Electronic ID: Services and Applications for Context-Aware Integrated Mobile Services

Dragos-George Comaneci*, Ciprian Dobre*
* University POLITEHNICA of Bucharest, Romania
E-mail: dragos.comaneci@cti.pub.ro, ciprian.dobre@cs.pub.ro

Abstract

Today smartphones integrate many sensors and provide large computing capacities. They enable the shift towards massive quantities of real-time information becoming access push rather than demand pull on a global case. CAPIM, a platform to support such a paradigm, integrates services to monitor and a context for adapt with the user's context using sensors and capabilities of smartphones, together with online social data. It integrates context-aware services that are dynamically configurable and use the user's location, identity, preferences, profile, and relations with individuals, as well as capabilities of the mobile devices to manifest themselves in many different ways and re-invent themselves over and over again. In the following paragraphs we present the design and development details of the security and user identification components to support these services. We propose a secure platform for user authentication and session management, based on public key infrastructure (PKI) services. We analyze its strengths and weaknesses, and present as a case study the particular extension of the platform to support secure user access to restricted areas of a building. We also discuss an analysis of the implementation, cost assessments and problems that might arise, as a methodology to support the construction of such mobile and context-oriented applications.

Keywords: PKI, Digital Certificates, Context-Aware Services, Secure Area Access, Bluetooth.

1. Introduction

As more people realize that having more sensing and computing capabilities in every-day situations is attractive for many reasons, smartphones become commodity hardware. Their success is the basis for a shift towards developing mobile applications that are capable to recognize and pro-actively react to user's own environment. Such context-aware mobile applications can help people better interact between themselves and with their surrounding environments. This is the basis for a paradigm where the context is actively used by applications designed to take smarter and automated decisions: mute the phone when user is in meeting, show relevant information for the user's current location, etc. CAPIM (Context-Aware Platform using Integrated Mobile services) [1] is a solution designed to support the construction of context-aware applications. It integrates services designed to collect context data (location, user's interests and characteristics, as well as the environmental data) and use it to provide a richer and simpler experience for the end user.

In the present work we approach CAPIM's design considerations for the management of *user identity* in context-aware applications. The user's identity is required by many context models. It can be used to infer preferences that are actively used in favor of the user, or it is used to provide personalized sets of services.

Today Public Key Infrastructures (PKI) solutions are generally accepted to support the management of identity. PKI provides a standardized and legally recognized service support [2]. Therefore it makes sense to use such services in providing electronic identity in context-aware integrated mobile services. PKI provides services such as confidentiality, integrity, authentication and non-repudiation [3]. By using the standards defined by PKI we can develop an approach to support the construction of rich context-aware applications that use the identity of the user as active context information. In particular, the security layer is used from the construction of social-aware mobile applications to intelligent housing, capable of actively recognizing the user entering the room for example.

A similar notable project in the area of secure user access to restricted areas of a building with the use of a mobile handset has been developed at the Disco Lab at Rutgers University ([3, 4]. Although the approach is different from our own (both in technology and system design), the goal is similar: to allow the use of a mobile handset as an electronic key, or, more generally, and electronic ID in order to access distributed services. However, we provide a more generic platform that actually include context as part of the entire process. Our proposed platform provides security guarantees as to who is accessing the contextual services, where people are. It instruments using context-oriented policies the interactions between peoples and services. In particular, we present a case study for the use of the platform as a tool to create a simple instrument to mediate the access for the user inside an intelligent building, capable of recognizing the user.

The rest of the paper is structured as follows. Section 2 presents the proposed Secure Service Communication Platform. In Section 3 we present the architecture of the Secure Area Access Service, and in Section 4 we make a detailed analysis of a real-world deployment case study. In Section 5 we conclude and present future work.

2. Secure Service Communication Platform (SSCP)

The proposed Secure Service Communication Platform (or SSCP) offers (1) a user session establishment mechanism, and a (2) session verification process that external services can use to verify the identity or authorization (based on various security policies) of the user operating the smartphone. The SSCP manages a session establishment mechanism between the smartphones and an authentication server. This is used for negotiating a secure connection with dual authentication (the client and server both present their digital certificates, so validation occurs on both sides). The server verifies the client certificate for validity (by using various verification mechanisms), retrieves the access rights for the user from a database, LDAP (Lightweight Directory Access Protocol) directory, or an ACL (Access Control List), and generates a session identifier that is transmitted back to the client.

2.1. The Main Components

The main components of SSCP are presented in Figure 1. On the mobile side two services are executed: the CAPIM Secure Communication Service, and the User Security Service. The *CAPIM Secure Communication Service* is responsible for session establishment, as well as for communication with other services that require user identification. The *User Security Service* implements the PKI operations for loading the user certificate, establishing of an SSL context for example, etc.

On the server side of the platform there are main two components responsible for authentication and authorization. The *authentication* CGI (Common Gateway Interface) component is responsible for verifying the credential of the user, for creating the appropriate session and registering it in a local database. The *session check* CGI component, as its name suggests, is used by the other services as an interface to check validity and retrieve information related to the user's context (such as the user's rights as defined by the security policy).

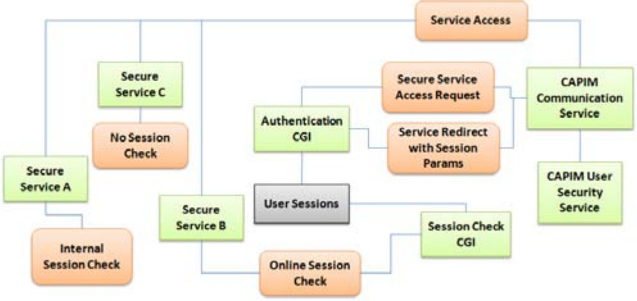

Fig. 1. The components of Secure Service Communication Platform (SSCP).

For more fine-grained audit requirements, the sessions (verifications and policy rights) can be established per service. In this case the communication service on client-side, upon authentication, also submits the name of the service that it requires access to. The authentication CGI verifies the name of the service in the local database, retrieves a certificate associated with that service, and uses the associated public key to encrypt a hash of the session key.

As illustrated in Figure 1, the services have three options for checking the user session, depending on the service requirements. The first option, used in the figure by *service A*, is an internal check of the session key. The session key is signed by the authentication service with its private key, so other services that have the certificate of the authentication service can use the public key stored within it to check the signature. This approach for verification is fast, but it involves processing on the service receiving the session key. Also, not much information about the user can be stored within the session key (which should be small, because it is transmitted with each request the user makes for a service). Therefore, this approach is suitable for services that only require a valid user and no other information (for example security rights).

The second option of verification, used by *service B* in Figure 1, is online session check. In this case the service receiving the session key from the user establishes a connection to the session check CGI. The session check CGI looks up the session in its' local database, retrieves the information associated with it (such as user details, user rights, etc…) and sends this information back to the requesting service. This approach involves minimal work on the service side but is also much slower since it requires communicating with the session check CGI. Another advantage is that the service can retrieve bundles of information associated with the user.

The third option (provided for consistency) is for services that do not require user identification. In this case the service simply ignores the session key.

2.2. Implementation details

The SSCP platform is designed to be modular and follows an object oriented design approach. The modular structure of the two CGI components is presented in Figure 2.

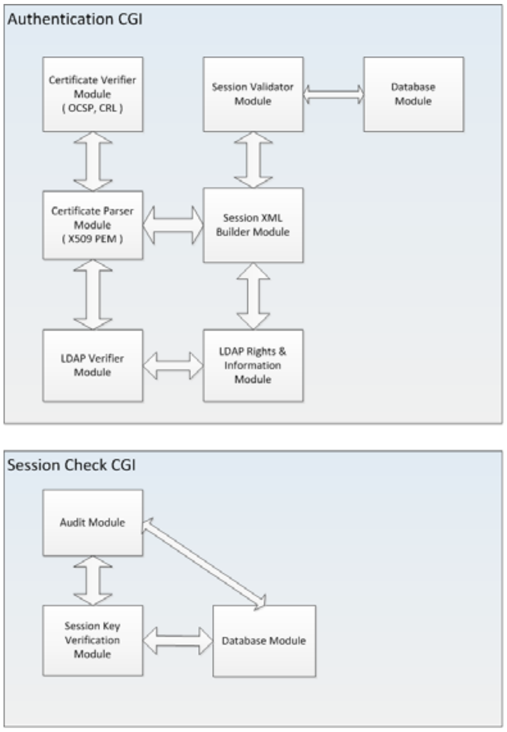
Authentication CGI
Certificate Verifier Module (OCSP, CRL)
Session Validator Module
Database Module
Certificate Parser Module (X509 PEM)
Session XML Builder Module
LDAP Verifier Module
LDAP Rights & Information Module
Session Check CGI
Audit Module
Session Key Verification Module
Database Module

Fig. 2. SSCP modular structure.

For standardization, the communication layer uses the HTTPS protocol. Modern smartphones have built-in support for working with HTTPS and, for some services, a web interface can also be provided. The authentication and session check services are implemented in C++, and use FastCGI ([14]) as the protocol for communicating with the web server (in our pilot implementation we use the Apache server). The technological choice allows the web server to spawn as many service instances as needed (or configured), each instance being responsible for sequential processing a number of requests. FastCGI also allows for distributed processing of requests; the web server, if configured, can spawn instances on different machines. The data storing layer uses unixODBC. Therefore, the services can be configured to run with any type of database. The authentication service also has support for verifying the user certificate and retrieving user rights using LDAP.

Since the system is using LDAP user group membership in order to verify permissions to access certain services, the security model undertaken is role based access control (RBAC). Each individual service is responsible for supplying the permitted user groups upon session key verification.

On the mobile side, implementation was carried out using the Java/Delvik API present on Android. The user certificate, along with its private key, are stored on the phone in PKCS#12 format encrypted by a combination of a user password and an unique information present on the smartphone. This ensures that the certificate and key are linked to a unique smartphone.

3. Secure Area Access Service

Access to a secure area of a building has always been a constant security problem and a lot of specialized solutions have been developed to facilitate this service [5-7]. Still, today they are either impractical for the user or lack the necessary security required for accessing sensitive areas. Also, the costs, both in equipment and training involved in implementing some of the solutions are prohibitive.

We propose a cost effective and practical solution for developing an Access Control service. The approach proposed in CAPIM is feasible because many users today carry at least one smartphone. The idea is to have a key in the form of a digital certificate and associated private key stored on the mobile smartphone and, after authenticating through SSCP and getting a session key, use the obtained session key to send an access request for a certain area to the service.

In order for the solution to work, the mobile smartphone has to be connected to the local Wi-Fi network. This is required to access the authentication service. Still, because the average Wi-Fi communication range is tens of meters, a more location sensitive solution is needed to determine that the user is in the presence of a door that protects access to a secure area. Hence, we also need to use Bluetooth for our propose access service.

The proposed solution works as follow. In the beginning the users' mobile smartphone is connected to the local Wi-Fi network and authenticated through SSCP, thus having a valid session key. Using the location service the mobile handset determines that it is in the proximity of a door that leads to a restricted area, and automatically turns on the Bluetooth receiver on the phone and scans the area for devices. The mobile handset finds the device corresponding to the Bluetooth dongle of the door and proceeds to generating a random shared-key that will be used for the association between the two Bluetooth devices. The random shared-key is posted along with the MAC address of the mobile handset via Wi-Fi to the Secure Area Access Service (SAAS). The mobile handset then begins the Bluetooth association procedure with the dongle of the door. The device controlling the Bluetooth dongle of the door detects the MAC of the device trying to associate and queries the SAAS for the random shared-key generated by the mobile handset, retrieves it and uses it to carry out Bluetooth association.

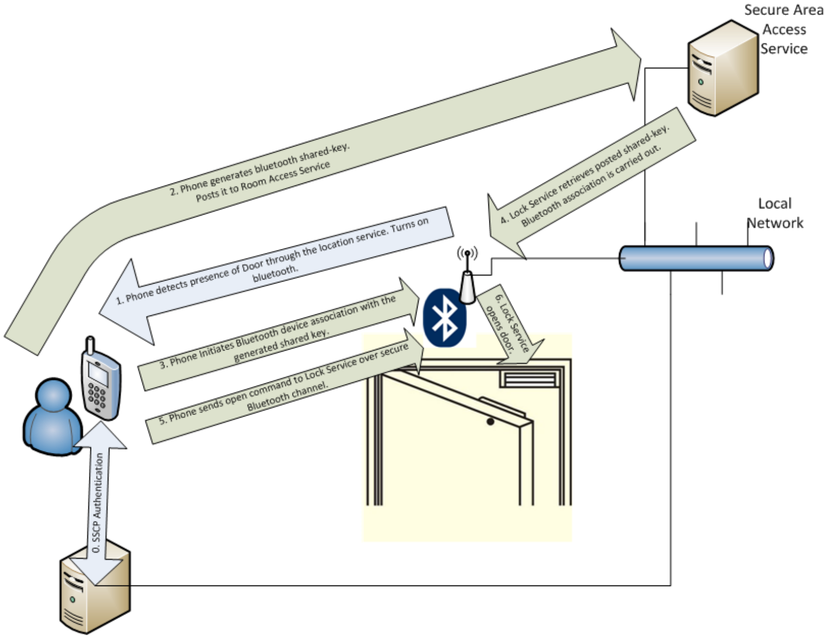
Secure Area Access Service
2. Phone generates bluetooth shared-key. Posts it to Room Access Service
4. Lock Service retrieves posted shared-key. Bluetooth association is carried out.
Local Network
1. Phone detects presence of Door through the location service. Turns on bluetooth.
3. Phone Initiates Bluetooth device association with the generated shared key.
6. Lock Service opens door.
5. Phone sends open command to Lock Service over secure Bluetooth channel.
0. SSCP Authentication

Fig. 3. Visual representation of the process for door opening.

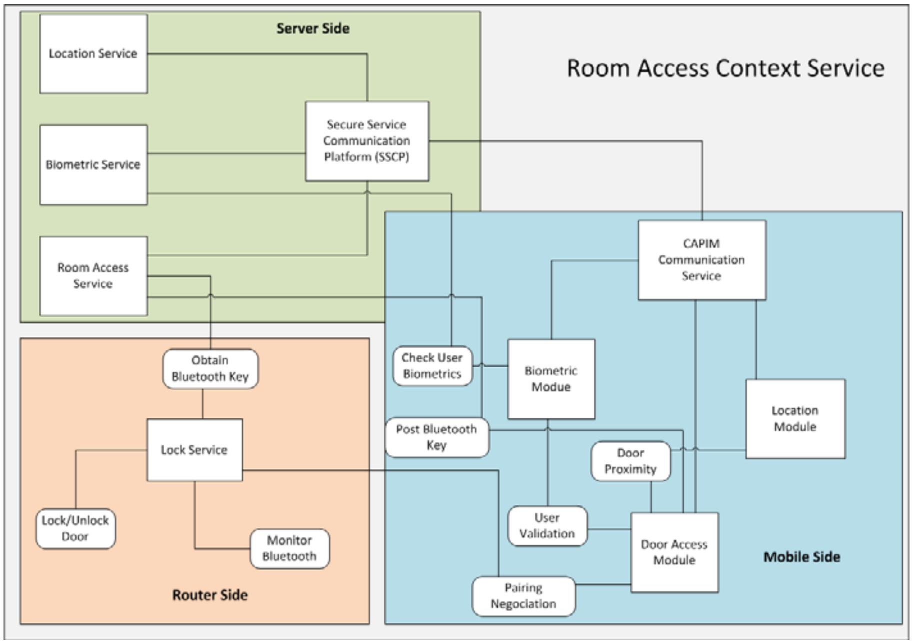

Fig. 4. Main components of the Secure Area Access Context Service.

After the association is complete the mobile handset sends a hello message and the door is opened. The association between the two devices is kept for a limited amount of time (for example 1-2 hours) so further access through the door are simple. A visual representation of this process is presented in Figure 3.

To ensure a higher level of security for extra-sensitive areas, the SAAS may require the user to prove its identity. This is done using a biometric service based on face recognition [10], but this can easily be adapted to alternate biometric inputs, such as voice or fingerprint.

The best device we found that can serve as a controller for physical access to the area, and to monitor incoming Bluetooth connection requests, is a router fitted with two USB ports (one for a controller for the magnetic lock on the door and another for a Bluetooth dongle). The router can also provide Wi-Fi network connectivity for the area.

3.1. Main components

The main components of the Secure Area Access Context Service are presented in Figure 4. Emphasis is given to the location of each component in the system. The *Lock Service daemon* resides on the router controlling access to one or more restricted areas of a building. The daemon is responsible for monitoring Bluetooth connection requests from different Bluetooth dongle receivers connected via USB to router and also servicing door open requests for the doors it controls.

On the server side there is the *Room Access Service*, which is a Bluetooth shared-key repository where the mobile handset posts the randomly generated shared-key for Bluetooth association, along with the MAC address of the handset, the Lock Service present on the router following up and retrieving the posted key. Also on the server side the *Location Service* offers information regarding the location of access points to restricted area, and the *Biometric Service* which checks user biometrics and can be used in case of highly restricted areas.

On the mobile side there is the *Door Access Module*, responsible for Bluetooth association, and the management of door access requests generated from door proximity alerts coming from the Location Module. Also, on the mobile side there is the *Biometric Module*, responsible for taking a photograph using the mobile handset integrated camera of the current user and sending it to the Biometric Service for verification.

3.2. Implementation details

For the implementation of the Lock Service an Asus 500gP V2 router with two USB ports was used. The firmware was replaced with the one provided by the open-source project DD-WRT [6] in order to have root access to the device and install the Bluetooth and door controller modules, along with the Lock Service. The Lock Service was developed in C++ and cross compiled for the MIPS32 platform to work with the processor present on the router. The Lock Service also uses the BlueZ Bluetooth library for accessing the dongle connected to the router and OpenSSL for accessing the Room Access Service.

An example of the hardware configuration is presented in Figure 5. The Room Access Service was developed, as in the case of SCCP, in C++ using FastCGI for communicating with the hosting web server. It stores the shared keys in a database and uses unixODBC for database access so that any flavor of database can be configured with it.

The Biometric Service was developed in Java as a distributed service that supports multiple dispatchers and workers and uses the EigenFace image matching algorithm [9]. A more detailed description of the Biometric Service can be found in [10]. Details regarding the Location Service and Location Module, developed within the CAPIM project, are available in references [11,12].

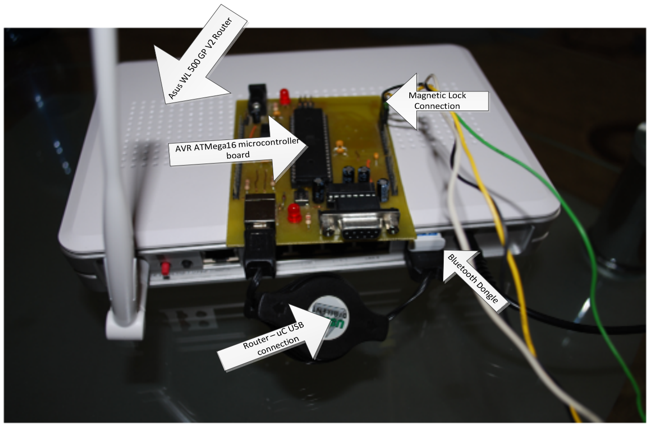

Fig. 5. Hardware configuration for SAAS.

The mobile components (Door Access Module, Biometric Module and Location Module) were developed in Java using the API provided by the Android platform.

4. Real-world deployment case study

Because the system is composed of many separate components that depend on one another, as well as on external services, a special care must be taken regarding their placement and deployment within an organization. As such, in this section the details regarding a proposed deployment of the system are presented and discussed.

4.1. Required components

PKI being at the base of the system, the most crucial component is the certification authority that will issue certificates for the users. It is recommended that each organization has its own internal CA for issuing certificates. Also, special protection measures must be considered to secure the CA private key. We used an LDAP server to store the user related information and certificates. Details regarding PKI and deployment recommendations are given in [13].

The next component on the list is a database system that will be used to store both the user sessions from the SSCP platform, as well as data from the Location, Biometric and Room Access services. The databases and their location must be scaled accordingly to the number of system users and the number of protected areas present within the building. Also, since the Biometric service deals with large amounts of image data (a set of 16 images with different lighting conditions is required for each user in order for the algorithm to work correctly), it is recommend that a separate database server be used for this service.

A further requirement is an Apache web server for hosting the FastCGIs for SSCP and the Room Access service. Also, for redundancy and load balancing purposes, a greater number of web servers may be configured.

For each secured area, a router is also required. The router connects the Bluetooth dongle and the magnetic lock controller for each controlled door. The router also hosts the Lock service

4.2. Cost assessment

The project was designed to be as cheap to implement as possible so, for the server components, existing computing hardware may be used. The only costs incurred are for the routers, Bluetooth dongles, magnetic locks and magnetic lock controllers. A two USB port WiFi router can be found at a medium price of 60€. A Bluetooth dongle compatible with BlueZ incurs a cost of 20€. The magnetic lock controller can be built out of a microcontroller board with an USB port and two relays that control the locking mechanism, the total component costs for it being around 20€. The most expensive component is the magnetic lock itself which, on average, has a cost of 100€. So, the total costs per room add up to 200€ for all the required hardware components.

The configuration time required for the server side components is, for an average configuration, around one hour. A preconfigured virtual machine containing all the components required for a functioning system is provided on CAPIM's official page.

4.3. Operating problems - assessment and mitigation

Several problems related to hardware or software failure or even planned attacks may arise during the operation of the proposed system. These problems must be identified and backup solutions must be provided for their resolution.

The most common problem is that of hardware failure. Many solutions exist for the server side components, so we will insist more on the router and mobile side. On the router side, three main components can fail: the router itself, the magnetic lock controller and the Bluetooth dongle. In order to detect these failures, the lock service must report periodically to a centralized service the status of its components. This is done using an already existing logging infrastructure such as syslog or a specialized communication between the lock service and a central reporting service.

In case the router itself fails (or the lock service residing on it), the failure can be detected by inspecting the time of the last report received from the lock service. Also, failure to contact the lock service can be detected and reported automatically by the mobile smartphone to a centralized reporting service. For the case when the router fails a manual override is provided for the magnetic lock.

If the Bluetooth dongle fails, the failure is detected and reported by both the lock service and the mobile handsets. Depending on security policy, access requests may be made directly from the mobile handset to the lock service residing on the router via Wi-Fi.

The magnetic lock controller may also fail, in which case, the lock service is the only one capable of reporting this issue. Special precautions must be taken when designing the magnetic lock controller such that, on failure, the magnetic lock operated by it must not be opened. Also, the magnetic lock controller firmware must report its status periodically to the driver present on the router.

On the mobile side, a smartphone can be stolen or information regarding the user identity stored on it may be copied. In order to prevent a malicious user from using a stolen mobile handset, a biometric service has been developed and integrated with the system. Although far from perfect, the biometric service adds an extra layer of security. Of course, extra precautions are taken in assuring that the image received by the biometric service did indeed come from the mobile handset integrated camera (the image is digitally signed using the user's private key).

In order to prevent the copying of the user identity (in our case the user's certificate and private key) to another handset, both the key and certificate are stored within a PKCS#12 certificate software store protected by a password containing both a user part (that will be inputted on each use of the certificate) and a device specific part derived from information found only on that certain device on which the certificate will be used. In order to detect a malicious user from extracting that specific information from the phone, special measures are considered in obfuscating the code portion that derives the certificate store password from it.

5. Conclusions

Smartphones are today becoming commodities. Considering that even today more than half a billion people have at least one smartphone, the previous affirmation is not so far-fetched. The advances in mobile technologies allowed people to have in their pockets, wherever they go, powerful computing devices, which can be of great help in their activities. CAPIM is a platform designed to support the shift towards massive quantities of real-time information becoming access push rather than demand pull on a global case. It integrates services to monitor and a context for adapt with the user's context using sensors and capabilities of smartphones. It integrates context-aware services that are dynamically configurable and use the user's location, identity, preferences, profile, and relations with individuals, as well as capabilities of the mobile devices to manifest themselves in many different ways and re-invent themselves over and over again. However, an important component in this architecture is the one responsible for user management. The user's identity is required by many context models. It can be used to infer preferences that are actively used in favor of the user, or it is used to provide personalized sets of services. In this we proposed a generic platform to support the identity management using PKI support. The approach to support the construction of reach context-aware applications uses the identity of the user as active context information. In particular, the security layer is used from the construction of social-aware mobile applications to intelligent housing, capable of actively recognizing the user entering the room for example. The Secure Service Communication Platform (SSCP) is as a way of providing user identification to all services present within the system. Based on this platform, we presented a security sensitive service for access to a secure area of a building. We presented implementation details and a real-world deployment case study.

Acknowledgments
The research presented in this paper is supported by national project: "TRANSYS – Models and Techniques for Traffic Optimizing in Urban Environments", Contract No. 4/28.07.2010, Project CNCSIS-PN-II-RU-PD ID: 238. The work has been co-funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

CAPIM's official site and source code repository are available at http://cipsm.hpc.pub.ro/capim.

References

[1] C. Dobre, "*Context-Aware Platform for Integrated Mobile Services*", in Proc. of International Workshop on Services for Large Scale Distributed Systems, Tirana, Albania, September 2011.

[2] E. Carayannis and E. Turner, "Innovation diffusion and technology acceptance: The case of PKI technology", *Technovation*, vol. 26(7), pp. 847-855, 2006.

[3] N. Ravi, P. Stern, N. Desai and L. Iftode, "Accessing Ubiquitous Services Using Smart Phones", *Proc. of the 3rd Intern. Conf. on Pervasive Computing and Communications*, 2005

[4] L. Iftode, C. Borcea, N. Ravi, P. Kang and P. Zhou, "Smart Phone: An Embedded System for Universal Interactions", *Proc. of the 10th IEEE Intern. Workshop on Future Trends of Distributed Computing Systems* (FTDCS 2004), May 2004

[5] I. Hwang and J. Baek, "Wireless access monitoring and Control System based on Digital Door Lock", *Consumer Electronics*, IEEE Transactions on, vol. 53(4), pp. 1724-1730, 2007

[6] C. Hsu, S. Yang and W. Wu, "Constructing intelligent home-security system design with combining phone-net and bluetooth mechanism", *Machine Learning and Cybernetics*, 2009 International Conference, vol. 6(1), pp. 3316-3323, 2009

[7] Y. Park, P. Sthapit and J. Pyun, "Smart digital door lock for the home automation", *TENCON 2009 - 2009 IEEE Region 10 Conference*, vol. 1(1), pp. 1-6, 2009

[8] D. Heldenbrand and C. Carey, "The Linux router: an inexpensive alternative to commercial routers in the lab", *J. Comput. Small Coll.*, vol. 23(1), pp. 127-133, 2007

[9] J. Zhang, Y. Yan and M. Lades, "Face recognition: eigenface, elastic matching, and neural nets", *Proc. of the IEEE*, vol. 85(9), pp. 1423-1435, 1997

[10] D. Comaneci and B. Vlad, "Face Biometric Distributed Authentication Service", Numeric Systems Architecture Course Project Description, Faculty of Automatic Control and Computers, Computer Science Department, University Politehnica of Bucharest, 2011.

[11] I. Militaru, "Indoor Localization Service", SCS 2011, University POLITEHNICA of Bucharest, Romania, 2011.

[12] D. Greceanu, "Platform and Services for Context Information Agregation and Visualization", SCS 2011, University POLITEHNICA of Bucharest, Romania, 2011.

[13] R. Housley, T. Polk, "*Planning for PKI: best practices guide for deploying public key infrastructure*", Wiley, Washington, 2001.

[14] Open Market, FastCGI, http://www.fastcgi.com/, last accessed June 10, 2011.