Chapter XXXIV
Distribution Patterns for Mobile Internet Applications

Roland Wagner
Johannes Kepler University Linz, Austria

Franz Gruber
RISC Software GmbH, Austria

Werner Hartmann
FAW Software Engineering GmbH, Austria

ABSTRACT

After the enormous success of the internet and mobile networks, the next upcoming boost for information technology will be the combination of both. But developing applications for this domain is challenging, because first, most mobile devices provide only small memory and processor footprints, prohibiting resource intensive code at client side and second, mobile networks offer only limited bandwidth, and the probability to connection losses is relatively high compared to wired networks. Selecting the appropriate software architecture in terms of distributing the functionality of the system between server and client device is crucial. Application distribution patterns, known from conventional system development, are analysed for their applicability for the mobile environment. After the more abstract analysis of the patterns, the IP multimedia subsystem (IMS) which is part of the current specification of 3G mobile networks is introduced and its support for different application distribution patterns is examined.

OVERVIEW

The success of mobile applications strongly depends on optimal utilization of client, server, and network resources. The distribution of the application functionality between client and server has strong impact on the grade of the resource utilization. Therefore, we present a

schema for application distribution patterns and analyze architectural locations where an application can be distributed and in this chapter we move our focus to the inherent problem of mobile applications to keep the data on the device and an the backend consistent.

With these distribution patterns, we will analyze several approaches for mobile Web access. In the last part of the chapter, we introduce an advanced architecture for representing mobile multimedia Web content: the IMS (IP multimedia system) with prerequisites and features as an example of a modern approach.

APPLICATION DISTRIBUTION PATTERNS

For the following we consider an application, whether it is mobile or not, consisting of three parts; The Presentation Layer, responsible for representing the visual parts of the application and doing the consummation of user input events. For distribution purposes, we divide the Presentation Layer in two sup-parts: The Dialog Representation, which is the visual painting and the reaction to events of the user and the Dialog Control which defines the sequence of the dialogs through the application.

The Business Logic Layer or the Application Kernel, responsible for the implementation of the business process, which means the origin and flow of data, which derives from the Backend Layer (Persistence Layer, Database Layer), which is responsible for retrieving and storing the data according to the requirements of the Business Logic Layer. Also for distribution analysis we divide the Backend Layer into the two subparts: The Database Access, which encapsulates the interface of the application programming language to the database (e.g., JDBC or ADO.NET) and the Database, which represents the database management system (DBMS) itself with tables, data, and stored procedures, etc.

Starting from this general architecture of an application, we want to derive a schema for making applications mobile, which means the separation of the whole application of parts of the application to a specified mobile device.

To be able to identify different software techniques to realize a device independent representation of an application, different architectural approaches have to be analyzed. First different levels where presentation and business logic can be separated are described. Finally, software techniques implementing the described design patterns are analyzed.

DISTRIBUTION LEVELS

This section describes possible distribution levels for every kind of client/server applications. The design of the client/server applications must provide functionality on the server. The client's functionality is mainly to display data ("thin client architecture"). This architectural design should ease distribution of new client versions. According to these needs, the design patterns "distributed presentation," "remote user interface" and "distributed application kernel" (taken from Renzel & Keller, 1997) are studied in more detail than "remote database" and "distributed database" pattern.

These patterns (see Figure 1) are interesting separating the representation from the business logic.

The remaining two design patterns "remote database" and "distributed database" in Figure 1 are applicable for realizing Web-based clients. A realization of these design patterns results in a "fat client" architecture, respectively the whole functionality of an application is located in the client.