

# A governance model for SOA

Pierre de Leusse \*/\*\*, Theo Dimitrakos \*\*, David Brossard \*\*

\* Newcastle University, \*\*BT Innovate

Pierre.de-leusse@ncl.ac.uk

## Abstract

Currently, business requirements for rapid operational efficiency, customer responsiveness as well as rapid adaptability are driving the need for ever increasing communication and integration capabilities of the software assets. Service Oriented Architecture (SOA) is generally acknowledged as being a potential solution to expose finely grained pieces of software components on a network that are reusable and composable. Provisioning of business services for different business purposes may require the rapid assembly of their core functionality with different infrastructure capabilities and policies in different contexts. In this paper, the authors propose a SOA based governance model that permits to handle non functional requirements in a dynamic way.

## 1. Introduction

The ever increasing amount of IT services along with all the potential states and types of configurations necessitate the development of adequate methods and tools for IT services governance. In [1], the concept of SOA governance is derived from corporate and IT governances. Corporate governance is referred to as the set of processes, customs, policies, laws and institutions affecting the way in which a corporation is directed, administered or controlled. IT governance is a subset of corporate governance that focuses on the control, performance and risk of IT systems. For SOA, the term governance refers to the processes used to oversee and control the adoption and implementation of a SOA in accordance with recognised policies, audit procedures and management policies. SOA governance aims at providing optimum service quality, consistency, predictability and performance. A SOA governance environment should offer the ability to define, administer and enforce a combination of processes, practices and tools that facilitate the management of the life-cycle of the services in the SOA as well as the life-cycle of the different policies that apply on these services.

## 2. Objectives of SOA governance

Functional decomposition into services, reuse, loose coupling, and distribution of resources are all perceived

benefits of the investment on SOA. This malleability can also bring about the risk of a more difficult oversight. The same service is used in different applications the infrastructure will have to adapt to these different contexts of use in order to provide variations in required functionality, quality of service, billing schemes and security requirements. Achieving such variations in a cost efficient way can be achieved by composing the core business function offered by a service with other services implementing infrastructure capabilities that fulfill varying non-functional requirements.

However, as the number of services increases and their use in different contexts proliferates, it becomes necessary to automate policy enforcement and compliance monitoring. Furthermore, the composition of services into different business applications over a common infrastructure intensifies the need for end-to-end monitoring and analysis to assess the business performance impact. Managing the full life-cycle of service definition, deployment, exposure and operation requires management processes that take into account their composition with the infrastructure capabilities that take of non-functional requirements. Finally, policies may change during the life-time of a service. Policy updates may be the result of various reasons including business optimisation, of reaction to new business opportunities, of risk / threat mitigation, of operational emergencies, etc. It becomes therefore clear that a well designed governance model is a prerequisite to successfully implementing a SOA. More details on the objectives of such a SOA governance framework are given in Table 1.

**Table 1. SOA governance objectives**

<i>Resource contextualisation</i>	Permits resources to be efficiently configured for and managed at an end-to-end level is one of the main objectives of SOA governance. This not only allows to adapt resources to specific transactions in function of an organisation's rules but to manage this adaptation in a more configurable, reliable and secured way.
<i>Resource visibility</i>	Brings the best fit for purpose visibility into the IT infrastructure used and its state. This aims at making sure that not only it is possible to find the resource but also

	<p>that its purpose and constraints are well understood.</p> <p>With complex systems comprised of many resources (e.g. Web service, policy) there is a strong requirement to increase the visibility of each resource. Indeed a same functionality could be provided by different services and advertised in different places.</p> <p>In addition, one of the strength of SOA being reuse and composition, there is an obligation to know how resources communicate and are wired together. The relevant management of dependences amongst resources is indeed a crucial element of the visibility.</p> <p>Furthermore, with a same resource involved in several collaborations or discussions it is necessary to keep track of how this same functionality is proposed (i.e. its attached constraints).</p> <p>Increasing the visibility includes advertising its functionality as well as non functional properties correctly, its issuer or provider.</p>	<p>well as non IT specialists.</p> <p><i>Resource life-cycle management</i></p> <p>Coordinates the lifecycle of policies throughout different stages of the infrastructure they support including its transition from development to operation.</p> <p>In SOA like on a production chain, resources go through different stages before they can be sold and eventually support can be offered on them or they are terminated. The appropriate management of the life-cycle of both the policies and the services they allow to manage is therefore a key element as to how visible, safe and reliable the SOA and its components are going to be.</p>	
<p><i>Policy administration</i></p>	<p>Administers policies coming from different sources of authority and that may apply to different, potentially interrelated, contexts and business collaborations.</p> <p>In a company, the different levels of hierarchy manage their resources according to their responsibilities. As such managers set up rules on how certain requests from clients are going to be dealt with when the directors will set up the roles and limits of the manager's authority. As IT services attempt more and more to support business functions, the same types of policies should be applied to them, allowing for different levels of authority that apply in particular or more general cases.</p> <p>The same apply for the different areas of expertise where an account manager will dictate the pricing policy for a client and the lawyer will know how to write legal contracts. IT services are dependants of the IT specialists at different levels (e.g. deployment, security) as</p>	<p><i>Policy management</i></p> <p>Manages the selection and integration of the best policy decision and policy enforcement mechanisms to support the optimal use of IT resources and services in a given context and in compliance with corporate agreements.</p> <p>As introduced in the three points above, a SOA will suffer from having many services that may be available in different contexts and at different stages of their life-cycle. The management of the SOA is made through the use of policies and as such it is crucial to be able to manage how these are going to be used and enforced.</p> <p>In addition, a SOA governance framework should allow detecting potential conflicts within the imbrications of services and their policies.</p>	
		<p><i>Process management</i></p> <p>Allows processes for different actors, at different stages of the services and policies to act upon them appropriately.</p> <p>The main objective of the process management is to allow different actors to influence the way both policies and services are created, maintained (i.e. from draft to finished version), shared, exposed and reported upon.</p> <p>As introduced above, with the SOA attempting to support a company's functions, the processes are the IT replication of the management processes in the</p>	

	physical world.
<i>Resource adaptation</i>	<p>Enables diagnosis and remediation in an as automated as possible fashion.</p> <p>SOA systems can potentially get very complex, with many different policies and services. Managing changes in the exposure of the different resources used and available is therefore a key aspect.</p>

### 3. Related works

SOA governance has been much talked about over the past few years. Industry middleware actors (e.g. SOA middleware vendors, consultants) have been the biggest sources of both hype and innovation. There has been little effort from academic research in this topic, although several related subjects such as service composition, service management and Service Level Agreement (SLA) have received interest.

The term “SOA governance” has also sometimes been treated as a marketing term for the packaging of the set of features that allow managing and improving the visibility of distributed resources. Such issues are well understood and solutions have been researched and developed for the past 10 years. In fact SOA governance frameworks build on top of such work by addressing the need to make the supporting service management and monitoring layer interoperable and introduce processes that allow governing multiple interrelated services and policies in SOA deployments as one whole.

Currently, there is no SOA governance technology and architecture that fulfils the requirements aforementioned [1]. Instead, vendors have a tendency to aggregate the different products they have developed over the years that supports the management of distributed resources.

ESB vendors, services deployment platforms and Service registry providers (e.g. HP, IBM) include what they define as governance tools in their products. These products such as HP Systinet or IBM WebSphere Service Registry and Repository (WSRR) are mainly providing service registry and metadata (e.g. policy) repository services along with their supporting features. Some of these products also provide some support for service life cycle management and policy as well as service management.

Governance Interoperability Framework (GIF) by HP Systinet is proposed as a specification for SOA Governance. This effort however is mostly inclined towards registry and repository support as underlined above.

These attempts are mainly directed towards the visibility and management objective of SOA governance and offer only little support, if any, for the adaptation and contextualisation.

Another area of interest that has focused on certain aspects of governance and that has received more interest from the research community is the management of Non Functional Properties (NFPs) as a way to improve the adaptation of a resource. Several projects have looked into different ways of defining and expressing NFPs, using either ADL [2], taxonomy [3] or ontology [4]. If some of these projects didn’t target any precise type of IT systems [2], a few were mostly interested in SOA [3]. Using this knowledge, some research was made on how to allow a concrete separation between a resource’s functionality and its NFPs. For instance in [5] a solution is proposed to manage a Web service NFPs using handlers. In [6], an architecture to handle the NFPs of sensors is presented. But none of the above papers proposed a SOA based solution to improve adaptability and increase ease of contextualisation in a more dynamic manner.

### 4. Requirements for a governance model

In this section we describe the requirements of a SOA governance framework. These requirements were gathered through the studying of a large number of business cases studies and pilots in research projects such as TrustCoM [7][8][9] and BEinGRID [10][11] and by working together with academia [15][16][17], customers and SOA vendors such as IBM, Layer 7 Technologies, Vordel, Microsoft and SAP.

To achieve the objective mentioned in Objectives of SOA governance, a flexible framework needs to be provided. As part of this infrastructure, the content of the following elements aim to be adjustable depending on the services provided, as well as the content and context of interactions:

- IT infrastructure profiles, including the selection of core infrastructure capabilities (c.f. following paragraphs on core infrastructure capabilities), and the corresponding policy schemes.
- Policy schemes and templates about protecting managing and monitoring resources, and transformations to realise these into concrete policy instances for specific target environments and contexts.
- Resource management processes that manage the life-cycle of IT resources and IT infrastructure services depending on the target environment and context.
- Governance processes that coordinate infrastructure service management and resource allocation across the enterprise.

The governance framework must allow adaptability in response to changes of the non-functional requirements of the resource exposed through it and also be capable of

adapting to different kind of events such as change in the requirements of the different components it uses. This has an impact on the way the consumed services must be presented, it influences the way the framework is architected and it affects the management of the profile.

## 5. Anatomy of a governance model

The architecture of governance model is summarised in the following paragraphs. A distinction is made between the operational, data and management models.

### 5.1. Governance: Operational model

Following is the list of the core operational elements part of the governance model and their basic properties.

*Business Capability:* This is an organisation traditional function (e.g. accountancy, fleet management, credit check). It is exposed as a service and can be the result of an aggregation of other business capabilities. In order to allow a more automated interaction and configuration of this type of service, it is assumed that it can be managed through a common service management abstraction layer (e.g. WS-DM).

*Infrastructure Capability:* This is a supporting capability fulfilling non-functional requirements such as identity management or access control. A set of infrastructures are typically aggregated to support the exposition a business capability. The non core infrastructure can include all type of non functional requirement providers (e.g. billing, audit). In the following paragraphs core infrastructures that are vital for SOA governance are presented.

*Access control:* An access control infrastructure is used in order to check authorization. It generally consists in a specialised service that check security assertions against access control requests. This is typically achieved through the use of access control policies and security assertions written in specialised grammar such as XACML or SecPAL. An example of a way such infrastructure can be defined provided in [18].

*Identity management:* The role of the identity broker is to allow users to identify themselves. Authentication of the entity that acts as user is indeed a key aspect of SOA where different domains (e.g. companies, branches) will interoperate. This is generally accomplished by using security tokens. An example of this type of infrastructure is further discussed in [18].

*Message interceptor:* The message broker often called mediator acts as an intermediary between two points. It can receive messages from multiple destinations, determine the correct destination and route the message to appropriate channels. An example of this type of infrastructure is described in [19].

*Metadata repository:* Often referred to as a policy store or simply repository, the role of this infrastructure is to allow storing metadata such as policies, taxonomies or ontology. Together with the service registry, this is the most commonly found element in existing governance solution.

*Policy management:* A policy management infrastructure makes sure that policies are written in compliance with organisational rules; these include the use of specific grammars, policy life-cycle management and control their access. Additionally, it ensures that there is no conflict between policies when these are assembled to control a resource.

*Profile management:* The profile manager controls the life-cycle of profile instances. This is done to guarantee that these are defined, enabled, monitored and disabled when relevant in agreement to clients needs.

*Service management:* An infrastructure that helps monitoring (e.g. availability, performance) services according to the policies set to this effect.

*Service registry:* The service registry is a repository where Web services are listed. On production of their credential, users can then discover the services which are potentially organised in different categories.

### 5.2. Governance: Object Model.

Following is the list of the core data elements part of the governance model and their basic properties.

*Infrastructure Profile:* Profiles are descriptors that define which composition of infrastructure capabilities (e.g. security services, audit) to use for the exposition of a business capability. Each profile associates infrastructures with their corresponding policy schemes, dependences (policy and service) and management processes.

*Context:* This is a combination of a potentially shared scope and state. They allow linking a profile to business capabilities, message exchanges or even operations.

**5.2.1. Policies.** Policies are rules describing behavior that a certain capability or process must comply with. They typically comply with different specific standards (e.g. WS-Policy, XACML). The main issues about policy in the governance framework are their life-cycle management; the shared nature of their authoring, enforcement and monitoring; the potential necessity to translate same type of policies from different grammars (e.g. an access control infrastructure could be using either XACML or SecPAL). In the following paragraphs, the main governance policy types of the SOA governance are introduced. This is particularly important for the governance model as the different types of policies can take precedence one over another.

*Profile policies:* Profile policies identify and define policies or template that applies within their domains. The

most important ones will regard the dependences and constraints related to the use of a profile.

*Infrastructure capability policies:* These policies are attached particular infrastructures and consider potential I/O metadata, usage, monitoring and management schemes.

*Business capability policies:* The business capabilities are similar to the infrastructure but for the possibility to assign exposure process schemes to them.

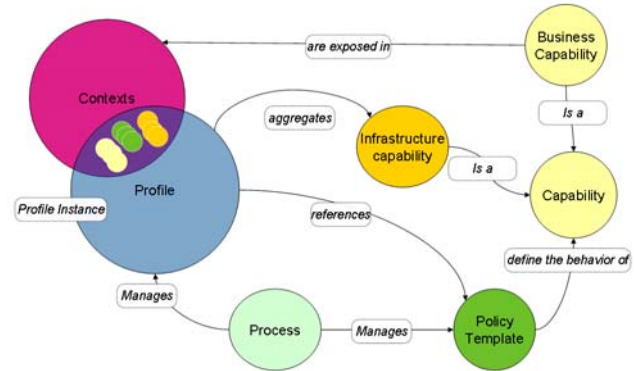
*Compliance policies:* Artifacts needed to identify and to define policies that implement regulatory compliance standards and other industry specific standards (e.g. Health Insurance Portability and Accountability Act). This can also be applied to technological standards for service modeling as well as data structuring. For instance WS-I and relevant versions of WS-A or SOAP can be applied to services; specific ontology, taxonomy or micro format such as the gene ontology or hCard can be applied to data.

*Metric policies:* In most reactive system, metrics need to be set in order to observe the system and allow reacting to specific sets of actions on the system and its components (e.g. service, policy) or changes. This can be used in order to monitor a resource, to control its usage, to modify its billing, etc.

**5.2.2. Processes.** A process is a procedure that uses the above building blocks in order to meet governance objectives. A distinction can be made between governance as well as policy and service management processes. The management processes target policies (e.g. authoring, association, enforcement, reporting) and services (e.g. publication, exposure). The governance processes aim at coordinating the different management processes aforementioned.

Most of the processes introduced above can be the responsibility of different entities and can potentially be delegated from one user to another. This can be done in order to follow an organisation’s hierarchy.

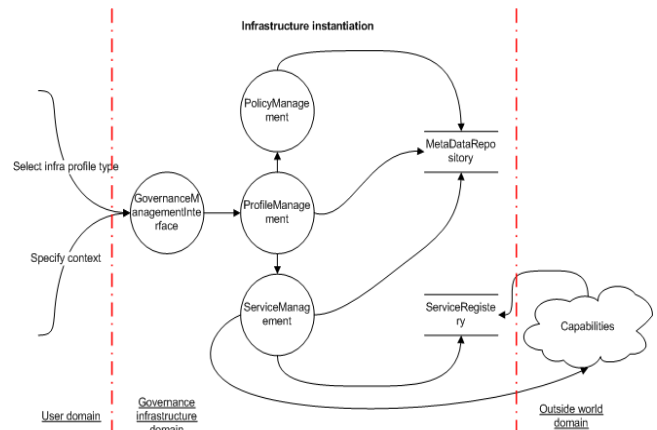
To summarise the governance infrastructure objectives with the concepts introduce in the two previous sections on operational and object models: the model aims at allowing users to expose business capabilities to clients. In order to manage the usage of this capability, a special set of infrastructure capabilities are composed and managed into a profile. Each particular exposure, with its constraints, policies and processes is then governed. This is recapitulated in Figure 1. Basic concepts and their relationships”. In different contexts, a business capability can become an infrastructure. Typically an access control service provider will manage its service as a business capability whereas this same capability will be defined as an infrastructure by other service providers.



**Figure 1. Basic concepts and their relationships**

**5.3. Governance: Management model**

The management model supports the interactions between the different elements of the infrastructure. Figure 2. Architectural diagram – top level view” presents a top level view of the model.



**Figure 2. Architectural diagram – top level view**

**5.3.1. Profile management.** Profile management, is divided into two main logical domains, the profile consistency management and the profile life-cycle management.

These domains can be respectively split in several steps: defining the infrastructure capabilities, the policy templates, the service dependencies and the information flow for the first one and defining the profile management process as well as publishing the infrastructure profile for the second.

The first aim of the profile management is to manage the life-cycle of the profiles. This consists in allowing the profile to be defined, instantiated, maintained accessible, updated and deleted. For instance, the Profile manager is to maintain the profile according to an agreement between this system and the resource owner regarding the profile instance’s availability (and ultimately that of the resource

it enhances) – e.g. the profile can be maintained at all time to enhance the performance or instantiated only when required, etc. If different infrastructures fail to comply with this requirement, the profile manager updates the profile instance with more relevant ones

In addition to the profile instance’s life-cycle management, an important task that is attributed to this manager is to handle the adaptability of the profile instance. According to the type and quality of the data held in the profile itself and the context it is aimed at, this architecture is capable of identifying threats or miss usage and react to them by modifying the profile.

Additionally, together with the Policy management infrastructure, the role of the Profile management is to determinate the best possible way to achieve the profile in the context requested. The decision making process is based on the requirements given by the user, the capabilities held by the system along with their associated constraints and the information contained in the context. The degree of automation of this activity is directly related to quality of the data held in the other core elements as introduced before.

More details on the specific stages of the profile management are given in Table 2. Creation of an infrastructure profile.

**Table 2. Creation of an infrastructure profile**

Define infrastructure Capability	1. Define Service Description 2. Define Capability policy scheme 3. Define Capability usage policy 4. Define Capability management process
Define policy Templates	5. Select Capability 6. Define policy template 7. Define domain of meta-data transformations (i/o meta-data ) 8. Define policy management processes
Define service dependencies	9. Select Infrastructure Capabilities 10. Define operation bindings 11. Define Capability invocation pattern 12. Validate Capability dependencies
Define information flow	13. Select Infrastructure Capabilities 14. Define policy meta-data transformations 15. Validate policy dependencies
Define profile Management processes	16. Select Capability management processes 17. Select policy management processes

18. Define coordination process 19. Bind management processes & coordination process 20. Validate dependencies
--

**5.3.2. Capability management.** Both capabilities management elements potentially comprise service factory and management interfaces. These elements are used to configure (e.g. setup with context aware policy) and/or replicate a service (e.g. copy service onto another server). The latter is particularly useful for infrastructure capabilities that may have to deal with heavy workloads (e.g. included in many or demanding profile instances) or different requirements (e.g. a Service Level Agreement could necessitate high availability). The management interface takes advantage of a management layer of a service, typically implemented using WS-DM, in order to configure the said service. This is also useful for infrastructures such as security service which require some sort of interaction and configuration before they can be used.

Once a profile has been instantiated and the instance made available, the enhanced business capability can be exposed. This allows insuring that an enhanced service is only used in the particular context that is relevant to its specific users.

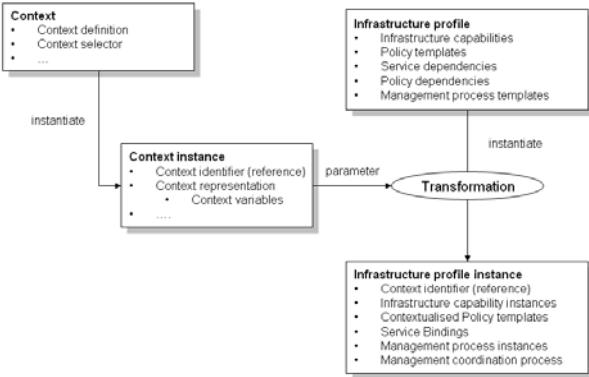
**5.3.3. Governance layer base.** The governance layer base is the middleware linking the different elements described previously.

The management interface allows resource owners (e.g. business service provider, governance infrastructure administrator) to define their requirements and/or to specify the context in which their resources should be exposed. In addition, it can be used by a different governance framework to request particular changes in certain profiles or context. The potentiality of this is defined in advanced or left to be negotiated as certain non functional properties and service exposure decisions could be notified as negotiable.

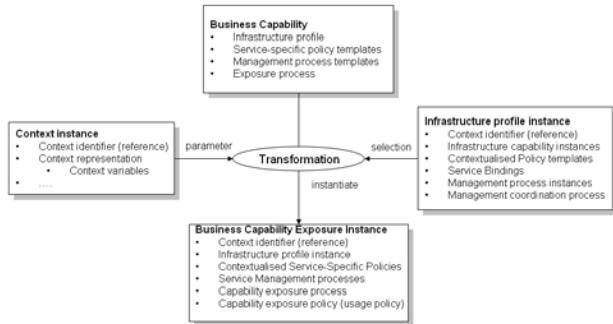
The Non Functional Requirements (NFRs) are given in term of a predefined profile (i.e. provided in an abstract form, or through directly through a list of NFRs). The request will then be processed as introduced in the previous chapter. Once this process is completed, the profile instance will consist in a composition of infrastructure services required by the requester (e.g. resource owner, other governance framework).

In addition and/or alternatively, the requester can provide a context for the business capability exposure. The context is typically formed by a transaction ID, a federation ID as introduced in [18], a WS-Addressing message ID or even an operation type the profile instance is required for (e.g. request or response).

In both cases, the request is expressed using semantics (e.g. taxonomy, XML Schemas) that are provided through the meta-data repository. The object model for these semantics is presented in Figure 3. Profile instantiation object model” and Figure 4. Capability exposure object model”.



**Figure 3. Profile instantiation object model**



**Figure 4. Capability exposure object model**

Potentially, constraints can be attached to a governance process request, these can include QoS (e.g. throughput, answer time) details for the components used as well as specific compliance requirements such as semantics to be used for certain operations (e.g. XACML, SecPAL).

Once a profile has been validated (c.f. Profile management), it can be used in order to allow the exposition of a Business Capability. In order to do so specific policies as well as exposure management processes need to be defined as illustrated in Table 3. Creation of an business capability exposure profile.

**Table 3. Creation of an business capability exposure profile**

Select Infrastructure profile instance	<ul style="list-style-type: none"> <li>21. Discover infrastructure Profiles (exposure context)</li> <li>22. Select infrastructure Profile(s)</li> <li>23. Define bindings to business capability</li> <li>24. Validate service dependencies</li> </ul>
25. Define Service-	<ul style="list-style-type: none"> <li>26. Select Infrastructure capability</li> <li>27. Refine policy template</li> </ul>

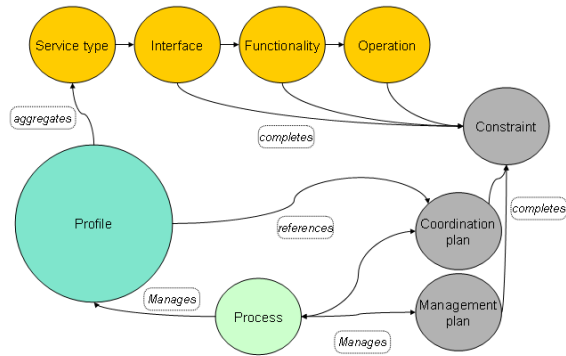
specific Policies	28. Update Capability policies
29. Define Information flow	<ul style="list-style-type: none"> <li>30. Select Infrastructure Capabilities</li> <li>31. Refine policy meta-data (service specific meta-data)</li> <li>32. Validate policy dependencies</li> </ul>
33. Define service Exposure management processes	<ul style="list-style-type: none"> <li>34. Select Profile management processes</li> <li>35. Select business Capability management processes</li> <li>36. Define coordination process</li> <li>37. Bind management processes &amp; coordination process</li> <li>38. Validate dependencies</li> </ul>
39. Publish Service instance	<ul style="list-style-type: none"> <li>40. Update Capability policy stores</li> <li>41. Update Infrastructure bindings</li> <li>42. Generate Capability exposure policy (C.E.P.)</li> <li>43. Expose business Capability to service endpoint</li> <li>44. Publish service &amp; attached C.E.P</li> </ul>

It is interesting to notice that when used by a business capability provider, this process triggers the creation a profile, but when used for an instance of this model, it instead allows configuring the governance infrastructure.

## 6. Evaluation

In order to demonstrate this governance model, the authors have developed a security governance gateway [21] that manages the security of web services that are exposed through it by the way of a security profile.

The security profile is defined by a taxonomy, presented in Figure 5. Profile description taxonomy, which describes the set of infrastructure services that are required for security (e.g. policy enforcement point, identity management, access control). This taxonomy is completed by sets of additional constraints such as policy templates; inter infrastructures coordination and management processes. Managed, these elements allow dynamically selecting and composing appropriate services to provide security and modify it on the fly when necessary (e.g. detection of a security threat, change of requirements).



**Figure 5. Profile description taxonomy**

## 7. Conclusions

In this paper, the authors have provided an overview of an architectural model for Service Oriented Architecture governance. This model is based on requirements that underline the need for policy and process management, resource life-cycle management, visibility and contextualisation. One of the roles and tested use case of the proposed model is to handle the security of web services expose through it by managing their security configuration.

In this domain, such a framework is a prerequisite for building solutions that can allow us to evaluate the profile generated against the customer's requirements and the limitations of the environment within which it will operate before it is instantiated. Future work includes integration of the model with trust and risk management frameworks and introduction of business intelligence that may adapt the choice of securing infrastructure services, their configuration and associated policies in response to detection of threats or other con-textual changes.

## 9. References

- [1] Kenney, F., Plummer, D., Magic Quadrant for Integrated SOA Governance Technology Sets, 2007, Gartner RAS Core research Note, 31 December 2007
- [2] Zhang, S. Integrating Non-Functional Properties to Architecture Specification and Analysis, in Third International Conference on Information Technology: New Generations, 2006
- [3] Galster, M., Bucherer, E., A Taxonomy for Identifying and Specifying Non-functional Requirements in Service-oriented Development, in IEEE Congress on Services 2008, 2008
- [4] Dobson, G., Hall, S., Kotonya, G., A Domain-Independent Ontology for Non-Functional Requirements, in IEEE International Conference on e-Business Engineering, 2007
- [5] Wada, H., Suzuki, J. and Oba, K., A Feature Modeling Support for Non-Functional Constraints in Service Oriented Architecture, in: Services Computing, 2007. SCC 2007. IEEE International Conference on, 9-13 July 2007, Salt Lake City, UT, 2007
- [6] Evy Troubleyn, Eli De Poorter, Ingrid Moerman, and Piet Demeester, AMoQoS: Adaptive Modular QoS Architecture for Wireless Sensor Networks, In The Second International Conference on Sensor Technologies and Applications, SENSORCOMM 2008, 25-31 August 2008, Cap Esterel, France, IEEE Computer Society, 2008
- [7] TrustCoM consortium. TrustCoM Framework for Trust, Security and Contract Management V4. Available at <http://www.eu-trustcom.com/>
- [8] TrustCoM consortium. Final TrustCoM Reference implementation and associated tools and user manual. Available at <http://www.eu-trustcom.com/>
- [9] Dimitrakos, Theo. TrustCoM Scientific and Technological Roadmap. Restricted TrustCoM deliverable available upon request. Contact: [theo.dimitrakos@bt.com](mailto:theo.dimitrakos@bt.com)
- [10] BEinGRID project resources: Website [www.beingrid.eu](http://www.beingrid.eu) - Gridipedia repository [www.gridipedia.eu](http://www.gridipedia.eu)
- [11] BEinGRID consortium. "Better Business Using Grid Solutions. Eighteen Successful Case Studies from BEinGRID". Booklet available at: <http://www.beingrid.eu/casestudies.html> See also BEinGRID industry days website: <http://www.beingrid.eu/beingridindustrydays.html>
- [12] UUID, RFC 4122, <http://www.ietf.org/rfc/rfc4122.txt>
- [13] Natis, Y.V., et al., Predicts 2007: SOA Advances. 2006.
- [14] Webinar Presentation: Sun and Layer 7 - Identity-Enabled SOA Governance, <http://www.sun.com/third-party/global/layer7/collateral/Sun-Layer7Identity-DrivenSOAGovernanceWebinar.pdf>
- [15] de Leusse, P., Periorellis, P., Watson, P. and Maierhofer, A, Secure & Rapid Composition of Infrastructure Services in the Cloud, In The Second International Conference on Sensor Technologies and Applications, SENSORCOMM 2008, 25-31 August 2008, Cap Esterel, France, IEEE Computer Society, 2008
- [16] de Leusse, P., Periorellis, P., Watson, P. and Dimitrakos, T., A semi autonomic infrastructure to manage non functional properties of a service, In UK e-Science All Hands Meeting 2008, 8-11 September, Edinburgh, UK
- [17] de Leusse, P., Periorellis, P., Dimitrakos, T. and Watson, P., An Architecture for Non Functional Properties Management in Distributed Computing, 3rd International Conference on Software and Data Technologies (ICSOFT 2008), 2008
- [18] Dimitrakos, T., Brossard, D., de Leusse, P., "Securing Business Operation in SOA", BT Technology Journal, vol.27, no.2, December 2008
- [19] de Leusse, P., P. Periorellis, and P. Watson, Enterprise Service Bus: An overview, in Technical Reports, S.o.C. Science, Editor. 2007, Newcastle University.
- [20] HP SOA Governance Interoperability Framework (GIF), Governance interoperability framework reference, February 2008
- [21] de Leusse, P. and Brossard, D., Distributed systems security governance, a SOA based approach, Third IFIP International Conference on Trust Management, Springer, June 15-19, 2009, Purdue University, West Lafayette, USA