Architecture Approach for Mobile Service Security

Younky Chung

*Department of Computer Engineering, Kyungil University, Republic of Korea
ykchung@kiu.ac.kr

Abstract

Mobile computing is a relatively new field. While the challenges arising from mobility and the limitations of the portable devices are relatively well understood, there is no consensus yet as to what should be done to address these challenges. A comprehensive solution has to address many different aspects, such as the issue of dynamically changing bandwidth, the power, computational, and other limitations of the portable devices, or the varying availability of services in different environments. A Mobile service is programmable application logic accessible using standard Internet protocols.

In this paper, we present our architecture for such adaptive mobile service security applications. We motivated the architecture by classifying likely mobile applications and identified common properties. The architecture intends to be more general than previous work with respect to adaptability, flexibility, and user mobility. We developed various pieces of the overall architecture and collected some preliminary experience with adaptive mobile applications. We give an overview of the intersection of the areas of software architecture and mobile applications. We consider the mobile applications, which represent the computing functionality designed to migrate across hardware devices at runtime and execute on mobile hardware platforms, and the mobile systems, which are computing applications that include mobile software and hardware elements.

Keywords: Mobile Service Architecture, SOA (Service Oriented Architecture), Mobile Service Frameworks, Mobile Service Security, Component Based Development, Component Architecture Model

1. Introduction

Mobile services combine the best aspects of component based development and the web. Like components, Mobile services represent black-box functionality that can be reused without knowing how the service is implemented. Unlike current component technologies, Mobile services are not accessed via object-model-specific protocol.

Service-Oriented Architecture (SOA) is a new software development paradigm introduced by recent research surrounding mobile services. SOA features [1, 2]

- Uniform data marshaling protocol (via Simple Object Access Protocol: SOAP and eXtensible Markup Language: XML),
- Standard communication infrastructure (via Mobile service Definition Language: WSDL and HyperText Transfer Protocol: HTTP), and
- Service query facility (via Universal Description, Discovery, and Integration: UDDI).

This architecture includes a service area tier to offer necessary roles and functionality for the consolidation of multiple services into a single composite service based on component. The component consists of security facade component and backside component. It also provides a tier for service operations management that can be used to monitor the correctness and overall functionality of aggregated/orchestrated services. Finally, we explained how component architecture model of mobile service-security can be used to
implement the security mobile service. The mobile applications support a much wider range of activities than desktop applications and leverage information about the user’s environment to provide novel capabilities. From a technology perspective, mobility shifts the global computing infrastructure from static, homogenous, powerful desktop computers to highly dynamic, heterogeneous, resource-constrained handheld and wearable computers. This new computing context demands entirely new software architectural paradigms that address the challenges of mobile software development, are specialized for the nature of mobile devices and wireless networks, and take advantage of the opportunities afforded by mobile systems. Mobile software is computing functionality designed to migrate across hardware devices at runtime and execute on mobile hardware platforms. The principles of software architecture are intrinsic to the development environments and runtime platforms that support mobile software models and processes. We motivated the architecture by classifying likely mobile applications and identified common properties. The architecture intends to be more general than previous work with respect to adaptability, flexibility, and user mobility.

In this paper, we developed various pieces of the overall architecture and collected some preliminary experience with adaptive mobile applications. We explain the core concepts and open research problems at the cross-section of the fields of software architecture and mobility. In parallel with (and supported by) these achievements, research activity that leverages this work to build mobile systems has expanded. Mobile systems are computing applications that include mobile software and hardware elements. These applications are characterized by customized software architectures that are designed for and intrinsically facilitate mobility. We present our architecture for such adaptive mobile applications. We motivated the architecture by classifying likely mobile applications and identified common properties. The architecture intends to be more general than previous work with respect to adaptability, flexibility, and user mobility. We developed various pieces of the overall architecture and collected some preliminary experience with adaptive mobile applications. We give an overview of the intersection of the areas of software architecture and mobile applications. We consider the mobile applications, which represent the computing functionality designed to migrate across hardware devices at runtime and execute on mobile hardware platforms, and the mobile systems, which are computing applications that include mobile software and hardware elements.

2. Related Works

2.1. 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 do provide many design guidelines that can prove to be useful in mobile applications. The guidelines include:

- Component decoupling, whereby the practical units of mobility are delimited by the 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 requestor’s or provider’s location.

2.2. Framework Description

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

A. Backend Server

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

- **Enterprise Mobile Applications:** integrate functionality from remote Servers that might host Mobile 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.

• Mobile services: are mobile 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 mini browsers that are designed for use on mobile devices and optimized to display mobile content most effectively on small screens. The framework 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.

2.3. Service Oriented Architecture

A Mobile service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-process able format (specifically MSDL). Other systems interact with the Mobile service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards[3].

The Mobile architecture and the Mobile services Architecture (MSA) are instances of a SOA. The Figure 2 illustrates the basic Mobile Services architecture, in which a service requestor and service provider interact based on the service’s description information published by the provider and discovered by the requester through some form of discovery agency. Service requesters and providers interact by exchanging messages. The
Mobile services architecture is based upon the interactions between three roles: service provider, service registry and service requestor. The interactions involve publish, find and bind operations.

Mobile service in the basic architecture can take on one or all of the following roles:

- **Service requester**: requests the execution of a Mobile service
- **Service provider**: processes a Mobile service request
- **Discovery agency**: agency through which a Mobile service description is published and made discoverable

**Figure 2. Service Oriented Architecture**

### 2.4. CBD Process

Development of e-business systems involves collaborative work of several different types of specialist with different areas of expertise; for example, business process consultants, software architects, legacy specialists, graphic designers and server engineers. e-Business process improvement provides the right business context for CBD, as shown in Figure 3. Of particular importance for transitioning to e-business using CBD are the overall e-business improvement plan, which provides business direction for architecture planning and the business models, which focus on understanding specific processes requiring e-business solutions. While the business improvement plan ideally encompasses the entire enterprise, the overall vision may be developed incrementally, leading to a succession of more narrowly focused action plans [4, 5].

The process is evolutionary. Results from software projects are fed back to e-business process improvement for reassessment in the light of experience with e-business. Similarly components are assessed with respect to architecture planning, in a process progressive refinement. Architecture planning must cover the overall business case for CBD and high-level enterprise component architecture, within the scope set by the business improvement plan. This includes the provisioning strategy (policy on build, reuse or buy decisions) and influences the choice of funding model. The enterprise component architecture provides a "big-picture" for projects to work to. This is particularly important on CBD projects as they typically involve incremental development and parallel work performed by relatively independent teams [6, 7].

**Figure 3. Component Based Development Process**
3. Deriving Architecture Model of Mobile Service-Security

3.1. Logical View Architecture

We propose component architecture of mobile service-security that consists of 4 layers. This logical view architecture that we have just looked at is a pre-requisite to thinking about the type of architecture required and horizons of interest, responsibility and integrity. For SOA there are three important architectural perspectives as shown in Figure 4. These contents are as following.

![Figure 4. Logical View Architecture of Mobile Services](image)

Business area layer contain user system, external system and presentation service. Service area layer offers mobile service through business logic. Also, there communicate response that handle to business area layer. The facade component and Mobile script is located at this layer. Operation area layer consists of component and package component. The component is common component or Commercial Off The Shelf(COTS) component that developed Component Based Development(CBD) methodology. Support area layer located database or component to offer requested data in operation area layer.

3.2. Component Meta Model for Mobile Service-Security

It can identify elements to construction mobile service and locate in logical layer through analysis relation between them. Also, mobile service in implementation viewpoint is possible that define relation between layers. Figure 5 show component meta model for mobile service-security.

Mobile service is consisted of single Core Facade component (Service oriented component) and many Backside components. Backside component provides mobile service through single Core Facade component. Facade component is specified by SOAP and MSDL, do so that mobile service may be available. Flow façade component has function that control flowing between mobile service. Security façade component achieves security function of authentication, authorization, encryption, log management etc.

![Figure 5. Component Meta Model for Mobile Service Security](image)
3.3 Component Architecture Model of Mobile service-Security

Component Architecture Model of Mobile service-Security constructed based on logical view architecture as figure 6. The structure of the inside is mapping with logical hierarchy structure. Business area contains external system and user system. A service area offer mobile service, located flow facade component, security façade component and core façade component. Operation area located backside component which offer legacy component and business logic. Support area is an outside resource which the system uses. Outside resources defines by everything (system, library, module, control and so on) in system outside.

Mobile infrastructure is located between a service area and business area. The system can be composed of the application of the low or ideal. One application is composed of Facade and Backside component and offers Mobile service. System offer Mobile service through core façade component or security façade component, when system consists of one application. Flow facade component offer integration and flow control between applications when inside of the system was composed of the application over 2.

4. Modeling and Implement of MSS

A brief demonstration of the prototype in this section illustrates how a component architecture model of mobile service-security could possible be implemented. We are currently working to build a complete prototype of our architecture and component model using UML. The product is a Mobile Service Security that provides history management service of stored data and privilege management service for each user.

Figure 6. Component Architecture Model of Mobile Service-Security

We present the component architecture model of mobile service-security through the case of a user of Mobile Service-Security. Figure 7 presents context diagram which expresses in system view. There are MSS (MOBILE SERVICE-SECURITY) that is mobile service between client and server. In Figures 8 and 9, workflow model, use case model and init architecture model are showed. Moreover, Figure 10 shows the Service Oriented Component Architecture of MSS(MOBILE SERVICE-SECURITY).

Figure 7. Context Diagram of Mobile Service-Security
We show the action of Mobile service confirm carrying with a service request. The result shows mobile browser because of mobile service based on web. It is shown in Figure 11 that an execution example of secure access mobile service by data operation service. Figure 12 shows the response message that is result of service call test and service request through interface of mobile service. It displays the result of the certification with a user certification screen for the utility of Mobile service. Figure 13 shows the Mobile Service Description Language (MSDL) specification of secure access mobile service that refine of design model.
5. Conclusion

A model base of mobile service-security has been designed using the component architecture model of Mobile service-Security. The model base uses UML constructs such as Class Diagrams, State Charts, Use Case Diagrams, Activity Diagrams and Collaboration Diagrams to architecture model and continuous systems which are the constituent sub-systems of Mobile Service-Security. The architecture model base is used as a template to implement the mobile service-security. The implementation consists of packages and processes which are executed concurrently using the technique of mobile service-security. Research in progress currently focuses on the refinement of the mobile service-security to include interactive error-checking as well as a diversification of the types of continuous models which can be input into the MSS (Mobile Service Security). The final objective is to endow the mobile service with security, audit, certification and authentication functionality.

Acknowledgement

This study was supported by the Kyungil University Grant.

References


Author

Younky Chung
Research interests: Network management, wireless network, multimedia communication, and ubiquitous sensor network.