Sun GlassFish Mobility Platform 1.1 Architectural Overview

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 820–7204

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

090217@21990

Contents

Preface

This guide provides an introduction to the architecture of Sun GlassFishMobility Platform 1.1.

Sun GlassFish Mobility Platform is a comprehensive mobility solution that enables offline data access, data synchronization, and secure access to EIS/EAI applications such as Siebel and SAP.

Sun GlassFish Mobility Platform is based entirely upon open standards, including the following:

- Java Platform, Mobile Edition (Java ME)
- Java Platform, Enterprise Edition (Java EE)
- The dominant industry standard OMA DS, formerly known as SyncML. The specifications for Open Mobile Alliance Data Synchronization V1.1.2 and V1.2.1 are available at `http://www.openmobilealliance.org/Technical/release_program/ds_v112.aspx` and `http://www.openmobilealliance.org/Technical/release_program/ds_v12.aspx`.

Who Should Use This Book

This guide is intended for customers, developers, and administrators who will use MEP.

Before You Read This Book

This book has no prerequisites.

Sun GlassFish Mobility Platform Documentation

The Sun GlassFish Mobility Platform 1.1 documentation set will be available at `http://docs.sun.com/coll/1918.1`. To learn about Sun GlassFish Mobility Platform, refer to the books listed in the following table.

TABLE P–1 Books in the Sun GlassFish Mobility Platform Documentation Set

Book Title	Description
Sun GlassFish Mobility Platform 1.1 Release Notes	Late-breaking information about the software and the documentation. Includes a comprehensive summary of the supported hardware, operating systems, application server, Java^{TM} Development Kit (JDK^{TM}), databases, and EIS/EAI systems.
Sun GlassFish Mobility Platform 1.1 Architectural Overview	Introduction to the architecture of Sun GlassFish Mobility Platform.
Sun GlassFish Mobility Platform 1.1 Installation Guide	Installing the software and its components, and running a simple application to verify that installation succeeded.
Sun GlassFish Mobility Platform 1.1 Deployment Guide	Deployment of applications and application components to Sun GlassFish Mobility Platform.
Sun GlassFish Mobility Platform 1.1 Administration Guide	System administration for Sun GlassFish Mobility Platform, focusing on the use of the Sun GlassFish Mobility Platform Administration Console.

For up-to-the-minute information about Sun GlassFish Mobility Platform from the Sun GlassFish Mobility Platform technical team at Sun, see the Enterprise Mobility Blog at `http://blogs.sun.com/mobility/`.

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Feedback.

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–2 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–3 Shell Prompts

Shell	Prompt
C shell	`machine_name%`
C shell for superuser	`machine_name#`

TABLE P–3 Shell Prompts *(Continued)*

Shell	Prompt
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell for superuser	`#`

CHAPTER 1

Overview of Sun GlassFish Mobility Platform

This document describes the architecture of the Sun GlassFish Mobility Platform product offering. It also describes how the product's components communicate with each other to provide a seamless data synchronization experience.

Introduction

Sun GlassFish Mobility Platform is an open standards-based platform that enables access between enterprise applications and mobile devices. Sun GlassFish Mobility Platform provides reliable two-way data synchronization with security, device management, and offline access. Sun GlassFish Mobility Platform helps companies maximize return on investment on their existing IT infrastructure by reducing the cost of developing and provisioning client applications. It also empowers corporate mobile workers to be more productive by providing them with access to mission-critical data, from anywhere, at any time.

Sun GlassFish Mobility Platform includes the following features:

- Out-of-the-box connectivity to many enterprise application platforms
- Support for a wide variety of mobile devices
- Offline access to data when there is no network coverage
- Encryption and data fading/wiping extends enterprise application security to mobile devices
- Support for over-the-air provisioning of mobile clients
- Tools and templates that simplify the development and customization of mobile applications
- Based on open industry standards and robust and scalable Java technologies

Sun GlassFish Mobility Platform overcomes the following challenges:

- Enterprise customers use multiple devices. Sun GlassFish Mobility Platform supports a wide variety of mobile devices.
- Enterprise solutions are proprietary. Sun GlassFish Mobility Platform provides the ability to communicate with a variety of databases and EIS/EAI systems by offering a simple API that makes it easy to develop both client applications and Enterprise Connectors.
- Mobile users need to work in both a connected and a disconnected mode. Sun GlassFish Mobility Platform supports offline access to client data on the mobile device and the ability to connect to the EIS/EAI system as needed.

New Features in the Sun GlassFish Mobility Platform 1.1 Release

The following product features have been introduced in the Sun GlassFish Mobility Platform 1.1 release:

- A new AES security manager, which client applications can use instead of the default triple-DES security manager
- Support for RESTful Web Services (JAX-RS)
- Dynamic Data support through RESTful Web Services
- Salesforce sample application, including a new Salesforce connector
- Support for record merging
- Blackberry and Palm Windows over-the-air (OTA) client provisioning
- Support for logging using the GlassFish logging capability
- Maven archetypes to simplify connector development

Sun GlassFish Mobility Platform Deployment Scenarios

The Sun GlassFish Mobility Platform software can be deployed in three different scenarios:

- “Service Provider Managed Deployment Scenario” on page 11
- “Service Provider Hosted Deployment Scenario” on page 11
- “Enterprise Deployment Scenario” on page 12

Service Provider Managed Deployment Scenario

In a service provider managed deployment scenario, such as the one shown in Figure 1–1, the Gateway engine and its associated sync database form the Gateway tier in the service provider's network. The corporate network includes the Sun GlassFish Mobility Platform Enterprise tier components and the EIS/EAI system.

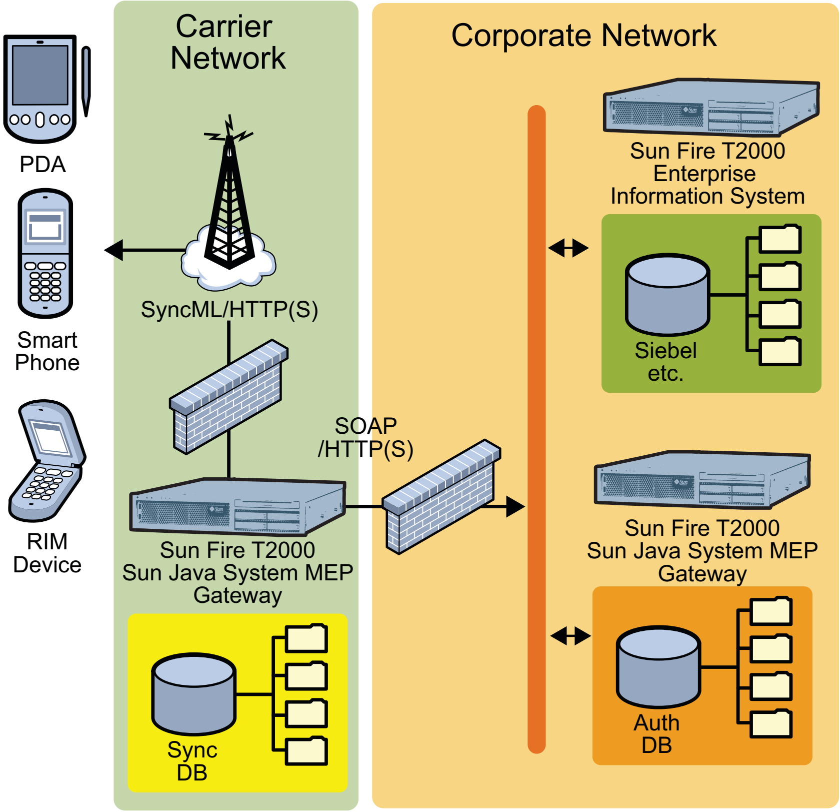

FIGURE 1–1 Service Provider Managed Deployment

Service Provider Hosted Deployment Scenario

In a service provider hosted deployment scenario, all of the Sun GlassFish Mobility Platform components and the database or EIS/EAI system are in the service provider's network.

Enterprise Deployment Scenario

In an enterprise deployment scenario, such as the one shown in Figure 1–2, all of the Sun GlassFish Mobility Platform components and the database or EIS/EAI system are in the corporate network.

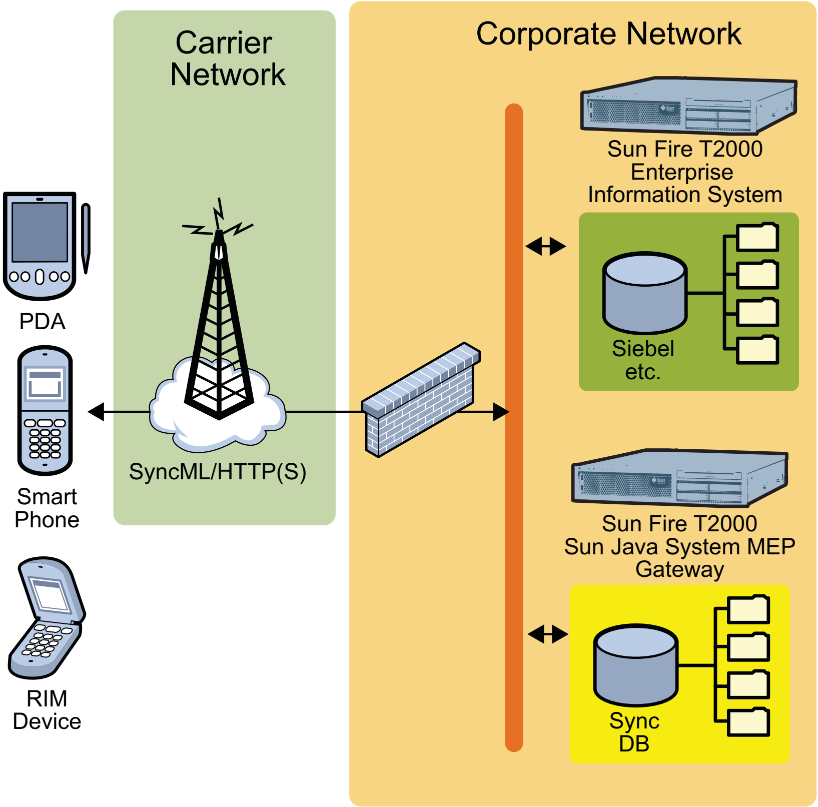

FIGURE 1–2 Enterprise Deployment Scenario

The service provider's network includes no Sun GlassFish Mobility Platform components. The corporate network, in essence, controls all of the Sun GlassFish Mobility Platform components as well as the database or EIS/EAI system. In the corporate network, the Sun GlassFish Mobility Platform components can be configured in a single-tier or two-tier configuration. These configurations are described in “Sun GlassFish Mobility Platform Architecture” on page 13.

Sun GlassFish Mobility Platform Architecture

Sun GlassFish Mobility Platform supports synchronization of enterprise data between Open Mobile Alliance Data Synchronization (OMA DS) enabled mobile phones and a database or EIS/EAI system.

The Sun GlassFish Mobility Platform architecture is based on open industry standards and is designed to operate a highly scalable, fault-tolerant environment tightly integrated with the existing infrastructure. Sun GlassFish Mobility Platform includes support for automatic failover and load-balancing, providing near-linear scalability. This architecture has been proven in carrier-grade deployments.

The major highlights of the Sun GlassFish Mobility Platform architecture are as follows.

Mobile Client Business Object API

The Mobile Client Business Object (MCBO) API provides an easy-to-use programming interface. It is provided as a Java ME library that the client application developer can use to develop a Sun GlassFish Mobility Platform Client application.

See *Sun Glassfish Mobility Platform 1.1 Developer's Guide for Client Applications* for information about using the MCBO API.

Sun GlassFish Mobility Platform Gateway

The Sun GlassFish Mobility Platform Gateway is the server component that interprets the incoming OMA DS messages and translates them into commands and data for Enterprise Connectors. The interface between the Gateway and the Enterprise Connectors is the Java Content Repository API.

Sync Database

The sync database contains the tables required by the Sun GlassFish Mobility Platform Gateway to store synchronization timestamps for client devices, mappings between client and server items, user information, configuration information, and synchronization message digest data.

Enterprise Connector Business Object API

The Enterprise Connector Business Object (ECBO) API provides an easy-to-use programming interface that makes it easy to build Enterprise Connectors.

See *Sun GlassFish Mobility Platform 1.1 Developer's Guide for Enterprise Connectors* for information about using the ECBO API.

Sun JCA Adapters

Sun JCA Adapters are Java EE Connector Architecture-based resource adapters that read and write data in the native format of specific databases or EIS/EAI systems. Sun GlassFish Mobility Platform includes adapters for SAP ERP, Siebel EAI, JDBC, and Oracle.

Figure 1–3 illustrates a single-tier Sun GlassFish Mobility Platform architecture. The client Sync App uses the APIs in the client library to communicate with the Gateway Engine, Enterprise Connector, and the Sun JCA Adapter. The adapter communicates with the EIS/EAI system.

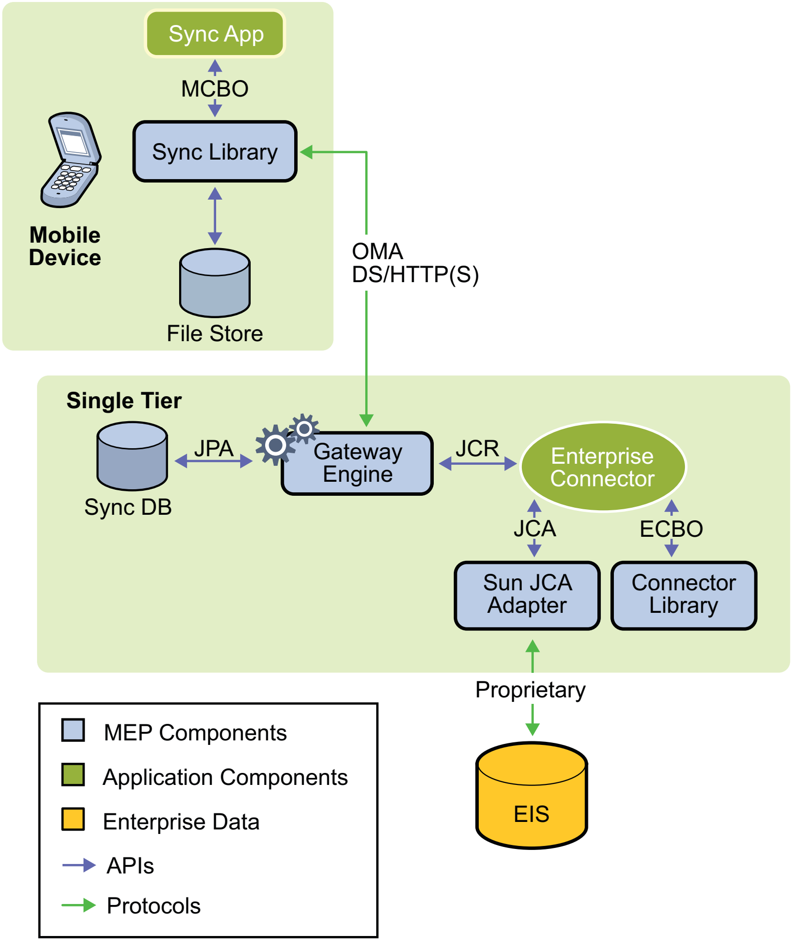

FIGURE 1–3 Single-Tier Sun GlassFish Mobility Platform Architecture

Figure 1–4 shows a two-tier architecture, consisting of a Gateway tier and an Enterprise tier. The Gateway tier includes the Gateway Engine and Web Service connector. The Enterprise tier includes the Web Service endpoint and Enterprise Connector, as well as the Sun JCA Adapter that communicates with the EIS/EAI system. The Web Service connector in the Gateway tier uses SOAP over HTTPS (or HTTP) to communicate with the Web Service endpoint in the Enterprise tier. The Enterprise Connector uses the ECBO APIs to communicate with the Connector Library and the JCA APIs to communicate with the Sun JCA adapter which, in turn, communicates with the EIS/EAI system.

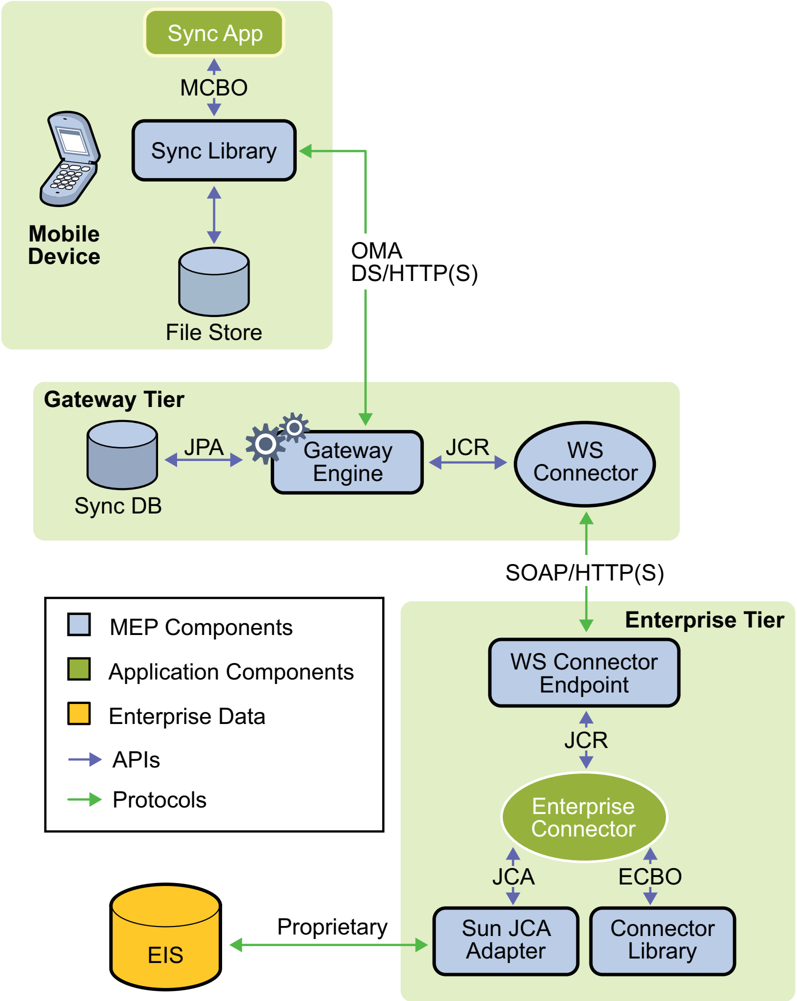

FIGURE 1–4 Two-Tier Sun GlassFish Mobility Platform Architecture

Synchronization Types

Sun GlassFish Mobility Platform supports the following types of client-initiated synchronizations:

- Both from server to client and from client to server:
  - "Two-way Sync (Fast Sync)" on page 17
  - "Slow Sync" on page 18
- From client to server only:

    - "One-way Sync from Client" on page 19
    - "Refresh Sync from Client" on page 20
- From server to client only:
    - "One-way Sync from Server" on page 20
    - "Refresh Sync from Server" on page 20

There are two ways to initiate the synchronization process: the user can trigger it manually by selecting the appropriate menu item in the client device's Java ME client, or the server can initiate the process (Push). As defined by the Open Mobile Alliance (OMA), for OMA DS Push, the server sends an SMS notification message to the device. If the client has been programmed to act on this message, it can synchronize with the server to retrieve any changes.

Both the client and the server store information about changes to their respective data stores since the last successful synchronization. When the next synchronization is performed, the client and server negotiate how the changes are resolved and propagated according to the type of synchronization being performed.

The following sections describe the synchronization types.

Two-way Sync (Fast Sync)

Two-way sync, also called fast sync, is the normal synchronization mode, in which the client and the server exchange modifications to the data that they have stored. An initial slow sync is used to populate the data on the client.

The client always initiates this exchange by sending client data modifications to the server. This is called a synchronization request.

The server processes the synchronization request, comparing and unifying the data from the client with the data in the server database by means of an Enterprise Connector.

Afterwards, the server sends the modified data to the client, which updates the local data store with the information from the server. Figure 1–5 illustrates this process.

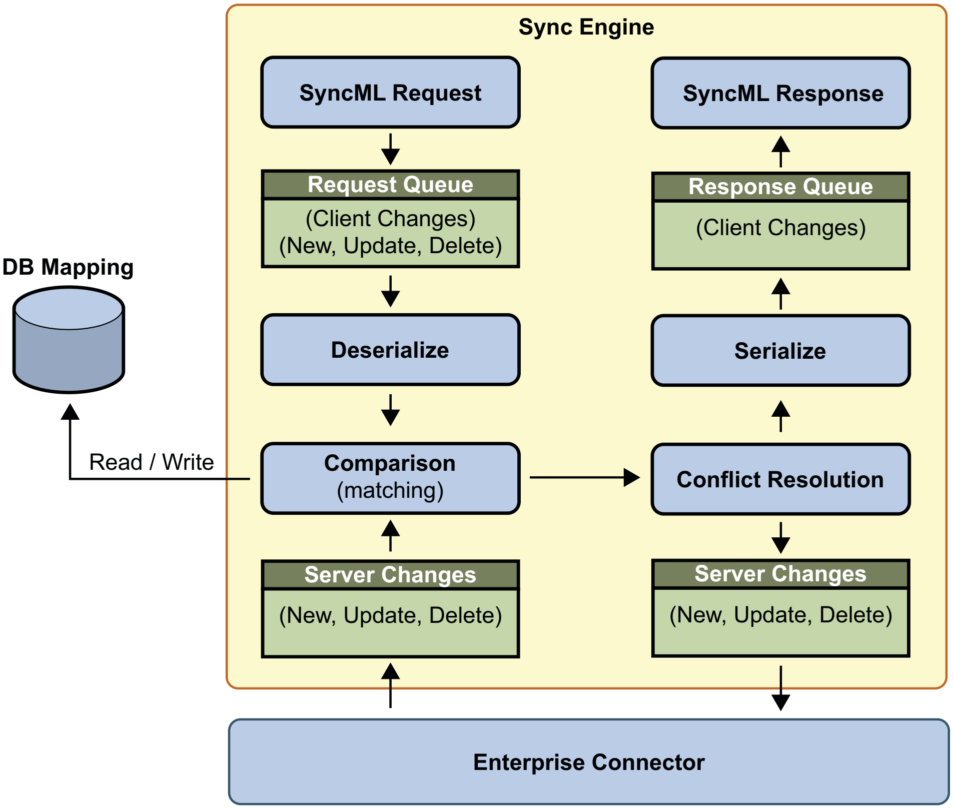

FIGURE 1–5 Two-way Sync

Slow Sync

Typically, the very first sync that a client performs is a slow sync. After that, the client performs fast syncs. If the data on the server or client is lost or corrupted, the next sync must be a slow sync.

In practice, a slow sync means that the client sends all its data to the server and the server compares its own data with that sent by the client. After the analysis, the server returns all the modification information to the client. In turn, the client returns the mapping information for all data items added by the server.

If the same record differs between the client and the server, the conflict will be resolved in favor of the server. If a record exists on the client but does not exist on the server, a slow sync will propagate the record to the server. If a record exists on the server but does not exist on the client, a slow sync will propagate the record to the client. If the client and the server were in sync before the slow sync, updates and deletions to records on the client are not propagated to the server. The client records will be overwritten by the corresponding record on the server, in this case.

Figure 1–6 illustrates this process.

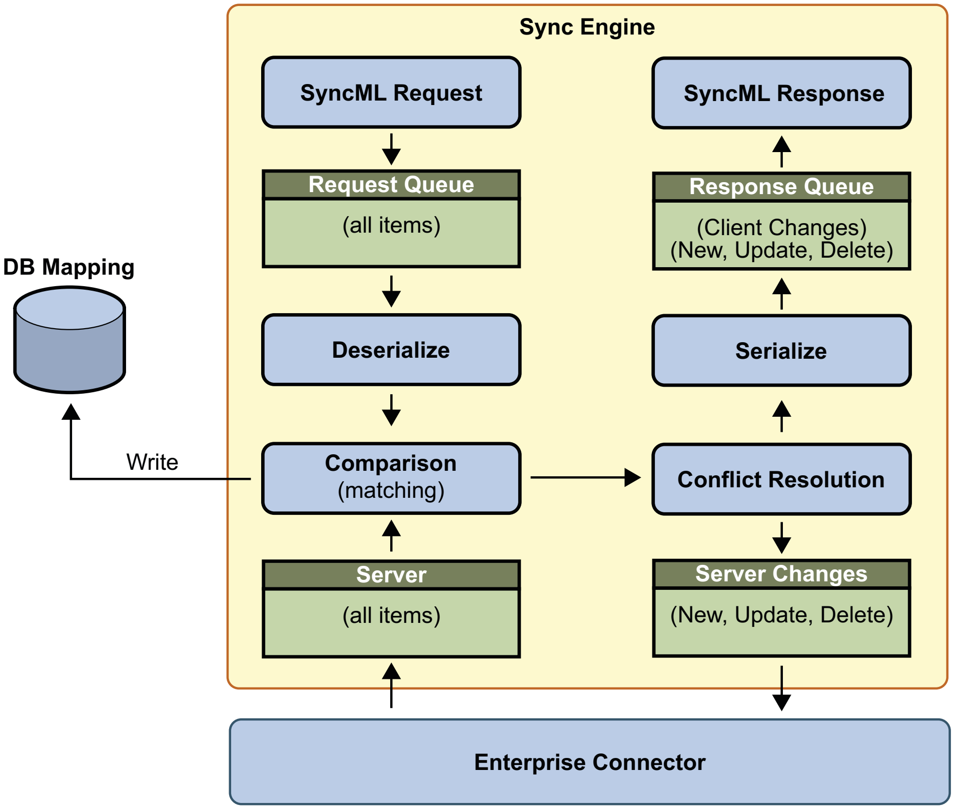

FIGURE 1–6 Slow Sync

One-way Sync from Client

This is one half of a two-way sync. In this mode, the client sends modifications of its data store to the server. The server updates its data store appropriately but does not send modifications of its data store to the client. After a one-way sync from client, the client and server data stores may not be in sync.

Refresh Sync from Client

In this mode, the client exports all its data to the server. The server is expected to replace all its data with the data sent by the client.

Note – Use this synchronization type with caution.

One-way Sync from Server

This is the other half of a two-way sync. In this mode, the server sends modifications of its data store to the client. The client updates its data store appropriately but does not send modifications of its data store to the server. After a one-way sync from server, the client and server data stores may not be in sync.

Refresh Sync from Server

In this mode, the server exports all its data from a database to the client. The client is expected to replace all its data with the data sent by the server.

Transport Layer

The transport layer for data synchronization between server and client can be HTTP or HTTPS.

- The HTTP protocol is used for insecure communication between the device and the Enterprise Server, and for accessing the administration portal.
- The HTTPS protocol is used for a secure connection between the device and the Enterprise Server.

Sun GlassFish Mobility Platform Security and Authentication

Sun makes every effort to ensure secure operation of Sun GlassFish Mobility Platform, which was designed with security in mind. Sun GlassFish Mobility Platform uses MD5 to generate a hash used as the symmetric key for 3DES to encrypt client data stored locally. When that data is sent over the wire from the client device, it is decrypted and sent using SSL/HTTPS. User data is at no time exposed to prying eyes. For security reasons, Sun GlassFish Mobility Platform does not duplicate the user's data to a local database, but only metadata required during the synchronization process.

Sun GlassFish Mobility Platform supports both client-side and server-side security:

- "Sun GlassFish Mobility Platform Client Security" on page 21
- "Sun GlassFish Mobility Platform Server Security" on page 21

Sun GlassFish Mobility Platform Client Security

Sun GlassFish Mobility Platform client security includes the following features:

- A simple PIN-based form of authentication
- A means to secure data at rest on the mobile device (data encryption)
- A means to securely synchronize with the Gateway Engine on the server (transport-layer security)
- A mechanism to destroy business data (data destruction)
- A means to prevent the client device from synchronizing (lockout)
- A means to remotely destroy all of the data on the device (poison pill)
- A means to notify the application that a certain quiet period has elapsed (data fading)
- An API that allows developers to replace the Sun GlassFish Mobility Platform default security manager implementation with their own

For details, see Chapter 3, "Client Security Architecture," in *Sun GlassFish Mobility Platform 1.1 Developer's Guide for Client Applications*.

Sun GlassFish Mobility Platform Server Security

Sun GlassFish Mobility Platform server security features include the following:

- TLS/HTTPS is used to provide authentication and encryption between the device and the Gateway Engine. The OMA DS protocol requires support for basic authentication and for verification of data integrity using a message digest created with MD5. The use of transport layer security (HTTPS) is assumed.
- In a two-tier Sun GlassFish Mobility Platform installation, TLS/HTTPS is used for communication between the Web Service connector and the Web Service endpoint.
- The Gateway Engine incorporates an Enterprise Server realm for user authentication. The default configuration uses a JDBC realm.

Client Provisioning

JSR 124, the Java EE Client Provisioning Specification, provides a framework and APIs for making client applications available on a Java EE server. Sun GlassFish Mobility Platform supports provisioning for client applications through its Administration Console. It provides a provisioning portal that offers the following features:

- **Content Management** - The provisioning portal manages the content repository and supports content versioning, a mechanism for administrators to upload applications, and the ability to search for applications.
- **Content Discovery** - The portal provides the user with a list of the available content and applications.
- **Application retrieval and installation** - Downloading the application is a two-part operation, handled by the application management system (AMS), the software in the device that manages the download, installation, execution, and removal of applications and other resources on the device. If an application description (in the form of a JAD file) exists, the AMS downloads it from the provisioning server's repository. Based on information found in the downloaded application descriptor, the AMS automatically downloads the application (the MIDlet suite JAR) from the repository. If required, the user is re-authenticated. If the application is retrieved successfully, installation is automatic.
- **Tracking** - Download status can be used to track the use of the application.

Sun GlassFish Mobility Platform Administration Console

Sun GlassFish Mobility Platform provides a web-based Administration Console that supports the following tasks:

- Sun GlassFish Mobility Platform product registration
- Addition, deletion, and modification of Gateway users
- Configuration of local and remote Enterprise Connectors
- Configuration of server push capabilities
- Management of client provisioning

The Administration Console provides different capabilities on each tier of a two-tier installation.

For details, see the *Sun GlassFish Mobility Platform 1.1 Administration Guide*.

NetBeans IDE Tooling

NetBeans IDE plugin modules simplify the development of applications for Sun GlassFish Mobility Platform as follows:

- For Enterprise Connector development, NetBeans modules can generate Object Type Definitions (OTDs), which are object models that support the Sun JCA Adapters for specific databases and EIS/EAI systems.
- Maven archetypes simply connector development
- For Java ME client application development, NetBeans IDE with the Mobility Pack includes full integration with the Java ME Wireless Toolkit and provides a choice of UI frameworks (including LCD UI and SVG).