

Architecture for Adaptive Mobile Applications

Haeng-Kon Kim

*School of IT Engineering, Catholic University of Daegu,
Kyungsan-si, Kyungbuk, 712-702, Korea
hangkon@cu.ac.kr*

Mobile applications can perform a wider range of activities than desktop applications and make use of information about the user's environment to provide novel capabilities. From a technological perspective, 'mobility' has shifted the global conception of computing infrastructure from static, homogenous, powerful desktop computers towards highly dynamic, heterogeneous, resource-constrained handheld and wearable devices. This new computing context demands entirely new software architectural paradigms to address the challenges of mobile software development, to be particularly designed for the nature of mobile devices and wireless networks, and take advantage of the opportunities afforded by mobile systems.

Latest research has rapidly advanced the state-of-the-art in architectures for mobile software and systems. Researchers have established the technical basis for mobile design by creating formal models of and engineering processes for mobile software. This particular software is designed to migrate across hardware devices at runtime and execute on mobile hardware. The principles of software architecture are intrinsic to the development environments and runtime platforms that support mobile software models and processes. We have characterized the mobile architecture by classifying likely applications and identified common properties; unlike previous research we approach the aspects of adaptability, flexibility and mobility from a wider perspective.

For the sake of research, we developed various pieces of the overall architecture and collected some preliminary experience with adaptive mobile applications, then we explain the core concepts and open research problems at the cross-section of the fields of software architecture and mobility. Moreover we analyze the possible impact and application of this analysis to build mobile systems tailored to mobile environments and present our architecture for such adaptive mobile applications.

Related Works

1.1.Mobile Applications Design

The key software architectural abstractions are components, connectors, their interfaces, configurations, and constraints on system structure, behavior, composition, and interaction. Architectural styles are essentially named sets of such constraints: client– server, peer to-peer (p2p), publish-subscribe (pub-sub), event notification, and so on. It is not readily obvious what level of support for mobility is yielded by, say, a given architectural style such as p2p or pub-sub. These traditional styles provide various design guidelines that can be useful in mobile applications. The guidelines include:

- ✓ Component decoupling, whereby the practical units of mobility are delimited – most architectural styles adhere to this guideline;
- ✓ Avoiding shared memory, whereby potential side effects of computations are limited and interaction “back doors” eliminated
- ✓ Styles such as pub-sub and event notification adhere to this guideline;
- ✓ Insulating components from execution context, whereby a single component may be effectively redeployed onto a number of sites
- ✓ Implicit invocation, whereby no component relies directly on the geographical presence or location of another
- ✓ Asynchronous interaction, whereby no component relies on the temporal presence of another
- ✓ Stateless components, whereby the migration process and system consistency assurance are greatly simplified – clients in some client–server applications are examples of stateless components
- ✓ Stateless interactions, whereby each interaction is self-contained and handled independently of the service requester’s or provider’s location.

1.2.Architecture Definition

As stated earlier, the architecture is a new programming paradigm that can ensure software reuse. Architecture is a template for working programs since it is a set of classes and interactions among the classes that embodies an abstract design for solutions to a family of related problems in one domain [9, 12]. Architectures are not simply collections of classes but also wired-in interactions between the classes that provide an infrastructure and architecture for the developers. To provide an infrastructure, we define the template classes that are the common and general classes to a group of related applications, and define the hot spots to a specific application. These hot spots are redefined or extended by the specific application developer later. It can be said that applications are successful when they satisfy the requirements of the customer. However, a

framework can be said to be successful when it is easily customizable for several different requirements. Therefore, designing an architecture is a little different from designing an application. We need more general and stronger abstract design concepts for designing an architecture.

2.2. Architecture Description

Figure 1 illustrates an overview of an architecture for Mobile applications development that involves two main participants: the mobile client environment and the backend server/mobile Web service. This architecture represents a logical structure and does not necessarily reflect a physical architecture. For example, the backend server may actually be distributed across several machines. In a large implementation, separate machines are used to run specific components of the architecture. A description of each module of our architecture presented in Figure 1 as overview of architecture for mobile applications is provided hereafter.

A. Backend Server

The backend server consists of eight components/modules that are: the enterprise web applications, business logic, content management system, workflow and profile management, user management, document repository, Web services, and database.

- **Enterprise Web Applications:** integrate functionality from remote Servers that might host Web services, portlets, *etc.* The backend server might support the J2EE standard Management System.
- **Business Logic:** represents the core of the mobile application that exposes functionalities. The business logic can be deployed on the backend server and used remotely by the mobile application to reduce the load due to limited resource available on the mobile devices. In other situation, a subset or the whole of the business logic can also be deployed on the mobile device.
- **Content Management System:** is a system (set of tools) used to manage different contents (data, multimedia information, and forums. These include for example, importing, creating, updating, and removing content to/from the mobile application.
- **Workflow and Profile Management:** manages the flow of data between the mobile application and the backend server and adapts the mobile application features and interface to different users' profiles.
- **User Management:** allows management of different types/classes of users with different levels of services and/or permission.
- **Database:** is used to store metadata about objects and content.
- **Document Repository:** is used to store different types of documents. The repository keeps track of all the updates performed by different users and provides version control capabilities.
- **Web Services:** are web components that can be deployed on backend server and can be invoked/consumed by the mobile application.

B. Mobile Device Client Environment:

The client environment consists of six components/modules that support the mobile client environment; these are mainly the business logic, secure layer interface, mobile application API, user interface, data storage, and the browser.

- **Business Logic:** as stated above, there are situations where it is suitable to deploy some of the business logic on the mobile device itself. For example, business logic required for offline transactions.
- **Secure Layer Interfaces:** handles secure user authentication for the Mobile environment using a protocol such as SSL, and provides different interfaces for different user's profile (e.g. admin, normal user, etc.).
- **Mobile Application APIs:** is a set of libraries and application programming interfaces (API) that implement a set of utilities that might be required by the mobile application (e.g.).
- **User Interface:** this is the graphical user interface components that mobile clients use to access different services. User interfaces includes mainly widgets, navigation facilities, and search tools.
- **Data Storage:** this is the place where persistent data is stored. This might include transactions data, users' personal information, profiles, and preferences.
- **Browser:** these consist of *minibrowsers* that are designed for use on mobile devices and optimized to display web content most effectively on small screens. The architecture components' provide a rich, drag-and-drop development environment for development of mobile applications. This environment supports developers in providing efficient and secure mobile applications. Efficiency, in terms of used resources, is highly required for such applications due to limited available resources in mobile devices and the drawbacks of wireless networks in terms of low bandwidth and high latency. Security is of prime importance whenever wireless and open networks are being used as a medium for data exchange. Figure 2 shows the overview of mobile device client application.