Secure Mobile Business Applications – Framework, Architecture and Implementation[1]

Thomas Walter
DoCoMo Euro-Labs
Landsberger Strasse 312
D-80687 München,
Germany
Tel: +49-89-56824 210,
Fax: +49-89-56824 300
walter@docomolab-euro.com

Laurent Bussard,
Yves Roudier
Institut Eurécom
2229 Route des Crêtes –
BP 193
06904 Sophia Antipolis,
France
Tel:+33-4-93-00-26-26
Fax :+33-4-93-00-26-27
bussard@eurecom.fr
roudier@eurecom.fr

Jochen Haller,
Roger Kilian-Kehr,
Joachim Posegga,
Philip Robinson
SAP Research
Vincenz-Priessnitz-Str. 1
76131, Karlsruhe,
Germany
Tel: +49 721 690210,
Fax: + 49 6227 7830361
Jochen.Haller@sap.com
Joachim.Posegga@sap.com
Roger.Kilian-Kehr@sap.com
Philip.Robinson@sap.com

Abstract

Emerging mobile technologies such as PDAs, laptops and smart phones together with wireless networking technologies such as WLAN and UMTS promise to empower mobile employees to become better integrated into their companies' business processes. However, the actual uptake of these technologies is still to come; one hindrance is security of mobile devices and applications.

In this contribution we present an in-depth analysis of the current situation enterprises are faced with in the mobile *arena*, both from a security and a management perspective. We argue that the currently predominant model of *perimeter security* will not scale for future mobile business applications that will require appropriate *application-level* security mechanisms to be in place.

We present a framework offering solutions for the development of *secure mobile business applications* that takes into account the need for strong security credentials, e.g. based on smart cards. This framework consists of software and abstractions that allow for the separation of the core *business logic* from the *security logic* in applications. Security management instruments in the form of enforceable enterprise policies are defined which target the security and trust-related deployment and configuration of mobile devices and business applications.

The presented architecture is open, in the sense that the actual mobile business application can span over heterogeneous client devices, forming a so-called *federation*.

1. Introduction

Modern infrastructures of enterprises tend to be based on a multi-tier architecture that – seen from a user's perspective – usually comprises:

- a desktop tier, e.g. a web browser or other client-resident application;
- a portal, which presents to a user the enterprise resources according to the specific role the user performs in a task-centred view;
- an application server tier, such as web application servers;
- a backend tier, often comprised of databases or other business-specific backend systems (e.g. in the production process of a company).

Application server and backend tiers are bound together using additional enterprise application integration systems, thus implementing the actual business processes.

1.1. Security models

Enterprise information technology (IT) infrastructures are today protected according to the so-called *perimeter security model*: its principle is to implement security at the network level using firewalls, intrusion detection systems, etc. Enterprises turn into *fortresses* by building network walls to separate trustworthy and less trustworthy parts of the network; security is managed and enforced at the network borders.

Mobile and wireless technologies such as laptops, PDAs, and smart phones with

1. IST-Programme / KA2 / AL: IST-2001-2.1.3. The project WITNESS was supported by the European Community. This document does not represent the opinion of the European Community. It is also the sole responsibility of the authors and not the responsibility of the European Community using any data that might appear therein.

1363-4127/04/

cellular and mobile network interfaces, e.g. for GSM/UMTS [23], potentially enable enterprises to let their mobile employees get access to the core information technology infrastructure through dedicated enterprise portals. Accessing corporate resources without duplicating the corporate infrastructure would make it possible to accelerate business processes and to integrate more tightly in-house and remote work, thus improving the usability, flexibility, and performance of information systems.

Seen from the enterprise's perspective, mobile access to networks will change the core business processes of an enterprise only to a limited extent, since the actual *business* of a company will most likely not be affected. However, mobility support will significantly change the way business processes are actually executed.

As a consequence of extending the enterprise perimeter to mobile devices, new threats arise such as theft and loss of mobile devices, wireless eavesdropping which in turn might lead to leakage of confidential data, uncontrolled access to corporate resources, etc. It should be noted that the value of some business transactions can be quite different from what is typically found in business-to-consumer scenarios. Consider for example a financial controller who monitors key performance indicators of different enterprise branches – information typically considered as insider information of significant value.

Compared to the overall number of existing business applications, only a fraction of these are actually enabled for mobile usage. We are convinced that one of the major reasons for the slow uptake of mobile business applications is the fact that suitable frameworks for the development of secure applications still do not exist or are not suitable for actual business needs [4]. Security measures in place should be able to cope with a diverse range of security requirements.

1.2. Problem statement

Future *mobile business applications* will span several trust domains from enterprises to personal domains, across different organisations and businesses. In such scenarios only the application will be able to determine the security requirements that need to be implemented. This is very much in contrast with the perimeter security model where the network *locations* determine whether mobile employees are allowed to enter the business domain.

Apart from the security problems emerging with mobile usage, enterprises must also be able to integrate these devices into their IT infrastructure. Most notably, seamless solutions that build on users working habits and empower them with flexibility and ease-of-use should be preferred. This also results in the following non-exclusive list of requirements:

- security modules in use (e.g. GSM/UMTS SIM cards [1], [2]) must be configurable and manageable by the enterprise itself (e.g. key management, authentication mechanisms, cryptographic algorithms);
- deployment, management, and maintenance of secure mobile business applications must be as far as possible expressible in corporate policies;
- trustworthiness of mobile devices must be made accessible to business applications to allow for fine-grained control over the information flow within an enterprise.
- security infrastructure in place on the client and server must be accessible for third-party application development.

Seen from the (usually) third-party application developers' and providers' point of view, additional requirements are:

- mobile business applications must be able to define their own security requirements in conjunction with corporate security policies, something which is usually not possible on the network level anymore;
- application development should allow for the separation of core *business logic*, which implements the actual business processes, and the *security logic*, which defines the security measures that must be enforced based on corporate policies.

1.3. Secure mobile business applications

Mobile devices equipped with wireless communications facilities are a platform to run business applications anytime and anywhere. The term *secure mobile business application* refers to the protection of critical corporate resources – data, processes and network equipment – accessed by applications deployed on mobile devices and beyond the security perimeter of an enterprise.

Security, however, always is a term relative to a given policy: assets are only as secure as what meets the explicit requirements. A business model is defined by its policies and practices, which govern how assets are used, how people interact, and how processes are carried out. Applications are the way business models are technically implemented. They should therefore reflect company policies (including security policies) through their implementation. Security policies clearly state both required security services and security mechanisms. The latter being the building blocks that establish a corresponding standard of security.

1.4. Contributions of this work

This paper presents a framework and architecture for developing and deploying secure mobile business applications. We concentrate on scenarios where mobile employees require access to their own corporate resources (business-to-employee, B2E) and to resources of business partners (B2B). Our work defines an integrated approach for designing, integrating and deploying secure mobile business applications.

The application security support architecture provides the flexibility and adaptability to cope with the above identified problems:

- A smart card hosted security module (Section 3.4.2) provides tamper-resistant storage and an execution environment for security critical data and operations;
- Policy management (Section 3.5) supports the definition, maintenance and enforcement of corporate security policies. Policies are bundled with an application and are pre-installed or dynamically downloaded to a device;
- As a sub-component of policy management, specific attribute certificates (Section 3.5.2) are employed to describe trustworthiness of mobile devices as well as to define rights for delegation of tasks between employees.
- Separation of business and security logic supports deployment of applications irrespectively of any security considerations which are dealt into the security logic layer of the framework (Section 3.7).

The described framework and architecture have been defined, designed and implemented in the WiTness – Wireless Trust for Mobile Business – project [24]. To our best knowledge no similar

comprehensive approach such as the discussed one exists.

1.5. Outline of paper

The following section provides additional background material and basic definitions. Section 3 discusses the framework and architecture for implementing secure mobile business applications. We argue for a strict separation of business and security logic but also for an integrated approach that takes security considerations into account from the very beginning. The building blocks and generic components of our framework are discussed. Section 4 discusses a use case that emphasizes applicability of concepts in a specific scenario. Before concluding the paper we briefly compare our results to related approaches (Section 5).

2. Federations of mobile devices – extending the security perimeter

Our proposed security framework extends security features of a corporate intranet to a mobile device and even further into the domain of co-operating mobile devices.

2.1. Usage scenario – pervasive salesperson

We consider a pervasive salesperson visiting customers and who enters new orders in the corporate database; the database is located in the corporate domain. The mobile devices in the salesperson's immediate vicinity are called the *personal domain*:

- To read corporate emails, the salesperson uses a mobile phone;
- To plan forthcoming customer visits, he or she uses his or her PDA;
- To enter order data, he or she uses his or her laptop.

If a data item has been sent to a mobile device it is beyond the security perimeter of the enterprise. The challenge is to what extent security can be enforced even under the assumption that, as in the described scenario, resources cannot be controlled explicitly.

In the following sections we describe our approach to support application security in various environments.

2.2. Federations of mobile devices – definition

The pervasive salesperson example demonstrated the advantage of exploring the capabilities of different mobile devices in performing business tasks. The notion of a federation captures this aspect as well as the co-operation of mobile devices and corporate servers.

A *federation* is a combination of mobile devices and corporate servers that executes distributed business applications and that offers the following features:

- access control (i.e. authentication and authorization), data confidentiality and integrity as well as accountability of communication events and non-repudiation of transactions are configured according to security policies;
- co-operation between mobile devices is constrained by federation policies and trust certificates;
- delegation of business tasks and data resources between mobile devices and between mobile devices and corporate servers is constrained by delegation policies and authorization certificates.

Federations make it possible to extend the capacity of each federated device including its deployed business applications with those of its federated counterparts.

For instance, using the screen of a laptop extends the display capabilities of a PDA; a laptop may extend the computing capabilities of a cell phone, and so on.

2.3. Security and federations

By their very nature, federations of mobile devices are more exposed to attacks than plain mobile devices since an application running within a federation spans across devices possibly owned by different administrations.

The enterprise must be able to implement and enforce its security policy regarding how its data are accessed by its employees and its partners. Securing data access in a federation is achieved in two separate steps: first, the company must ensure that sensitive data do not get disclosed in an uncontrolled fashion; second, the employee must be able to designate used appliances in a context-dependent manner, based on his location, history, and last but not least defined policies.

Federating a device may increase its security, and may be required with respect to authentication in the corporate policy. For instance, a user may access some specially critical data on his or her laptop only if he or she authenticates using a fingerprint reader located on a PDA; or the user may carry in his or her clothes a device proving his or her presence in a certain distance range so that his or her mobile phone will lock if the user is too far away from the device, thus reducing the risks incurred if the mobile phone is stolen.

3. Application security architecture – design and implementation

Given the identified requirements (Section 1.2) and mobile working scenarios (Section 2.1) it becomes obvious that application security can only be orchestrated from within applications themselves. The application security architecture takes into consideration the needs for *enhanced security management*, enabling true end-to-end security through *application layer security*, and allowing established security mechanisms to be leveraged.

Enhanced security management is achieved by, first, a separation of business and security logic and, second, a comprehensive security policy management which includes security policy enforcement as well. Application layer security is supported end-to-end between mobile devices and enterprise networks and between mobile devices used by the employee in his or her personal domain – a federation.

3.1. Components of the WiTness framework

This section introduces the WiTness framework components that are the atomic software parts provided by the framework which promote a (mobile) business application to a secure mobile business application (Figure 1).

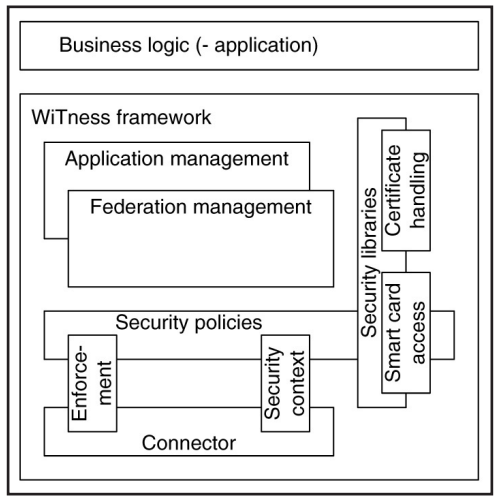

Figure 1: WiTness framework

Application management provides an environment for hosting the software components of a business application which are termed *partlets* in our Java™ based framework. Application security requirements are expressed as *security policies* which accompany partlets. Evaluation of policies during the execution of a partlet is done by application management. Policies are bound to security services and mechanisms as available and implemented by installed *security libraries* on the hosting mobile device. The selection and instantiation of security services and mechanisms determine a partlet's *security context*. To a large extent, application security logic is implemented by the respective security mechanisms of the security context. Security libraries implement basic security services and mechanisms but as well specific security services for certificate handling and smart card access.

Multiple partlets can run in parallel and can communicate with partlets on federated devices or partlets in the enterprise intranet.

Federation management handles requests from one partlet to federate with other partlets in either enterprise intranet or personal domain. Note that, vice versa, federation set-up requests may originate from other external partlets as well. Thus, federation management is in charge of identifying federated devices and controlling outgoing and incoming federation requests.

3.2. Application management

The WiTness framework takes a policy driven approach to meet an application's security requirements. All business logic is implemented as partlets running in the application execution environment called *application manager*. The application manager is the main enforcement point for partlet related security policies as well. Its duties encompass:

- Partlet lifecycle operations;
- Dynamic addition and deletion of partlets during runtime;
- Partlet communication operations.

Partlets are accompanied by a set of policies specifying the lifecycle behaviour and authorisations of this partlet such as conditions to be met before starting the partlet or communication constraints. The application manager then starts and stops available partlets or dynamically downloads partlets based on policy decisions. Since communication handles in form of connectors (Section 3.6) are set up from within the partlet and therefore in the application manager controlled execution space, it is also possible to enforce communication related policies.

The following subsections describe in detail the relations of these application management duties to other specialized WiTness framework components.

3.3. Federation management

A federation of mobile devices is a centrally organized structure that develops from the trust assignment in the personal domain. A federation itself is seen as a secure platform for applications in the form of partlets. Security determination is hereby based on assigned policies regarding the different components, e.g. if the application managers on each device are able to enforce all participating partlets' policies.

The central component receiving federation related requests from surrounding devices in order to form and maintain a federation, is called the *federation manager*. Its communicating counterpart, the *federation handler*, is

available in each mobile device with an installed WiTness framework.

Federations are initiated by federation handlers contacting the federation manager based on a discovery mechanism which is specified independently from implemented network layers or frameworks. The WiTness project implemented the framework specification in different ways experimenting with federations based on lower network layer mechanisms, e.g. UDP/IP broadcasts, and high level virtual networks such as Project JXTA [11].

3.4. Security libraries and mechanisms

Security in the WiTness framework is provided from the beginning as potentially available, mostly static mechanisms collected in the so-called *security library* (Figure 1). Security policies state – and ideally parameterize – where and how to make use of the security mechanisms to meet the application's security requirements. A distinction is made between different kinds of security data and related operations. Tokens for mobile device authentication such as device certificates are less critical since these are only valid and utilized in the personal domain. They have no impact in the enterprise domain where the high risk – high impact damage would be inflicted. Critical data consist, for instance, of the mobile user's credentials, and operations working on them such as digital signatures utilizing the user's private key. These security properties obviously have to be adequately protected in the so-called *security module* and will be described separately in the following subsection.

3.4.1. Security libraries

The security libraries (Figure 1) are present on each device containing the static mechanisms needed by the WiTness security framework. These mechanisms are the essential tools to establish confidential communication channels, verify authentication tokens, etc.

The security libraries also provide the basic mechanisms for federations to enable the federation manager to control their lifecycle and constantly monitor their state throughout their existence. The basic operations required for these duties are operating on certificate chains, and include, for instance, in chain verification or issuance/signing of certificates.

3.4.2. Security module and services

The security module in the WiTness security framework is assigned the crucial role to provide a secure storage and execution environment.

The secure and personal storage may be assured by using tamper-resistant hardware such as *smart cards*. Security critical data, e.g. user credentials used for authentication or associated administrative data such as a chain of trusted root certificates, are stored in the smart card (Figure 2). The smart card also provides a secure execution environment, offering security services to the federated devices. These security services are actively answering to requests from the federated devices. They hereby provide a protective layer between the secured data and the outside world of the security module. In this manner, the protected data are never exposed outside of the security module.

Since the security module deals with critical user data, there is always one security module present per federation of mobile devices. The security module is needed throughout the federation's lifetime. Every device which performs validation of a

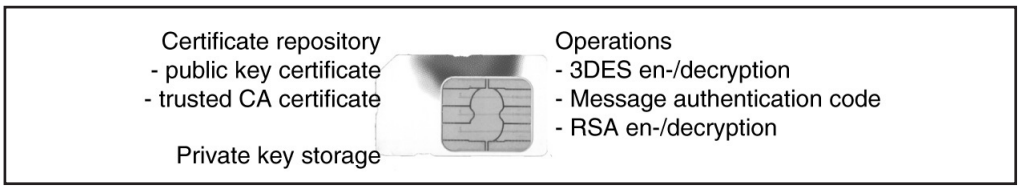

Figure 2: Security module – data and operation

certificate needs access to such services to validate the certification authority (CA) against the chain of trusted root certificates stored inside the security module.

3.5. Security Policies

Security policies are the rules for creating a mobile business application's *security context*, and are defined during application development and integration process. Within the WiTness framework (Figure 1) they are bundled with an application or partlet, when the partlet is installed in the corporate network and on mobile devices or dynamically downloaded to a device after it was issued.

Basically, policies are concerned with:

Access control: the corporate server must verify that resources are accessed only by authorized users. It can be based on different types of credentials such as encrypted password, fingerprint, challenge-response (private key), authorization certificates, etc. in combination with access control lists. Policies that restrict access are primarily specified in the corporate domain, although its enforcement is distributed between mobile devices.

Communications security: the corporate server must maintain a secure communication link with its clients. It defines the type of security properties (integrity, confidentiality, algorithm, key size) that have to be negotiated for communication between devices.

Compliance with these properties is enforced both in the corporate domain as well as in the personal domain.

Device trust: if and how data are distributed between members of a federation depends on whether a device can be trusted to execute a given operation or how data and code have to be pre-processed before being given to the members of a federation. However, such a process is restricted by the computational capabilities of a device, criticality of data (e.g. public vs. confidential), as well as the trust level that can be assumed. Although such policies are basically set forth in the corporate domain they have to be enforced on federated devices themselves.

Overall, however, one has to keep in mind that security and trust requirements may be specific for an application. They are, in general, hard-coded when the application is developed. For example, an application requiring non-repudiation or fair exchange will have to be developed with those features included at the places where they are needed. Such security properties do not characterize an application but are a part of its logic: a non-repudiation property may not apply to an application while it can indeed apply to a message origin. Policies that go with such requirements are enforced through the use of specific code dispersed in the application. This code may either be programmed in an ad hoc way, using pre-determined mechanisms available in security libraries, or be automatically generated in the corporate domain, before distributing

the application. The latter approach thereby potentially permits retrofitting legacy application code, and may be enabled by use of aspect oriented programming [7].

3.5.1. Automating security policies

The WiTness framework offers four types of templates for automating security policies, namely *authorization*, *configuration*, *delegation* and *federation* (Figure 3). The latter three types belong to the core of the WiTness framework; however, we include the authorization policies for completeness.

assertions regarding the validity of devices to be included in a federation. Trust certificates are the containers that store information on devices which is then evaluated against federation policies.

The above described policies define a hierarchy as shown in Figure 3. At the top, the authorization policies define the rules applicable in the corporate domain. Configuration policies are defined such that they provide evidence that the fulfilment of the configuration policies implies the fulfilment of the authorization policies. Configuration policies are refined into delegation and federation policies. This

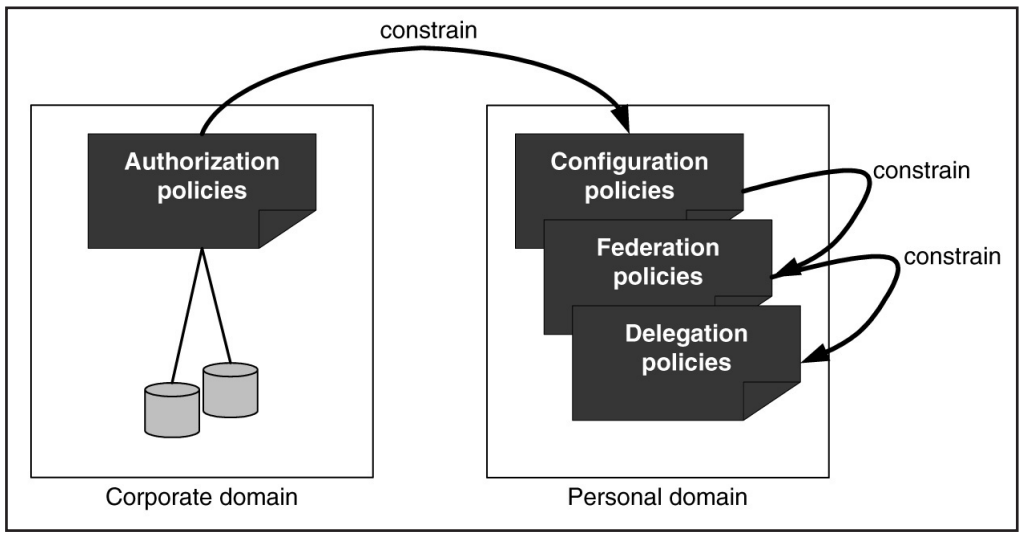

Figure 3: Security policy types

Authorization policies typically exist within a corporate network and control the access to resources as well as the condition for a secure and trusted exchange of these resources within the network (corporate and personal domain). The *configuration policies* are derivatives of the authorization policies. The framework specifies a *delegation policy* that states the rules before a task can be delegated between personal domains of employees. The framework specifies a *federation policy* for providing

refinement, however, must preserve the configuration policies in the sense that the above mentioned implication, i.e. configuration policies implies authorization policies, still holds.

3.5.2. Attribute certificates

Enforcement of security policies related to access control is based on the validation of chains of authorization certificates. The enforcement of policies related to device

trust is based on the validation of chains of trust certificates.

Authorization certificates are bound to employees. They specify and determine the role of an employee and his or her rights. For instance, an employee in the role of a sales department manager has the right to access the order database, to perform approval of orders and to revise budget figures. The sales manager may also be entitled to delegate certain tasks, e.g. the task of approving orders.

Trust certificates are assigned to devices. They provide evidence on the trustworthiness of devices. We can think of a situation where secret data should only be processed on devices that come from the employee's company. Before delegation of a task to another device takes place, the device's trust certificate is evaluated. If the certificate proves that the device is owned by the company, the application task and data can be delegated and are transferred to the device.

Additionally, co-operating companies may mutually agree to maintain an appropriate security and trust level on their devices. Then some tasks and data may be shared among devices that are owned by different companies. All this can be expressed respectively by attribute certificates and delegation and federation policies.

3.5.3. Enforcement of policies

For federation policies, first the scope of federative devices is specified which might be any device in the personal domain, any device in another employee's personal domain, or no restriction applies. Second, the assertions of a federation policy provide details on the utilities and resources that are used for validating the device's scope.

Whenever an employee attempts to federate mobile devices in order to transfer application tasks or data, e.g. transfers confidential order information from an email attachment on a mobile phone to a laptop, the WiTness framework checks the federation policy associated with the task; it retrieves the mobile device's trust certificate and performs the mentioned validation (Figure 4). Upon successful validation of policies and certificates, the application task can be transferred to the federated device for execution. [26] discusses this example in more detail. The validation of delegation policies is performed in a similar way, but additionally makes use of authorization certificates.

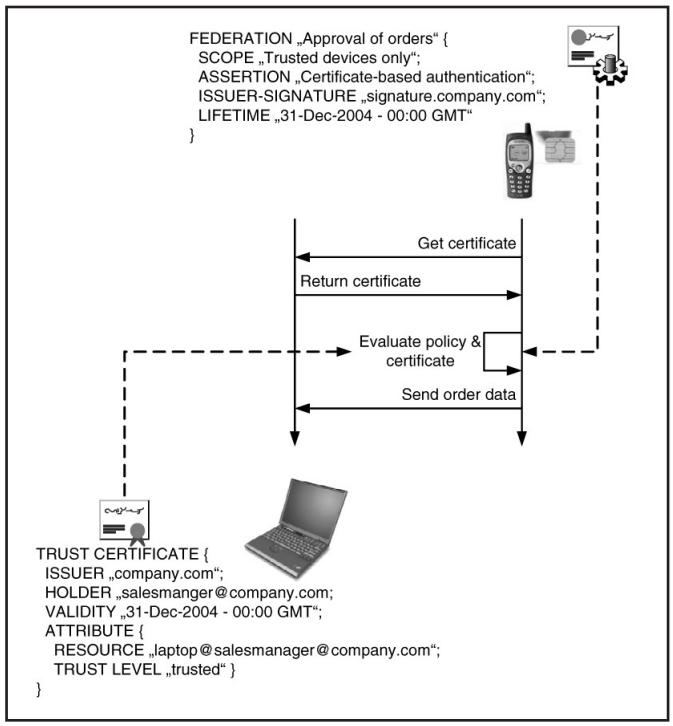

Figure 4: Policy enforcement – Federation policy and trust certificate

3.6. Connector

The connector is a framework component facilitating an abstraction from different network bearers for business applications.

A connector is always used as a communication endpoint within federated secure mobile business applications. The advantage for application design and development is clearly that special properties of network bearers used to communicate in a federation are hidden from the application logic. Regardless of the bearer, e.g. Bluetooth [3] or Wireless LAN [14], an application always uses the same communication methods offered by the connector.

As the dedicated framework component for communication handling, a connector also is the ideal enforcement point for security policies regarding communication security. Communication security policies state certain requirements about communication security which have to be fulfilled in order to establish a communication channel between applications.

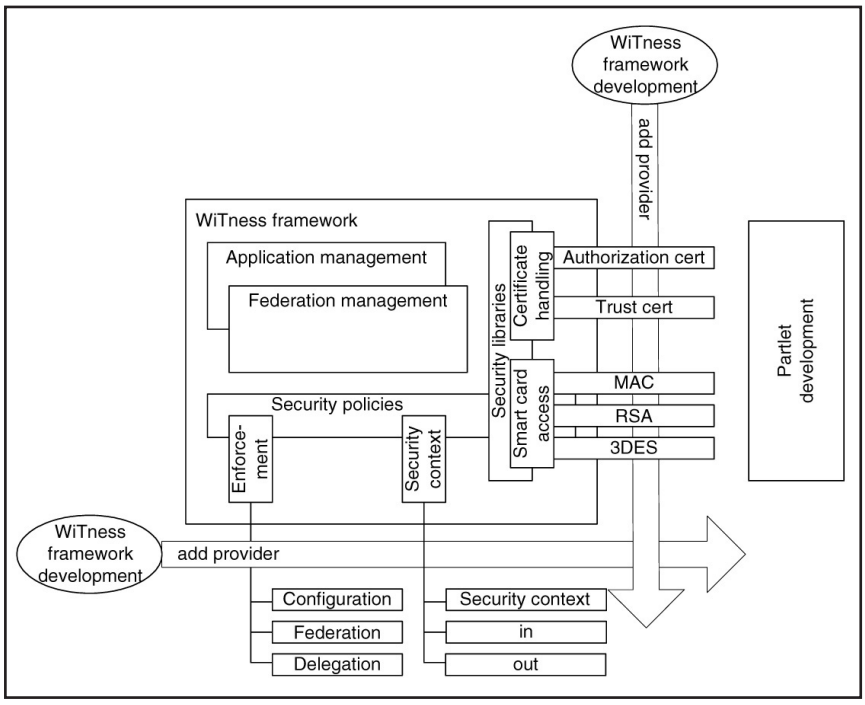

Figure 5: WiTness framework development – separation of concern

The instantiation of a connector is orchestrated by security logic, ultimately in the environment controlled by the application manager. Therefore, the application manager is still able to control communication and more importantly, is still able to override policies which forbid certain types of communication.

3.7. Development of the framework

The modular structure of the WiTness application framework supports several levels of separation of concerns. First, separation of business and security logic, which is further discussed in Section [4]. Second, the framework separates mechanism definition from its parameterization during application development as shown in Figure 5. For the framework, this separation means introducing a possibility to add, update or augment additional functionality by adding, updating or augmenting respective providers, i.e. to centrally offer frequently used security mechanisms as well as to adapt existing security policies and their enforcement procedures. A new security mechanism is simply supported by installing its respective implementation in the security library. To use the new feature, one may change a given policy, add a constraint, and bind the respective security provider installed in the previous step. The updated policy is sent to specific mobile devices, installation is done instantaneously, and the policy, and thus the new security feature, becomes active as soon as the bundled partlet is executed.

4. Use case – secure mobile workflow

Typically, applications are designed and implemented with only functionality (i.e. business logic) in mind but it would often be advantageous to separate different

business logic segments, serving different purposes. The WiTness framework introduces a solution for the separation of *business* and *security logic*. This separation of concerns brings an immediate advantage for application developers: they can solely concentrate on business logic functionality.

We have used the WiTness framework to implement a secure mobile workflow application which served to prove how to separate business and security logic; the former is encapsulated in specific partlets installed on the client device as well as on the server, while the latter is centralized in corporate defined security policy files.

The workflow models a simple authorization process. An employee submits an authorization request which is stored in a database (Figure 6). Under control of a mobile workflow manager the authorization request is passed through a sequence of approval steps. At each step, a manager may either approve or reject a request. Rejection of a request immediately terminates the workflow; otherwise the workflow terminates if the top manager in the hierarchy approves the authorization request.

The instantiation of the WiTness framework for the specific application is as follows:

- Application management as an execution environment for either client as well as for CAP (corporate access point) partlet;
- Security libraries for encryption, decryption, and data integrity;
- Certificate handling for access to certificates – public key as well as CA certificates;
- Smart card access for handling of smart card operation requests (client site only); and
- Network control, which provides access to lower layer communication handlers (such as sockets [18]).

Enterprise security policies require that the client and the server mutually authenticate. These policies also require that exchange of data is protected for confidentiality and data integrity. Approval of an authorization request is performed by

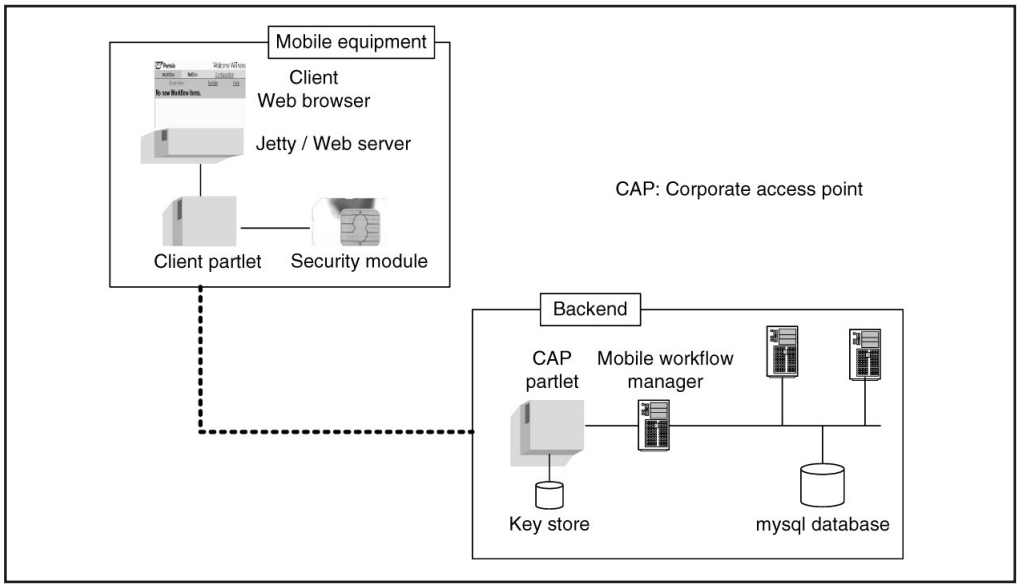

Figure 6: Mobile workflow – Mobile equipment and backend

digitally signing the authorization request. We build upon security mechanisms available which are supported on client as well as corporate side, as detailed below.

4.1. Mobile equipment and client software

The hardware platform in the personal domain consists of a laptop equipped with a smart card reader facilitating low level access to the security module implemented as WIM smart card [12]. The WIM smart card provides the secure environment for the user's private key and can perform the digital signature operation.

The application typically runs in a Windows® environment because of a native library containing the methods for smart card access. Furthermore, JDK 1.4.2 [20] is used. Part of the security library is provided by a Bouncy Castle Java® Cryptographic Extension (JCE) crypto provider [8]. It is complemented with the necessary WiTness extensions to make use of the smart card functionality from within a Java Virtual machine (JVM).

The application itself consists of a partlet implementing the functionality for the client application creating authorization requests and for the workflow inbox where all processed and pending workflow items are shown. The partlet is executed in the WiTness framework context based on a small application server deployed on the mobile device itself, called *Jetty* [6].

4.2. Application server and corporate access point

The corporate domain consists of two components, an application server hosting the corporate access point (CAP) and the information provider system (Figure 6), i.e. a database. The CAP is implemented as a partlet and is the endpoint facing partlets from the personal domain.

The CAP is responsible for security functionality on the application layer. Any communication channel to and from the corporate domain firstly requires mutual authentication; and secondly a session is created and secured by a session key (see [17] for an in-depth discussion of cryptographic methods and network security). The endpoints of this communication channel are each located within the WiTness secure mobile application framework, i.e. within a partlet.

The CAP partlet implementation is executed as a standalone Java application implementing the security logic and using its own application management

The so-called mobile workflow manager includes an implementation of the workflow model (the workflow steps) and keeps track of all the workflow states. The mobile workflow manager is designed with flexibility in mind not only regarding communication with different personal domain connections in mind, but also with different information provider systems. In this case, the mobile workflow manager uses a relational database [9] without integrated application logic as its information provider system.

5. Related work

5.1. SIM cards as ubiquitous crypto-processors

GSM SIM cards aim at protecting mobile network operators and belong to them. The widespread availability of Java-enabled SIM cards opens this model by allowing the execution of personalized applications in a secure environment. Numerous projects and applications are based on those new

features. It is worth noting that different proposals like WLAN-SIM [27], EAP/SIM [5], or SecurID [15] use SIM cards to secure user or corporate assets. The use of smart cards as integrated cryptographic processors in business applications, such as Netscape [10], is becoming increasingly common., but although their acceptance for such use can be observed, the main hindrance to their widespread use is the cost or unavailability of readers; the WiTness approach makes it possible to forego this issue thanks to the federation concept.

5.2. Secure platform for mobile applications

It is becoming common to develop applications to be deployed in mobile environments. Java 2 Micro Edition (J2ME) [19] has been proposed to write and to deploy applications in mobile environments, although it makes little account of security, especially from a distributed application perspective. The Small Terminal Interoperability Platform (STIP) project [21] provides a framework for writing secure applications on mobile devices with limited resources such as payphones, parking meters, or vending machines. WiTness goes one step beyond by securing a distributed platform that takes federations into account and by extending the scope to mobile business applications.

5.3. Trust evaluation of surrounding devices

The execution of mobile code must be secured, both in terms of integrity of execution as well as in terms of confidentiality of execution. Theoretical work has shown that results can be achieved without depending on a reliable environment [15]. However, those approaches are computationally expensive and restricted to specific cases (e.g. protecting Boolean circuits). A more realistic approach is to rely on some trust in the execution environment. It is possible to distribute some secure hardware that will run the mobile code, e.g. SIM cards or secure coprocessors [25]. Alternately, it is possible to certify some environment so that one can check that it is secure to upload a piece of code, for instance, based on TCG [22]. In comparison, the approach taken in WiTness differs by three features: (1) the secure hardware modules used are SIMs or USIMs, which are ubiquitously available in most mobile terminals, unlike dedicated coprocessors. (2) Secure hardware modules need not be neutral, which may imply complex certification processes with a central authority. Instead the secure module can be owned and managed by the company that equips its mobile workforce itself. (3) WiTness focuses on interactions with other entities, not on the separation from other entities, i.e. this approach makes pervasive scenarios more explicit.

5.4. Security policies for mobile applications

Generic Policy Languages such as Ponder [13] are sometimes replaced by dedicated policy languages that are simpler but less flexible. Some policy languages are dedicated to security (RBAC [16]). Defining a policy language dedicated to security in a mobile context is challenging. WiTness' policy language is based on Ponder.

6. Conclusion

The number of mobile employees is ever increasing and the demand for supporting them with access to corporate networks is one that must be met. We presented and discussed an architecture and framework that provides a comprehensive and evolutionary approach for the

implementation of secure mobile business applications. The architecture and framework allows for an integration of existing technologies (if useful, e.g. Bouncy castle Java Cryptographic Extension) as well as for their complementation (if required – WiTness security module); overall it consists in an open architecture and framework based on standards (e.g. WIM smart card specification).

As was proven by the implementation of a use case, the WiTness framework provides the flexibility that only parts of its elements may be employed in a real implementation.

We regard as a particularly important contribution of our framework that we have achieved a separation of concerns that supports the separation of business and security logic. It allows for a dynamic update of security mechanisms without the need to re-install the whole business application.

As further work we regard as most important the generalization of the architecture and framework so that other business application implementation paradigms can be supported.

Acknowledgements

We would like to thank our partners in the WiTness project for fruitful discussions and contributions to the presented framework. Special thanks go to Laurent Gomez, Cedric Hébert, Lars Brückner, and Marco Voss for their major contribution to the framework implementation.

References

[1] 3GPP, Subscriber Identity Modules (SIM); Functional characteristics (GSM 02.17 version 8.0.0 Release 1999), GSM 02.17, November 1999.

[2] 3GPP, USIM and IC card requirements (Release 6), SGPP TS 21.111, March 2004.

[3] Bluetooth.org, Bluetooth Specification including Core v1.2, www.bluetooth.org, November 2003.

[4] Regina Casonato, Mobility and Business-to-Employee Applications, Gartner Research, Note Number LE-16-2297, April 2002.

[5] H. Haverinen, J. Salowey. Extensible Authentication Protocol Method for GSM Subscriber Identity Modules (EAP-SIM). April 5th, 2004. draft-haverinen-pppext-eap-sim-13.txt

[6] jetty://, Jetty:// Web Server – Servlet Container, jetty.mortbay.org/jetty/index.html.

[7] G.Kiczales, J.Lamping, A.Mendhekar, C.Maeda, C.V.Lopes, J.M.Loingtier, and J.Irwin. Aspect-Oriented Programming, in ECOOP'97 Proceedings, LNCS, pages 220-242, 1997.

[8] Legion of the Bouncy Castle, bcprov-jdk14-123.jar, www.bouncycastle.org.

[9] MySQL, MySQL database server and standard clients, http://dev.mysql.com/downloads/.

[10] Netscape DevEdge web site. Smart Cards, developer.netscape.com/tech/security/certs/cards.html

[11] S. Oaks, B. Traversat, L. Gong, JXTA in a Nutshell, O'Reilly, September 2002.

[12] Open Mobile Alliance, Wireless Identity Module Specification, WAP-260-WIM-20011207-a, www.openmobilealliance.org/tech/affiliates/wap/wapindex.html, 2001.

[13] Policy Research Group, Ponder Policy Framework, www-dse.doc.ic.ac.uk/Research/policies/index.shtml.

[14] N. R. Prasad, A.R. Prasad (Eds.), WLAN Systems and Wireless IP for Next Generation Communication, Arthech House, 2002.

[15] RSA Security, RSA SecurID Authentication, www.rsasecurity.com/node.asp?id=1156.

[16] R. S. Sandhu, E. J. Coyne et al., Role-Based Access Control Models, IEEE Computer 29(2), 1996.

[17] W. Stallings, Cryptography and Network Security – Principles and Practice, Prentice Hall, 1999.

[18] W. Richard Stevens, UNIX network programming, Prentice Hall, 1990.

[19] Sun Microsystems, Connected Limited Device Configuration, Specification v1.1, March 2003, java.sun.com/j2me.

[20] Sun Microsystems, Java 2 Platform, Standard Edition – version 1.4.2, java.sun.com/j2se/1.4.2/download.html.

[21] STIP Consortium, STIP Specification 2.1 Overview, 2002, www.stip.org.

[22] The Trusted Computing Platform Alliance, Building a Foundation of Trust in the PC, White paper, www.trustedcomputing.org, January 2000.

[23] B. H. Walke, Mobile Radio Networks – Networking, Protocols and Traffic Performance – Second Edition, Wiley, 2002.

[24] WiTness consortium, WiTness – Wireless Trust for Mobile Business, www.wireless-trust.org.

[25] B. S. Yee, Using Secure Coprocessors, PhD thesis, Carnegie Mellon University, 1994.

[26] T. Walter, L. Bussard, P. Robinson, Y. Roudier, Security and trust issues in ubiquitous environments – the business-to-employee dimension, SAINT 2004 Symposium on Applications and the Internet, Tokyo, January 2004.

[27] WLAN Smart Card Consortium. WLAN-SIM specifications (WLAN-SIM and EAP-SIM handler). www.wlansmartcard.org/specifications.html