Abstract:
This document describes the security framework of the SORMA Open Grid Market. It starts from a review about the SORMA architecture and security requirements which motivate the security concepts and design. As most of the common security mechanisms for encryption and authentication are well understood, this document emphasizes on distributed identity and trust management concepts, the challenges and how they will be addressed in SORMA. In addition, the document also describes the security requirements and issues of the individual SORMA components that will be addressed in the next version of the prototype.

Keyword list (optional): security, identity management, single sign-on
Table of Contents

1 About This Document .......................................................... 1
   1.1 Purpose of Document ....................................................... 1
   1.2 Change History .............................................................. 1
   1.4 Contact Persons ............................................................. 2
2 Introduction and Overview ....................................................... 2
   2.1 SORMA Architecture Review ............................................ 3
      2.2.1 Layer 4: Grid Application .......................................... 4
      2.2.2 Layer 3: Intelligent Tools .......................................... 4
      2.2.3 Layer 2: Open Grid Market ........................................ 6
      2.2.4 Layer 1: Core Market Services .................................... 7
   2.2 Security Requirements .................................................... 8
      2.2.1 Certificate and GSI Issues ......................................... 8
      2.2.2 Single Sign-On ........................................................ 10
      2.2.3 Distributed Identity Management ................................ 10
      2.2.4 Trust Management ................................................... 11
      2.2.5 Open Standards and Interoperability ............................ 12
3 SORMA Security Framework .................................................. 12
   3.1 Basic Security in SORMA ................................................ 12
      3.1.1 User Authentication .................................................. 12
      3.1.2 Securing Communication ......................................... 13
   3.2 Distributed Identity Management ..................................... 14
      3.2.1 Identity Management Standards and Tools .................... 14
         3.2.1.1 Liberty ............................................................ 14
         3.2.1.2 WS-Federation ............................................... 15
         3.2.1.3 OpenID ......................................................... 16
         3.2.1.4 Security Assertion Markup Language (SAML) ........... 17
         3.2.1.5 Shibboleth ..................................................... 18
         3.2.1.6 Sun Access Manager / OpenSSO ............................ 18
      3.2.2 Identity Management in SORMA ................................ 21
         3.2.2.1 SORMA Identity Provider ................................... 21
         3.2.2.2 Web Portal Login ............................................. 21
         3.2.2.3 Security and Identity Service ............................... 27
      3.2.3 Credential Exchange Service .................................... 27
   3.3 Trust as Soft Security ..................................................... 29
      3.3.1 Definition of Trust .................................................. 29
      3.3.2 Existing Trust Mechanisms ....................................... 30
      3.3.3 Reputation system in SORMA .................................... 32
4 Security in SORMA Components .............................................. 33
   4.1 Core Services .............................................................. 33
   4.2 Trading Management Component ..................................... 35
4.2.1 Introduction to C-Space.................................................................36
4.2.2 Security Related Requirements..................................................36
4.3 Payment Component....................................................................38
  4.3.1 Input information needed by Payment Component...................39
4.4 Economically Enhanced Resource Manager.................................40
4.5 Contract/SLA Management Enforcement and Billing Component...41
Tables...............................................................................................43
Figures..............................................................................................43
References.........................................................................................44
SORMA Consortium........................................................................47
1 About This Document

1.1 Purpose of Document

This document describes the security framework of the SORMA Open Grid Market. In particular, this document comprises:

- Review of the SORMA architecture and security requirements (section 2): This section presents the logical view of the architecture and the components. Next, we discuss the security requirements that is described in the Description of Work, as well as some of the issues and trends in security research.

- A detail description of the security framework (section 3): In this section, we describe the SORMA security concepts, focusing on distributed identity and trust management. We also introduce some of the current work and technology that are available today.

- Description of security requirements in the SORMA components (section 4): In the next phase of the prototype development, we will integrate the security concepts into the components. This section describes what are the requirements and issues that need to be addressed by each of the components.

1.2 Change History

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Comments</th>
<th>Version</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 2008</td>
<td>Melvin Koh</td>
<td>Structure and content</td>
<td>0.1</td>
<td>Initial Draft</td>
</tr>
<tr>
<td>22 February 2008</td>
<td>Arun Anandasivam</td>
<td>Added section 3.3</td>
<td>0.2</td>
<td>Draft</td>
</tr>
<tr>
<td>25 February 2008</td>
<td>Matteo Villa, Enrico Del Grosso</td>
<td>Added section 4.3</td>
<td>0.3</td>
<td>Draft</td>
</tr>
<tr>
<td>26 February 2008</td>
<td>Mario Macías</td>
<td>Added section 4.4</td>
<td>0.4</td>
<td>Draft</td>
</tr>
<tr>
<td>29 February 2008</td>
<td>Niklas Wiström, Lars Rasmusson</td>
<td>Added section 4.2</td>
<td>0.5</td>
<td>Draft</td>
</tr>
<tr>
<td>10 March 2008</td>
<td>Pablo Chacin</td>
<td>Added section 4.1</td>
<td>0.6</td>
<td>Draft</td>
</tr>
<tr>
<td>14 March 2008</td>
<td>Vikas Deora</td>
<td>Added section 4.5</td>
<td>0.7</td>
<td>Draft</td>
</tr>
<tr>
<td>14 March 2008</td>
<td>Melvin Koh</td>
<td>Proof reading and minor re-organization</td>
<td>1.0</td>
<td>Final</td>
</tr>
</tbody>
</table>
1.4 Contact Persons

Table 2 lists the institutions which are responsible for the Intelligent Tool and Open Grid Market development. The task leader is one person from those institutions.

<table>
<thead>
<tr>
<th>Task</th>
<th>Name</th>
<th>Institutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>T4.1 Specification of the Security and Trust Concept of</td>
<td>Simon See, Melvin Koh</td>
<td>SUN</td>
</tr>
<tr>
<td>the Open Grid Market</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4.2 Implementation of the Bid Generator</td>
<td>Jens Nimis</td>
<td>FZI</td>
</tr>
<tr>
<td>T4.3 Evaluation of the available Open Grid Market</td>
<td>Torsten Eymann</td>
<td>BFM</td>
</tr>
<tr>
<td>components for usage in SORMA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4.4 Implementation of missing Open Grid Market</td>
<td>Lars Rasmusson</td>
<td>SICS</td>
</tr>
<tr>
<td>components and amendment of given components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4.5 Testing of the Open Grid Market software components and refinement</td>
<td>Mark Baker</td>
<td>URE</td>
</tr>
</tbody>
</table>

Table 2: Contact Persons

2 Introduction and Overview

The overall objective of SORMA project is the development of methods and tools for establishing an efficient market-based allocation for resources in a more efficient way in order to enable resource accessibility for all users and to increase user’s satisfaction, profit and productivity. The underlying interdisciplinary methodology will ensure that the resource allocation process is autonomously conducted – albeit tailored to the requirements of the users – which leads to the realization of a self-organizing resource management system.

The mediated resource allocation and delivery over the market will allow better utilization of available resources, which automatically directs those resource provided to the clients who, value them most. Hence, the SORMA lays a foundation for an efficient resource management, which keeps the amount of waste at a very low level.

More specifically, SORMA aims to create a framework for realizing self-organizing resource
management that will enable the establishment of economically sound and a technically feasible market over which resources, i.e. computational resources, as well as hard- and software can be traded to allow an efficient allocation of given resources. The framework will be solidly based on the analysis of users’ needs and the state of the art brought by both researchers and practitioners.

2.1 SORMA Architecture Review

In this section, for completeness, the SORMA architecture's logical view is presented. The SORMA architecture is described in detail in D2.1 and D2.2. Figure 1, taken from D2.1, describes what the SORMA system does in terms of its functional components, their responsibilities and their dependencies. In this view boxes represent functions (not necessarily components) and arrows represent their dependencies (not necessarily data or control flows).

![Figure 1: SORMA Architecture Logical View](image)
D1.1 described a five layered architecture; in this document the architecture is refined four layers in order to reflect that from a market point of view, the consumers with their applications and the providers with their resources, belong to the same layer. Furthermore, the former Economically Enhanced Virtualization Middleware is not a layer on its own but rather a component (the Economically Enhanced Resource Management) within the Open Grid Market layer. The remainder of this section describes the layers and associated components of the updated architecture. The intention is to provide a high-level overview of the system design here and then later zoom down into greater detail as each of the components are described in subsequent sections.

2.2.1 Layer 4: Grid Application
Layer 4 is the home of the Grid applications and Grid resources to be traded on the SORMA market and of the involved humans.

At the provider side a provider IT specialist makes use of the intelligent tools in layer 3 to model the provider's business strategies and the offered Grid resources. “Grid resource” in this context means a physical resource, a raw service and/or a complex service. SORMA will initially focus on the trading of “physical resources”, but from a logical architecture standpoint, it is correct to consider that any type of resource or service could be offered at this layer.

On the consumer side it has to be distinguished between the Grid application's end user(s) and the consumer's IT support staff who will use the intelligent tools to model an application's resource requirements and the consumer's preferences.

2.2.2 Layer 3: Intelligent Tools
The users (consumers and providers) are supported by intelligent tools for an easy access to the SORMA market.

**Consumer preference modelling:** This component allows for the consumers to describe their economic preferences that will determine their bidding strategies on the Open Grid Market, e.g. they could define if they prefer cheap over reliable resources. One approach would be to provide the users with a GUI in the form of a simplified ontology modelling tool to instantiate a given consumer
Demand modelling: The user needs a tool to specify the technical requirements of her Grid application. It will be part of the discussions if these requirements will be specified in terms of plain resources or aggregate services and how this process must be supported by the Grid applications themselves. If the requirements are specified as aggregate services and the offered resources are of finer granularity this component could also be used for the request decomposition. The technical approach for this component could be similar (i.e. ontology-based) to consumer the preference modelling and the according GUIs could be integrated.

Business modelling: Analogously to the consumer preference modelling, the providers have to specify their business models to determine the generation of their offers on the Open Grid Market. For example one part of such a description could be a pricing model that specifies if the consumer has to pay for booked time-slots or for the actual usage. As the example indicates, the models specified by means of this component depend on the implemented market type. (Technical approach analogue to consumer preference modelling.)

Supply modelling: The resource (or service...) modelling component is the counterpart of the demand modelling component. Aided by this component the providers can technically specify their offers. (Technical approach analogue to request description.)

Bid generation: The bid generation is the intelligent (agent-like) component that generates and places the bids of the consumer on the Open Grid Market. For this purpose it considers the user preferences, the (decomposed) technical requirements and the current state of the market and derives the bids. The bids (which should be supervised by the consumer) are submitted to the trading Management component of the Open Grid Market. The bid generation component could be implemented with the help of a rule engine for logical inference over the mentioned inputs.

Offer generation: The offers are assembled from the technical resource descriptions and the business model of the respective provider by the offer generation component. It also publishes the offers at the Grid market directory. It will be part of the discussion, how intelligent this component has to be as this depends on the market type. An intelligent component again could be rule engine-powered.
2.2.3 Layer 2: Open Grid Market

The Open Grid Market in layer 2 is the place, where the offered resources/services are assigned to the Grid applications of the consumers, following certain market organizations.

**Trading management:** The trading management component is the access point for the consumers to the Open Grid Market where they can find the offered services and place their according bids. Therefore as a first step, the trading management matches the technical descriptions of the request (received from the consumers' bid generation) to the suitable technical descriptions of the offered resources (collected from the associated Grid market directories). In the second phase the trading management orchestrates the bidding process from the (possibly competing) consumers according to a given market organization (e.g. English auction). If the bidding process finishes successfully the corresponding bid and offer are submitted to the contract management.

**Contract management:** The contract management component transforms corresponding pairs of bids and offers to mutually agreed contracts. One important part of these contracts are the service level agreements (SLAs) which define the agreed terms of usage of the resources and the pricing. The contract management also initiates the enforcement of the contract, especially the allocation of the sold resources (aided by the extended resource management) and the payment process.

**SLA enforcement and billing:** The SLA enforcement and billing component is responsible for the surveillance and enforcement of the contracts it receives from the contract management. It keeps track of the actual usage of the resources, makes comparisons to the SLA and (if appropriate) initiates the billing and clearing according to the results of the comparison. Thereto, it can make use of the payment interface component.

**Payment:** The payment service offers a unified interface to payment, isolating the rest of the components from the particularities of the payment mechanism. The payment service also generates appropriate logging/auditing information.

**Security management:** The security management component is intended as the entry point for a single sign-on mechanism and is responsible for a tamper-proof identity management for the consumers, the suppliers and the constituent components of the SORMA system. Thus, probably all layers that are developed as part of the SORMA system will have to provide security connectors that build the
technological bridges from the respective layers to the security management of the Open Grid Market layer. (For sake of clarity the security connectors are not depicted in the logical view diagram.)

**Economically Enhanced Resource Management (EERM):** This component provides a standardized interface to typical Grid middleware (e.g. Globus Toolkit or Sun Grid Engine). The EERM can shield clients from resource platform specific issues and also enhance or complement the management functions provided by job scheduling and submission systems. The EERM’s main duties include:

- **Resource management:** Management functions to achieve the expected resource service levels and notify of variations. Coordinates independent resources to allow co-allocation (if not provided by the fabrics).

- **Resource monitoring:** The resource monitoring subcomponent monitors the state of the resources in terms of technical parameters.

- **Resource fabrics:** Standardized interfaces to create instances of resources and later make use of them from the application.

### 2.2.4 Layer 1: Core Market Services

Standard Grid middleware does not provide all the infrastructure services necessary for an open marketplace. Layer 1 extends the standard Grid middleware by additional infrastructure services:

**Trusted market exchange service:** All communication among participants (sellers, buyers, supporting services of the open market layer) is mediated by this service, which assures that information is routed to the appropriate party in a secure and reliable way. Routing is done by applying rules on content and also from the context of the communication (for instance, existing negotiations). Also, this service enforces policies defined on specific messages for logging, encryption, signing, etc.

**Logging:** All relevant transactions executed on the market will be registered in a secure log for auditing purposes (for instance, for non-repudiation of agreements).

**Market directory:** The market directory is a market-enabled extension to the commonplace service registries in (Grid) middleware. The technical service offers in the Grid market directories are enriched by economically relevant information like pricing or quality of service parameters. The enriched information serves as input to the trading management component.
Market information: The market information service allows participants to publish information and to gather information from other participants (for instance, prices, resource usage level). Participants can query the service or subscribe to topics. Information queries can be about instantaneous values or the history of the value.

2.2 Security Requirements

The SORMA Open Grid Market (hereby referred to only as SORMA), in its final form, is the establishment of economically sound and a technically feasible market over which resources, i.e. computational resources, as well as hard- and software can be traded to allow an efficient allocation of given resources. Therefore, resource owners have an incentive to supply the most demanded resources because they are directly rewarded in monetary terms. So, SORMA can enfold its potential best when real money is involved, and when real money is involved, security becomes a major factor. In computer systems, the 3As – Authentication, Authorisation and Accounting (AAA) are the three main requirements in computer security.

- Authentication – Establishing or confirming users identity is authentic.
- Authorization – Allowing access to resources only to those permitted to use them.
- Accounting – process of collecting and sending security server information used for billing, auditing, and reporting

Besides “AAA”, there are other requirements that are defined in SORMA – single sign-on, open standards and interoperability, distributed identity management and trust management. However, first and foremost, we need to understand the issues and problems with the Grid Security Infrastructure (GSI), which is regarded as the defacto standard for Grid computing today.

2.2.1 Certificate and GSI Issues

Most of the established Grid infrastructures and Grid middlewares today follows the authentications mechanisms defined by the Globus Toolkit. The Grid Security Infrastructure (GSI) functionality is based on the Public Key Infrastructure (PKI). Every user and service on the Grid is identified via a digital certificate encoded in X.509 format, which contains four primary pieces of information:
Some researchers have argued that using the PKI approach is inflexible (Vullings, Dalziel et al. 2007) and has several inherent limitations (Gutmann, 2001):

1. Key lookup: When a user apply for the Grid access, the certificate are mostly mailed around.
2. Enrolment: How does a user obtain a trusted certificate in the first place? Some countries do not have a national CA, so it is difficult to obtain an appropriate certificate.
3. Validity checking: Currently, checking of the validity of a certificate is often not done, as it is very time-consuming, and the information is often outdated.
4. User identification: Although a PKI certificate contains a Distinguished Name or DN, the way this DN is constructed varies from place to place, and is often also influenced by national laws.
5. Quality control: Has the current certificate and the CA who issued it been properly verified?

Currently, the International Grid Trust Federation (IGTF) community has defined guidelines and authentication profiles to address these limitations to a certain extend. However, the effectiveness of the PKI depends heavily on the users to safe keep their private keys. In reality, there are Grid users who do not understand or care about the consequences of losing their private keys, thus compromising the security of the whole Grid. Such users may keep multiple copies of their private keys in different, possibly untrusted computers. In additional, users may belong to multiple Grids, thus they have to manage different private keys. Even if the users understand the importance of the private keys, they are still prone to human error. Like all other online e-Banking website, the SORMA platform need to minimise the chances of security risk due to human factor.

Authorisation, which is commonly based on a set of user attributes, however, cannot be effectively addressed by using certificate. Although a certificate can contain a set of user attributes for determining the user’s authorisation, these attributes however have to remain static. Another problem is that the same set of attributes does not apply to all. For example, in SORMA, the payment component may
requires the PayPal ID and the contract management component may require the identification number. As a result, PKI is not a scalable solution for passing changeable attributes for authorization, and you would rather have a delegated or distributed model: one service for authenticating the user; another service will provide the authenticated user’s accompanying attributes like name, email, or role, and manage which attributes need to be released to a certain component.

2.2.2 Single Sign-On

The main function of security in SORMA refers to the typical security in terms of perimeter access control. Participants in the Open Grid Market must authenticate themselves before they are eligible for access. Authentication and authorization across company or system borders to identify users and grant access permissions on the assigned resources can be inherently complex, but is of crucial concern within a commercial usage of Grids. To make participation easier, authentication must be single-sign on (SSO). SSO refers to the method of access control that enables a user to authenticate once and gain access to the resources of multiple software systems. It is the main requirement of a Grid infrastructure – to allow users different remote resources without having to repeatedly produce their pass phrase. It is also the main reason why standardization bodies have come up with standards for identity management.

2.2.3 Distributed Identity Management

To achieve true distributed identity management, a circle-of-trust needs to be established to enable federated identity. Here, identity federation refers to linking of two or more different identities stored across multiple distinct identity management systems belonging to the same user to form a assembled identity. It is designed to solve a long-running problem in both IT and systems security. From e-business transactions over the Internet to logins for the employee HR portal, uniform access control and robust management tools are required to securely enable connectivity for customers, partners and employees. Yet user databases and access policies are often fragmented, requiring multiple logins for users and repetitive tasks for systems administrators. Traditional SSO can address this to a certain
extend but it is not realistic to have a unified database. In a Grid where the participants are from multiple organizations, such a database might have to include up-to-date information on each of the organization's participants. There are many complexities in terms of provisioning user accounts in an environment consisting of tens of thousands of users from hundreds of organizations. If SORMA can setup several federated identity providers, the arduous task of managing user accounts can be shared. For example, there can be different identity providers to take care of users in different geographic regions. Another possible scenario is where SORMA delegates the authentication management of the users to their own organization. The benefits of this approach is that the users identity and information will be managed by their own identity providers and we can avoid privacy issue since any sensitive information do not need to be stored in SORMA. Another advantage is that each organizations can use their own authentication mechanisms. It does, however, require every authentication system to agree upon and be able to provide a common token, such as the SAML Assertion.

2.2.4 Trust Management

The issue of trust is prevalent in many online open trading systems. The trust in this context refers to the confidence of the buyers with the sellers, which is termed as “Soft Security”, in contrast to “Hard Security” (Rasmusson and Jansson 1996) which refers to traditional security mechanisms like authentication and access control. Many online websites like eBay, Yahoo and Amazon have their own reputation systems with slightly different recommendation mechanisms. Like those websites, it is obvious to see why having a reputation mechanisms is important in the SORMA.

The Open Grid Market functions just like any electronic marketplace. The appeal of reputation mechanisms is that, when they work, they facilitate cooperation without the need for costly enforcement institutions. They have, therefore, the potential of providing more economically efficient outcomes in a wide range of moral hazard settings where societies currently rely on the threat of litigation in order to induce cooperation. The “open-ness” of SORMA also means that buyers will interact with sellers who are complete stranger to them. With no known past history of the sellers, buyers would be reluctant to pay full prices given the uncertainty about the seller’s quality of service, hence would scale back their offers. High-quality sellers, however, would be reluctant to accept
discounted prices. Over time, high quality sellers would desert out of the market. Eventually, only the lowest quality sellers would remain. However, with an effective reputation mechanism, low quality sellers receive lower prices, leaving a healthier market with a variety of prices and service qualities. For example, sellers with stellar reputations may enjoy an extra premium on their services – a premium that users may be willing to pay for the security and the comfort of high quality services. Benefits of informative reputation systems return to both buyers and sellers.

2.2.5 Open Standards and Interoperability

In order to provide seamless integration with existing tools and applications, another requirement of SORMA is to leverage on existing standards and specifications for interoperability. For example, as the usage of X.509 certificate has become pervasive in many Grid deployments and tools, it is one of the requirements that is to be incorporated into the SORMA security infrastructure. Other standards that are considered include SAML, ID-WSF and WS-Security.

3 SORMA Security Framework

The security framework needs to address four different type of security issues – user authentication, securing communication, distributed identity management and trust management. The framework is composed of two main components, the Identity Provider component provides basic security and identity services, and the Credential Exchange which does mapping of security tokens and credentials. The framework will be implemented in the SORMA Prototype as the Security Provider, supporting the security requirements that the SORMA components required.

3.1 Basic Security in SORMA

This section describes the basic security concepts like authenticating user and securing communication between the SORMA components.

3.1.1 User Authentication

User authentication refers to the perimeter access control to prevent unauthorized access. In Grid
computing, the common way of user authentication is through digital certificate. The SORMA security framework takes a “certificate free” approach where users are not expected to own a Grid certificate. To be more precise, we do not mandate the means that authentication will be establish. The idea is that each organizations that is in the “federation” are able to use their own way of authenticating their users and still sign-on to SORMA using the same security credential. For the SORMA prototype, we will first use the simple approach of username and password as the authentication mechanism to simplify development. However, this can be easily change to more sophisticated authentication like multi-factor authentication.

3.1.2 Securing Communication

As the SORMA architecture is designed as a distributed platform, it is important to secure the communication where sensitive information may be transferred over the network for the following reasons:

- You cannot always be sure that the entity with whom you are communicating is really who you think it is.
- Network data can be intercepted, so it is possible that it can be read by an unauthorized third party, sometimes known as an attacker.
- If an attacker can intercept the data, the attacker may be able to modify the data before sending it on to the receiver.

SSL addresses each of these issues. It addresses the first issue by optionally allowing each of two communicating parties to ensure the identity of the other party in a process called authentication. Once the parties are authenticated, SSL provides an encrypted connection between the two parties for secure message transmission. Encrypting the communication between the two parties provides privacy and therefore addresses the second issue. The encryption algorithms used with SSL include a secure hash function, which is similar to a checksum. This ensures that data is not modified in transit. The secure hash function addresses the third issue of data integrity.
3.2 Distributed Identity Management

As described in Section 2.2 – Security Requirements, distributed identity management is one of the requirements specified by the SORMA description of work. This means that the user accounts are not centralized in any one single repository and possibly even have multiple accounts of same user in different location. This creates a challenge of federating the different repositories, or in identity management term, better known as identity providers. In this section, we will first describe the various popular identity standards and tools that exist today and in the next section, discuss about how this issue is addressed in SORMA.

3.2.1 Identity Management Standards and Tools

In this section, we describe the various identity management standards and tools that are commonly used today.

3.2.1.1 Liberty

Liberty is the project formed by the Liberty Alliance to work on the federated identity management system driven by real business needs. It specifies and provides single sign-on feature, access controlling of the principals through exchanging data using SAML. The objectives of Liberty are to:

- Enable consumers to protect the privacy and security of their network identity information
- Enable businesses to maintain and manage their customer relationships without third-party participation
- Provide an open single sign-on standard that includes decentralized authentication and authorization from multiple providers
- Create a network identity infrastructure that supports all current and emerging network access devices

The Liberty Identity Federation framework (ID-FF) protocol are based on SAML 1.0 (and later SAML 1.1). In 2005, the Liberty Alliance recognized the need for a unified federation standard and decided to contribute ID-FF back into SAML 2.0. As such, SAML 2.0 is a critical step towards full convergence for federated identity standards. Figure 2 demonstrates the relationship between the OASIS SSTC and
Microsoft, along with its partner IBM, created their own identity federation specification called WS-Federation as part of the overall effort to build a Webservice Security Framework. It has been created to support the federation of security domains, which allows users whose identity information exists in one security domain to access resources in another security domain. This is enabled by the ability to broker identity data including identity, authentication, authorization and attribute information between security domains, through the use of mechanisms selected based on the relationship of trust that exists between the security domains. WS-Federation provides a standard way for security domains to exchange meta-data in order to indicate the trust relationships between these security domains.

WS-Federation exists within a family of other WS-*standards. It rests on a foundation of WS-Security and WS-Trust that contain the primitives necessary to define security tokens, trust topologies and security infrastructures. The primitives defined in WS-Policy and WS-SecurityPolicy, along with extensions that are defined as part of the WS-Federation specification, are intended to support the definition of policies that identify what the supported and required components of a federation are by participating members, and what the choices are for communicating the policy information between federation participants.
WS-Federation has been submitted to OASIS standardization consortium for advancement and input from the broader community. It is currently still under on-going standardization process.

3.2.1.3 **OpenID**

OpenID is a user-centric decentralized single-sign-on system that eliminates the need for multiple usernames across different websites, simplifying the online experience. OpenID is designed to be a lightweight method of identifying individuals that uses the same technology framework that is used to identify websites. It originated from the open source community and the OpenID foundation was formed to assist the open source model by providing a legal entity to be the steward for the community by providing needed infrastructure and generally helping to promote and support expanded adoption of OpenID.

OpenID starts with the concept that anyone can identify themselves on the Internet the same way websites do – with a URI (also called a URL or web address). Since URIs are at the very core of Web architecture, they provide a solid foundation for user-centric identity. A user with an OpenID identity can login to any websites that is OpenID-enabled by entering their OpenID URI, without the need to create separate accounts. The basic idea is that there are services such as blogs and wikis where users come and leave their comments or modifications without actually having user accounts at the service providers. In a small-scale identity system, an identity server can tell e.g. a blog server that the user has logged in at the identity server (which can even be running on the user's own computer) with their OpenID identity. For example, a website such as example.com wants to enable OpenID logins for its users will need to place a login form which ask for users OpenID identifier, instead of user name and password. A user name John has registered an account with an OpenID provider (myOpenID.net) and wants to login to example.com using his OpenID identify. He simply goes to example.com and enter his identifier, which in this case is john.myopenid.net, in the OpenID login box. The replying party (example.com) first access the web page located at the URL http://john.myopenid.net and discover the provider server. If John has not logged in to his provider yet, his browser will be redirected to a login page from his provider, myOpenID.net. After he is authenticated, his provider will ask for authorization to send example.com details about his identity. If authorization is given, John is considered logged in to example.com as “john.myopenid.net” and the site may store the session, or, if this is his first logon,
prompt John to enter some information specific to example.com, to complete the registration.

3.2.1.4  Security Assertion Markup Language (SAML)

SAML is the XML-based language for exchanging authentication and authorisation data between an identity provider and a service provider. The fundamental problem that SAML is trying to solve is the web browser single sign-on (SSO) problem. It defines message formats for Queries and Responses, as well as a request-response protocol in SOAP over HTTP for carrying the SAML messages. SAML Queries are sent to a decision-making service whilst Responses, in the form of SAML Assertions, are returned. These assertions can then be coupled with a further Query and sent to other decision making services to aid them in their own decisions. In the SAML model there are three decision-making services: the Authentication decision-making service, the Attribute decision-making service and the Authorisation decision-making service.

SAML consists of SAML profile, binding, protocol and assertion

- SAML assertion is the statements sent by identity provider used to communicate to service provider. The assertion will be used by service provider to response to user's request of services.
- SAML protocol is the set of rules to govern how the SAML elements are included into SAML request or response elements. Example of SAML request element are attribute query. Identity provider replies attribute query by sending SAML response whose content is SAML assertion.
- SAML binding is the mapping of the SAML protocol message into standard messaging or protocol used for communication. Usual binding is with SOAP and HTTP POST.
- SAML profile is the description of mechanism how the SAML assertion, protocol and binding is combined to perform defined SAML use-case.

The latest version of SAML is version 2.0 and is incompatible with the previous version 1.1. SAML 2.0 unifies the previous disparate federated identity building blocks of SAML 1.1 with input from both higher education's Shibboleth initiative and Liberty's Identity Federation Framework (Liberty ID-FF). As such, SAML 2.0 is a critical step towards full convergence for federated identity standards. It is
strongly supported by many major vendors like Oracle, IBM and HP, and is already widely adopted in
the industry. Thus, SAML will be the chosen protocol that is to be used in SORMA.

3.2.1.5 Shibboleth

Shibboleth is an Internet2 Middleware Initiative project that has created an architecture and open-
source implementation based on SAML 1.1. It provides a federated single sign-on and attribute
exchange framework, as well as extended privacy functionality allowing the browser user and their
home site to control the attribute information being released to each service provider.

With Shibboleth, a user can authenticate against his own user database to access a web resource on
another institutions domain, regardless of his location. Users are not logged onto the resource itself,
instead, attributes for the user are sent to the resource from the home institutions database, these
attributes are then used to decide the type of access the user has on the site. For example, a student
authenticates against their own user database to gain access to a repository at another university,
specific attributes for the user will be released to the repository, in this case it may just be the course
the student is on, or simply the institution they are from, the repository will then use these values (and
not the initial logon credentials) for any authorisation decisions that need to be made. Besides an
identity provider and service provider, Shibboleth defines an optional “Where are you from?” (WAYF)
service. The WAYF can be used by the SP to determine the user's preferred IdP, with or without user
interaction. The WAYF is essentially a proxy for the authentication request passed from the SP to the
SSO service at the IdP.

Shibboleth is mostly used in the academic world today. Although the current Shibboleth version 1.3
supports only SAML 1.1 today, the next version 2.0 will be based on SAML 2.0.

3.2.1.6 Sun Access Manager / OpenSSO

Sun Java System Access Manager is Sun Microsystems' web access management product and a
component of Sun Java Enterprise System. Access Manager helps organizations manage secure access
to an enterprise's web applications both within the enterprise and across business-to-business value
chains. It is an an open, standards-based product that provides centralized authentication and policy-
based authorization from a single, unified framework. Access Manager meets the current needs of the enterprise for secure protection of essential identity and application information and supports their future business needs through implementation of the latest identity federation standards for tighter integration with business partners. It improves user experience through single sign-on to all of an organization's web-based applications and creates revenue opportunities through deepened relationships with partners, suppliers, and customers.

Sun has released the source code of Access Manager publicly under the Open Web Single Sign-On (OpenSSO) project. Currently, OpenSSO supports multiple security and identity management standards like SAML, WS-Security and Liberty. Figure 3 shows the internal architecture of the Access Manager.

![OpenSSO Architecture](image)

**Figure 3: OpenSSO Architecture**

Supported Standards:
Java Authentication and Authorization Service (JAAS)

Kerberos

Liberty Identity Federation Framework (Liberty ID-FF)

Liberty Identity Web Services Framework (Liberty ID-WSF)

Lightweight Directory Access Protocol (LDAP)

Security Assertion Markup Language (SAML)

SOAP (Simple Object Access Protocol)

SSL (Secure Sockets Layer)

WS-I Basic Security Profile tokens

XML Digital Signature

XML Encryption

In upcoming release, the product will be called Federated Access Manager (FAM) 8.0 that will be built from the same code base from OpenSSO. Besides the standards that are already supported, FAM will also include support for WS-Federation and OpenID. The key idea is to develop FAM as a multi-protocol federation hub for providing single sign-on across service providers that are using different protocols. For example, a user that is already authenticated to FAM can access to all service providers using SAML, WS-Federation and OpenID.

Today, there are very few open source software that are able to support such a wide variety of security protocols. Since in some part, the success of the SORMA rests on its acceptance by the industry, so using an open source software that is backed by a major vendor like Sun Microsystems will help to a certain extend. Access Manager also allows the flexibility of extending SORMA to support other security protocols in the future, therefore Access Manager has clear advantage over other open source software from both technical and commercial perspectives.
3.2.2 Identity Management in SORMA

In this section, we discuss how is distributed identity management supported in SORMA.

3.2.2.1 SORMA Identity Provider

As there may be cases where users or small groups of users who are not affiliated with any organizations that have their own identity provider (IdP), SORMA will have its own identity provider for them. The SORMA IdP provides the necessary service for these prospective users to register for an account. When users register for an account, they will choose a username and password which they can be subsequently use to authenticate with the SORMA IdP.

As part of the requirement in the SORMA Description of Work is to achieve identity federation, the SORMA IdP needs to support federation standards and mechanisms, like those specified by SAML. The Sun Federated Access Manager (FAM) provides all the SAML functionality and is able to achieve most of the requirements regarding identity federation. However, to achieve federation requires complex configurations in setting up the circle-of-trust. Therefore we will not first consider multi-IdPs scenario in the coming version of the prototype implementation, and address it in the later stages of the project.

3.2.2.2 Web Portal Login

The front end interface of SORMA is a portal interface hosted in GridSphere Portal Framework. As we will be using FAM as our IdP, we need to modify the login mechanisms in GridSphere to delegate the authentication to it. To do this, we will be adding a SAML Handler method which will interact with the SORMA IdP using the protocol defined by the SAML Web Browser SSO Profile. Figure 4 shows the flow when a user logs on to the SORMA GridSphere portal.
1. User requests for a webpage from the portal server.

2. The SAML Handler intercepts the request and looks for a security token. If the user does not have a valid token, the user is redirected (via the browser) to the SORMA IdP for authentication.

3. User authenticates him/herself to the SORMA IdP which acts as the authentication authority. Once authenticated, the IdP issues a security token to the user (which is stored as a cookie by the web browser).

4. If login is successful, user is redirected back to the requested webpage. As before, the SAML Handler intercepts the request and validates the security token. If the token is valid, the user is granted access to the webpage.

Now that the user is authenticated by the IdP and has received the security token, the user can access any web resources that trust the IdP by presenting the token. We want to highlight that this scenario is using web SSO where communications are between browser requesting for a webpage from the webserver. Figure 5 shows in more detail the interactions of the login process.
It is important to note that the HTTP redirections are transparent to the user. We have done a preliminary integration of GridSphere with the Sun Federation Access Manager. The following screen captures show what the user will see when using the SORMA portal. First, when the user accesses the portal, he/she will see the main page as show in figure 6.
The main page will show list of identity providers that is trusted by SORMA (there is only one IdP in this test). Users have to choose the one that they have an account with. Selecting one of the links will direct the user to their identity providers login page, such as the Sun Access Manager as shown in figure 7.
Note that the login page shows that the Access Manager is configured to use LDAP authentication, thus username and password are need. Since Access Manager can be configured to use other authentication modules, the means of authenticating the user can be different for other identity providers. Once the user is authenticated successfully, the identity provider will respond with a HTML form and perform an automatic HTTP Post using Javascript to the portal with a SAML assertion. This is completely transparent to the user. Now the portal receives the SAML assertion with the request, and assuming that it is valid, the user will be allowed access to the request page, as shown in figure 8.

Figure 7: Access Manager login page
Our integration of GridSphere with Sun Access Manager using the SAML Web Browser SSO Profile with the POST Binding works well. However there are one issue that still need to address. GridSphere keeps their own database of users account, so when a newly registered user who is authenticated and log in to GridSphere but do not have an account in the GridSphere database, some of the user management portlets will fail. Thus we need to make modification to the GridSphere source code to either automatically create a default user account or change the user management portlets to interface with Sun Access Manager.
3.2.2.3 Security and Identity Service

Besides the Portal interface, SORMA also provides Intelligent Tools that perform actions on user's behalf to interact with the SORMA system (example, places a bid for a resource). Thus these tools need to have a programmatic mechanisms of authentication and achieving single sign-on. For this purpose, the SORMA IdP provides a web service interface that allows remote authentication, authorisation and querying of user's attributes. The WSDL of the interface is shown as figure 9. Most of the details are left out for brevity.

```xml
...<operation name="log">...<operation>
<operation name="attributes">...<operation>
<operation name="authenticate">...<operation>
<operation name="authorize">...<operation>
</portType>
...
```

Figure 9: Identity Service provides four operation

This service issues and uses single sign-on token (SSOToken) as the security token. The authenticate operation, for example, returns the SSOToken ID, which is an unique identifier to user's authenticated session, if the authentication is successful. Using the ID, the intelligent tools and other SORMA components can query user related information from the SORMA IdP.

3.2.3 Credential Exchange Service

As SORMA aims to connect to different type of Grid middleware (e.g. Globus Toolkit, GridSAM, Sun Grid Engine), it poses a huge security challenge since the security needs of different systems vary widely. In particular, independently managed Grid Services vary with regard to the type of security token (credential) used to prove user identity (username and password, X.509 signing, Kerberos, etc.).
Thus using a single credential for authentication across the different SORMA resource/service providers is out of the question. A naïve way of addressing this is to let users manually manage their security tokens. Users could acquire all possible credentials upfront for each session, then select and present the necessary credentials on a per-service basis. Even for especially skilled and vigilant users, this approach is tedious and insecure.

The purpose of the Credential Exchange Service is to address this need of enabling single sign-on (SSO) to the resource/service providers. The objective is to allow users to store their credentials (e.g. certificate A for provider A, certificate B for provider B, username/password for provider C) in a trusted SORMA service. To access the SORMA system, a user will login and acquire his or her SORMA security token. In order to engage a service from one of the providers that do not recognize this credential, the user's SORMA security token is exchanged for another credential that is recognized by that provider.

The idea of exchanging security token has already been tried and tested. MyProxy (Novotny, Tuecke et al. 2001) from the Globus Toolkit supports exchanging of username/password to Grid credential (GSI). Dorian (Langella, Oster, et al. 2006) is another grid user management service that provide an Identity Federation Service that works like MyProxy, but accepts SAML Assertion instead of username/password. Another tool called the CredEx (Del Vecchio, Basney et al. 2005) is inspired by MyProxy but supports more than just the password-Grid credential exchange and uses WS-Security and WS-Trust standards. It is the first freely-available general-purpose credential storage and exchange service that solves most the token management problems. Except lacking the support for SAML, CredEx appears to be the best tool that matches most of our needs. However, the project seems to be no longer active since 2005 and the source code of CredEx is no longer available for download. As such, we will be building our own credential exchange service based on the design and concept of CredEx.

The implementation of the Credential Exchange Service will proceed in parallel with the development of the SORMA prototype. In order not to delay the implementation of the security concept in the prototype, a dummy web service interface that only allow exchanging of SAML Assertion with a default grid credential will be provided. In the longer term, we will work on extending the service to support multiple types of resource credential.
3.3 Trust as Soft Security

Trading over the Internet has become popular of the last decade. Experience has shown drawbacks in electronic trading which were not known to this extent in common face-to-face transactions. A bilateral exchange of resources inherits the risk that the party who moves first is exposed to the behavior of the other party (Dellarocas, 2001). The second mover has the opportunity not to deliver the resource as a seller or to pay for the resource as a buyer (moral hazard). Bolton et al. examined in an experiment that especially in a market with a reputation system, participants tend to defect less than in a stranger market, where the parties do not have information about the counterpart (Bolton, Katok et al. 2002). Trust is based on the successful transactions of the parties on the platform. Thus, trust can be seen as a precondition for a successful market (Dunn, 1984; Gambetta, 1990).

Electronic markets like the SORMA Open Grid Market can gain a large set of sellers and buyers. Formal institutions can increase trust by sanctioning misbehavior. However, in the scope of SORMA for a wide application of the Open Grid Market several jurisdictions and conflicting legal systems have to be considered (Johnson and Post, 1996). In addition to hard security (like Identity Management), trust as a soft security is required to set incentives for cooperative behavior and publish information about the behavior of other participants. Another obstacle is to identify whose fault it was, when a computing job was not successfully finished. It can be the breakdown of a machine or an intended cancelation of the job (seller’s fault) or a programming mistake in the source code (buyer’s fault) (Humphrey and Thompson, 2001). To overcome these kind of problems, online reputation mechanisms are a promising approach to enhance trust on the Open Grid Market.

3.3.1 Definition of Trust

The term 'Trust' is widely discussed in the social science community and there are numerous definitions of trust. Some researchers stated that this term has become more confusing than clarifying over the years (Barber, 1983; Taylor, 1989; Hosmer, 1995). Trust has been described as an elusive concept (Yamagishi and Yamagishi, 1994). In computer science trust gained in interest with the advent of P2P-Computing and Grid Computing. Many research was done on this topic (Alice and Friedman, 2005), (Friedman and Resnick, 1998; Resnick and Zeckhauser, 2000; Buragohain, Agrawal et al. 2003; Damiani, di Vimercati et al. 2002; Kamvar, Schlosser et al. 2003; Kwok, Song et al. 2005; Papalilo,
Friese et al. 2005; Alunkal, Veljkovic et al. 2003). In this context three different terms have to be distinguished: trust, recommendation and reputation. We refer to the definitions of Josang, Ismail et al. (2004), Haller (2006), Azzedin and Maheswaran (2002), Alunkal et al. (2003), Golembiewski and McConkie (1975):

- Trust is a subjective probability by which one party expects another to behave cooperatively to achieve the desired output. This expectation is only valid within a certain timeframe for a specific context. Hence, trust implies some degree of uncertainty as to the outcome and the resulting cooperative behavior implies the acceptance of risk.

- Reputation is the objective, business context specific aggregation of trust values from multiple independent sources to support one party’s decision making process with respect to an intended collaboration with another party.

- Recommendation (also refereed as opinions or ratings) is a single subjective value of experience with another party. The aggregation of different recommendation values from different parties will result in the objective reputation value.

It is obvious that these terms are closely related. In a stranger market, where an organization A has no experience with organization B, A will derive his initial trust from the reputation of B. The reputation of B is the record of recommendations of other parties in the network. Consequently, trust is based on the recommendation as well as on the behavior of all the participants in the network.

### 3.3.2 Existing Trust Mechanisms

In an (electronic) trading environment trust is exposed to the moral hazard and adverse selection problem. It is a common assumption that participants in a network are self-interested acting agents. E.g. in a bilateral exchange the buyer faces the risk to be cheated, if he pays first. In the moral hazard case reputation mechanisms enforce participants to act cooperatively and honestly. Otherwise the participants are sanctioned with lower reputation (Dellarocas, 2005). Issues related to adverse selection are unknown behavior of the counterpart. A reputation mechanism documents past behavior and provides the participant with additional information about the expected future outcome, if he trades with this counterpart. Thus, in the adverse selection case the objective of a reputation mechanism is to provide sufficient information to learn about the counterpart’s behavior.
P2P-Computing is an insecure environment due to the distributed nature of the network structure. There was a need to ensure trust in P2P networks without a central authority. Several reputation systems were proposed during the last decade for these distributed networks, e.g., Aberer and Despotovic (2001), Kamvar, Schlosser et al. (2003), Damiani, di Vimercati et al. (2002), Xiong and Liu (2003), Wang and Vassileva (2003). However, all of these approaches assume that there is no central authority, which leads to a more complex scenario (from the incentive engineering perspective as well as the technical perspective). This assumption is not entirely applicable to the Grid network. There are already central authorities (CA) for ‘hard securities’ to create certificates for the participants. These CAs can provide a global reputation base with a web service, where participants can request a reputation value of a certain member. This approach was considered by Hwang and Tanachaiwiwat (2003), and Goel and Sobolewski (2003).

The idea of Azzedin and Maheswaran (2002) was to schedule incoming computing jobs based on trust level and the load balance of a machine. Each job has an expectation about the required trust level based on the importance of the job. The weighted local and global reputation of the machine are calculated with respect to a decay function (older recommendations are lower weighted). The load balance is included by the minimum completion time of each machine. The authors propose two more heuristics for batch mode scheduling. However, this model is not suitable for the Open Grid Market, as the allocation of jobs is not directly based on reputation. The price and the defined SLA are the major criteria for allocating the jobs. Reputation can be a part of the defined SLA, but not the main criterion.

Goel and Sobolewski (2003) focus on Grid Services, which are selected by the quality rating of the services. Each services is evaluated after usage by the buyers. A composition of services induces a complex contemplation of all recommendation of each service. Then, for every service a requested reputation level has to be defined according to the necessity of each service. The proposed model was not validated by simulations or an experimental testbed. Though an integration into the Open Grid Market can be a part of the SLA. Reputation as one criterion has an influence for complex service selection, if the price plays a minor role for the consumer.

Lin, Varadharajan et al. (2004) present an architecture for trust management. They break the term trust down into authentication trust, execution trust, code trust, direct trust and recommended trust. In particular, execution trust defines the trust a resource consumer has in the provider that the provider
will not harm the participant’s running job or send a faked result back. From the provider’s view code
trust comprises the belief of the provider that the job does not contain and execute malicious code on
his machine. As the authors themselves state, it is difficult to identify and measure these kinds of trust.
However, these could be a metric to evaluate the reputation of each provider and consumer. A
participant can view the aggregated value and derive more information by requesting detailed trust
metrics like code trust. This model, however, was only described as a possible solution without any
validation of the system drawback like ballot-stuffing (increase own reputation with fake account) or
bad-mouthing (decrease reputation of counterparts).

3.3.3 Reputation system in SORMA

A reputation system can enhance trust in the SORMA Open Grid Market. The design of such a system
provides a variety of implementation possibilities. Gupta and Somani (2004), for example, present a
reputation based allocation mechanism for P2P networks, where participants are not allowed to
download files from sites with a higher reputation than their own. Policies like these are already
influencing the allocation mechanism. In SORMA the allocation is mainly based on economic
parameters. The Trading component communicates with the Economic Enhanced Resource Manager
(EERM) and calculates the optimal outcome for a given set of resource request. Thus, introducing a
new parameter like reputation can lead to a new allocation of jobs and resources. The SORMA market
currently applies two markets mechanisms: the Decentralized Local Greedy Algorithm and a
continuous double auction. The primary factor of the allocation process is the price. Further parameters
like timeslot, duration or quality of service are defined as constraints. A reputation system could be
included in the allocation process as a constraint, too. E.g. A participant can ask for a site, which has at
least 90% of positive feedback to ensure to receive an almost reliable site. Obviously, jobs will always
ask for the most reliable site. In case of scarce resources, jobs have to decide between paying more for
a reliable site and less risk of job loss or vice versa.

In SORMA the intention is to test different reputation systems, which are suitable for Grid Computing
(Kwok et al. 2005), (Alunkal et al. 2003), (Papalilo et al. 2005). Although there were proposed a wide
variety of reputation systems for P2P networks, in Grids the requirements differ from the requirements
of P2P networks. In the latter there are n rational, self-acting agents, whereas in Grids the agents belong
to a site or they are member of a VO. Instead of providing a reputation value for every single agent, the value could be aggregated for each site or each VO, since not the participant is providing the resources, but the entire site with his members.

At first, it is important to test the effects of behavior to the Open Grid Market without reputation. A possibility is that due to the necessity of certification authority known from the scientific Grid, trust is not an issue. In a second step, a simple reputation mechanism with binary feedback can be introduced, if necessary, to isolate and analyze the impact of reputation mechanisms on the Open Grid Market. Reputation is based on the evaluation of a consumer about a site. Decision parameters are the accuracy of expected job duration time, machine breakdown or useless results. In the extended prototype a more complex scenario can be considered, where users’ behavior is evaluated as well and every user’s behavior has an impact on the site’s reputation, the user belongs to. This setting incorporates new incentives for users and sites, which need a deeper analysis. Since incentives and punishment have to be defined in every contract, the analysis will be done in conjunction with the Contract Management.

4 Security in SORMA Components

In the initial version of the SORMA prototype, most of the security issues are not considered as it will needlessly increase the complexities of the design. The next version of the prototype will incorporate identity management and basic security concepts. In this section, we discuss the security issues that need to be addressed in the SORMA components.

4.1 Core Services

The security model of the core services provides an interface to external security providers for authentication of participants. It is based on a set of generic objects and functions that allows the integration of diverse security mechanisms like simple user/passwords or certificates.

The three main abstractions of the security model are based on the terminology proposed in the Internet Security Glossary (IETF, 2000)

- **Token:** is a data object used to verify an identity in an authentication process,
- **Credential**: A Credential is the "data that is transferred or presented to establish either a claimed identity or the authorizations of a system entity. Includes the principal's name, an authentication token and optionally a public key.

- **Access Ticket (or capability)**: gives the bearer or holder the right to access the core services. Possession of the ticket is accepted by a system as proof that the holder has been authorised to access the resource(s) named or indicated by the token.

Based on this two concepts, the process for authenticating an user and validating his/her access to the GMM's diverse services can be resumed as shown in the following figure.

![Diagram of Security verification process in the GMM](image)

**Figure 10: Security verification process in the GMM**

1. When a participant wants to enter the GMM, it must first obtain the security **Token** according to
the security mechanism in place in its institution (e.g. user/password or a security certificate) and uses it to build its *Credential*.

2. The participant then authenticates up front to the SORMA Security Provider presenting its *Token*, which validates them using an external, pluggable security provider.

3. If the *Token* is accepted, the Security Provider, accordingly to the security policies in place, returns an *AccessToken* which grants the participant the access for a given period to a set of specified services.

4. This *AccessToken* can then be presented by the participant to the other services in the GMM framework (Exchange, Information, Directory) which in turn call the Security Provider to validate it and return a *Session* representing this interaction of the participant with the service.

5. When the participant request an operation to a service specifying a *Session*, the service can check with the Security Service if the participant is allowed to perform it.

Each of the services in the GMM also offer security related functions

- **Market Exchange Service**: provides a secure communication mechanism to the participants in a market by:
  - Encrypting messages to avoid tempering and eavesdropping
  - Automatically signing offers and agreements between participants to provide non-repudiation of transactions
- **Logging Service**: provides a secure storage of transactions for auditability
- **Directory Service**: Supports policies to restrict the visibility of the resources and participants to queries from other participants.

### 4.2 Trading Management Component

Conversation Space (C-Space) constitutes the Trading Management component in SORMA. This
section first gives a short introduction to C-Space and then specifies the security related requirements of the C-Space model.

4.2.1 Introduction to C-Space

C-Space is a framework for creating and executing conversations. In SORMA, conversations are typically instances of auctions or other kinds of market protocols. A conversation follows a certain protocol that determines who can say what, and when. These protocols are written in the Java programming language. The protocols are executed in a sandboxed computing environment by the component responsible for executing the protocols, the Protocol Executor. The Protocol Executor provides a controlled access to system resources such as libraries, native code and externally observable state through virtualized functions.

Conversation protocols are defined in terms of *in-words* and *out-words*. The in-words of a specific protocol is the set of “words” that the agents (traders) are allowed to say in a conversation defined by that protocol. The out-words is the set of words produced in a conversation as an effect of an in-word. The out-words of a conversation should be “understood” by the agents participating in that conversation. A user can participate (i.e. to say something) in a conversation by sending an in-word to a Protocol Executor. In addition to processing the in-word, the Protocol Executor stores the in-word, the conversation’s identifier, a timestamp and the identity of the user, for auditing purposes.

4.2.2 Security Related Requirements

A number of requirements need to be provided by the trading infrastructure.

1. Users must be able to identify themselves in a legally binding way to the SORMA infrastructure.

   The state of a particular conversation is regarded as evidence that certain events have occurred. For example, if a conversation is an English auction with variables specifying the current highest bid $b$ and the identity $u$ of the user submitting the bid, then its state is a proof of that user $u$ has bid $b$, and that $b$ is the highest bid. This makes it necessary for users to have identities, and that these identities can be authenticated by the Protocol Executor.

2. Users must be able to bid anonymously.
In many applications, such as auction protocols, it is desirable to keep user’s identities secret. This requires that messages (in-words and out-words) are encrypted to prevent eavesdropping, and that the states of conversations are not visible other than through the out-words explicitly produced by the conversations. The latter requirement is met by running the Protocol Executor within a trusted infrastructure. Given that the Protocol Executor is trusted, there must also be a way for users to authenticate out-words they receive.

3. Assured message delivery.

Messages should not be lost so that users are not informed of changes to the auction state.

In SORMA, users' agents and Protocol Executors use SOAP to communicate with each other. Currently, message encryption and two-way authentication is provided by TLS (Transport Layer Security), which is a layer between HTTP and TCP. The users are currently identified with x509 certificates that have been signed by a SORMA-trusted Certificate Authority.

It is under consideration to move this responsibility to inside the SOAP messages by using WS-Security. This would enable the Protocol Executor to also store the signature of each message received.

Anonymous bidding is achieved by running the Protocol Executors on a trusted site. That guarantees that the messages (in-word) are not revealed to the parties unless specified by the protocol.

The protocol execution is completely logged, so that they can later be replayed during an audit. This requires capturing any protocol dependencies of external state such as the time (for time outs – “going, going, gone”) or native java calls.

C-Space itself does not guarantee assured message delivery, but instead relies on the use of a message delivery mechanism that routes, delivers and resends the messages, as necessary.

The C-Space model itself needs no means to authorize users to participate in conversations, because it is up to the protocol to determine who can say what and when. It is assumed that if a user can be authenticated, then he is also authorized to participate.
4.3 Payment Component

The Payment Component is a SORMA component capable of managing payments for the services offered through SORMA system. The current implementation is based on (but not bound to) the popular PayPal® system.

The component itself is composed of two modules:

- A PayPal® wrapper, where the objects from PayPal® SDK are created and used.
- A business logic module, which manages all the input/output data from/to external tools.

The following figure shows the system architecture of the Payment Component:

![Payment Component Architecture]

Since the PayPal® wrapper and the business logic are within the same object, and there is no connection to other external resources when moving information from one module to the other, there’s no need to implement particular security policies within the module.

It is anyway necessary to implement a security connection between the web service (which is responsible for passing the user data to the Payment Component) and the Payment Component itself,
and between the web service and the portlet GUI (a graphical user interface for gathering necessary data). All these connections will be made on HTTPS security channels.

When communicating to PayPal®, it is necessary to use the PayPal SDK, so to rely on their security mechanism. PayPal® official documentation reports that every connection among its servers and other external servers is performed in a secure way using HTTPS protocol.

4.3.1 Input information needed by Payment Component

In order to understand the level of security that is required, we report the information that is required by the Payment Component. Such input is divided into two categories:

1. End-user information (the one PERFORMING the payment).
2. Provider-user information (the one RECEIVING the payment).

End-user information:

- Mail: a string which contains the mail of the user, as registered in PayPal®.
- Service Name: a string which contains the name of the service purchased.
- Service Id: a string which contains the ID of the service as assigned by SORMA system.
- Service Cost: a double value with the amount of money to be paid.
- Currency: a string with the code of the currency in use.
- Credit Card Type: a string with the name of the credit card.
- Credit Card Number: a string with the number of the credit card.
- Expiration Date: an integer with the expiration data of the credit card in the format mmyyyy.
- CVV2: an integer with the three-digit security code of the credit cards.
- First Name: a string with the first name of the owner of the credit card.
- Last Name: a string with the last name of the owner of the credit card.
- Street: a string with the address of the owner of the credit card.
City: a string with the city of the owner of the credit card.

State: a string with the state (if living in America) where the owner of the credit card lives.

Zip: an integer with the postal code of the city of the owner of the credit card.

Country Code: a string with the international code of the country where the owner of the credit card lives.

Provider-user information:

- **User Name**: a string with the user name of the person who receives the payment, as registered in PayPal®.
- **Password**: a string with the password of the person who receives the payment, as registered in PayPal®.
- **Signature**: a string with the check signature of the person who receives the payment, this is assigned by PayPal® when registering as a vendor.

### 4.4 Economically Enhanced Resource Manager

The overall aim of the Economically Enhanced Resource Manager (EERM) is to isolate SORMA economic layers from the technical layers and orchestrates both economic and technical goals in order to achieve maximum economic profit and resource utilisation. The EERM has interactions with several components that have some security requirements.

- **Client Application to EERM**:
  - **Job submission**: when a client requests EERM to submit a job, it will require a valid agreement ID as defined by the contract management. It is also required that the clientis authenticated and has a valid security token.
  - **Request for checking Job Status or retrieve results**: clients needs to be authenticated and authorised and has a valid security token.
● EERM to Monitoring:
  ○ Requests for Monitoring data: EERM must be authorised and have an account on the gateway (or be in the Virtual Organization that the gateway is subscribed to). This security system is external to SORMA security.

● EERM to Tycho-GridSAM:
  ○ Job submission, read status, read results: EERM must be able to convert the credential into an appropriate type for the underlying resource, i.e. the type of credential token used, and appropriate user account detail. Probably Resource Fabrics (Tycho-GridSAM) should be able to access Credential Exchange Service in order to check credentials.

● Intelligent tools to EERM:
  ○ EERM needs to be sure that communications from Intelligent Tools is secure and encrypted. It also needs to authenticate and authorise requests from the Intelligent Tools.

● SLA component to EERM:
  ○ EERM needs to authorise and authenticate requests.

### 4.5 Contract/SLA Management Enforcement and Billing Component

The overall aim of the SLA management components is to define, monitor and enforce a service level agreement as per the terms and condition negotiated between the resource provider and consumer. The SLA Management/Enforcement/Billing interacts with other SORMA components, which imposes certain security requirements. These requirements can be divided into three main stages:

1. Sign SLA
2. Authenticate/Authorize SLA monitoring
3. Payment/Billing Processing

- Sign SLA: In the first case it is required that once a service level agreement has been agreed between the provider and consumer, to make it legally binding it has to be signed by the security...
Authenticate/Authorize SLA monitoring: In the second case, it is assumed that the request to access monitoring data from SLA enforcement component to EERM will be based on simple Authentication/Authorization and that it will be based on a one-shot process where the request to access the data is made and the security service grants or denies it.

Payment/Billing Processing: In the third case, the billing component needs to check consumer and provider credentials so a payment can be processed. Once the credentials have been verified the security service needs to establish a signed token from consumer authorizing to process billing amount agreed in an SLA. Once this is established, the token needs to be sent to the billing component. The billing component sends the SLA details, signed token, consumer/provider certificates and account details to the payment component so the billing can be processed. The SLA details include the consumer id, provider id, job id, cost and currency. The information contained in the token includes, credit card details. The consumer/provider account details include names, addresses and paypal account id. Using this information a payment component can process the required billing for an SLA.
Tables

Table 1 .................................................................................................................................................. 1
Table 2 .................................................................................................................................................. 2

Figures

Figure 1 ................................................................................................................................................ 3
Figure 2 ............................................................................................................................................... 15
Figure 3 ............................................................................................................................................... 19
Figure 4 ............................................................................................................................................... 22
Figure 5 ............................................................................................................................................... 23
Figure 6 ............................................................................................................................................... 24
Figure 7 ............................................................................................................................................... 25
Figure 8 ............................................................................................................................................... 26
Figure 9 ............................................................................................................................................... 27
Figure 10 ............................................................................................................................................ 34
Figure 11 ............................................................................................................................................ 38
References


Barber, B. (1983), The Logic and Limits of Trust. New Rutgers University Press, Brunswick, NJ.


Bolton, G., Katok, E., Ockenfels, A. (2002). How effective are online reputation mechanisms?.


Golembiewski, R. T. and McConkie, M. (1975). The Centrality of Interpersonal Trust in Group
SORMA D4.1: Description of the Security Framework of the Open Grid Market


Gutmann, P. (2001). Everything you never wanted to know about PKI but have been forced to find out, Tutorial, http://www.cs.auckland.ac.nz/~pgut001/.


SORMA Consortium

This document is part of a research project partially funded by the IST Programme of the Commission of the European Communities as project number IST-FP6-034286.

Barcelona Supercomputing Center

BF/M Bayreuth, Universität Bayreuth

Cardiff University

Correlation Systems Ltd

FZI Forschungszentrum Informatik

Hebrew University

Institut für Informationswirtschaft und -management (IISM), Universität Karlsruhe

Sun Microsystems

Swedish Institute of Computer Science

TXT e-Solutions

Universitat Politècnica de Catalunya

University of Reading