

ACTIVE

Deliverable 3.4.1

Security in Knowledge Processes

Editor:	Carlos Ruiz Moreno, iSOCO
Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	M12 (28 February 2009)
Actual delivery date:	M12 (28 February 2009)
Suggested readers:	
Version:	1.0
Total number of pages:	42
Keywords:	Security, security policies, knowledge processes, KPs

Abstract

The importance of Knowledge Processes management is fundamental for enhancing the value of a corporation. The typical ACTIVE scenario is collaborative, decentralized and heterogeneous, where knowledge is defined, used and shared across different groups and domains. Articulating enterprise knowledge in form of Knowledge Processes allows users to create social relationships forming working and interest knowledge-based groups beyond the own domain and organization. Those communities are Knowledge Spheres, which define virtual boundaries around a group of members who can create, modify, and share Knowledge Processes in the ACTIVE Knowledge Community.

A key issue for the ACTIVE platform adoption and success is the ability to handle security and privacy and automate these protocols for the use of Knowledge Processes by Knowledge Workers in every-day work. The purpose of this deliverable is to consider the possibilities and implications of security and privacy issues in the ACTIVE project.

[End of abstract]

Disclaimer

This document contains material, which is the copyright of certain ACTIVE consortium parties, and may not be reproduced or copied without permission.

All ACTIVE consortium parties have agreed to full publication of this document.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the ACTIVE consortium as a whole, nor a certain party of the ACTIVE consortium warrant that the information contained in this document is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.

[Full project title] ACTIVE – Enabling the Knowledge Powered Enterprise

[Short project title] ACTIVE

[Number and title of work-package] WP3 – Pro-active Knowledge Processes Support

[Document title] Security in Knowledge Processes

[Editor: Name, company] Carlos Ruiz Moreno, iSOCO

[Work-package leader: Name, company] Marcel Tilly, EMIC

[Estimation of PM spent on the Deliverable] 4 PM

Copyright notice

© 2009 Participants in project ACTIVE

Executive summary

Nowadays, sharing and collaboration are established on a global scale implying many administrative domains and organizations which exchange sensitive and valuable data. This is a crucial issue because of its importance in order to increase collaboration and productivity for knowledge workers, and therefore, for enhancing the value of a corporation. Also, due to organizational knowledge must keep secret and confidential to competitors.

In this deliverable we have studied the different security factors related with Knowledge Process within the ACTIVE project. We have used the term ACTIVE Knowledge Community to define the social structure based on shared knowledge composed of Knowledge Spheres, virtual boundaries around a group of members or organizations, and tied by different security mechanism. We have to take into account that an ACTIVE Knowledge Community is composed by a set of activities and resources dispersed geographically and institutionally, so security mechanisms are really important in order to allow organizations to control who access to what under which circumstances. Also, organizations must agree on how to interoperate. Thus, such a framework needs to be powerful, flexible, semantically rich and simple to be able to simplify and automate as much as possible.

As part of this deliverable, we have also studied the security and sharing mechanisms for our use cases. Although we have established a general scenario where resources and knowledge are distributed around a broad number of different administrative domains and organizations, the features needed by the use cases are much more limited: users belong to the same organization where they are authenticated formally (ID and password), group permissions are created at design time, resources are stored in a central repository, and there is no negotiation. However, they all use security policies to manage access-control and privacy.

Finally, we have presented a general security framework combining the use of grid technologies, semantic policies and trust management as basis for an ACTIVE security framework. In this sense, Grid had been used in the past to create virtual organizations in order to support the sharing and coordinated use of resources in dynamic and distributed systems. We follow this same point to create Knowledge Spheres with the combination of semantic annotations to establish relationship between them. Inside such a security model, some aspects can be addressed by using semantic security policies. Using security policies can help to automate the behaviour of ACTIVE, and including semantic in security policies brings a lot of interesting properties: simplicity, readability, analyzability, scalability, etc. At last, we have presented how a Semantic Binding from the Ontogrid project can help to combine metadata and security factors to suggest, or even automate, security relationships among Knowledge Spheres.

List of authors

Company	Author
iSOCO	Carlos Ruiz Moreno (cruiz@isoco.com)
iSOCO	José Manuel Gómez-Pérez (jmgomez@isoco.com)
iSOCO	Jesús Contreras (jcontreras@isoco.com)

Table of Contents

Executive summary	3
List of authors.....	4
Table of Contents	5
List of figures	7
List of tables	8
Definitions.....	9
1 Introduction	10
2 Motivation and scope	11
1.1. Knowledge Spheres: ACTIVE Knowledge Community	12
1.2. Security in Knowledge Processes within ACTIVE	13
3 Requirements for the ACTIVE security framework from use cases	15
3.1 Introduction.....	15
3.1.1 A general overview of the use case regarding security issues	15
3.1.2 A detailed description of particular security requirements for use case	15
3.2 Requirements for the BT use case.....	19
3.2.1 A general overview of the use case regarding security issues	19
3.2.2 A detailed description of particular security requirements	19
3.3 Requirements for Accenture use case	20
3.3.1 A general overview of the use case regarding security issues: Accenture Use Case 1.....	20
3.3.2 A detailed description of particular security requirements: Accenture Use Case 1.....	20
3.3.3 A general overview of the use case regarding security issues: Accenture Use Case 2.....	20
3.3.4 A detailed description of particular security requirements: Accenture Use Case 2.....	21
3.4 Requirements for Cadence use case.....	21
3.4.1 A general overview of the use case regarding security issues	21
3.4.2 A detailed description of particular security requirements	21
4 Related work	23
4.1 Security in Grid.....	23
4.1.1 Grid Security Infrastructure (GSI).....	23
4.1.2 OGSA (Open Grid Service Architecture) Security Infrastructure	23
4.2 Security policies.....	23
4.3 Trust management.....	24
4.3.1 Policy-based trust management	24
4.3.2 Reputation-based trust management.....	24
4.4 Security and trust management policy languages and frameworks	25
4.4.1 KAoS	25
4.4.2 Ponder.....	25
4.4.3 Rei.....	26
4.4.4 Protune.....	26
4.4.5 Cassandra.....	26
4.4.6 XACML.....	26
4.4.7 PeerTrust.....	26
4.4.8 Summary comparison	27
5 Security for Knowledge Processes in ACTIVE	29
5.1 Security framework overview	30
5.2 Security into the ACTIVE Knowledge Workspace	32
6 Conclusions	34
References	36
Annex A Security requirements questionnaire for use case partners	38
A.1 Introduction.....	38
A.1.1 A general overview of the use case regarding security issues	38
A.1.2 A detailed description of particular security requirements	38
A.2 User case: <<Name>>	39
A.2.1 A general overview of the use case regarding security issues	39
A.2.2 A detailed description of particular security requirements	39

List of figures

Figure 1 Levels of privacy and confidentiality in ACTIVE.....	11
Figure 2 Communication among domains and organizations in ACTIVE.....	12
Figure 3 A Reputation-based model [6]	25
Figure 4 The ACTIVE Knowledge Community	29
Figure 5 Security approach for ACTIVE	31

List of tables

Table 1 General security goals	13
Table 2 Requirements description for use cases.....	16
Table 3 Requirements description for BT use case	19
Table 4 Requirements description for Accenture use case 1	20
Table 5 Requirements description for Accenture use case 2	21
Table 6 Requirements description for Cadence use case.....	22
Table 7 Security policy approaches classified by the kind of evaluation	27
Table 8 Security policies according formalism and language used.....	27
Table 9 Security policies cross-comparison	28

Definitions

Access Control [24] limits the use of resources. Access may be allowed to specific people, programs or devices, and these are often explicitly permitted to access or otherwise use a particular resource. Hence, access control typically includes the specification of a particular access type (e.g. an agent may be permitted to read a file but not to amend it).

Authentication [24] is a process where a person, program, device or any other agent proves their identity to access a particular resource. The identity may be a simple assertion, usually in a form of user name, login ID or similarly. Authentication is based upon the notion of proof – i.e. to authenticate itself an agent usually provides a proof, which is generally known to the system (such as e.g. a password or a unique token).

Authorization [24] is the actual act of granting (or permitting) an agent, program or device to make use of resources in a secured environment. The act of authorization is often tightly linked to the act of authentication – an agent usually has to authenticate itself first, and this assigns it a particular (often pre-determined) access right. The assigned rights represent the authority to take a specific action or do something with the resource. The authorization or access permissions are often determined by one of the following ways: policies, agent's identity, agent's roles in the organization, organizational capabilities, or any combinations of these.

Credentials [24] are a set of information that the agent presents in order to establish its identity to the access control system. This is a general term, and in practice is used for a wide range of credentials.

Context in ACTIVE ¹ is a set information objects which may be mapped by more than one context. The information objects may be static (e.g. an email or a document) or dynamic (e.g. a task).

Identity [24] Sometimes also digital identity comprises an identity assertion and the characteristics (called attributes) that can be collected and observed in the access control system. Probably the most typical kind of identity is the combination of user name (login) and password. Nonetheless, many other types may be in use (see also authentication).

Security policies are used to dynamically control and protect any system posing constraints on security and privacy. More details regarding this issue are described in Section 4.2 and Section 4.4.

Knowledge Process in ACTIVE² is a loosely defined and structural ramified collection of tasks. The structure of such a process and the order of tasks are not fully defined at the start of a Knowledge Processes. Many tasks require a decision by an actor about the follow-up task. At such a decision point the actor uses his (tacit) knowledge and the current working context to decide for the successor action. To complicate matters, as circumstances change, the actor may decide to work in a different context, rather than follow the normally expected pattern.

Knowledge Sphere is a virtual boundary around a group of members, domains and/or organizations where Knowledge Processes are shared. A Knowledge Sphere can be a couple of ACTIVE users, or communities of users linking different groups or organizations. ACTIVE Knowledge Community is a social structure based on knowledge made of Knowledge Spheres that are tied by different security mechanisms.

Trust [24] is usually defined as a relationship between a trustor and a trustee. The trustor is the subject that trusts the target entity, whereas the trustee is the entity which is trusted. More details on trust and its differentiation from access rights are described in Section 4.3.

Trust Management [3] is the approach to specifying and interpreting security policies, credentials, and relationships which allow direct authorization of security-critical actions.

Virtual organizations [25] is a term from Grid computing defined as a set of individuals and/or institutions sharing resources and services under a set of rules and policies governing the extent and conditions for that sharing.

¹ http://wiki.active-project.eu/wiki/Knowledge_process

² <http://wiki.active-project.eu/wiki/Context>

1 Introduction

The importance of Knowledge Processes management is fundamental for enhancing the value of a corporation. The ACTIVE project is thus concerned with the support of informal Knowledge Processes in terms of a loosely defined and structural ramified collection of tasks and resources. A key issue for the ACTIVE platform adoption and success is the ability to handle security and privacy and automate these protocols for the use of Knowledge Processes by knowledge workers in every-day work.

The typical ACTIVE scenario is collaborative, decentralized and heterogeneous, where knowledge is defined, used and shared across same and different groups, domains, and organizations. Articulating enterprise knowledge in form of Knowledge Processes allows users to create social relationships forming working and interest knowledge-based groups beyond the own domain and organization. On the one hand, knowledge workers contact their social network when they perform their Knowledge Processes, creating social relationships in terms of users who are involved in those Knowledge Processes. Such a virtual boundary linking different users and groups through the sharing of knowledge is here called Knowledge Sphere. On the other hand, Knowledge Processes can be interested for other users, not directly involved (e.g. as a good practices or guideline for a particular process), so they can be shared in a semi-open or open environment, linking and mapping different Knowledge Spheres in an ACTIVE Knowledge Community. In those cases, the use of some kind of trustworthiness and reputation for collaboration and sharing knowledge might be taken into account. Therefore, an ACTIVE Knowledge Community can be seen as a social structure based on knowledge made of Knowledge Spheres that are tied by different security mechanisms.

The possibilities and implications of security and confidentiality issues in Knowledge Processes and how knowledge is shared across domains and companies can affect an ACTIVE Knowledge Community. Those virtual communities may have an own security policies, which must be merged at the organizations level, or even further, at the community level. Under this situation, organizations need to control who access to what and under which circumstances in, as much as possible, flexible, semantically rich, automatic and simple enough fashion.

The purpose of this deliverable is to consider the possibilities and implications of security and privacy issues in the ACTIVE project. This document covers the next points:

- General security and privacy aspects and terminology related to ACTIVE, in terms of procedures for managing and controlling access and sharing of Knowledge Processes, contexts and resources, and introduce related vocabulary such as policy, trust, and reputation.
- The possibilities and implications of security and confidentiality issues in Knowledge Processes and how knowledge sharing cross boundaries or different domains can affect the ACTIVE social networks in terms of trust models and management (e.g. coalition formation, teamwork facilitation, cross-organizational agreements, negotiation ...).
- How to operate without constant human supervision, based on intelligent security behaviour according to some runtime situation (e.g. based on the context).

The document is structured in the following way:

- **Chapter 1** provides introductory explanations for the deliverable.
- **Chapter 2** addresses the motivation and scope regarding security in Knowledge Processes within ACTIVE.
- **Chapter 3** offers the description and evaluation of general and use case requirements;
- **Chapter 4** outlines precious security efforts in Grid systems, and regarding security and trust policies and frameworks.
- **Chapter 5** describes a security model and its main components.
- **Chapter 6** provides the conclusions.

2 Motivation and scope

The aim of ACTIVE is to convert tacit and unshared expertise and knowledge into transferable, interoperable and actionable knowledge in enterprises through the definition, refinement, transferring, sharing, and use of Knowledge Processes. This is fundamental as the successful knowledge management marks the competitiveness in organizations and builds their intellectual capital [19].

In that context, knowledge creation, sharing, and transfer is critically important because of collaboration and exchange of knowledge is the essence of everyday work of knowledge workers such as managers, researchers, engineers or consultants and because successful knowledge work relies more and more on effective and efficient communication, sharing and collaboration on a domain and global scale. This definitively means many administrative domains and organizations establishing relationships and exchanging sensitive and valuable data that can be disclosure internally or to some allied organizations, but kept secret and confidential to competitors. In this multilateral networked vision, privacy and security are a key issue for protecting Knowledge Processes by appropriate mechanisms [7].

The possibilities and implications of security and confidentiality issues in Knowledge Processes and how knowledge is shared across boundaries or different administrative domains can affect the ACTIVE social networks in terms of coalition formation, teamwork facilitation, cross-organizational agreements, etc.

On the one hand, users can establish different disclosure levels for their data. For example, Figure 1 shows how knowledge is disclosed beyond user desktop: data can be private and use for personal purposes, but, and most important, data can be available to a limited group, or public to community users. Knowledge is therefore disclosed at different levels: in an every day work, a regular user interacts with several kinds of users who belong to different domains and organizations at different levels of privacy and confidentiality.

On the other hand, different domains and organizations share part of their data, and must agree on a data disclosure and a security policy. For each one, they internally define their own intra-domain access control and authentication mechanism which must able to interoperate with other intra-domain approaches. In other words, organizations want to control who access to what and under which circumstances.

Moreover, the nature of a Knowledge Process is not atomic, but composed by a set of activities, tasks and resources (including files, and emails, but also all kinds of communication and collaboration tools including IM or wikis) located across the same or even different types of organization, dispersed geographically and institutionally.

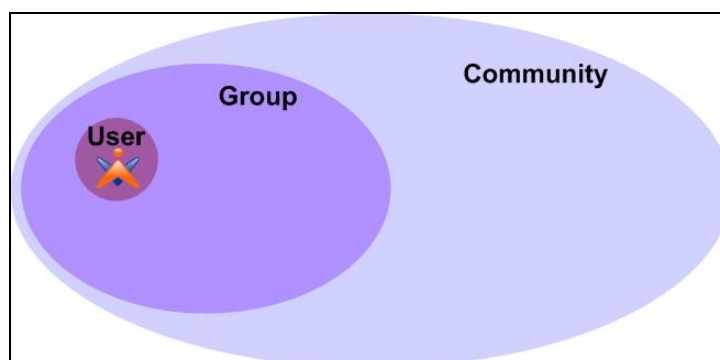


Figure 1 Levels of privacy and confidentiality in ACTIVE.

The size and complexity of the aforementioned scenario where Knowledge Processes are shared among different organizations raise a number of security and confidentiality issues which can not be handled with simple and manual security mechanisms such as plain authorization lists. In complex organizations, different departments may have different data disclosure and security policies. For example, user identification and access to resources may depend on the enterprise identification system (e.g. LDAP). However, such a global approach cannot be used for decentralized and distributed systems where knowledge can be disclosed between different domains, departments or companies, or even share in open communities where may exist different security mechanisms with different level of security. Besides, having a decentralized and open environment implies to establish negotiation mechanisms and trust relationships among interacting users, specifying and interpreting security policies, credentials, and relationships which allow direct/indirect

authorization of accessing and sharing. In addition, sharing Knowledge Processes influences how relationships are established among users and organizations. Generally speaking, social networks allow users to create social relationship forming working and interest groups among different domains and organizations. In the case of ACTIVE, knowledge workers contact their social network when they perform their processes creating social relationships in terms of users who are involved. But also, there are organizational environments, or even inter-organizational environments, where users can share their Knowledge Processes for the benefit of other users (e.g. sharing good practices for a particular process) who are not initially in their contact list or involved in the Knowledge Process.

In those circumstances, users and organizations are linked in virtual boundaries termed as Knowledge Sphere. A Knowledge Sphere is a basic node (can be a couple of users, a complete domain or an organization) of interaction in the ACTIVE ubiquitous environment where Knowledge Processes are shared. All the Knowledge Spheres, with different security and confidentiality policies, form an ACTIVE Knowledge Community tied by those different security mechanisms. Besides the typical security issues, e. g. such as authentication, there are some other operations like publish and subscribe, joining and leaving Knowledge Spheres and so on which must be supported.

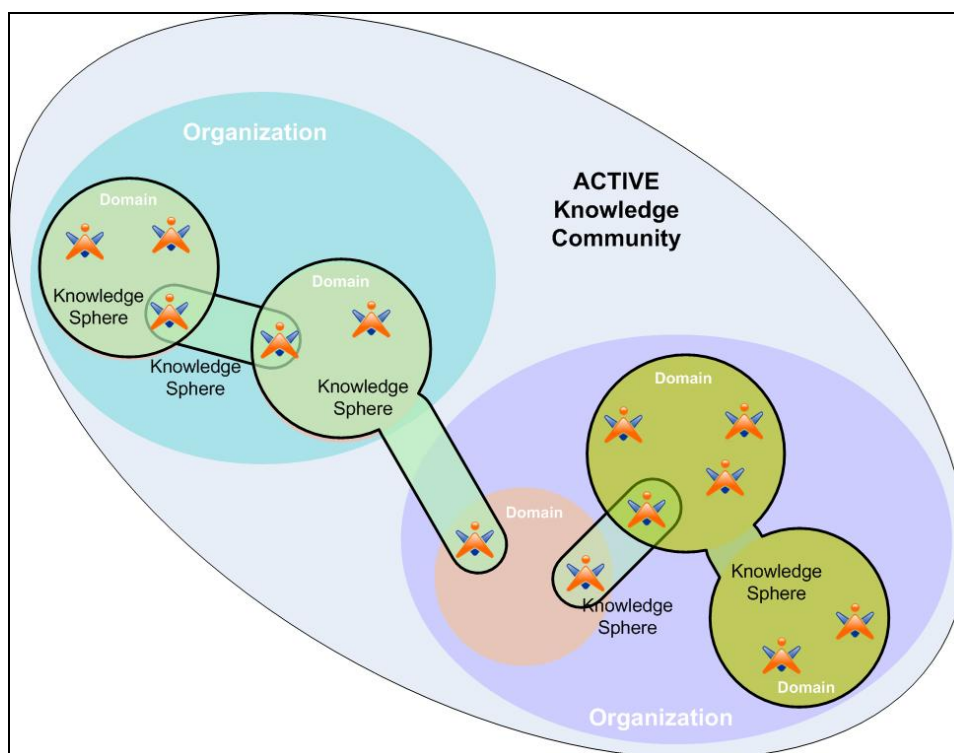


Figure 2 Communication among domains and organizations in ACTIVE

1.1. Knowledge Spheres: ACTIVE Knowledge Community

An ACTIVE Knowledge Community can be seen as a social structure based on shared knowledge made of Knowledge Spheres that are tied by different security mechanisms. A Knowledge Sphere is a node defining a virtual boundary around a group of individuals or organizations that can create, modify, and share Knowledge Processes forming virtual communities. Such an ACTIVE Knowledge Community works as a social network based on knowledge, where particular rights and permissions by users and organizations determinate the social capital of each Knowledge Sphere, and where resulting structures are often very complex. Figure 2 depicts an example of how domains and companies can share knowledge: In those complex systems, again, the security interest of all system stakeholders must be considered, interest conflicts must be identified, and methods for negotiating these conflicts must be proposed [7].

In [9], four kinds of privacy are distinguished in similar environments in order to understand the security risks by emerging technologies from a sociological point of view [8]:

- Freedom from intrusion: This means that we have the right to be left alone, or in ACTIVE, to respect our sphere of autonomy.
- Construction of the public/private divide: This distinction concerns the social negotiation of what remains private and what becomes public.
- Separation of identities: It gives individuals the right to control, edit, manage and delete information about them.
- Protection from surveillance: This refers to the creation and managing of social knowledge about population groups.

According to this, Knowledge Spheres can have a wide variety of security mechanisms that guide interactions regarding resource management, security and access control. When a user joins a Knowledge Sphere or a Knowledge Sphere attempts to join another, there are certain privileges which guide the interaction, and must be resolved in order to come to an agreement. This process of decentralized policy resolution is termed negotiation. Such a negotiation is usually based on the user’s trust and the resource’s reputation, measured from local experiences together with the feedback given by the others. Similar procedures of trust management are typically used to collect user properties in open environments, where the set of potential users spans over the entire web. Moreover, trust management in virtual communities makes security transparent to end users so that the system must have an intelligent behaviour according to the runtime situation at hand (e.g. based on the context).

In general, there are some security requirements which should be satisfied in any open-networked environment [10]. These security goals are referred in the following Table 1:

Table 1 General security goals

Confidentiality goals	<p><u>Confidentiality</u>: information is accessible only to those authorized to have access.</p> <p><u>Anonymity</u>: A user can use services or resources without publishing the real identity.</p> <p><u>Pseudonymity</u>: A user can use services or resources but is still accountable for her actions</p> <p><u>Unobservability</u>: A user can use a service or resources without others being able to observe that the service or resource is being used. The communication process is private but the partners involved in the communication.</p>
Integrity goals	<p><u>Integrity</u>: Any modifications made to the content of communication can be recognised by the communication partners</p> <p><u>Accountability</u>: Communication partners can prove the existence of a communication to a third party</p>
Availability goals	<p><u>Availability</u>: Communicated messages or the resources are available when the user wants to use them</p> <p><u>Reachability</u>: An entity (i.e., a user, machine etc.) either can or cannot be contacted depending on user interests</p>

In addition to the above goals, we should point out that there is a lack of security expertise in users, so that we cannot expect them to be aware of complex security issues applied by their systems, or how those issues can be changed and improved. As a consequence, security facilities for regular users must be based on simplicity too [2].

1.2. Security in Knowledge Processes within ACTIVE

In this context, the ACTIVE Knowledge Community needs a security framework to secure Knowledge Processes and manage access and sharing across a huge number of resources in different domains and

organizations. In other words, it requires some security functions such as authentication, access control, integrity, privacy, and non repudiation.

In a more precisely manner, there are three elements that we consider for security within ACTIVE: virtual organizations from grid systems, semantic security policies from distributed systems, and trust and reputation management from open environments.

Firstly, and as explained before, the typical ACTIVE scenario is decentralized and heterogeneous, where knowledge is created, used and shared across boundaries or different administrative domains within Knowledge Spheres. This scenario has been partially addressed in the past within grid approach using virtual organization [25] as a means to offer a transparent approach for enabling advanced distributed applications to manage and access different types of resources from multiple administrative domains, with complex communication structures and security requirements [20][22].

Secondly, and taking into account that nature of ACTIVE environment and the interaction among Knowledge Spheres for creating and sharing knowledge, a security framework needs to be flexible, semantically rich and simple enough to automate as much as possible. In this sense, policies have been used in a broad sense [7], although the most common type are security policies which are used to pose constraints on system's behaviour and to dynamically control and automate the behaviour of complex environments without requiring code changing or the global cooperation of all system components [1][2].

Thirdly, knowledge is shared across different organizations (or even in open environments), where the parties involved have to establish some kind of trustworthiness for collaboration and sharing knowledge [12]. Though we are still far from achieving such a goal, the importance of trust management has focused special attention in modern open and decentralized systems in relation with the Semantic Web and Web 2.0 applications. In those systems, identification of users is quite flexible, and the only way to determine whether a user can be trust is based on their previous interactions. This fact can affect how Knowledge Processes are disclosure in an open environment [18].

In this deliverable, we investigate the use of these three elements: virtual organizations, security policies, and trust management in the context of the ACTIVE project.

3 Requirements for the ACTIVE security framework from use cases

3.1 Introduction

As part of this deliverable, we have identified the security requirements for each of the use cases (BT, Accenture, and Cadence) in order to define the basic features that a security framework for Knowledge Processes in ACTIVE should provide. Since the environment of our use cases has previously defined their own organizational security features, the main task of this analysis is to identify those requirements and the new desired features.

For this purpose, we have developed a security questionnaire which has been provided to the use cases and filled by them. The questionnaire partially follows the “Common Criteria for Information Technology Security Evaluation”³, an international standard for identifying and defining the security requirements within the questionnaire. It is divided into two parts: a general overview of the use case regarding security issues, and a detailed description of particular security requirements. Following subsections explain each part in detail.

3.1.1 A general overview of the use case regarding security issues

The goal of this step is to describe the use case identifying major security factors. So as to define the security context of the use case environment, each system is briefly presented, giving a short description and pointing out the main security features (e.g. relevant organizational security policies, assumptions, and threats). Also, it can be identified some other desired security properties. This is the basis to build the second step.

3.1.2 A detailed description of particular security requirements for use case

From the general overview provided in the previous section, we derive security requirements for each use case. Thus, each use case evaluates its environment against more specific individual functions using a basic security functionality list based on the “Common Criteria for Information Technology Security Evaluation” (e.g. how a user acting a particular role might be authenticated). So, we have developed a questionnaire with a list of possible answers for the list of security functionality.

Annex A shows the questionnaire provided to use case partners whom have selected the most appropriate ones

The current list of security functionalities are grouped as follows:

1. **Authentication.** This is a process where a person, program or any other agent verifies their identity. The identity may be a simple assertion, usually in a form of user name, login ID or similarly with the combination of a password. Questions related to this point are: How do users identify in the use case? Does the use case support anonymity?
2. **Security Management.** Many systems require some sort of management (e.g., to control who can do what), generally by those who are given a more trusted role (e.g., administrator). Questions related to this point are: Are there different roles for the same user? How are groups and domains discovered and configured? Is there some kind of trust management in the use case?
3. **User Data Protection.** This feature specifies requirements for protecting user data (access control or information flow rules), and develop various means to support off-line storage, import, and export, and provide integrity when transferring data between users. Questions related to this point are: How user resources are made public? How data protection is established?
4. **Resource Utilization.** This point refers to the availability of required resources such as sharing mechanisms, processing capability and storage capacity. Questions related to this point are: Where are resources allocated? Is there some kind of protection against run-time unavailability of a resource?

³ http://en.wikipedia.org/wiki/Common_Criteria

For each of these functionalities we have defined a set of questions with different possible answers:

Table 2 Requirements description for use cases

Type	ID	Requirement	Description
Authentication	A-SR1	Type of identification	Indicates how users are identified by the system. At this point in time, we have identified: <ul style="list-style-type: none"> • No identification. • Informal identification (e.g. a simple register using an email address). • Formal identification in a domain, but not the whole organization (e.g. identified in a department, but not in the whole organization) • Formal identification in an organization.
	A-SR2	Support anonymity.	Indicates whether the use case supports anonymity. We have identified the following answers: <ul style="list-style-type: none"> • Yes. • No. • Other (please indicate).
	A-SR3	Support unobservability	Indicates whether user cannot determine whether a resource is being accessed. We have identified the following answers: <ul style="list-style-type: none"> • Yes. • No. • Other (please indicate)
Security Management	SM-SR1	Support roles	Indicates whether the use case support user roles. We have identified the following answers: <ul style="list-style-type: none"> • Yes. • No. • Other (please indicate).
	SM-SR2	Groups and domains discovery	Indicates how groups and domains are discovered and configured. We have identified the following answers: <ul style="list-style-type: none"> • At design time (e.g. a user has an account in the domain or group). • At run-time (e.g. there is a sort of negotiation). • Other (please indicate)
	SM-SR3	Collaboration between groups and/or domains	Indicates how the relationship and collaboration is between those groups or domains. We have identified the following answers: <ul style="list-style-type: none"> • Collaboration among groups in the same

			<p>organization.</p> <ul style="list-style-type: none"> • Collaboration among groups in different organization • Collaboration among different organizations. • Other (please indicate)
	SM-SR4	Negotiation support	<p>Indicates whether there is a sort of negotiation, and how this negotiation is managed. We have identified the following answers:</p> <ul style="list-style-type: none"> • Using security policies. • Using a reasoning mechanism. • Using a trust measure. • Other (please indicate)
	SM-SR5	Trust support	<p>Indicates whether there is some kind of trust management in the use case. We have identified the following answers:</p> <ul style="list-style-type: none"> • No. • Yes, it is associated to a particular role. • Yes, it is associated to a trust measure. • Other (please indicate).
User Data Protection	UDP-SR1	Resources availability mechanism	<p>Indicates how user resources are made public. We have identified the following answers:</p> <ul style="list-style-type: none"> • Manually (e.g. uploading resources) • Automatic (e.g. sharing folders in p2p systems) • Other (please indicate)
	UDP-SR2	Privacy of user data	<p>Indicates how data protection is established. We have identified the following answers:</p> <ul style="list-style-type: none"> • Manually • Using security policies. • Allowing access to a particular domain or group. • Other (please indicate)
	UDP-SR3	Permissions with an organization and/or domain	<p>Indicates how users access to resources associated to a domain or an organization. We have identified the following answers:</p> <ul style="list-style-type: none"> • Manual (e.g. permission given by an administrator) • Automatic (e.g. permission given with the identification in the system) • Other (please indicate)

Resource Utilization	R-SR1	Resources emplacement	<p>Indicates where resources allocated are allocated. We have identified the following answers:</p> <ul style="list-style-type: none"> • User's desktop. • Central repository. • Some distributed repositories. • Other (please indicate)
	R-SR2	Run-time availability	<p>Indicates whether there is some kind of protection against run-time unavailability of a resource. We have identified the following answers:</p> <ul style="list-style-type: none"> • Yes. • No. • Other (please indicate).
	R-SR3	Access to third-part resource	<p>Indicates when a user accesses to a third-part resource. We have identified the following answers:</p> <ul style="list-style-type: none"> • A copy of the resource is downloaded to the user's desktop. • A copy of the resource is created. • The resource is blocked, and the user modifies the original resource. • Other (please indicate).
	R-SR4	Resource management	<p>Indicates how new versions of resources are managed. We have identified the following answers:</p> <ul style="list-style-type: none"> • Using versions in a resource repository • Manual renaming of the file • Other (please indicate)
	R-SR5	Resource recommendation	<p>Indicates whether there is some kind of recommendation for related resources. We have identified the following answers:</p> <ul style="list-style-type: none"> • Yes. • No. • Other (please indicate).
	R-SR6	Resource recommendation mechanisms	<p>Indicates how recommendation is performed. We have identified the following answers:</p> <ul style="list-style-type: none"> • Based on users in the same domain. • Based on users' role. • Based on a trust measure.

3.2 Requirements for the BT use case

3.2.1 A general overview of the use case regarding security issues

This use case describes the creation of a customer proposal. It involved one or more proposal writers going through a number of steps leading to the creation of a document. Steps include writing the document; looking for relevant information, e.g. from previously written proposals; using a costing tool to determine costs and hence pricing; liaising with various experts; checking consistency of contributions from different people; etc.

In our discussions with trialists no explicit requirements for security have arisen. All the people involved in the process are working within the BT firewall. They may be working from home, or some third party premises. However, they will be using a secure BT VPN to access BT resources.

3.2.2 A detailed description of particular security requirements

Table 3 offers detail information regarding security requirements from BT use case.

Table 3 Requirements description for BT use case

Type	ID	Requirement	Description
Authentic ation	A-SR1	Type of identification	Formal identification in the organization
	A-SR2	Support anonymity.	No.
	A-SR3	Support unobservability	No.
Security Management	SM-SR1	Support roles	No.
	SM-SR2	Groups and domains discovery	At run-time
	SM-SR3	Collaboration between groups and/or domains	Currently case: collaboration among groups in the same organization. <i>Currently this is the case</i> Collaboration among different organizations. <i>This might be the case in the future.</i>
	SM-SR4	Negotiation support	Not applicable – there is no negotiation.
	SM-SR5	Trust support	No.
User Data Protection	UDP-SR1	Resources availability	Manually (e.g. uploading resources)
	UDP-SR2	Privacy of user data	Using security policies. <i>User profiles are not shared.</i>
	UDP-SR3	Permissions with an organization and/or domain	Automatic (e.g. permission given with the identification in the system)
Resource Utilization	R-SR1	Resources emplacement	Central repository.
	R-SR2	Run-time availability mechanism	No.
	R-SR3	Access to third-part resource	A copy of the resource is downloaded to the user's desktop.
	R-SR4	Resource management	Manual renaming of the file.
	R-SR5	Resource recommendation	Yes.
	R-SR6	Resource recommendation mechanisms	Based on content and metadata.

3.3 Requirements for Accenture use case

3.3.1 A general overview of the use case regarding security issues: Accenture Use Case 1

The use case is that of an enterprise search and browsing portal which supports a wide variety of tasks. The user logs in to the system using their *userid* that is unique for each user in the enterprise.

3.3.2 A detailed description of particular security requirements: Accenture Use Case 1

Table 4 offers detail information regarding security requirements from Accenture Use Case 1.

Table 4 Requirements description for Accenture use case 1

Type	ID	Requirement	Description
Authentication	A-SR1	Type of identification	Formal identification in the organization
	A-SR2	Support anonymity.	No.
	A-SR3	Support unobservability	Yes
Security Management	SM-SR1	Support roles	No
	SM-SR2	Groups and domains discovery	At design time (e.g. a user has an account in the domain or group).
	SM-SR3	Collaboration between groups and/or domains	Collaboration among groups in the same organization
	SM-SR4	Negotiation support	Using security policies.
	SM-SR5	Trust support	No
User Data Protection	UDP-SR1	Resources availability	Manually (e.g. uploading resources)
	UDP-SR2	Privacy of user data	Manually
	UDP-SR3	Permissions with an organization and/or domain	Automatic (e.g. permission given with the identification in the system)
Resource Utilization	R-SR1	Resources emplacement	Central repository. Some distributed repositories.
	R-SR2	Run-time availability mechanism	No.
	R-SR3	Access to third-part resource	A copy of the resource is downloaded to the user's desktop.
	R-SR4	Resource management	Manual renaming of the file.
	R-SR5	Resource recommendation	No.
	R-SR6	Resource recommendation mechanisms	Not applicable

3.3.3 A general overview of the use case regarding security issues: Accenture Use Case 2

The use case is that of a enterprise search and browsing portal which supports a wide variety of tasks. The user logs in to the system using their *userid* that is unique for each user in the enterprise.

3.3.4 A detailed description of particular security requirements: Accenture Use Case 2

Table 5 offers detail information regarding security requirements from Accenture Use Case 2.

Table 5 Requirements description for Accenture use case 2

Type	ID	Requirement	Description
Authentication	A-SR1	Type of identification	Formal identification in the organization.
	A-SR2	Support anonymity.	No
	A-SR3	Support unobservability	Yes
Security Management	SM-SR1	Support roles	No
	SM-SR2	Groups and domains discovery	At design time (e.g. a user has an account in the domain or group).
	SM-SR3	Collaboration between groups and/or domains	Collaboration among groups in the same organization.
	SM-SR4	Negotiation support	Using security policies.
	SM-SR5	Trust support	No
User Data Protection	UDP-SR1	Resources availability	Manually (e.g. uploading resources)
	UDP-SR2	Privacy of user data	Manually
	UDP-SR3	Permissions with an organization and/or domain	Automatic (e.g. permission given with the identification in the system)
Resource Utilization	R-SR1	Resources emplacement	Central repository. Some distributed repositories.
	R-SR2	Run-time availability mechanism	No
	R-SR3	Access to third-part resource	A copy of the resource is downloaded to the user's desktop.
	R-SR4	Resource management	Manual renaming of the file.
	R-SR5	Resource recommendation	No
	R-SR6	Resource recommendation mechanisms	Not applicable

3.4 Requirements for Cadence use case

3.4.1 A general overview of the use case regarding security issues

The use case is the data acquisition for ProjectNavigator, a typical application scenario in microelectronic design, setting the focus for the early phase of prototyping on digital back end design and on the verification part of digital design flow.

In general there are designers providing information which should not be revealed to other users (project/program managers & designers).

3.4.2 A detailed description of particular security requirements

Table 6 Requirements description for Cadence use case

Type	ID	Requirement	Description
Authentication	A-SR1	Type of identification	Formal identification in the organization.
	A-SR2	Support anonymity.	No.
	A-SR3	Support unobservability	Yes.
Security Management	SM-SR1	Support roles	Yes.
	SM-SR2	Groups and domains discovery	At design time (e.g. a user has an account in the domain or group).
	SM-SR3	Collaboration between groups and/or domains	Collaboration among groups in the same organization.
	SM-SR4	Negotiation support	No
	SM-SR5	Trust support	Yes, it is associated to a particular role.
User Data Protection	UDP-SR1	Resources availability	Never
	UDP-SR2	Privacy of user data	Using security policies
	UDP-SR3	Permissions with an organization and/or domain	Automatic (e.g. permission given with the identification in the system)
Resource Utilization	R-SR1	Resources emplacement	Central repository.
	R-SR2	Run-time availability mechanism	Not yet decided
	R-SR3	Access to third-part resource	The resource is blocked, and the user modifies the original resource.
	R-SR4	Resource management	Using versions in a resource repository.
	R-SR5	Resource recommendation	Not yet decided
	R-SR6	Resource recommendation mechanisms	Not applicable

4 Related work

This section outlines the main previous related work regarding the main three topics highlighted in the motivation section.

Firstly, we describe how security is managed in Grid systems. Secondly, we explain how security policies have been used in the past to control permissions and accessing resources in distributed, grid, and open environments. Thirdly, we describe different security and trust management languages and frameworks.

4.1 Security in Grid

A computational grid [21], a large-scale distributed computing environment, is a collection of heterogeneous computers and resources across multiple administrative domains with the goal of providing easy access to these resources [12]. In concrete, grid applications are distinguished by use of resources from multiple administrative domains, use a large number of resources, complex communications structures, etc. As any other distributed environment, this approach raises some security requirements that need to be satisfied.

The following subsections summaries the different approaches for providing security in grid environments.

4.1.1 Grid Security Infrastructure (GSI)

The Grid Security Infrastructure (GSI) was initially described in [20] for achieving grid security goals (e.g. management of access to resources from multiple administrative domains interconnected). It is the facto security standard in the grid community [12], which provides basic security properties such as sign-on, delegation and authentication. GSI is built on security standards such as X.509 certificate data structures, SSL protocol and the Generic Security Service API (GSS-API) [12].

4.1.2 OGSA (Open Grid Service Architecture) Security Infrastructure

The Open Grid Service Architecture proposes a security architecture and a set of security components that encapsulate the required security functionalities in grid systems [23]. This approach is gaining popularity among the scientific as well as the industrial grid communities [12].

4.2 Security policies

In the past, the term policy has been used in the literature in a very broad sense referring from security policies or trust management to business rules [7]. In general, we can split them into two main approaches: On the one hand, security policies relying on strong security mechanisms based on user identity and authentication e.g. trusted certification authorities. On the other hand, trust management relying on procedures for establishing and maintaining trust relationship among users in a large open system where anyone can contribute and access in somehow.

Policies are used to dynamically control and automate the behaviour of complex environments without requiring code changing or the global cooperation of all system components. In [3] pointed out the main benefits of this approach: reusability, efficiency, extensibility, context-sensitivity, verifiability, support for both simple and sophisticated components, and protection from poorly designed components, and reasoning about component behavior. Moreover, they are used in the literature in a broad sense either that encompasses [3] [11]:

- Access control and privacy policies protect any system open to the Internet and assist users while they browse and interact with different resources and services. In general, Security Policies pose constraints on the behaviour of a system (e.g. used to control permissions of users and groups while accessing resources).
- Network administration policies often applied to automate and govern network administration tasks, such as configuration, security, recovery, or quality of service.
- Trust Management policy languages are used to collect user properties in open environments, where the set of potential users spans over the entire web.

- Action Languages are used in reactive policy specification to execute actions like event logging, notifications etc.
- Business Rules are statements about how a business is done and are used to formalize and automate business decisions as well as for efficiency reasons.

In general, policy-based systems relies on strong security mechanisms such as signed certificates and trusted certification authorities (CAs) in order to regulate access control permissions, monitoring, and other actions to be taken.

In the past, policies have been specified in many different ways and multiple approaches have been proposed. In the context of this deliverable, we concentrate on three well-known web languages for policy representation and reasoning, KAOs, Rei, and Ponder, and provide a summary on the main features (a more complete description is described in [3]).

4.3 Trust management

The importance of trust management has focused special attention in modern open and decentralized systems in relation with the Semantic Web and Web 2.0 applications as a large and open system where anyone can contribute and access. However, this implies a number of drawbacks in terms of trustworthiness of the users accessing to resources and systems. The goal of trust management is to establish and maintain relationships among users and systems in open environments.

A definition is given in [13]: “the activity of collecting, encoding, analyzing and presenting evidence relating to competence, honesty, security or dependability with the purpose of making assessments and decisions regarding trust relationships”.

There are two approaches used for trust management: policy-based and reputation-based trust management [13].

4.3.1 Policy-based trust management

This approach has been proposed in the context of open and distributed services architectures as well as in the context of Grids systems as a solution to the problem of authorization and access control in open systems. The focus here is on trust management mechanisms employing different policy languages and engines for specifying and reasoning on rules for trust establishment. The goal is to determine whether or not a certain priori unknown user can be trusted, based on a set of credentials and a set of policies.

4.3.2 Reputation-based trust management

Reputation-based management [3] relies on an approach to the problem of trust in open environments. In this case, trust is typically computed from local experiences together with the feedback given by other users in the network. In Figure a general model reputation-based model is outlined [6].

For example, in the case of eBay, buyers and sellers rate each other after each transaction, reflected in a global trust measure by the eBay community. The reputation-based approach has been fundamental for open environments such as Peer-to-Peer (P2P) or Semantic Web, where the existence of certifying authorities can not always be assumed but where a large pool of individual user ratings is often available [4].

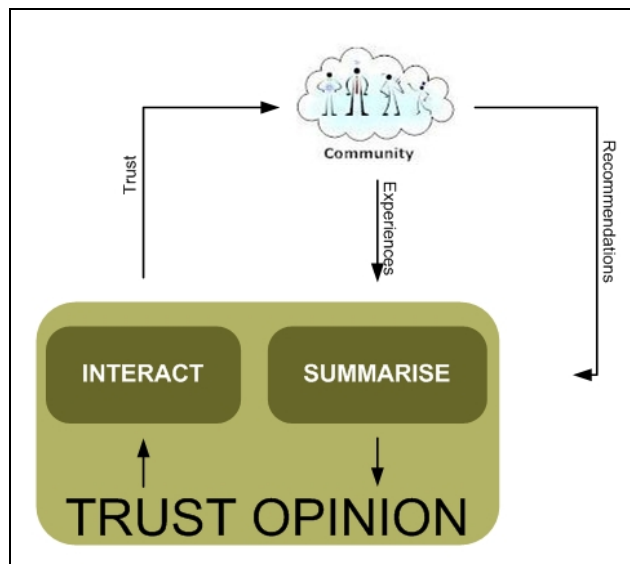


Figure 3 A Reputation-based model [6]

4.4 Security and trust management policy languages and frameworks

There exist a number and variety of policy languages justified by the different requirements they have to accomplish [13] [14]. This section makes a summary and a comparison among some of the languages and frameworks proposed in the past for this research field.

4.4.1 KAoS

KAoS [15] [16] is a set of independent-platform policy services originally oriented to dynamic and complex software agents applications, and later adapted to grid computing and Web Services. KAoS allows the creation, management, and conflict resolution for policies, providing the capability for groups of people, resources and any other entity to be structured into domains within organizations [1]. KAoS poses a set of very interesting features:

- Policies are represented as ontologies in OWL [16] in the KAoS Policy Ontologies (KPO): it distinguishes between authorization and obligation for an action to be performed. Other policy actions (for example, role-based authorization) are built from the basic domain primitives plus the policy types.
- The KAoS framework allows the use of additional domain ontologies to express related concepts and actions.
- The KAoS framework supports dynamic runtime policy change and update due to a logical inference engine which resolve policy selection and conflict and runtime.
- The KAoS framework defines ontology-based mechanisms to load new platforms and applications. Moreover, it adapts a wide range of well-known computing and software environments such as Web Services, grid computing, and CORBA, or Cougar, Nomads.
- The KAoS framework implements a sophisticated graphical tool called KAoS Policy Administration Tool (KPAT) that facilitates security designers to focus on high-level policies specification, visualization and monitoring.

4.4.2 Ponder

Ponder⁴ [26] is a declarative and object-oriented policy language for distributed systems developed at Imperial College. It covers concepts like domains, roles in a organizations, and relationships to groups the object to which policies apply. It also supports obligation policies in form of condition-action rules,

⁴ <http://www-dse.doc.ic.ac.uk/Research/policies/ponder.shtml>

registration of users and audio event for security violations. A complete toolkit has been developed to support the users of the language (a grammar compiler, a policy editor, and a management toolkit)

4.4.3 Rei

Rei⁵ [27] is a policy language based in OWL-Lite developed by L. Kagal concerned with support pervasive computing applications. In Rei, policies can be specified as constraints over allowable and obligated actions on resources. It also includes meta policy specifications for conflict resolution and policy analysis specifications like what-if analysis and use-case management. It offers an reasoning engine to provide answers about the current permissions and obligations of an entity in order to guide its behaviour.

4.4.4 Protune

Protune⁶ (PROvisional TrUst NEgotiation) provides a framework focusing on trust management combining distributed trust negotiation and credentials for access-control related actions. For this purpose, it features a declarative metalanguage for supporting critical negotiation decisions and actions based on attributes (e.g. *allow(download(resource) ← public(Resource))*), and integrity constraints for monitoring negotiations and credential disclosure [4] [7].

Protune also provides an explanation facility for policies and negotiations in form of three different queries: how-to, why/why-not, and what-if [18]. Such queries are very important to understand which pieces of information are actually used or needed during a negotiation, or just to monitor some access control or credential.

4.4.5 Cassandra

Cassandra [28] is a role-based policy framework for large-scale distributed systems. It defines different kind of predicates for expressing a wide range of policies including role hierarchy, role delegation, separation of duties, automatic credential discovery and trust negotiation. It uses a goal-oriented distributed policy evaluation algorithm.

4.4.6 XACML

XACML⁷ [17], standing for eXtensible Access Control Markup Language, is an XML-based standard for policy representation developed by OASIS consortium (<http://www.oasis-open.org/home/index.php>). This approach specifies schemas for authorization policies and for authorization decision requests and responses, assuming the following summarised usage model:

- A Policy Enforcement Point (PEP) is responsible for protecting some resources.
- When a resource access is attempted, the PEP sends a description of the attempted access to a Policy Decision Point (PDP) in form of an authorization request. This request is evaluated against its available policies and produces an authorization decision.
- The PEP is responsible for enforcing the decision. Also, it may obtain other request related attributes from on-line Attribute Authorities or from Attribute Repositories.

Furthermore, XACML defines distributed policies but evaluation is centralized. Semantics are defined in the functional-language Haskell. The current version of XACML is V3.0 Working Draft 08 (3 November 2008).

4.4.7 PeerTrust

PeerTrust⁸ [29] was a language focused on automated trust negotiation on the Semantic Web based on a distributed query evaluation [14]. It develops a trust mechanism which has the ability to reason about

⁵ <http://rei.umbc.edu/>

⁶ <http://policy.l3s.uni-hannover.de:9080/policyFramework/protune/>

⁷ http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

⁸ <http://www.l3s.de/peertrust/>

statements made by others peers, measuring the trustworthiness of other peers and perform trusted interactions based on their past interaction histories without trusted third parties. Its language is based on first order Horn rules ($\text{literal}_0 \leftarrow \text{literal}_1, \dots, \text{literal}_n$), built upon the rule layer of the Semantic Web layer cake (e.g. RuleML).

4.4.8 Summary comparison

There are some comparisons of the different languages showed in the former section attending the different possible features of the languages.

In [17], Bonatti and Olmedilla classify those approaches with regards to the kind of evaluation they perform (e.g. centralized evaluation) and whether a policy is well-defined (e.g. independent of implementation). Some other authors classify those approaches with regards to the different formalisms used to specify the security policy. Both classifications are showed in the following Table 7 and Table 8.

Table 7 Security policy approaches classified by the kind of evaluation

	Centralized evaluation	Distributed policies, Centralized evaluation	Distributed evaluation
Well-defined semantics	RBAC	KaOS, Rei	PeerTrust, Protune, Cassandra
No formal semantics	ACL, Java Policies	Ponder, XACML, P3P	TPL

Table 8 Security policies according formalism and language used

Formalisms used	Language/Framework
XML	XACML
OWL	KAoS
RDF/RDF Schema	Rei
Object-Oriented Language	Ponder

However, the most complete evaluation so far is performed by De Coi and Olmedilla in [14] where the next properties are taken into account to evaluate the different approaches:

- Well-defined semantics: the semantic within a security language is considered well-defined if it is independent of the particular implementation.
- Underlying formalism: the security language is based on a well-known formalisms (e.g. description logics)
- Delegation: the security policy supports delegation, the assignment of authority to an agent or component to carry out specific activities on its behalf of (e.g. passing rights to open a file).
- Type of evaluation: this property defines how evaluation is performed during negotiation (e.g. distributed policy evaluation).
- Evidences: the policy evaluation needs some information in order to establish its identity. Such information is usually called credentials.
- Negotiation: the security framework supports negotiation between peers (e.g. to define a policy or evaluate a request).
- Extensibility: the security language supports extensibility in order to adapt the language to new needs. The following Table 8 gathers and summaries the related information based on [14]:

Table 9 Security policies cross-comparison

	KAoS	Ponder	Rei	Protune	Cassandra	XACML	PeerTrust
Well-defined semantics	Yes	No	Yes	Yes	Yes	No	Yes
Underlying formalism	Description logics	Object-oriented paradigm	Description Logics, Logic programming	Logic programming	Constraint DATALOG	-	Constraint DATALOG
Delegation	No	Yes	Yes	Yes	Yes	No	Yes
Type of evaluation	Local	Local	Distributed policies, local evaluation	Distributed	Distributed policies, local evaluation	Distributed policies, local evaluation	Distributed
Evidences	No	-	-	Credentials, Declarations	Credentials	No	Credentials, Declarations
Negotiation	No	No	No	Yes	Yes	No	Yes
Result format	Allow/Deny	Allow/Deny	Allow/Deny	Explanations	Allow/Deny ad a set of constraints	Allow/Deny, not applicable, indeterminate	Allow/Deny
Extensibility	Yes	Yes	Yes	Yes	Yes	Yes	Yes

5 Security for Knowledge Processes in ACTIVE

The creation and management of the ACTIVE Knowledge Community poses a number of security challenges. Users from multiple domains and organizations share their diverse resources and Knowledge Processes in Knowledge Spheres creating an overall ACTIVE Knowledge Community while they belong to different real organizations governed by their own internal security rules, policies, and mechanisms. Besides, this sharing must be highly controlled because of the trade of confidential knowledge, defining clearly what is shared, who is allowed to share, and the conditions under a sharing occurs. In Figure 4, we depicted this situation: in an organization, different set of users share their Knowledge Processes creating internal Knowledge Spheres which are part of the organizational ACTIVE Knowledge Community. At the same time, diverse Organizational Knowledge Communities can be part of a global ACTIVE Knowledge Community sharing resources and Knowledge Processes according some own security policies and protocols. As part of the ACTIVE Knowledge Community, some other organizations can be assigned or not, but in any case, we have to provide security and confidentiality mechanisms to the others.

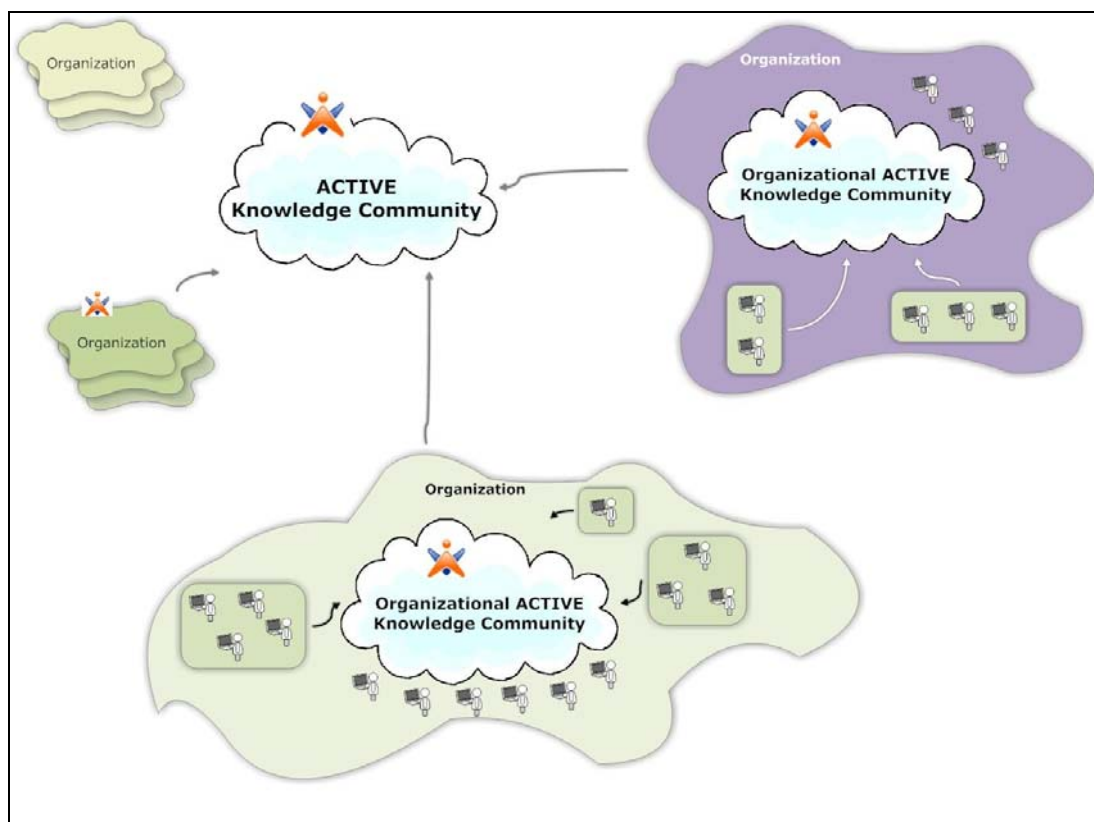


Figure 4 The ACTIVE Knowledge Community

The situation described above can be addressed with the use of a comprehensive Grid security architecture. Firstly, because it supports and integrates different and popular security models, protocols, and technologies in a way that enables a variety of systems interoperate securely. Also, one of its basic requirements is that security mechanisms have to be pluggable and discoverable, and allow the federation of security mechanisms. Secondly, because the idea of virtual organizations within Grid infrastructure perfectly fits with the problems faced by ACTIVE Knowledge Community and follows the same goal and purpose: support the sharing and coordinated use of diverse resources in dynamic and distributed systems. In other words, a virtual organization from Grid computing can be used as a basic infrastructure for the ACTIVE Knowledge Community in order to virtualize the access and security mechanism.

Moreover, using a Grid security model allows us to satisfy general security requirements: interoperability with local security solutions, plug point for authentication mechanisms, facilities for delegation of access rights, a single logon for authentication over a reasonable period of time, facilities for defining and enforcing privacy policies, control access based on authorization policies (e.g. who can access a resource under what

conditions) or specify invocation policies (e.g. what you trust to provide the requested resource), exchange particular information for establishing dynamic and negotiated security environments (e.g. such information can contain authentication requirements or supported functionalities), message integrity and secure logging.

The security requirements described in Section 3 are satisfied inside such a grid security model [20] [22]. Although more information of each point is given in the following section, an overview of how those general security requirements are satisfied is given:

1. Aspects like access-control to resources and authentication can be addressed by using security policies [1] in order to pose constraints on system's behaviour and to dynamically control and automate the behaviour of complex environments without requiring code changing or the global cooperation of all system components (e.g. using authorization policies indicating who can access a resource under what conditions attached to each resource).
2. Aspects like security management between different administrative domains is performed through the exchanging dynamically security information (e.g. policies, credentials) to establish a negotiated security context between them.
3. Aspects like identity in different domains are addressed by the notion of trust relationships and federation between security mechanisms.

5.1 Security framework overview

In order to define a security framework within ACTICE, we have followed the security principles of Grid computing [20] taking into account the different aspects and factor involved and described in the previous sections. Thus, we have followed the following assumptions:

1. A trust domain is an administrative structure within a single, consistent security mechanism.
2. A Knowledge Sphere is a virtual boundary around a group of trust domains sharing knowledge in form of resources and Knowledge Processes. Members of a Knowledge Sphere can be from the same or different real administrative domains or organizations integrated in the same virtual organization.
3. An ACTIVE Knowledge Community is a set of Knowledge Spheres, and maps each Knowledge Sphere into the global virtual view of the community
4. Operations confined to a trust domain are only subject to local security mechanisms, not imposing any grid security method.
5. Operations between different administrative domains or organizations require mutual authentication using some kind of credentials

Moreover, it is reasonable that a number of operations can occur in a Knowledge Sphere: a user log on into a Knowledge Sphere mapping him from a local to a global subject; a user share resources and Knowledge Processes into the Knowledge Sphere under some security policy; a user log out of a Knowledge Sphere; even, a Knowledge Sphere can join another Knowledge Sphere offering to its users accessed to the first one resources according its internal security mechanisms; A Knowledge Sphere offers access to resources according its own security policy. ACTIVE platform may automate the belonging to a Knowledge Sphere based on other security interactions and annotations. These operations are explained in more detail:

- Knowledge Spheres may grow and shrink according operations of logging on and out. These operations can be manual or dynamic depending on the nature of the required credentials (e.g. a login/password or a private key) and how the Knowledge Sphere can obtain these credentials. Also, these can be restricted by a time duration or provide a temporary credential based on a public key infrastructure, or so. Thus, the use of credentials is quite simple and useful, since involved entities simply need to check the other one's credentials (e.g. for validating they belong to the same group, guaranteeing the identity of others, negotiate a session key).
- As part of the log on and log out operations, an important element of the security policy is a valid mapping between a local subject and the corresponding global one. Although the local identity may be established by a login or an ID, the global identification can be given by some kind of ticket or certificate. In order to achieve this conversion, a mapping table must be maintained by Knowledge Spheres.

- A user shares some resources into the Knowledge Sphere according some security policy and allocation and identity is mapped globally by the Knowledge Sphere. One of these resources is accessed by somebody else. In this case, the operation is controlled by the user’s local policy within Knowledge Sphere which is responsible for mapping the access to the resource. The Knowledge Sphere determines the identity of the user requiring access, and whether the request is successful, the access to the resource. The request can fail because: there is an authentication failure, the user is not a recognized user and doesn’t prove their identity to access to the resource; there is an authorization failure, the user is identified but has no access right into the resource; there is a allocation failure, the identification and authorization were right but the resource is not available at runtime.
- Similar considerations are taken when Knowledge Spheres are involved (e.g. a Knowledge Sphere wants to join into a Knowledge Sphere).
- An ACTIVE Knowledge Community may suggest, or even automate, the belonging to a Knowledge Sphere based on other security interactions and annotations. For example, some users could share some files annotated with the name of a project, which is related with some topics according to an ontology. Although some other users are not part of such a Knowledge Sphere, those topics can be relevant for their interest according to their metadata, and based on the security policies they would be allowed to participate in that Knowledge Sphere.

Figure 5 depicts the main layers and components of the security framework which can be used to construct the particular security architecture: a local layer with local security polities and protocols, a second layer based on a comprehensive Grid Security infrastructure, and finally the ACTIVE Knowledge Community.

Firstly, an ACTIVE local layer describes all the resources and Knowledge Processes shared by an user. These are described by ontologies which users can annotate using also domain ontologies (e.g. roles in an organization). This annotation process generates metadata in order to be later related with different Knowledge Spheres through Semantic Bindings (explained further on). As explained, the ACTIVE platform could use those annotations in order to suggest or automate the belonging to different Knowledge Spheres based on the metadata relationships.

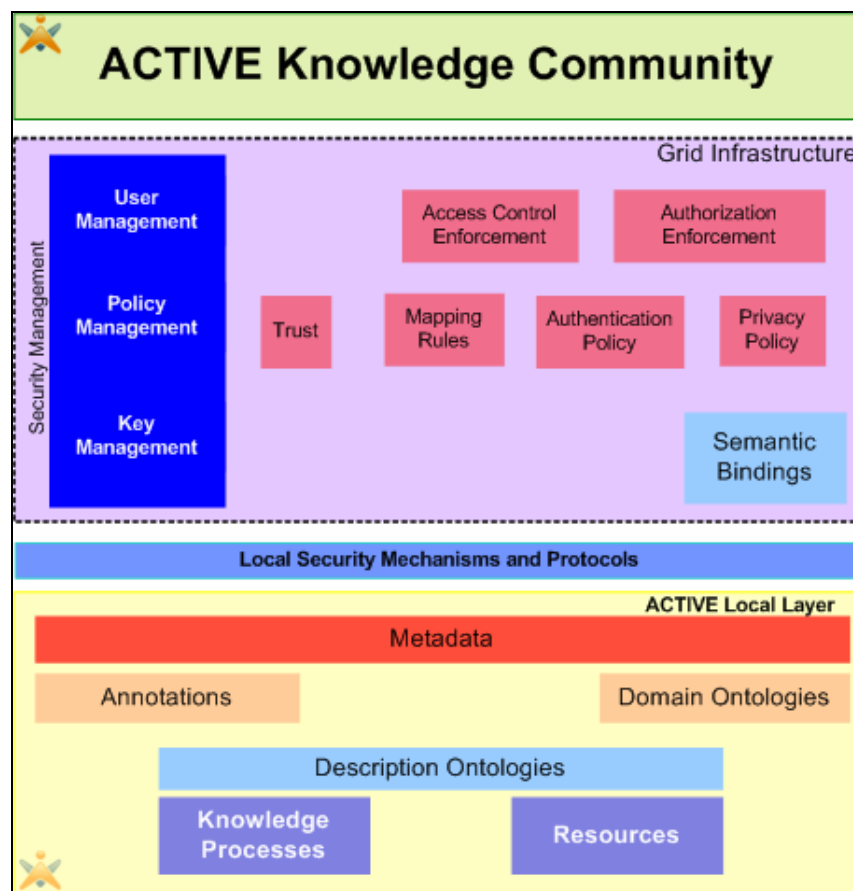


Figure 5 Security approach for ACTIVE

Secondly, we can find the Grid security layer as proposed in [21]. We have just specified some of the components which encapsulate the main functionalities for our purposes:

- Security management component groups all security management functions applicable to policy and federation. These include key and user management, authorization, privacy and trust policy management and the mapping needed for federation.
- Access-control enforcement component prevents unauthorized users for accessing or modifying resources and Knowledge Process for which they have no rights.
- Authorization enforcement component validates that each domain satisfies its own authorization method in order to control the access to resources. Sometimes privacy policies have been treated as part of the authorization policies. The Grid Security Framework defines a mechanism to express, expose and exchange policies through a component called policy expression and exchange.
- Trust management is needed for different administrative domains and/or organizations. Security policies and authentication credentials belonging to the same domain are manageable within the scope of the organization. However, trust management for traverse organizations are fundamental for sharing and accessing resources from one domain to another. Such a trust mechanisms have to define mutual trust relationships between different domains and/or organizations mapping identities and credentials among the involved domains and organizations.

Finally, the ACTIVE Community is at the top level which can be treated as a global Knowledge Sphere.

Regarding the use of security policies, some considerations are of our interest: there have been different approaches in the past with different advantages and disadvantages, but none can be used as a common approach for all situations. In any case, the choice of a particular security policy should be driven by the following features [1]: simplicity, readability, analyzability, scalability, and flexibility in order to automate and reuse them as much as possible. Among the different alternatives presented in Section 4.4, the adoption of ontologies for policy representation seems clear because of a series of advantages: represent elements at multiple level of abstraction, help to model policies at a high level of abstraction avoiding implementation details, allowing users to use concepts to describe entities and environment simplifying the description and facilitating the analysis and empowering the reasoning. On the disadvantages side, an ontology-based policy specification can be difficult to implement.

Apart of the common Grid model, we have included a Semantic Binding component from the Ontogrid project [23]: it contains explicit metadata about one or several Grid Entities (e.g. resources) and relates that metadata to one or several Knowledge Entities (e.g. ontologies). In this case, we describe resources and Knowledge Processes with description ontologies, and allow users to generate annotation with some domain ontologies. Then, we can use the elements provided by the Semantic Binding to query explicit metadata, also combined with content, in order to suggest, or even automate, the belonging to a Knowledge Sphere based on other security interactions and annotations. For example, some users could share some resources and Knowledge Processes annotated with some tags like “template”, “Schedule a meeting”, “BT”, “sales group”, “sales assistant”, which are related with some topics like “role”, “company”, “group” or “type or resource” according to some domain ontologies. Although some other users are not part of such a Knowledge Sphere, those topics can be relevant for their interest according to their metadata, and based on the security policies they would be allowed to participate in that Knowledge Sphere.

5.2 Security into the ACTIVE Knowledge Workspace

The use of services for the ACTIVE security framework achieves a level of abstraction that helps to provide an integrated security environment. Besides, existing security technologies had been exposed as services, and may be reused (e.g. WS-Policy⁹ or any from WS-Security) within ACTIVE. Like any other service, security services should be described as Web Services and should expose their functionalities while hiding implementation details.

In this sense, the use of security services is also considered in the design principles followed by the ACTIVE Knowledge Workspace (fully described in D5.1.2 ACTIVE Knowledge Workspace architecture and design).

⁹ <http://www.w3.org/Submission/WS-Policy/>

In a nutshell, the ACTIVE Knowledge Workspace is structured as a set of cooperative software services with well defined interfaces, and where those services support Enterprise Knowledge Structures and Knowledge Processes as executed in an enterprise by knowledge workers. The proposed architecture is flexible and expandable to support extensions and changes. In such an architecture, the use of security services is originally described, and a generic access control and an user/group services are defined in the Workspace Infrastructure Service. Therefore, security services assure seamless integration and interoperability with the ACTIVE Knowledge Workspace.

The ACTIVE security framework may use different security function in form of services for the basic security components described in Section 5.1. These security services should include:

- An authentication service concerned with verifying the identity of users and organizations to Access to a particular resource. agent proves their identity to access a particular resource
- A policy service concerned with the management of security access policies followed by organizations (and virtual organizations).
- A federation service concerned with the management of virtual organizations Because different organizations involved may use different security technologies, it may be necessary to mediate and translate the different security mechanisms between the parties involved. Also, an identity mapping service transforming local and global identities from different domains is needed.
- A privacy service concerned with the management of personal information among different users and organizations. This service should combine with other services like the policy and trust services.
- A trust service concerned with the management of trust relationships among users and organizations.

In recent years, there have been some efforts to standardise the use of web services for security. The WS-Security is a standard released by the Open OASIS organization providing a means for applying security to Web services¹⁰. It provides a flexible and extensible unified model that uses existing technologies (e.g. SOAP or WSDL) and that allows applications to exchange secure communications. WS-Security defines draft specifications for the majority of the security requirements (WS-Authorization, WS-Policy, WS-Trust, WS-Policy, WS-Federation, WS-Trust ...)

¹⁰ <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>

6 Conclusions

Nowadays, sharing and collaboration are established on a global scale implying many administrative domains and organizations which exchange sensitive and valuable data. Security mechanisms are really important because organizations want to control who access to what under which circumstances, so in order to interoperate they must agree on how security mechanisms are established. Moreover, such a framework needs to be flexible, semantically rich and simple enough to simplify and automate as much as possible.

In this deliverable we have studied the different security factors related with Knowledge Process within the ACTIVE project: sharing knowledge is very important and crucial to increase collaboration and productivity Knowledge Workers, and therefore, for enhancing the value of a corporation. Also, organizational knowledge must be kept secret and confidential to competitors.

We have used the term Knowledge Sphere to define a virtual boundary around a group of members, domains or organizations where Knowledge Processes are shared within ACTIVE Knowledge Community. We have to take into account that a Knowledge Process might compose by a set of activities and resources dispersed geographically and institutionally.

As part of this deliverable, we have also studied the security and sharing mechanisms for our use cases. Although we have established a general scenario where resources and knowledge are distributed around a broad number of different administrative domains and organizations, the features needed by the use cases are much more limited: users belong to the same organization where they authenticate formally (ID and password), group permissions are created at design time, resources are stored in a central repository, and there is no negotiation. However, they all use security policies to manage access-control and privacy.

We have presented a general security framework combining the use of grid technologies, semantic policies and trust management as basis for an ACTIVE security framework. In this sense, Grid had been used in the past to create virtual organizations in order to support the sharing and coordinated use of resources in dynamic and distributed systems. We follow this same point to create Knowledge Spheres with the combination of semantic annotations to establish relationship between them. Inside such a security model, some aspects can be addressed by using semantic security policies. Using security policies can help to automate the behaviour of ACTIVE, and including semantic in security policies brings a lot of interesting properties: simplicity, readability, analyzability, scalability, etc. At last, we have presented how a Semantic Binding from the Ontogrid project can help to combine metadata and security factors to suggest, or even automate, security relationships among Knowledge Spheres.

References

- [1] Tonti, G.; Bradshaw, J. M.; Jeffers, R.; Montanari, R.; Suri, N. & Uszok, A. Fensel, D.; Sycara, K. P. & Mylopoulos, J. (ed.) Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei, and Ponder. *International Semantic Web Conference, Springer, 2003*, 2870, 419-43
- [2] Dokoohaki, N. Semantic Web Services: Role of Security, Authorization, Privacy and Trust in Semantic Web *Proceeding of the OASIS Adoption Forum, 2006*.
- [3] P.A. Bonatti, C. Duma, N. Fuchs, W. Nejdl, D. Olmedilla, J. Peer, and N. Shahmehri, "Semantic Web Policies - A Discussion of Requirements and Research Issues". *3rd European Semanti Web Conference, LNCS 4011*, pp. 712-724, Budva, Montenegro, June 2006
- [4] P.A. Bonatti, C. Duma, D. Olmedilla, N. Shahmehri, An Integration of Reputation-based and Policy-based Trust Management, *Semantic Web and Policy Workshop (in conjunction with 4th International Semantic Web Conference)*, Galway, Ireland, November 2005.
- [5] NEOON Project, D4.4.1 The role of access rights in ontology customization.
- [6] Alfarez Abdul-Rahman , Stephen Hailes, Supporting Trust in Virtual Communities, *Proceedings of the 33rd Hawaii International Conference on System Sciences, Volume 6, 2000*
- [7] Piero A. Bonatti, Daniel Olmedilla: Driving and Monitoring Provisional Trust Negotiation with Metapolicies. *Proceedings of the 6th International Workshop on Policies for Distributed Systems and Networks (POLICY 2005)*, Stockholm, Sweden, 2005.
- [8] S. F. Gürses, B. Berendt, and Thomas Santen. Multilateral Security Requirements Analysis for Preserving Privacy in Ubiquitous Environments. *Ubiquitous Knowledge Discovery for users (UKDU'06)*. Berlin, Germany, 2006.
- [9] D.J. Phillips: Privacy Policy and PETs: The Influence of Policy Regimes on the Development and Social Implications of Privacy Enhancing Technologies. *New Media & Society* 6(6): 691 – 706.
- [10] G. Wolf and A. Pfitzmann. Properties of protection goals and their integration into a user interface. *Computer Networks*, 32:688-699, 2000
- [11] Bonatti, P., Olmedilla, D. Driving and monitoring provisional trust negotiation with metapolicies. *Sixth IEEE International Workshop on Policies for Distributed Systems and Networks*, pp 14 – 23, 2005.
- [12] Li, T., Zhu, H., and Lam, K. A Novel Two-Level Trust Model for Grid. *Lectures Notes in Computer Science, Information and Communication Security*, pp. 214-225, Volume 2836/2003, 2003.
- [13] REWERE Project, I2-D1 Rule-based Policy Specification: State of the Art and Future Work.
- [14] De Coi, J. L., and Olmedilla, D. A review of trust management, security and privacy policy languages. *SECURITY2008: International Conference on Security and Cryptography*, Porto, Portugal, 2008.
- [15] Uszok, A., Bradshaw, J. M., Jeffers, R., Suri, N., Hayes, P. J., Breedy, M. R., Bunch, L., Johson, M., Kulkarni, S., and Lott, J. KAoS policy and domain services: Towards a description-logic approach to policy representation, deconfliction, and enforcement. In *POLICY 2003*. IEEE Computer Society.
- [16] Uszok, A., Bradshaw, J.M., Johnson, M., Jeffers, R., Tate, A., Dalton, J., Aitken, S. KAoS policy management for semantic Web services. *Intelligen Systems, IEEE*, pp. 32 – 41, 2004.
- [17] Bonatti, P.A., Olmedilla, D. Rule Based Policy Representation & Reasoning for the Semantic Web. *REWERSE Reasoning on the Web Summer School*, Dresde, Germany, 2007.
- [18] REWERE Project, I2-D4 Advanced Policy Queries.
- [19] E. Bertino, L. R.Khan, and R. Sandhu. Secure Knowledge Management: Confidentiality, Trust, and Privacy. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 36, N°. 3, 2006.
- [20] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. A Security Architecture for Computational Grids. *ACM Conference on Computer and Communication Security*, pp. 83 – 92, 1998.

- [21] I. Foster and C. Kesselman. *Computational Grids: The Future of High Performance Distributed Computing*. Morgan Kaufmann, 1998.
- [22] N. Nagaratnam, P. Janson, J. Dayka, A. Nadalin, F. Siebenlist, V. Welch, I. Foster, S. Tuecke. *The Security Architecture for Open Grid Services*. GGF OGSA Security Workgroup.
- [23] O. Corcho, P. Alper, An overview of S-OGSA: a Reference Semantic Grid Architecture. *Journal of Web Semantics* 4(2):102-115. June 2006.
- [24] Security glossary. <http://www.rsasecurity.com/glossary>
- [25] I. Foster, C. Kesselman, S. Tuecke. *The Anatomy of the Grid: Enabling Virtual Organizations*. *International Journal of Supercomputing Applications*, 15(3), 2001.
- [26] N. Damianou, N. Dulay, E. Lupu, and M. Sloman. *The Ponder Policy Specification Language*. In *Proceedings of POLICY 2001: Workshop on Policies for Distributed Systems and Networks*. Lecture Notes on Computer Science, pp. 18-39, 2001.
- [27] L. Kagal, T. W. Finin and A. Joshi. *A Policy Language for Pervasive Computing Environment*. In *Proceedings of POLICY 2003: Workshop on Policies for Distributed Systems and Networks*. Lecture Notes on Computer Science, 2003
- [28] M. Y. Becker and P. Sewell. *Cassandra: Distributed access control polices with tunable expressiveness*. In *Proceedings of POLICY 2004: Workshop on Policies for Distributed Systems and Networks*. IEEE Computer Society, 2004
- [29] R. Gavriloaie, W. Nejdl, D. Olmedilla, K. E. Seamons, M. Winslett. *No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web*. In *Proceedings of ESWS 2004: 1st European Semantic Web Symposium Lecture Notes in Computer Science 2053*, 2004.

Annex A Security requirements questionnaire for use case partners

A.1 Introduction

D3.4.1 is focused on security related issues around Knowledge Processes (e.g. how a Knowledge Process is shared or how users are identified in ACTIVE). As part of D3.4.1, we have to identify the security requirements for each of the use cases (BT, Accenture, and Cadence) in order to define a security framework for Knowledge Processes in ACTIVE. We gently ask ALL use case partners to provide their security requirements using the following template for defining such an ACTIVE Security framework.

For this purpose, we partially follow the “Common Criteria for Information Technology Security Evaluation”¹¹, an international standard for identifying and defining security requirements. Since our use cases environments have previously defined their own organizational security features, the main task of this analysis is to identify them and new desired features. This simplified analysis is divided into two steps:

A.1.1 A general overview of the use case regarding security issues

The goal of this step is to identify the use case system that is the target of this analysis. So as to define the security context of the use case environment, each system will be briefly presented, giving a short description and pointing out the main security features (e.g. relevant organizational security policies, assumptions, and threats). Also, it can be identified some other desired security properties. This will be the basis to build the second step.

A.1.2 A detailed description of particular security requirements

From the general overview of the use case target, you derive security requirements so that they meet the target. Thus, the target is evaluated against more specify individual functions (e.g. how a user acting a particular role might be authenticated).

Initially, we have selected some questions and a list of possible answers for a list of security functionalities. Use case partners have to select the most appropriate ones.

The current list of security functionalities is as follows:

1. **Authentication.** This is a process where a person, program or any other agent verifies their identity. The identity may be a simple assertion, usually in a form of user name, login ID or similarly with the combination of a password. Questions related to this point are: How do users identify in the use case? Does the use case support anonymity?
2. **Security Management.** Many systems require some sort of management (e.g., to control who can do what), generally by those who are given a more trusted role (e.g., administrator). Questions related to this point are: Are there different roles for the same user? How are groups and domains discovered and configured? Is there some kind of trust management in the use case?
3. **User Data Protection.** This feature specifies requirements for protecting user data (access control or information flow rules), and develop various means to support off-line storage, import, and export, and provide integrity when transferring data between users. Questions related to this point are: How user resources are made public? How data protection is established?
4. **Resource Utilization.** This point refers to the availability of required resources such as sharing mechanisms, processing capability and storage capacity. Questions related to this point are: Where are resources allocated? Is there some kind of protection against run-time unavailability of a resource?

¹¹ http://en.wikipedia.org/wiki/Common_Criteria

A.2 User case: <<Name>>**A.2.1 A general overview of the use case regarding security issues**

<<To be completed>>

A.2.2 A detailed description of particular security requirements

<<For the proposed questions, please select the most appropriate one for your use case>>

A.2.2.1 Authentication

This is a process where a person, program or any other agent verifies their identity. The identity may be a simple assertion, usually in a form of user name, login ID or similarly with the combination of a password. Questions related to this point are:

- How do users identify in the use case?

No identification (e.g. public access)

Identification with public registration (e.g. a simple registration process using an email address)

Formal identification in a domain, but not the whole organization (e.g. identified in a department, but not in the whole organization)

Formal identification in the organization

Others (please indicate)

- Does the use case support anonymity?

Yes

No

Other (please indicate)

- If so, how is this anonymity managed?

- Does the use case support unobservability (subjects cannot determine whether a resource is being accessed)?

Yes.

No.

Other (please indicate)

- Please indicate any other information that you consider relevant:

A.2.2.2 Security Management

Many systems require some sort of management (e.g., control who can do what), generally by those who are given a more trusted role (e.g., administrator)

- Are there different roles for the same user?

Yes.
No.
Other (please indicate).

- How are groups and domains discovered and configured?

At design time (e.g. a user has an account in the domain or group).
At run-time (e.g. there is a sort of negotiation).
Other (please indicate)

- How is the relationship and collaboration between those groups or domains?

Collaboration among groups in the same organization
Collaboration among groups in different organization
Collaboration among different organizations
Other (please indicate)

- Whether there is a sort of negotiation, how is this negotiation managed?

Using security policies
Using a reasoning mechanism
Using a trust measure
Other (please indicate)

Some other times there are special operations that only trusted roles can invoke, or even, there is a trust measure to specify and interpret security policies, credential, and relationships which allow direct authorization of security-critical actions between a trustor and a trustee. The trustor is the subject that trusts the target entity, whereas the trustee is the entity which is trusted. This process is usually called trust management.

- Is there some kind of trust management in the use case?

No
Yes, it is associated to a particular role
Yes, it is associated to a trust measure
Other (please indicate).

- Please indicate any other information that you consider relevant:

--

A.2.2.3 User Data Protection

This feature specifies requirements for protecting user data (access control or information flow rules), and develop various means to support off-line storage, import, and export, and provide integrity when transferring data between users.

- How user resources are made public?

Manually (e.g. uploading resources) Automatic (e.g. sharing folders in p2p systems) Other (please indicate)

- How data protection is established?

Manually Using security policies Allowing access to a particular domain or group Other (please indicate)

- How do users access to resources associated to a domain or an organization?

Manual (e.g. permission given by an administrator) Automatic (e.g. permission given with the identification in the system) Other (please indicate)
--

A.2.2.4 Resource Utilization

This point refers to the availability of required resources such as sharing mechanisms, processing capability and storage capacity.

- Where are resources allocated?

User's desktop Central repository Some distributed repositories Other (please indicate)
--

- Is there some kind of protection against run-time unavailability of a resource?

Yes. No. Other (please indicate).

- When a user accesses to a third-part resource:

A copy of the resource is downloaded to the user's desktop.
A copy of the resource is created.
The resource is blocked, and the user modifies the original resource.
Other (please indicate).

- How are new versions of resources managed?

Using versions in a resource repository
Manual renaming of the file
Other (please indicate)

- There is some kind of recommendation for related resources?

Yes
No
Other (please indicate)

- If so, how this recommendation is performed?

It is based on users in the same domain
It is based on users' role
It is based on a trust measure

- Please indicate any other information that you consider relevant:

[end of document]