Security in the Development Process of Mobile Grid Systems

David G. Rosado[1], Eduardo Fernández-Medina[1] and Javier López[2]
[1]University of Castilla-La Mancha. GSyA Research Group, Ciudad Real
[2]University of Málaga. Computer Science Department, Málaga
Spain

1. Introduction

Grid computing has emerged to cater the need of computing-on-demand (Jana et al., 2009) due to the advent of distributed computing with sophisticated load balancing, distributed data and concurrent computing power using clustered servers. The Grid enables resource sharing and dynamic allocation of computational resources, thus increasing access to distributed data, promoting operational flexibility and collaboration, and allowing service providers to scale efficiently to meet variable demands (Foster & Kesselman, 2004).

Security is considered as the most significant challenge for Grid computing (Humphrey et al., 2005), due to the fact that resources are shared between organizations; expensive resources, that may go from computers and other hardware facilities, to potentially valuable, sensitive and confidential data files.

In recent years the mobile computing community has been successful in utilising academic and industry research efforts to bring products to the commercial market. We have seen a proliferation of consumer electronic devices taking advantage of wireless technology enrich our daily lives with increased productivity thanks to higher connectivity.

At first glance, it seems that the marriage of mobile wireless consumer devices with high-performance Grid computing would be an unlikely match. After all, Grid computing to date has utilised multiprocessors and PCs as the computing nodes within its mesh. Consumer computing devices such as laptops and PDAs are typically restricted by reduced CPU, memory, secondary storage, and bandwidth capabilities. However, therein lies the challenge. The availability of wirelessly connected mobile devices has grown considerably within recent years, creating an enormous collective untapped potential for resource utilisation. To wit, recent market research shows that in 2008, 269 million mobile phone and 36 million smartphone (Gartner, 2009) were sold worldwide, and that in 2006, 17 million PDAs (Gartner, 2007) were sold worldwide. Although these individual computing devices may be resource-limited in isolation, as an aggregated sum, they have the potential to play a vital role within Grid computing (Phan et al., 2005).

On the other hand, the idea of developing software through systematic development processes to improve software quality is not new. Nevertheless, there are still many information systems such as the Grid Computing ones, that are not developed through methodologies adapted to their most differentiating features (Kolonay & Sobolewski, 2004). That is to say, generic development processes are used to develop specific systems without taking into consideration either the subjacent technological environment or the special features and particularities of these specific systems.

Additionally, the growing need for constructing secure systems, mainly due to the new vulnerabilities derived from the use of the Internet and that of the applications distributed in heterogeneous environments, encourages the scientific community to demand a clear integration of security into the development processes (Bass et al., 2004; Breu et al., 2003; Haley et al., 2006; Jürjens, 2005; Lodderstedt et al., 2002; Mouratidis & Giorgini, 2006). The main reason is that security aspects are only considered at the implementation stages causing that security solutions are not perfectly coupled with the design and the rest of requirements of the system (Artelsmair and Wagner, 2003; Mouratidis & Giorgini, 2006).

Therefore, the goal of the paper is to define a systematic development process for Grid systems that supports the participation of mobile nodes and incorporates security aspects into of all sofware lifecycle will thus play a significant role in the development of systems based on Grid computing. The reasons that led us to focus on this topic are several: Firstly, the lack of adequate development methods for this kind of systems since the majority of existing Grid applications have been built without a systematic development process and are based on ad-hoc developments (Dail et al., 2004; Kolonay & Sobolewski, 2004), suggests the need for adapted development methodologies (Giorgini et al., 2007; Graham, 2006; Jacobson et al., 1999; Open Group, 2009). Secondly, due to the fact that the resources in a Grid are expensive, dynamic, heterogeneous, geographically located and under the control of multiple administrative domains (Bhanwar & Bawa, 2008), and the tasks accomplished and the information exchanged are confidential and sensitive, the security of these systems is hard to achieve. And thirdly, because of the appearance of a new technology where security is fundamental together with the advances that mobile computation has experienced in recent years that have increased the difficulty of incorporating mobile devices into a Grid environment (Guan et al., 2005; Jameel et al., 2005; Kumar & Qureshi, 2008; Kwok-Yan et al., 2004; Sajjad et al., 2005).

The rest of paper is organized as follows: In section 2, we will present the related work. In section 3, we will define the proposed development process with the models used and activities and tasks. In section 4 we will apply the proposed process to a real case showing the results obtained. We will finish by putting forward our conclusions as well as some research lines for our future work in section 5.

2. Related work

There are some proposals which try to integrate security into the software development process, even from the first stages, but however, none of them are defined for Grid Computing based systems. For instance, authors in (Steel et al., 2005) present a methodology for the integration of the security on software systems. This methodology is based in the Unified Process (Kruchten, 2000) and it is called Secure Unified Process (SUP). The problem is that it only offers a solution to a very high level without offering "practical mechanisms" (e.g. Grid-specific security artifacts or a security architecture of reference) that permits to implement his approach in a short space of time and with minimal effort. Other approach (Jurjens, 2001; Jurjens, 2002; Jürjens, 2005) concentrates on providing a formal semantics for UML to integrate security considerations into the software design process. UMLSec is more focused on access control policies and how these policies can be integrated into a model-driven software development process (Mouratidis et al., 2005). This aspect is important but is very specific and it is applicable only to certain stages of the development process. Other approach is CLASP (Comprehensive, Lightweight Application Security Process) that is an activity-driven, role-based set of process components guided by formalized best practices (Graham, 2006). CLASP suggests a number of different activities across the development lifecycle in order to improve security, but does not define a whole development process only the activities which can be integrated into any development process. Finally, AEGIS (Appropriate and Effective Guidance for Information Security) (Flechais et al., 2003) is a secure software engineering method that integrates security requirements elicitation, risk analysis and context of use, bound together through the use of UML. This approach is the only approach found in which the authors attempt to apply the methodology to Grid systems, although they do not explain how to do this, and do not define guides and practices for capturing specific security aspects in Grid systems.

On the other hand, there are numerous approaches related to Grid architectures, middleware, infrastructures, projects, and so on, such as OGSA (Foster et al., 2002) that is a specification that defines a set of concepts, rules, features, properties, services, capacities and behaviour for implementing Grid Computing applications, but it does not indicate the steps and methods to develop for obtaining a Grid application. Other approach is UNICORE (Uniform Interface to Computing Resources) (Romberg, 2002) which develops a software infrastructure for seamless access to distributed supercomputer resources. The problem is that does not support either dynamic environment or mobile devices. The gLite approach (EGEE Middleware Design Team, 2004) provides a framework for building Grid applications tapping into the power of distributed computing and storage resources across the Internet. These security features are sufficient for many Grid applications, but insufficient when the applications are more complex and even more when mobile devices are included in the application as Grid resources. Also, the Legion project, developed at the University of Virginia, is an attempt to provide GRID services that create the illusion of a virtual machine. This architecture does not support the dynamic behaviour of the Grid.

Finally, some of the most interesting approaches for the incorporation of mobile devices into Grid systems are: LEECH (Leveraging Every Existing Computer out tHere) (Phan et al., 2005) that takes into account security aspects for wireless network and mobile devices through the proxy which serves as an intermediary between mobile devices and the Grid. This proxy must additionally be protected to safeguard Grid systems with mobile devices so that is a weak spot by the attackers. Mobile-To-Grid (Jameel et al., 2005; Kalim et al., 2005; Sajjad et al., 2005) is other approach that defines a middleware which permits heterogeneous mobile devices access to Grid services. This approach treats mobile devices like external elements and the security must be implemented outside the Grid environment. Other of the approaches is the Surrogate approach wich defines a middleware that will allow handheld devices, e.g. PDA units, to interact with Grid services while inducing minimal burden on the device itself (Riaz et al., 2005). Security to Grid level is not considered, only to gateway level that is not sufficient for the complexity of Mobile Grid systems with thousand of mobile resources and different VOs and domains. Finally, the Akogrimo architecture (Access to Knowledge through the Grid in a Mobile World) (The AKOGRIMO project) is intended to support business process designs that both support and take advantage of dynamic, mobile services and users. This approach defines a design and implementation solution for Grid with mobile devices but it is not development guides or methodologies to apply these solutions of controlled and systematic way.

Therefore, after studying and analyzing each of the approaches related to the development and security in Mobile Grid computing, we conclude that the existing proposals are not specific enough to provide a complete solution of security under a systematic development process for Mobile Grid environments. This is due to the fact that none of the approaches defines a systematic development process for this specific kind of systems that incorporates security from the earliest stages of the development. Moreover, the existing security approaches to Mobile Grid computing are more focused on offering a technological solution than on the requirements and design. Neither of them offers solutions that integrate mobile devices as own resources of the Grid under a global security environment.

3. Secure development process

3.1 Overview

This process is designed for building software systems based on Mobile Grid computing with security aspects. It is a process which builds, from initial requirements and needs of Mobile Grid systems, a secure executable software product. It is not a process for including only security in a development process but it is a development process in itself incorporating security aspects during all the process.

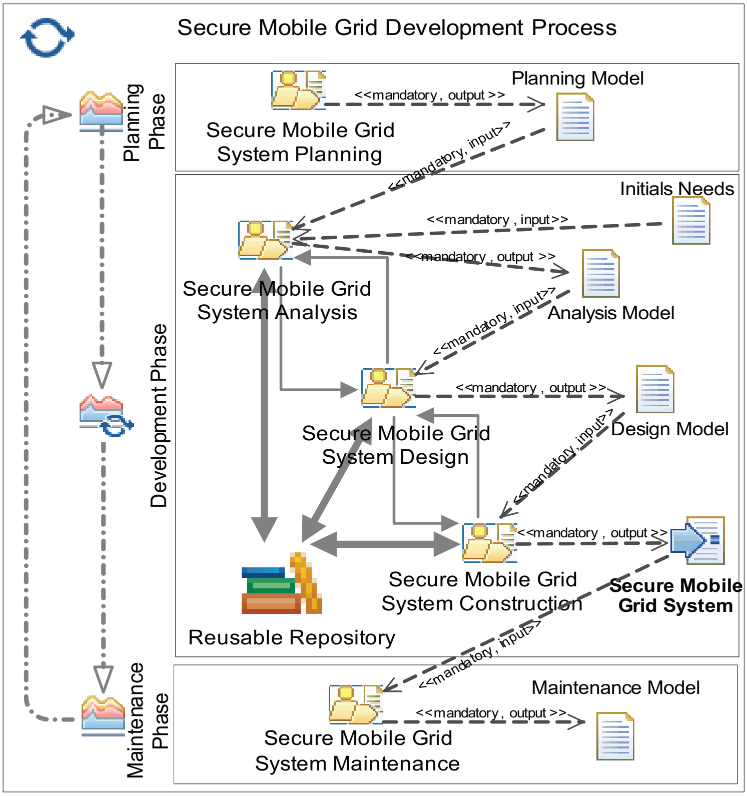


Fig. 1. Development process for secure Mobile Grid systems with SPEM 2.0

The structure of the process which we propose follows the classical cycle (see Fig. 1), in which we find a planning phase, a development phase including analysis, design and construction and finally a maintenance phase. However, the proposed process is specially designed for this kind of systems because we consider their particular aspects and features in each of the activities of the process

Our systematic process of development is an iterative and incremental process. An iterative approach refers to the cyclic nature of the process in which activities are repeated in a structured manner and proposes an understanding of the problem through successive refinements, and an incremental growth of an effective solution through several versions. Thus, in each iteration of the process, new and necessary characteristics can be added and extended so that a complete final design is obtained. Also, it is a reusable process in the sense of the utilization of artifacts built in others executions of this process or in previous iterations which have been validated and tested and that improve the quality of the new artifacts built and save developers' time and effort.

3.2 Models of the process

This section presents the defined models identified for the process and which are necessary for carrying out the different activities and tasks. These models are: i) an extension of UML (GridUCSec-profile) to define Grid use cases that include security use cases and misuse cases oriented to the characteristics of Mobile Grid systems, and that are used in the analysis activity (Rosado et al., 2009a; Rosado et al., 2010b; Rosado et al., 2010c); ii) the reference security architecture used in the design activity, which ensures the fulfillment of the security requirements for these systems and that is service-oriented (Rosado et al., 2010a; Rosado et al., 2010d); iii) the association rules model between security requirements and security services for identifying what services are necessary for fulfilling what requirements, and it is used in the design activity when the security services are defined; and finally, iv) the reusable elements of the repository that are used in the different tasks of the process and that help us build artifacts in an easy, fast and reliable way (Rosado et al., 2009a; Rosado et al., 2009b).

3.3 Activities of the process

As we have seen above, the process consists of 3 phases, planning, development and maintenance. The phases of planning and maintenance are common phases which any development of information systems has to define, so we move on a generic development process to carry out the activities and tasks of these phases. Thus, our work focuses on defining what is really specific and differentiating in developing systems based on Grid computing, the development phase. This phase consists of three activities, analysis, design and construction, and each of them defines the specific tasks necessary, the artifacts to be used, and the steps to take to analyze, design and build specific information systems as Mobile Grid systems are.

3.3.1 Analysis activity

Analysis focuses on ensuring that the system's security and functional requirements are elicited, specified and modelled. In our approach, this activity is driven by use cases and supported by the reusable repository. This obtains, builds, defines and refines the use cases of the secure Mobile Grid systems which represent the functional and non-functional requirements of this kind of systems. Both the wide set of elements which are common to these systems and that are stored in the repository, such as secure Mobile Grid use cases, interaction diagrams, UML profiles, templates, etc., and the GridUCSec-profile model aforementioned, help the analyst define all the requirements (functional and non-functional, and security in particular) and build the necessary diagrams with which to complete the analysis activity from beginning to end.

The analysis activity is based on use cases in which we define the behaviour, actions and interactions with those implied by the system (actors) to obtain a first approach to the needs and requirements (functional and non-functional) of the system to be constructed. This activity is supported by repositories in which several types of elements appear: Firstly, the elements that have been developed in earlier stages; secondly, those that have been built at the beginning of the process and finally, those that come from other executions of the process from which we have obtained elements that can be reused by other applications. Reuse is appropriate here thanks both to the common features of applications based on Grid computing (CPU intensive, data intensive, collaborative and so on) and to the fact that these applications use mobile devices. Therefore, we must abstract all the common features (by analyzing the main features of Grid applications and constructing, for example, generic use case diagrams in which all these common features are represented) and make them available for the process (through the repository) in order to be able to use the common elements in any activity and adapt them to our needs.

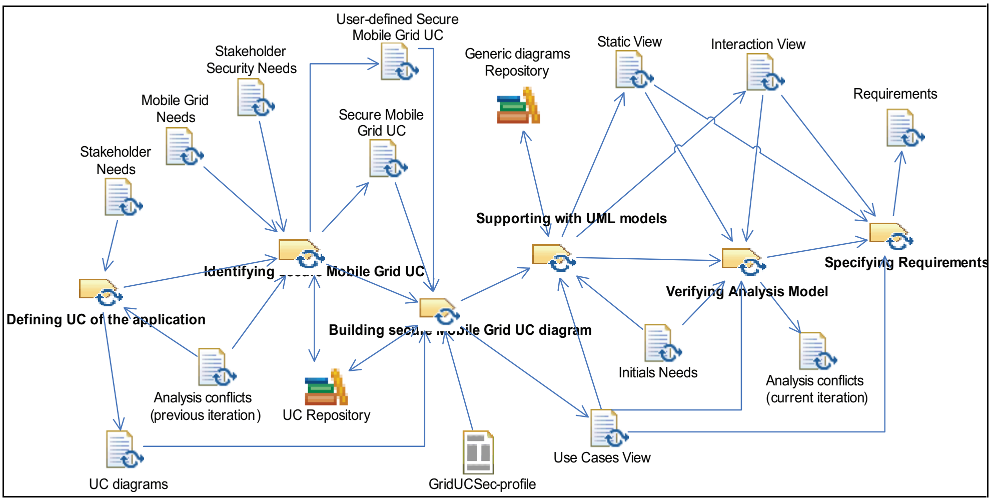

User-defined Secure Mobile Grid UC
Stakeholder Security Needs
Generic diagrams Repository
Static View
Interaction View
Requirements
Mobile Grid Needs
Secure Mobile Grid UC
Stakeholder Needs
Supporting with UML models
Identifying Mobile Grid UC
Specifying Requirements
Defining UC of the application
Verifying Analysis Model
Building secu bile Grid UC diagram
Initials Needs
Analysis conflicts (current iteration)
Analysis conflicts (previous iteration)
UC Repository
Use Cases View
UC diagrams
GridUCSec-profile

Fig. 2. Tasks and artifacts of the Secure Mobile Grid System Analysis activity

The analysis activity is composed of tasks which build use case diagrams and specifications to obtain the analysis model in which the requirements are defined. This activity produces internal artifacts which are the output of some tasks and the input of others. Fig. 2 shows a graphical representation of the analysis activity tasks using SPEM 2.0 diagrams. Initially, in the "Defining UC of the application" task, we define the functional use cases of the application identified from the stakeholder needs and study the interactions with the user without considering the specific aspects of Mobile Grid environments. Next, in the

"Identifying secure Mobile Grid UC" task, we study the security aspects of the application within the Mobile Grid context and identify the possible security use cases and misuse cases that can be reused from those defined in the repository, for the system in development. Once the use cases have been identified and defined, we build the overall use case diagram (or diagrams) in which we define the relationships between all the use cases and actors previously identified, and we describe the information from all the diagram's elements by following a new UML profile for Mobile Grid use cases. We can also reuse and integrate some diagrams with common features of the repository which have been previously built for Mobile Grid environments. This is carried out in the "Building secure Mobile Grid UC diagram" task. In the "Supporting with UML models" task, we complete the analysis model with different UML models such as the sequence and collaboration diagrams according to use cases and scenarios, or class diagrams for an initial structural description of the system from the use cases diagrams built in previous tasks. Then, in the "Verifying Analysis model" task, we have to verify that the artifacts have been correctly generated and the possible conflicts or errors in the analysis model have to be identified and analyzed for their subsequent refinements and corrections in next iterations of this activity. Finally, the "Specifying Requirements" task consists of the formal definition of the requirements identified in previous tasks (functional requirements and non-functional requirements including security) in natural language (though a template of requirements specification will be defined in the future).

3.3.2 Design activity

Design focuses on ensuring that the system's security and functional requirements are fulfilled, covered and validated with the design, on the one hand, of a security architecture, and the other hand, of a software architecture. In our approach, this activity is supported by a reference security architecture which covers and fulfills the security requirements of the Mobile Grid system specified in the analysis activity. This reference security architecture is instantiated to a concrete security architecture (depending on the security requirements required) and is integrated with the software architecture designed (using a generic development process as the Unified Process), obtaining a Secure Software Architecture for Mobile Grid systems which is an input artifact for the construction activity.

The design activity is centred on building the secure software architecture of the system through UML views, services and a reference security architecture stored in the repository. The design activity is centred on architecture that is the main element of the system and, on the one hand, it covers the necessities and requirements identified and specified in the analysis activity, and on the other hand, it serves as a guide in the next activity of system construction. The designed software architecture will be a secure architecture because we have incorporated a security architecture for this kind of systems into the general software architecture fullfiling with the security requirements specified in the previous activity. This activity is supported by repositories in which several types of elements appear: Firstly, the elements that have been developed in earlier stages; secondly, those that have been built at the beginning of the process and finally, those that come from other executions of the process from which we have obtained elements that can be reused by other applications. A reference security architecture has been defined in the repository for its use together with UML diagrams for the system construction.

The design activity is composed of tasks which build the software architecture, the security architecture and their specifications with different views and using architectural elements of the repository, obtaining the design model where the architecture is defined. Fig. 3 shows a graphical representation of the design activity tasks using SPEM 2.0 diagrams. Initially, the "Designing Mobile Grid Software Architecture" task designs a software architecture following the steps, methods and techniques of the typical development processes (as the Unified Process (Jacobson et al., 1999), TOGAF (Open Group, 2009), etc.) using UML diagrams, views and realizations of use cases from the use cases model and analysis model defined in the analysis activity for Mobile Grid systems. The "Designing Mobile Grid Security Architecture" task defines the security architecture using the reusable elements of the repository and following the security Requirement-Service association rules of use cases to security services from the use cases of the analysis model. The reusable elements are generic elements which should be instantiated and integrated into the security architecture that is built with the help of security patterns. Once we have built both the software architecture and the security architecture, now we have to build and define the final architecture that is the security architecture integrated into the software architecture defining the relationships between elements (classes, collaboration, sequence, objects, diagrams, etc.), obtaining the secure Mobile Grid software architecture. This is executed in the "Designing/Integrating Secure Mobile Grid Software Architecture" task. Next, in the "Specifying Secure Mobile Grid Software Architecture" task, the final architecture built is specified using natural language or the IEEE 1471-2000 standard (IEEE, 2000). Last, in the "Validating and Verifying Secure Mobile Grid Software Architecture" task, the architecture obtained in the previous task has to be validated with the requirements specified in the analysis activity and the traceability between artifacts has to be verified, and the possible conflicts or errors in the design have to be identified and analyzed for their subsequent refinements in next iterations of this activity.

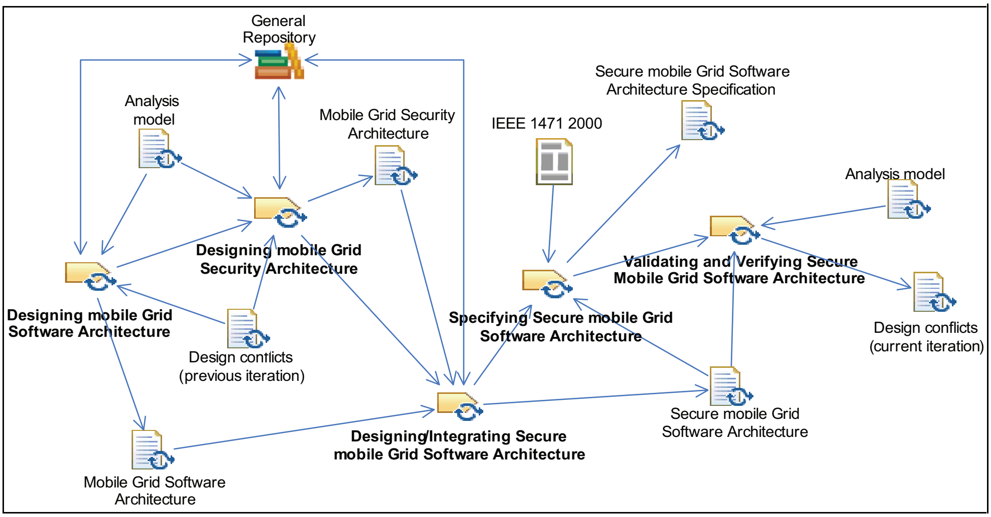

General Repository
Analysis model
Mobile Grid Security Architecture
IEEE 1471 2000
Secure mobile Grid Software Architecture Specification
Analysis model
Designing mobile Grid Security Architecture
Validating and Verifying Secure Mobile Grid Software Architecture
Designing mobile Grid Software Architecture
Design conflicts (previous iteration)
Specifying Secure mobile Grid Software Architecture
Design conflicts (current iteration)
Secure mobile Grid Software Architecture
Designing/Integrating Secure mobile Grid Software Architecture
Mobile Grid Software Architecture

Fig. 3. Tasks of the Secure Mobile Grid System Design activity

3.3.3 Construction activity

Construction focuses on implementing the elements designed in the previous activity using a technological platform for the building of Grid systems. Mainly, the goal of this activity is to generate code of services and interfaces of the secure software architecture and to configure and compile files, data, packages, interfaces and classes under a Grid platform which gives life to the design and makes the system available for users, resources and organizations to be able to communicate between them sending tasks and obtaining information from the Grid.

The construction activity is centred on implementing the architecture designed in the design activity through a technological platform for Grid computing previously studied whose tools and libraries can be obtained from the implementation repository. The construction activity consists of implementing the designed architecture (interfaces, components, classes, etc.) under a technological environment related to Grid computing and supported by mobile computing. This environment will be installed and configured so that it helps us implement the design model with the tools, mechanisms, methods and libraries available in the selected technological environment. In the repository the most used tools to build this kind of systems together with common services implemented in different programming languages, generated in other executions of the process are available. The aim of this activity is to obtain the final product, the secure Mobile Grid system, depending on a specific technological environment for Grid computing.

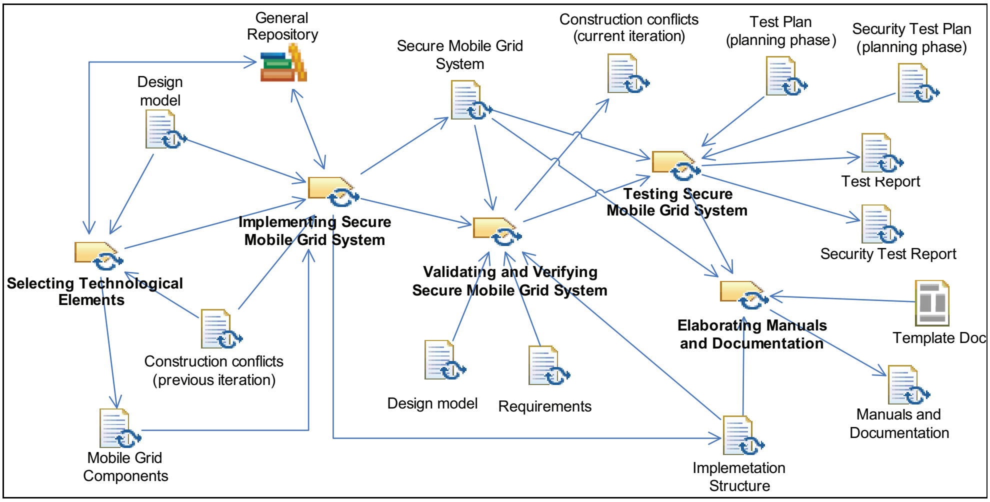

General Repository
Design model
Secure Mobile Grid System
Construction conflicts (current iteration)
Test Plan (planning phase)
Security Test Plan (planning phase)
Test Report
Testing Secure Mobile Grid System
Implementing Secure Mobile Grid System
Selecting Technological Elements
Validating and Verifying Secure Mobile Grid System
Security Test Report
Elaborating Manuals and Documentation
Template Doc
Construction conflicts (previous iteration)
Design model
Requirements
Manuals and Documentation
Mobile Grid Components
Implemetation Structure

Fig. 4. Tasks of the Secure Mobile Grid System Construction activity

The construction activity is composed of tasks which implement the designed secure software architecture for obtaining an executable final product with the help of the tools and technologies available for Mobile Grid computing. Fig. 4 shows a graphical representation of the construction activity tasks using SPEM 2.0 diagrams. Initially, the "Selecting Technological Elements" task refers to study and analyze the different current technological approaches for Grid computing (Globus, gLite, PERMIS, VOMS, etc.) and choose the most appropriate that facilitates the implementation of the system. This choice together with the tools requirements and features are defined in the Mobile Grid Components artifact. The "Implementing Secure Mobile Grid System" task is the main task of this activity and converts the design elements into executable and deployed components running in different resources of different organizations. To implement the secure software architecture, we must take into account the chosen technological components and the specified aspects in the design activity. Once we have implemented the services interfaces, classes, services and protocols under the technological environment, we will generate executable code (files, packages, executable classes, objects, etc.) which represents the designed architecture and that offers solutions to the necessities and requirements initially raised, especially of security, for this kind of systems. Also, we must describe and indicate, in the implementation structure artifact, the decisions of implementation carried out as the generated files of a class, the relations between files and packages, the final deployment structure, what services interfaces and operations are being implemented and with what programming language, used communications protocols, exchanged messages format, standards and specifications used, and so on. Then, with the implemented architecture, the design model and the requirements, in the "Validating and Verifying Secure Mobile Grid System" task we validate that the system implementation fulfills and covers the requirements and we verify the right traceability from design to implementation. The tests and security tests of all components of the implemented system have to be checked in the "Testing Secure Mobile Grid System" activity, obtaining the possible failures and issues not considered in previous activities and which are important for protecting the system and for ensuring that the system is reliable and robust. Finally, we elaborate, in the "Elaborating Manuals and Documentation" activity, all the necessary documentation on how we have designed and implemented the application (decisions, configurations, tools requirements, files, languages, etc.) and the manuals of development for its maintenance and updateness, and user and execution manuals.

3.4 Prototype tool

To help in the building of use cases diagrams following the UML extension aforementioned, we have developed a tool, SMGridTool (Secure Mobile Grid Environments Tool) which provides a simple, automatic and intuitive way of building use cases diagrams especially designed for Secure Mobile Grid systems. This tool is focused on the construction and definition of secure Grid use cases diagrams, and on the management of the repository that stores reusable artifacts which can be reused in the construction of diagrams.

SMGridTool performs automatically the validation of the use cases diagrams, checking the constraints defined by the GridUCSec-profile for each use case and relationship. The tool automates the analysis activity of the development process for Secure Mobile Grid Systems, but our objective is that in future versions it will also support the design activity. To facilitate the user the completion of this activity, the prototype offers an easy to use graphic interface supported by Microsoft Visio controls for building diagrams. The use of Microsoft Visio 2007 activex controls also provides an easy way to reuse diagrams, export to other formats like images or xml files, and many more advantages.

In short, the SMGridTool has been developed to facilitate building and managing specific use cases diagrams for Mobile Grid environments which are necessary for carrying out an exhaustive and deep analysis of the needs and requirements of this kind of systems.

4. Case study

The real case study selected to apply the process is the media and journalism application that is a pilot Grid application of GREDIA, which is an FP6 project funded by the European Union and that provides a Grid application development platform, which will enable the integration of secure Grid middleware for managing mobile services and allowing mobile devices to participate in a protected data Grid as service providers, in a peer-to-peer manner, where journalists and photographers make their work available to a trusted network of peers at the moment it is produced, either from desktop or mobile devices. This pilot will bring together the market orientation and pervasiveness of mobile communication technology with the promise of a dynamically concerted use of resources and services provided through Grid infrastructures.

We want to build a system that will cater for the reporter who is on the move with lightweight equipment and wishes to capture and transmit news content. This user needs to safely and quickly upload the media to a secure server to make it easier for others to access, and to avoid situations where his/her device's battery dies or another malfunction destroys or makes his/her media unavailable.

To develop this pilot application, we will use an iterative and incremental approach so that for the end of the iterations cycle, we have developed the final product. Therefore, in a first iteration, that is shown here, we must select a part of the aims and goals of the system, of an acceptable size, so that we can apply the different activities and tasks to this part of the system and we can obtain reasonable artifacts, even a reduced version of the product. In the next iterations, that have been omitted here, this part of the system will be refined and extended with new elements and the process will again be applied to obtain a new version of the product. In the last iteration, the process will be applied to the whole system obtaining the final product.

In this case, we will describe the first iteration of the process for a set of initial needs, not all, and we will obtain in the analysis activity the functional and security requirements of the case study using the defined model GridUCSec-profile and the SMGridTool which help us build security use cases diagrams for Mobile Grids. The set of requirements have to be incorporated into the secure software architecture in the design activity and later, we will implement from this architecture, a first version of the product. We will put more emphasis on the part of security that is the original of this process omitting the software part that is generally built making use of other generic development processes.

4.1 Application of the analysis activity

The aim of this first iteration in the analysis activity is to define a set of requirements of the system (most of them) through the use cases diagrams built with the help of the SMGridTool and the repository of reusable elements of the process. We mainly focus on security requirements but without forgetting the rest of requirements.

Task A1. Defining UC of the application

Based on the initial needs and the scenarios defined for the media sector, a set of use cases have been identified along with a first level analysis. It is certainly expected that these will be refined during the design and construction activities according to a continuous feedback loop to be finalised with the final delivery of the Grid system, and with the subsequent iterations of the process. These use cases identified are: Add/edit Mobile user; Login to the system; Formulate the 'news' task; Notify Human Resources of task assignment; Search for news; Get query results; Record audio/video; Create news item; Submit news item; Join a Community; Review news item; Archive news item; Approve news item; Annotate news item; Assign news item publishing position; Display breaking news.

Task A2. Identifying secure Mobile Grid UC

Once we have identified the functional use cases of the application, now, we must identify all the use cases and security use cases for the Grid system that are related to the functional use cases of the application. These use cases for the Grid system include Grid use cases, security use cases, Grid security use cases, misuse cases and mobile use cases together with Grid actors and Misactors, all of them defined with the GridUCSec-profile. We use the reusable artifacts of the repository where many of these use cases for Grid systems and diagrams that can be easily used in this application and that help us obtain use cases, actors and associations that are necessary in this application are defined.

Step A2.1. Identify generic Grid UC for the application. We must act on the repository of Grid use cases to identify the generic Grid use cases that are needed to extract and that are related to the use cases defined in the previous task. To define the Grid use cases we will use the GridUCSec-profile defined as a model of the process and using the repository where a large set of Grid use cases are defined, we can build the Grid use case diagram, with the SMGridTool, where security use cases and misuse cases oriented to Mobile Grid are present. In the repository we have a set of generic use cases which have a common behaviour for any Grid systems and have been identified in other executions of the process and that can be used in this application. We select some of these generic Grid use cases that have relation with the functional use cases identified previously and are show in Fig. 5.

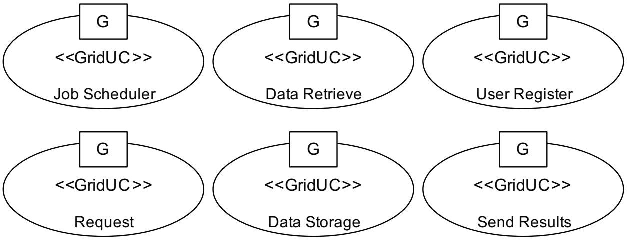
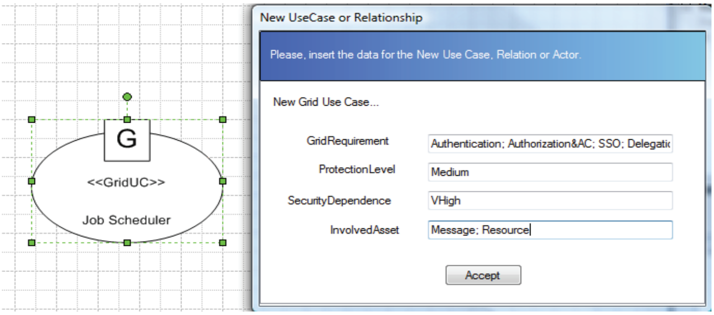


Fig. 5. Generic Grid Use Cases of the repository

These generic Grid use cases are defined based on the GridUCSec-profile, so that in the repository, we have all information related to these Grid use cases. For example, the information related to the "Job scheduler" Grid use case stored in the repository is shown in Fig. 6.



Please, insert the data for the New Use Case, Relation or Actor.

New Grid Use Case...

GridRequirement: Authentication; Authorization&AC; SSO; Delegatic

ProtectionLevel: Medium

SecurityDependence: VHigh

InvolvedAsset: Message; Resource

Fig. 6. Defining Tagged Values for "«GridUC» Job Scheduler" with SMGridTool

Step A2.2. Identify security Assets of the application in a Mobile Grid environment. In Mobile Grid environments we can identify a set of generic assets that we must protect for obtaining a secure Grid system, which are the following: User and system data (stored, transmitted); Identity information; Credentials (private keys, passwords); Accounting; CPU-/Storage-/Mobile devices-/Network-resources; General system.

In this first iteration of our case study, we define the most important assets related to the use cases aforementioned that we must protect and that are the reference for the identification of threats, attacks and security use cases. These assets are: Personal information about the journalist or editors; Media information used: photos, articles, recordings, videos, intellectual property rights; Exchange information: messages, queries, transactions.

Step A2.3. Identify Threats, Attacks and Risks of the application in a Mobile Grid environment. The set of threats and attacks that can occur in a Mobile Grid system is similar to that produced in a distributed system by adding those occurring in the mobile environment with wireless network and limited resources. Examples of threats are unauthorized disclosure of information, attacks to the content of a message through wireless links, denial-of-service attacks, network authentication related attacks, physical node attacks, alteration of information, and so on. In this first iteration, we can identify several possible types of threats to Information: Unauthorized access to Grid system; Unauthorized disclosure and alteration of information; Masquerade.

Step A2.4. Identify the Security UC and Misuse cases from the repository. Once we have defined the most significant threats and major assets to be protected in this first iteration, we start with the identification, definition and incorporation of security use cases and misuse cases for the application. In the repository, the main security use cases for Mobile Grid environments, and misuse cases that capture the behaviour of the main threats identified in these environments are defined. We can identify those security use cases and misuse cases that fit in with the attacks and threats for this application identified in the previous step.

In this first iteration, the misuse cases that we have found in the repository and that fit in with the threats identified for this application are: Alteration info, Disclosure info, Unauthorized access and Masquerade.

With these misuse cases, we can identify security use cases that mitigate them observing the information offered by the repository for security use cases and the diagrams defined where we can see the relationships of mitigation between security use cases and misuse cases. In case that the required use cases are not in the repository we can define them and specify relationships as it is convenient.

We find in the repository the security use cases (including Grid security use cases and Grid actors) that are related to the misuse cases identified. These security use cases are: Authenticate, Authorize access, Ensure Confidentiality and Ensure Integrity.

Step A2.5. Security Assessment. Finally, it is necessary to assess whether the threats are relevant according to the security level specified by the security objectives. Therefore we must estimate the security risks based on the relevant threats, their likelihood and their potential negative impacts, in other words, we have to estimate the impact (what may happen) and risk (what will probably happen) to which the assets in the system are exposed. We must therefore interpret the meaning of impact and risk. In Table 1 we define the impact and risk for some of the threats identified previously. We are going to evaluate risk and impact with five possible values: Very Low, Low, Medium, High and Very High. The likelihood of a threat could be: Very Frequent (daily event), Frequent (monthly event), Normal (once a year), Rare (once in several years).

As we can see in Table 1, all threats have to be dealt with because they cause a high or very high value of risk in the worst case, therefore, misuse cases that represent these threats must be studied and analyzed in this first iteration and will take part of the Grid use cases diagram that we will build in the next task.

Threat	**Unauthorized access to Grid system**	
Impact	MEDIUM if the authorization privileges are very limited (i.e. only reading).	VERY HIGH if the opposite is the case
Attack	Unauthorized access	
Probability	Normal	Normal
Risk	HIGH	VERY HIGH
Threat	**Unauthorized alteration of information**	
Impact	LOW if there is no personal information modified	HIGH if the opposite is the case
Attack	Modification of information	
Probability	Frequent	Frequent
Risk	LOW	HIGH

Table 1. Assessment of Impact and Risk

Task A3. Building secure Mobile Grid UC diagram

After identifying functional use cases of the application, generic Grid use cases related, misuse cases that represent the threats identified and security use cases that protect from those threats, we are able to build the overall use cases diagram and to define the relationships between them following the UML GridUCSec-profile and using the SMGridTool which help us build the diagram and manage the repository where there are defined use cases diagrams that we can incorporate into the overall diagram.

In previous tasks, we have been studying and analyzing artifacts of the repository (use cases) that define common features for Grid environments and that can be usable for this specific application, as it is. These reusable artifacts have been identified and used in the previous tasks in an isolate way resolving the tasks suggested and obtaining concrete results of the application analysis. Now we have to add all these defined artifacts to the use cases diagram of the application and define the relationships, according to the GridUCSec-profile, between functional use cases, generic Grid use cases, security use cases and misuse cases.

All use cases and relationships defined in the overall diagram have associated a set of properties and values that indicate the behaviour, function and semantic of each use case and relationship. These properties may be already predefined for reusable use cases in the repository, or have to be defined by the developer within the diagram identifying specific properties of each use case and the different relationships between them.

These properties are defined with the GridUCSec-profile model that has been elaborated for capturing the specific features and values of the use cases for Mobile Grid systems. As the overall diagram is complex to define here, we will describe a sub-diagram for showing how the GridUCSec-profile is used in the definition of the use cases and relationships involved. Fig. 7 shows a sub-diagram of the overall diagram which will be used for describing its elements according to the GridUCSec-profile model.

The "«GridSecurityUC» Authenticate" models the authentication service of the application and is responsible for protecting the "Login" UC and for mitigating the "«MisuseCase»

Unauthorized access" misuse case which threatens the "Login" UC. The "«GridSecurityUC» Authorize access" models the authorization service and is responsible for protecting the "«MobileUC» Search news" UC, for mitigating the "«MisuseCase» Unauthorized access" misuse case and for permitting the execution of "Login" and "«GridUC» Request". We also have the "«MisuseCase» Alteration info" misuse case that threatens the modification or alteration of the information exchanged in the messages every time that a request is sent to the system. This threat is mitigated by the "«GridSecurityUC» Ensure Confidentiality" and "«GridSecurityUC» Ensure Integrity" UCs which are part of the reusable sub-diagram stored in the repository. Finally, the "«MobileUC» Search News" UC is identified as a mobile UC due to the possible mobility of the user who requests information from the system from the mobile devices. This mobile UC includes the "«GridUC» Request" UC which is responsible for making the request in a secure manner.

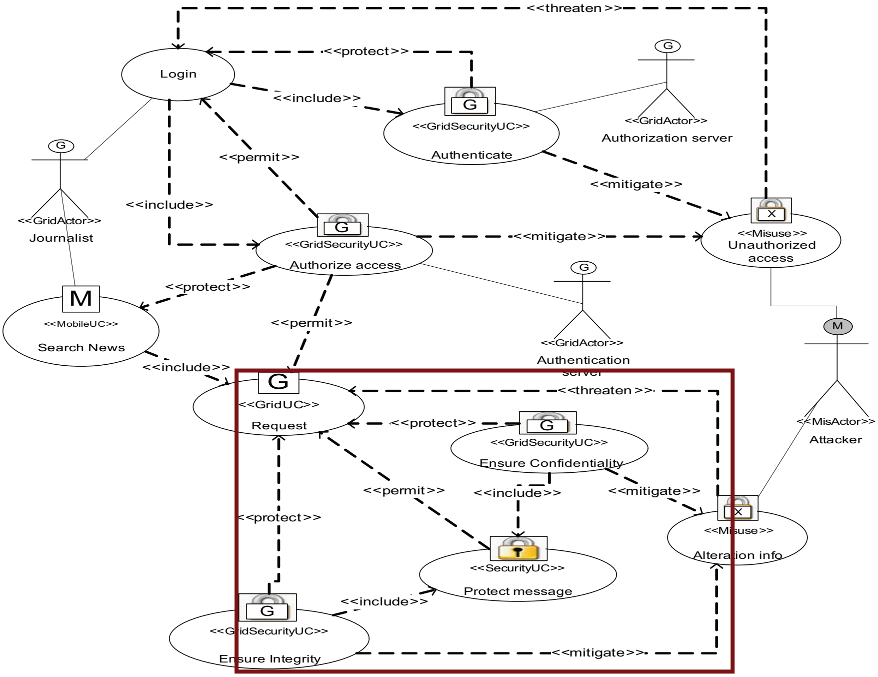


Fig. 7. Sub-diagram of the overall diagram of the application

In order to build the resulting diagram, we have used a reusable UCs diagram (framed sub-diagram shown in Fig. 7) which is available in the repository and is defined by using our UML profile, to model a common scenario that ensures confidentiality and integrity of a request in Grid environments, which our application requires. This sub-diagram shows how the "«GridUC» Request" UC is protected, through «protect» relationships, by the "«GridSecurityUC» Ensure Confidentiality" and "«GridSecurityUC» Ensure Integrity" security UCs which mitigate the "«MisuseCase» Alteration info" misuse case that threatens "«GridUC» Request". It also establishes a «permit» relationship from the "«SecurityUC» Protect message" security UC, meaning that once the message is protected, the request can be carried out.

Last, we should validate this diagram checking if the relationships between use cases are well defined and there is not redundancy or faults. This can be made in a first iteration or can go refining in successive iterations of this task when we add new use cases and relationships. This validation is automatic using the SMGridTool that is responsible for controlling the constraints of the use cases, the relationships between elements and that the possible values selected are within the range specified in the GridUCSec-profile.

The output artifact of this task is Use Cases View which contains all use cases diagrams generated in this task and the description of each use case, actor and relationship defined within the diagram.

Task A4. Supporting with UML models

UML models as interaction diagrams are used in this task for completing the capture of requirements and their dynamic behaviour. With these models we can define the actions between actors and use cases, and the flow of events produced between the elements of the diagram. With UML models we want to complete the definition of use cases knowing better the behaviour of the system and all the involved elements for refining the use cases model with new aspects identified thanks to these UML models.

In the repository there is a defined set of generic interaction diagrams which are related to reusable use cases. So we can instantiate these generic diagrams for the specific use cases of this application. In this first iteration, we have identified the Grid security use case called "«GridSecurityUC» Ensure Integrity" of the reusable repository, which has also a generic sequence diagram associated that we have to instantiate with the actors involved in this scenario. To finish this task, besides of interaction diagrams, other diagrams that completely define the behaviour of the use cases and facilitate the specification of requirements are defined. In this step, we will follow the analysis stage of the Unified Process, which defines classes diagrams, use case realizations, and initially describes packages and development diagrams which will be used and refined in the design activity.

The output artifacts of this task are: Static view whit diagrams generated applying the Unified Process, and Interaction view with the sequence and collaboration diagrams associated with the use cases and scenarios.

Task A5. Verifying Analysis model

Once the artifacts have been defined, we must verify that they have been correctly generated, that is, that the UML diagrams, such as for example the sequence diagram for message integrity, define the correct elements involved in the use case or in a scenario of use cases, in this case, the "«GridSecurityUC» Ensure Integrity" use case. This verification should also be made with the remaining diagrams and guides by the use cases defined in this activity.

Task A6. Specifying requirements

This is the last task of analysis activity and it specifies the set of requirements extracted from the use cases identified during all the previous tasks obtaining a description of functional and non-functional requirements of the system. This description should indicate all elements involved in the definition of requirements together with the interactions between use cases and actors and the attributes of these elements. All this information has been generated in previous tasks through use cases models using the GridUCSec-profile and with UML models. Here we define the requirements in natural language but we know that there are templates for the definition of requirements where the main characteristics of the models generated in this activity can be summarized and formally described in a document which will be part of the analysis model in future works. This task is carried out through review meetings by involved stakeholders in this task and the artifacts elaborated in previous tasks.

Step A6.1. Specify functional requirements. The functional requirements together with the related use cases can be clustered into the following categories: Network specific requirements; File and database management requirements; Query and retrieval requirements; User Interface requirements; Requirements for providing miscellaneous functionality.

Step A6.2. Specify security requirements. The security requirements are distinguished according to the different security services. They relate to safeguarding the accuracy and completeness of data and processing methods and the prevention of unauthorized entities to access and modify data. A first vision of security requirements that can be extracted from the functional use cases and security use cases of the application are: System requires authentication mechanisms for user identification; Users are classified into groups; System needs to give the capability to the administration users to define user group access permissions; Users can access to specific content based on their role; System uses data encryption; Single authentication process is required for accessing the functionality of the system. These security requirements extracted from use cases together with the security requirements that are defined in the security use cases can be specified into the following general categories: Data confidentiality; Trust and reputation; Authorisation and access control; Data integrity and authenticity; Authentication.

Step A6.3. Specify non-functional requirements. Based on the general requirements identified in (Fenkam et al., 2002; Jacob et al., 2005; Manish Parashar et al., 2005), this step presents the non-functional requirements, which should be considered during the development of the Grid system: Performance; Scalability/Expandability; Resource Management Capabilities; Semantic Grouping of Information; Availability; Security; Fairness; Robustness, Fault Detection and Recovery; Maintainability; Portability; Distribution; Usability; User interface; System stability; Deployment.

The output artifacts of this task are: "Requirements" artifact that defines all requirements (functional and non-functional) specified in this task, and "Analysis conflicts" artifact that identifies the possible conflicts or errors found in the specification of requirements and in the building of the use cases diagrams. This last artifact will be an input artifact for the subsequent iterations of this activity, but we have found none.

4.2 Application of the design activity

With these requirements, in the design activity, we can design the secure software architecture which covers all these requirements (functional, non-functional and those of security). The software architecture has been omittied for simplicity, and we focused on the reference security architecture specifically defined for the process.

Task D1. Designing Mobile Grid Software Architecture

This task defines the software architecture of the system using traditional methods and mechanisms of software engineering as the Unified Process, OPEN, OpenUP, etc. The software architecture is designed from functional requirements and use cases of the application that have been analyzed following the techniques and guides offered by some development processes but its input artifact is the analysis model elaborated in the previous activity of this process.

The output artifact is a software architecture oriented to Mobile Grid systems (that we have omitted here).

Task D2. Designing Mobile Grid Security Architecture

This task defines the security architecture where all security aspects of the application are taken into account. The aim of this task is to design a security architecture whose services and components cover the security requirements and needs for this case study and can be integrated with the software architecture designed in the previous task.

The input artifact is the analysis model with all use cases and requirements specified in the analysis activity.

Step D2.1. Find architecturally significant security UC. In this first iteration of the process, we are working with a reduced set of security use cases for simplicity, so that the same security use cases identified in the analysis activity will be selected for developing the different tasks of the design activity of the process. These security use cases selected are: Authenticate, Authorize access, Ensure Confidentiality, and Ensure Integrity. Each one of these security use cases are associated with some or many security services of the reference security architecture which will be identified in the next step.

Step D2.2. Relate Security UC – Security Services. Following the association rules defined for this process between security requirements and security services, we have that the security services to consider for the security architecture of this application from the security use cases identified in the previous activity are: Authorization, Authentication, Credential Management, Identity Management, Trust Management, Confidentiality and Integrity. We cannot forget the Grid Security Policy service and the Mobile Policy service which are needed to manage the different policies of the system. These security services are obtained of the relationship with the security requirements which are defined as tagged values in the GridUCSec-profile.

Step D2.3. Instantiate Reference Security Architecture. Each security use case is associated with one or more security requirements, and the security services associated with these security use cases cover the security requirements represented by these security use cases. Therefore, the aim is to obtain a set of security services which cover and take into account the security requirements represented by the four security use cases. In Fig. 8 we can see the set of security services necessary for fulfilling the security requirements specified in the first iteration with the four security use cases selected.

Step D2.4. Define security policies. Security policies must be defined for the services identified in the previous step and they must indicate the security rules allowed for each service along with protocols, mechanisms, attributes, etc., admitted by the service. The policies are stored in one or more LDAP repositories distributed throughout the Grid system but always available online.

We must have security policies associated with each Grid security service of the instantiated architecture (for example, the security policy of the authorization service); we must also define security policies for messages exchanged between services, or inbound to the system (for example, each message must be protected against attacks on the integrity and confidentiality of the message body); and we must define security policies for inter-, intra-organizational partnerships (the sending of messages should be protected from attacks on privacy, integrity, confidentiality, and the sender entity must be trusted).

Grid Security Policies govern and control the overall security of the Grid system and are associated with the security services, with inter-, intra- organizational partnerships, with

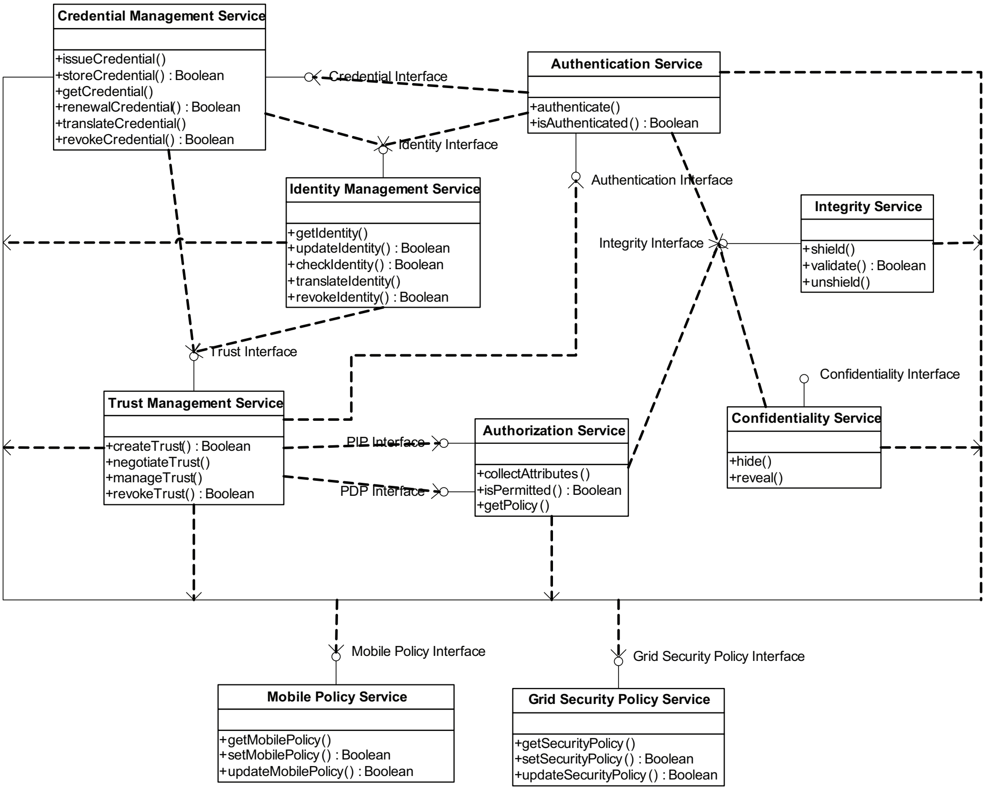


Fig. 8. Services and interfaces of Security Architecture instantiated for this case study

communication and exchange of information between resource users and Grid entities. Mobile Policies govern the use of mobile devices and wireless communications by establishing rules, standards, norms and advices of how resources must act on the mobile environment, protect their data, communicate via protocols, sending data, etc.

The output artifact is a part of the security architecture designed from the set of use cases selected and with the help of the reference security architecture.

Task D3. Designing/Integrating Secure Mobile Grid Software Architecture

This task cannot be applied because for this case study we have omitted the part of the software of the architecture and we cannot see how the integration is carried out. Basically we have to define how the various elements of the system, software architecture and security architecture, potentially constructed at different times, should be brought together into a single architecture, a secure software architecture.

Elements of the security architecture should be integrated into the software architecture because many software elements have to make use of the security elements, and the security elements have to act over the software elements to provide security in the system. Therefore, we must define the relations between these elements through the interfaces of the services, defining the protocols and mechanisms of exchange, policies, permissions and constraints.

Moreover, the static, dynamic and deployment views of all these elements of the secure software architecture are very useful to understand and know better the architecture and the interactions between all its elements when the decisions of implementation are made in the implementation activity. These views are defined using traditional methods with the secure software architecture as input artifact.

Task D4. Specifying Secure Mobile Grid Software Architecture

Systems Architect and Security Architect created an initial version of the document of secure software architecture in which all the information outlined in the previous points is widespread. This document follows the basic points of the document specifying the functional architecture consistent with IEEE 1471-2000 adding new players, views, views, and packages of views of security.

Task D5. Validating/Verifying Secure Mobile Grid Software Architecture

This task validates and verifies the designed architecture. We can only validate and verify the security architecture that is that we are developing in this case study. The analysis model with the specified use cases and requirements and the architecture designed in this task are the input artifacts to validate and verify the architecture.

Step D5.1. Validate designed architecture. Using as input the document of specification of secure software architecture (although in this case study we will only have the security architecture, the software architecture has not been shown), the process of validating the security architecture was done through review meetings by the involved roles in this task and it was found that the security architecture response to the needs of involved stakeholders and, moreover, it is integrated without any impact into the designed software architecture.

Step D5.2. Verify design model. It was verified that the traceability between requirements and services designed through the rules of association was successfully performed. Therefore, artifacts of the activity analysis are needed as input to the design activity and are the basis for developing the artifacts of the design model.

Step D5.3. Identify, study and analyze conflicts. In addition, it was observed that this architecture was not in conflict with any other non-functional requirements, and that all elements designed in this iteration were well integrated and there was no irregularity or inconsistency.

4.3 Application of the construction activity

Finally, in the construction activity, a possible implementation of the elements previously designed is carried out. In this first iteration, we can implement a set of elements of the designed architecture that are independent of the rest of elements or with little relationships for being implemented in an isolated way.

Task C1. Selecting Technological Elements

This task defines the technological environment that we must use in order to implement the system with the help of tools and libraries oriented to Grid computing.

Step C1.1. Study Grid middleware, tools and libraries. Within the Gredia project, a series of tools and platforms to develop the different proposals that are being carried out with reference to Grid systems and mobile devices are being used. These same tools will be used here for their perfect integration into GREDIA and because they are the most used and appropriate for these systems. This tool is Globus toolkit which provides software tools that make it easier to build computational grids and grid-based applications. Also PERMIS and VOMS can be used as security infrastructure for authorization and access control, and Java CoG kits that is a tool which provides the implementation in Java of some components of Globus toolkit.

Step C1.2. Install and Configure Grid environment. With this set of tools and libraries identified we can begin the installation and configuration in a way that all components are available under the same platform. Globus defines a wide set of manuals and documentation[1] for the installation and configuration of GT4, such as the quickstart guide and the administrador's guide where the steps to carry out for that GT4 is implemented and prepared to its use are indicated. The Java CoG Kit provides convenient access to Grid middleware through the Java framework. The version 4.1.2 of the Java CoG Kit has been released and guides, manuals and documentation are available to be downloaded[2]. Finally, PERMIS[3] will be used as an authorization infrastructure integrated into the Globus toolkit[4] that helps us implement the authorization service designed in the previous activity. The Java libraries, PERMIS package, Globus toolkit, XML files, LDAP repository, policy files, etc. are the output artifacts of this task.

Task C2. Implementing Secure Mobile Grid System

For this first iteration, we will only consider the authorization service because we believe it is the most comprehensive one that takes into account many aspects of the technologies chosen and that requires more effort to install and configure. We have as input artefact, just one element of the design model (the authorization service, we have omitted the rest) and all grid components identified and installed previously. The Authorization service has two interfaces, PIP interface and PDP interface. The first has one operation that is collectAttributes() and the latter has two operation that is isPermitted() and getPolicy(). Globus provides the ideal framework for the authorization and permits us to create our own authorization framework. We have to implement the PDP interface and PIP interface of our authorization service. Globus shows how we can change and implement our own PDPs and PIPs into the Globus platform.

The authorization service of the reference security architecture designed in the design activity defines one operation for PIP interface and two operations for PDP interface. All these operations define a set of parameters that Globus uses and implements. Therefore, if we define the interfaces of Globus for the security services of the architecture, we can easily implement a robust authorization service. This activity belongs more to the programming field than to the research field, which is out of the scope of this paper to get and to implement Java code for security services.

As output artifacts, we obtain a part of the secure Mobile Grid system artifact (the part corresponding to the implementation of the authorization service), and the implementation structure of the authorization service.

Task C3. Validating and Verifying Secure Mobile Grid System

This task validates and verifies that the system which is being built fits to the requirements and follows a right traceability of artifacts during the development. We need as input

[1] www.globus.org/toolkit/docs/4.0/

[2] www.cogkit.org/release/4_1_2/

[3] http://sec.cs.kent.ac.uk/permis/

[4] http://sec.cs.kent.ac.uk/permis/integrationProjects/GT.shtml

artifacts the security requirements (because the rest of requirements are not dealt with here) specified for the elements selected in the design, the part of the design model developed, the structure of how the authorization service has been implemented and the own authorization service (classes and packages).

Step C3.1. Validate implemented system. In this first iteration we can only validate the components that have been implemented because the end system has not been constructed yet. We can validate that the security services of authentication, authorization, confidentiality and integrity which have been designed in the previous activity are implemented with security mechanisms that carry out the protection of the system complying with a set of requirements specified in the analysis activity. In this case the implemented services cover authentication, mutual authentication, confidentiality, integrity, trust, interoperability, flexibility and single sig-on. These implemented services help us fulfill other requirements but must be supported with more security services that will be implemented in the next iterations. When the system is totally built, apart from validating elements which have not been validated yet, it validates the whole set of elements, i.e. how the elements relate and interact to provide more security and cover more requirements.

Step C3.2. Verify construction model. We verify that the implemented services, interfaces and any other element have followed the methods and techniques indicated in the process ensuring the traceability between artifacts thus justifying that all artifacts previously defined in the process are necessary for achieving the aim of implementation.

Step C3.3. Identify, study and analyze conflicts. In a first iteration in which a few elements are selected for their development incompatibilities or errors do not generally appear. It is in the latest iteration in which all elements are implemented and integrated when we can identify a greater number of them. We declare that all elements implemented in this iteration were well built and there was no irregularity or inconsistency with the rest of elements.

Task C4. Testing the Secure Mobile Grid System

The tests of the system are carried out once the system has been completely built. Nevertheless in this first iteration we have implemented the authorization service into Globus that can be tested. As this case study is focused on the part of security, we only have security elements implemented, such as the authorization service; therefore the test must be carried out over this element.

These tests consist in adding users, roles and policies to the system and perform calls to the Grid services requesting the access to some data or execution of a simple job (information appeared in the security test plan and test plan of the planning activity of the process). In these tests, we must study some cases where the authorization service permits access (returning PERMIT) and other cases in which access is denied (returning DENY) to check their correct implementation and logical functioning. For the authorization service we can confirm that the tests obtained the expected result.

The output artifacts indicate the tests executed adding to the reports the results obtained from a set of input data, and their valoration.

Task C5. Preparing Manuals and Documentation

This task defines the manuals and documentation of development, configuration, installation and user but once the system has been completely developed, therefore until the last iteration it is not recommendable to elaborate them due to the possible changes and refinements of the system.

5. Conclusion

This paper has proposed a development process for secure Mobile Grid systems (SecMobGrid), which enables to develop in a systematic and secure way a Grid system with mobile devices incorporating security from the earliest stages of development and based on the reuse of artifacts which have been generated in the different applications of the process.

This is an iterative and incremental process and exploits the reusability of the different artifacts that are generated during the process. This process is composed of several activities and among them, we can highlight the following: the analysis activity where all system requirements (especially security requirements) are captured and specified basing on specific use cases, which are defined using a UML profile (GridUCSec-profile); the design activity focused on the construction of a security architecture for this kind of systems from the reference security architecture developed that defines the minimum set of security services covering all possible security requirements that can be specified in mobile Grid systems; and the construction activity that is in charge of the implementation of the system using the existing tools for the development of Grid systems.

Moreover, the prototype tool called "SMGridTool" that gives automated support for the development of the tasks of the analysis activity of the SecMobGrid process is presented. Finally, in order to validate and improve the SecMobGrid development process, the result of its application to a case study of a real Mobile Grid system is offered.

As future work, we will concrete and refine the generic tasks of the used development processes that have been incorporated into the SecMobGrid process. We will refine and improve the parameters and tagged values of the GridUCSec-profile for capturing the most important aspects and features of Mobile Grid systems and we will improve the reference security architecture for that the security aspects considered in the analysis activity through the GridUCSec-profile can easily be incorporated as parameters into the interfaces of the security architecture, into the definition of policies of the system or into the decisions of implementation. Finally, we will carry out new case studies for a continuous improvement of the SecMobGrid process in other Grid environments and dominions.

6. Acknowledgments

This research is part of the following projects: QUASIMODO (PAC08-0157-0668), SISTEMAS (PII2I09-0150-3135) and SEGMENT (HITO-09-138) financed by the "Viceconsejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha" (Spain) and FEDER, and MEDUSAS (IDI-20090557), BUSINESS (PET2008-0136) and PEGASO/MAGO (TIN2009-13718-C02-01) financed by the "Ministerio de Ciencia e Innovación (CDTI)" (Spain). Special acknowledgment to GREDIA (FP6-IST-034363) funded by European Commission.

7. References

Artelsmair, C. and R. Wagner, 2003. Towards a Security Engineering Process. *The 7th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, Florida, USA.

Bass, L., F. Bachmann, R. J. Ellison, A. P. Moore and M. Klein, 2004. "Security and survivability reasoning frameworks and architectural design tactics." *SEI*.

Bhanwar, S. and S. Bawa, 2008. Securing a Grid. *World Academy of Science, Engineering and Technology*.

Breu, R., K. Burger, M. Hafner, J. Jürjens, G. Popp, V. Lotz and G.Wimmel, 2003. Key issues of a formally based process model for security engineering. *International Conference on Software and Systems Engineering and their Applications*.

Dail, H., O. Sievert, F. Berman, H. Casanova, A. YarKhan, S. Vadhiyar, J. Dongarra, C. Liu, L. Yang, D. Angulo and I. Foster, 2004. Scheduling In The Grid Application Development Software Project. *Grid resource management:state of the art and future trends*: 73-98.

EGEE Middleware Design Team. (2004). "EGEE Middleware Architecture." from https://edms.cern.ch/document/476451/.

Fenkam, P., S. Dustdar, E. Kirda, G. Reif and H. Gall, 2002. Towards an Access Control System for Mobile Peer-to-Peer Collaborative Environments. *Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'02)*.

Flechais, I., M. A. Sasse and S. M. V. Hailes, 2003. Bringing Security Home: A process for developing secure and usable systems Workshop on New Security Paradigms. Ascona, Switzerland, ACM Press: 49--57.

Foster, I. and C. Kesselman, 2004. *The Grid2: Blueprint for a Future Computing Infrastructure*. San Francisco, CA, Morgan Kaufmann Publishers; 2 edition.

Foster, I., C. Kesselman, J. M. Nick and S. Tuecke, 2002. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum.

Gartner. (2007). "Gartner Says Worldwide PDA Shipments Top 17.7 Million in 2006." Gartner Press Release, from http://www.gartner.com/it/page.jsp?id=500898.

Gartner. (2009). "Gartner Says Worldwide Mobile Phone Sales Declined 8.6 Per Cent and Smartphones Grew 12.7 Per Cent in First Quarter of 2009." Gartner Press Release, from http://www.gartner.com/it/page.jsp?id=985912.

Giorgini, P., H. Mouratidis and N. Zannone, 2007. Modelling Security and Trust with Secure Tropos. *Integrating Security and Software Engineering: Advances and Future Visions*. H. M. a. P. Giorgini, Idea Group Publishing: 160-189.

Graham, D. (2006). "Introduction to the CLASP Process." from https://buildsecurityin.us-cert.gov/daisy/bsi/articles/best-practices/requirements/548.html.

Guan, T., E. Zaluska and D. D. Roure, 2005. A Grid Service Infrastructure for Mobile Devices. *First International Conference on Semantics, Knowledge, an Grid (SKG 2005)*, Beijing, China.

Haley, C. B., J. D. Moffet, R. Laney and B. Nuseibeh, 2006. A framework for security requirements engineering. *Software Engineering for Secure Systems Workshop*, Shangai, China.

Humphrey, M., M. R. Thompson and K. R. Jackson, 2005. "Security for Grids." *Lawrence Berkeley National Laboratory. Paper LBNL-54853*.

IEEE, 2000. Recommended Practice for Architectural Description of Software-Intensive Systems (IEEE Std 1471-2000). . New York, NY, Institute of Electrical and Electronics Engineers: 29.

Jacob, B., M. Brown, K. Fukui and N. Trivedi, 2005. *Introduction to Grid computing*, IBM Redbooks.

Jacobson, I., G. Booch and J. Rumbaugh, 1999. *The Unified Software Development Process*, Addison-Wesley Professional.

Jameel, H., U. Kalim, A. Sajjad, S. Lee and T. Jeon, 2005. Mobile-To-Grid Middleware: Bridging the gap between mobile and Grid environments. *European Grid Conference EGC 2005*, Amsterdam, The Netherlands, Springer.

Jana, D., A. Chaudhuri and N. B. Bhaumik, 2009. "Privacy and Anonymity Protection in Computational Grid Services." *International Journal of Computer Science and Applications* 6(1): 98-107.

Jurjens, J., 2001. Towards Development of Secure Systems Using UMLsec. *Fundamental Approaches to Software Engineering (FASE/ETAPS)*.

Jurjens, J., 2002. UMLsec: Extending UML for Secure Systems Development. *5th International Conference on the Unified Modeling Language (UML)*, Dresden, Germany.

Jürjens, J., 2005. *Secure Systems Development with UML*, Springer.

Kalim, U., H. Jameel, A. Sajjad and S. Lee, 2005. Mobile-to-Grid Middleware: An Approach for Breaching the Divide Between Mobile and Grid. *4th International Conference on Networking*, Reunion Island, France, Springer.

Kolonay, R. and M. Sobolewski, 2004. Grid Interactive Service-oriented Programming Environment. *Concurrent Engineering: The Worldwide Engineering Grid*, Tsinghua, China, Press and Springer Verlag.

Kruchten, P., 2000. *The Rational Unified Process: An Introduction*, Addison-Wesley.

Kumar, A. and S. R. Qureshi, 2008. Integration of Mobile Computing with Grid Computing: A Middleware Architecture. *2nd National Conference on Challenges & Opportunities in Information Technology (COIT-2008)*, Mandi Gobindgarh, India.

Kwok-Yan, L., Z. Xi-Bin, C. Siu-Leung, M. Gu and S. Jia-Guang, 2004. "Enhancing Grid Security Infrastructure to Support Mobile Computing Nodes." *Lecture Notes in Computer Science* 2908/2003: 42-54.

Lodderstedt, T., D. Basin and J. Doser, 2002. SecureUML: A UML-Based Modeling Language for Model-Driven Security. *5th International Conference on the Unified Modeling Language (UML), 2002*, Dresden, Germany, Springer.

Manish Parashar, Zhen Li, Hua Liu, Vincent Matossian and C. Schmidt, 2005. Enabling Autonomic Grid Applications: Requirements, Models and Infrastructure. *Self-star Properties in Complex Information Systems*, Springer. 3460: 273-290.

Mouratidis, H. and P. Giorgini, 2006. *Integrating Security and Software Engineering: Advances and Future Vision*, Idea Group Publishing.

Mouratidis, H., P. Giorgini and G. Manson, 2005. "When security meets software engineering: A case of modelling secure information systems." *Information Systems* 30(8): 609-629.

Open Group. (2009). "TOGAF™ Version 9 -- The Open Group Architecture Framework." from http://www.opengroup.org/architecture/togaf9-doc/arch/.

Phan, T., L. Huang, N. Ruiz and R. Bagrodia, 2005. Chapter 5: Integrating Mobile Wireless Devices Into the Computational Grid. *Mobile Computing Handbook*. M. Ilyas and I. Mahgoub, Auerbach Publications.

Riaz, M., S. L. Kiani, A. Shehzad and S. Lee, 2005. Bringing Handhelds to the Grid Resourcefully: A Surrogate Middleware Approach. *Computational Science and Its Applications – ICCSA 2005*, Singapore, Lecture Notes.

Romberg, M., 2002. "The UNICORE Grid Infrastructure." *Scientific Programming* 10(2): 149-157.

Rosado, D. G., E. Fernández-Medina and J. López, 2009a. Applying a UML Extension to build Use Cases diagrams in a secure mobile Grid application. *5th International Workshop on Foundations and Practices of UML, FP-UML 2009*, Gramado, Brasil, LNCS 5833.

Rosado, D. G., E. Fernández-Medina and J. López, 2009b. Reusable Security Use Cases for Mobile Grid environments. *Workshop on Software Engineering for Secure Systems, in conjunction with the 31st International Conference on Software Engineering*, Vancouver, Canada.

Rosado, D. G., E. Fernández-Medina and J. López, 2010a. "Security Services Architecture for Secure Mobile Grid Systems." *Journal of Systems Architecture. Special Issue on Security and Dependability Assurance of Software Architectures* (article in press).

Rosado, D. G., E. Fernández-Medina, J. López and M. Piattini, 2010b. "Analysis of Secure Mobile Grid Systems: A Systematic Approach." *Information and Software Technology* 52: 517-536.

Rosado, D. G., E. Fernández-Medina, J. López and M. Piattini, 2010c. "Developing a secure mobile Grid system through a UML extension." *Journal of Universal Computer Science* (to be published in 2010).

Rosado, D. G., E. Fernández-Medina, J. López and M. Piattini, 2010d. "Systematic Design of Secure Mobile Grid Systems." *Journal of Network and Computer Applications* (under review).

Sajjad, A., H. Jameel, U. Kalim, S. M. Han, Y.-K. Lee and S. Lee, 2005. AutoMAGI - an Autonomic middleware for enabling Mobile Access to Grid Infrastructure. *Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services - (icas-icns'05).*

Steel, C., R. Nagappan and R. Lai, 2005. Chapter 8.The Alchemy of Security Design Methodology, Patterns, and Reality Checks. *Core Security Patterns:Best Practices and Strategies for J2EE™, Web Services, and Identity Management*, Prentice Hall PTR/Sun Micros: 1088.

The AKOGRIMO project. from http://www.akogrimo.org/.