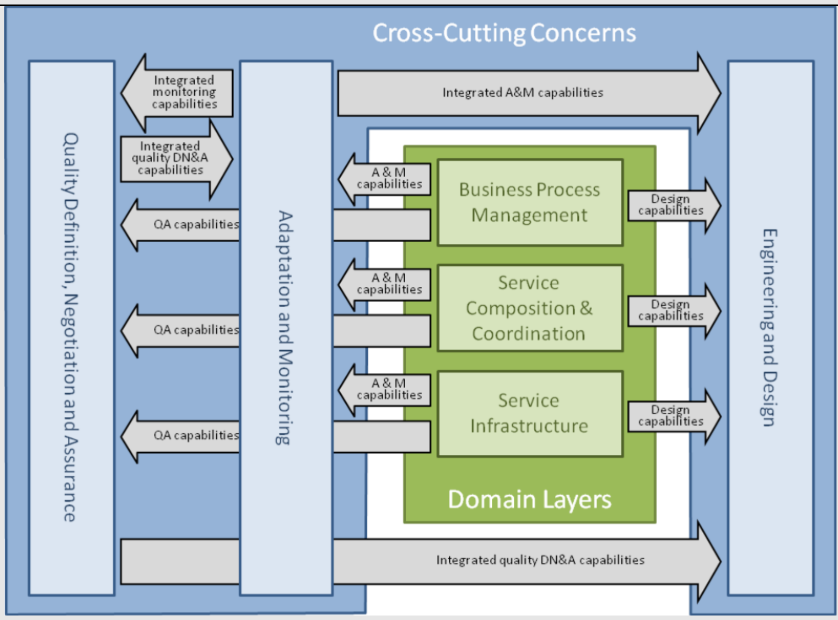
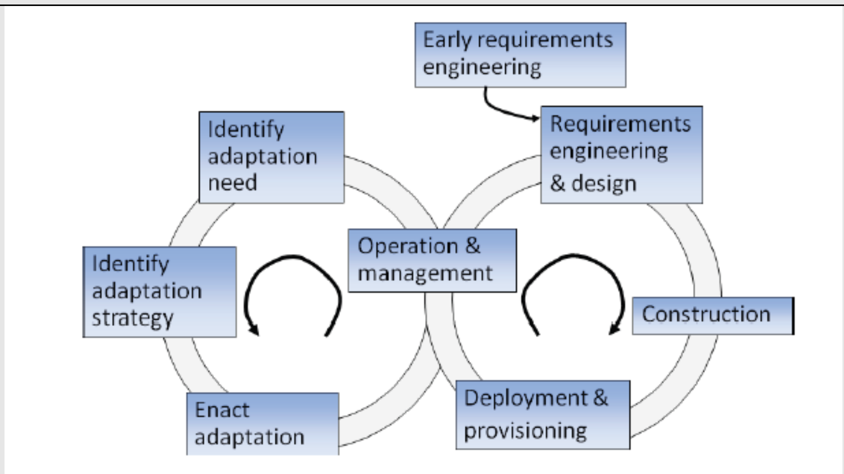
1. Views

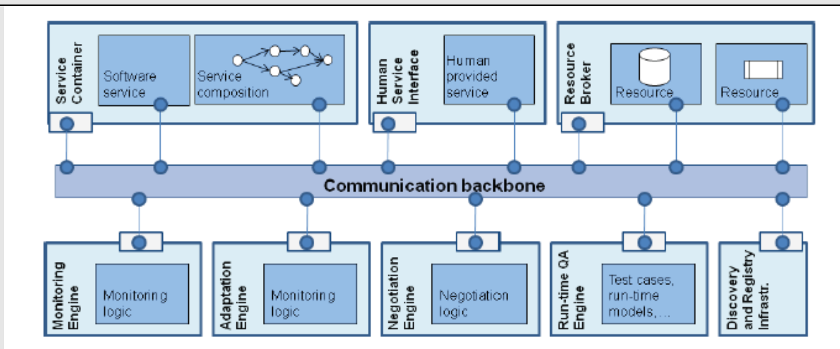
1.1. Views

Name	Conceptual Research Framework
Synopsis	The conceptual research framework provides a high-level conceptual view of the S-Cube research activities.
Authors	-
Description	 The conceptual research framework is the core element in the definition of the IRF. Its aim is to organise the joint research activities within S-Cube by providing a high-level conceptual architecture for the principles and methods for engineering service-based applications, as well as for the technologies and mechanisms which are used to realize those applications. The framework consists of six components, which are in 1-to-1 relation with the six research work-packages of the network. Moreover, the framework distinguishes between the horizontal components corresponding to the "traditional" domain layers of a SBA, i.e., "Service Infrastructure", "Service Composition and Coordination", and "Business Process Management", and the vertical components, which correspond to the cross-cutting issues addressed by the project, namely "Engineering and Design", "Adaptation and Monitoring", and "Quality Definition, Negotiation and Assurance". We note that the distinction between the two kinds of components is one of the core elements of the S-Cube approach. Indeed, an element that makes the S-Cube framework unique when compared to the traditional "layered" approach is that the framework systematically addresses cross-cutting issues. The framework sets out to make explicit the knowledge of the horizontal layers that is relevant for these cross-cutting issues, and that currently is mostly hidden in languages, standards, mechanisms, and so on that are defined and investigated in isolation at the different layers. More precisely, the approach underlying the framework is that the domain layers offer (design, monitoring, adaptation, verification) capabilities that are relevant for the cross-cutting issues. The research

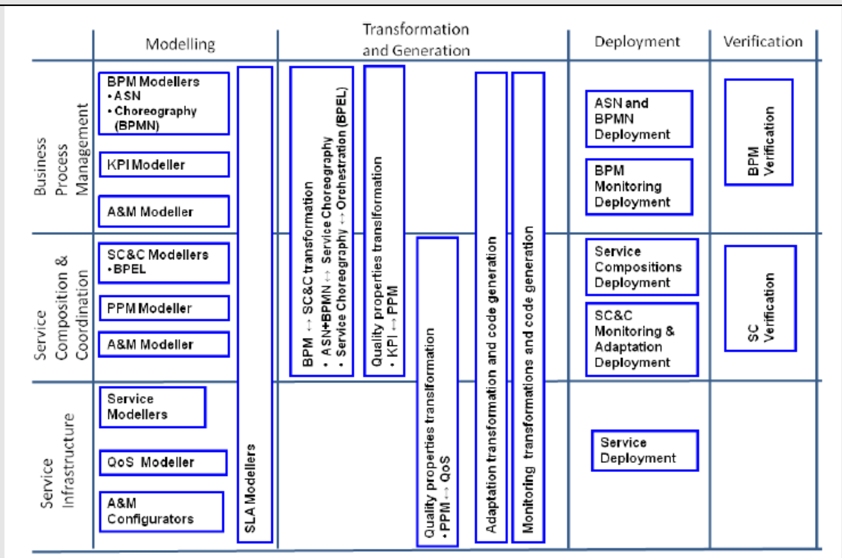
	efforts in the vertical components are responsible of defining over-arching principles and methodologies for addressing cross-cutting issues by exploiting in suitable ways the capabilities exposed by the horizontal components.
References	M. Pistore, R. Kazhamiakin, A. Bucchiarone (Eds.). *Integration Framework Baseline*, S-Cube Deliverable CD-IA-3.1.1
Glossary	SBA, Service Infrastructure, Service Composition and Coordination, Business Process Management, Adaptation, Monitoring, Design for Adaptation, Quality Attributes, Quality of Service Negotiation, Quality of Service-based Adaptation.
Keywords	-
	-

Name	Reference life-cycle
Synopsis	The reference life-cycle shows how the different research activities contribute to support a coherent design and run-time life-cycle of an SBA
Authors	-
Description	 The key purpose of this view is to complement the static view provided by the conceptual research framework and to relate the research efforts undertaken by the network to the different phases of the life of the SBAs. It is composed of two main cycles: the one on the right hand side corresponds to the *classical application design*, deployment and provisioning; the one on the left hand side corresponds to the *adaptation perspectives*. By adopting this two-cycle approach, not only must applications undergo the transition between the runtime operation and the analysis and the design phases in order to be continuously improved and updated (i.e., evolution), but they must provide mechanisms that, during runtime, continuously and automatically a) detect new problems, changes, and needs for adaptation, b) identify possible adaptation strategies, and c) enact them. These three steps (on the left hand side) lead to deployment and provisioning of the modified application. The identification of the changes in the environment and of the problems in the execution of the SBA (e.g., failures) is obtained through monitoring and run-time quality assurance. The monitoring activity triggers the iteration of the adaptation cycle, whose effects is to inject changes directly into the application being operated and managed.

References	M. Pistore, R. Kazhamiakin, A. Bucchiarone (Eds.). Integration Framework Baseline, S-Cube Deliverable CD-IA-3.1.1
Glossary	Life cycle model, Requirements Engineering, Design Principles, Adaptation, Evolution, Adaptation Strategy, Adaptation Requirements, Monitoring, Monitoring Requirements, Software Quality Assurance
Keywords	

<table>
<tr><td>Name</td><td>Logical Run-Time Architecture</td></tr>
<tr><td>Synopsis</td><td>The logical run-time architecture, shows how joint research activities undertaken by S-Cube are aligned</td></tr>
<tr><td>Authors</td><td>-</td></tr>
<tr><td>Description</td><td>


The key purpose of this view is to guarantee a coherent picture for all run-time mechanisms studied by S-Cube, that is, for all mechanisms that are adopted in the operation and management phases and in all the left hand side of the reference life-cycle. The proposed run-time architecture is service-oriented, that is, it assumes that all the run-time mechanisms and components are realized as services and are exposed on the same communication backbone. We can have two kinds of services called core and application-specific services. The core services are middleware services that the run-time architecture provides to all SBA in order to support the different aspects of the SBA execution. Examples of such core services are a discovery service, an engine for executing service compositions, or an engine for monitoring the behaviour of a SBA or the performance of a business network. These core services may belong to the "Adaptation and Monitoring" and "Quality Definition, Negotiation and Assurance" components. Some of these core services act as containers for application-specific services, i.e., services that are specific of the SBA in execution, and that encapsulate part of the application-specific logic. This is the case of the engine for executing service compositions. Other core services contain other parts of the application-specific logics, which are however not exposed as services. This is the case, for instance, of monitoring engine, which will contain the application-specific properties to be monitored.</td></tr>
<tr><td>References</td><td>M. Pistore, R. Kazhamiakin, A. Bucchiarone (Eds.). Integration Framework Baseline, S-Cube Deliverable CD-IA-3.1.1</td></tr>
<tr><td>Glossary</td><td>Service Runtime, Service Registry, Service Discovery, Autonomy, Self-*, Dynamic Binding, Quality of Service-based Adaptation, Adaptation, Monitoring. Service Composition, Monitoring Infrastructure.</td></tr>
<tr><td>Keywords</td><td>-</td></tr>
</table>

<table>
<tr><td>Name</td><td>Logical Design Environment</td></tr>
<tr><td>Synopsis</td><td>The logical design environment aims at providing a logical description of the foreseen design environment for SBAs.</td></tr>
<tr><td>Authors</td><td>-</td></tr>
<tr><td>Description</td><td>

This view is complementary to the run-time architecture and its purpose is to provide a context where to place the envisioned techniques and mechanisms that support the analyst and designer in the design of a SBA. The design environment covers phases corresponding to the right hand side of the life-cycle view, from early requirements engineering to deployment and provisioning. By adopting this logical design environment, the efforts undertaken by the different research work-packages define a coherent picture that supports the different aspects of the SBA design and engineering. The design environment should provide different functionalities (i.e. Modelling, Transformation and Generation, Deployment and Verification) at different application layers (i.e., Business Process Management, Service Composition and Coordination and Service Infrastructure). Moreover we should have also cross-layer techniques that span over more application layers.</td></tr>
<tr><td>References</td><td>M. Pistore, R. Kazhamiakin, A. Bucchiarone (Eds.). Integration Framework Baseline, S-Cube Deliverable CD-IA-3.1.1</td></tr>
<tr><td>Glossary</td><td>Business Process Modeling, Service Composition, Service Deployment, Verification, Service Level Agreement, Adaptation, Monitoring, Key Performance Indicator.</td></tr>
<tr><td>Keywords</td><td>-</td></tr>
</table>

1.2. Elements

1.2.1 Elements of the Conceptual Research Framework

Name	Adaptation and Monitoring
Synopsis	This element comprises research on languages and methods for monitoring and managing the adaptation of a SBA.
View	Conceptual Research Framework
Authors	-
Description	This element covers the issues related to the adaptation of a SBA.

	Specifically, this comprises languages and methods for defining adaptation goals and different adaptation strategies, which are triggered by monitoring events. An example for an adaptation technique that falls into the responsibility of this aspect is a strategy that correlates the monitoring events across the functional layers, thereby avoiding conflicting adaptations, or the one that aims to predict the potential SBA problems and perform adaptation activities pro-actively.	
Related elements	***Element***	Integrated A&M capabilities
	Relation	Provides
	Element	A&M capabilities
	Relation	Uses
	Element	Integrated quality DN&A capabilities
	Relation	Uses
	Element	Integrated Monitoring capabilities
	Relation	Provides
References	J. Hielscher, A. Metzger, R. Kazhamiakin (Eds.), *Taxonomy of Adaptation Principles and Mechanisms*, S-Cube Deliverable CD-JRA-1.2.2.	
Glossary	Service adaptation, service monitoring	
Keywords	Service-based Applications, Adaptation, Monitoring	

Name	Engineering and Design	
Synopsis	This element comprises research on principles and methods for engineering and design of a SBA as well as its adaptation and monitoring tools.	
View	Conceptual Research Framework	
Authors	-	
Description	This element covers the issues related to the life-cycle of services and SBAs. This includes principles and methods for identifying, designing, developing, deploying, finding, applying, provisioning, evolving, and maintaining services, while exploiting novel technologies from the functional layers. In particular, this aspect focuses on the quality of the SBA development process, on the roles and placement of the contextual properties of SBAs and human involvement, and on exploiting future service search engines for bottom-up SBA design.	
Related elements	***Element***	Design capabilities
	Relation	Uses
	Element	Integrated quality DN&A capabilities
	Relation	Uses
	Element	Integrated A&M capabilities
	Relation	Uses
References	Vasilios Andrikopoulos (Ed.), *Separate design knowledge models for software engineering and service based computing*, S-Cube Deliverable CD-JRA-1.1.2.	
Glossary	Service Engineering, Service Design, SBA Engineering, SBA Design	
Keywords	Service Engineering, Design	

Name	Quality Definition, Negotiation and Assurance
Synopsis	This element comprises research on principles and methods for quality attributes and SLAs of SBA.
View	Conceptual Research Framework

Authors	-	
Description	This element involves principles and methods for defining, negotiating and ensuring quality attributes and Service Level Agreements (SLAs). Negotiating quality attributes requires understanding and aggregating quality attributes across the functional layers as well as agreeing on provided levels of quality. To ensure agreed quality attributes, techniques which are based on monitoring, testing or static analysis (e.g., model checking) are employed and extended by novel techniques exploiting future technologies (e.g., Web 2.0).	
Related elements	***Element***	Integrated quality DN&A capabilities
	Relation	Provides
	Element	QA capabilities
	Relation	Uses
	Element	Integrated monitoring capabilities
	Relation	Uses
References	A. Gehlert, A. Metzger (Eds.), *Quality Reference Model for SBA*, S-Cube Deliverable CD-JRA-1.3.2.	
Glossary	Quality Attributes, Service Level Agreement, Negotiation	
Keywords	Quality Assurance, SLA	

Name	Business Process Management	
Synopsis	This element comprises research on the "Business Process Management" functional layer of SBA.	
View	Conceptual Research Framework	
Authors	-	
Description	This element addresses the aspects related to the modeling, designing, deploying, monitoring and managing of service networks, business processes and Key Performance Indicators (KPIs).	
Related elements	***Element***	A&M Capabilities
	Relation	Provides
	Element	Design Capabilities
	Relation	Provides
	Element	QA Capabilities
	Relation	Provides
References	Branimir Wetzstein (Ed.), *Initial models and mechanisms for quantitative analysis of correlationsbetween KPIs, SLAs and underlying business processes,* S-Cube Deliverable CD-JRA-2.1.2	
Glossary	Key Performance Indicator, Agile Service Network, Business Activity, Business Process	
Keywords	-Business Process Management	

Name	Service Composition and Coordination
Synopsis	This element comprises research on the "Service Composition and Coordination" functional layer of SBA.
View	Conceptual Research Framework
Authors	-
Description	This element focuses on novel service composition languages and techniques. In particular, it provides mechanisms to adapt and monitor service compositions.

Related elements	***Element***	A&M Capabilities
	Relation	Provides
	Element	Design Capabilities
	Relation	Provides
	Element	QA Capabilities
	Relation	Provides
References	Martin Treiber (Ed.), *Models and Mechanisms for Coordinated Service Compositions*, S-Cube Deliverable CD-JRA-2.2.2	
Glossary	Service, Service Composition, Process Performance Metrics	
Keywords	-Service Composition and Coordination	

Name	Service Infrastructure	
Synopsis	This element comprises research on the "Service Infrastructure" functional layer of SBA.	
View	Conceptual Research Framework	
Authors	-	
Description	This element studies a high-performance execution platform supporting adaptation and monitoring of SBAs (e.g., self-* mechanisms). The platform provides a set of core services, like service registries, discovery capabilities, and virtualization services to the other layers.	
Related elements	***Element***	A&M Capabilities
	Relation	Provides
	Element	Design Capabilities
	Relation	Provides
	Element	QA Capabilities
	Relation	Provides
References	Jean-Louis Pazat (Ed.), *Basic Requirements for self-healing services and decision support for local adaptation*, S-Cube Deliverable CD-JRA-2.3.2	
Glossary	Service Realization, Resources, Service Discovery and Selection, Service Registry, Service Metrics	
Keywords	Execution Platform	

Name	Integrated A&M capabilities
Synopsis	This element comprises research on defining overall, cross-layer monitoring and adaptation strategies.
View	Conceptual Research Framework
Authors	-
Description	This element is responsible of defining overall, cross-layer monitoring and adaptation strategies that are then realized by exploiting the capabilities offered by the domain layers. These overall monitoring and adaptation strategies are in turn capabilities that the "Adaptation and Monitoring" component offers to the "Engineering and Design" component. Indeed, the knowledge of the capabilities and mechanisms for monitoring and adaptation, which will be available at run time, is crucial at design time in order to design and construct a SBA that is able to exploit those capabilities. Indeed, by "design for monitoring" and "design for adaptation" we refer to the possibility of designing SBAs whose behavior relies on a full exploitation of the monitoring and adaptation capabilities offered by the framework.

Related elements	***Element***	Adaptation and Monitoring
	Relation	Uses
	Element	Engineering and Design
	Relation	Provides
References	-	
Glossary	Cross-layer Adaptation, Cross-layer Monitoring	
Keywords	Adaptation, Monitoring	

Name	Design Capabilities	
Synopsis	This element comprises research on languages and mechanisms for designing SBA layers.	
View	Conceptual Research Framework	
Authors	-	
Description	Each functional layer provides capabilities to the "Engineering and Design" of SBAs; these capabilities correspond to languages and mechanisms for modeling and specifying those aspects of a SBA that are specific to a domain layer. For example, the "Business Process Management" layer offers capabilities for modeling business processes (e.g., BPMN, or UML Activity Diagrams), as well as for specifying aspects related to the integration and execution of these business processes. The "Service Composition and Coordination" layer provides capabilities for modeling the single services, as well as service compositions (e.g., WSDL, BPEL). Finally, the "Service Infrastructure" layer provides capabilities for service discovery, for accessing service registries, and for managing service execution.	
Related elements	***Element***	Business Process Management
	Relation	Uses
	Element	Service Composition and Coordination
	Relation	Uses
	Element	Service Infrastructure
	Relation	Uses
	Element	Engineering and Design
	Relation	Provides
References	Vasilios Andrikopoulos (Ed.), *Separate design knowledge models for software engineering and service based computing*, S-Cube Deliverable CD-JRA-1.1.2.	
Glossary	Business Process Design, Service Composition Design, Service Design	
Keywords	-Design	

Name	QA Capabilities
Synopsis	This element comprises research on quality assurance capabilities.
View	Conceptual Research Framework
Authors	-
Description	Each domain layer provides capabilities that are exploited to achieve an end-to-end, cross-layer quality definition and assurance for the SBA. At the "Business Process Management" layer, these capabilities correspond to understanding how to express the relevant quality at tributes (e.g., KPIs) and the possibility of doing a static verification of the business process models, as well as of running simulations in order to predict and analyze the expected behavior of these models. At the "Service Composition and Coordination" layer, the capabilities cover

	understanding the relevant quality attributes and how to do both static verification and simulation of single services and of service compositions. At this layer, capabilities may also concern the possibility of testing the service composition. The "Service Infrastructure" layer, finally, provides capabilities for expressing relevant infrastructural quality attributes, and capabilities for exploiting the infrastructures for running simulations or to test cases on SBAs.	
Related elements	***Element***	Business Process Management
	Relation	Uses
	Element	Service Composition and Coordination
	Relation	Uses
	Element	Service Infrastructure
	Relation	Uses
	Element	Quality Definition, Negotiation and Assurance
	Relation	Provides
References	A. Gehlert, A. Metzger (Eds.), *Quality Reference Model for SBA*, S-Cube Deliverable CD-JRA-1.3.2.	
Glossary	Quality Attributes, Testing	
Keywords	-Quality Analysis	

Name	Integrated quality DN&A capabilities	
Synopsis	This element comprises research on integrated quality definition, assurance and negotiation capabilities.	
View	Conceptual Research Framework	
Authors	-	
Description	These capabilities of the "Quality Definition, Negotiation and Assurance" component are offered to the "Engineering and Design" component, so that they can be exploited during the design and construction of a SBA. More precisely, these capabilities concern languages that can be exploited for defining the expected quality of a SBA; they concern mechanisms for negotiating quality attributes between service consumers and providers; and mechanisms for static analysis, simulation and testing of SBAs. These capabilities are also offered to the "Adaptation and Monitoring" component, for the purpose of enabling pro-active adaptation on the basis of the analysis of the past, current and future quality of the SBA. Indeed, pro-active adaptation will exploit the testing, simulation and quality prediction mechanisms studied by the "Quality Definition, Negotiation and Assurance" component.	
Related elements	***Element***	Quality Definition, Negotiation and Assurance
	Relation	Uses
	Element	Engineering and Design
	Relation	Provides
References	-	
Glossary	Quality Attributes, Negotiation, Quality Assurance	
Keywords	-Quality Assurance, Negotiation	

1.2.2 Elements of the Reference life-cycle

Name	Early Requirements Engineering	
Synopsis	This element comprises research on requirements engineering with the objective to analyze and understand the problem by studying existing organizational and business setting.	
View	Reference Life-cycle	
Authors	-	
Description	This element related to the requirements expression in terms of high-level concepts that correspond to the actors that are relevant in the setting, and to their goals, needs, and mutual dependencies, without any reference to the system-to-be. This element studies requirements that exist *a priori* in the organizational and business setting, and that are hence largely independent from the solution. They are collected from the stakeholders and cover not only the functional aspects; they should cover also quality expectations, adaptation requirements and expectations of the actors.	
Related elements	***Element***	Requirements Engineering & Design
	Relation	Beforehand
References		
Glossary	Requirement, Requirements Engineering, Adaptation Requirements, Monitoring Requirements	
Keywords	Requirement Engineering	

Name	Requirements Engineering and Design	
Synopsis	This element comprises research on usual requirements engineering and design taking into account both functional and quality aspect of the SBA.	
View	Reference Life-cycle	
Authors	-	
Description	The main objectives of this element are similar to the ones of any classical software development, there are some peculiarities that make development of SBAs different from others. The first difference is that the availability of services drives the requirement engineering (RE) as well as the design phase in such a way that the usage of these services is possible. The second difference is that RE and design of a SBA have to be performed taking into account the three domain layers that define such an application. A third difference in that the SBA has to be built to be able to react to new and/or critical conditions by triggering proper adaptation actions. It means that new classes of requirements have to be elicited and understood. These include adaptation and monitoring requirements. At the level of design this means that proper adaptation strategies have to be designed together with monitoring mechanisms that allow the adaptation needs to be identified.	
Related elements	***Element***	Construction
	Relation	Beforehand
	Element	Early Requirements Engineering
	Relation	Next
References	-	
Glossary	Requirements Engineering, Service-Oriented Requirements Engineering, Adaptation Requirement and Objectives, Monitoring Requirements, Business Process Modelling, Service-based Applications.	

Keywords	-Requirement Engineering, Design

Name	Construction	
Synopsis	This element comprises research on the SBA construction integrating different services.	
View	Reference Life-cycle	
Authors	-	
Description	This element of the reference life-cycle covers the issues related to the integration of different services. This means that for that for establishing the desired end-to-end quality of those SBAs, contracts between the service providers and the service requestors on quality aspects of services have to be established. Typically this requires some form of SLA negotiation and agreement. the service composition construction should cover not only the functional requirements, but also the QoS aspects and the adaptability requirements for the SBA. In addition to the service composition, the construction phase will also realize all those mechanisms that are necessary for supporting the monitoring, adaptation, and quality assurance of the SBA.	
Related elements	***Element***	Requirements Engineering and Design
	Relation	Beforehand
	Element	Deployment and Provisioning
	Relation	Next
References	-	
Glossary	Service Composition, Service Coordination, Service Orchestration, Service Choreography, Quality of Service Negotiation, Service Level Agreement, Quality of Service-based Application, Adaptation Mechanisms, Monitoring Mechanisms.	
Keywords	-Construction	

Name	Deployment and Provisioning	
Synopsis	This element comprises all the activities needed to make the SBA available to its users.	
View	Reference Life-cycle	
Authors	-	
Description	This element covers the issues related to the publishing of the SBA. It can be itself a service: in this case, a proper description of its interface should be provided and published on some registry. Moreover semantic service descriptions of various kind should be proposed. These include the description of the QoS characteristics of a service and enable for the definition of SLAs. In the case of adaptable SBAs, we could imagine that QoS and SLA information includes data on the adaptation characteristics of the SBA.	
Related elements	***Element***	Construction
	Relation	Beforehand
	Element	Operation and Management
	Relation	Next
References	-	
Glossary	Service, Service-based Application, Automatic Service Deployment,	

	Semantic Web Services, Service Level Agreement, Quality of Service-Based Adaptation.
Keywords	-Deployment

Name	Operation & Management	
Synopsis	This element is used to specify all the activities needed for operating and managing a SBA	
View	Reference Life-cycle	
Authors	-	
Description	This element covers the issues related to the activities that govern the correct execution of SBAs and related services by ensuring that they respect the expected QoS level during execution. In this context, the identification of problems in the SBA (e.g., failures) plays a fundamental role. This identification is obtained by means of monitoring mechanism and, more in general, of mechanisms for run-time quality assurance. These mechanisms are able to detect failures, or critical conditions requiring the triggering of an adaptation mechanism needed to adapt SBAs.	
Related elements	***Element***	Deployment and Provisioning
	Relation	Beforehand
	Element	Identify Adaptation Need
	Relation	Next
	Element	Requirements Engineering and Design
	Relation	Next
References	-	
Glossary	Service Governance, Service runtime, Service-based application, Service runtime management process, Service Analysis, Monitoring mechanisms, Failure, Error.	
Keywords	-Management, Failures, Execution, Monitoring	

Name	Identify Adaptation Need	
Synopsis	This element comprises the decision on the needs for the SBA to adapt.	
View	Reference Life-cycle	
Authors	-	
Description	This element provides way to use information gathered during execution, the observation of the properties of the application, and the context of SBA constitute the elements on which the decision on the need for the SBA to adapt is based. Such decision may be automatically taken on the basis of monitoring requirements derived from adaptation requirements, or it may require human intervention (end user, system integrator, application manager). Moreover, such decision may be taken in a reactive way, when the problem has already occurred, or in a pro-active way, where the need is to prevent a potential problem.	
Related elements	***Element***	Operation and Management
	Relation	Beforehand
	Element	Identify Adaptation Strategy
	Relation	Next
References	-	

Glossary	Adaptation, Self-adaptation, Human Computer Interaction, Monitoring Requirements, Adaptation Requirements, Reactive Adaptation, Proactive Adaptation.
Keywords	-Adaptation

Name	Identify Adaptation Strategy	
Synopsis	This element covers the issues to define a set of possible adaptation strategies and related them with the adaptation needs.	
View	Reference Life-cycle	
Authors	-	
Description	This element covers the issues related to the identification and selection of adaptation strategy and their relation with adaptation needs. The decision on what strategy use at run-time may be may be automatic if either the SBA or the execution platform decide the action to perform, or it can be done by a human user. Among the possible adaptation strategies we mention service substitution, SLA re-negotiation, SBA re-configuration or service re-composition.	
Related elements	***Element***	Identify Adaptation Need
	Relation	Beforehand
	Element	Enact Adaptation
	Relation	Next
References	-	
Glossary	Adaptation Strategy, Self-Adaptation, Human Computer Interaction	
Keywords	-Adaptation	

Name	Enact Adaptation	
Synopsis	This element covers the issues to define a set of adaptation mechanisms that implement adaptation strategy and its run-time activation.	
View	Reference Life-cycle	
Authors	-	
Description	This element of the reference life-cycle covers the issues related to the implementation of adaptation mechanisms that realize adaptation strategies. For example service substitution, re-configuration, re-composition may be obtained using automated service discovery and dynamic binding mechanisms, while re-composition may be achieved using existing automated service composition techniques. As these examples show, the enactment of an adaptation strategy usually requires the exploitation of mechanisms provided by different layers, in particular by the "Service Composition and Coordination" and by the "Service Infrastructure" layers.	
Related elements	***Element***	Identify Adaptation Strategy
	Relation	Beforehand
	Element	Operation and Management
	Relation	Next
References	-	
Glossary	Adaptation Mechanism, Service Discovery, Dynamic Binding, Service composition.	
Keywords	Adaptation, Run-Time	

1.2.3 Elements of the Logical Run-Time Architecture

Name	Service Container	
Synopsis	In the run-time architecture of S-Cube services are deployed in containers called "Service Containers".	
View	Logical Run-Time Architecture	
Authors	-	
Description	The run-time architecture is service-oriented, it means that all the run-time mechanisms and components are realized as services and are exposed on the same communication backbone. We distinguish between *core* and *application-specific* services. The core services are middleware that the run-time architecture provides to all SBA in order to support the different aspects of the SBA execution (i.e., discovery service, an engine for monitoring the behaviour of a SBA, etc..). Application-specific services are specific service of the SBA in execution, and that encapsulate part of the application-specific logic. All these kind of services are deployed onto a container and the communication backbone allows accessing both services deployed within the containers.	
Related elements	***Element***	Software Service
	Relation	Contains
	Element	Service Composition
	Relation	Contains
	Element	Communication Backbone
	Relation	Is exposed to
References	-	
Glossary	Service	
Keywords	Service	

Name	Human Service Interface	
Synopsis	This element provides the fact that we can have also human-services that can be integrated in the SBAs.	
View	Logical Run-Time Architecture	
Authors	-	
Description	Human Computer Interaction is the study of the interaction between humans and computers (in their broadest sense, including computerized devices and large scale computer systems as well as stand-alone computers). It is concerned with the design, evaluation and implementation of interactive computing systems which it aims to make more usable and useful for human use. With this element we should be able to provide interfaces among Humans that provide services and the SBA.	
Related elements	***Element***	Human Provided Service
	Relation	Contains
	Element	Communication Backbone
	Relation	Communicates with
References	-	
Glossary	Human Computer Interaction	
Keywords	HCI	

Name	Resource Broker	
Synopsis	This element provides the way to select resources in automatic way during the SBA execution.	
View	Logical Run-Time Architecture	
Authors	-	
Description	Brokering is used to automate resource selection. For example the role of grid brokers is to provide an interface for the users to access grids, accept and understand user jobs, discover resources, find a suitable resource for a job with scheduling, submit jobs to resources and provide the output of the jobs to the user. The S-Cube run-time architecture have to provide	
Related elements	***Element***	Resource
	Relation	Contains
	Element	Communication Backbone
	Relation	Communicates with
References	-	
Glossary	Grid Brokering	
Keywords	Resource Management	

Name	Monitoring Engine	
Synopsis	To execute monitoring the run-time architecture must provide a monitoring engine.	
View	Logical Run-Time Architecture	
Authors		
Description	With monitoring we mean a process of collecting and reporting relevant information about the execution and evolution of SBA. Such information, namely monitoring events, represents evolution of SBA and changes in the environment. Run-time monitoring has to be supported by monitoring engines that should be included in the infrastructure. Moreover a precise monitoring logic have to be provided to specify monitoring properties.	
Related elements	***Element***	Monitoring Logic
	Relation	Contains
	Element	Communication Backbone
	Relation	Communicates with
References	-	
Glossary	Monitoring	
Keywords	Monitoring	

Name	Adaptation Engine	
Synopsis	This element provides the way to execute different types of adaptation during the SBA execution	
View	Logical Run-Time Architecture	
Authors	-	
Description		
Related elements	***Element***	Adaptation Logic
	Relation	Contains

	Element	Communication Backbone
	Relation	Communicates with
References	Adaptation is the process of modifying an SBA in order to satisfy new requirements and to fit new situations dictated by the environment on the basis of adaptation strategies designed by the system integrator. The run-time architecture must provide adaptation engine that realizes the different adaptation strategies.	
Glossary	Adaptation, Adaptation Strategy	
Keywords	Adaptation	

Name	Negotiation Engine	
Synopsis	This element provides the way to execute negotiation among parties that are involved in a SBA	
View	Logical Run-Time Architecture	
Authors	-	
Description	Negotiation is a process carried out between Service Providers and Requesters by formulating, exchanging and evaluating a number of Agreement proposals that may end with the stipulation of a contract in the form of an Service Level Agreement. The S-Cube run-time architecture have to deploy a Negotiation Engine able to execute this process using a precise Negotiation Logic.	
Related elements	***Element***	Negotiation Logic
	Relation	Contains
	Element	Communication Backbone
	Relation	Communicates with
References	-	
Glossary	Negotiation, Quality of Service Negotiation	
Keywords	Negotiation	

Name	Run-time QA Engine	
Synopsis	This element provides the way to execute quality analysis techniques on the SBA.	
View	Logical Run-Time Architecture	
Authors	-	
Description	To assure the desired quality of a service-based application, two complementary strategies can be employed: constructive and analytical quality assurance. Where the goal of constructive quality assurance is to prevent the introduction of faults (or defects) while the artifacts are created (in the sense of 'correctness by construction'), the goal of analytical quality assurance is to uncover faults in the artifacts after they have been created. The run-time architecture should provide an engine able to verify the quality of SBAs using different techniques like Testing, Statical Analysis, Monitoring, etc..	
Related elements	***Element***	Test cases
	Relation	Contains
	Element	Run-time Models
	Relation	Contains
	Element	Communication Backbone
	Relation	Communicates with
References	-	

Glossary	Quality of Service-based Application	
Keywords	Quality	

Name	Discovery and Registry Infrastructure	
Synopsis	This element provides the way to discover and add services in the S-Cube platform.	
View	Logical Run-Time Architecture	
Authors	-	
Description	A Service Registry is a repository that contains service related meta information (e.g. Web service descriptions). The S-Cube run-time architecture has to provide mechanisms to find new services and add them in the deployed registry. The purpose of this element is to capture the basic requirements for decision support in service execution, deployment and runtime management for services including core services such as discovery and registries.	
	Element	Communication Backbone
	Relation	Communicates with
References	Jean-Louis Pazat (Ed.), *Basic Requirements for self-healing services and decision support for local adaptation*, S-Cube Deliverable CD-JRA-2.3.2	
Glossary	Service Registry	
Keywords	Service Discovery, Servie Registry	

Name	Communication Backbone	
Synopsis	This element has the objective to support the communication among any kind of services.	
View	Logical Run-Time Architecture	
Authors	-	
Description	This element supports the communication among any kind of services, regardless of whether they are core services or application-specific services.	
Related elements	***Element***	Service Container
	Relation	Communicates with
	Element	Human Service Interface
	Relation	Communicates with
	Element	Resource Broker
	Relation	Communicates with
	Element	Monitoring Engine
	Relation	Communicates with
	Element	Adaptation Engine
	Relation	Communicates with
	Element	Run-time QA Engine
	Relation	Communicates with
	Element	Negotiation Engine
	Relation	Communicates with
References	-	
Glossary	Service adaptation, service monitoring	
Keywords	Communication	

1.2.4 Elements of the Logical Design Environment

Name	Modelling Techniques	
Synopsis	The element covers the issues to define a set of possible techniques for modelling a SBA.	
View	Logical Design Environment	
Authors	-	
Description	The objective of this element is to provide a set of techniques for modelling a SBA at the different domain layers (i.e., Business Process Management, Service Composition and Coordination and Service Infrastructure), as well as for modelling the cross-cutting aspects of a SBA. More precisely, for each layer we must be able to provide techniques for modelling our SBA, for modelling the indicators that are used to evaluate the quality of the SBA, and for modelling the monitoring and adaptation aspects that are used to control and adapt the application at run-time.	
Related elements	***Element***	Business Process Management Layer
	Relation	Used by
	Element	Service Composition and Coordination Layer
	Relation	Used by
	Element	Service Infrastructure Layer
	Relation	Used by
References	-	
Glossary	Business Process Modelling, Service Design, Design for Adaptation, Design for Monitoring, Design Principles,	
Keywords	Modelling	

Name	Transformation and Generation Techniques	
Synopsis	This element has the objective to provide techniques to realize model-to-model transformations.	
View	Logical Design Environment	
Authors	-	
Description	This element has the objective to provide techniques that allow for transforming high-level models of the behaviour of a SBA into lower-level executable models, and vice-versa. They include for instance transformation techniques that generate BPEL code from BPMN, or that transform choreographies into orchestrations, and vice-versa. Moreover, they contain techniques to transform high-level specifications of quality properties into lower-level specifications of the same properties and vice-versa, e.g. KPI to/from PPM models. Finally, they include techniques for generating code in automatic way from the design models, as well as mechanisms to transform adaptation and monitoring specifications from one layer to another one. An example are mechanisms for transforming the monitoring and adaptation strategies specified by the designer into engine mechanisms that the service infrastructure will provide.	
Related elements	***Element***	Business Process Management Layer
	Relation	Used by

	Element	Service Composition and Coordination Layer
	Relation	Used by
	Element	Service Infrastructure Layer
	Relation	Used by
References	-	
Glossary		
Keywords	-Model Tranformation, Code Generation	

Name	Deployment Techniques	
Synopsis	The element comprises techniques for deploying artifacts of a SBA specification.	
View	Logical Design Environment	
Authors	-	
Description	This element provides a set of techniques for deploying the artifacts corresponding to a SBA specification at the different layers. This corresponds to deploying service networks, as well as the real/physical deployment of services on a service infrastructure. This functionality also covers deployment techniques for the adaptation and monitoring mechanisms and specifications.	
Related elements	***Element***	Business Process Management Layer
	Relation	Used by
	Element	Service Composition and Coordination Layer
	Relation	Used by
	Element	Service Infrastructure Layer
	Relation	Used by
References	-	
Glossary	Automatic Service Deployment, Manual Service Deployment, Automated Service Composition.	
Keywords	Deployment	

Name	Verification Techniques	
Synopsis	This element provides ways to verify and validate different SBA models.	
View	Logical Design Environment	
Authors	-	
Description	This element provides validation techniques to validate different models with respect to functional to functional and non functional properties. The design environment must provide techniques to verify their correctness and completeness. Such verification techniques are available both at the Business Process Management and at the Service Composition and Coordination layers.	
Related elements	***Element***	Business Process Management Layer
	Relation	Used by
	Element	Service Composition and Coordination Layer
	Relation	Used by
References	-	
Glossary	Validation, Verification, Completeness	
Keywords	-Verification and Validation	

2. Research

2.1. Research Challenges

2.1.1. Challenges from JRA-1.1

Name	Definition of a coherent life cycle for adaptable and evolvable SBA
Synopsis	A software life cyle is the total set of software engineering activities necessary to develop and maintain software products. Adaptable Service Based applications need a life cycle taking adaptation into account in a holistic way.
Authors	Elisabetta Di Nitto, Valentina Mazza
Description	The life cycle for the development of adaptable service based applications should include the ability to compose services in complex applications and to adapt and evolve applications. In fact, the service-oriented paradigm enables a high degree of flexibility of SBAs. This means that the SBA can be more easily adapted to new requirements than traditional software systems. The life cycles for SBAs that are currently presented in the literature are mainly focused on the phases that precede the release of software and, even in the cases in which they focus on the operation phases, they do not consider the possibility for SBAs to adapt dynamically to new situations, contexts, requirement needs, service faults, and the like. When dealing with adaptation, on the one side, the requirements engineering phase can be shortened to enhance the time-to-market of the SBA as the missing or misunderstood requirements can later be implemented through adaptation of the running SBA. On the other side, the application has to be designed and developed in such a way that it is able to recognize an adaptation need and to act accordingly. Indeed, not only the application-specific requirements have to be elicited and addressed in the resulting implementation, but also the requirements for adaptation needs to be identified and have to result in a corresponding implementation.
IRF elements	**Life cycle** **Framework:** Engineering and Design
Related challenges	Run-time Quality Assurance Techniques Proactive SLA negotiation and agreement Multi-level and self-adaptation
References	CD-JRA-1.1.2 "Separate design knowledge models for software engineering and service based computing." CD-JRA-1.1.4 "Coordinated design knowledge models for software engineering and service-based computing."
Glossary	Life cycle model, Requirements Engineering, Design for Adaptation, Service Based Application Construction, Service Deployment, Adaptation Strategy, Adaptation Mechanism, Service Composition, Adaptation Requirements and Objectives
Keywords	Life cycle

Name	Measuring, controlling, evaluating and improving the life cycle and the related processes.
Synopsis	Adapting service based application in order to react to changes or to deviations from the desired behavior requires the need to continuously

	monitor the processes and the life cycles. So, there is the need to identify proper approaches for process measurement, control, evaluation and improvement.
Authors	Ita Richardson and Stephen Lane
Description	The definition of approaches for providing the necessary guidelines, procedures and processes to measure, control, evaluate and improve the engineering of SOA is a challenging task primarily due to variations in the existing service-engineering principles, techniques, methodologies and mechanisms both used in the industry and recommended by the research community. In addition, the practices in SOA have been found to be still immature. The problem gets further complex due to the growing requirement of integration of self-* properties such as self-adaptation and self-evolution in the service-based applications as the incorporation of self-* properties need capturing and handling dynamic operational requirements by the system. Therefore, the guidelines, procedures and processes for efficiently and effectively measuring, controlling, evaluating and improving the engineering of SOA, their self-adaptation and self-evolution could be only defined by considering the dynamic operational environment of a service-based application apart from considering the engineering process itself. Although different industry leaders and researchers are conducting the research in the SOA domain yet the definitions of standard guidelines, procedures and processes to measure, control, evaluate and improve the engineering of SOA are still missing.
IRF elements	**Life cycle:** Requirement Engineering and Design, Construction, Deployment and Provisioning, Operation and Management, Identify Adaptation Needs **Framework**: Service adaptation and Design **Infrastructure:** Adaptation Engine, Monitor Engine
Related challenges	-Definition of a coherent life cycle for adaptable and evolvable SBA -Quality Prediction Techniques to Support Proactive Adaptation -Multi-level and self-adaptation
References	CD-JRA-1.1.2 "Separate design knowledge models for software engineering and service based computing."
Glossary	Self-Adaptation, Adaptable Service Based Application, Adaptation, Business Process Measurement, Monitor, Monitoring in Service Composition, Monitoring Requirements, Monitored Property
Keywords	Adaptation, Monitoring

Name	HCI and context aspects in the development of service based applications
Synopsis	The emergence of some requirements for adaptation/evolution implies the triggering of some adaptation and/or evolution actions. service-based application development. The human beings involved in the execution of SBA could raise such requirements. In order to identify the requirements for adaptation/evolution is needed to understand how to characterize the context of the SBA and codify the human-computer interaction knowledge (user task knowledge, user task knowledge, accessibility knowledge).

Authors	Angela Kounkou, Neil Maiden
Description	Humans are involved in service-oriented computing as end users and consumers, but also as service designers and providers (e.g. Human-Provided Services). A foreseen change in the use and distribution of services, as exemplified in the vision of an upcoming Internet of Services, is expected to further draw humans within the "service loop" and to promote human-to-application interaction as well as application to-application interaction. However, to this day, there has been little intersection between research in service-centric systems and Human-Computer Interaction. Human specificities, diversity and tasks characteristics are currently not taken into account in SBA design and delivery - despite being properties that could be powerful drivers for SBAs configuration and personalization. Thus, an integration of HCI knowledge in the engineering of SBAs is necessary to address the need for SBAs to be designed and delivered in ways fitting to human use wherever appropriate. Such integration is also required for the exploration of new opportunities afforded by the exploitation of HCI knowledge - for the enhancement of SBAs' existing capabilities, and for the delivery of new capabilities. It's needed the identification of HCI knowledge that delivers enhanced or new capabilities for SBAs; moreover the codification of this knowledge for its application to the development and use of SBAs it's required. Moreover, another important issue is represented by the characterization of the context of SBA in order to enable the identification of the adaptation requirements; the observation of the context could guide the adaptation process.
IRF elements	**Life cycle:** • Early Requirement Engineering, • Requirement Engineering and Design, • Construction and Quality Assurance, • Deployment and Provisioning, • Identify Adaptation Requirements, • Identify Adaptation Needs **Framework:** Service adaptation and Design
Related challenges	Context and HCI -aware SBA monitoring and adaptation
References	PO-JRA-1.1.3 Codified Human-Computer Interaction (HCI) Knowledge and Context Factors
Glossary	Human Computer Interaction, Context, Adaptable Service Based Application
Keywords	Self-adaptation, self-evolution, HCI, Context

Name	Understand when an adaptation requirement should be selected
Synopsis	In the context of an Adaptable SBA we need to identify the requirements for adaptation and the objectives of the adaptation on the basis of the context and execution information.
Authors	Elisabetta Di Nitto, Valentina Mazza
Description	Observing the context and the properties of the application during execution by means of the monitors, critical events are detected

	triggering the adaptation. The process could be automatic or requiring human involvement: in this case, the user, on the basis of monitored information, decides to trigger the adaptation. When the process is automatic without human involvement, the system is considered self-adaptable. There is the need to identify proper modeling means to enable the automatic identification and analysis of adaptation requirements. These issues require a suitable design for adaptation phase for the identification of the requirements for adaptation, the strategies and the related mechanisms.
IRF elements	**Framework:** Engineering and Design Adaptation and Monitoring **Life Cycle:** Requirements Engineering and Design, Identify Adaptation Needs, Identify Adaptation Strategy and Enact adaptation. **Infrastructure:** Adaptation and Monitoring Engine
Related challenges	Definition of a coherent life cycle for adaptable and evolvable SBA Measuring, controlling, evaluating and improving the life cycle and the related processes.
References	CD-JRA-1.1.2 "Separate design knowledge models for software engineering and service based computing."
Glossary	Adaptation Requirements and Objective, Adaptation Strategy. Adaptation Mechanism, Design for Adaptation
Keywords	Self-adaptation, self-evolution, Adaptation Requirements

Name	Exploiting the concept of service-based applications in the internet of things setting
Synopsis	Accessing to services, anywhere and anytime, and using different devices is on the basis of the pervasive computing. In such scenario, the Internet of Things is related to the possibility to interconnect all the devices through the Internet.
Authors	Elisabetta Di Nitto, Valentina Mazza
Description	By the term Internet of things it is meant the possibility for special purposes devices such as navigation systems, PDAs, cellular phones, sensors, actuators to operate and be visible and accessible through the Internet. Such possibility that is being realized quite fast thanks to the technological advances in the areas of hardware and telecommunication systems is opening new very interesting challenges. While in the past, relatively complex computations running on things were not possible, now these are being experimented in research. This, of course, opens up a huge number of new possibilities in terms of systems that pervasively influence the life of people and help them in several tasks and situations. For instance, through these devices we can imagine users access complex information systems, but also, in the opposite direction, information systems access software services available on these devices to actuate local-scope operations such as the execution of a temperature monitoring function on some critical patient or the invocation of a "turn red for 5 min" service on all the semaphores on some critical paths.
IRF elements	**Life cycle:** Requirement Engineering ad Design, operation and

	management **Framework:** Engineering and Design
Related challenges	Quality Prediction Techniques to Support Proactive Adaptation
References	List of link to paper and web page, in which this challenge was used.
Glossary	Adpatable Service Based Application
Keywords	Service Based Application, Internet of Things, Devices

2.1.2. Challenges from JRA-1.2

Name	Comprehensive and integrated adaptation and monitoring principles, techniques, and methodologies
Synopsis	Current solutions for SBA adaptation and monitoring are highly fragmented and isolated; they address specific domains or aspects, specific functional layers or a particular phase of the SBA life-cycle. A holistic framework is needed that provides a comprehensive and integrated vision of the adaptation and monitoring problem.
Authors	Raman Kazhamiakin, WP-JRA-1.2
Description	To overcome the isolation and fragmentation of existing A&M solutions, the target holistic integrated A&M framework will aim to provide a uniform model of adaptation and monitoring that covers different domains, disciplines, and SBA elements. This framework will accommodate the integration of the existing solutions in different directions: - Cross-layer adaptation and monitoring, where the problem is addressed for SBA as a whole propagating and exploiting specific actions, mechanisms, and tools at different functional SBA layers. - Cross-boundary adaptation and monitoring, where the problem is considered across the boundaries of SBAs, addressing the issue of distribution of information, control, and effects to other applications, external systems, and services. - Cross life-cycle adaptation and monitoring, where the knowledge and models available at different phases of SBA life-cycle (e.g., design-time or post-operational data) is exploited in order to devise new monitoring approaches (e.g., post-mortem analysis for prediction) and adaptation decisions (e.g., to learn from previous decisions and adaptations)
IRF elements	**Framework:** - Adaptation and Monitoring - Integrated A&M capabilities - BPM - SCC - SI **Life Cycle:** - Operation and management - Identify adaptation need - Identify adaptation strategy - Enact adaptation **Infrastructure:** - Monitoring engine - Adaptation engine **Logical design environment:** - A&M modeller

	- A&M configurations - A&M transformation and code generation
Related challenges	Understand when an adaptation requirement should be selected
References	- CD-JRA-1.2.2 Taxonomy of Adaptation Principles and Mechanisms - PO-JRA-1.2.3 Baseline of Adaptation and Monitoring PTMs across Functional SBA Layers CD-JRA-1.2.4 Integrated adaptation and monitoring PTMs across functional SBA layer
Glossary	Adaptable Service-based application, Adaptation, Adaptation Strategy, Adaptation Requirements and Objectives, Adaptation Mechanism, Monitoring, Monitoring Event, Monitoring Mechanism, Business Process, Service Composition, Monitoring in Grid, Self-*
Keywords	- Cross-layer SBA monitoring and adaptation, adaptation and monitoring framework

Name	Proactive Adaptation and Predictive Monitoring
Synopsis	To anticipate the needs for critical changes and to prevent problems in SBA functioning, proactive adaptation aims to exploit predictive monitoring capabilities. In this way, potential problems will be identified before they may happen, and the necessary adaptation actions are driven by the predicted quality deviations or functional problems.
Authors	Raman Kazhamiakin, Barbara Pernici, WP-JRA-1.2
Description	In existing SBA approaches the adaptation aims to react to events that have already happened in the SBA execution or context. However, if the identified event is generated because of a very critical problem the change should be prevented. There is a need for solutions that do not define reactions to the critical changes and problems, but try to avoid them; the shift in SBA adaptation should be directed towards proactive management of undesirable situations. A key element for proactive SBA adaptation is the possibility of predicting future problems or undesirable situations, i.e., to understand what the symptoms representing future problems are, how to represent and detect them. It may be necessary to consider the solutions and mechanisms that traditionally are not applied to the monitoring problem (e.g., run-time testing and validation post-mortem analysis and data mining to predict certain trends. It is also important to identify the minimum set of observables that allow the diagnosis or the prediction of faults in the SBA.
IRF elements	**Framework:** - Adaptation and Monitoring - Quality Definition, Negotiation, and Assurance **Life Cycle:** - Operation and management - Identify adaptation need - Identify adaptation strategy **Infrastructure:** - Monitoring engine - Adaptation engine
Related challenges	- Quality Prediction Techniques to Support Proactive Adaptation - Comprehensive and integrated adaptation and monitoring principles,

	techniques, and methodologies
References	- CD-JRA-1.2.2 Taxonomy of Adaptation Principles and Mechanisms - CD-JRA-1.2.4 Integrated adaptation and monitoring PTMs across functional SBA layer
Glossary	Adaptable Service-based application, Monitoring, Adaptation, Proactive Adaptation, Adaptation Requirements and objectives, Reactive Adaptation
Keywords	Proactive adaptation, predictive monitoring

Name	Context- and HCI-aware SBA monitoring and adaptation
Synopsis	Changes in the context must be reflected in the SBA and managed in appropriate ways; otherwise the system falls out of use. SBAs should be equipped with the required mechanisms to adapt quickly to changes in the system's context, particularly at run-time.
Authors	Andreas Gehlert
Description	The context, e.g., everything, which is outside the boundaries of the software system including stakeholders, other IT systems, rules and regulations as well as business objects, end-user settings and even physical environment, plays an important role for developing and maintaining SBAs. SBAs should be equipped with the mechanisms to model and represent critical context factors, to recognize relevant changes in those factors, and to transform them into the adaptation strategy at run-time. This amounts to modelling and capturing various context aspects, such as business context, user context, human-computer interactions, or execution context; to the development of novel monitoring techniques specifically focusing on the those aspects; and to the definition of new adaptation mechanisms that devise and realize appropriate adaptation strategies for those situations.
IRF elements	**Framework:** - Adaptation and Monitoring - Engineering and Design -**Life Cycle:** - Requirements engineering and design - Deployment and provisioning - Operation & management - Identify adaptation need **Infrastructure:** - Monitoring engine - Adaptation engine **Logical design environment:** - A&M modeller
Related challenges	- HCI and context aspects in the development of service based applications - Understand when an adaptation requirement should be selected
References	- CD-JRA-1.2.2 Taxonomy of Adaptation Principles and Mechanisms - PO-JRA-1.2.3 Baseline of Adaptation and Monitoring PTMs across Functional SBA Layers
Glossary	Adaptation, Monitoring, Adaptation requirements and objectives, Context, Human-Computer Interaction, Personalization, User modelling
Keywords	-

Name	Mixed initiative SBA adaptation
Synopsis	While most of the approaches aim to provide solutions for self-adaptation, in many applications the user has to control the way the system operates and is adapted. The adaptation process should consider and support human roles and activities from the very beginning interacting with them and realizing their decisions.
Authors	Raman Kazhamiakin, JRA-1.2
Description	Most of the existing approaches aim to develop self-adapting systems, i.e., the SBAs that identify and react to changes autonomously. While this approach suites very well in the level of service infrastructure, this is often not the case for the systems oriented towards end users (user-centric systems, B2C applications). The end user has to control the system works (make appropriate decisions or intercept adaptation activities), or drives the way the system is adapted (i.e., system is personalized to fit a particular user through preferences, HCI aspects). To achieve this, it is necessary to consider the human roles in the adaptation process from the very beginning, properly designing the adaptation infrastructure, the models and interfaces to express the adaptation needs, to interact with the user, and to reflect his decisions. The research objectives are to come up with (i) new models that are able to adequately capture the adaptation problem solutions at run-time, are easily understood by humans, and able to capture their intentions and requirements; (ii) novel adaptation infrastructures that specifically target the human actions and decisions and transfer them into the internal system actions; (iii) new interfaces that enable interaction with the adaptation infrastructure based on the corresponding models.
IRF elements	**Framework:** - Adaptation and Monitoring - Engineering and Design -**Life Cycle:** - Requirements engineering and design - Deployment and provisioning - Operation & management - Identify adaptation need - Identify adaptation strategy **Infrastructure:** - Monitoring engine - Adaptation engine **Logical design environment:** - A&M modeller
Related challenges	- HCI and context aspects in the development of service based applications - Multi-level and self-adaptation
References	- CD-JRA-1.2.2 Taxonomy of Adaptation Principles and Mechanisms
Glossary	Adaptation, Monitoring, Adaptation requirements and objectives, User modelling
Keywords	-

2.1.3. Challenges from JRA-1.3

Name	End-to-End Quality Reference Model
Synopsis	To support end-to-end quality provision, the dependencies between different kinds of quality attributes need to be made explicit. In addition, the dependencies between quality attributes at the same and different functional levels of an SBA need to be understood. To achieve a shared understanding of quality attributes between the S-Cube layers and disciplines, a common S-Cube Quality Reference Model will be defined.
Authors	Andreas Metzger, WP-JRA-1.3
Description	**Motivation:** Different kinds of quality attributes are important in an SBA. There is thus a strong need for methods that address quality attributes in a comprehensive and cross-cutting fashion across all layers of a service-based application. Due to the dynamism of the world in which service-based applications operate, techniques are needed to aggregate individual quality levels of the services involved in a service composition in order to determine and thus check the end-to-end quality during run-time. This aggregation will typically span different layers of a service-based application and thus a common understanding of what the different quality attributes mean within and across these layers is needed. **Challenge:** To support end-to-end quality provision, S-Cube will aim at making the dependencies between different kinds of quality attributes explicit. For instance, the interrelation between the fulfilment of different QoS attributes across the various layers will be modelled. In addition, S-Cube aims at understanding the dependencies between QoI attributes on the infrastructure layer, the satisfaction of QoE on the service composition layer and the achievement of QoBiz (business value or business KPIs). One key means to achieve the above objective is to achieve a shared understanding of quality attributes between the S-Cube layers and disciplines by defining the S-Cube Quality Reference Model. Based on the S-Cube Quality Reference Model and the quality definition language (see Challenge "Rich and Extensible Quality Definition Language"), foundations for techniques will devised, which allow aggregating individual quality levels of the services involved in a service composition in order to determine and thus ultimately check end-to-end quality.
IRF elements	**Framework:** - Quality Definition, Negotiation & Assurance - BPM - SCC - SI **Life Cycle:** - requirements engineering **Infrastructure:** - N/A
Related challenges	- Rich and Extensible Quality Definition Language
References	- PO-JRA-1.3.1 Survey of quality related aspects relevant for SBAs - CD-JRA-1.3.2 Quality Reference Model for SBA - CD-JRA-1.3.3 Initial Concepts for Specifying End-to-End Quality Characteristics
Glossary	Quality Attribute, Quality of Service Characteristic, Quality of Service

	Constraint, Quality of Service Dimension, Quality of Service Level, Quality of Service-Based Adaptation, Service Level Agreement, Level of Service
Keywords	-

Name	Rich and Extensible Quality Definition Language
Synopsis	To describe every relevant aspect of quality for services and SBAs, including metrics, units, measurement functions and directives, constraints, value types, etc, a quality definition language is required. This quality definition language will also encompass a rich set of domain-dependent and global quality attributes (i.e., the ones referenced in the S-Cube Quality Reference Model; see Challenge "End-to-End Quality Reference Model") and will be extensible so as to allow the addition of new quality dimensions when needed. Further, this quality definition language will be semantically enriched – where feasible – to be machine-processable or machine-interpretable. Finally, this language must be applicable in complex SBAs, in which services can be invoked and composed with variable quality profiles.
Authors	Andreas Metzger, WP-JRA-1.3
Description	**Motivation:** For what concerns quality modelling and definition, a lack of a well established, rich, extensible, and semantically enriched quality definition language has been observed. As a result, quality capabilities and requirements, as well as service SLAs are described by many different formalisms and languages. **Challenge:** S-Cube strives to develop a quality definition language, which allows describing every relevant aspect of quality for services and SBAs, including metrics, units, measurement functions and directives, constraints, value types, etc. In addition, this quality definition language will encompass a rich set of domain-dependent and global quality attributes and will be extensible so as to allow the addition of new quality dimensions when it is needed (e.g., for a application domain which has currently not been considered). As a starting point, the set of quality attributes as defined in the S-Cube Quality Reference Model (see Challenge "End-to-End Quality Reference Model") will be exploited. Further, this standard quality definition language will be semantically enriched – where feasible – to be machine-processable or machine-interpretable. This quality definition language will be created to be applicable in complex service-based applications, in which services can be invoked and composed with variable quality profiles. Such a quality definition language should thus be capable of expressing quality capabilities and SLAs by using functions, operators and comparison predicates on quality metrics. It should also allow the description of composition rules for possible combinations of composition constructs and quality metrics.
IRF elements	**Framework:** - Quality Definition, Negotiation & Assurance - BPM - SCC - SI

	Life Cycle: - requirements engineering - construction - deployment & provisioning - identify adaptation need **Infrastructure:** - Monitoring engine - Run-time QA engine - Discovery and registry - Negotiation engine - Adaptation engine
Related challenges	- End-to-End Quality Reference Model
References	- PO-JRA-1.3.1 Survey of quality related aspects relevant for SBAs - CD-JRA-1.3.2 Quality Reference Model for SBA - CD-JRA-1.3.3 Initial Concepts for Specifying End-to-End Quality Characteristics
Glossary	Quality Attribute, Quality of Service Characteristic, Quality of Service Constraint, Quality of Service Dimension, Quality of Service Level, Quality of Service-Based Adaptation, Reactive Adaptation, Service Level Agreement, Level of Service
Keywords	-

Name	Exploiting user and task models for automatic quality contract establishment
Synopsis	To devise advanced automated negotiation techniques and protocols (thereby enabling automatic quality contract establishment), one key challenge is how to exploit user and task models, which codify user preferences and characteristics. Those advanced techniques could lead to service negotiators (e.g., autonomous components provided as core services) that perform the negotiation process on behalf of the service consumers (requestors) and providers.
Authors	Andreas Metzger, WP-JRA-1.3
Description	**Motivation:** Service negotiation and agreement involves selecting one out of many service providers based on his quality offer so as to agree on and thus establish the contracts for the delivered service. To address dynamic adaptations of service-based applications, a growing need for automating the negotiation and agreement of quality attributes (e.g., as stipulated by SLAs) can be observed. However, this issue requires considering user interaction and experience (e.g., QoE) issues that may impact on the negotiation itself. This aspect requires a multi-disciplinary effort in which technology researchers will have to interact with researchers addressing user interaction issues. **Challenge:** One key research objective regarding quality contract establishment is to exploit user and task models, which codify user preferences and characteristics (see Challenge "HCI and context aspects in the development of service based applications"), in order to devise advanced automated negotiation techniques and protocols. Those advanced techniques could lead to service negotiators (e.g., autonomous components provided as core services) that perform the negotiation

	process on behalf of the service consumers (requestors) and providers.
IRF elements	**Framework:** - Quality Definition, Negotiation & Assurance - Engineering and Design - SCC **Life Cycle:** - deployment and provisioning - operation & management - enact adaptation **Infrastructure:** - Negotiation engine
Related challenges	- Proactive SLA negotiation and agreement - HCI and context aspects in the development of service based applications
References	- PO-JRA-1.3.1 Survey of quality related aspects relevant for SBAs - CD-JRA-1.3.2 Quality Reference Model for SBA - CD-JRA-1.3.3 Initial Concepts for Specifying End-to-End Quality Characteristics
Glossary	Quality Attribute, Quality of Service Characteristic, Quality of Service Constraint, Quality of Service Dimension, Quality of Service Level, Quality of Service-Based Adaptation, Service Level Agreement, Quality of Service Negotiation, Service Level Agreement Negotiation, Level of Service
Keywords	-

Name	Proactive SLA negotiation and agreement
Synopsis	Based on the envisioned advances in automated negotiation, S-Cube aims to address the current state-of-the-art limitations by starting negotiation when there is evidence that the need for deploying a new service and/or change the conditions of deploying a current service is likely to arise but has not arisen yet. Thus, the challenge is to forecast at run-time a number of factors related to the deployment of services, as the availability of accurate forecasts can lead to effective proactive run-time negotiation strategies for service clients.
Authors	Andreas Metzger, WP-JRA-1.3
Description	**Motivation:** Similar to proactive adaptation (see Challenge "Quality Prediction Techniques to Support Proactive Adaptation"), proactive SLA negotiation and agreement is a key prerequisite for effective run-time SLA negotiation since negotiation does not have a negligible computational cost and, therefore, undertaking it when there is an immediate need to use a new service can be unlikely or unfeasible at run-time. **Challenge:** The challenge for quality contract negotiation and agreement is how to negotiate the terms and conditions under which a service can be offered before the need for deploying or invoking these services arises. Based on the envisioned advances in automated negotiation, we aim to address the limitations introduced above by starting negotiation when there is evidence that the need for deploying a new service and/or change the conditions of deploying a current

	service is likely to arise but has not arisen yet. Thus, our proactive negotiation approach is based on forecasting at run-time a number of factors related to the deployment of services. Those include, for example, the expected demand for a service, the expected levels of service provision, and the expected service terms and conditions that a service negotiator is likely to agree. The availability of accurate forecasts can lead to effective proactive run-time negotiation strategies for service clients. Prediction also plays a role in quality prediction for proactive adaptation (see Challenge "Quality Prediction Techniques to Support Proactive Adaptation"). Although the factors which are relevant differ in both situations, we expect to be able to exploit synergies between the principles and techniques that are developed.
IRF elements	**Framework:** - Quality Definition, Negotiation & Assurance - SCC **Life Cycle:** - deployment and provisioning - operation & management - enact adaptation **Infrastructure:** - Monitoring engine - Discovery and registry - Negotiation engine - Adaptation engine
Related challenges	- Exploiting user and task models for automatic quality contract establishment - Quality Prediction Techniques to Support Proactive Adaptation
References	- PO-JRA-1.3.1 Survey of quality related aspects relevant for SBAs - CD-JRA-1.3.2 Quality Reference Model for SBA - CD-JRA-1.3.3 Initial Concepts for Specifying End-to-End Quality Characteristics
Glossary	Proactive Adaptation, Quality Attribute, Quality of Service Characteristic, Quality of Service Constraint, Quality of Service Dimension, Quality of Service Level, Quality of Service-Based Adaptation, Reactive Adaptation, Service Level Agreement, Software Quality Assurance, Quality of Service Negotiation, Service Level Agreement Negotiation, Level of Service
Keywords	-

Name	Run-time Quality Assurance Techniques
Synopsis	S-Cube will investigate how standard and consolidated offline software quality assurance techniques can be extended to be applicable while the application operates. In addition to extending the quality assurance techniques to the operation phase, synergies between the different classes of analytical quality assurance techniques will be exploited.
Authors	Andreas Metzger, WP-JRA-1.3
Description	**Motivation:** Given the need for adapting service-based applications at run-time, quality assurance techniques that can be applied at run-time are essential. The major type of run-time quality assurance techniques used today is monitoring. Monitoring observes the service-based

	application (or its constituent services) during their current execution, i.e. during their actual use or operation. However, monitoring only allows the assessment of the quality of 'representative' applications (in fact the application in operation) and thus key problems might only be discovered by coincidence. In contrast, standard and consolidated software quality assurance techniques employed during design time, can uncover problems that might only occur after many invocations of the SBA. As an example model analysis can examine classes of executions, thereby leading to more universal statements about the properties of the artefacts. **Challenge:** S-Cube will investigate in how standard and consolidated offline software quality assurance techniques can be extended to be applicable while the application operates. For instance, we will investigate into run-time model analysis techniques and other online techniques such as online testing. In addition to extending the quality assurance techniques to the operation phase, synergies between the different classes of analytical quality assurance techniques will be exploited. As an example, we will investigate how testing can be combined with monitoring in such a way that when a deviation is observed during monitoring, dedicated test cases are executed in order to determine – with high confidence – the cause for the deviation. In order to achieve feasible results from run-time quality assurance, it is essential that the artefacts exploited for run-time analysis or testing are a consistent and up-to-date representation (abstraction) of the running service-based application. For example, this leads to the challenge on how to "synchronize" the model with the SBA in operation in order to achieve valid analysis results. Existing quality assurance techniques appear to be not yet fully incorporated into a comprehensive life-cycle. These aspects are particularly critical as the designers find that understanding what will happen as a result of some self-adaptation design choice quite difficult. Research, jointly with WP-JRA-1.1, will thus address the consistent and comprehensive integration of quality assurance into the service life-cycle (see Challenge "Definition of a coherent life cycle for adaptable and evolvable SBA").
IRF elements	**Framework:** - Quality Definition, Negotiation & Assurance - Engineering and Design **Life Cycle:** - deployment & provisioning - operation & management - identify adaptation need (- identify adaptation strategy) **Infrastructure:** - Monitoring engine - Run-time QA engine - Adaptation engine
Related challenges	- Quality Prediction Techniques to Support Proactive Adaptation - Definition of a coherent life cycle for adaptable and evolvable SBA
References	- PO-JRA-1.3.1 Survey of quality related aspects relevant for SBAs - CD-JRA-1.3.2 Quality Reference Model for SBA - CD-JRA-1.3.3 Initial Concepts for Specifying End-to-End Quality

	Characteristics
Glossary	Analytical Quality Assurance, Failure, Failure Semantics, Fault, Monitoring, Quality Attribute, Quality of Service Characteristic, Quality of Service Constraint, Quality of Service Dimension, Quality of Service Level, Quality of Service-Based Adaptation, Service Fault, Service Level Agreement, Software Quality Assurance, Static Analysis, Testing, User Error, Validation, Verification, Quality of Service Negotiation, Service Level Agreement Negotiation, Level of Service
Keywords	-

Name	Quality Prediction Techniques to Support Proactive Adaptation
Synopsis	To support the vision of proactive adaptation, novel quality prediction techniques need to be devised. Depending on the kind of quality attribute to be predicted, these can range from ones that built on traditional techniques to ones that exploit modern technologies of the Future Internet.
Authors	Andreas Metzger, WP-JRA-1.3
Description	**Motivation:** To respond in a timely fashion to changes implied by the highly dynamic and flexible contexts of future SBAs and to promptly compensate for deviations in functionality or quality, SBAs have to be able to self-adapt. In current implementations of service-based applications, monitoring events trigger the adaptation of an application. Thus self-adaptation often happens after a change or a deviation has occurred. Yet, such reactive adaptations have several drawbacks, such as: (1) Executing faulty services can lead to unsatisfied users and typically requires the execution of additional activities (e.g., compensation or roll-back); (2) Execution of adaptation activities takes time and thereby can reduce the system performance; (3) It can take time before problems in the system lead to monitoring events (e.g., time needed for the propagation of events from the infrastructure to the business process level), thus events might arrive so late that an adaptation of the system is not possible anymore (e.g., because the system is in a deadlock situation). Proactive adaptation presents a solution to address these drawbacks, because – ideally – the system will detect the need for adaptation and will self-adapt before a deviation will occur during the actual operation of the service-based application and before such a deviation can lead to the above problems. Key to proactive adaptation is to predict the future quality (and functionality) of a SBA and to proactively respond if the prediction uncovers deviations from expected quality (or functionality). **Challenge:** To support the vision of proactive adaptation, S-Cube will work on devising novel quality prediction techniques need. Depending on the kind of quality attribute to be predicted, these can range from ones that built on traditional techniques (see Challenge "Run-time Quality Assurance Techniques") to ones that exploit modern technologies of the Future Internet. As an example for the first case, correctness or performance (QoS) could be predicted by building on techniques similar to online testing or run-time model analysis. As an example for the latter case, usability of services (QoE) could be

	predicted by extending existing principles of reputation systems. In this context, one of the possible dimensions to explore is to analyze and predict the properties of networks arising from the interactions between various services. For instance if service A invokes service B, a link between these two services is established. The set of all services and their interactions constitutes a network, which can be represented as a graph structure that can be analyzed by means of traditional link analysis techniques. However, novel and more targeted analysis approaches are needed to support quality prediction.
IRF elements	**Framework:** - Quality Definition, Negotiation & Assurance - BPM - SCC - SI **Life Cycle:** - requirements engineering - construction - deployment & provisioning - operation & management - identify adaptation need (- identify adaptation strategy) **Infrastructure:** - Monitoring engine - Run-time QA engine - Negotiation engine - Adaptation engine
Related challenges	- Run-time Quality Assurance Techniques
References	- PO-JRA-1.3.1 Survey of quality related aspects relevant for SBAs - CD-JRA-1.3.2 Quality Reference Model for SBA - CD-JRA-1.3.3 Initial Concepts for Specifying End-to-End Quality Characteristics
Glossary	Analytical Quality Assurance, Failure, Failure Semantics, Fault, Monitoring, Proactive Adaptation, Quality Attribute, Quality of Service Characteristic, Quality of Service Constraint, Quality of Service Dimension, Quality of Service Level, Quality of Service-Based Adaptation, Reactive Adaptation, Service Fault, Service Level Agreement, Software Quality Assurance, Static Analysis, Testing, User Error, Validation, Verification, Quality of Service Negotiation, Service Level Agreement Negotiation, Level of Service
Keywords	-

2.1.4. Challenges from JRA-2.1

Name	End-to-end processes in Service Networks
Synopsis	How to develop and validate design-time concepts, mechanisms and languages for specifying, analyzing, and simulating end-to-end processes in agile service networks?
Authors	JRA-2.1
Description	**Motivation:** Design time concepts, mechansisms and languages for specifying, analyzing and simulation of end-to-end processes –including

	the protocols that govern them- are still ill understood. **Challenge:** In particular, this challenge involves at least overcoming the following three impediments: • Exploring, developing and validating effective techniques, concepts, languages and mechanisms for analyzing, modelling and simulating end-to-end business processes in ASNs. In particular, deeper understanding of existing service engineering methodologies is needed in collaboration with SED. • Developing and validating approaches exist for analysis and formal verification of business protocols involving bi-lateral and multi-lateral agreements between network nodes. Solutions will be grounded on existing approaches and techniques in protocol engineering in connection with SED, as well as devising Quality of Service for SBAs and Service Level Agreements in SQDN. • Developing and validating analysis and design of business-aware transaction concepts and mechanisms to support business protocols in ASNs are typically very traditional in nature addressing traditional, short-running database transactions ignoring important business semantics including multi-party agreements on QoS. In particular, this sub-challenge is also related to the SQDNA and SED.
IRF elements	**Framework:** - BPM - SCC - SED - SQDN **Life Cycle:** - analysis/design **Infrastructure:** - N/A
Related challenges	- Business transactions in service networks
References	- PO-JRA-2.1.1/2.1.2/2.1.3 -
Glossary	- business process management, optimization, end-to-end processes, protocols, simulation, analysis, choreography, conversations, QoS, composition
Keywords	-

Name	Business Transactions in Service Networks
Synopsis	How to develop and validate concepts, mechanism and languages for run-time monitoring of business transactions?
Authors	JRA-2.1
Description	**Motivation:** Business transactions are the heart-and-soul of agile service networks, and as such need to be better understood. **Challenge:** To overcome this challenges, a better understanding is required of existing adaptation and monitoring approach, techniques and solutions,

	which are scrutinized in the Service Adaptation & Monitoring (SAM) plane, as well as existing (automatic) approach for quality assurance of SEBs (SQDNA). This challenge involves resolving the following two deficiencies of existing techniques and solutions: • Existing transaction monitors typically limit themselves to sniffing and aggregating system-level events. An integrated approach including mechanisms and concepts for monitoring and measuring business events raised by business-aware transactions and related protocols and processes is currently lacking. This sub-challenge will particularly benefit from ongoing research with regarding to system monitors and business activity monitors in the SAM plane. • While existing business transaction monitors may be able to detect and measure system-level errors and anomalies in service-based applications, mechanisms and concepts for adapting business-aware transactions and related protocols and processes in ASNs are not effectively supported. In particular, development of adaptation of business-aware transactions will be grounded on existing adaptation techniques and methodologies that will be assessed in the SAM plane.
IRF elements	**Framework:** - BPM - SCC - SAM - SQDNA **Life Cycle:** - analysis/design/monitoring/adapatation **Infrastructure:** - N/A
Related challenges	- Business protocols in service networks
References	- PO-JRA-2.1.1/2.1.2/2.1.3 -
Glossary	- business process management, end-to-end processes, business transactions, transaction models, long-running transactions, ACID, composition, business activity monitoring
Keywords	-

2.1.5. Challenges from JRA-2.2

Name	Formal Models and Languages for QoS-Aware Service Compositions
Synopsis	This challenge will deal with formal models and Languages for QoS-aware service compositions. The challenge is substantiated by the facts, that firstly, there are no formal models for service compositions available that take into account the QoS and behavioural characteristics of these compositions and secondly, that the formal models are extremely important to guarantee that the final result of a composition services possesses the required characteristics.
Authors	Manuel Carro, Dimitris Plexousakis, Dimka Karastoyanova, WP-JRA-2.2
Description	**Motivation:** When composing several services into an aggregated one, it is usually necessary to fulfil several characteristics in the composed service: the composite service needs to deliver the information requested, behave

	as desired, and meet the quality standards required from it. In general, extremely difficult to ensure that a complex, final product will deliver what is required from it without resorting to a model of the system, its environment, and the requirements. The degree to which this model really reflects the real product / environment and to which reasoning within the model is feasible and accurate with respect to the modeled entities greatly impacts the applicability of such a model. Formal models have the advantage of being equipped with a non-ambiguous meaning and a way to reason on instances of the model in such a way that sound results are achieved—i.e., inferred properties are not in contradiction with the semantics of the model. Given the complexity of Service Oriented Computing and service composition, it is difficult to find a single existing proposal which can seamlessly and in a uniform way tackle all the issues. **Challenge:** The primary research objective will be to devise novel models for QoS-aware services and service compositions, based on the expertise on formal models of the partners. Models of QoS-aware service compositions need to provide means for reasoning about services and their compositionality based both on their functionality in a wide sense (i.e., semantics / behaviour) and on their QoS attributes. Such models need to be sufficiently expressive to describe a wide class of service compositions and QoS attributes, while at the same time constrained enough to ensure that the standard reasoning tasks performed on the model are decidable (at least in the common cases) and reasonably efficient. Determining (QoS-aware) compositionality assumes that service behavior is exposed in a declarative manner with the use of formal specification languages. As far as reasoning on service functionality is concerned, rich semantic formal models will need to be devised. These models should aim at describing the behavior of services and service compositions and offer a complete description of what the services provide under all circumstances. Among the formal basis to use in order to construct more general formal models and languages to describe and reason about service compositions, we plan to explore the use of temporal logic to specify message exchange patterns between software services and QoS constraints with respect to time. On the semantic side (utterly necessary in order to be able to perform automatic, dynamic service compositions), we foresee that description logics can be used to model service structures and, with suitable extensions still to be fully developed, QoS constraints. Modeling of service metadata is also an important aspect. QoS attributes of services will have to be included in the description of the services and of their compositions. The application of soft constraints for modelling and reasoning about QoS will also be examined. Formal models will form the formal substrate of execution languages which can be used as input for execution, monitoring, and later analysis.
IRF elements	**Framework:** - SCC

	- SI **Life Cycle:** - construction - deployment & provisioning **Infrastructure:** - Modelling Tool for Service Composition
Related challenges	-
References	- PO-JRA-2.2.1 Overview of the state of the art in compositions and coordination of services - CD-JRA-2.2.2 Models and Mechanisms for Coordinated Service Compositions - First Draft - CD-JRA-2.2.4 Models, Mechanisms and Protocols for Coordinated Service Compositions - CD-JRA-2.2.6 QoS-aware, Coordinated Service Compositions – Mechanisms and Techniques
Glossary	Service Composition, Process Model, Service Model, Formal Specification
Keywords	-

Name	Monitoring of Quality Characteristics of Service Orchestrations and Service Choreographies
Synopsis	In the context of QoS-aware service compositions, our focus lies on monitoring of quality characteristics of service orchestrations and service choreographies. As service compositions implement business processes and at the same time run on IT infrastructure, their quality characteristics are influenced by both process-level and infrastructure-level metrics. A holistic monitoring approach for quality characteristics of service compositions involves monitoring of service orchestrations in terms of both process-level and infrastructure level factors and in addition monitoring of quality characteristics across participants in service choreographies.
Authors	Branimir Wetzstein, Martin Treiber, Manuel Carro, Dimka Karastoyanova, WP-JRA-2.2
Description	**Motivation:** Monitoring is the process of collecting relevant information from the execution data of service composition and involved services in order to evaluate properties of interest and report results of that evaluation. Monitored properties can be based on functional aspects (e.g., correctness properties) or non-functional aspects (e.g., QoS properties). In the context of QoS-aware service compositions, our focus lies on monitoring of quality characteristics. Current solutions to service composition monitoring mostly focus and are constrained to one layer or very specific aspects, e.g., process metrics as part of business activity monitoring, or QoS metrics as part of SLA monitoring and do not integrate information from all layers and deal with their dependencies. As service compositions implement business processes from the BPM layer, and at the same time are based on technical QoS properties of Web services and IT infrastructure used, monitoring of service compositions should take into account and integrate both business related metrics and technical QoS metrics.

	Challenge: S-Cube will devise mechanisms and corresponding development methods which aim to support a holistic monitoring approach for service compositions which integrates monitoring information from different layers and across choreography participants in the SCC layer. In particular, mechanisms will be devised which support: – Integrated Monitoring of process and QoS characteristics of service compositions: We want to be able to monitor metrics which define time, cost, and quality related properties of business processes (a.k.a. process performance metrics) and correlate them with technical QoS metrics of the underlying IT infrastructure. – Monitoring of quality characteristics in service choreographies: Mechanisms will be devised which enable monitoring of processes in service choreographies in cross-organizational scenarios. This type of monitoring has to take into account that on choreography level only public processes of the participating service orchestrations are available.
IRF elements	**Framework:** - SCC - Adaptation and Monitoring **Life Cycle:** - Operation & Management **Infrastructure:** - Monitoring Engine
Related challenges	- Analysis and Prediction of Quality Characteristics of Service Compositions - QoS Aware Adaptation of Service Compositions - Comprehensive and Integrated Adaptation / Monitoring Principles, Techniques and Methodologies
References	- PO-JRA-2.2.1 Overview of the state of the art in compositions and coordination of services - CD-JRA-2.2.2 Models and Mechanisms for Coordinated Service Compositions - First Draft - CD-JRA-2.2.4 Models, Mechanisms and Protocols for Coordinated Service Compositions - CD-JRA-2.2.5 Derivation of QoS and SLA specifications - CD-JRA-2.2.6 QoS-aware, Coordinated Service Compositions – Mechanisms and Techniques
Glossary	Service Composition, Service Orchestration, Service Choreography, Quality Attribute, Quality of Service Characteristic, Monitoring, Business Activity Monitoring
Keywords	-

Name	Analysis and Prediction of Quality Characteristics of Service Compositions
Synopsis	When monitoring of quality characteristics of service compositions reveals that KPIs do not meet their target values, users are interested in finding out the causes and the most influential factors in order to be

	able to adapt the composition to prevent those violations in future. Analysis and prediction mechanisms for quality characteristics will be devised, which are integrated with the monitoring mechanisms and provide input to the adaptation framework on which quality characteristics to adapt.
Authors	Branimir Wetzstein, Martin Treiber, Manuel Carro, Dimka Karastoyanova, WP-JRA-2.2
Description	**Motivation:** While *monitoring* focuses on reporting of values of monitored properties (what?) in a timely fashion, *analysis* is based on monitoring results and tries to find explanations for monitored values (why?) or predict future values. In this respect, analysis of service compositions may also be performed ahead of time (i.e., before the actual execution takes place) in order to infer emerging properties (or, quite often, approximations thereof) which are guaranteed to be universally valid — i.e., true any particular execution when the initial assumptions for the execution hold. Based on the results of monitoring and analysis the service composition can be optimized (QoS-aware Adaptation of Service Compositions). **Challenge:** Based on Monitoring of Quality Characteristics of Service Orchestrations and Service Choreographies, mechanisms will be devised which provide explanations and prediction of monitored values. When KPIs do not meet their target values, business users are interested in finding out the causes and the most influential factors. In our case, we want to be able to derive the most influential factors and dependencies of KPIs on process performance metrics and QoS characteristics of used services. In this context, prediction of KPI and QoS values will be supported, which should enable pro-active service adaptation. In that context, one possible approach is to use data mining techniques (to perform online and post-mortem analysis) and also design time/static analysis which can be used to warn of possible (and sometimes certain) problems before they appear.
IRF elements	**Framework:** - SCC - Adaptation and Monitoring **Life Cycle:** - Operation & Management **Infrastructure:** - Monitoring Engine
Related challenges	- Monitoring of Quality Characteristics of Service Orchestrations and Service Choreographies - QoS Aware Adaptation of Service Compositions - Proactive SBA Adaptation and Predictive Monitoring
References	- PO-JRA-2.2.1 Overview of the state of the art in compositions and coordination of services - CD-JRA-2.2.2 Models and Mechanisms for Coordinated Service Compositions - First Draft - CD-JRA-2.2.4 Models, Mechanisms and Protocols for Coordinated Service Compositions - CD-JRA-2.2.5 Derivation of QoS and SLA specifications - CD-JRA-2.2.6 QoS-aware, Coordinated Service Compositions – Mechanisms and Techniques

Glossary	Service Composition, Service Orchestration, Service Choreography, Quality Attribute, Quality of Service Characteristic, Monitoring, Quality of Service-Based Adaptation
Keywords	-

Name	QoS Aware Adaptation of Service Compositions
Synopsis	Adaptation of Service Compositions driven by changes in the environment and in particular by the changes in QoS characteristics still remains a major challenge in service-based applications. Mechanisms for enabling such adaptation will be developed as well as the major drivers for adaptation will be defined. The influence of the BPM and SI layers of SBAs on the adaptation of SC must be taken into account to ensure consistency of the adaptation steps.
Authors	Dimka Karastoyanova, Olha Danylevych, Salima Benbernou, WP-JRA-2.2
Description	**Motivation:** In general, QoS-aware adaptation refers to the approaches and mechanisms for adaptation that enable reaction to changes in QoS requirements on the service composition. This means that adaptation of Service Compositions (SCs) must be considered in relation to the measurement, aggregation and disaggregation of QoS parameters of the compositions (usually called Process Performance Metrics (PPMs)) and of the services they employ (QoS characteristics of the services). There is a gap in the current SOC related research with respect to classifications of adaptation types and adaptation drivers and identifying those types and drivers with particular importance for QoS-aware adaptation. Furthermore, mechanisms for reacting to such drivers must be developed, which is the major concern in WP-JRA-2.2 where such mechanisms will be devised and realized. The special focus is on service compositions. The classifications and mechanisms must be refined to include the requirements for integrated cross-layer adaptation of SBAs. Several areas with inadequate or missing solutions can be identified so far: cross-layer adaptation of SBAs and its influence on SCs driven mainly by changes in QoS characteristics; proactive adaptation based on monitoring and analysis results; Process fragmentation of service composition to improve reusability and flexibility of SBAs, including coordination protocols between process partitions: Leveraging the emerging Web 2.0 techniques related to service composition and adaptation will also be taken into account. **Challenge:** Our main objective is to devise adaptation mechanisms for service compositions to react to and predict different triggers, including those from the BPM and Service Infrastructure levels thus accounting for the interplay among the layers of SBAs. The focus will be mainly on mechanisms that consider QoS-awareness as a major criterion to trigger adaptation. Mechanisms for reactive adaptation will be provided to enable different adaptation types and will take into account QoS characteristics of services, QoS requirements of the SCs and those imposed by the BPM layer in terms of KPIs. Additionally, the mechanisms will consider the

	SLAs between the SC and the participating services. Pro-Active adaptation based on monitoring and analysis results (in particular based on prediction) is necessary in some cases in order to adapt instances of a service composition based on information provided by the execution of other instances of the same composition. The information used to enable this and trigger that kind of adaptation is the same as the one used during composition monitoring. Monitoring information about services and the business processes may also be used. Note that in this case our focus is on QoS characteristics measurements as well. Proactive changes are enabled using the same adaptation types as in the cases of reacting to changes due to unexpected situation. The difference to existing approaches is that there must be additional means to analyse process instances constantly to recognize possible critical situations in future. Process fragmentation of service composition will be utilized to improve reusability and flexibility of SBAs. The corresponding coordination protocols (if applicable) will also be the subject of our work. We shall also investigate the possibility to introduce adaptation features of Web 2.0 service composition models.
IRF elements	**Framework:** - Quality Definition, Negotiation and Assurance - Adaptation and Monitoring - SCC **Life Cycle:** - Operation and Management, Identify Adaptation Need, Enact adaptation **Infrastructure:** - Engine, Bus, modelling tools for SCs
Related challenges	- Comprehensive and Integrated Adaptation / Monitoring Principles, Techniques and Methodologies - Proactive SBA Adaptation and Predictive Monitoring - - Multilevel and Self-adaptation
References	- PO-JRA-2.2.1 Overview of the state of the art in compositions and coordination of services - CD-JRA-2.2.2 Models and Mechanisms for Coordinated Service Compositions - First Draft - CD-JRA-2.2.3 Algorithms and Techniques for splitting and merging service compositions
Glossary	Service Composition, Adaptation, QoS-based adaptation, evolution, optimization, Design for Adaptation, Proactive Adaptation, Rebinding, Service Orchestration, Workflow
Keywords	-

2.1.6. Challenges from JRA-2.3

Name	Multi-level and self-adaptation
Synopsis	Provide support for dynamic adaptation of service-based applications
Authors	Françoise André, Jean-Louis Pazat
Description	Service-based applications must be dynamically adaptable in order to accommodate the continuous evolution of their environment. Existing approaches to the adaptation problem do not fully meet the requirements of highly dynamic, large-scale service ecosystems. Our objective is to support building adaptable service-based applications; not

	only individual adaptable services (addressed mainly in the challenge "Self-* in service execution, discovery and registries") but also adaptable compositions of services. The adaptations can be performed either because monitoring has revealed a problem or because the application identifies possible optimizations or because the execution context has changed. The context here includes the set of services available to compose the service-based application, the parameters and protocols being in place, the user preferences, and other environment characteristics (location, time, other running applications). Three levels of adaptation should be considered. The lowest level concerns adaptation of one service on its own. The second level concerns adaptation between services within a service composition in order to satisfy the needs of an application. Finally, the highest level concerns the adaptation of several applications running in parallel, each application being itself a composition of services.
IRF elements	*Conceptual research framework*: Service infrastructure, Service composition and coordination, Adaptation and monitoring *Reference life-cycle*: Identify adaptation need, Identify adaptation strategy, Enact adaptation, Operation & management *Logical run-time architecture*: Adaptation engine *Logical design environment*: AM modeling
Related challenges	Self-* in service execution, discovery and registries, Deployment and execution management, Design for adaptation, Proactive adaptation
References	
Glossary	adaptable SBA, adaptation, monitoring, self-*
Keywords	multi-level, self-adaptation

Name	Deployment and execution management
Synopsis	Provide support for on-demand, dynamic provisioning of services
Authors	Zsolt Nemeth
Description	Deploying and decommissioning services in an on-demand, dynamic way is useful for establishing adaptability, self-healing, and other self-* properties. On-demand, dynamic service provisioning is a subset of general adaptation techniques and thus presents many similar research problems. This type of adaptation should be supported by past experience (learning), be able to take into consideration a complex set of conditions and their correlations, act proactively to avoid problems before they can occur and have a long lasting, stabilizing effect. The decision-making mechanism of such on-demand service provisioning should be investigated, which involves problem identification, analysis of symptoms, policies for various deployment scenarios, and a knowledge base for provisioning strategies. The realization of on-demand service provisioning includes discovery and analysis of discovery, which should also be investigated. Other specific research issues include on-demand service image creation, distribution and replication for recovery or preemption purposes, and offering various deployment features.
IRF elements	*Conceptual research framework*: Service infrastructure, Adaptation and monitoring *Reference life-cycle*: Deployment and provisioning, Operation & management

	Logical run-time architecture: Service container, Discovery and registry infrastructure, Adaptation engine *Logical design environment*: Service deployment
Related challenges	Multi-level and self-adaptation, Self-* in service execution, discovery and registries
References	
Glossary	on-demand service deployment, automatic service deployment, service deployment
Keywords	deployment, dynamic provisioning

Name	Process mining for service discovery
Synopsis	Enable the discovery of human-provided activities in addition to traditional services and business process
Authors	Fabrizio Silvestri
Description	A modern discovery facility should support the discovery of human-based processes in addition to traditional services and business processes. In other words, we want to leverage the knowledge coming from how services (including human-provided services) are invoked and composed. There is a whole body of work in the literature showing how human activities can be traced down and analyzed in a very effective way. In our case, data may come from different sources. The most obvious one is data coming from the monitoring activity, which contains traces from the activities of processes, tasks, etc. The log of those activities can be used, for instance, to derive a new business model, or to detect failures and unexpected behaviour. In particular, we intend to study a new problem, which is related to process mining. We called it mashup discovery and it consists of discovering implicit human user activities in logs of events. One particular case study will be the case of Web search engines' query logs but the techniques developed will be also applicable to other fields, such as touristic activities.
IRF elements	*Conceptual research framework*: Service infrastructure *Reference life-cycle*: Deployment and provisioning, Operation & management *Logical run-time architecture*: Discovery and registry infrastructure *Logical design environment*: service modellers
Related challenges	Feedback-based service discovery
References	
Glossary	service discovery, process mining
Keywords	process mining, mashup discovery