

NEXOF-RA

NESSI Open Framework – Reference Architecture

IST- FP7-216446



Deliverable D5.2.1

Open Architecture Specification Process

PROGRAMME

Frederic Gittler (HP Labs)

With contributions from most NEXOF-RA Partners

Due date of deliverable: 01/09/2008

Actual submission date: 15/10/2008

This work is licensed under the Creative Commons Attribution 3.0 License.

To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

This work is partially funded by EU under the grant of IST-FP7-216446.

Change History

Version	Date	Status	Author (Partner)	Description
BASELINE	21-Jul-08	NA	Multiple NEXOF-RA partners	Publication of Programme in 1 st Invitation to Contribute
DRAFT	31-Jul-08	Draft	F. Gittler (HP Labs)	Initial version – Capture of Programme in this document

EXECUTIVE SUMMARY

This document describes the initial topics of the first Open Construction Cycles for the NEXOF-RA Open Architecture Specification Process. The content of this programme has been first published in the Invitation to Contribute released on July 21st, 2008.

Document Information

IST Project Number	FP7 – 216446	Acronym	NEXOF-RA
Full title	NESSI Open Framework – Reference Architecture		
Project URL	http://www.nexof-ra.eu		
EU Project officer	Arian Zwegers		

Deliverable	Number	D5.2.1	Title	Open Architecture Specification Process - Programme
Work package	Number	5	Title	Open Architecture Specification Process

Date of delivery	Contractual	01/09/2008	Actual	15/10/2008
Status	Version 1, dated 15/10/2008		final <input checked="" type="checkbox"/> (for this version)	
Nature	Report <input checked="" type="checkbox"/> Demonstrator <input type="checkbox"/> Other <input type="checkbox"/>			
Abstract (for dissemination)	List of topics considered for the Open Construction Cycles of the NEXOF-RA Open Architecture Specification Process			
Keywords	Programme			

Internal reviewers	Reviewer1 (Organization)		
	Reviewer2 (Organization)		
Authors (Partner)	Frederic Gittler (HP Labs) + many other contributors in the project		
Responsible Author	Frederic Gittler		Email Frederic.gittler@hp.com
	Partner	HP Labs	Phone +33 (0)476 144 289

TABLE OF CONTENTS

EXECUTIVE SUMMARY	3
TABLE OF CONTENTS	5
1 INTRODUCTION	6
2 SCHEDULE	7
3 TOPICS FOR 1ST INVITATION TO CONTRIBUTE	8
3.1 Service Description	8
3.2 Design Time Service Composition	11
3.3 Service Discovery	13
3.4 Interoperability of Message-Based Service Interaction	15
3.5 Declarative Authoring Language for User Interfaces	17
3.6 Context Model and Universal APIs	19
3.7 Definition of Infrastructure Services	21
3.8 Dynamic identity management for SOA	23
3.9 Privacy Management in SOA	25
3.10 Scalable Approaches to Service Oriented Infrastructures	27
3.11 Highly Availability for Multi-Tier Architectures	29
4 TOPICS FOR FUTURE CALLS	31

1 INTRODUCTION

NEXOF-RA has defined an Open Architecture Specification Process (OASP) which includes a set of Open Construction Cycles (OCC) executing in parallel. Each OCC has the purpose of addressing a single, well defined topic. Each of these topics addresses a part of the NEXOF Reference Architecture. This document presents the topics which have been the object of the first Invitation to Contribute, as well as the list of candidate topics (title only) for the first following calls. This content has been originally published in the first Invitation to Contribute.

2 SCHEDULE

The OCC section of the OASP defines a series of milestones for the execution of each OCC. At this time, only the first two batches (calls) of OCC have been scheduled, as follows:

Table 1: OCC Schedule (first two batches)

Milestone	CALL 1	CALL 2
Topics Identified	03-Jul-2008	20-Nov-2008
PHASE 1 –		
Invitation to Contributed Release	21-Jul-2008	05-Dec-2008
Deadline to register interest	05-Sep-2008	09-Jan-2009
Deadline to submit Position Paper	03-Oct-2008	06-Feb-2009
PHASE 2 –		
Investigation Teams Kickoff Meeting	20-Oct-2008	18-Feb-2009
Investigation Teams Final Results	16-Jan-2009	30-Apr-2009

All the above dates are indicative only. The project adopts a balanced approach between schedule and quality. Results of the first Invitation to Contribute may affect the schedule and content of the second batch.

3 TOPICS FOR 1ST INVITATION TO CONTRIBUTE

The content of this section has been first published on 21-Jul-08.

3.1 Service Description

<p>Contact</p> <p>Piero Corte (Engineering) – piero.corte@eng.it</p>
<p>Overview</p> <p>This call addresses the problem of describing services. The particular challenge is to identify methods for describing and representing services that allow to enhance reuse and automate the composition of services.</p>
<p>Problem Statement</p> <p>Adoption of Service Oriented Architecture (SOA) is expected to improve the way how enterprises effectively cope with the ever changing and dynamic businesses of today in a timely way. This relies on the capability provided by SOA to support and accommodate new business solutions dynamically by supporting reuse of functional assets (services).</p> <p>The fundamental principles of the SOA paradigm concern the ability to create and exploit functional assets, organize them and enhance their reuse and composition for the realization of new or modified business processes. Effective methods for describing and representing services are needed to enable and eventually automate all these activities. For this reason, service description is one of the most fundamental characteristic of SOA.</p>
<p>Scope</p> <p>The problem of describing and representing services can be analysed from different perspectives. Description languages, methods and tools can be designed specifically for supporting different service related activities, such as service creation, management, discovery, invocation, and composition. Therefore, to fully qualify a solution to service description, it is important to focus on the question <i>“For which activity is a given description useful?”</i></p> <p>This call focuses on description languages, their expressiveness and their easiness of usage with respect to their primary purpose (that is, discovery, composition, etc), but not on matching or search algorithms, neither on service composition itself which are respectively addressed by the “Service Discovery” and “Service Composition” calls.</p> <p>Even if quality of services, policies and usage contracts are commonly and rightly recognized as service descriptions, this call does not specifically address these kinds of descriptions since they are directly addressed by other calls. A similar case are description methods that deal with the deployment and configuration of services.</p>

Contributions

Contributions to this call should have the form of architectural patterns. These patterns will constitute pieces of designing solutions that will be integrated with other provided patterns to specify the overall design of a SOA-based system.

The documentation of the proposed design patterns is expected to supply sufficient details about the pattern. In particular, it should contain a description of

- the problem the pattern solves, its intent, the forces and consequences of its application,
- its interfaces and usage,
- a guideline for its implementation,
- its internal structure or/and implementation,
- standards to which it is related.

Baseline

In the following we list some of the major and widely adopted models and standards for service description and position them with respect to the perspective previously introduced.

Service Creation	Service Discovery (querying, matching, browsing)	Service Invocation (message delivery and mediation)	Service Composition (process creation)
UML	UDDI WSMO SAWSDL	WSDL WSMO SAWSDL	BPEL WSMO UML

Please keep in mind that the standards specified in the above table are only examples and the contribution can concern any other standard or language.

The approach of a contribution can support more than one activity.

Classifying the contributions according to the activities is very valuable, since it helps to distinguish alternative or complementary approaches and allows us to select the ones that are most suitable in a given case.

Thus any approach should be classified accordingly the activities it is specifically conceived for. For instance, WSDL is mainly conceived to enable Service Invocation and although it may be used for Service Discovery it should probably not be advocated for such an activity. While the activities mentioned above are coarse grained, finer distinctions are encouraged. For example, both, WSDL and WSMO, are useful for Service Invocation, but while WSDL can be used for the delivery of messages and does not specifically address protocol mediation or semantic mediation, WSMO addresses them.

Each proposal should compare the proposed approach to other competitive

approaches, using the same perspectives.

For further information

http://www.nexof-ra.eu/service_description_techniques

3.2 Design Time Service Composition

Contact

Jesús Gorroñoigoitia (Atos Origin) – jesus.gorronogoitia@atosresearch.eu

Overview

This call addresses the problem of composing services by aggregating other services participating in a common business process that is reified by the composite service. It focuses on the composition of services at design time and on the aspects that complicate or impede a wide adoption of composition techniques by SOA practitioners with diverse background and expertise. A list of such aspects may, for instance: include the integration of humans as first class participants in composite services, the question of graphical vs. executable modelling of compositions, composition design patterns, selection of choreography vs. orchestration descriptions, decomposition and planning of complex composition tasks, etc.

Problem Statement

One of the main SOA principles is composability, a concept originated from the area of component oriented architecture. Closely related to composability are other fundamental SOA principles like reusability, autonomy and loose coupling.

Composite services are those built by aggregation of other services that are invoked according to some structural and behavioural patterns described in an orchestration and choreography specification.

Composite services can be specified at design time, planning a workflow (process and data flow) that accomplishes the service capability by leveraging on invocations to other services. Flexible orchestration and choreography languages and tools (taken from the BPM domain) satisfy in most cases that specification.

However, there remain still some topics that need improved solutions:

- In practice, composite services are mostly used to implement business processes, but there are still gaps between Business Process Modeling (BPM) techniques and the techniques used for service composition.
- Support, during the entire design cycle, for the participation of different actors with greatly differing expertise like, for instance, BP analysts or integrators of composite services is largely missing. Back and forth tracking, even during composite execution, should be possible.
- For humans, it is still difficult to interact with composed services in a given workflow if the currently widely adopted service composition techniques are applied.

The assistance for users struggling with the complexity of service composition is still weak. Additional engineering means for the development of service compositions are required, as, for instance, pattern orientation in composition design (including a set of composition design patterns), supporting for service search based on requirements and on composition design patterns,

management of choreography constraints, etc.

Scope

This call focuses on techniques concerning composability of web services. Concrete studies on BPM are not considered necessary unless they pertain to aspects that clarify the relationship between BPM and the modelling of business processes with composite services. Furthermore, no deep analysis of orchestration and choreography is required but their role and their strengths and weaknesses with respect to service composition should be investigated. Neither is a deep analysis on service discovery expected (addressed by the call on Service Discovery), but studies how discovery strategies can be applied when services are to be composed. Eventually, service mediation in general (partially addressed by the call on Service Interoperability) is not in the scope of this call but those mediation aspects complementing the choreography of services in the data and process workflow of composite services.

Contributions

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

- Standard specifications and languages to define compositions of services (including wide accepted graphical notations) covering the wide range of user's expertise, allowing tracking back/forth graphical and executable notations.
- Contributions like techniques, specifications, tools, best practices and guidelines that assist the users in the specification of compositions:
 - Best architectural and design patterns for composition, including guidelines for composition from scratch and task decomposition including planning techniques that also consider the granularity/latency trade-off.
 - Discovery assisted composition strategies and techniques/tools (including requirement based discovery assisted composition, architecture/pattern based discovery assisted composition, etc).
 - Lightweight vs. Heavyweight composability approaches, highlighting techniques to hide composability complexity.
- Best practices and strategies for adopting orchestration and choreography descriptions of compositions. Complementary descriptions, adopting orchestration for workflows and choreography for interoperability
- Service mediation techniques, including process and data mediation, service negotiation, etc in the context of service composition.
- Standard specifications and techniques to allow an integrated participation of humans in processes implemented by service compositions. Standard specifications for describing human tasks within those compositions.

Baseline

The baseline for this call is the WS technological stack; this implies the full compatibility of the expected contributions with this technology. BPEL4WS/WS-CDL is also considered a basic underlying technology; however, also other options may be taken into account.

For further information

http://www.nexof-ra.eu/design_time_service_composition

3.3 Service Discovery

Contact

Jesús Gorroñoigoitia (Atos Origin) – jesus.gorronogoitia@atosresearch.eu

Overview

That services are “discoverable” is one of the key factors for the success of SOA. Services are fully described and these descriptions are published in publicly accessible SOA registries so the services may be later discovered. However, discovering services could be a cumbersome process if it had to be done manually by browsing in distributed registries with a large number of service descriptions, without the aid of semi-automatic integrated searching facilities, especially in the composite service development lifecycle.

Problem Statement

SOA strength is that services can be consumed as many times a particular user requires them. This is possible since services are discoverable. This capacity is based on the ability of services to describe themselves through well-defined formal descriptions that are published on SOA registries, and, once there, are publicly available for retrieval and inspection through browsing or other discovering mechanisms.

Discovering of services by browsing is not adequate for SOA registries with large content. Therefore, fast and accurate semi-automatic search and selection facilities are required which perform queries within local or remote, centralized or distribute, single or federate SOA registries.

Service discovery may rely on: a) the user’s ability of precisely describe its request, b) in the algorithms applied to match that user request with the capabilities of candidate services, and c) in the algorithms to rank and select the best candidate among those discovered.

Service discovery is intensively used by SOA practitioners when they compose other services or when they require invoking an external service from some application or process. Hence, the service discovery process is quite relevant in SOA engineering cycles.

Even if service discovery is, to some extent, well covered by the SOA techniques and tools, there are still some challenges, especially with techniques to specify user requests, semi-automatic service discovery, and the ranking and selection facilities. Also determining the role of service discovery in the SOA

engineering phases like service composition specification, runtime, etc., requires still further attention.

Scope

This call focuses on service discovery techniques and strategies and their applicability during certain service development phases (composition, invocation, etc.). It is not concerned with service description specifications, except in those cases where describing user request and services capabilities are relevant for the discovery process.

Contributions

This call expects contributions for a general conceptual and technological framework for service discovery that should be aimed as much complete and self-consistent as possible. Therefore, it is expected:

- Techniques and language specifications for representing the user's request. The technology should also support the creation of consumer's goals, isolating as much as possible the underlying technological complexity.
- Strategies, techniques, best practices, etc. for accurate and precise matching between consumers' goals and services' capabilities. Most appropriate matching algorithms and guidelines to accommodate then to particular searching scenarios are also expected.
- Techniques, algorithms, best practices, etc. for semi-automatic service ranking and selection among those service specifications retrieved by the discovery process. As above, guidelines are expected to accommodate selection algorithms to particular searching scenarios.
- Strategies, techniques for service discovery applied to some frequent service discovery scenarios in composite services: requirement based service discovery (that is, based on stakeholders or analysts requirements elicitation process), architecture based service discovery (that is, based on design patterns and choreography constraints applied to the composite service) and runtime service discovery (that is, postponing concrete binding to services to execution time), among others.

It is expected to explore other aspects concerning service discovery and selection such as consumers' and providers' context, SLA, negotiation, etc

Baseline

Baseline for this call is the WS technological stack; contributions are assumed to be full compatibility with this technology. There are no other assumptions or constraints.

For further information

http://www.nexof-ra.eu/service_discovery

3.4 Interoperability of Message-Based Service Interaction

Contact

Katharina Mehner (Siemens) – katharina.mehner@siemens.com

Overview

Services operate by exchanging messages with each other. Each service needs to understand each others' messages completely and unambiguously.

Services, however, are developed independently according to different standards and techniques. Furthermore, the same standards are often used in different ways. This jeopardizes the interoperability between services.

Problem Statement

Interoperability of message exchange is concerned with (data) format interoperability, protocol interoperability and most importantly the semantics of these messages. Interoperability is also concerned with higher level, but still application and domain independent protocols that describe how sequences of messages are interrelated, for instance, if they are defining transactions or sessions.

In the presence of standards, interoperability is often impeded by ambiguities and incomplete specifications. Here, additional constraints or new versions are used to unify and formalize the intent of a standard. The former approach is, e.g., adopted by WS-I. As an example of the latter, SOAP 1.2 excludes certain elements in the body that SOAP 1.1 missed to prohibit.

Regarding higher level protocols, standards are not commonly adopted or are still missing and best practices vary a lot. In particular, sessions are implemented using very different standards.

In the absence of standards or in the presence of conflicting standards, interoperability becomes a mediation challenge. Messages and protocols have to be transformed. In practice, tools like ESB (Enterprise Service Bus) or SCA (Service Component Architecture) runtime environments provide transformations between different messages and protocols, thereby aiming at hiding the use of different communication mechanisms. Nevertheless, interoperability remains a problem in practice because different vendors, which address varying business domains, adopt standards to a different extent. In such situations, best practices and respective patterns are needed.

Scope

The scope of this call is interoperability related to the exchange of single messages and to the exchange of a set of interrelated messages. It addresses standards for data formats, for message formats and for application and domain independent protocols together with the necessary mediation. It does not cover domain specific high level process protocols.

Contributions

Contributions to this call are expected in the area of standards; constraints for standards; and cross-standard mediation (which is needed when interacting services use different but overlapping standards).

We expect best practice patterns and constraints on patterns for mediation. Here, we expect documentation and analysis of the different existing solutions for a given problem/requirement, including a discussion of the lack of standards or the weaknesses of existing ones that gave rise to these solutions. The contribution is further expected to include a thorough analysis of the essential commonalities and variabilities of the involved patterns.

Baseline

The primary focus of this call is on web services. We ask for general solutions that can be applied to web services and WSDL.

For further information

http://www.nexof-ra.eu/service_interoperability

3.5 Declarative Authoring Language for User Interfaces

Contact

Nikolaos Tsouroulas (Telefonica) – nik@tid.es
José Manuel Cantera (Telefonica) – jmcf@tid.es

Overview

The aim of this topic is to identify a declarative language for user interface authoring to be adopted by NEXOF-RA.

Problem Statement

Traditional user interface development approaches are insufficient for supporting the new generation service front-ends. On the one hand they are oriented to specific devices or modes of interaction (normally a PC device). On the other hand they promote platform specific imperative development approaches which increase the effort, as UI designers cannot fully concentrate on the real application requirements. For example, Web & AJAX-based user interfaces are developed using HTML (which it is not device / modality independent) and scripting (which is both device dependent and imperative).

Going one step further, traditional UI platforms and toolkits lack formalisms to deal with Context-Aware service front-ends. For example, there is no a declarative mechanism to specify how an interface should adapt according to different Delivery Contexts. Instead, the developer needs to do it manually, using an ad-hoc costly approach which does not promote reuse or standardization.

New techniques based on abstract declarative languages are a better fit for this new scenario. A declarative language enables developers to concentrate on what the application needs to do, rather than the details of how that is to be achieved on a particular platform. The approach will be intrinsically extensible, and skilled programmers can add support for new mark-up and associated implementation classes as needed for new kinds of controls.

Scope

The scope of this topic focuses on declarative authoring languages suitable for specification of adaptable user interfaces for the ubiquitous web. The ultimate goal will be to achieve device independency and context sensitivity of the service front-end UI.

More abstract languages that go one step further and allow modality independent interface specifications will also be part of the investigation performed. The results will be used to define future strands of evolution of the selected language or languages.

Contributions

The expected result of this topic is to identify a Declarative Authoring language for User Interfaces that will form part of the NEXOF-RA building blocks. The contributions sought must take the form of:

- A specification of the language either developed within a project or available as an open standard.
- A roadmap for the evolution of Declarative Authoring Languages in general and the one selected in particular.

Baseline

A layered approach to the development of UI for services front-end is envisioned:

- Abstract UI (device independent)
- Concrete UI (device dependent)
- Physical realisation for specific devices

All of the layers (with the possible exception of the physical realization) can be represented in XML. Each layer embodies a model of behaviour at a progressively finer level of detail. The model view controller pattern should be used to cleanly separate the user interface, the dialog behaviour and the data it operates on.

Each layer can be considered as the result of a transformation, driven by different adaptation policies, of the layer immediately above it. Transformations will have access to the Context, which models user preferences, device and web browser capabilities and environmental conditions.

The physical realisation can be implemented via an optional mapping to a low level markup language or through compilation.

For further information

http://www.nexof-ra.eu/declarative_UI_authoring_languages (a white paper with further details is available on the web site)

3.6 Context Model and Universal APIs

Contact

José Manuel Cantera (Telefonica) – jmcf@tid.es
Nikolaos Tsouroulas (Telefonica) – nik@tid.es

Overview

This is an invitation to contribute with a Model and Universal APIs to the NEXOF-RA framework for context-awareness in service front-ends.

Problem Statement

The main challenge introduced by context-awareness is to come up with a flexible and unambiguous representation of the Context - a Context Model. Using this Context Model, applications will be able to adapt seamlessly to the target environment. Nonetheless, the heterogeneous nature of context-aware applications makes it impossible to have a universal, unique representation of the Context. A good compromise can be achieved if context models are able to manage a set of universal properties, useful for any application, in conjunction with application-specific custom properties.

Once a Context Model has been defined, the next critical step will be the adoption of a Universal API that supports such Context Model. Such API should be platform and language independent, shielding developers from the mechanisms used to gather distributed context information.

Scope

The following context aspects are under the scope of this invitation to contribute:

- User Profile: global preferences, interests, skills and social network
- Delivery Context: device, network, user agent and local settings (font size, volume, brightness ...)
- Environment: location and moment in time

The adopted Context Model must be extensible allowing other properties and aspects (standard or application-specific) to be included in the future.

The Universal API adopted must support the Context Model and in addition it should have the following functionalities:

- Platform independent
- Generic and extensible, allowing to work with different vocabularies of contextual properties
- It should support the notion of properties, aspects and components of the Context.
- It should provide not only query-response functions but also publish and subscribe mechanisms for notifying contextual changes to applications

Contributions

At an initial stage we invite contributions to the following work items:

- An standardised model for representing the fundamental aspects of the Context that are under scope
- A Universal Context API that meets at least the requirements under scope

These building blocks are considered the most urgent to be adopted, as they enable the minimal infrastructure for context-awareness. There are remaining building blocks that will be the subject of future invitations to contribute. Such invitations will be based on a roadmap that might also be part of the outcome of this process.

Baseline

The W3C's Delivery Context Ontology might be an starting point for a standard, minimal and universally-accepted Context Model. Such specification could be generalized and extended with additional modules capable of representing new general-purpose entities.

The DDR Simple API and DCCI are two W3C emerging standards for dealing with contextual information, thus they should be considered with regards to the Universal Context API.

For further information

http://www.nexof-ra.eu/context_model_universal_APIs (a white paper and further references are available on the web site)

3.7 Definition of Infrastructure Services

Contact

Mike Fisher (BT) – mike.fisher@bt.com

Overview

The NESSI vision of an open services ecosystem is underpinned by a universally accessible information and communications technology infrastructure which can provide execution environments for a very wide range of software components and systems. This infrastructure should also be exposed as services. This topic aims to identify an initial range of services that are attractive for both consumers and providers, and which will provide a focus for the specification of infrastructure service descriptions.

Problem Statement

Infrastructure as a service is a topic of increasing interest, both in offerings from commercial providers and in research projects. It is also an important constituent of the NESSI Open Framework. The problem addressed here is to identify a set of infrastructure services that will be attractive to users and feasible for providers to offer.

There is a wide range of possible infrastructure services, including network connectivity, storage, processing and execution environments for software components and virtual machines. Provision of software appliances for remote deployment, encapsulating complex functionality such as databases and application servers may be of interest. Desktop and application virtualisation may also be considered. Combinations of these into more integrated environments will also be of interest. In addition to generic functionality, specialised application requirements for features such as high bandwidth, high data throughput and low response time (latency) will be considered. Optimisations for specific workloads and application topologies may also be included.

There are existing infrastructure service offerings which will be taken into account – including some from Amazon, Google, IBM and others. These provide concrete examples of services that NEXOF-RA should be able to accommodate. In addition, a number of research projects have views on how infrastructure might be offered as services. This investigation topic will attempt to develop a unified view and to extract an initial set of infrastructure service characteristics which are expected to be common to sets of services. These will include appropriate service metrics and their units, which will be needed to support comparison of similar services from different providers as well as to ensure that the consumer and provider of a single service can have a consistent view of the quality of the service provided.

In exploring the limits of functionality that could be exposed as a service, it will be necessary to consider the viewpoint of both consumers (is the functionality useful and valuable?) and providers (can the service be deployed and managed

effectively?).

Scope

The scope of this topic is a set of services that expose low-level ICT functionality. These can be considered the baseline environment on which the rest of the NESSI Open Framework is built. It focuses on describing a set of infrastructure services that can be used to motivate and validate specifications (to be developed later) which will enable machine-readable descriptions of the functional and non-functional characteristics of infrastructure services. Formal specification (of, for example, service and deployment descriptors) is out of scope at present but, for example, requirements for common information models are in scope.

Contributions

- catalogue and describe a set of infrastructure services to underpin NEXOF
- identify common service characteristics to support automated service comparison
- define appropriate metrics for each service with well-defined measurement methodologies and units

The intention is that this output will provide a concrete foundation for future development and validation of service description languages and reference point (interface) specifications.

Baseline

The baseline for this topic is very open. It includes existing or anticipated market offerings of Infrastructure as a Service, plus views of contributors which may be explicit or implicit in the approach of contributing research projects. Flexible infrastructure, including server, storage and network automation and virtualization technologies is expected to be significant.

For further information

http://www.nexof-ra.eu/definition_infrastructure_services

3.8 Dynamic identity management for SOA

Contact

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

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

Overview

This call addresses the area of dynamic identity management for SOA. A potential problem is that users still have to manage multiple identities and credentials. This call concerns architectural schemes and patterns identity management, user interaction design, the federation of identity, and the access right framework based on semantic, in particular with regard user centric identity and high-level identity assurance.

Problem Statement

The concept of profiles can be developed into the more general idea of “identity management.” Users have several identities which can be used to perform different online transactions. For example, users could have an “anonymous identity” to surf general web sites, a “domestic identity” for accessing retail web sites, and an “office identity” for accessing corporate intranets. Decoupling identities from individuals can reduce the information collected about a single individual. However, identity management technologies are rather complex. So far, allowing easy definition of policies and simple awareness active personas has proven to be a difficult task.

In addition, identity federation can be defined as the set of agreements, standards and technologies that enable a group of service providers to recognise user identifiers and entitlements from other service providers within a federated domain. In a federated identity domain, agreements are established between Service Providers so that identities from different Service Providers specific identity domains are recognised across all domains. These agreements include policy and technology standards. A mapping is established between the different identifiers owned by the same client in different domains that links the associated identities. The federation of isolated identifier domains gives the client the illusion that there is a single identifier domain.

The user can still hold separate identifiers for each service provider. However, they do not necessarily need to know or possess them all. A single identifier and credential is sufficient for him to access all services in the federated domain. However, a potential problem is that users still have to manage multiple identities and credentials, even if they are not actively using all of them. In centralised user identity models, there exists a single identifier and credentials provider that are used by all service providers, either exclusively, or in addition to other identifier and credentials providers. From a user perspective, an increasing number of identifiers and credentials rapidly becomes unmanageable. A user-centric approach to identity management is a very promising way improving the user experience, and thereby the security of online service provision as a whole.

This call concerns architectural schemes and patterns identity management, user interaction design for identity management, expressing trustworthiness of identity management to users and privacy-enhancing identity management, logs tools required for forensic purposes (but not limited to).

Also, the call addresses solutions at the same time with regard to federation of identity including (but not limited to) methodologies and interfaces for managing multiple identities and credentials including delegation, separate identity management at each providers of services, synchronization with repositories of record.

In addition, the call looks for contributions concerning access right framework based on semantic, in particular with regard user centric identity and high-level identity assurance.

Scope

The scope is dynamic identity management for SOA.

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

Topics of particular interest include (but are not limited to) user interaction design for identity management, user centric identity, expressing trustworthiness of identity management to users, methodologies and interfaces for managing multiple identities including delegation, privacy-enhancing identity management, separate identity management at each providers of services, enterprises in cluster, risk management practices for issuing end-user credentials, synchronization with repositories of record, high-level identity assurance, and logs required for forensic purposes.

We envision also access rights framework based on semantics as an important step in the future of identity management search.

Recommendations: To propose identity management and federation identity to support e-service projects having realistic implementation plans and budgets.

Contributions

The contributions can take different shapes: 1) They can be around on how to attain identity management with federation of identity for SOA; 2) Architecture Patterns, schemes, components for identity management and federation of identity; 3) Concrete architectures for federation of identity including interface specifications; 4) How to extend a semantic approach to deal with management of access rights framework.

Baseline

The baseline is composed of web services standards (W3C, OASIS), J2EE, and the standards from the identity management and federation of identity forum.

For further information

http://www.nexof-ra.eu/identity_management

3.9 Privacy Management in SOA

Contact

Pascal Bisson (Thales) – pascal.bisson@thalesgroup.com
Daniel Gidoïn (Thales) – daniel.gidoïn@thalesgroup.com

Overview

This call addresses the area of privacy management for SOA. It focuses on the privacy in order to protect enterprise privacy and private infosphere, composed by a multitude of numeric shadow, in each internet webservice. Privacy issues addressed fall into two broad categories: users' data privacy and location privacy. Topics include: management, risk analysis, architecture, patterns, standards.

Problem Statement

Data privacy involves control over personal information contained on the devices and within services providers and in associated database(s). Location privacy involves control over the information regarding the individual's physical location and movement. Security controls that protect data privacy may not address location privacy and vice versa.

The call concerns threat and risk analysis methodology. When we talk about information privacy, we are usually talking about the privacy of sensitive data. This data can be the object of privacy threats. It is important to know what data is in the possession of the service providers, how many copies of the data exist, and where the data is stored.

Also, the call also looks for solutions with regard to personal vs. enterprise data protection. Personal privacy in pervasive computing is the process by which individuals selectively disclose personal information-such as e-mail address, shopping history, or location to organizations or other people. We suggest extending the concept of privacy to the typically personal data of the Enterprise.

Finally, the call looks for contributions concerning privacy standards and transaction anonymization. Next areas can be identified as relevant for pre-standardisation actions, namely the risk analysis methodology and an inventory of the threats, risks and privacy, data protection models, patterns and components.

Scope

The scope is Privacy Management for SOA. Privacy solutions answering these various problems are welcomed, including risk evaluation in the context of information privacy, risk being the aggregate of the likelihood that a threat will actually occur, the vulnerability to a threat if it did occur and the impact or consequences if a threat did happen. Concerning data protection - the focus is on protecting private and sensitive's enterprise data by regulating: how, when and for what purpose data can be collected, used and disclosed. Concerning the transaction anonymization- concrete solutions will be proposed (based on models, schemes, patterns) allowing the implementation of the anonymization

for SOA.

Contributions

The contributions can take different shapes: 1) They can be around on how to attain privacy management for SOA; 2) Privacy threat and risk analysis; 3) Architecture patterns, schemes, components for anonymization and privacy of enterprises and individuals discussing briefly the different nature of the solutions for both scenarios; 4) Concrete architectures for individual and enterprise data protection including interface specifications; 5) How to extend standards to deal with privacy and data protection models, and support the identified architectural patterns in better ways.

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/privacy_management

3.10 Scalable Approaches to Service Oriented Infrastructures

Contact

Ricardo Jimenez-Peris (UPM) – rjimenez@fi.upm.es

Overview

We invite for contributions on scalable solutions for Service Oriented Infrastructures (SOI). The emphasis is on clustering approaches that scale-out - that is, that by adding additional sites to a cluster, the cluster is able to cope with higher loads (e.g. higher number of concurrent clients). We look for scalability approaches to single and multi-tier architectures with special emphasis on stateful and/or transactional services. Scalability approaches in the context of new SOA paradigms is also sought such as Cloud Computing, SaaS, ...

Problem Statement

Current approaches to provide scalability for SOI rely on scale-up approaches, that is, buying a more powerful mainframe for dealing with higher loads. In here, solutions to provide scalability based on scale-out approaches are sought. Scale-out approaches rely on clusters that by growing with additional sites are able to cope with higher loads. We expect contributions on the scalability for systems with single and multiple tiers. Proposed approaches are expected to deal with consistency issues, consistency/cost trade-offs, autonomic aspects such as self-provisioning.

Contributions on systems modelling for predictable performance and scalability are also welcomed.

Contributions on new paradigms to scalable SOA are also called for, such as Cloud Computing, Software as a Service, Data Streaming, Complex Event Processing, Cluster Computing, Web farms, Edge Computing, ...

Scope

Contributions on scalability approaches to SOI are welcomed. Targeted servers include the typical tiers of multi-tier systems, web server, application server (e.g. J2EE) and databases. Scalability for servers in the area of web services, service composition, etc. are also in scope, as well as scalability approaches for new service paradigms as Cloud Computing, Software as a Service, Data Streaming, Complex Event Processing, Cluster Computing, Web farms, Edge Computing, ...

Hardware based scale-up solutions are out of scope.

Contributions

The contributions can take the form of: 1) Architectural patterns for attaining scalability in SOI; 2) Specification of SOI interfaces to enable scalability solutions; 3) Addressing scalability issues in SOI standards at the architectural level.

Baseline

The baseline is composed of web service stack (W3C, OASIS), and supporting multi-tier SOI standards such as the J2EE framework.

For further information

http://www.nexof-ra.eu/scalable_approaches_SOI

3.11 Highly Availability for Multi-Tier Architectures

Contact

Ricardo Jimenez-Peris (UPM) – rjimenez@fi.upm.es

Overview

We invite for contributions in the area of high availability for multi-tier architectures. The call focuses on the high availability in multi-tier systems for both single and multiple tiers (also federation of replicated tiers). Self-healing protocols are also in scope, with emphasis on how to attain performability using techniques such as online recovery. Contributions are also expected around consistency issues, cost awareness, and consistency/cost tradeoffs.

Problem Statement

With the increasing pervasiveness of eServices in all areas of life resulting in a high dependency on them. This results in a need for highly available services. High availability is attained by introducing redundancy in the underlining Service Oriented Infrastructures (SOI) typically by means of replication (either data or process replication). We look for contributions at how to introduce this redundancy in multi-tier systems to attain high service availability. We look for solutions for both single and multiple tiers including federation of replicated tiers.

One of the most important issues that should be dealt with in the approaches for high availability is data consistency in the advent of failures and concurrent access by multiple clients. High data consistency typically results in poor performance. Approaches dealing with relaxed level of consistency are therefore also of interest in which the consistency/performance tradeoff is identified. Another important issue that is called for is how to overcome real life problems when implementing replication such as how to enforce determinism, how to deal with sources of non-determinism, etc.

We also invite for contributions addressing performability. That is, how to deliver the same performance during failures and recoveries as in the failure-free execution. Non-intrusive solutions for fault-tolerance and recovery, such as online recovery, are therefore sought.

High availability in geographically distributed services is also an open topic. How to attain low latency for services in WANs (e.g. by resorting to edge computing approaches), how to interconnect data centres across WANs, how to guarantee consistency in the advent of network failures (e.g. partitions), etc.

Scope

The scope is high availability for SOI. High availability solutions for any kind of service (and underlying server) are welcomed, including typical tiers of multi-tier architectures such as web servers, application servers, and databases. The high availability of specific servers such as composition (e.g. BPEL) engines, service directories, etc. are also in scope. High availability for SOI deployed in

any kind of network is in scope: LAN, WAN, high bandwidth, mobile, ad hoc, ...
Hardware based solutions for availability (such as Tandem solutions, etc.) are out of scope.

Contributions

The contributions can take different shapes: 1) Architectural patterns for attaining high availability in single and multi-tier architectures; 2) Architectural patterns for high availability for particular service standards (with emphasis on web services and multi-tier middleware frameworks such as J2EE); 3) Concrete architectures for high availability including interface specifications for supporting high availability; 4) Standards and specification extensions to deal with high availability with better consistency, performance, performability, etc.

Baseline

The baseline consists of the web service stack (W3C, OASIS), J2EE framework, and the standards from the service availability forum.

For further information

http://www.nexof-ra.eu/highly_available_SOA

4 TOPICS FOR FUTURE CALLS

The following list, initially published on 21-Jul-2008, includes a series of topics which could potentially be the subject of the second (and subsequent) Invitation to Contribute. Actual topics will be confirmed at a later date.

- Core Service Framework Area
 - Federated Registries
 - Distributed Intelligent Deployment
 - Service Lifecycle Management
 - Event-Driven Architecture
 - Access to Current Assets / Legacy Integration
 - Service Creation
 - Semantic Interoperability
 - Semi Dynamic Service Composition and Service Goal Decomposition
 - Intelligent Service Discovery - Service Capability Extraction
- User Interaction Area
 - Context Binding Models and Policies
 - A Vocabulary (and Representation Format) for Expressing Service Front-End Resource's Metadata
 - Standard APIs for the Service Front-End Resource's Execution Environment
- Infrastructure Area
 - Service and deployment description languages for infrastructure services
 - Infrastructure service reference points (usage and management)
- Security Area
 - Dynamic Security in SOA and SOI
 - Multi-level Security in Interconnected Systems
- Quality of Service Area
 - Description of non Functional Aspects
 - SLA and QoS
 - Federated and Autonomic Management in SOA
 - Transactional Support for SOA