TITAN

SOFTWARE

AN AFFILIATE OF TITAN CONSULTING

TITAN MOBILE APPLICATION FRAMEWORK

TITAN MOBILE APPLICATION FRAMEWORK

OVERVIEW

The SAP Mobile infrastructure consists of three major components:

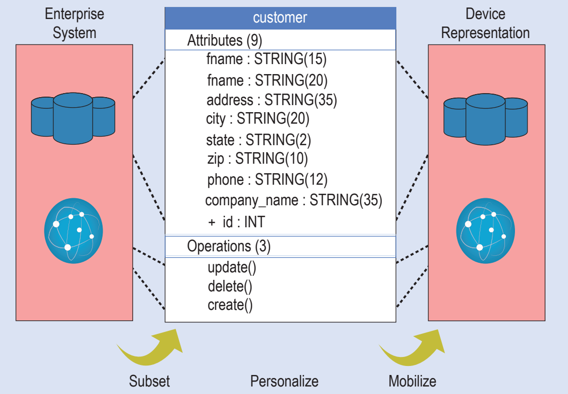
- SAP back-end system(s);
- SAP Unwired (Mobile) Platform server;
- Client devices
  - iPhone
  - iPad
  - Android device
  - Blackberry
  - Windows Mobile device

There are, in fact, several other components involved; however, the above is a simplified description that shows only the components involved within the Titan framework.

In a nutshell, data objects such as customers, vendors, materials, sales orders, delivery notes, invoices, etc. are stored on the SAP back-end systems. These data objects are spread across several database tables, and take up a lot of space.

Users of mobile devices will typically require access to view and/or modify only a small percentage of any given SAP data object. In some cases, the mobile user may wish to see components of more than one data object combined on a mobile device – for example, customer number, name and address from the customer object together with document number and value for all orders for that customer from the sales order object.

SAP uses an entity called a Mobile Business Object, or MBO, to provide the mobile device a view of back-end data objects. The challenge to mobile developers is to map the appropriate components of the back-end system objects to the MBO, both in the initial development of a mobile application and in subsequent enhancement stages of the application as user requirements change.

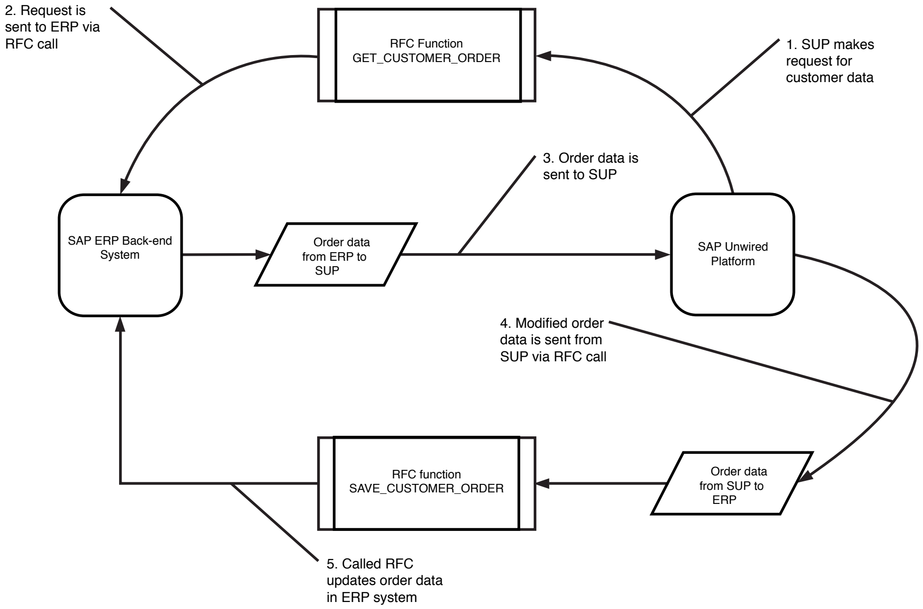


The diagram above shows a simple representation of this mapping process

This mapping process will typically be done via Remote Function Calls, or RFC's, from the SAP Unwired Platform (SUP) to the SAP back-end system as described below:

1. The SUP server makes a call to the **GET_CUSTOMER_ORDER RFC**-enabled function module.
2. The **GET_CUSTOMER_ORDER** function is executed on the SAP ERP system.
3. The **GET_CUSTOMER_ORDER** function reads the required data on the ERP system and transfers it to the SUP server.
4. After the mobile device has modified the order data, the SUP server makes a call to the **SAVE_CUSTOMER_ORDER RFC**-enabled function module, passing the updated order data in the function parameters.
5. The **SAVE_CUSTOMER_ORDER** function is executed on the ERP system; the modified order data is saved in the ERP database tables.

These steps are shown in the flow diagram below.



2. Request is sent to ERP via RFC call
RFC Function GET_CUSTOMER_ORDER
1. SUP makes request for customer data
3. Order data is sent to SUP
SAP ERP Back-end System
Order data from ERP to SUP
SAP Unwired Platform
4. Modified order data is sent from SUP via RFC call
RFC function SAVE_CUSTOMER_ORDER
Order data from SUP to ERP
5. Called RFC updates order data in ERP system

Implementing the mapping process requires a substantial amount of development. The MBOs must be developed and deployed to the SUP server, and the function modules that map the MBO components to the back-end system data objects must be written. Knowledge about the data objects on the back-end system is required to implement the function modules in an efficient manner.

Over time, as the end-user business evolves, changes will be required to the MBOs on the SUP server, and also to the back-end function modules. Development and maintenance of these components are not a trivial exercise.

PRODUCT DETAILS

The Titan Mobile Application Framework simplifies the development of the function modules that feed the MBOs on the SUP server.

A core set of extractor classes (for reading back-end data), and persistence classes (for writing back-end data) is provided to process source data from each underlying table of a business object within the ERP system (for example, Material Data Object, consisting of tables MARA, MARC, MARD, MAKT, etc.). Each table component is processed by a dedicated READ method in the extractor class. A set of Extractor Configuration tables determine (a) which data elements from the table that will be extracted by the READ method and (b) which target fields of the MBO that the extracted data will be mapped to.

Similarly, the Persistence Configuration tables drive the execution of the persistence class WRITE methods to write the data returned in the MBO to the ERP database tables.

A higher-level set of generation programs with corresponding Generator Configuration tables is provided to define the extractor and persistence objects to be used for each function module. The generation programs and matching Generator Configuration tables are used to generate the code for the function modules. As business requirements change, modifications are made to the Extractor and Persistence tables. After which the generator programs are executed to generate a new version of the function module(s).

Following common SAP practice, a set of user exits is provided in the form of BAdI classes to allow user-specific functionality to be added to the function modules.

A default set of configuration data is provided as a starting point. Customers may modify the provided objects (or copies of them) to suit their unique requirements. Specialized code can be added as necessary in the user exit BAdI classes.

It is also possible to create customer-specific objects using the supplied configuration as a template. These customer-specific objects could be created by customer technical staff members, or as a consulting project by Titan.

unleash the power of together®

www.titansoftware.mobi

GLOBAL HEADQUARTERS

3411 Preston Rd., Ste. C13-PMB 205 / Frisco, Texas 75034 / Main: 972.377.3500 / Fax: 972.692.7436