USP - Universidade de São Paulo
IME - Instituto de Matemática e Estatística

An overview of NEXOF

Victor Williams Stafusa da Silva

2011 - April - 04

What is NEXOF?

- NEXOF - NESSI Open Framework
- Organized by NESSI - Networked European Software and Services Initiative
- NEXOF will be an open service framework, ranging from the infrastructure up to the user interfaces.
- A infrastructure for building, deploying and running SOA applications in a cloud-computing internet environment.

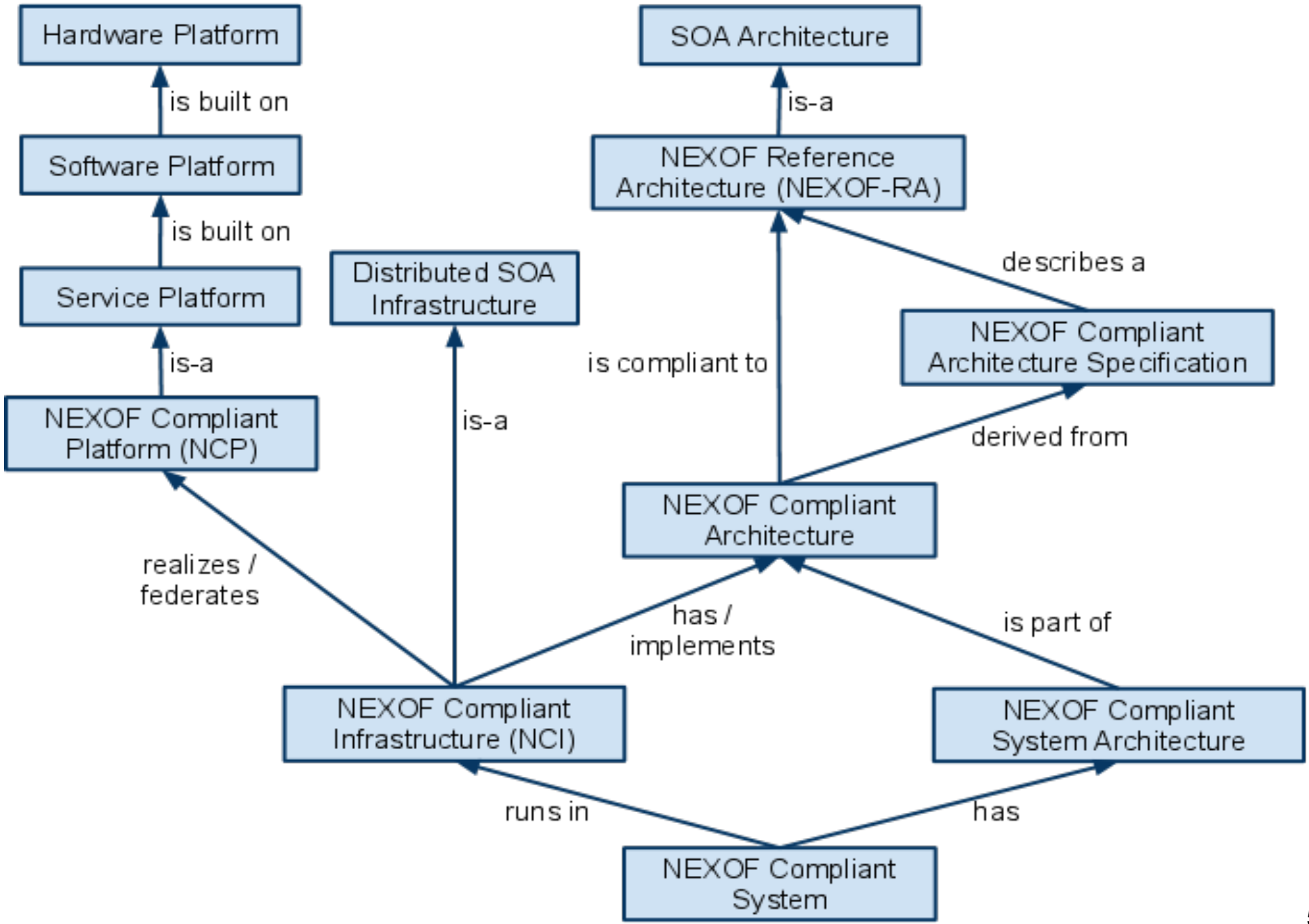
NEXOF Objectives

- Deliver a coherent open service framework, ranging from the infrastructure up to the user interfaces.
- Deliver applications and enable the creation of service-based ecosystems where service providers and third-parties easy collaborate.
- Be implemented into a broad range of application domains supporting any business size by all user communities using different technologies.
- Try to establish strategies and policies to speed up the dynamics of the services eco-system.
- Be platform independent.
- Be domain independent.
- Be open.

NEXOF Divisions

- NEXOF Reference Model - *"a (technology neutral and application domain independent) conceptual model that defines the types of entities and relationships that constitute service-based systems as well as key elements of their contexts."* [3] - Consists of a set of standarts
- NEXOF Reference Architecture - Consists of a set of patterns
- NEXOF Reference Implementation - Planned for future
- Conformance test - Planned for future

NEXOF Compliant Platform



NEXOF Compliant Platform

Characteristics of a NEXOF Compliant Infrastructure:

- In most cases, realized by a federated collection of subsystems.

Characteristics of a NEXOF Compliant Platform:

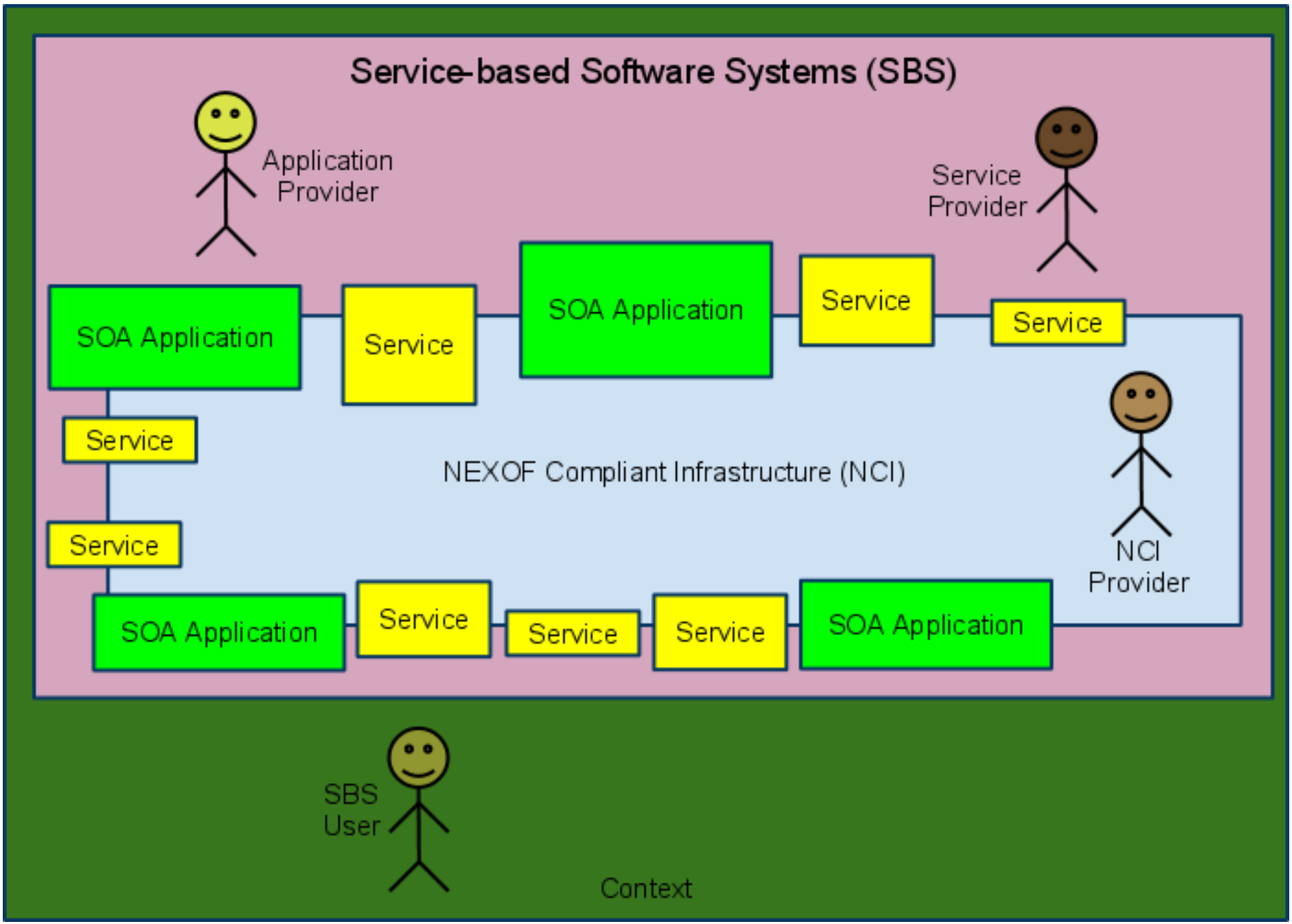
- Deployed in different interconnected computational nodes.
- Built by integrating a set of plugable platform-components.
- Co-operates with other NEXOF Compliant Platforms.
- Openness - interaction not only through API, but through deployment too.
- Extendable - allows the execution in it or publishes an API to made it public accessible, adding new features, rendering existing functionalities in new ways.

NEXOF Compliant Platform

Characteristics of a NEXOF Compliant Platform (cont.):

- Federation - access to all platforms through one, or through an access layer.
- Business level interoperability - loosely coupled business processes.
- Business-platform interoperability - It should be possible to deploy and perform business process in different platforms.
- Platform components interoperability - components from different vendors can be plugged-in or plugged-out.
- Platform services interoperability - interoperability of services provided in different platforms.

NEXOF Reference Model Context



NEXOF Reference Model - definitions

- Entity - A person or organization.
- Agent - *"Running software components that provide services".* [1]
- Actor - An entity or agent.
- Service - *"An action made by one entity (provider) that matches the request of another (requesting entity), according to the interpretation of the latter".* [1]
- Observable Action - A set of observable actions characterises a service. i.e. it is what a service serves. Observable actions can be ordered, each one having predecessor and sucessor observable actions.

NEXOF Reference Model - definitions

- Software Service - *"A service in which software agents, a requester and a provider agent, mediate the interaction between requester entity and provider entity."* [1]

Services can be human-to-human, human-to-agent, agent-to-human or agent-to-agent. Software Services are always agent-to-agent.

The Nine concerns

The requeriments of a NEXOF Compliant Infrastructure are: [1]

1. Service - The infrastructure has to support the creation and execution of services.

2. Messaging - The infrastructure has to enable service interaction by means of messages.

3. Discovery - The infrastructure has to support the discovery of (provided and required) services.

The Nine concerns

The requeriments of a NEXOF Compliant Infrastructure are: [1]

4. Composition - The infrastructure has to support the composition of services.

5. Analysis - The infrastructure has to provide tools to analyse the services it supports with respect to the business goals.

6. Presentation - The infrastructure has to support presentation interfaces to enable human users to interact with services and applications.

The Nine concerns

The requeriments of a NEXOF Compliant Infrastructure are: [1]

7. Management - The infrastructure has to support the management and monitoring of its core components and of services and applications it supports.

8. Security - The infrastructure has to provide mechanisms and policies to accomplish different level of security.

9. Resources - The infrastructure requires computational resources to be executed.

The Nine concerns: 1 - Service

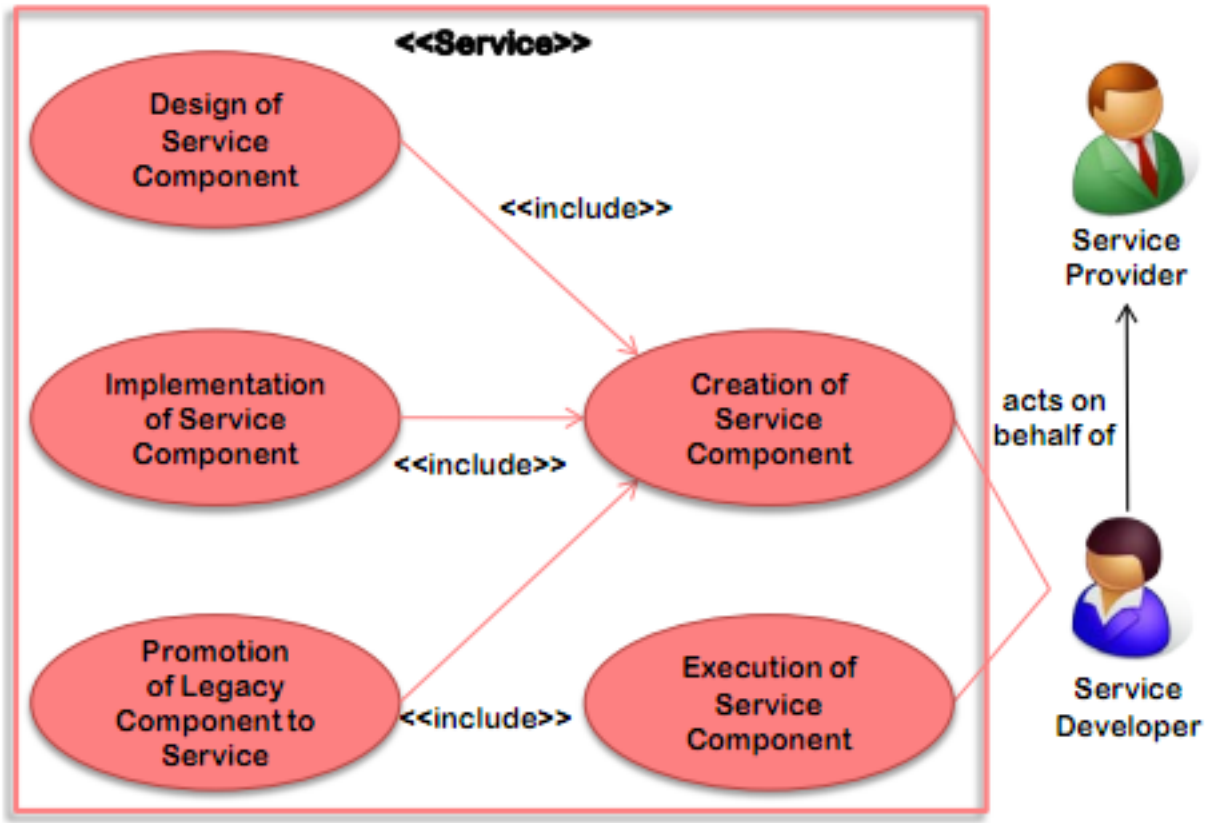


Figure from [1] - See bibliographic references

The Nine concerns: 1 - Service

Requirement: "The infrastructure has to support the creation and execution of services".

Reason: To capture all the functionalities to support creation and execution of services. This addresses the underlying building blocks of SOA.

Not concerned to service invocation or composition (these are addressed in other requirements).

Creation functionality subdivides in three: design, implementation, and wrapping legacy software components to promote them as services.

The Nine concerns: 1 - Service

- Service description - Represents information that describes a service. Comprises information of several facets. It is used in the creation of a service.
- Service facet - A particular piece of information that describes a particular aspect of a service, may be in natural language plain text, XML, RDF, OWL, etc.
- Connector service component - "*wraps legacy application components in order to promote them as services*". [1]
- Service component specification - Constraints the implementation of the service (i.e. says what the service does).

The Nine concerns: 1 - Service

The design of a service creates a service description, which should contains at least a subset of:

- The functional description of the service to define what the service does.
- The interface of the service to define how a service will be accessed.
- The quality of service (QoS).
- The definition of the transactional support.
- Possibly other things.

The implementation should expose a remote interface of the component and allow the generation of a client stub.

The Nine concerns: 2 - Message

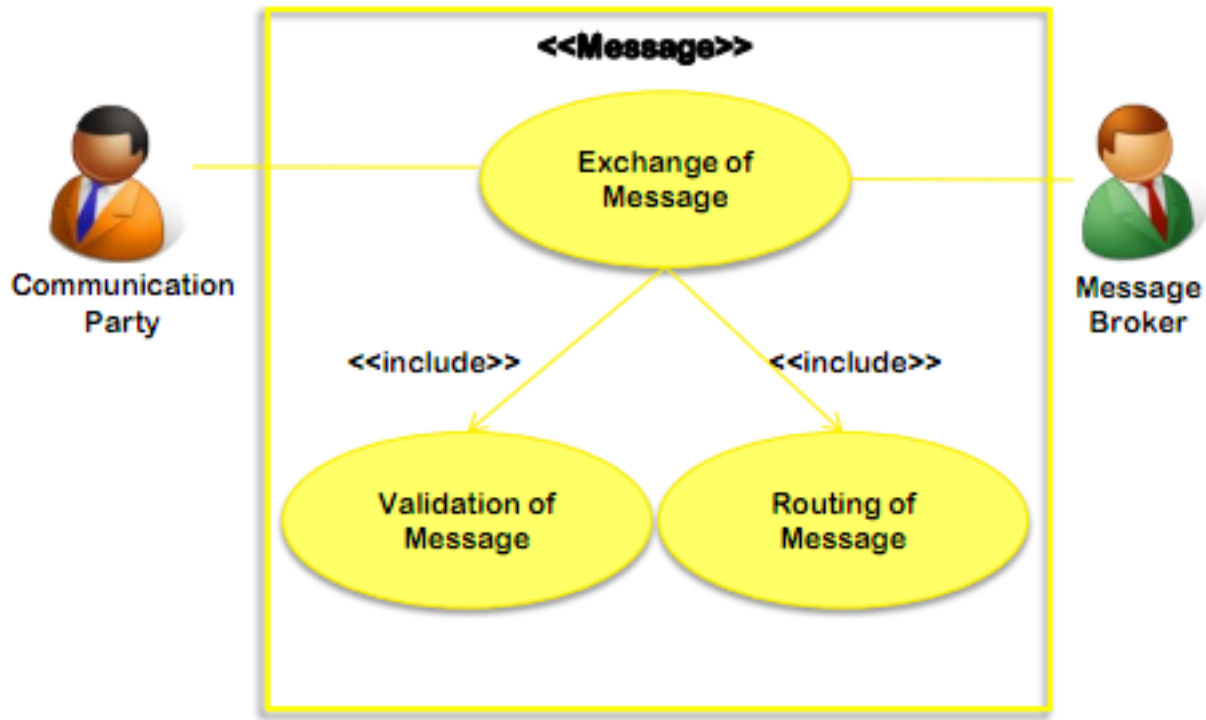


Figure from [1] - See bibliographic references

The Nine concerns: 2 - Message

Requirement: "The infrastructure has to enable service interaction by means of messages".

Reason: To address the communication capability that allows applications or services to interact with other services. This enables services to communicate and interact.

Not concerned with intelligent routing that alter process flows based on business and policies conditions. This should be addressed in composition.

Message exchange includes message validation and routing.

The Nine concerns: 2 - Message

- Message broker - Mediates the communication between communication parties.

- The connectivity is connection independent - The service requestor and service provider communication parties should be loosely-coupled. usually (but not always) through asynchonous messaging.

Providing the connectivity is not enough. Data formats need to be manipulated by different programs, so message adapting and mapping is frequently needed. This way, each program receives and outputs the message in the native format and the message layer transforms one format in another.

The Nine concerns: 2 - Message

- Message - A basic unit of communication exchanged between a consumer and a provider.
- Message type - Defines the message structure, how it should be interpreted.
- Message metadata - Information associated with the message used to process and/or deliver it.
- Endpoint - Entry point of a given service. Uniquely identifies the service and contains the address to send and receive messages.

The Nine concerns: 2 - Message

Interation model - Defines the pattern for exchange of messages between communications parties.

- Should define if the message is synchronous or asynchronous.
- Should define if it is supported by request/response, one-way, peer-to-peer.

The arriving message should be validated against validation properties, and a rejection notification should be produced if invalid. The validation should check if the arriving message is a well-formed document.

The Nine concerns: 2 - Message

The forwarding message is sent by the communication party to the broker. If the metadata does not specifies the endpoint, the broken selects it. If necessary, the message is adaptated and finally the broker dispatches the message.

Adaptations can be needed to:

- To convert between different protocols.
- To convert between data formats.
- To enhance the content of the message to match the recipient agent expectations.
- To remove not required information.
- To split, agreggate and sequence the messages.

The Nine concerns: 3 - Discovery

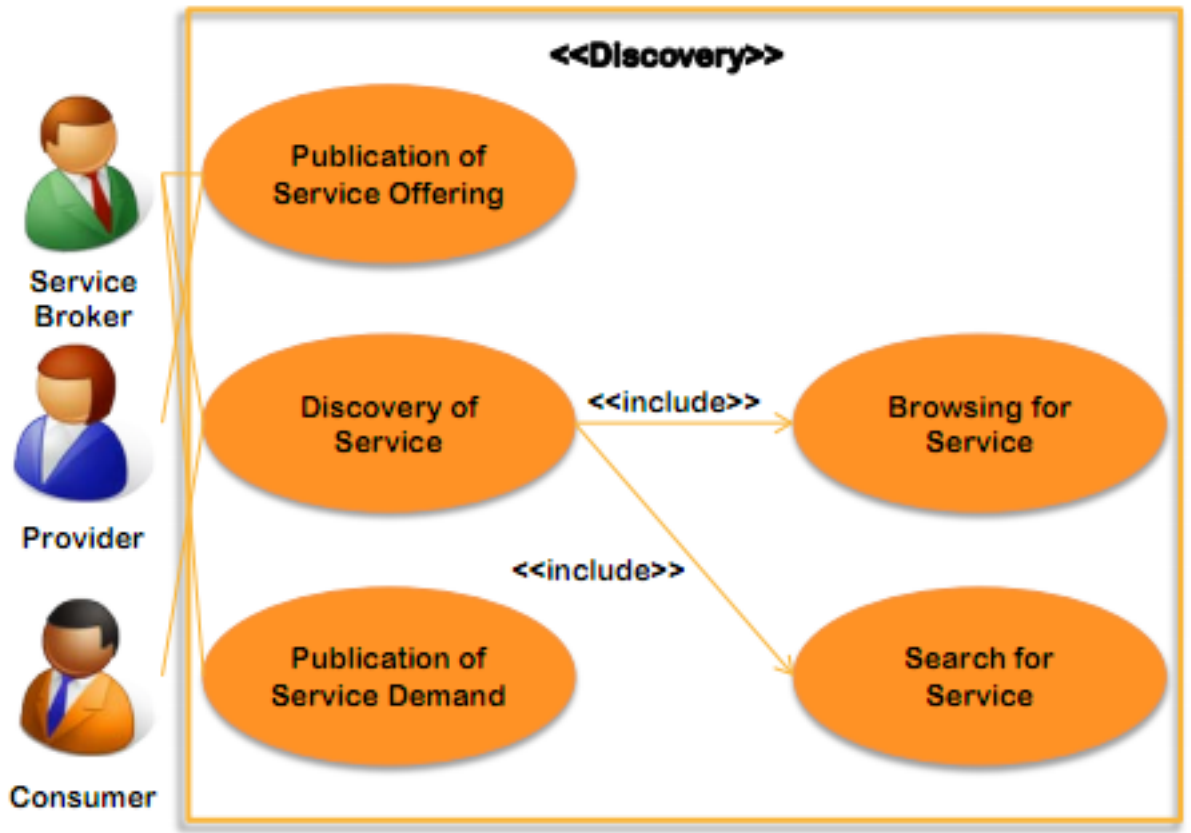


Figure from [1] - See bibliographic references

The Nine concerns: 3 - Discovery

Requirement: "The infrastructure has to support the discovery of (provided and required) services".

Reason: To cover all functionalities that are needed to support the discovery of services and making them effective. This provides the bases for reuse.

Includes service publications and registries. To encourage software reuse to avoid developers to duplicate already existing services, should include easy-to-use browse facilities and notifications in service changes.

Discovery occurs through browsing or search. In search, a service ranking is needed in the service candates result list.

The Nine concerns: 3 - Discovery

Registries should hold associated attributes for each service, including possibly business policies. E.g. criptography or installation attributes.

Discovery can be:

- In design-time, where a developer query the service catalog to find one or more services to incorporate in processes.
- In run-time, where the system queries the service catalog to find services.

Discovery policies can be:

- Pull policy - Consumers search the registry.
- Push policy - Subscribed consumers are notified of service publication events.

The Nine concerns: 3 - Discovery

- A search in the registry contains a search criteria.
- Each search criterion matches one service facet of the service description.
- Search criteria may be a search query, a subscription criteria or a catalogue schema criteria.
- The search results are a list of candidate services with their respective service descriptions.
- The service descriptions are published in the registry.
- A catalogue schema consists of categories.
- The published information should contain details of the service to enable users to invoke it (e.g. endpoint address, transport protocol, data format and what the service does).
- A search may be refined resulting in a multi-step search.

The Nine concerns: 3 - Discovery

- Publication of service offering - Publishes a service based on the service description to make in searchable and browseable by users. Also able to update or delete the information.
- Publication of service demand - Allows users to publish their demand (i.e. subscribe), for push-mode discovery. Users receives events of publications and updates that matches their subscription criterias.

The Nine concerns: 4 - Composition

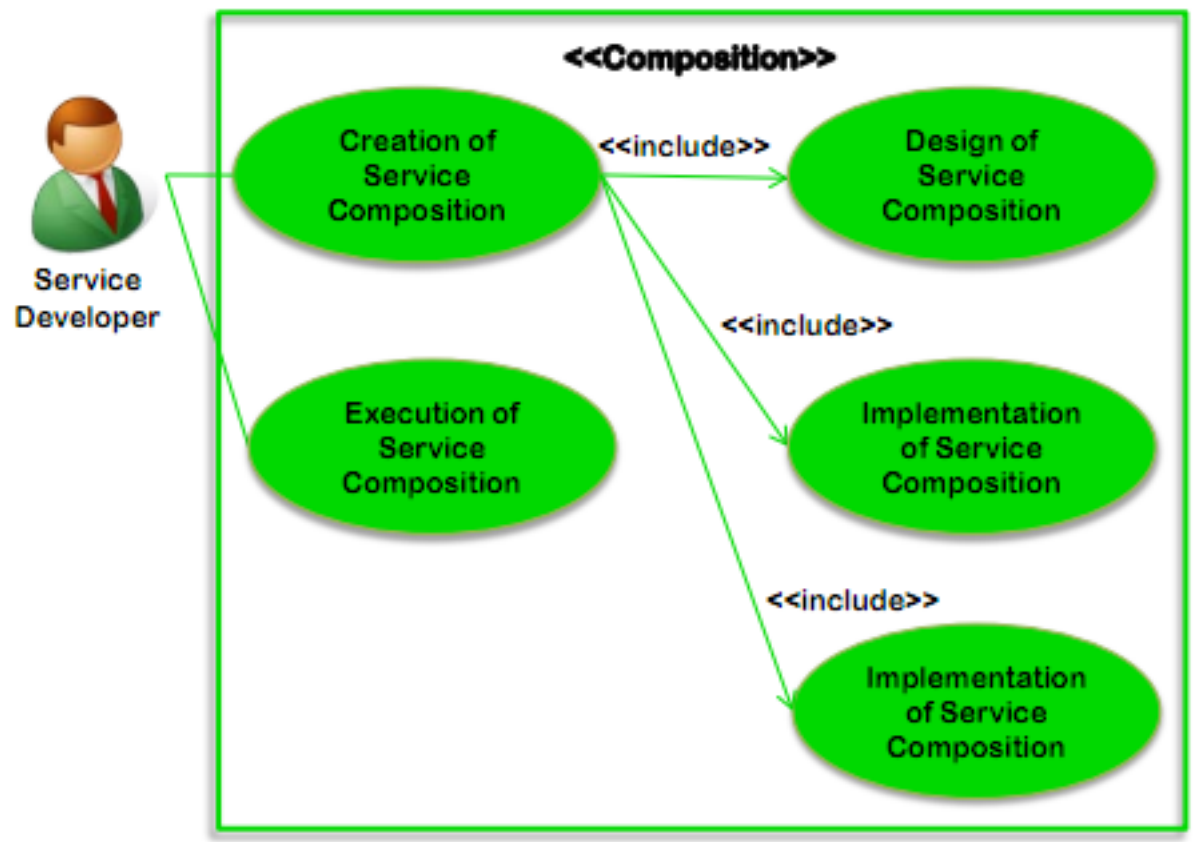


Figure from [1] - See bibliographic references

The Nine concerns: 4 - Composition

Requirement: "The infrastructure has to support the composition of services".

Reason: To address the creation and execution of software processes. This links services into business processes.

- The result of a composition is called process.
- A composition may be (optionally) published as a composite service with a service description, and then be discovery or itself used in another composition.
- Covers both, the support at design-time (orchestration, choreography) and at run-time (static/dynamic composition).

The Nine concerns: 4 - Composition

- Addresses the ability to support dynamic process reconfigurations.
- Includes design, implementation and execution of compositions.
- Sometimes, support dynamic compositions.
- Also concerned with human workflows, which can lasts days, and can lead the requeriment to hold transaction states for long periods.
- From a consumer point of view, it is not visible if a given service is a composite service or not.
- To compose a service based on existing ones, discovery may be needed.
- The implementation can be detailed into transformation in executable code, binding and testing.

The Nine concerns: 4 - Composition

- The execution handles service invocations and data transformations.
- The execution can at runtime be adapted to some events, like a publication of a new service.
- The execution may monitor the data to adapt its workflow based in some criteria.
- The execution may be interrupted and resumed.

The Nine concerns: 5 - Analysis

Requirement: "The infrastructure has to provide tools to analyse the services it supports with respect to the business goals".

Reason: To deal with business performance, need to track, record and manage. This enables continuous process improvement.

Currently, mostly left out of the Reference Model and just barely and superficially discussed and defined, due to time and priorisation constraints.

To deal with process metrics, business events, policy requirements, detect out-of line situations, respond and mitigate compliance failures and associated penalties.

The Nine concerns: 6 - Presentation

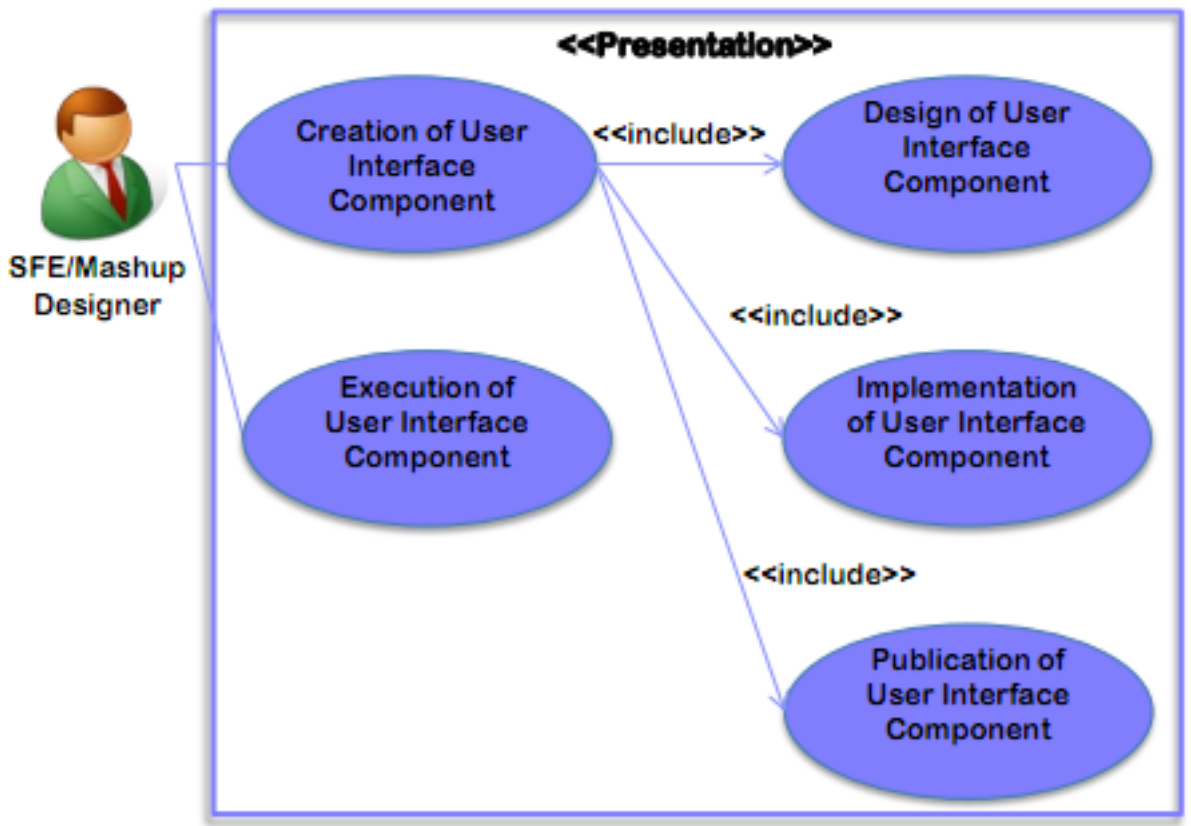


Figure from [1] - See bibliographic references

The Nine concerns: 6 - Presentation

Requirement: "The infrastructure has to support presentation interfaces to enable human users to interact with services and applications".

Reason: To enable human users to interact and make use of the functionality provided by the overall platform. This incorporates people into the SOA equation.

Includes GUI and APIs for creation, customization and execution of these GUIs.

Should made possible to design a composite end-user application by just assembling together a set of uniform building blocks, with no coding at all.

The Nine concerns: 6 - Presentation

- Service Frontend Resource (SFER) - Can be a gadget, a gadget group, or a workspace.
- Description - a SFER consisting of several resource properties with a certain value.
- Gadget group - A group of gadgets connected via channels. Are part of the workspace.

Workspaces are owned by users and provides highly personalized and simplified environments for building gadget groups.

A SFER can be created, published in a catalogue and executed.

The Nine concerns: 6 - Presentation

- User interfaces components can be designed, implemented and published.
- A composition of user interfaces components is a mashup.
- A simple SFER just implements a GUI and its logic.
- A mashup search and selects SFERs.
- A mashup is composed from a discovery process of other SFERs, and then added to the workspace.

The Nine concerns: 7 - Management

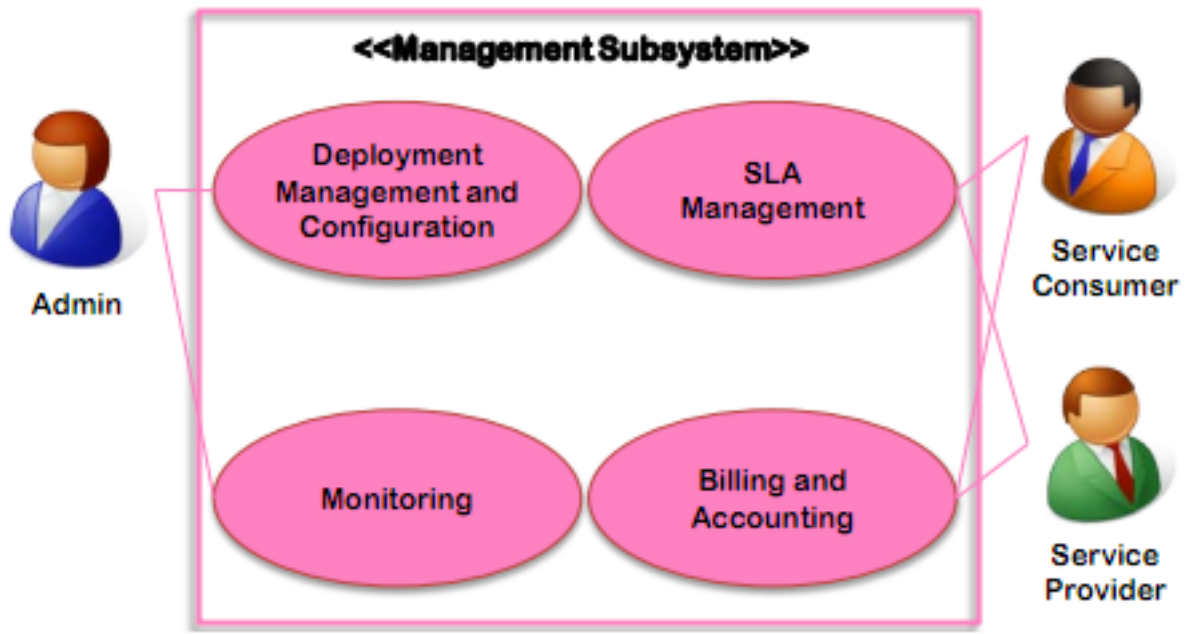


Figure from [1] - See bibliographic references

The Nine concerns: 7 - Management

Requirement: "The infrastructure has to support the management and monitoring of its core components and of services and applications it supports".

Reason: To allow management and monitoring service, processes and usage of platform. This addresses service levels and governance.

Services can be decomposed in smaller components, software and hardware. Services can be deployed and configured in many different ways with different QoS and costs, and management decides how to do this.

Aims to guarante SLAs and minimize costs.

The Nine concerns: 7 - Management

- To the management of a system be possible (including its service and resources), monitoring is needed.
- The monitoring observes and quantifies the behaviour of the system.
- Monitoring drives management decisions and compiles billing and accounting informations.
- Management (including self-management) needs an interface to modify, deploy, undeploy, start, stop and configure services and service metadatas, possibly in heterogeneous NCIs. That interface should be implemented by a management entity.
- Monitoring needs facilities to monitor QoS aspects (like response time) and may include logging, event tracking and SLA tracking.

The Nine concerns: 7 - Management

- Monitoring requires reporting facilities, may be in real-time or off-line.
- Fault avoidance and fault tolerance mechanisms must be in place.
- Fail-over support.
- Re-routing around failures.
- A service offering always has a Service Level Agreement (SLA).
- A SLA consists of Service Level Objectives (SLO).

The Nine concerns: 7 - Management

A SLO may specify one of:

- A requeriment (goal) of the service (like service time, capacity, recoverability, avaiability, security...)
- A constraint of goal
- A configuration parameter of the service behaviour
- A rule (policy) that handles events. A policy can be a management or a monitoring policy.

SLOs are defined by metrics (for usage, performance, etc).

SLA Management - Creates, describes, negotiates and monitors SLAs based on metrics. The SLAs are translated to a form interpretable by software systems.

The Nine concerns: 8 - Security

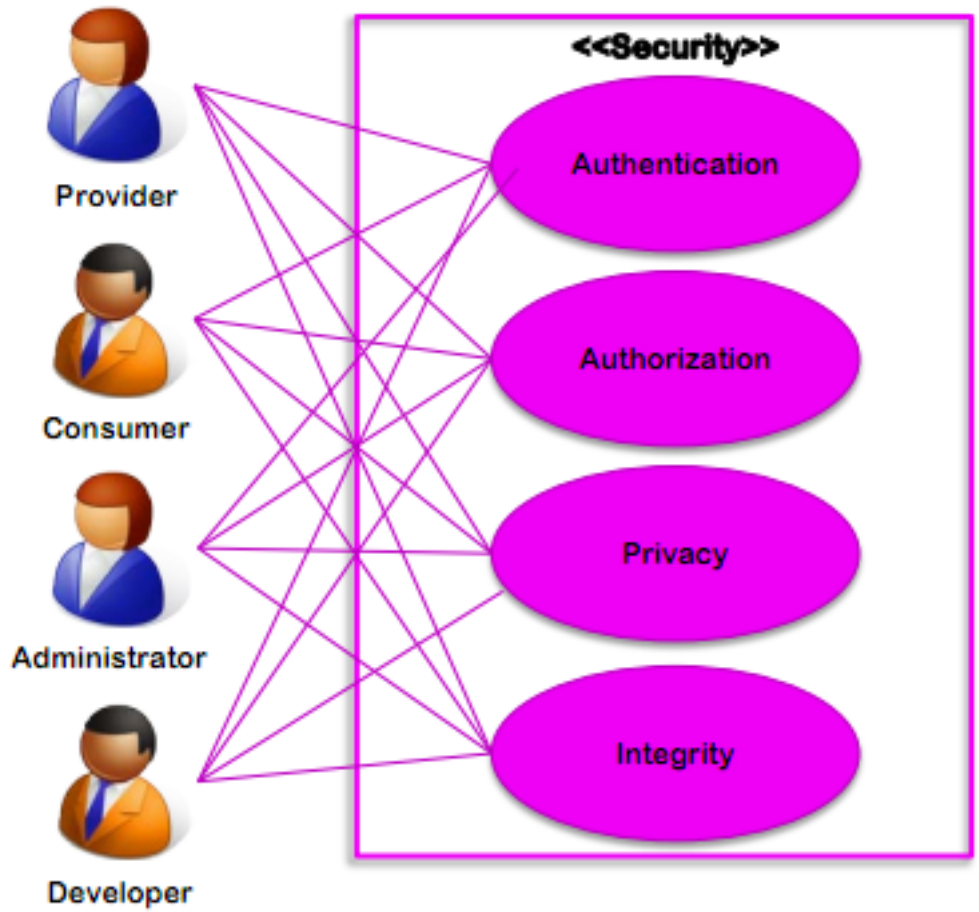


Figure from [1] - See bibliographic references

The Nine concerns: 8 - Security

Requirement: "The infrastructure has to provide mechanisms and policies to accomplish different level of security".

Reason: To addresses the support to manage the security of the system. This makes SOA reliable.

Includes authentication, authorization and encryption. Handles DoS attacks and tampering. Concerned about privacy and message integrity too.

Policy-based security service approach provides the relevant security concerns to NEXOF.

The Nine concerns: 8 - Security

- Identity data - A credential consisting of a set of authentication protocols, i.e. login.
- Authentication data - Data used to authenticate the requesting-entity, normally consisting of passwords, digital certificates and signatures, etc.
- Authentication result - Informs if the access was authenticated or not, computed from the identity data.
- Access rights - Informs what the user can access, computed from the authorization data.

Authorization may rely on digital signatures, roles and timestamps and provides confidentiality and privacy.

The Nine concerns: 8 - Security

- Integrity verifies if the message was not unexpectedly modified, by checking the digital signature and/or timestamps.
- Authentication may be provided through "something you know" (passwords), "something you hold" (tokens and PINs), or "something you are" (biometrics).
- Privacy handle user-tracking, signature, accountability, encryption and legislation aspects.

The Nine concerns: 9 - Resources

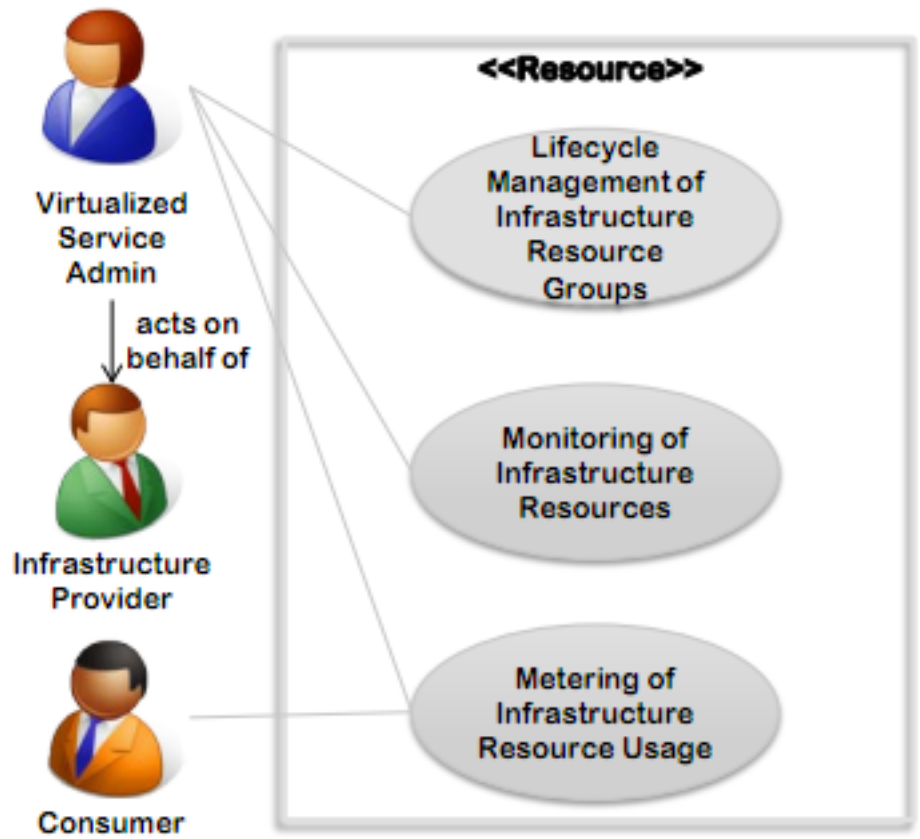


Figure from [1] - See bibliographic references

The Nine concerns: 9 - Resources

Requirement: "The infrastructure requires computational resources to be executed".

Reason: To deals with the infrastructure of NCIs. This makes SOA effective.

Includes computational resources like virtualization, storage, network resources, CPU, RAM, etc. Infrastructure components, including process engines, are not considered.

The Nine concerns: 9 - Resources

The virtualized service manager (VSM) is responsible for:

- Deploying, undeploying, starting, stoping and suspending infrastructure resources.
- Interacting with service providers.
- Negotiate pricing.
- Monitoring the environment and the resource.
- Gathering billing information from monitoring.
- Receiving service manifests.
- Ensuring SLAs compliances.
- Adjusting the service application capacities through elasticity rules.
- Provisioning, migrating and removing resources.
- Release and reserve capacity.

The Nine concerns: 9 - Resources

- A service manifest is a request for deploying a service in a infrastructure provider and includes all the information needed to manage the service lifecycle.
- A virtualized service repository (VSR) keeps virtualized services and their attributes (SLA, business rules, service definition, elasticity rules).
- The service management interface (SMI) is provided by the VSR to the VSM.

The Nine concerns: 9 - Resources

- A virtual execution environment (VEE) is a generic virtual resource, abstracting away physical characteristics.
- A virtual execution environment host (VEEH) is the physical machine and the virtualization software that hosts and controls VEEs.
- A virtual execution environment manager (VEEM) places VEEs into VEEHs optimally and moves VEEs between VEEHs.
- The VEE host interface (VHI) is a generic API between VEEH and VEEM. This abstracts away specific virtualization technologies.
- The VEE management interface (VMI) is the interface used by the VSM to interact with the VEEM.

Overview of the NEXOF-RA

- Guidelines and principles - Defines modularity, Layering, Partitioning, Extensability and Reuse. Used to guide the evolution of NEXOF-RA, specifying how to create a design pattern to NEXOF and what types of patterns relationships are expected to exist (Extends, IsPartOf, CanBeSolved, Uses, MayUse, etc).

- Reference architectural models - Describes the entities and concerns of the NEXOF, mainly the Reference Model and a glossary.

- Standards catalogue - Maps external entities patterns and standards (like OASIS, W3C, JCP, OSGi, OMG) to NEXOF-RA patterns.

Overview of the NEXOF-RA

- Components catalogue - *"Groups both, abstract descriptions of components as well as product- or software-based components. Each description refers to the standards it implements, the concepts it addresses as well as its behavioural characteristics".* [1]

- System of Patterns - Comprises a set of (currently 60 and counting) patterns specified by the NEXOF-RA. The specifications of these patterns are the main goal of NEXOF-RA. These patterns provides solutions to implement the functionalities related to the nine concerns in NEXOF Compliant Infrastructures and NEXOF Compliant Systems. Currently the majority of them is still under development.

Patterns of the NEXOF-RA

The NEXOF-RA Patterns are classified in three domains. A few patterns do appears in more than one of these domains. These domains are:

- Internet of Services domain - "*Provides an architectural design of a world-wide internet-based platform to enable anyone to deliver, consume and prosume services at global scale*". [5]
- Cloud domain - Provides Cloud Service Providers with computational resources and execution environments to enable its customers to deliver their own business services on the Internet.
- Enteprise SOA domain - *"Describes the characteristics of a SOA infrastructure specifically designed to support enterprise service-based applications"*. [5]

NEXOF-RA - Internet of Services domain

The Internet of Services domain patterns are:

- Assisted Composition Designer (4)
- Authorization (8)
- Dynamic Binding of Services during Composition (4)
- Federated Distributed Service Bus (2)
- Infrastructure as a Service (IaaS) (7, 9)
- Internet of Service (1, 2, 3, 4, 6, 7, 8, 9)
- Mashup as a Service (MaaS) (6)
- Multi-phase Discovery (3)
- Semantic Annotation Composition (4)
- Semantic Based Federated Registry (3)
- Service Discovery (3)
- Service Matchmaking and Ranking (3)
- Template-based Discovery (3)
- Trust Based Model Registry (1, 3)

NEXOF-RA - Cloud domain

The cloud domain patterns are:

- Authorization (8)
- Cloud migration enabled by OSGi – step one (1, 4, 7, 9)
- Cloudified Application Servers (1, 7)
- Elastic Reliable Filesystem (7)
- Infrastructure as a Service (IaaS) (7, 9)
- Map Reduce (1, 7)
- Not only SQL (NoSQL) Storage (7)
- Platform as a Service (PaaS) (1, 7, 8, 9)

NEXOF-RA - Enterprise SOA domain

The enterprise SOA domain patterns are:

- Active Replication (none)
- Assisted Composition Designer (4)
- Authorization (8)
- Black-Box Database Replication (none)
- Cloud migration enabled by OSGi – step one (1)
- Data Mediation (2)
- Designer and Runtime Tools for E-SOA (1, 4, 6, 7)
- Distributed ESB in ESOA (2)
- Dynamic Binding of Services during composition (4)
- Enterprise SOA (1, 2, 3, 4, 6, 7, 8, 9)
- Front-End in E-SOA (3, 6, 7)
- Generic Group communication (none)
- Gray-Box Database Replication (none)

NEXOF-RA - Enterprise SOA domain

The enterprise SOA domain patterns are (cont.):

- Horizontal Replication with Replication Awareness (none)
- Infrastructure as a Service (IaaS) (7, 9)
- Log-Mining Writeset Extraction (none)
- Models Manager (4)
- Monitoring in Enterprise SOA (7)
- Multicast-Based Replica Discover (3)
- Multi-phase Discover (3)
- Multi-Tenancy Enabler (9)
- Multi-Tier Transactional Service Runtime (1)
- Non-Repudiation (8)
- OSGi-based SCA-Container (1, 4)
- Reflective Database Replication (none)
- Registry-Based Replica Discovery (3)

NEXOF-RA - Enterprise SOA domain

The enterprise SOA domain patterns are (cont.):

- Security in Enterprise SOA (8)
- Service Discovery (3)
- Service Matchmaking and Ranking (3)
- Session Replication with Multi-Tier Coordination (none)
- Template-based Discovery (3)
- Transparent Replication Proxy (none)
- Trigger Writeset Extraction (none)
- Trusted Timestamping (8)
- Vertical Replication (none)
- Virtual ESB (2)
- Virtualization of Computational Resources in E-SOA (7, 9)
- White-Box Database Replication (none)
- Writeset Extraction Based on Extended Interface (none)

Other NESSI projects related to NEXOF

- EzWeb - For mashups, concern 6.
- Master - For management and security, concerns 7 and 8.
- Reservoir - For infrastructure, virtualization and cloud computing, concerns 7 and 9.
- SOA4All - Mainly for infrasctructure, but for services and messaging too, concerns 1 and 2.
- SLA@SOI - For SLA management, concern 7.

Other NESSI projects - EzWeb

"The EzWeb project delivers an open web mashup platform for the next-generation Internet of Services. It delivers the main components of the front-end web access platform to NEXOF. Its open source approach is intended to facilitate liaison with other open source communities and allows for the incorporation of additional partners. A fully functional beta version is already available."
(quoted from http://en.wikipedia.org/wiki/Networked_European_Software_and_Services_Initiative/)

Other NESSI projects - MASTER

MASTER - Managing Assurance Security and Trust for sERvices

"MASTER will provide methodologies and infrastructure that facilitate monitoring, enforcement, and auditing of security compliance, especially where highly dynamic service oriented architectures are used to support business process enactment in single, multi-domain, and iterated contexts"
(quoted from http://www.master-fp7.eu/)

Other NESSI projects - Reservoir

"The goal of the Reservoir project is to introduce a powerful ICT infrastructure for the reliable and effective delivery of services as utilities. This infrastructure will support the setup and deployment of services on demand, at competitive costs, across disparate administrative domains, while assuring quality of service. The architecture and reference implementation for a service-oriented infrastructure will be built on open standards and new technologies to provide a scalable, flexible and dependable framework for delivering services as utilities. Based on the emerging “cloud computing” model, Reservoir will use virtualization techniques to allow physical resources from different service providers across the cloud to essentially be pooled while still guaranteeing security and quality of service to customers".

(quoted from http://en.wikipedia.org/wiki/Networked_European_Software_and_Services_Initiative)

Other NESSI projects - SOA4All

"SOA4All aims at realizing a world where billions of parties are exposing and consuming services via advanced Web technology. The main objective of the project is to provide a comprehensive framework and infrastructure that integrates complementary and evolutionary technical advances into a coherent and domain-independent service delivery platform"
(quoted from http://en.wikipedia.org/wiki/SOA4All/)

Other NESSI projects - SLA@SOI

"The SLA@SOI project delivers the following elements to NEXOF: an e-contracting platform between service consumers and providers, a framework for mapping, planning and coordination within multiple levels in an organizational/IT structure and the access and provisioning layer for SLA-aware infrastructure."
(quoted from http://en.wikipedia.org/wiki/Networked_European_Software_and_Services_Initiative/)

Questions?

???

Bibliographic references

[1] http://www.nexof-ra.eu/sites/default/files/D6.3_v1.0.pdf

[2] http://www.nexof-ra.eu/sites/default/files/NEXOF_Roadmap_Public_v1.0.pdf

[3] http://fif.kr/fisc2009/doc/swkang.pdf

[4] http://www.soasymposium.com/soasymposium2009/pdf/Frederic_Gittler_An_Approach_for_Service_based_System_Architectures.pdf

[5] http://www.nexof-ra.eu/sites/default/files/D7.5b%20RA%20Specification%20V0.4a%20%28Pattern%20Compass%29.pdf

[6] http://sla-at-soi.eu/tag/nexof/

[7] http://www.nexof-ra.eu/

[8] http://www.nessi-europe.com/?Page=open_innov

Bibliographic references

[9] http://www.nexof-ra.eu/?q=newsroom

[10] http://en.wikipedia.org/wiki/Networked_European_Software_and_Services_Initiative

[11] http://ezweb.morfeo-project.org/

[12] http://www.nexof-ra.eu/sites/default/files/D7.5b%20RA%20Specification%20V0.1%20%28Executive%20Summary%29.pdf

[13] http://www.nexof-ra.eu/sites/default/files/D7.5b%20RA%20Specification%20V0.4a%20%28Pattern%20Compass%29.pdf

[14] http://www.nexof-ra.eu/?q=node/757

[15] http://www.nexof-ra.eu/?q=node/695

[16] http://www.reservoir-fp7.eu/

Bibliographic references

[17] http://www.nexof-ra.eu/?q=rep/term/141

[18] http://www.nessi-europe.eu/default.aspx?page=home

[19] http://www.lero.ie/project/nexof

[20] http://cordis.europa.eu/fetch?CALLER=FP7_PROJ_EN&ACTION=D&DOC=2&CAT=PROJ&QUERY=012eb40587f4:8c6d:5d1867c2&RCN=85366

[21] http://www.ist-world.org/ProjectDetails.aspx?ProjectId=4c08f0a34d3444418ff6a6571ea0864b

[22] http://www.nexof-ra.eu/sites/default/files/D7.2c%20Definition%20of%20an%20architectural%20framework%20and%20principles%200.9.pdf

[23] http://en.wikipedia.org/wiki/SOA4All

[24] http://sla-at-soi.eu/2011/04/sla-enabled-infrastructure-scheduling/

Bibliographic references

[25] http://www.nexof-ra.eu/?q=node/529

[26] http://www.soa4all.eu/

[27] http://www.master-fp7.eu