurityGroup" minOccurs="0" type="string/>
      <element name="WLanguage" minOccurs="0" type="string/>
      <element name="Org_Id" minOccurs="0" type="string/>
    </sequence>
  </complexType>
</definitions>
```

©RapidValue Solutions
ABOUT RAPIDVALUE

A global leader in digital transformation for enterprise providing end-to-end mobility, omni-channel, IoT and cloud solutions. Armed with a large team of experts in consulting, UX design, application development, integration and testing, along with experience delivering projects worldwide, in mobility and cloud, we offer a wide range of services across industry verticals. We deliver services to the world’s top brands, fortune 1000 companies, Multinational companies and emerging start-ups. We have offices in the United States, the United Kingdom and India.