Analysis of NEXOF-RA RA Model v 2.0

August 24, 2009

Sungwon Kang

sungwon.kang@kaist.ac.kr

KAIST

TOC

I. NEXOF, NEXOF-RA and NEXOF-RA Model

- **"Networked European Software & Services Initiative"**
- **A *European Technology Platform* dedicated to Software and Services**
  - ETP works by the process "*Vision* –> *Strategic Research Agenda (SRA)* –> *Implements its SRA*"
- Launched in Sept 2005 by industry
- **Community**: 24 partners, > 400 members
  - Industries, SMEs, Academia and users
- **Activities**
  - Structuring research
  - Defining and fostering the creation of
    NEXOF(NESSI Open Service Framework)

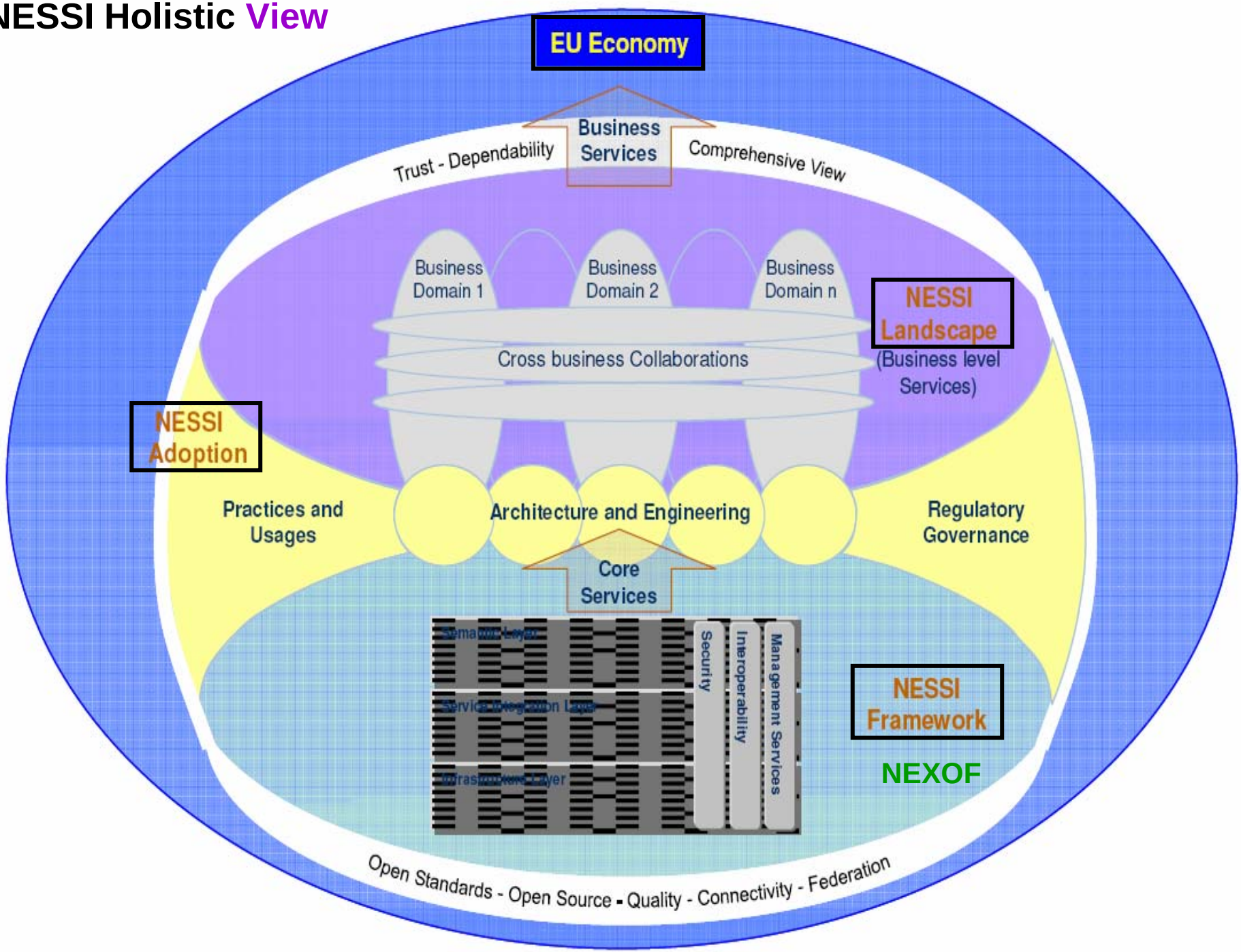
Vision

- Long term strategy of *software* and *services*
- To contribute to Europe's *competitiveness*, *job sustainability* and *quality of life*

- **8 Horizontal NWGs**
  - Business Process Management NWG
  - Semantic Technologies WG
  - Service Engineering WG
  - Service-Oriented Infrastructure WG
  - Services Sciences WG
  - Software Engineering WG
  - Trust, Security and Dependability WG
  - User-Service Interaction WG

- **2 Vertical WGs**
  - eHealth WG
  - iGovernment WG
- **2 Adoption WGs**
  - SME NWG
  - Open Source Software WG

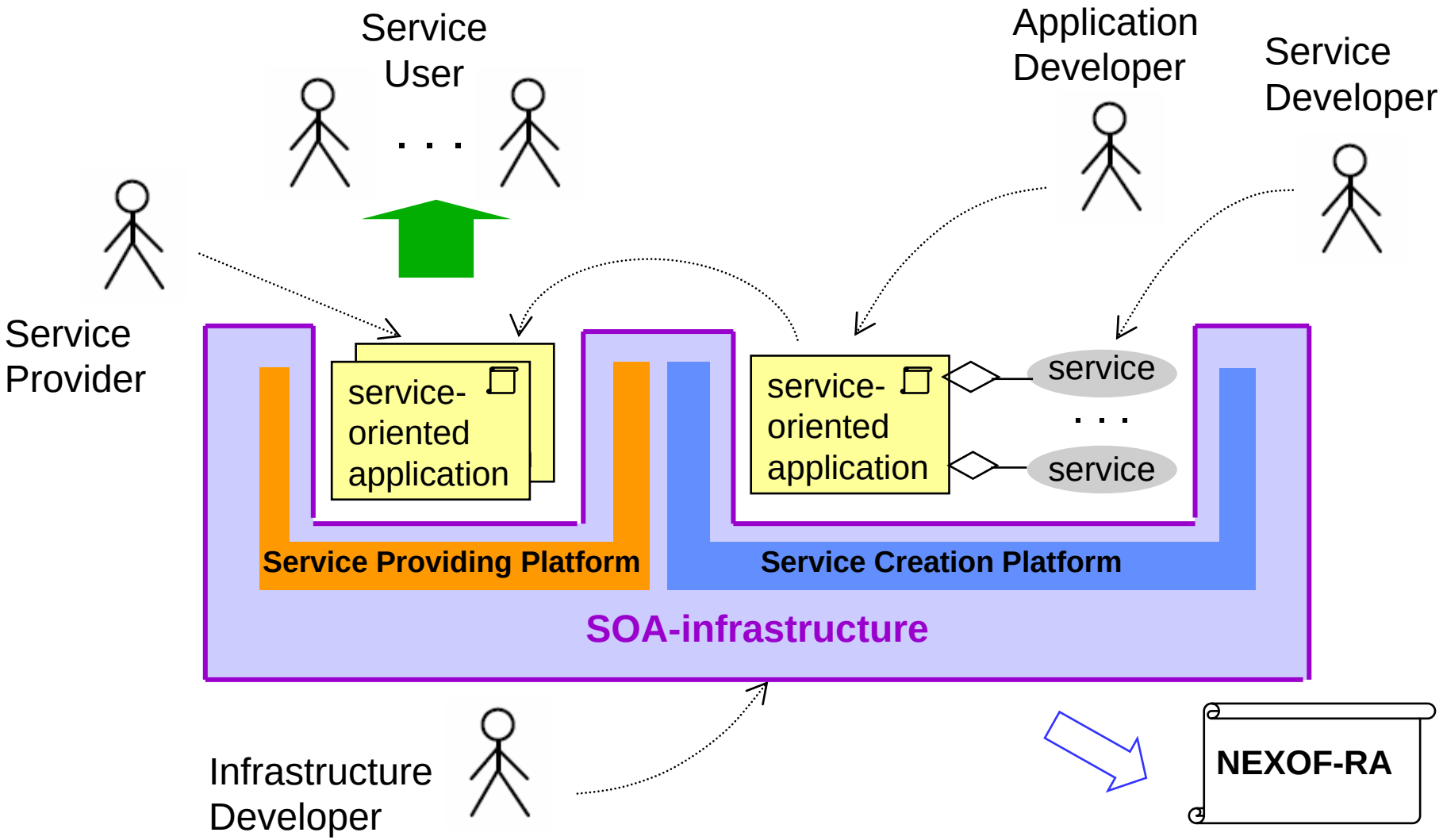
NESSI Holistic View



NEXOF

- **NESSI Holistic Model**
  - Embraces the whole service area
  - Puts a NESSI approach and technologies as key elements in the EU economy transformation.
  - **NEXOF (NESSI Open Service Framework)**
    - Integrated set of technologies, methods and tools.
    - *Open platform for creating and delivering applications*
      - To promote and make real the transformation of the European economy to fully employed service economy
        - Towards ecosystems
        - All the parties involved coexist
        - Can develop into a new economic model.
      - Technological environment to support the whole NESSI model

NEXOF-RA and NEXOF-RA Model



NEXOF-RA and NEXOF-RA Model

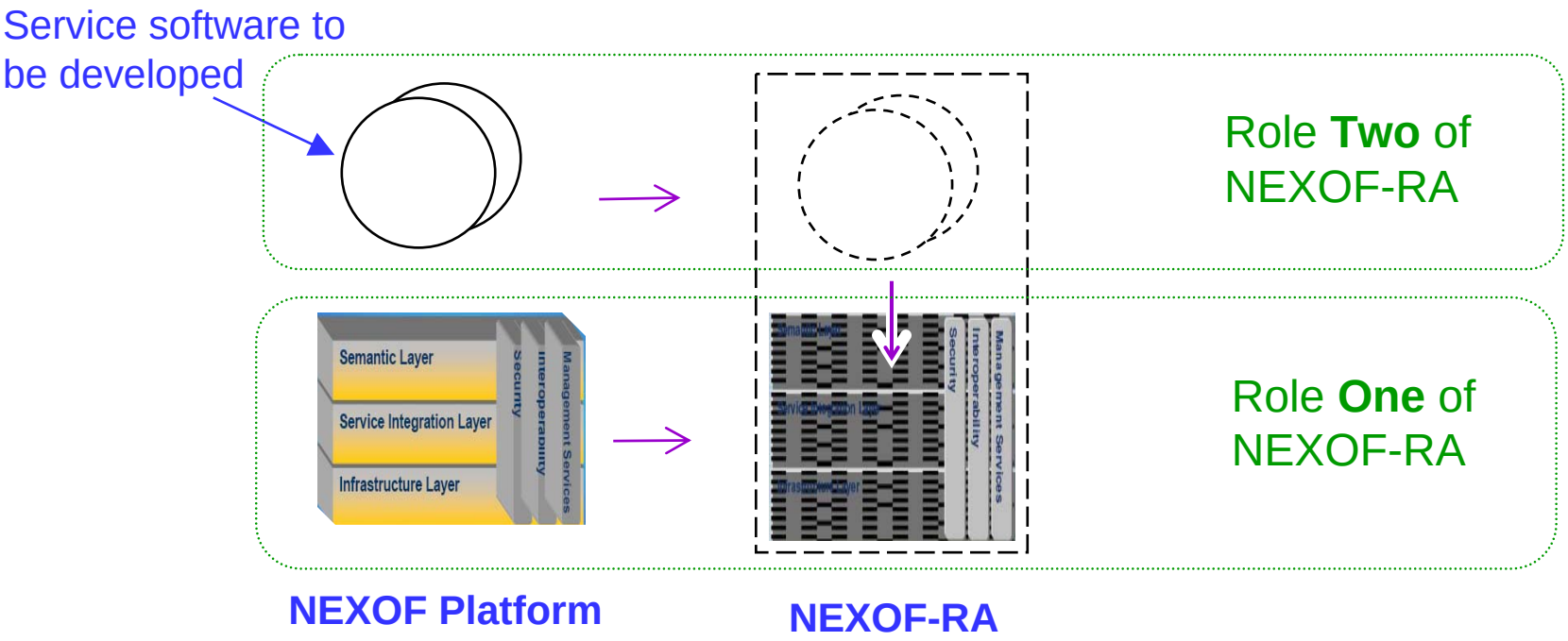
- NEXTOF-RA consists of:
  - A comprehensive set of concepts, which are defined by a conceptual model
  - A set of specifications consisting of standards, components, and patterns
  - Guidelines and principles for system architects to design architectures of service-based software systems

- NEXOF-RA Model
  - includes a well-structured set of different model parts representing different viewpoints.
  - describes elements of the NEXOF-RA on a conceptual level

- NEXOF-RA considers different levels of abstraction and granularity, which are closely interrelated with the specification of building blocks.

☛ Model is a subset of RA and can be used almost *interchangeably*.

NEXOF, NEXOF-RA and NEXOF-RA Model



→ : conforms to

NEXOF-RA Project

- FP7(Seventh Framework Programme)
  - European Union's main instrument for funding research in Europe
  - 2007-2013 (53.2 Billion Euros)
- FP7 Future Internet Projects (79 projects)
  - Area 1 "Future Networks"(21)
  - Area 2 "Services Architectures"(13)
    - NEXOF-RA - **NESSI flagship strategic Project**
    - …
  - Area 3 "Networked Media Systems"(13)
  - Area 4 "Internet of Things"(5)
  - Area 5 "Security"(14)
  - Area 6 "Experimental Test Facilities"(13)
- **NEXOF-RA Project**
  - 6,5 m €
  - *FP7 Objective ICT-2007.1.2: Service and Soft ware Architectures, Infrastructures and Engineering*
  - Projects coordinator: *Mr. Stefano De Panfilis*

II. NEXOF-RA Model V 2.0

TOC (V 2.0)

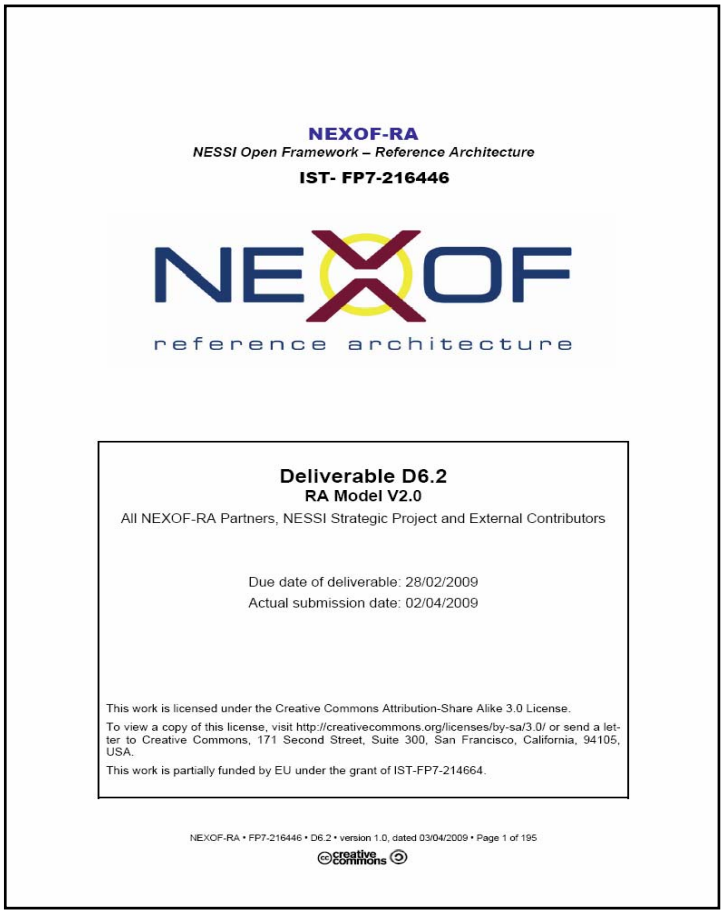
1. Introduction

2. NEXOF Reference Architecture Approach
3. The Basic Characteristics of
   a NEXOF Compliant Infrastructure
4. The Basic Concepts and the Main Concerns

5. Top-Level View

6. Overview: Functionality Description
7. Services
8. Messaging
9. Discovery
10. Presentation
11. Management
12. Security
13. Resources

14. Conclusion and Future Work



NEXOF-RA
NESSI Open Framework – Reference Architecture
IST- FP7-216446

NEXOF reference architecture

Deliverable D6.2
RA Model V2.0
All NEXOF-RA Partners, NESSI Strategic Project and External Contributors

Due date of deliverable: 28/02/2009
Actual submission date: 02/04/2009

This work is partially funded by EU under the grant of IST-FP7-214664.

Written by 45 contributors!

2. NEXOF Reference Architecture Approach

2.1 Architecture-related Terms in Traditional Software Engineering

2.2 Usage Purpose of the Reference Architecture: Design Process

2.3 Structure of the NEXOF Reference Architecture

2.1 Architecture-related Terms in Traditional Software Engineering

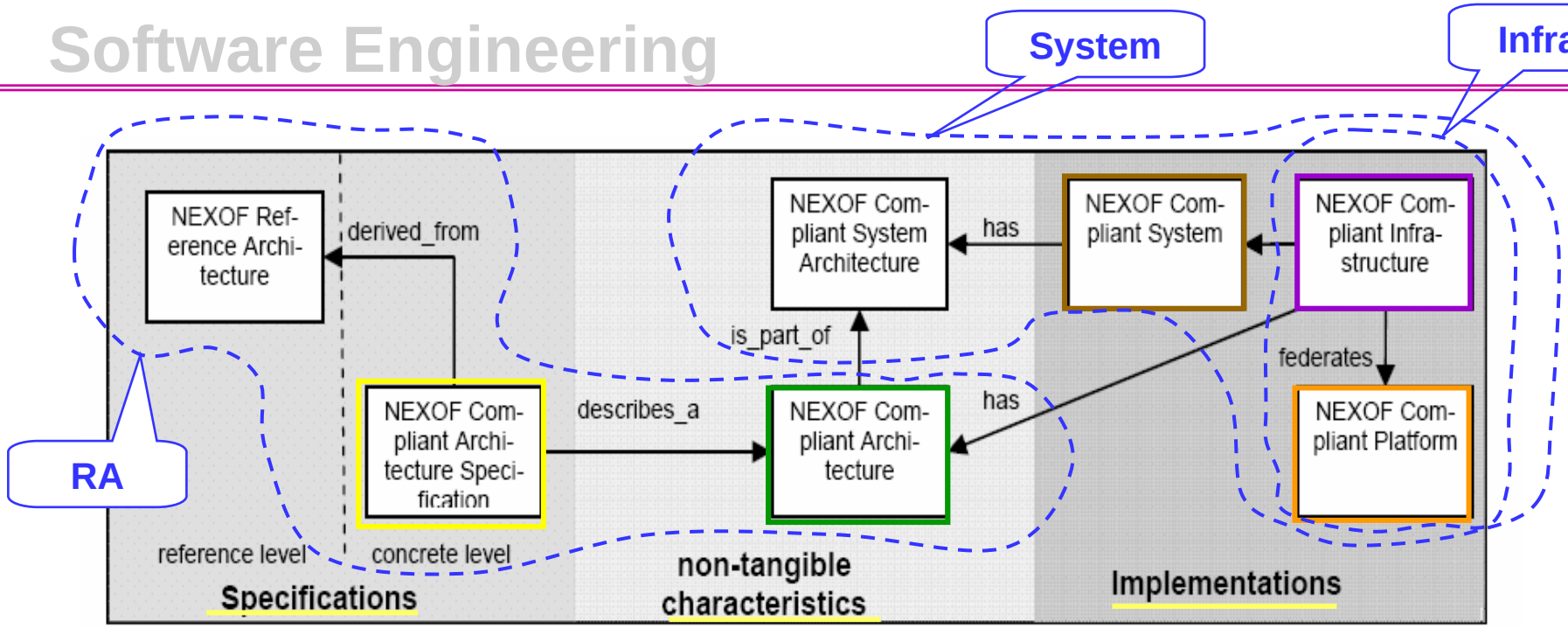


Figure 3: Architecture terminology in the NEXOF context

- **NC Architecture *only addresses organizational structure characteristics of the* hardware and software *infrastructure*.**
- **Infrastructure implementation with NC Architecture is called a NC Infrastructure.**
- **NC Infrastructure is an open federation of several smaller service-base software infrastructures called NC Platforms.**
- ***A system that is based on such a NC Infrastructure is called* a NC System.**
- **SOA is a *non-tangible characteristic* of a software system**

2.2 Usage Purpose of the Reference Architecture: Design Process

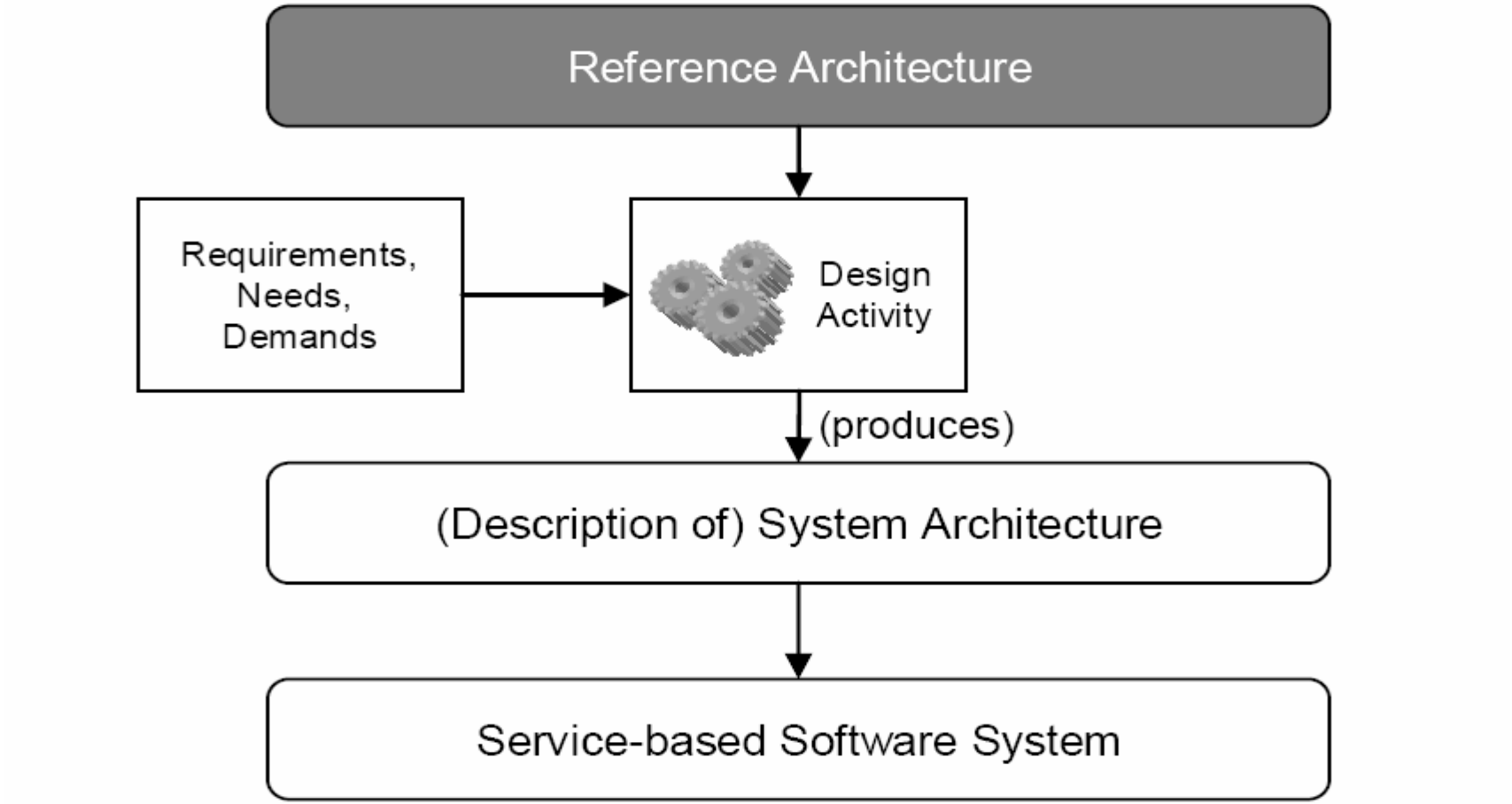


Figure 4: The NEXOF Reference Architecture Design Process

2.3 Structure of the NEXOF Reference Architecture

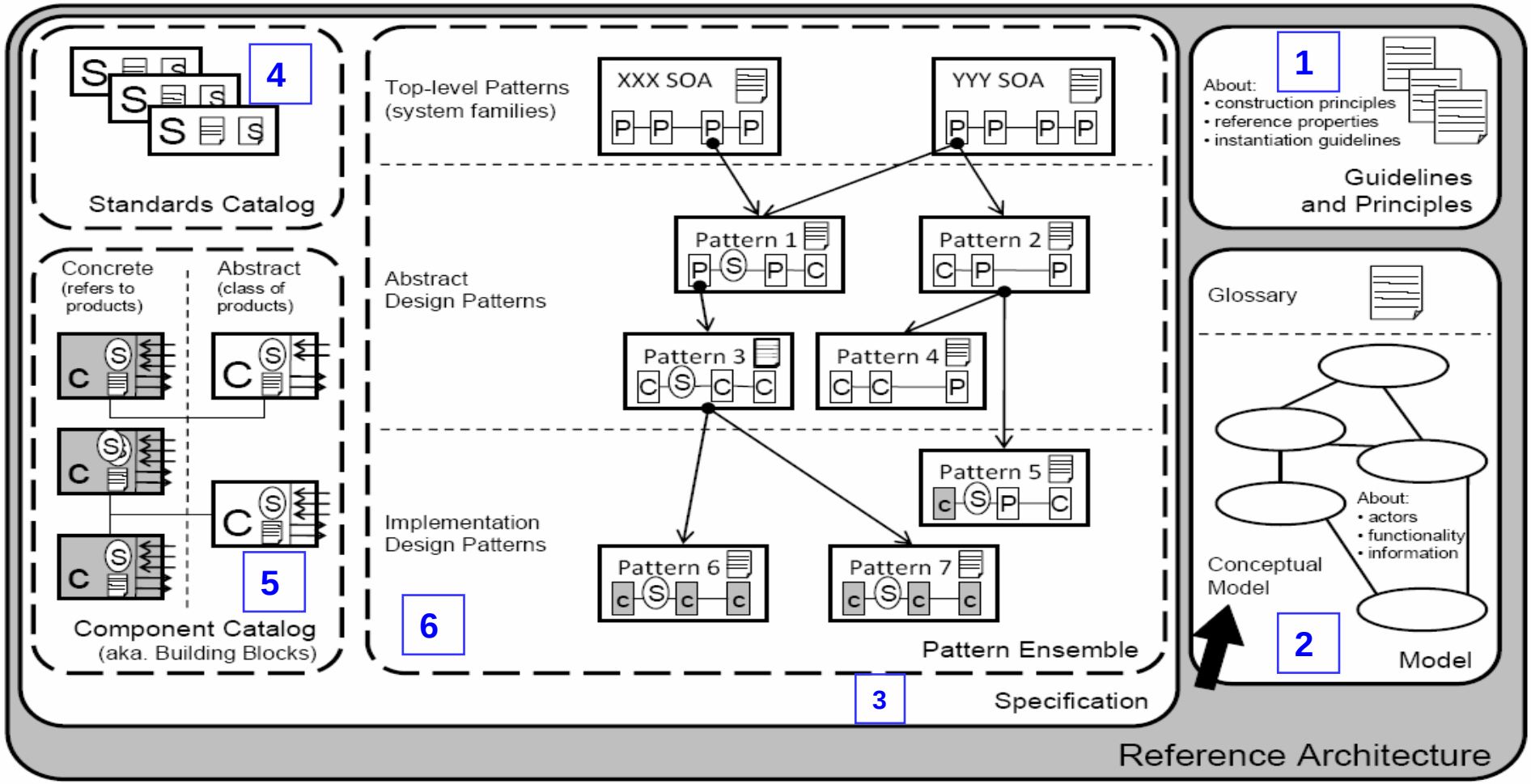


Figure 5: The NEXOF Reference Architecture Structure

3. The Basic Characteristics of a NEXOF Compliant Infrastructure

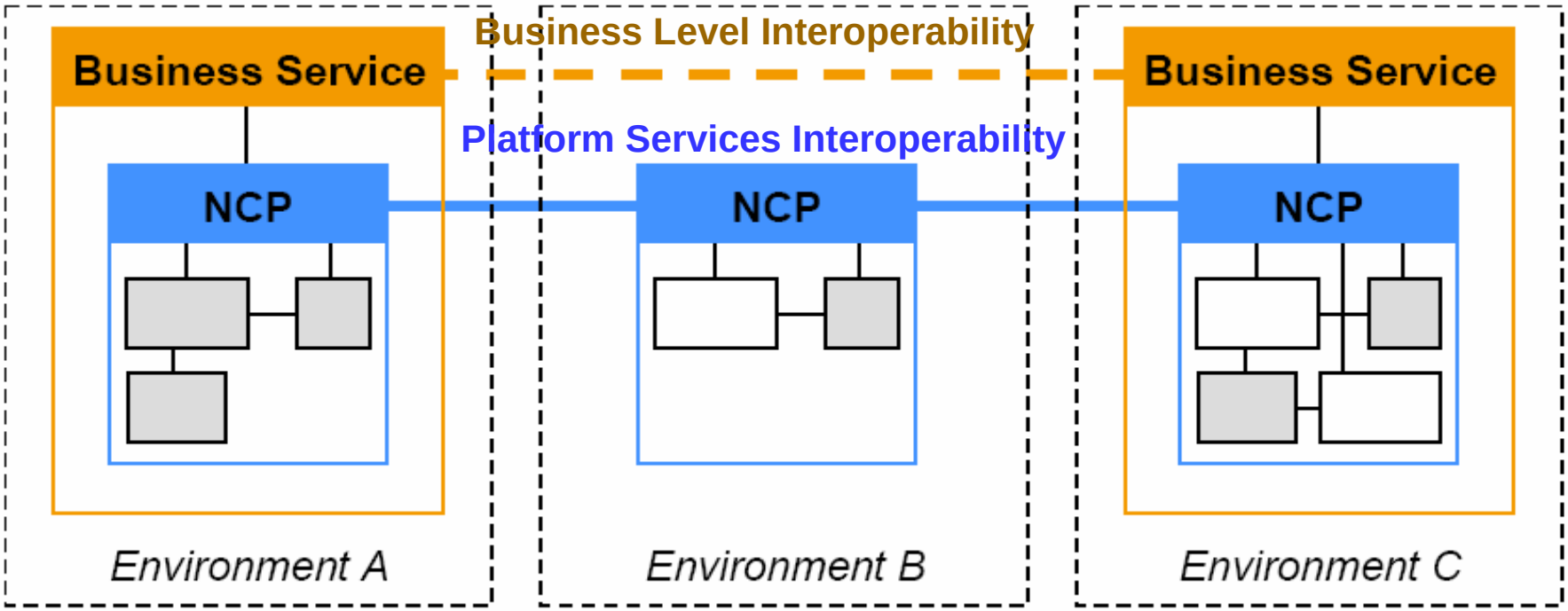


Figure 6: NEXOF Compliant Infrastructure

- Business services and applications are developed, deployed and executed on top of the overall NEXOF Compliant Infrastructure.

3.1 Platform Definition

- **Hardware Platform:** E.g. Gaming, mobile, Internet of Things domain
- **Software Platform:** E.g. OSs, Java
- **Service Platform**
  - A place to execute software and execute services
  - Examples
    - Amazon Elastic Compute Cloud E2C
      - Execute software application
      - Combine execution with service characteristics => "elasticity rules".
        - Access to hardware resources dynamically scaled with needs
        - Dynamically adapt (together with the application) to the characteristics of a service (e.g. defined by SLAs).
    - Facebook
      - Allows the programming and execution of social applications.
      - As applications become more popular, access to more resources
      - Application can be built based on the many social network specific APIs and data sources that the platform provides.

3.2 Characteristics of NEXOF-RA Compliant Platforms

- Popular service platforms: e.g. Facebook, OpenSocial etc.
- Service-based platforms differ in terms of the following four:
  - **Openness**
    The degree to which a platform can be programmed/adapted by users
  - **Expandability**
    - **Different types of extension**
      1) Rendering an existing functionality in a new way
      2) Can add new features
  - **Interoperability**
    1. Business Level Interoperability
    2. Business – Platform Interoperability
    3. Platform Components Interoperability
    4. Platform Services Interoperability
  - **Federation**
    - The outside developer can access the functionalities of all platforms via one – or use an access layer service to access all underlying platforms.

4. The Basic Concepts and the Main Concerns

4.1 Services and Software Services
4.2 The "SOA Infrastructure" Puzzle
4.3 Separation of Concerns

4.1 Services and Software Services (1/2)

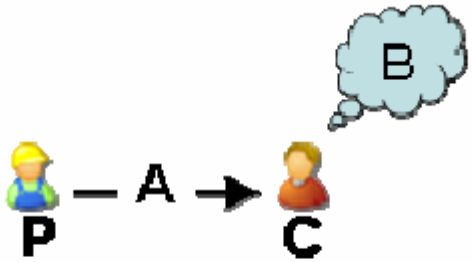


Figure 7: Action for benefit - graphical notation

- $ServiceConcept_1$
  $=_{def}$ [**P**,**A**(B),C]

=> An action performed by one entity (provider) for the benefit of another (consumer)

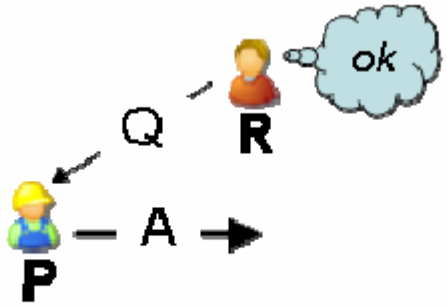


Figure 9: Action on request - graphical notation

- $ServiceConcept_2$
  $=_{def}$ [**P**,**A**(Q),**R**]

=> "An action performed by one entity (provider) that matches a request of another (requesting entity), according to the interpretation of the latter"

4.1 Services and Software Services (2/2)

- **NEXOF-RA Definition of Service**: "*An action performed by one entity (provider) that matches a request of another (requesting entity), according to the interpretation of the latter*"

- **NEXOF-RA Definition of Software Service: "**a Service in which software agents that are an agent requester and an agent provider mediate the interaction between requester entity and provider entity."

=> The direct interaction never happens between humans.

=> Human interaction is not excluded.
  – To fulfill their duty software agents may need human interaction.

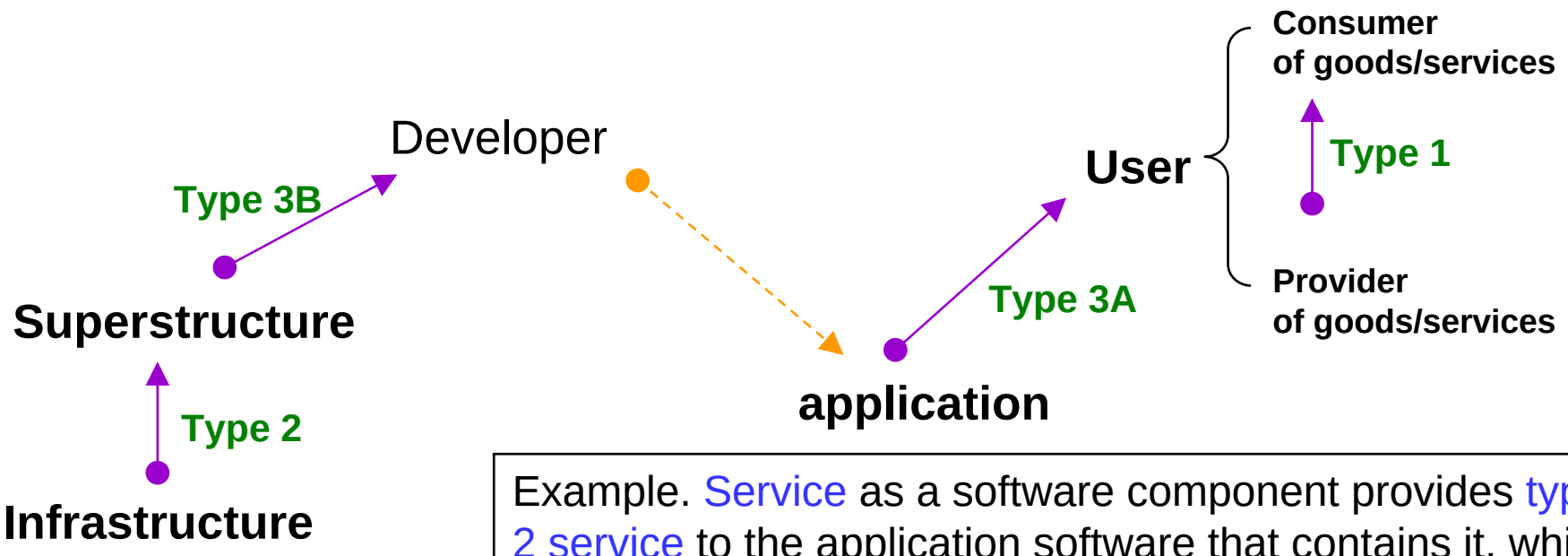
Service Concept (Extended from [Kang 09])

: service provisioning relation

: "develop" relation

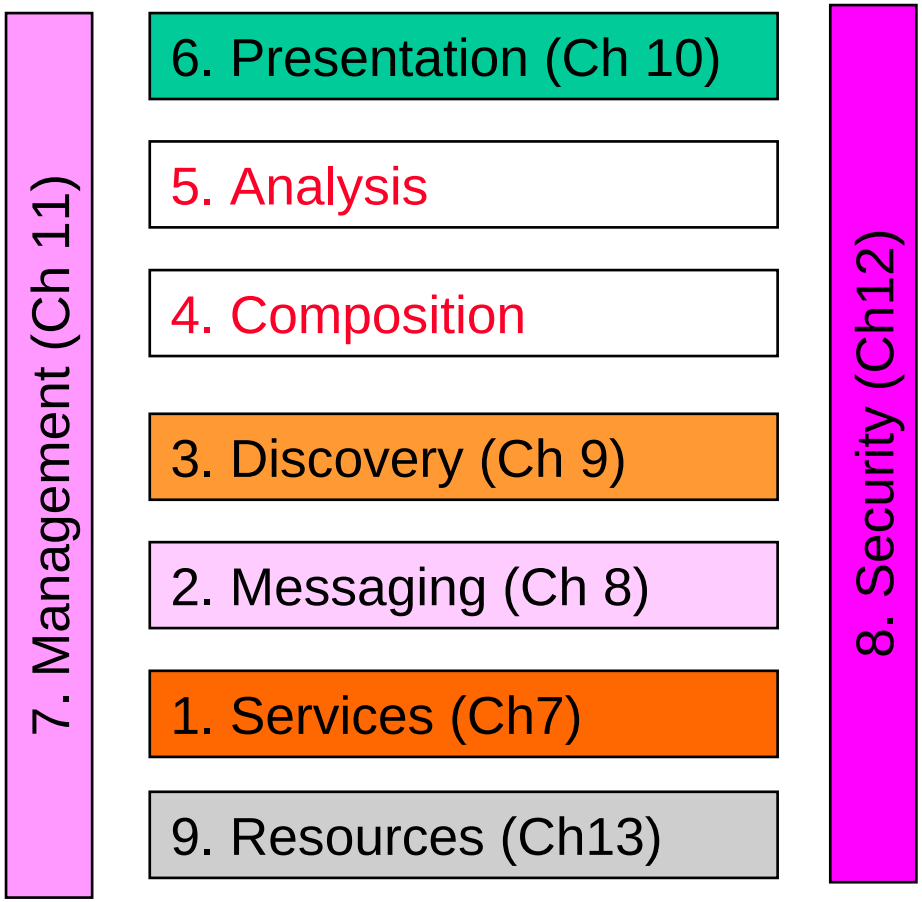
Type 1 service: human-to-human
Type 2 service: agent-to-agent
Type 3 service: agent-to-human
Type 4 service: human-to-agent

Agent: machine, i.e. software or hardware



Example. Service as a software component provides type 2 service to the application software that contains it, which in turn provides type 3 service to its human user.

4.2 The "SOA Infrastructure" Puzzle



- **Pieces that make "SOA Infrastructure" puzzle**

9 Concerns

Each concern will be discussed in section 5.4 on "the Overview Model"

5. Top-Level View

- The *conceptual architecture view* that is constituted by the *description of the provided functionality* is closely related to the RA-Model.

- NEXOF-RA Model is 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.

5.1 Creation Process of NEXOF-RA
5.2 Stakeholders of NEXOF-RA (Discussed in …)
5.3 Separation of Function, Behavior, Structure, and Information
5.4 The Overview-Model

5.3 Separation of Function, Behavior, Structure, and Information

- Due to the complexity of SOA, need to employ diverse ***viewpoints***:
  1) Behavior viewpoint
  2) Structure viewpoint
  3) Data viewpoint
  4) Function viewpoint <– To be described in natural language

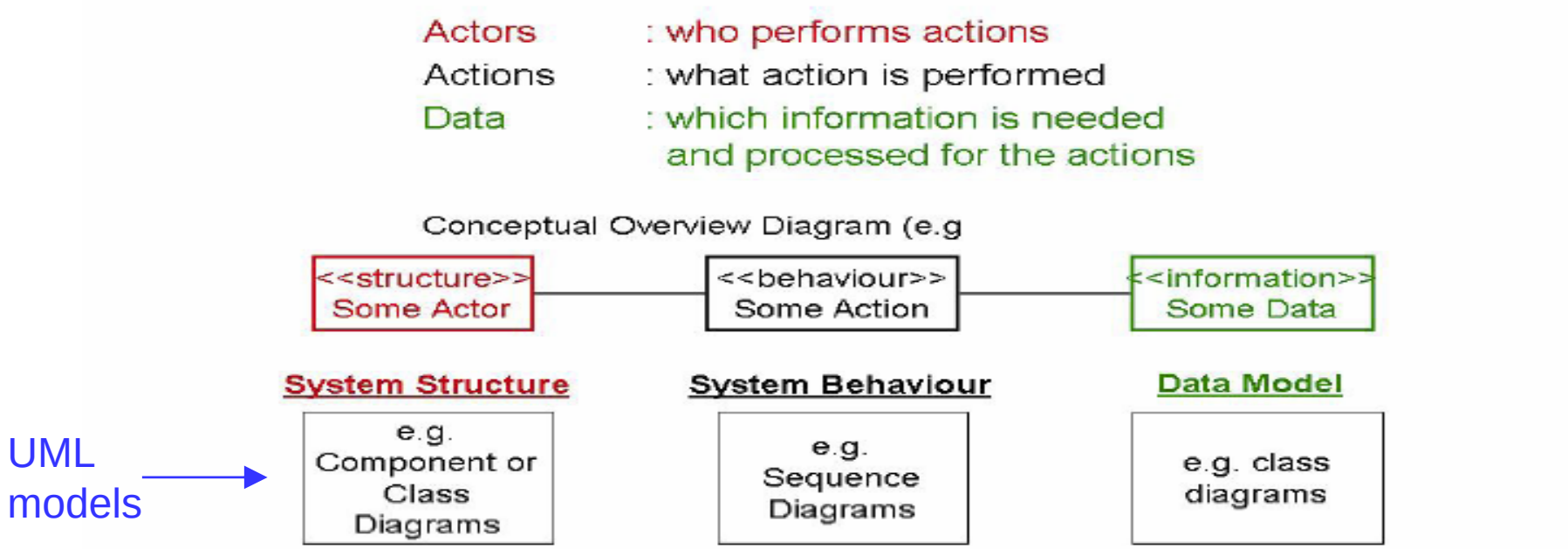


Figure 17: Separation of Structure, Behaviour and Information

5.4 The Overview-Model (1/2)

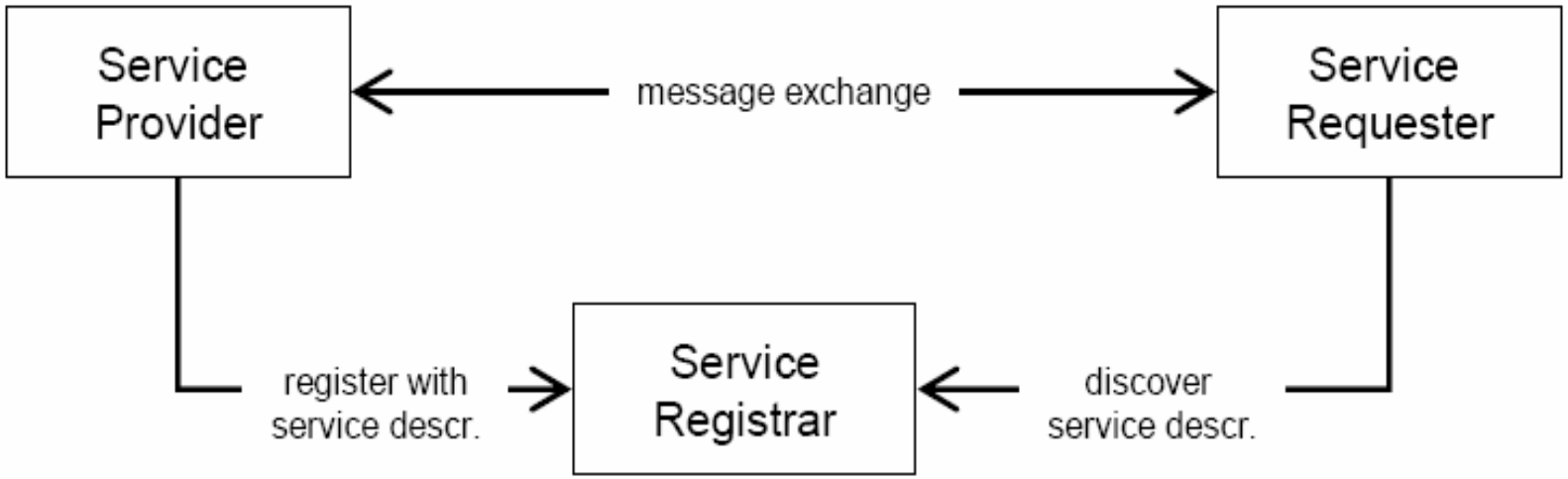


Figure 18: Three principle concepts of a SOA from a structural perspective

- 3 main concepts that are part of every SOA
  - *Service provider* is an entity that offers one or more services.
  - *Service requester* is an entity that consumes one or more services.
  - Arrow represents an exchange of information (*Service provider* and *service requester* communicate by exchanging messages.)

5.4 The Overview-Model (2/2)

- The top-level-view model of NEXOF-RA describes the relevant *concepts* and *relationships between the basic concepts* of a service-oriented architecture.

5.4.1 Services
5.4.2 Service Composition
5.4.3 Communication between Service Provider and Service Requester
5.4.4 Service Discovery
5.4.5 Management

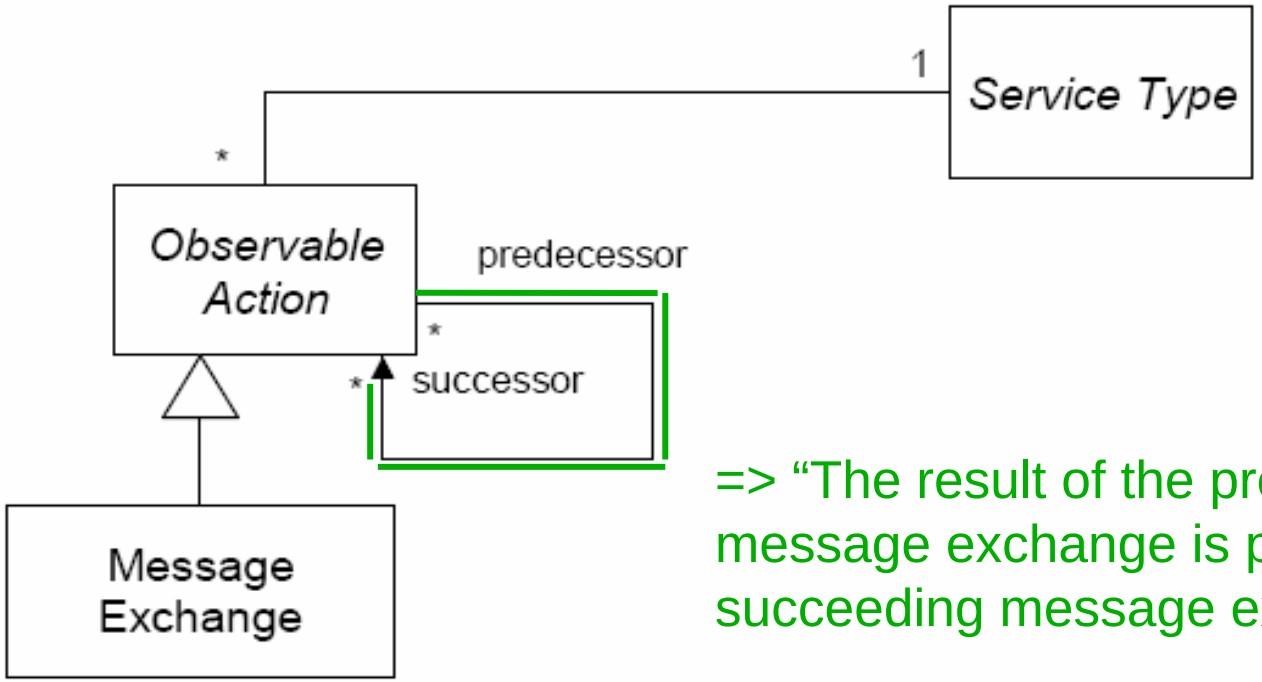
1. Services
2. Messaging
3. Discovery
4. Composition
5. Analysis
6. Presentation
7. Management
8. Security
9. Resources

SOA = Architecture of SO Application ?

5.4.1 Services (1/3)

- **Services**
  - Creation (design, implementation, modify, testing, deployment)
    - Method
      - new service to be created
      - existing applications to be exposed as service implementations
        - Using adapter, wrapper
  - Execution
    - Computation performed by the provider agent to reply to the service invocation

5.4.1 Services (2/3)



=> “The result of the preceding message exchange is passed to the succeeding message exchange.”

Figure 19: Service type (Information)

- Observable actions characterize the type of the service that is offered.
- An important type is a *message exchange*, which implements the communication between *service provider* and *service requester*.

5.4.1 Services (3/3)

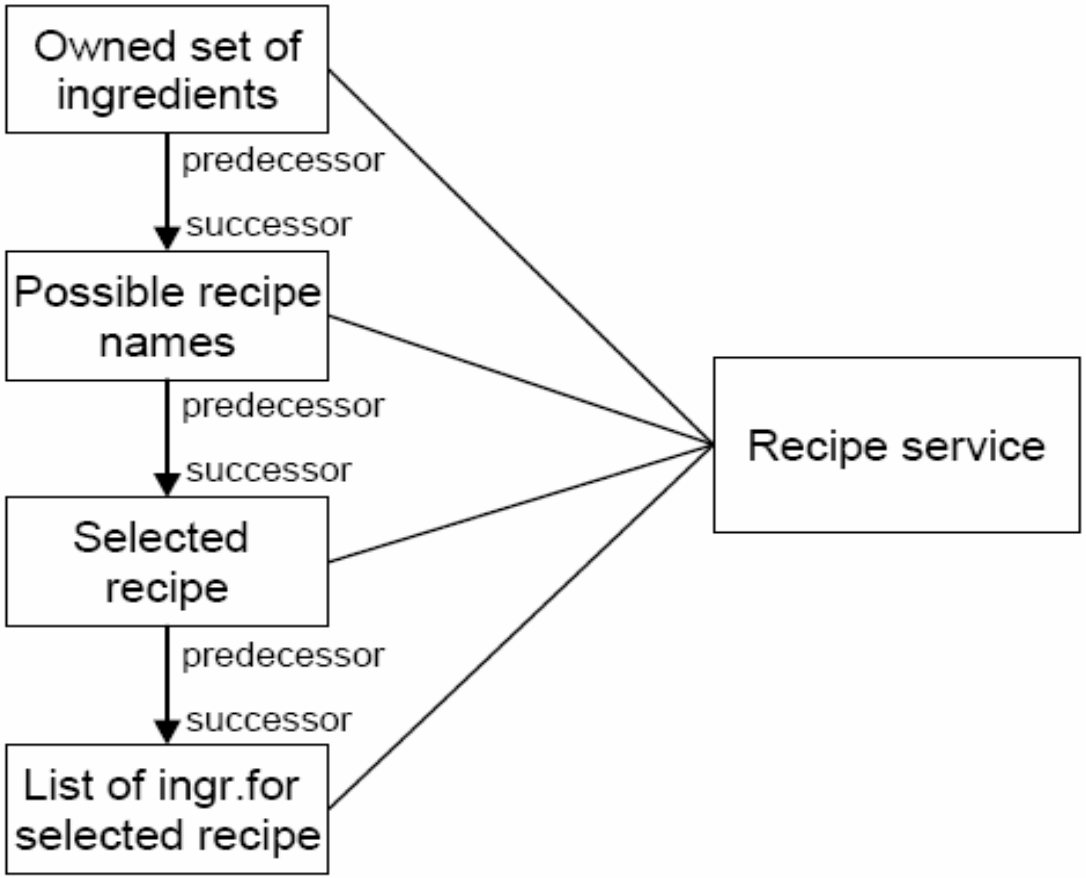


Figure 20: Example for a service

- "Recipe service" allows you to identify a cooking recipe for the set of ingredients that you have at home.

5.4.2 Service Composition

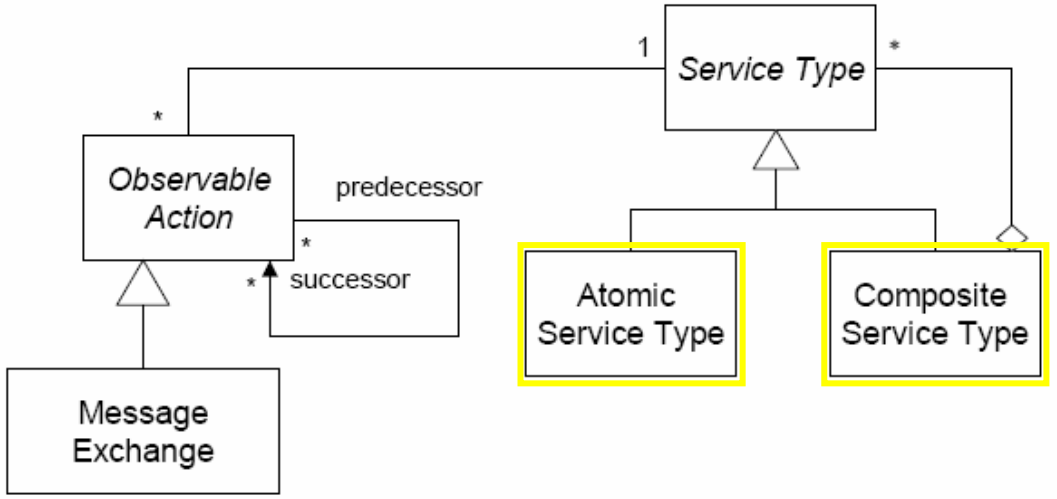


Figure 21: Service composition (Information)

• In order to enable service composition, the model of service type in Figure 19 is extended.

•Two levels:
  – design-time (orchestration, choreography)
  – run-time (static/dynamic composition)
• Result of composition of services is called process.
• Process
  –used as an application or as a service (composite service)
  –described, published, discovered and composed transparently
  –support dynamic process reconfigurations, delivering business agility.

5.4.3 Communication between Service Provider and Service Requester (1/4)

- **Messaging**
  - Communication capability
    - Connection-independent : service requestor/provider => loosely coupled
    - Map one data format into another
      - Messaging layer transforming required
      - Consider reliability and performance of messaging layer

5.4.3 Communication between Service Provider and Service Requester (2/4)

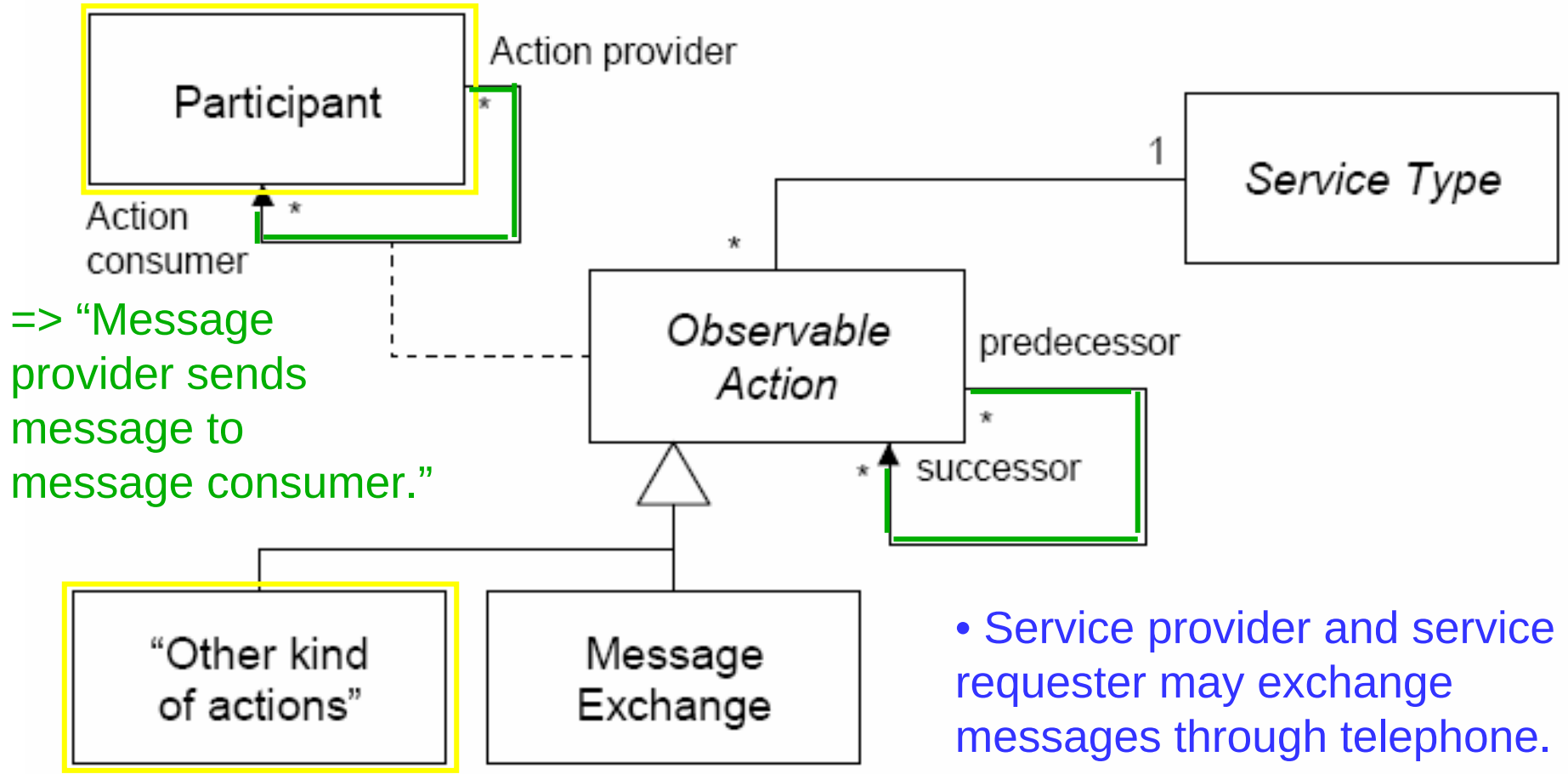
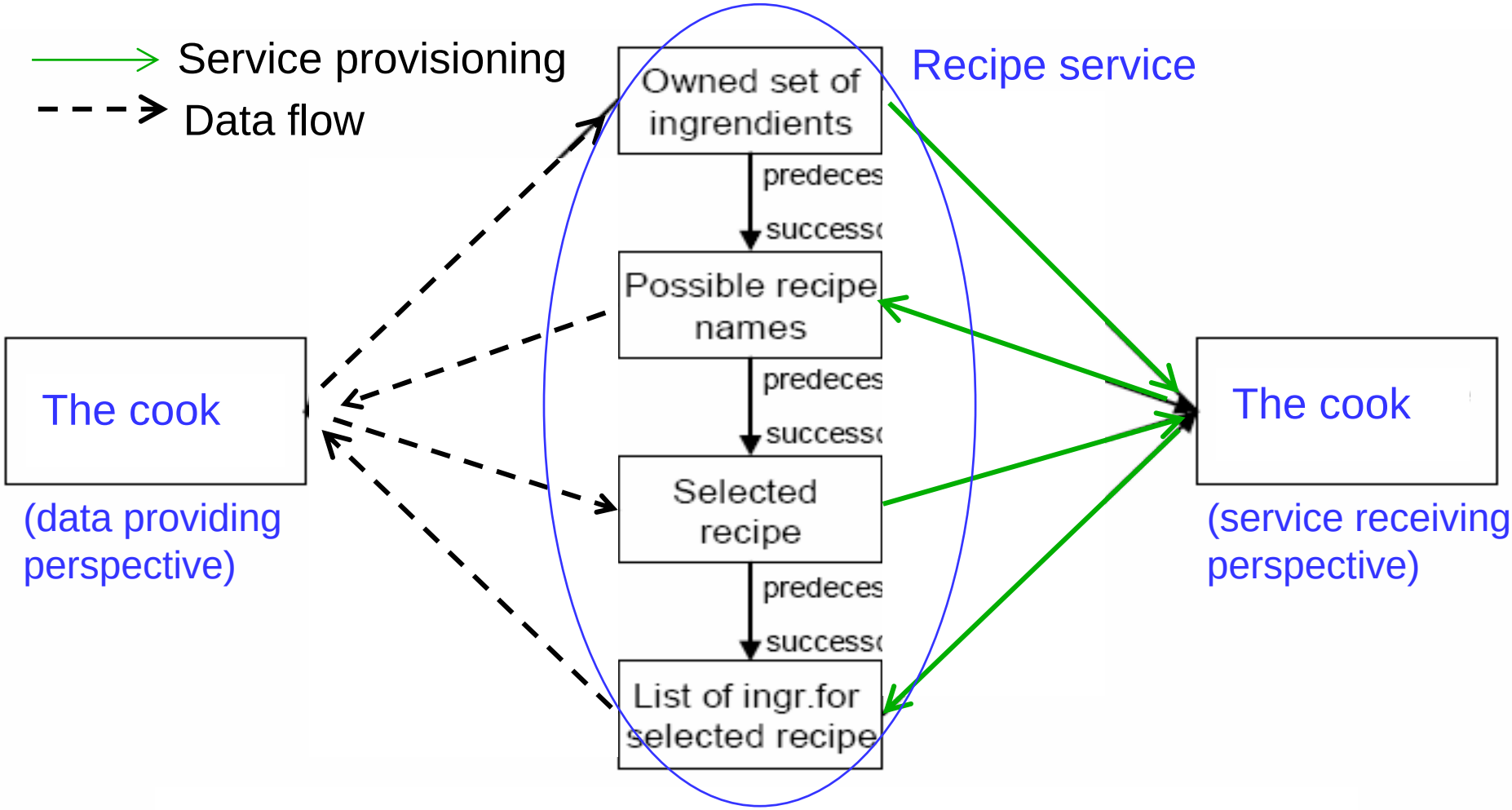


Figure 22: Principle communication in a SOA (Information)

5.4.3 Communication between Service Provider and Service Requester (3/4)



5.4.3 Communication between Service Provider and Service Requester (4/4)

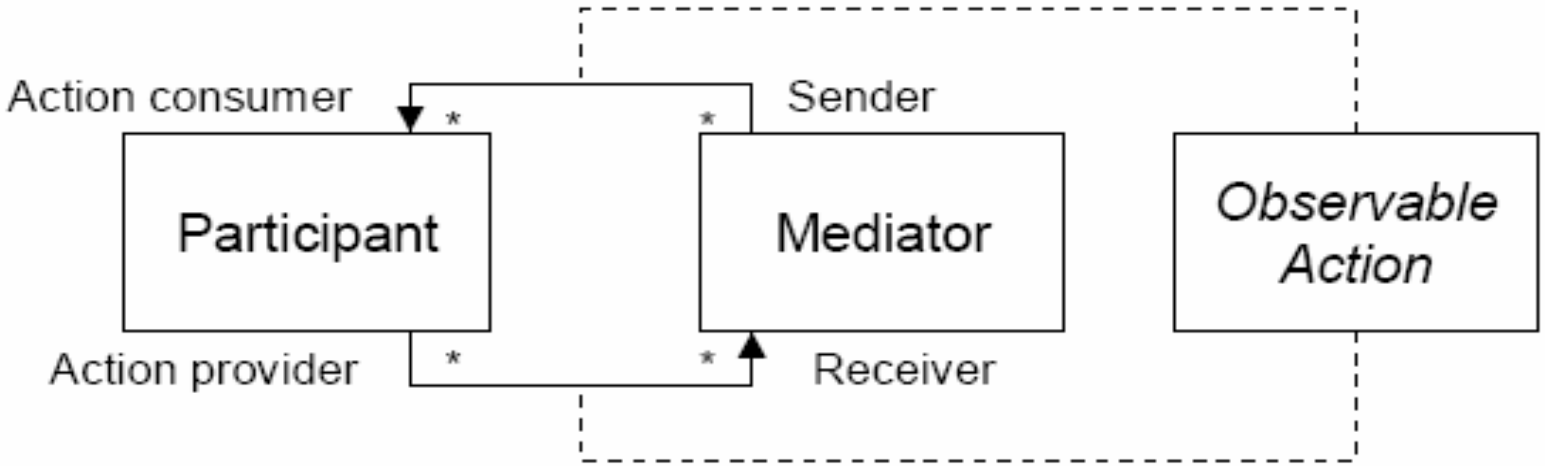


Figure 24: Principle communication in a SOA (Information)

- In a *federated architecture* between different companies, the communication between a service provider and a service requester is performed with the help of a mediator.

5.4.4 Service Discovery

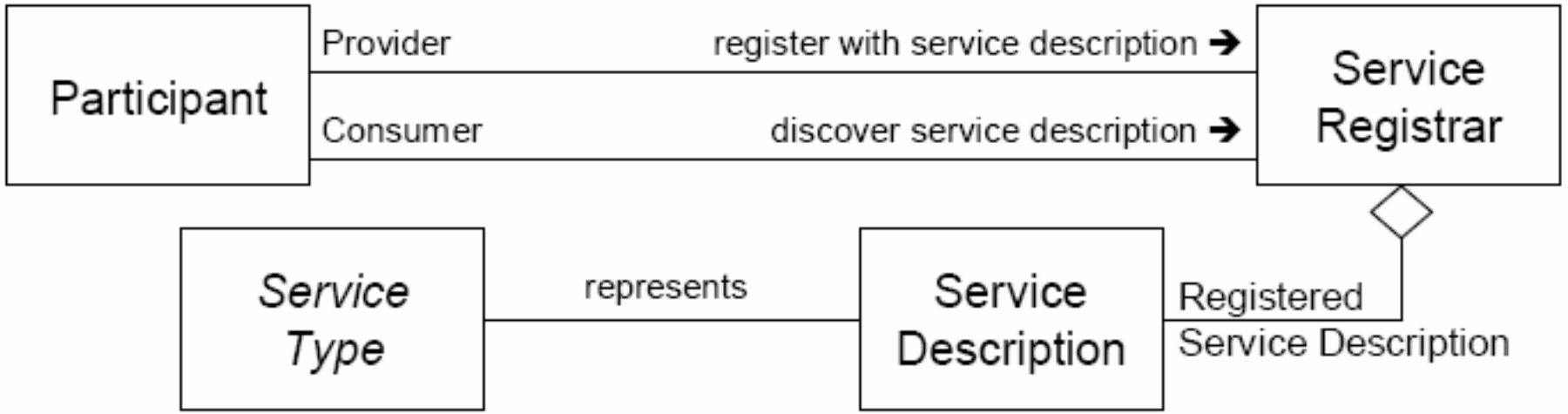


Figure 25: Service registrar (structure)

- An SOA can exist without an explicit service registrar.
    - How a consumer becomes aware of the existence of services
        - Pull policy: consumers directly search on a registry
        - Push policy: a consumer is notified

5.4.4 Service Discovery (2/2)

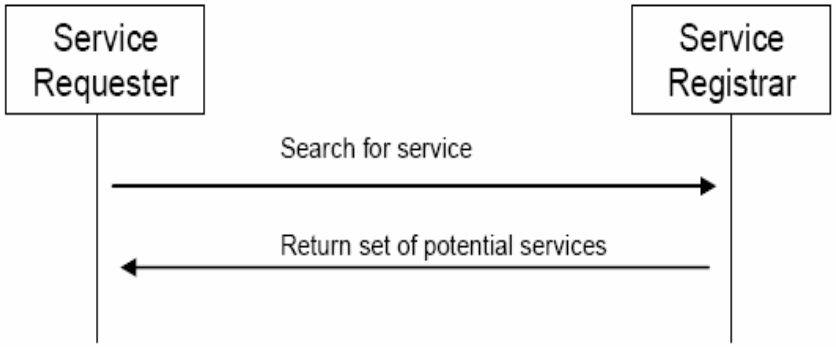


Figure 26: Service discovery with pull (behaviour)

Examples of behavior view

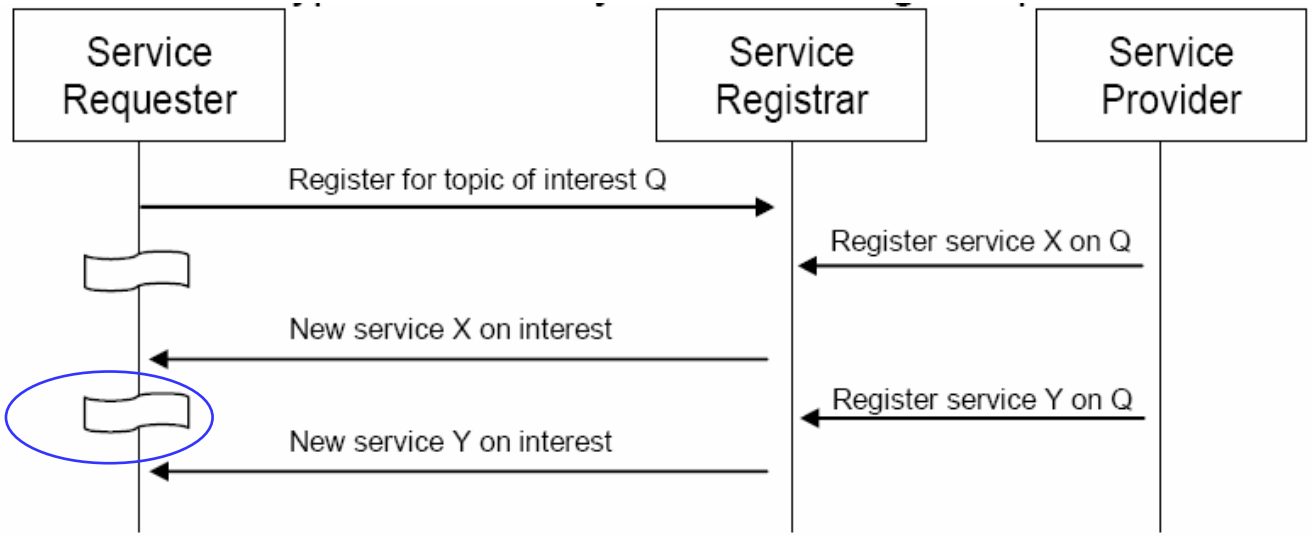


Figure 27: Service discovery with push (behaviour)

5.4.5 Management

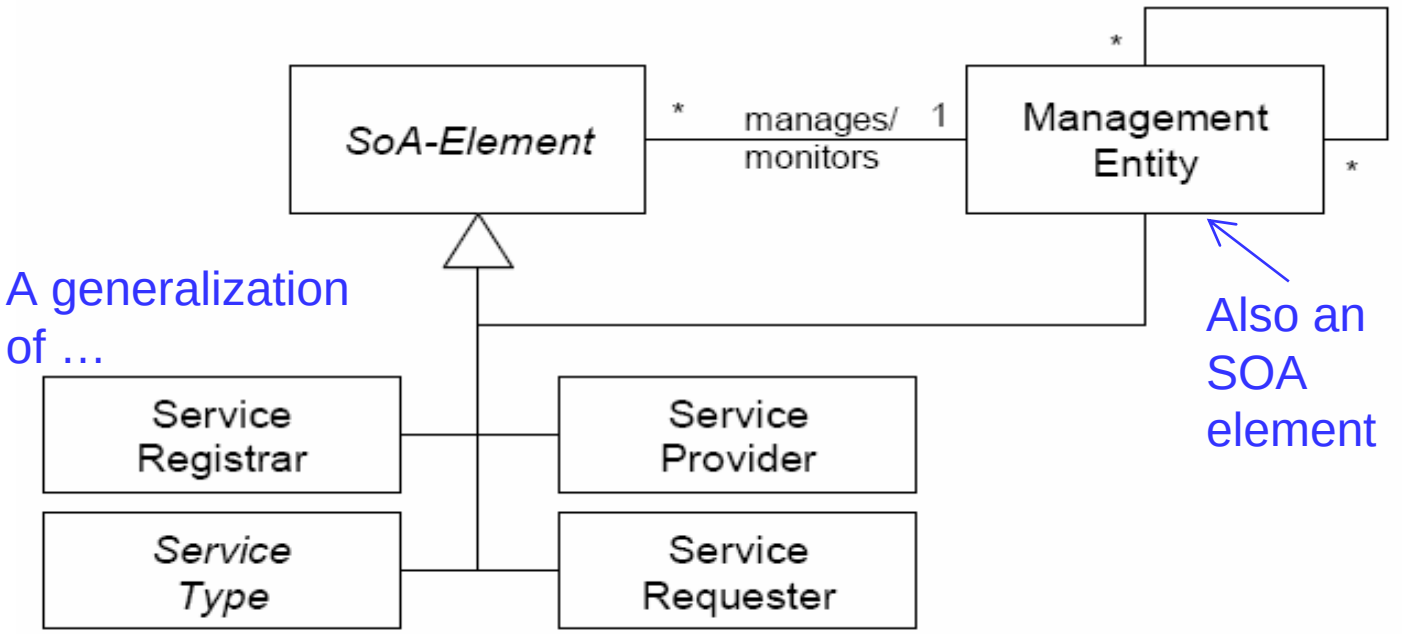


Figure 28: Management in a SOA (Structure)

- Management is the management and monitoring of
  - *services and processes*, and
  - *the usage of all platform functionalities*.

Other Concerns - Analysis

- Functionalities that support the analysis of information related to the execution of services and processes with respect to business requirements.

- To deal with business performance, need to track, record, and manage
  - process metrics
  - business events

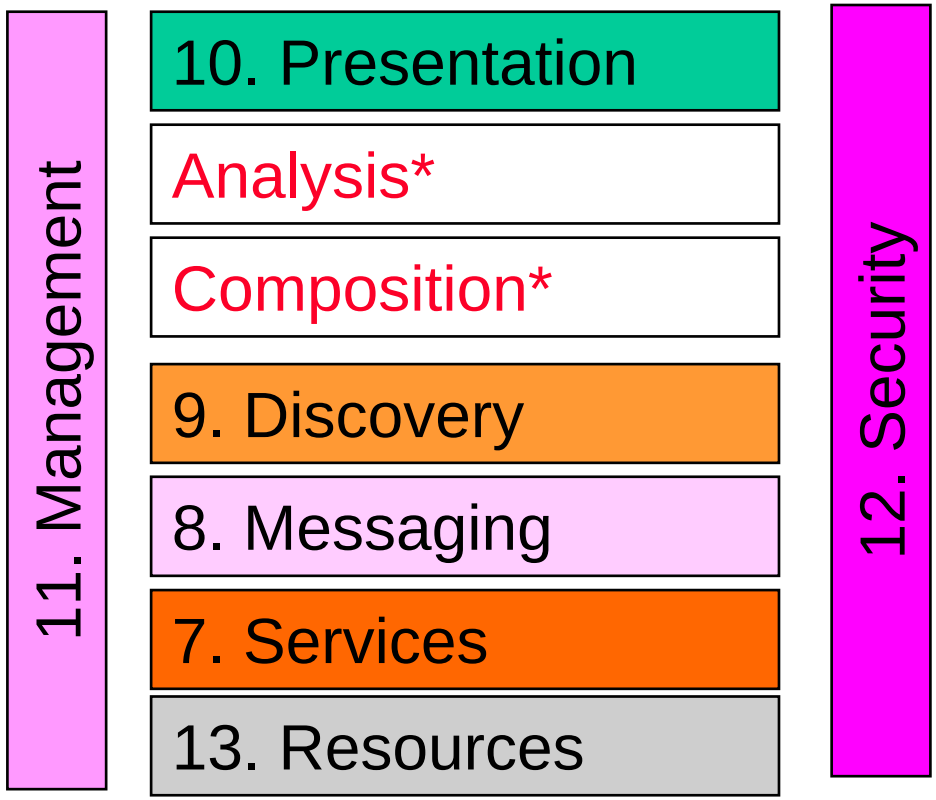
- A facility should display the event-related information in order to support improved management decision-making.

- Monitoring the business behavior of the real-time system
  => match that behavior to regulatory or corporate policy requirements
  => detect out-of-line situations
  => companies can respond much more quickly and effectively to compliance failures, mitigating any associated penalties.

Other Concerns – Presentation, Security, Resources

- **Presentation**
  - All mechanisms to enable human users to interact and make use of the functionality provided by the overall platform
    - GUI
    - APIs to enable the creation, customization and execution of such graphical user interfaces.
- **Security**
  - User/client authentication
  - Access control authorization
  - Link level encryption or network segregation of messages
  - Denial of service attacks
  - Tampering with information flows
- **Resources**
  - Computational resources needed to support the execution of all the software components that constitute the platform
  - Software component is not resource

6. Overview: Functionality Description



* Analysis: not considered because higher priority topics should be solved first.
* Composition : Considered but due to delay the results could not be integrated.

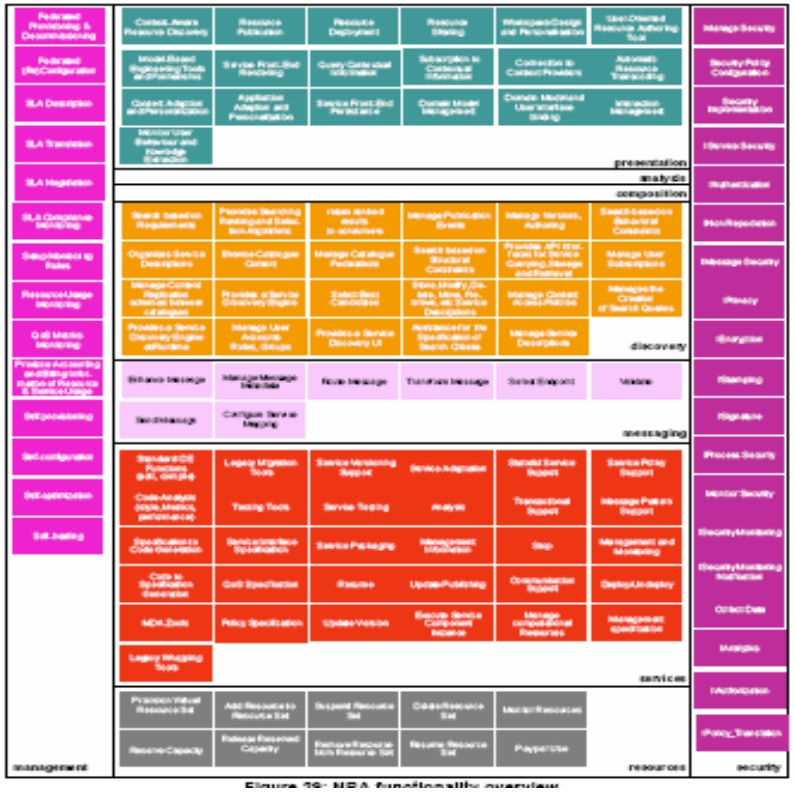


Figure 29: NRA functionality overview

7. Services

- **Service Concerns**
  - The creation of service implementation (design-time support)
    - **From existing or newly written**
  - The deployment of service implementation
    - **Where to deploy a service**
    - **When to start or stop it**
  - The execution of service implementation (run-time support)
    - **Computation performed by the service provider in order to reply to a service consumer request for a specific service**

7.1 System Requirements

7.2 Functionalities and Information Entities

7.3 Rationale

7.1 System Requirements

- Lists the systems requirements related to the Service subsystem

7.1.1 System Requirements for Service Creation

7.1.2 System Requirements for Service Deployment

7.1.3 System Requirements for Service Execution

7.1.1 System Requirements for Service Creation

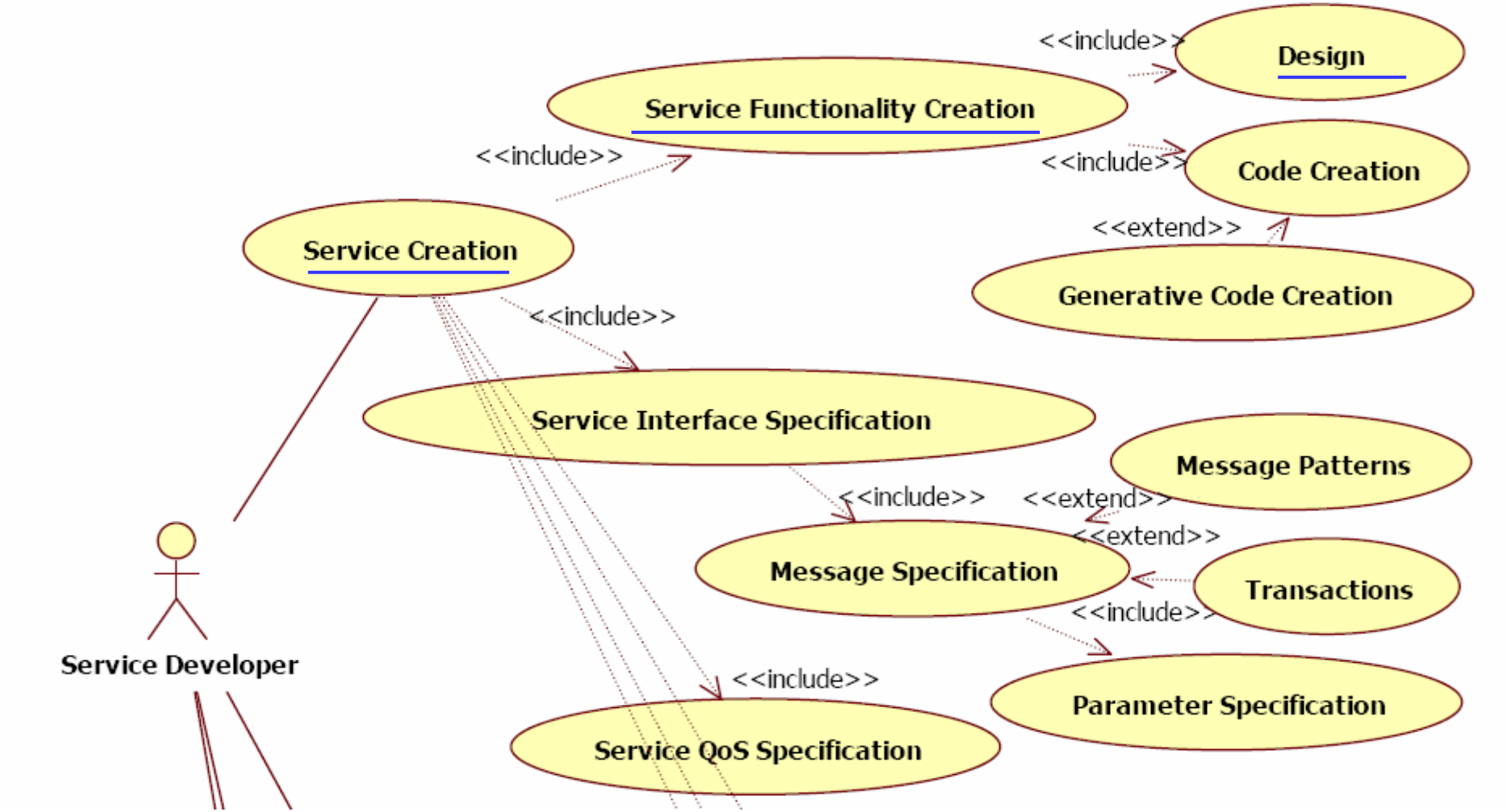


Figure 30: Service Creation requirments

7.1.1 System Requirements for Service Creation

Textual Description of Requirements

SR name: Service Creation

SR description: Facilities are needed to create a service implementation and all necessary specifications.

Related business requirements: NA

SR name: Service Functionality Creation

SR description: Facilities to develop service from scratch. The service has to be developed in accordance with all its interface and non-functional specifications disregarding if they will be developed before or after the functional parts.

Related business requirements: NA

SR name: Design

SR description: Facilities for designing service code.

Related business requirements: NA

7.2 Functionalities and Information Entities

- Describes *functionalities and the information that the platform has to provide and manage.*

7.2.1 Service Creation Functionality and Information Model

7.2.2 Service Deployment Functionality and Information Model

7.2.3 Service Execution Functionality and Information Model

7.2.1 Service Creation Functionality and Information Model

The only stakeholder

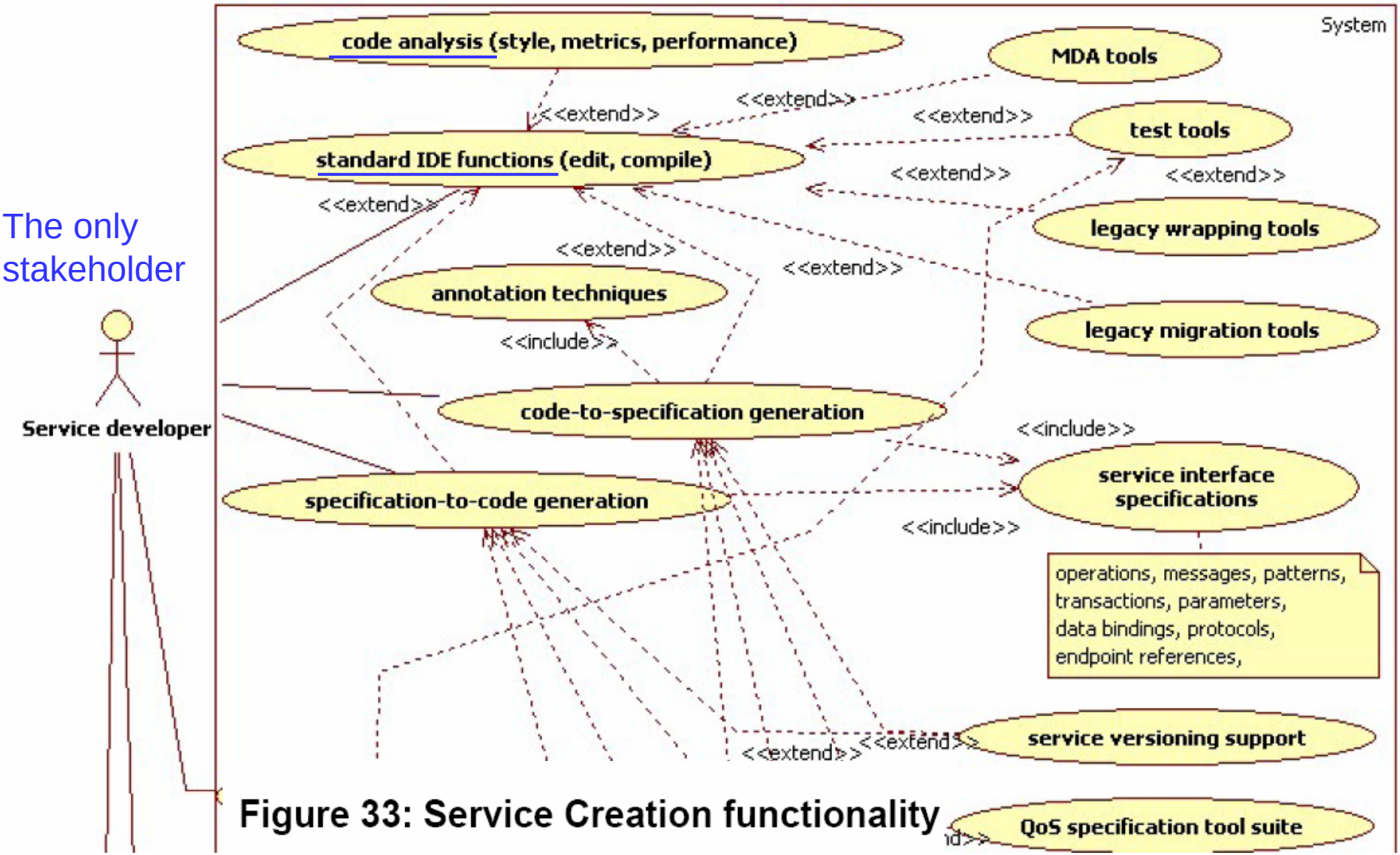


Figure 33: Service Creation functionality

7.2.1 Service Creation Functionality and Information Model

- Textual description of functionalities.

  (Functionalities refer to the information model to be introduced later.)

Functionality name: standard IDE functions

Description: best practice facilities for editing and compiling code that will be used as service implementation

Related system requirements: Code Creation

Related information model entities (optional): Service Implementation, Operation Signature

Related functionalities (optional): NA

Source: Products: Eclipse, VisualStudio

Functionality name: code analysis

. . .

7.2.1 Service Creation Functionality and Information Model

- The model of the information managed by the functionalities.
- Information for design and testing that is not of primary relevance for architectural decisions is abstracted in as few classes as possible.

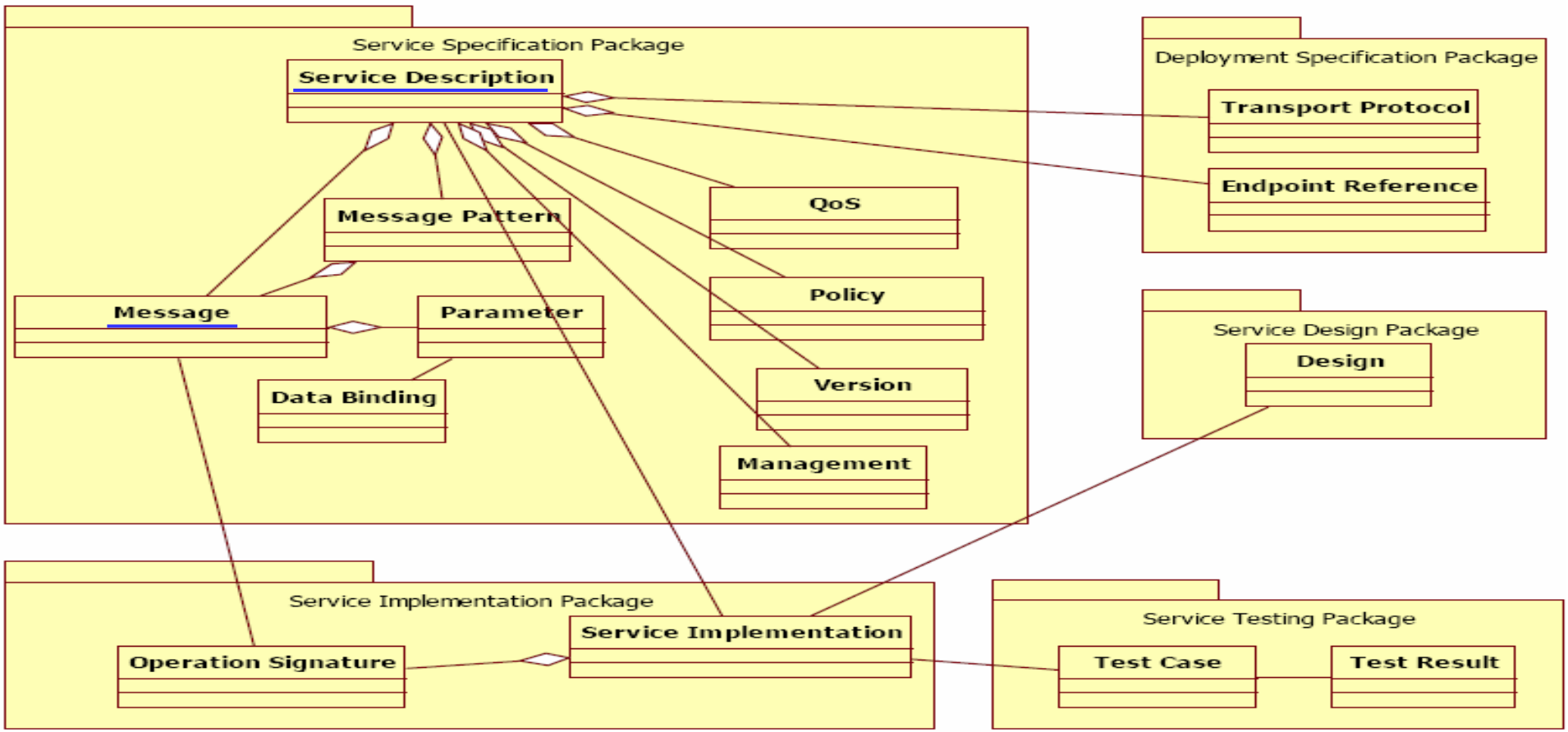


Figure 34: Service Creation information entities

7.2.1 Service Creation Functionality and Information Model

- **Textual Description of Information Entities**

Information entity name: Service Description

Description: This bundles all information created during service creation. It accompanies the service code and has consistent with it. Here, we assume it relates to the implementation of one service operation. Optionally, the service description can contain information for deployment.

Related Entities (optional): NA

Related Functionality (optional): NA

Information entity name: Message

Description: Describes a single message sent to a service. A message and its parameters should be consistent with the operation signature.

III. Conclusion

Conclusion

- A holistic view: a framework in which all elements of the conceptual model can be integrated
- Separation of structure, behavior and functions
- Different levels of granularity
- Perspectives of different stakeholders

  => Recognize the socio-economic environment in which the RA will be deployed.
- Two approaches
  - Top-down view: architectural overview
  - Bottom-up view: functionalities

Future Work

- Integration and consolidation is still in progress.
- This version is not fully consistent, coherent or complete.
  E.g. composition
- Proven recurring and innovative patterns for service-oriented systems.

References

[Kang 09] Sungwon Kang, “Service Architecture for the Future Internet – Software Engineering Perspective”, KRNET 2009, June 2009.

[Andressen 07] Andreessen, Marc; *The three kinds of platforms you meet on the Internet*; Blog;Sep 16, 2007; http://blog.pmarca.com/2007/09/the-three-kinds.html

[W3C 046] Booth, D.; Haas, H.; McCabe, F.; Newcomer, E.; Champion, M.; Ferris, C. & Orchard, D.: *Web Services Architecture*. Technical Report, W3C, 2004.

[Colombo 05] Colombo, M.; Nitto, E. D.; Penta, M. D.; Distante, D.; Zuccalà, M.: *Speaking a Common Language: A Conceptual Model for Describing Service-Oriented Systems*. Proc.3rd Int’l Conf. on Service Oriented Computing (ICSOC 2005), Springer.

[OASIS 06] MacKenzie, C. M.; Laskey, K.; McCabe, F.; Brown, P. F.; Metz, R.; *OASIS Reference Model for Service Oriented Architecture 1.0; Committee Specification1*; http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf, 2006.