Toward an Efficient Framework for Designing, Developing, and Using Secure Mobile Applications

Mohamed Adel Serhani, Abdelghani Benharref, Rachida Dssouli, and Rabeb Mizouni

Abstract—Nowadays, people are going more and more mobile, both in terms of devices and associated applications. Moreover, services that these devices are offering are getting wider and much more complex. Even though actual handheld devices have considerable computing power, their contexts of utilization are different. These contexts are affected by the availability of connection, high latency of wireless networks, battery life, size of the screen, on-screen or hard keyboard, etc. Consequently, development of mobile applications and their associated mobile Web services, if any, should follow a concise methodology so they will provide a high Quality of Service. The aim of this paper is to highlight and discuss main issues to consider when developing mobile applications and mobile Web services and then propose a framework that leads developers through different steps and modules toward development of efficient and secure mobile applications. First, different challenges in developing such applications are elicited and deeply discussed. Second, a development framework is presented with different modules addressing each of these challenges. Third, the paper presents an example of a mobile application, Eivom Cinema Guide, which benefits from following our development framework.

Keywords—Mobile applications, development of mobile applications, efficient mobile application, secure mobile application.

I. INTRODUCTION

THE way people communicate has drastically changed since the early days of communication. Users' requirements drove the revolution from the earlier fire signals, optical and electrical telegraphs, home phones, to wireless communications. Nowadays, wireless networks and devices are supposed to convey far more than traditional voice conversations. In fact, the cheapest cell phone is likely to have a built-in camera allowing its users to take pictures and then share them with friends and family members. Multimedia players are another basic feature of today's cell phones and mobile gadgets that let users download multimedia files from remote servers and play them afterwards. These new features are empowered by a modern and rich set of mobile

Mohammed Adel Serhani is with the College of Information Technology at U.A.E University, UAE (phone: 971-3-713-5572; fax: 971-3-7672018; e-mail: serhanim@uaeu.ac.ae).

Abdelghani Benharref is with the College of Engineering and Computer Science at Abu Dhabi University, UAE (phone: 971-2-501-5762, fax 971-2-5860182, e-mail: abdelghani.benharref@adu.ac.ae).

Rachida Dssouli is with Concordia University, Montreal, Canada (phone: 1-514-848-2424 Ext 4162, e-mail: dssouli@ciise.concordia.ca).

Rabeb Mizouni is with the College of Information Technology at U.A.E University, UAE (phone: 971-3-7135552 9; fax 971-3-7672018; e-mail: mizouni@uaeu.ac.ae).

Mobile clients are leveraging the barriers, again, higher when they foresee to have more advanced applications on their mobile phones and devices ([2], [3], [4], [5]). They are no longer limited to pure telephoning devices or even standalone applications (local address book, local calendar, etc). While surfing the web and exchanging emails, mobile clients, mainly business users, are expecting to have miniaturized versions of applications they actually have on their laptops and desktops. After all, most users from outside the technical IT fields have the perception that "a mobile phone or device is just a small computer"; which is true to some extent but not at a broader look. A mobile application is any application running on a mobile device other than a notebook/laptop. A sales' manager, for instance, wants to have both view and modify accesses to previous and recent orders even though she/he is on the road with no computer access.

Service providers see these expectations as new promising and high standing business opportunities. With the almost-saturated traditional mobile phones' market, mobile applications are to generate more benefits for networks' carriers and service providers. A mobile Web service is any Web service that might be invoked by a mobile application.

Developing mobile applications and their corresponding mobile Web services is quite different from developing traditional Web services and desktop-oriented applications. Despite the limited available resources (CPU, memory, battery, etc) and communication aspects in wireless environment (connection availability, security, etc), mobile devices are nowadays running a variety of proprietary platforms and incompatible operating systems. This makes mobile applications highly platform-dependent, that is, they just can be used on the platform for which they are developed. Migration to other platforms usually requires code modification, tuning and adaptation, recompilation, and redeployment.

Two main platforms are emerging to host platform-independent mobile applications: 1) browser-based platform and 2) java-based platform. The browser-based approach relies on the availability of advanced browsers on most platforms (e.g. Opera Mobile [6], Internet Explorer Mobile [7], Safari Mobile [8]). Those browsers offer a standard set of features comprising the support for scripting languages such as Asynchronous JavaScript and XML (AJAX) [9]. The Java approach is boosted up by the availability of Java Micro Edition (ME) virtual machine [10] on most mobile devices. Depending on the used platform, solutions to overcome

mobile applications issues presented above might be tackled, adapted, and customized.

The paper is organized as follows. Section II reviews existing work on mobile applications and mobile Web services. Section III describes and analyses the main requirements and challenges of mobile applications development. Section IV proposes a framework for mobile applications and describes its components. Section V presents and discusses a case study of implementing a mobile application that adopted our development framework. The last section highlights the main contributions and points to ongoing and potential future work.

II. RELATED WORK

At the early days of mobile technologies, the typical utilization of handheld devices was mainly basic personal communication purposes (e.g. initiating calls, exchanging messages, etc.). Nowadays, providers of these devices exploded the market with a large variety of mobile devices such as portable media players, PDAs, Smart phones, and Pocket PCs. Besides, with new adopted networking and telecommunication technologies such as WiFi and GPRS, the use of handheld devices becomes very popular and these devices are used for a wide range of services including internet navigation, emailing, location-based and context-aware services, and mobile learning.

State of the art shows that many mobile applications have been developed to enrich the appliance of these devices and to benefit from their features. Most of these applications are very sensitive to performance (Quality of Service), security, and availability. Examples of these applications use video streaming, image processing and mobile learning services (m-Learning).

Nowadays, very few mature works have been done on the development and management of mobile applications and mobile Web services. However, considering usability and quality of these applications will play a key element in using them. On the one hand, providers of these applications will need to guarantee a better usability of mobile applications and their corresponding mobile Web services to remain competitive and maximize the return on investment from their businesses. On the other hand, mobile clients will have the possibility to look for mobile applications and mobile Web services with higher usability, quality, and lower cost.

Achieving such requirements necessitates a concise development approach of both mobile Web services and mobile applications. State of the art in this issue still lack solid and proven development approaches.

In [11], authors propose an approach for developing mobile applications with separation of concerns: functional and adaptive behaviors in an ad-hoc network. The authors, however, did not study specific aspects of mobile applications as listed above. Authors in [12] propose a framework to ease the development of platform-dependent mobile applications based on J2ME. Joeng et al. in [13] proposes a mobile

application streaming service in which server-side infrastructure does most of the job and customizes screen display for the mobile device. This approach assumes a permanent availability of a connection to mobile Web services.

The difference between Web services and mobile Web services depend on the way mobile applications interact with these Web services. There are two main ways: 1) direct interactions and 2) interactions over gateways. In the first category, the mobile Web service must customize messages exchanged with mobile applications. The Web service might implement XML compression [14], encryption [15], and/or light versions of standard Web services technologies (e.g. kSOAP [16] instead of SOAP) in addition to their core functionalities. In the gateway approach ([17], [18]), extra features are implemented at the gateway and the (mobile) Web service may even not notice it is interacting with a mobile application.

Each of these related works tries to tackle a particular issue at once. Nevertheless, different issues are dependent and have mutual impact on each other and proceeding with an individual issue at a time usually misses this impact. For example, extensive use of compression of exchanged XML messages [14] might drastically consume the battery [19]. Same can be applied to XML security [15] including encryption, digital signature, and complicated key exchange protocols. All the issues should be studied together to get a broader view of challenges that developers of mobile applications and mobile Web services should carefully consider.

Main issues in developing mobile applications and mobile Web services are studied in the next section.

III. MOBILE APPLICATIONS AND MOBILE WEB SERVICES CHALLENGES

Before mobile applications can be adopted at a large scale and for heavy transactions, there are plenty of technical aspects that have to be addressed. Transaction-related data should "always" be accessible to mobile applications. This accessibility is mainly affected by three factors:

1. Availability of connections between mobile applications and mobile Web services,

2. Available bandwidth, and

3. Network latency.

When there is no connection, data should be stored locally using appropriate data persistency mechanisms. Later on, both offline transactions and data must be updated, committed, and/or rolled-back as soon as a connection is available. Limited bandwidth and high latency in wireless networks imply a set of decisions on how much data should be located on the device, how much data should travel backward-and-forward between the mobile application and the mobile Web service and how to optimize the transfer of this data. There are two extremes:

1. Put all data at the mobile side and synchronize it with backend warehouses whenever needed and/or possible, or

2. Put all data at backend warehouses and communicate it to mobile applications whenever requested.

The first approach is utterly impractical since resources on mobile devices are very limited. Efficiency of the second approach is downed by the bandwidth and latency properties of wireless networks. A better solution would be a compromise between these extremes in which some data will be on the mobile device and other will be requested from the mobile Web service. The main problem that rises in such approaches is how to decide which information to keep remote and which to bring to the mobile device. Decision criteria include available resources, cost of storage, cost of communication, significance and sensitivity of data, and cost and benefits of compression.

Furthermore, communication over wireless networks drives a wide range of security flaws: from user authentication, access lists, to data integrity. Mobile users must have enough confidence that their sensitive data is protected all along the way from their devices to mobile Web services. Moreover, they want to make sure that no one has access to specific data except authorized users with valid credentials.

While developing a mobile application and its mobile Web services, developers should carefully consider the following questions:

• How to present the User Interface (UI)? Mobile devices' screens are very small compared to the smallest laptop. This is aggravated when the mobile device provides a soft (on-screen) keyboard.

• How to support offline operations? Even though roaming is a very basic feature of actual wireless networks, there are situations where a connection is not available, possible, and/or desirable. A mobile user should be able to use a mobile application in the absence of a connection.

• How to deal with limited bandwidth? Even when connected, the bandwidth available in wireless networks is somehow limited when compared to wired networks (modern switched Ethernet, Optical Fiber, etc). The size of data exchanged between mobile applications and mobile Web services should be optimized.

• How to hide high and unpredictable latency? Wireless networks are known for their variable delays and connection drops. These bad properties should be made hided to the mobile user.

• How to ensure security? While transiting between mobile Web services and mobile devices or stored locally on the latter, data should be appropriately secured. Business information is very sensitive and its users should be confident that this information is safely moving back-and-forth. However, when implementing security policies on mobile devices, required resources by security-related operations should be minimized. Efficient security procedures, such as data encryption and digital signature, must be elaborated and optimized for mobile environments.

• How to provide advanced business logic? A mobile application might interact with many mobile Web services to complete a transaction. The business logic orchestrating different interactions should be smoothly defined and automatic composition patterns and flows should be made efficient. Moreover, developers have to decide on where to deploy this business logic: on the mobile device, mobile Web service, or the gateways (if any).

• How to ensure browser's compatibility? When designing browser-based mobile applications, and even though mobile browsers are offering somehow similar functionalities, some application tuning is required for different platform and quite old browsers (Internet Explorer Mobile 5 for example).

• How to decide on local or remote computation? As discussed above, resources in mobile devices are very limited (CPU, memory, etc) and their use should be optimized. This optimization might imply delegation of processing to mobile Web services rather than performing them locally on the device depending on the resulting overhead. Criteria and threshold to decide when to perform local or remote processing must be clearly specified.

• How to adapt mobile Web services? Web services are usually invoked from desktop applications where resources (CPU, RAM, Bandwidth, etc) are somehow abundant. Unless they are designed since the early beginning to take into account mobile applications requirements, mobile Web services should be adapted to reduce size of messages and required processing at mobile devices.

The next section presents our framework for developing mobile applications that addresses most of the challenges discussed above. It stretches different modules and components that characterize efficient and secure mobile applications.

IV. FRAMEWORK DESCRIPTION

Fig. 1 illustrates an overview of our framework for Mobile applications development that involves two main participants: the mobile client environment and the backend server/mobile Web 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 Fig. 1 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

(JSP, Servelet, Web services) or other Content 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 (see section IV.B).
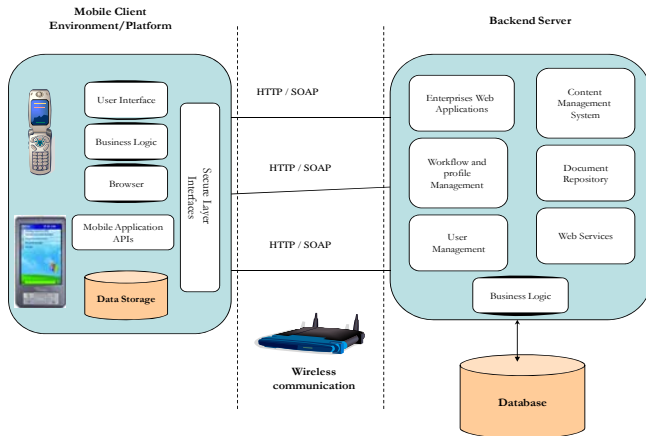

Fig. 1 Mobile Applications Development Framework

• **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 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 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. All aspects of security, including user authentication and encryption, are supported through a well-defined and efficient set of API easily accessible within the development environment.

V. CASE STUDY

A. Description

To experiment with some of the features of our framework, we developed a mobile application called "Eivom". Eivom is a cinema guide application that integrates data from different information systems of different cinemas through Web services technology to provide users with various services that engage them in a truly entertaining experience. The aim of this case study is to deliver an application that would integrate both mobile applications and web technologies together, and provide efficient mobile services to wide a range of users while maintaining good performance and assuring security of transactions. Fig. 2 presents the components of the mobile movie system.

a) *Backend Eivom Server:* an application server that receives requests from clients and cinemas, processes them and then responds. It also holds a database that maintains all system related data, receives queries from the application server and responds accordingly.

b) *Cinema Server:* an application server that receives clients' and Eivom's requests, processes them and responds

accordingly through the published Web service. It also holds a database server that maintains all cinema related data, receives queries from the application server and responds accordingly.

c) *Mobile Client:* a mobile application that enables users to access the system and use all its features. The client exchanges SOAP requests/responses with the application servers (Eivom server and the cinema servers).
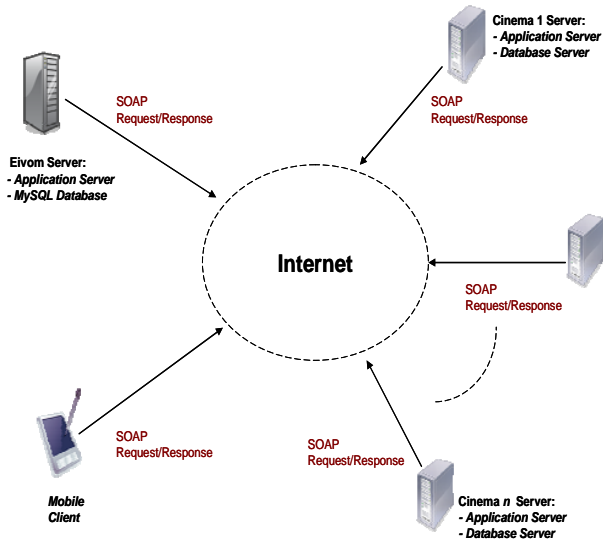

Fig. 2 Mobile Movie System

B. Application Features Appliance

Beside the functional features, a set of non-functional requirements was derived to insure the quality of the application as defined in section III. These requirements (design principals) concern the following:

- Reliability: by implementing error handling procedures.

- Security: by authenticating users and encrypting using secured transport protocol; HTTPS.

- Flexibility, by facilitating application's further enhancement and maintenance.

- Ease of use: by providing a simple user interface.

- Highly available.

- Scalability: by having the possibility and the ease of integrating other cinema services.

- Allow variety of users with diverse profiles, thus adapting the application (e.g. interface) to each user preferences.

- Offline operation: using cashing mechanism data used by some of the frequently used operations of Eivom are cashed at the mobile device.

- Local vs. remote computation: some of the operations of Eivom are executed locally and do not consume lot of resources, other are delegated to remote Web services thus executed remotely at the Web service environment.

- User interface design: the user interface of Eivom is designed to be user friendly. Some UI components are depicted in Fig. 3.

- Advanced business logic: by using Web services and composition of Web services, advanced business logic can be implemented based on the invocation of diverse Web services that cooperatively provide more complex business logic.

C. Implementation

We have used the following technologies to develop Eivom application: J2ME, J2EE, MySQL, KSOAP, and Web services technologies. Eivom is designed to run on HP iPAQ Pocket PC with CrE-Me virtual machine provided by NSIcom [20]. By using the suitable virtual machine that supports Swing library, the application will be pseudo-automatically adapted to run in any PDA or smart phone.

We have applied Model View Controller (MVC) methodology in building the architecture which separates core business logic from the presentation. MVC model allows multiple views to share the same enterprise data model, which makes supporting multiple clients easier to implement, test, and maintain.

Back-end architecture mainly consists of a J2EE application containing methods that are deployed as Web services. These methods access authorized databases using JDBC, process the required queries and respond to the client. We have implemented a couple of cinemas system prototypes and expose them as deployed Web services to simulate working scenarios of the system.

Cinemas' Web services retrieve all required information from their databases using JDBC and deliver it to clients such as movie schedule and movie premiers. Eivom Web services provide user with many functions such as authenticate user, create user profile and store user preferences so that when the user logs in to the application, he will be authenticated through Eivom web server.

The system front-end is the interface from which users can directly interact and execute the features of the system. It provides means for collecting input from the user, validating and processing it in such a way that it conforms to a specified structure that the back-end can handle. The Eivom front-end is a mobile Java application implemented using J2ME Personal Profile. J2ME Personal Profile is set of APIs used on top of Connected Device Configuration (CDC) such as PDAs and smart phones. To develop Eivom's front-end, we have worked with both profiles of J2ME which are MIDP and Personal Profile and we have decided to use Personal Profile as it provides richer GUI capabilities.

To consume the deployed Web services, HTTPS connection should be established and SOAP messages, which are based on the XML protocol, are sent to connect the back-end and the front-end. However, as Eivom front-end is a wireless client, J2ME lacks the standard XML API since they are too heavy for small devices. To overcome this limitation, we have used

kSOAP project [16] which is an open source SOAP API suitable for J2ME to implement wireless Web services client.

We have implemented most of the system features such as search cities, search movies, user bookmarks, user cinemas, movie premier, stream movie trailer, movie details, movie schedule time, offline access to the application, buy tickets, tell friend about movie and rate movie. Fig. 3 shows examples of some implemented features.

Fig. 3 Search cities and movie information snapshots

We will show hereafter how the proposed framework has been used to develop Eivom application and solved most of the mobile application development challenges stated previously. We will show as example how the framework supported in two issues, namely (1) implementing advanced business logic and (2) compromising local/remote computation.

• How Eivom provide advanced business logic? Eivom has been fully implemented around mobile Web services that are invoked to expose the application functionalities. These Web services are deployed on remote servers and invoked using SOAP messages. Web services are composed to provide more complete transaction and richer functionalities. For instance, *UserPreferences* and *UserProfile* Web services are composed to provide two features in one operation.

• How Eivom decides on local or remote computation? Due to limited resource on the mobile device, Eivom optimizes the resource usage by delegating processing to remote Web services depending on the resulting overhead. Eivom application has been implemented to handle some of transaction processing on the mobile environment and others on remote Web services environments to allow better performance and availability. Computations that need considerable resource for processing are delegated to remote Web services, which are deployed on powerful servers. Some of other operations, those frequently asked by the user, are executed locally. The table below shows some of Eivom operations along with their processing environment:

D. Evaluation

The Eivom application attracted a lot of interests during the National Mobile Applications Contest in 2007 and was ranked 4th. During this event, sponsored by Nokia and organized in the United Arab Emirates, Eivom has been thoroughly evaluated by experts from Nokia, UAE main Telecommunication provider, and university professors. Evaluation criteria include originality, usability, quality, and focus on cultural aspects.

We exposed the mobile application to diverse users to evaluate their features and to test the efficiency of the application to handle a variety of users while ensuring the performance and the security of the executed transactions. Users showed an excellent interaction and satisfaction with regards to the ease of navigation and the variety of exposed features.

TABLE I
EIVOM REMOTE/LOCAL PROCESSING

peration	Description and processing location
getTrailer	It processes and displays a video trailer which requires considerable resources, thus processed remotely.
searchMovies	Requires considerable resources for searching movies in all theaters of all UAE cities, therefore is processed remotely.
viewSchedule	Returns the schedule of given movie in all theaters. It is better to use remote computation to handle this request.
getUserPrefrences getBookmark getUserProfile	These operations require fewer resources since it will process only user data, therefore processed locally.

VI. CONCLUSION

With exceptional rise in the number of mobile devices that are being used by an extensive number of users to explore a wide range of applications and access to online services. The need for a well defined development environment which handles most of challenges related to communication, security, limited resources, etc is of prime necessity. Therefore, we have presented a new framework to provide a concise methodology for secure development of mobile application and their related Web services.

The aim of this paper was to (1) highlights and discusses main issues to consider while developing mobile applications and their associated Web services, and (2) propose solutions to these challenges through a framework that supports developers to build up efficient, high quality, and secure mobile applications.

The framework has been used to develop the Eivom Cinema guide mobile application where developers followed

most of the framework development methodologies and best practices.

As future work, we plan to expose the Eivom application for quantitative evaluation where performance metrics can be measured to evaluate the performance of the application in terms of response time, availability, etc. We also, intend to evaluate our framework using diverse other mobile applications with a large number of users. We also foresee to extend the framework to provide QoS support and management such as QoS monitoring, etc.

REFERENCES

[1] B. Sofien and K. Nabil El, "An Overview of Mobile Applications Architecture and the Associated Technologies," 4th International Conference on Wireless and Mobile Communications, 2008, pp. 77-83.

[2] J. Nutzel and M. Kubek, "A mobile peer-to-peer application for distributed recommendation and re-sale of music," 2nd International Conference on Automated Production of Cross Media, 2006 2nd International Conference on Automated Production of Cross Media, IEEE Computer Society, 2006, pp. 93-98.

[3] K. Petrova, "Mobile learning as a mobile business application," International Journal of Innovation and Learning, vol. 4, no. 1, 2007, pp. 1-13.

[4] N. Delic and A. Vukasinovic, "Mobile payment solution - symbiosis between banks, application service providers and mobile network operators," 3rd International Conference on Information Technology: New Generation, Proceedings. Third International Conference on Information Technology: New Generation, IEEE Computer Society, 2006, pp. 346-350.

[5] O. Mayuk and T. Torabi, "Framework for mobile application development and content integration," 4th IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education, Proceedings - Fourth IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education, WMUTE 2006, Institute of Electrical and Electronics Engineers Computer Society, Piscataway, NJ 08855-1331, United States, 2006, pp. 69-73.

[6] Opera, "Opera Mobile," 2008; http://www.opera.com/products/mobile/.

[7] Microsoft, "IE Mobile," 2008; http://www.microsoft.com/windowsmobile/en-us/downloads/microsoft/internet-explorer-mobile.mspx.

[8] Apple, "Safari Mobile," 2008; http://www.apple.com/iphone/features/safari.html.

[9] AJAX, "AJAX," 2008; http://www.ajax.org/.

[10] SUN, "Java Micro Edition," 2008; http://java.sun.com/javame/index.jsp.

[11] N. Paspallis and G.A. Papadopoulos, "An approach for developing adaptive, mobile applications with separation of concerns," International Computer Software and Applications Conference, 30th Annual International Computer Software and Applications Conference COMPSAC 2006 (IEEE Cat. No. 06P2655), IEEE, 2006, pp. 299-306.

[12] C. Ming-Chun and Y. Shyan-Ming, "An adaptive mobile application development framework," The International Conference on Embedded and Ubiquitous Computing, Embedded and Ubiquitous Computing - EUC 2005. International Conference EUC 2005. Proceedings (Lecture Notes in Computer Science Vol. 3824), Springer-Verlag, 2005, pp. 765-774.

[13] K. Joeng, et al., "An application streaming service for mobile handheld devices," IEEE International Conference on Services Computing, 2006 IEEE International Conference on Services Computing, IEEE Comput. Soc, 2006, pp. 323-326.

[14] W. Ng, et al., "Comparative analysis of XML compression technologies," World Wide Web, vol. 9, no. 1, 2006, pp. 5-33.

[15] A. Ekelhart, et al., "XML security - a comparative literature review," Journal of Systems and Software, vol. 81, no. 10, 2008, pp. 1715-1724.

[16] ObjectWEB, "kSOAP," 2008; http://ksoap2.sourceforge.net/.

[17] J. Ala-Kurikka, et al., "Improving multiple mobile application interaction with unified session management," IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, Institute of Electrical and Electronics Engineers Inc., Piscataway, NJ 08855-1331, United States, 2006, pp. 1-5.

[18] R. Hens, et al., "Design of a multimedia gateway for mobile devices," SPIE - The International Society for Optical Engineering, Proc. SPIE - Int. Soc. Opt. Eng. (USA) 5684, 1 ed., SPIE - The International Society for Optical Engineering, 2005, pp. 137-145.

[19] N. Ravi, et al., "Context-aware battery management for mobile phones," 6th IEEE International Conference on Pervasive Computing and Communications, 6th Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2008, Institute of Electrical and Electronics Engineers Computer Society, Piscataway, NJ 08855-1331, United States, 2008, pp. 224-233.

[20] NSICOM, "Cr-E-Me," 2008; http://www.nsicom.com/.