SUP: Concepts

Applies to:

This document applies to Sybase Unwired Platform 1.5.2. For more information, visit the Mobile homepage.

Summary

As Enterprise Mobility is gaining more and more importance day by day, the acquisition of Sybase by SAP has a huge significance. And this article provides an overview on Sybase Unwired Platform 1.5.2.

Author: Rahul Muraleedharan

Company: Applexus Software Solutions (P) Ltd

Created on: 24 November 2010

Author Bio

Rahul Muraleedharan is working as SAP Technology Consultant with Applexus Software Solutions (P) Ltd. He has an experience of over 1 year in SAP ABAP.

Table of Contents

Introduction

As Enterprise Mobility is gaining more and more importance day by day, the acquisition of Sybase by SAP has a huge significance. SAP is world's one of the leading EIS and Sybase one of the leading MEAP. So there will be a huge leap in the field of Enterprise Mobility by SAP Sybase combination within a couple of years. Sybase Unwired Platform (SUP) provides a set of comprehensive services that allow customers to mobilize appropriate data and business processes for enterprises using virtually any mobile device.

Challenges in Enterprise Mobility

1. Device Heterogeneity.
2. Data partitioning for Mobile Applications.
3. Data Transport and Application Security.
4. Bandwidth, Latency and Connectivity Issues.
5. Managing devices and the Application Life Cycle.

SUP meets these challenges by hiding system complexity while efficiently and flexibly handling complex mobility issues. SUP provides many opportunities for productivity gains through a well-defined life cycle, repeatable use of existing investments, and mobility knowledge.

Platform Components

1. Server Tier
2. Data tier
3. Client tier

Server tier components integrate with back-end enterprise systems, data access and transaction services, device and application deployment, and system management functionality.

Data Tier is used to store data retrieved from the backend data sources and other runtime related metadata.

Client Tier consists of device applications built on top of the Unwired Platform client runtime.

Unwired Platform includes a complete application development environment. Unwired Platform offers a choice of multiple application models—native application development, Device Application Designer development, or simple codeless business process mobilization—by reusing your development and deployment investments in the Unwired Platform server tier.

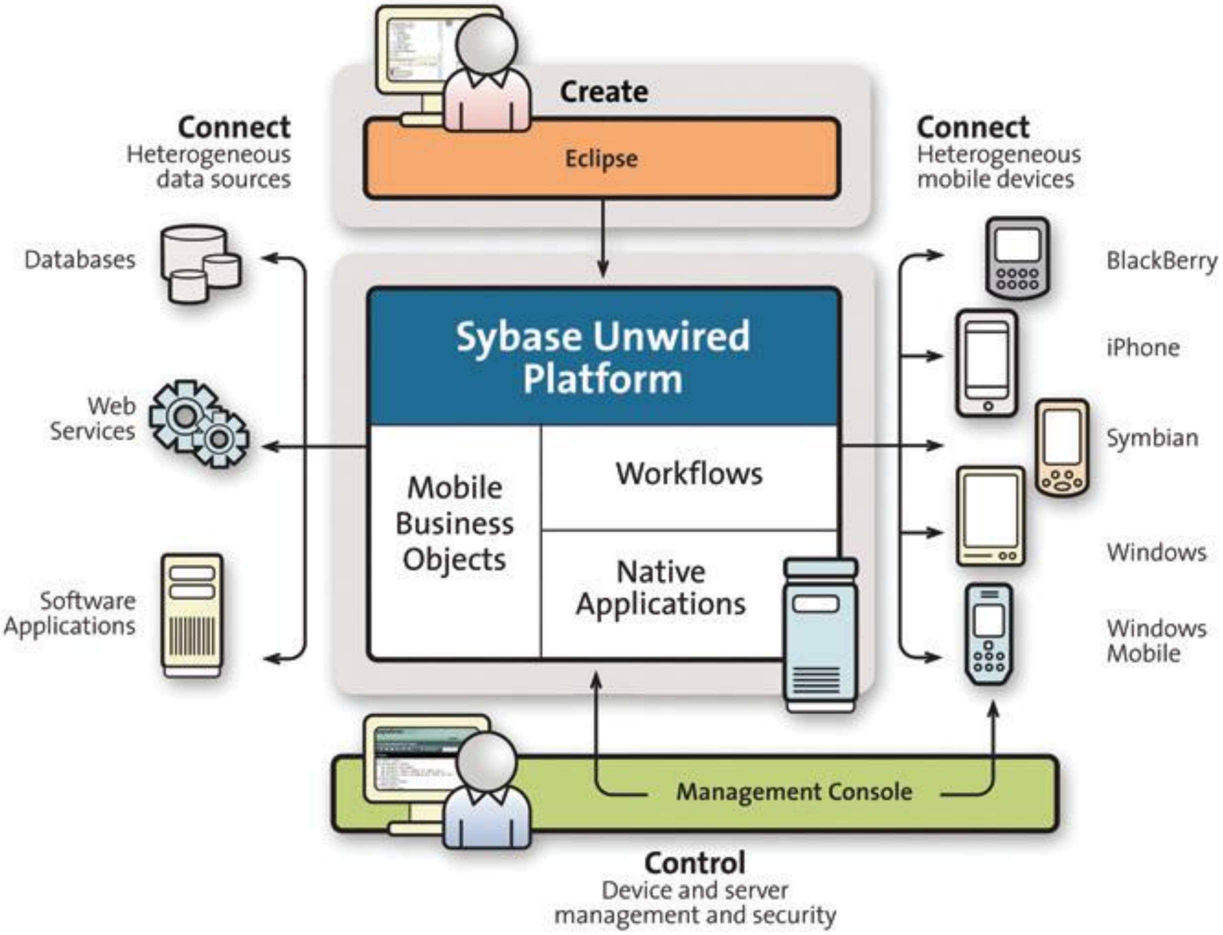
Advantages of SUP

- Single Robust Mobile Infrastructure: SUP provides scalable, secure and reliable environment for multiple types of device platforms.
- Mobile Application Enablement: SUP enables partners and customers to develop and deliver robust multiple mobile applications.
- Complete Enterprise Integration: It enables existing enterprise applications utilizing a multitude of existing SUP backend connectors.
- Application Life Cycle Management: The developed application can be managed through its life cycle using Afaria.

SUP tames complexity

SUP allows you to strategically adapt to the rapidly changing enterprise World. SUP offers seamless integration to a variety of backend data sources. It provides a consistent development platform, ie., the existing skills and technologies can be made use. It also enables rapid deployment of mobile applications across multiple device types.

SUP Architecture



1. Connect the SUP to the backend server, the backend data source can be either database, web services or other software applications. Now the data has to be modeled into Mobile Business Objects (MBO) which contains business logic.
2. The entire mobile application can be developed and deployed within Eclipse plug-in.
3. Connect various device types using SUP.
4. Deploy and manage the device settings and applications. It provides a centralized control of the entire system from a single control center web console.

Key Elements of SUP

1. **Mobile Business Objects:** It help form the business logic for mobile applications.
2. **Device Applications:** The building blocks of the device application include the user interface (UI) layer, business logic layer, Unwired Server client object APIs for retrieving data and executing operations, frameworks for passing data and handling events, and device platform and third-party component APIs.
3. **Synchronization Methods:** Developers can use either replication-based or message-based synchronization to move data and transactions between device application clients and Unwired Server.
4. **Security:** End-to-end security is an important component of the Unwired Platform solution architecture. Security comprises multiple components, including system security to control access to data and transactions, transport security for over-the-air and network level data protection, local data security for protecting enterprise data on the device, and device security to protect devices in case of loss or theft.
5. **Application Deployment:** During development, to test the application, the developer deploys mobile business object (MBO) packages to Unwired Server, and device application binaries and their dependencies to emulators or actual devices. Deployment to devices can be achieved by physically connecting the device to the developer's machine, and copying binaries using device-supported tools.
6. **System Management:** The Unwired Platform management console offers integrated, Web-based access to all Unwired Servers running in the corporate network.

Mobile Business Objects

A mobile business object (MBO) is derived from a data source (such as a database server, Web service, or SAP server). MBOs are deployed to Unwired Server, and accessed from mobile device application clients. u can define MBOs using either a top-down approach—first designing attributes and parameters, then binding them to a data source; or a bottom-up approach—first specifying a data source, then automatically generating attributes and parameters from it.

Data and Transaction Models

Mobile business objects (MBOs) implement both a data model and a transaction model.

Attributes

Attributes define the structure for the data associated with the MBO instance on the mobile device.

Operations

Mobile business objects (MBOs) may incorporate operations that change the data retrieved from the enterprise information system (EIS).

Relationships

Relationships define the data association between two MBOs by linking attributes and parameters in one MBO to attributes and parameters in another MBO.

Cache Groups and Synchronization Groups

The package definition can include cache groups and synchronization groups.

Roles

To secure access to an MBO and its operations, developers can assign logical roles, which, together with the physical role (which exists on the underlying security provider repository), helps Unwired Server perform real-time authorization checks against the user identity before allowing the request to go to the enterprise information system (EIS).

Data Sources

A data source is the enterprise information system where data is retrieved from and transactions are executed. A connection profile is a design-time connection to a data source. Connection profiles are created to specific data source by providing connection information such as host, port, login, and password among others. The connection profiles are used to define MBOs and operations, and mapped to existing, or used to create new, server connections when the package is deployed to Unwired Server..

Data Caching

Data caching is initial loading (or filling) the consolidated database (CDB, or Unwired Server cache) with enterprise information system (EIS) data, then continuing to refresh the consolidated database with changes from the EIS or mobile device on an ongoing basis.

EIS to Server Cache Integration

You can use multiple mechanisms to update the data in the Unwired Server cache.

Other Key Concepts

Other key concepts for understanding mobility include object queries, synchronization parameters, result set filters, result set checkers, and personalization keys.

Object Code Generation

To access and integrate MBOs in a device application, developers have the option to generate object code for the target device platform, and then use their IDE of choice to build the native device application. If they use Sybase Unwired Workspace Device Application Designer, the object code is internally generated when the device application code is generated and deployed to emulator. The object code generation step is the bridge from the Unwired Server server-side development (MBOs) to client-side development (device application).

Deployment

The last step of mobile business object (MBO) development is to deploy the MBO definitions to Unwired Server as a deployment unit generated from a design-time deployment package using Sybase Unwired Workspace.

Mobile business objects (MBOs) implement both a data model and a transaction model.

- Data model – attributes provide the abstract model that describes how data is represented and accessed in Unwired Platform. The goal is secure access to enterprise information system (EIS) data from the mobile application.
- Transaction model – operations provide the abstract model for data transactions. The goal is to deliver updated data from mobile devices to the EIS.

Device Applications

Unwired Platforms offers the following options for developing device applications:

- Object code generation – enables you to generate object code for a device platform, then use the code in the native IDE—such as JDE for BlackBerry, and Visual Studio for Windows Mobile—to build an application. This option is suitable for developers with advanced knowledge of the target device platform development.
- Device Application Designer – enables you to use graphical device application development tools to design and create device applications for BlackBerry and Windows Mobile platforms. This method uses the MBO metadata to automatically generate the required screens to support the data viewing and operation execution of MBOs. You can also use this method to quickly create a prototype. This option is suitable for developers new to mobile application development or a device platform.
- Mobile Workflow Forms Editor – enables you to create a simple business workflow type of application.

Application Types

Sybase Unwired Platform supports two choices for application type. First is the native application type, and the other is the container-based business workflow type.

Application Components

The device application components you select depend on the functionality you require in your application.

Client Object APIs

The Unwired Server client object APIs form the core building block of device applications and provide the common set of APIs for consistency across platforms, thus following the "design once and deploy anywhere" paradigm of building applications.

Synchronization Methods

Developers can use either replication-based or message-based synchronization to move data and transactions between device application clients and Unwired Server.

The choice depends on the target device platform, application requirements, target platform, and the nature of data changes and activity between Unwired Server and clients, for example, mobile workflow forms always use message-based synchronization.

Unwired Server manages and maintains data freshness between multiple data sources and device application through synchronization.

Replication-Based Synchronization

The replication-based synchronization model uses relational database replication, and is session-based to reduce the overhead required to maintain a continual connection. During the session, the clients submit pending operations and obtain the latest changes.

Message-Based Synchronization

The message-based synchronization model is inherently asynchronous. The interaction, however, can either be synchronous, or asynchronous for messaging-based synchronization. During transmission, the client and server submit pending operations and obtain the latest changes.

Security

End-to-end security is an important component of the Unwired Platform solution architecture. Security comprises multiple components, including system security to control access to data and transactions, transport security for over-the-air and network level data protection, local data security for protecting enterprise data on the device, and device security to protect devices in case of loss or theft.

System Security

System security is built using a component-based extensible model of common security infrastructure that allows for pluggable mechanism to integrate Unwired Server with security providers including the most common LDAP providers— Active Directory, Windows OS, Remedy, and others—for authentication and authorization.

Transport Security

End-to-end data encryption support is based on Transport Layer Security (TLS) and Secure Sockets Layer (SSL), which secures client/server communication using digital certificates and public-key cryptography.

Device Data Security

Optionally you can encrypt device data so that data at rest is always encrypted and available only after users have been successfully authenticated. Unwired Client runtime APIs include an API to encrypt the device database.

Application Security

Application security is based mainly on the mapping of a mobile business object (MBO) package to a security configuration.

Authentication

Unwired Server authenticates users accessing packages using the security provider of the mapped security configuration.

Role-Based Access Control

Access to mobile business objects (MBOs) and operations that are mapped to logical roles is authorized by delegating authorization checks to mapped security provider for the package. The given user's principal is checked for its membership in the corresponding physical role that is mapped to the logical role assigned to the MBO or operation.

Client Credential Propagation

Access to an enterprise information system (EIS) can be granted either by using a fixed set of credentials for all users, or by passing in the client user credentials.

Device Security

Unwired Platform supports Afaria device management and security functionality, which includes features such as remote device locking, remote data cleanup, data fading (a feature that enables the IT administrator to lock, wipe, or reset a device that has not communicated with the corporate network or Afaria server after a predetermined number of days), and password expiration management. Even without Afaria, the Unwired Server administrator can lock or unlock devices from accessing applications deployed to the server.

Application deployment

During development, to test the application, the developer deploys mobile business object (MBO) packages to Unwired Server, and device application binaries and their dependencies to emulators or actual devices. Deployment to devices can be achieved by physically connecting the device to the developer's machine, and copying binaries using device-supported tools.

Device Testing

Device applications are typically tested by developers in an emulator environment. All target platforms provide emulator support from their native IDE, which enables deployment and execution of the application to the emulator.

Large-Scale Device Deployment

Some device platforms require signing the binaries before they can be deployed. For production scale deployment, application deployment is done using proven products such as Afaria.

Device Registration

Application users and their devices are registered in Unwired Server for licensing and monitoring.

Related Content

Sybase info center

NetWeaver Mobile / Sybase

CD121 An Overview of Sybase Mobility Solutions for SAP

For more information, visit the Mobile homepage

Copyright

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.