

NEXOF-RA

*NESSI Open Framework – Reference Architecture*



**OPEN ARCHITECTURE SPECIFICATION PROCESS**

**OPEN CONSTRUCTION CYCLE #2**

**INVITATION TO CONTRIBUTE**

**ACTION REQUIRED BY February 23<sup>rd</sup>, 2009**

Date of publication: 5-Feb-2009

## 1 OVERVIEW

ICT-based services will fuel economic growth and well-being of citizens in the future of our society. In order to speed up this movement, NEXOF-RA, the NESSI flagship Strategic Project within the EU Seventh Framework Programme, is building an Open Reference Architecture for service frameworks through a collaborative process. We invite you to join with leading actors from both industry and academia, and contribute to this important community effort. In doing so, you will influence and build the service architecture of the future and give visibility to the results of your research and development.

### **DEADLINES – to participate to this second cycle, you must:**

- ▶ **register your Intent to participate by February 23<sup>rd</sup>, 2009**
- ▶ **submit a position paper by March 5<sup>th</sup>, 2009**
- ▶ **plan to join us for the kickoff meeting on March 23<sup>rd</sup> & 24<sup>th</sup> (Brussels)**

NEXOF-RA is creating an initial set of Investigation Teams, focusing on each of the topics below. Invitations to join us will be issued prior to the kickoff meeting and will be based on the position paper submitted and your motivation. Each Investigation Team is due to finish work on June 5<sup>th</sup>, 2009.

We seek contribution at this time on the following topics (see section 4 for details):

- Core Service Framework Area
  - Runtime Service Composition
- User Interaction Area
  - Metadata for Service Front End Resources (Phase I)
  - APIs for Service Front End Resources (Phase I)
- Infrastructure Area
  - Infrastructure Usage and Management Interfaces
- Security Area
  - Multilevel security for SOA
  - Dynamic security in SOA
- Quality of Service Area
  - Service Level Agreements (SLAs) and Quality of Service (QoS)
  - Federated and Autonomic Management in SOA

To obtain further information or check if there is any updated document, please visit the NEXOF-RA website ([http://www.nexof-ra.eu/open\\_construction\\_process](http://www.nexof-ra.eu/open_construction_process)). You can contact the coordinator of each topic of interest with additional questions you may have.

## 2 BACKGROUND AND RATIONALE

Globally, the question is not whether the delivery of software is shifting towards a service-oriented model, but rather *how* and *when* can this evolution be achieved. Leading industry and academic institutions have joined forces to provide a practical answer to this question, forming NESSI, the European initiative for software and services, which now counts 300 member organizations of all sizes, from SME to multinationals, including academics, technology providers and users.

Based on these requirements, NESSI has defined NEXOF, a coherent and consistent open service framework designed to guarantee quality of service, ubiquitous operation and continuous availability. NESSI has also identified the strategy and plan to build this framework, starting with its reference architecture (RA). The NEXOF-RA project implements the initial phase of this plan.

The strategy is based on open contribution mechanisms which will ensure that NEXOF meets the requirements of the user community, incorporates the best open technologies, and triggers research with concrete, applicable results. Ultimately, this process is essential in gaining wide acceptance and building a very dynamic community.

Contributors to this project will join forces in an active community, fostering partnerships with significant industrial and academic participants. The framework will be promoted and used in significant industrial deployments; this will give visibility and impact to your contributions. All contributors will gain expertise on NEXOF, which will be a key asset once the framework is deployed. Significant contributors will be recognized as co-authors.

The mission of NEXOF-RA is to address comprehensive service-oriented software system architectures and specifications. The approach of the project is to use patterns, partitioning the domain in a series of distinct problems and identifying reusable solutions for each. The project has a parallel focus on the composition of these patterns to create coherent, consistent and interoperable instances of service-oriented software systems. The reference architecture will incorporate relevant open specifications through patterns. More information about this topic is available on the NEXOF-RA website, in the document "[Definition of an architectural framework and principles](#)".

The objective of NEXOF-RA is not to replace standards bodies, but rather to leverage their work, and complement it with a strong focus on coherence and interoperability. Results from NEXOF-RA could be used as input for next versions of current standards or for new ones. Further information can be obtained from the NESSI (<http://www.nessi-europe.eu>) and NEXOF-RA (<http://www.nexof-ra.eu>) websites.

### 3 PARTICIPATION

Participation to our open process is easy:

#### 1. **REGISTER before February 23<sup>rd</sup>, 2009**

Register your intent to submit a position paper and participate to the Investigation Team. A specific web page on the project portal is dedicated to this purpose ([http://www.nexof-ra.eu/register\\_in\\_topics](http://www.nexof-ra.eu/register_in_topics)). Once registered, you may be contacted by one of our project members to discuss your participation. Participation is open to all motivated parties, including individuals, organizations and project consortiums.

#### 2. **SUBMIT before March 5<sup>th</sup>, 2009**

After registration, submit a position paper describing in a few pages your background in the domain as well as a summary of your intended contribution (your actual contribution is not expected at this stage). The position paper can be as long as required, but must contain a summary not exceeding 4 pages in case the document is longer. The papers submitted must not contain any confidential or proprietary information. Instructions on how to submit the position papers will be available on the dedicated section of the project portal ([http://www.nexof-ra.eu/open\\_construction\\_process](http://www.nexof-ra.eu/open_construction_process)).

#### 3. **PARTICIPATE between March 23<sup>rd</sup>, 2009 and June 5<sup>th</sup>, 2009**

After reviewing all the position papers, the project will issue invitations to join the Investigation Teams. The kickoff meeting for the Investigation Teams will be held in Brussels on **March 23<sup>rd</sup> and 24<sup>th</sup>**. The Investigation Team will create input which will be used as a basis to construct the Reference Architecture according to the approach described in the document "[Definition of an architectural framework and principles](#)" available on the website. Effective participation requires the following: attend face-to-face meetings, participate to phone conferences, actively engage in email discussion on the topic, perform off-line investigations or research, and co-author sections of the document. You are requested to keep the same representative through the duration of the activity. The Investigation Teams are expected to complete their work by **June 5<sup>th</sup>**.

The copyright for your contributions must allow the publication of the NEXOF Reference Architecture as an open document with derivative rights (the project is currently using the Creative Commons Attribution 3.0 License). We encourage you to check eligibility as part of one of your funded activities since NEXOF-RA will not be able fund your participation to the project.

## 4 SCOPE AND SPECIFIC EXPECTATIONS OF THIS CALL

The following pages provide details for each of the topics for which we invite you to contribute.

### 4.1 Runtime Service Composition

<p><b>Contact</b></p> <p>Jesús Gorroñogoitia – <a href="mailto:jesus.gorronogoitia@atosresearch.eu">jesus.gorronogoitia@atosresearch.eu</a> Francisco Javier Nieto – <a href="mailto:francisco.nieto@atosresearch.eu">francisco.nieto@atosresearch.eu</a></p>
<p><b>Overview</b></p> <p>This invitation to contribute (ITC) is the natural follow-up of Call 1 ITC on Design Time (DT) Service Composition. While ITC on DT Service Composition addressed the common challenges concerning service composition at design time, this new ITC completes that picture by addressing the runtime concerns and the innovative features which can be provided by execution engines.</p>
<p><b>Problem Statement</b></p> <p>Commonly, business processes are implemented, according to SOA principles, by designing and executing some composite services that perform, driven by some work and data flow, a set of tasks leveraged on external services.</p> <p>Even if the composite service (process hereafter) can be mostly specified at design time, there may be some aspects of the process specification that requires to be addressed at runtime. Besides, at runtime, processes are activated, executed, monitored, adapted, managed, etc.</p> <p>In the European context, some IP6/IP7 EC projects have covered, to some extent, the challenge of the specification, execution, management, monitoring, etc. of business process implemented as SOA composite services, providing some promising results, but there are still some concerns that require further investigation. The purpose of the topic is to identify and describe those concerns on service composition at runtime and propose some widely-accepted solutions based on the current research done under those projects and other initiatives, which may improve processes execution and increase their robustness, flexibility and automation.</p> <p>A non-exhaustive and incomplete collection of service composition at runtime concerns includes:</p> <ul style="list-style-type: none"> <li>• Effective dynamic hot deployment and activation of processes into the execution engine, integrated within the overall SOA governance system.</li> <li>• Parameterisation of abstract processes specified at design time using abstract composition, templates, etc., by exploiting execution context. This may include late-binding, re-binding policies, etc.</li> <li>• Support to the negotiation process on the basis of agreed SLAs, which may drive the service selection.</li> <li>• Management of long-lasting process execution, and their interaction with users and/or external applications.</li> <li>• Improved management of exceptional situations during the process execution</li> </ul>

and its dynamic behaviour. Process self-healing, self-configuring and self-optimization support, including service replacement, compensation, re-planning, context adaptation, etc.

- Process lifecycle management, including non-intrusive monitoring, which may also drive corrective actions.
- Improvements of mechanisms for non-functional aspects support, such as transaction protocols, transparent security approaches, etc.

### **Scope**

This ITC focuses on the dynamic concerns of SOA processes lifecycle at runtime: execution, monitoring, adaptation, management, etc. For instance, process adaptation in reaction to some monitoring feedback could be covered by the topic, concerning the monitoring of process execution. However, no concrete contributions on service monitoring are expected, since they should be covered by other ITCs.

Similarly, SOA governance could be partially covered by this topic, regarding the management of its internal state (for instance, the current execution point within the workflow).

We foresee other links to Call 1 ITCs, such as, for instance, Service Discovery, which may be relevant at runtime to support some self-healing techniques (i.e. service replacement), but not a deep analysis is expected since it was covered by that topic.

### **Contributions**

This call expects to define a general conceptual and technological framework for service composition that is as much as possible completed and self-consistent. Therefore, the following contributions are expected:

- Reports on the topic challenges that contribute for their better understanding, specification and description.
- Identification and description of design patterns to be applied to address some of the challenges aforementioned.
- Additional contributions like techniques, specifications, standards, frameworks, tools, best practices and guidelines that may help to face the topic challenges in different scenarios.

### **Baseline**

The baseline for this call is the standard WS technological stack, for backward compatibility reasons. Besides, orchestration and choreography technologies (as BPEL4WS/WS-CDL) are considered baseline technologies, although they were not conceived to address some of the concerns of this topic, so extensions over this baseline are expected. We expect to incorporate to this baseline, as part of this IT results, those wide consensual techniques obtained from foregoing/ongoing aforementioned projects.

### **For further information**

[http://www.nexof-ra.eu/?q=runtime\\_service\\_composition](http://www.nexof-ra.eu/?q=runtime_service_composition)

## 4.2 Metadata for Service Front End Resources (Phase I)

<p><b>Contact</b></p> <p>Marcos Reyes (Telefónica) – mru@tid.es          José Manuel Cantera (Telefónica) – jmc@tid.es          Nikolaos Tsouroulas (Telefónica) – nik@tid.es</p>
<p><b>Overview</b></p> <p>It is necessary to establish the metadata describing all the information associated to Service Front End Resources (SFERs) to allow its integration and interaction into different platforms and environments, such as mashup platforms or web runtimes. Metadata will include, among others, cataloguing information (creator, version, icon...), external properties (persistence, configurable preferences, context subscriptions, data published / consumed...), required APIs, etc.</p>
<p><b>Problem Statement</b></p> <p>In order to exploit the SFER-platform capabilities today SFERs must be coupled to their execution environments through platform dependant metadata. Having standard metadata schemes will allow the usage of those SFERs in different environments increasing the decoupling between SFERs and the target execution platforms.</p> <p>A solution based on minimums is not desirable, so it is necessary to define a set of abstract capabilities described by the metadata and interpreted by the platform, so each SFERs can take advantage of the best functionalities available in each execution environment.</p> <p>SFERs metadata should be restricted as far as possible to declarative information, letting the implementation issues to other parts integrating the SFR development.</p>
<p><b>Scope</b></p> <p>As the list of metadata items can be long this investigation team will focus on a specific subset. Future investigation teams might deal with the rest.</p> <p>The following groups of metadata items are in scope:</p> <ul style="list-style-type: none"> <li>• Cataloguing Information (author, icon, version ...)</li> <li>• Published / Consumed data items</li> <li>• Persistency requirements</li> </ul>
<p><b>Contributions</b></p> <p>The contributions accepted by this IT might take the form of:</p> <ul style="list-style-type: none"> <li>• Formal vocabularies for SFER metadata</li> <li>• Formats for declaring SFER (based on XML or RDF)</li> </ul>
<p><b>Baseline</b></p> <p>There is no specific baseline identified for this Investigation Team. Nonetheless it is anticipated that the IT should study existing W3C and Open AJAX Alliance working drafts.</p>

**For further information**

[http://www.nexof-ra.eu/?q=metadata\\_for\\_service\\_front\\_end\\_resources](http://www.nexof-ra.eu/?q=metadata_for_service_front_end_resources)

(a white paper with further details is available on the web) site)

## 4.3 APIs for Service Front End Resources (Phase I)

<p><b>Contact</b></p> <p>José Manuel Cantera (Telefónica) – jmc@tid.es          Marcos Reyes (Telefónica) – mru@tid.es          Nikolaos Tsouroulas (Telefónica) – nik@tid.es</p>
<p><b>Overview</b></p> <p>This IT will be in charge of identifying a set of client-side APIs for Service Front End Resources (SFERs).</p>
<p><b>Problem Statement</b></p> <p>It is necessary to identify a set of APIs to enable the creation of user interfaces that fully exploit client-side platform capabilities. Such APIs will provide uniform interfaces to the functionalities provided by runtime execution environments (mash-up platforms, web runtimes, etc.). The APIs might include, among others, network connections, off-line access, publish / subscribe, persistence, clipboard, drag &amp; drop, device capabilities, etc.</p>
<p><b>Scope</b></p> <p>As the list of APIs can be long this investigation team will focus on a specific set and future investigation teams might deal with the rest.</p> <p>The following APIs are under the scope of this invitation to contribute:</p> <ul style="list-style-type: none"> <li>• SFERs interconnection through a publish-subscribe paradigm.</li> <li>• Persistence.</li> <li>• Network Connections.</li> </ul>
<p><b>Contributions</b></p> <p>The contributions accepted by this IT might take the form of:</p> <ul style="list-style-type: none"> <li>• API formal specifications, provided that they are both language and platform independent</li> <li>• List of functionalities that would need to be addressed by the APIs that are under scope</li> </ul>
<p><b>Baseline</b></p> <p>There is no a specific baseline identified for this Investigation Team. Nonetheless it is anticipated that the IT should study existing W3C working drafts, such as XMLHttpRequest Level 2, File Upload, Network API or File I/O.</p>
<p><b>For further information</b></p> <p><a href="http://www.nexof-ra.eu/?q=apis_for_service_front_end_resources">http://www.nexof-ra.eu/?q=apis_for_service_front_end_resources</a>          (a white paper with further details is available on the web)</p>

## 4.4 Infrastructure Usage and Management Interfaces

### Contact

Mike Fisher – mike.fisher@bt.com

### Overview

This topic addresses the interactions between an ICT resource infrastructure (computing, storage, network and execution environments) and the applications or components that use it. It deals with the interactions involved when an infrastructure service is being used – including information exchanges between application and infrastructure as well as facilities to allow the user to monitor and control infrastructural aspects of the service.

### Problem Statement

NEXOF services are underpinned by a flexible, heterogeneous set of resource infrastructure services. These will be provided by a number of independent stakeholders to meet specific technical and market needs, and we can already see examples emerging in the form of various on-demand computing or hosting services, typically with usage-based charging (e.g. Cloud computing, Platform as a service). NEXOF aims at an extensible, decentralized global computing environment, which is open in the sense that there are no unnecessary barriers to participation.

The NEXOF architecture should make it possible for users to deploy software components that make use of any infrastructure resources which are appropriate to their needs, including functional, non-functional and commercial. The architecture therefore needs to specify how software can make use of a range of resource infrastructure services, identifying common features and capabilities.

Adaptive behaviour by both the infrastructure and the application is desirable to make the user experience more dependable. This means that the infrastructure provider needs knowledge of the components he is hosting that goes beyond the “black box”. Infrastructure management facilities should be available to the infrastructure service user or application developer. These should include both access to monitoring information, such as whether there are any faults or performance issues, and control interfaces, such as the ability to request additional resources or to migrate components between geographical locations or service providers.

It is expected that this topic will address:

- the specification of approaches to software deployment eg.descriptors
- approaches to interoperability/portability between infrastructure providers, possibly including unification of interfaces or brokering
- information models, mechanisms and protocols for exchanging management information between user and resource provider or between resource providers

### Contributions

Specific proposals for deployment descriptors, interface definitions, management information models and communication protocols are particularly welcome.

**Baseline**

Existing Internet and Web Service standards are expected to form the starting point for this topic.

**For further information**

[http://www.nexof-ra.eu/?q=infrastructure\\_usage\\_and\\_management\\_interfaces](http://www.nexof-ra.eu/?q=infrastructure_usage_and_management_interfaces)

## 4.5 Multilevel Security for SOA

### Contact

Pascal Bisson (Thales) – pascal.bisson@thalesgroup.com

Daniel Gidoïn (Thales) – daniel.gidoïn@thalesgroup.com

### Overview

This call addresses the area of the multilevel security for SOA. Service-oriented architectures are dynamic, flexible and compositional in nature. Security is a significant challenge for Service-Oriented-Architectures (SOA) in a multi-domain environment. Security incorporates the concept of Multi-Level Security (MLS). MLS has been until recently a niche market with only a few government agencies needing it. However, in recent years, there has been an emphasized need for multiple government agencies to share information on a need-to-know basis. Hence, there is a government push to migrate the existing isolated MLS infrastructures to a single integrated MLS infrastructure. Therefore, supporting MLS in large scale distributed enterprise systems becomes an urgent and critical requirement for intra- and inter- enterprise collaborations.

### Problem Statement

The MLS concept was originally described in the DoD Orange Book on the needs of common evaluation criteria. At that time, there was almost no concept of distributed computing, Web services, policy management or metadata technologies.

An MLS system is supposed to operate as follows: all resources are assigned a security label denoting the sensitivity of the resource; users are issued security clearances denoting their trustworthiness and the types of information they need to know; mandatory access control compares each user's clearance with each resource's label before access is granted.

With the advancement of technologies such as web services, SOA, ontology and the deployment of networks, to achieve MLS in a distributed computing environment today, in reality, it must do the following: provide mechanisms at the hosts and network nodes to enable security services at each specified classification level; provide the ability to enforce accountability by logging an audit trail of all events; guarantee impenetrable barrier between treatments, services and information of different levels of sensitivity, according to security classification.

### Scope

The scope is multilevel security for SOA.

The aims of this call are to provide solutions for making implementable and deployable improvements to the usability of multilevel security.

The main control functions expected of an MLS system includes:

- Access control. This is accomplished through the use of access control lists that identify the users that can access a given resource (service, data and their level of authority with regard to that resource);
- Auditing: Audit records associate security-related events (such as file access) with the user that caused the event;

- Name-hiding: The names of files, data sets and directories are only displayed to users with access authority. Users without a need-to-know will not see the file or object listed or displayed;
- Write-down prevention: To prevent users from declassifying data, in order to grant access to users without a need-to-know and or of lower level of classification.

With the availability of SOA and the capabilities of applications in today's distributed computing environment, there are other low level functions required in order to ensure a true single integrated MLS system environment. An MLS system design should achieve the following goals:

- To establish controls that prevent users from accessing information at a higher classification than their authorization permits;
- To ensure that the controls prevent unauthorized users from declassifying information.
- To enable information on an as-needed basis among multiple administrative domains.

Recommendations: to propose MLS having realistic implementation plans and budgets.

### **Contributions**

MLS can be one of the services provided in a SOA environment. The MLS service requires the deployment of security mechanisms at different layers. So, because of the maturity of technologies, we think to expand to support MLS in a SOA environment is very feasible, in particular for the following reasons: availability of standard web-services interfaces, languages and protocols; almost all security mechanisms are standard based implementations; and availability of MLS functions. In recent years, open-source operating systems became ideal platforms for implementing MLS.

The key components of the MLS architecture and expected contributions concern:

- Integration of diverse MLS services and tools into the architecture;
- Establishment Services. This service interfaces with a policy manager to determine if the user is authorized to ask for the requested classification level. If yes, the service gets the security resources requirements and determines what actions are permitted for each object state given the user's security classification and need-to-know level;
- Security Configuration Service: This service matches the security resources requirements and the configuration of the security infrastructure that are involved.

### **Baseline**

The baseline is composed of web services standards (OASIS..), J2EE, technologies in the areas of dynamic configuration management, object metadata model, rules execution tools.

### **For further information**

[http://www.nexof-ra.eu/?q=multi-level\\_security\\_for\\_soa](http://www.nexof-ra.eu/?q=multi-level_security_for_soa)

## 4.6 Dynamic Security in SOA

### Contact

Pascal Bisson (Thales) – pascal.bisson@thalesgroup.com

Daniel Gidoïn (Thales) – daniel.gidoïn@thalesgroup.com

### Overview

This call addresses the area of dynamic security for SOA.

The evolution of dynamic execution environments increasingly requires security policies that are also dynamic in nature to address such events as process migration, changes in personnel, shifts in alliances, and detected intrusion that cannot be well anticipated or addressed by static policies.

In addition, Web Services (WS) will play a significant role in the next generation Web. However, the attractive features of WS such as platform independence, XML and SOAP reliance, and simplicity to use, make them vulnerable to many security threats including new unexplored and inherited old problems.

Dynamic separation of duties, delegation and other dynamic security constraints requires the state of the security system to be managed explicitly at run-time. So, dynamic separation of duty constraints are a form of history-based access control. The permission for an actor to take certain actions in some context will depend on him/her not having performed related actions in that same context already. For example, a clerk may be authorised to sign or countersign a given cheque, but a single clerk is not authorised to carry out both actions.

### Problem Statement

The evolution of dynamic execution environments and dynamic security adaptability requires architectures that adapt to changing security policies during runtime with minimal loss of functionality and with little or no manual assistance.

In this context, we must take into account the ability to reconfigure the global security policy at any time to address events such as shifts in alliances, changes in personnel, changes in the execution environment (e.g. transition from trusted execution environment to untrustworthiness execution environment), crisis situations...

Security agility is a software flexibility technique to address security properties and their dynamic evolution. An agile software component is cognizant of the security environment in which it executes, is aware of its responsibilities for enforcing "its part" of a more global policy, and contains internal mechanisms that adapt its functionality in coordination with authorized external policy changes. The heterogeneous nature of a dynamic execution environment presents some significant obstacles to developing dynamic security. The first such impediment is the wide range of possible security semantics. A variety of access control policies might be employed, for example, including information disclosure policies, role based policies.

The variety of architecture components employed in a heterogeneous and/or ubiquitous environment, including operating systems, system software, and mission-specific software require the development of flexible techniques that are not bound to a single environment.

**Scope**

To help overcome the heterogeneity obstacles, dynamic security techniques can employ various strategies to address security properties and their dynamic evolution. For example, they can embed components with pre-formulated security policy models (security patterns) and mechanisms to provide policy awareness in support of security reconfiguration.

They can also provide a flexible component architecture that allows dynamic code extensions for adding new security semantics or changing security-relevant behaviour to maintain compatibility with new security rules or execution environment.

**Contributions**

The contributions can take different shapes: 1) security policies and dynamic models. Security policy models allow a component to be aware of the security policy governing their operation. 2) Policy components awareness. One important consideration in dynamic execution environments is the likely fluctuation in resource availability resulting from security policy reconfiguration. When processes find themselves unable to access resources they expect, undesirable events may occur, including termination of critical processes. Process dependencies, such as client/server relationships, are often subtle, or even unknown, and may be overlooked when dynamic policy changes are implemented, particularly in time-critical situations such as in response to an intrusion detection event. 3) Dynamic security architecture (model, mechanisms, patterns, components, toolkits). Once components have been made aware of the security policy changes that could affect their execution, they can be extended with adaptive functionality that reacts positively to these changes, rather than failing in some manner. Positive responses might include terminating connections invalidated by policy changes, temporarily suspending or reducing normal operations until lost resources become available once again, reacquiring lost resources, or switching to alternate algorithms to produce equivalent results. The dynamic security toolkit architecture facilitates the coordination of policy awareness and adaptive behavior functionality.

**Baseline**

The baseline is composed of web services standards (W3C, OASIS), J2EE, and the standards from the service privacy forum.

**For further information**

[http://www.nexof-ra.eu/?q=dynamic\\_security\\_for\\_soa](http://www.nexof-ra.eu/?q=dynamic_security_for_soa)

## 4.7 Service Level Agreements (SLAs) and Quality of Service (QoS)

<p><b>Contact</b></p> <p>Ricardo Jiménez-Peris – <a href="mailto:rjimenez@fi.upm.es">rjimenez@fi.upm.es</a></p>
<p><b>Overview</b></p> <p>This topic covers two issues. The first one is service level agreements (SLAs) in all its aspects including SLA description, SLA translation, SLA monitoring, and SLA negotiation. The second issue covered by the topic is how to guarantee and/or enforce quality of service (QoS) to satisfy requirements coming from SLAs.</p>
<p><b>Problem Statement</b></p> <p>The ITC focuses on SLAs and how to enforce QoS requirement derived from the SLAs. On the SLA side expected contributions include languages for describing SLAs, approaches to translate SLAs, architectures to monitor SLAs, interfaces, protocols, and standards to negotiate SLAs, etc. On the QoS side contributions are expected to concentrate on how to enforce QoS requirements set by SLAs on regular and large scale systems (e.g. in cloud computing).</p>
<p><b>Scope</b></p> <p>Contributions on all aspects of SLAs and on how to enforce QoS requirements derived from SLAs are welcomed. QoS contributions without a link to SLAs are not in scope.</p>
<p><b>Contributions</b></p> <p>Architectural contributions for both SLAs and QoS are expected to take the shape of architectural patterns. Contributions regarding SLA description are expected to be in the form of languages for describing SLAs. Also descriptions of interfaces are welcomed for SLA negotiation and SLA monitoring.</p>
<p><b>Baseline</b></p> <p>The baseline is SOI in any shape either traditional (multi-tier) or more innovative (SaaS and cloud computing).</p>
<p><b>For further information</b></p> <p><a href="http://www.nexof-ra.eu/?q=service_level_agreement_and_quality_of_services">http://www.nexof-ra.eu/?q=service_level_agreement_and_quality_of_services</a></p>

## 4.8 Federated and Autonomic Management in SOA

<p><b>Contact</b></p> <p>Ricardo Jiménez-Peris – <a href="mailto:rjimenez@fi.upm.es">rjimenez@fi.upm.es</a></p>
<p><b>Overview</b></p> <p>This topic focuses on two main issues. First, how the management of different service infrastructures can be federated to obtain a holistic management of the whole service infrastructure (including the federation of the management virtualization infrastructures). Second, how to enrich SOA management with autonomic capabilities to obtain self-management (self-healing, self-provisioning, self-optimization, self-configuration).</p>
<p><b>Problem Statement</b></p> <p>The first issue lies in how given different service infrastructures used in combination to support a particular set of services how can be managed as a single logical entity by federating the individual management of the different infrastructures. For instance, in a multi-tier architecture with web, application and database tiers how the management of the three tiers can be federated to obtain a holistic management.</p> <p>The second issue consists in how to incorporate autonomic capabilities into the management of service infrastructures. More concretely how to obtain self-* properties for SOI such as self-healing, self-provisioning, self-configuration, and self-optimization.</p>
<p><b>Scope</b></p> <p>The scope is on federated and autonomic management. Regular management such as Life Cycle Management is out of scope since it will be covered by a specific ITC.</p>
<p><b>Contributions</b></p> <p>The contributions on federated management are expected to take the form of architectural patterns and/or management interfaces to enable federation. The contributions on self-management are expected to take the form of architectural patterns for generic and concrete SOI.</p>
<p><b>Baseline</b></p> <p>The baseline lies in current approaches to SOI, either traditional ones such as multi-tier architectures or newer ones such as SaaS and cloud computing.</p>
<p><b>For further information</b></p> <p><a href="http://www.nexof-ra.eu/?q=federated_and_autonomic_management_in_soa">http://www.nexof-ra.eu/?q=federated_and_autonomic_management_in_soa</a></p>