EUROPEAN COMMISSION
DIRECTORATE GENERAL
COMMUNICATIONS NETWORKS,
CONTENT AND TECHNOLOGY
UNIT E2 - SOFTWARE AND
SERVICES, CLOUD

Cloud Computing Service Level Agreements

Exploitation of Research Results

Editor: Dimosthenis Kyriazis

Brussels, June 2013

Abstract

The rapid evolution of the cloud market is leading to the emergence of new services, new ways for service provisioning and new interaction and collaboration models both amongst cloud providers and service ecosystems exploiting cloud resources. Service Level Agreements (SLAs) govern the aforementioned relationships by defining the terms of engagement for the participating entities. Besides setting the expectations by dictating the quality and the type of service, SLAs are also increasingly considered by the providers as the key differentiator to achieve competitive advantage. In this context, the current report surveys the research outcomes stemming from European and National projects, and discusses how these outcomes address the complete SLA lifecycle. In addition, this report introduces a set of recommendations to support the on-going policy work on SLAs of the Cloud Select Industry Group (SIG), while identifying the research outcomes that can be exploited for the implementation of the recommendations. What is more, the report examines the potential impact of the realization of the listed recommendations in different domains and areas.

Executive Summary

The dynamic and technology-rich digital environment and the market economic constraints has shifted service provisioning from a pre- and strictly- defined to an *on-demand* orientation. The cloud services industry is addressing this challenge through the commoditization of IT assets and provision of services following on-demand usage patterns. This relatively broad *cloud ecosystem* comprises of various interacting entities (i.e. providers, brokers, customers and end-users) with *different expectations and objectives*. Service Level Agreements (SLAs) provide the fundamental ground for the aforementioned interactions by setting: (i) goals through *Quality of Service* (QoS) attributes, (ii) privacy and protection constraints through *Quality of Protection* (QoP) attributes, (iii) expectations through the *description of actions* that need to be taken in order to deliver the service according to the QoS attributes, (iv) *responsibilities* through the inclusion of obligations of parties including penalties and exclusion terms, (v) evolvement cases through the *definition of rules* that enable efficient adaptation of resource provisioning based on the dynamic demands of the applications and the end users.

In recent years, extensive research has been conducted in the area of SLAs in cloud computing environments. Representative outcomes of this research are presented in the current document. Even though the outcomes of ***European and National research projects*** on cloud technologies are emphasized, given that SLAs is a core concept in the IT domain, the report also presents outcomes stemming from projects focusing on networking technologies and infrastructures. In the area of ***SLA specifications and term languages***, various innovative approaches have been developed such as the manifest in OPTIMIS, the blueprint in RESERVOIR and 4CaaSt, the quality model in CONTRAIL, the QoS-oriented specification in Q-ImPrESS, the virtualised service network in IRMOS, and the service description in SLA@SOI. ***Business aspects*** in the SLA lifecycle have also been considered - a representative example would be the business-enhanced template in ETICS, as well as frameworks supporting ***composite services*** as the cloud federations proposed in CONTRAIL, the eMarketplace in 4CaaSt or the mechanisms in ETICS and GEYSERS. As the basis for the provision of QoS guarantees, interesting works regarding ***performance estimation*** and ***workload prediction*** have been developed in Cloud-TM, service network ***risk, uncertainty and dependability*** for critical infrastructures in SERSCIS, ***data reliability and safety*** in PrestoPRIME, while enhancements for ***trade-off analysis*** have been proposed in Q-ImPrESS. The unified monitoring interface from Cloud4SOA, the adaptable monitoring tools from IRMOS, the SLA-driven monitoring from SLA@SOI, the scalable and efficient monitoring from Stream, and the network monitoring from mPlane cover the ***monitoring*** aspects for SLAs. Novel ***negotiation*** approaches enabling dynamicity, automation, scalability and re-negotiation during runtime have been implemented by Cloud4SOA, OPTIMIS, SLA@SOI and IRMOS respectively. Regarding ***SLA enforcement***, CloudScale tools for automatic root cause analysis, 4CaaSt developments for elasticity management, VISION cloud approaches for proactive SLA violation detection, as well as CumuloNimbo and Cloud-TM outcomes with respect to enforcement for transactional systems are worth mentioning.

These research outcomes have demonstrated important innovations in the respective fields and their exploitation is expected to offer clear potential to cloud stakeholders. Furthermore, in today’s cloud service industry, the lack of standardization in SLAs and the use of SLAs as a potential marketing vehicle have resulted in an SLA jargon. On the other hand, the users are becoming more demanding in terms of service requirements, offered and guaranteed levels of

quality, data protection, etc. Taking these facts into consideration, the report includes a set of *recommendations* (to support the on-going policy work on SLAs of the Cloud Select Industry Group - SIG) and proposes the exploitation of specific research outcomes in order to form the basis for the realization of the recommendations.

The first recommendation focuses on the cornerstone, the ***SLA specification***. Term languages should be sufficiently expressive to allow concise and clear description of terms (including penalties), service quality attributes addition, metrics and KPIs definition. Moreover, it is recommended to capture the SLA through a structured representation (e.g. in XML format) in order to make it machine-readable and use it during the complete SLA lifecycle (from selection of providers to automated and dynamic negotiation, enforcement and conclusion). On the same topic, it is recommended to ***differentiate the contents and scope of SLAs and contracts***, and introduce ***legal attributes in SLAs*** in order to clarify the responsibilities and obligations of all involved entities. Legal attributes will cover aspects related to data (such as processing or placement), QoP terms to reflect the responsibilities, and exclusion terms. ***Outcome-based***, ***user-oriented (or experience-oriented) SLAs*** (that will embrace SLA specifications) are also proposed, aiming to increase the cloud market pool for non-technical users through simplicity and relieving the users from the need to be aware of all service and infrastructure parameters.

An additional recommendation is proposed to address the users' requirements for ***composite services*** that consist of services offered by different providers - current market fragmentation and cloud service models contribute to the increasing rate of such requirements since many organizations provide services that depend on services from other organizations. To address such a multi-provider environment, SLA specifications should capture in a parametric way the dependencies and interactions between the services, while handling of the dependencies should also be feasible through SLA management framework.

Furthermore, one of the main users concern refers to the validation and supervision of the quality of the provided services: users require greater levels of transparency through ***accurate*** and ***on-time*** delivery of ***SLA monitoring*** information. Nevertheless, monitoring is also fundamental for providers since SLAs are expected to be used by cloud vendors as their certification in order to establish themselves when entering the competitive cloud market. To this end, accurate monitoring is a key to demonstrate their commitment to the agreed quality levels. We recommend delivering monitoring information on the level of service attributes included in the SLAs, thus providing both application- and infrastructure- related monitoring data. Frameworks collecting and managing the data should meet specific latency requirements, while minimizing the footprint on the system. Finally, it is recommended that cloud vendors develop APIs to provide unified monitoring data (of major importance in the cases of composite services) or enable Trusted Third Parties (TTP) to undertake the monitoring responsibility.

What is more, Future Internet applications and mission-critical applications increasingly rely to cloud environments, raising the need for infrastructures that can facilitate real-time, interactivity and allow ubiquitous service provisioning. To tackle this challenge, one of the recommendations focuses on the ***certification of provider's liability*** in order to identify their "guaranteed" offerings and the evolvement of SLAs at runtime, i.e. automatic ***SLA re-negotiation***; while another one highlights the need for ***SLA enforcement*** through proactive SLA violation detection mechanisms and models for automatic root cause analysis.

Finally, the adoption of current SLA standards (i.e. WS-Agreement by OGF and WSLA by IBM) highlights the success potential and need for standards. It is therefore recommended to develop ***domain agnostic standards*** and to encourage SLA-relevant standards (e.g. Open Cloud Computing Interface - OCCI, also developed by OGF) to incorporate enhancements which further enable SLA support. The domain agnostic standards should target different elements and parts of the SLA lifecycle: the SLA specification (covering also the case of composite services), the monitoring tools and the management frameworks.

The report concludes with a discussion of the potential *envisioned impact of the realization of the recommendations* in different domains and areas, ranging from increased competitiveness enabled through the consideration of SLAs as a means to certify providers (similar to the concept of the Cloud Auditor - proposed by NIST [1]), to wider adoption of cloud solutions by end users, increased market pool of cloud computing to non-technical users, enhanced cost and performance trade-offs, optimized service deployment and operation through the use of third party specialized services, and broader service offerings through the ability to provide composite services and guarantee QoS for future internet and mission critical applications.

Acknowledgements

Several collaborators and researchers, participating in European research projects, have contributed greatly to Cloud Computing SLAs, developing remarkable outcomes in the respective projects. We (contributors and editor of the report) wish to record our sincere acknowledgments to their research work.

We also owe a debt of gratitude to the European Commission, DG ConNECT, Unit “Software & Services, Cloud” for initiating and supporting this effort, and especially to Ms. Maria Tsakali, Scientific Officer and Dr. Ken Ducatel, Head of Unit.

List of Contributors

Lorenzo Blasi (HP, Italy); Gunnar Brataas (SINTEF, Norway); Michael Boniface (IT Innovation Centre); Joe Butler (Intel, Ireland); Francesco D'andria (ATOS, Spain); Michel Drescher (European Grid Infrastructure, Netherlands); Ricardo Jimenez (Universidad Politécnica de Madrid, Spain); Klaus Krogmann (FZI, Germany); George Kousiouris (National Technical University of Athens, Greece); Bastian Koller (High Performance Computing Center Stuttgart, Germany); Giada Landi (Nextworks, Italy); Francesco Matera (Fondazione Ugo Bordoni, Italy); Andreas Menychtas (National Technical University of Athens, Greece); Karsten Oberle (Alcatel-Lucent Bell Labs, Germany); Stephen Phillips (IT Innovation Centre); Luca Rea (Fondazione Ugo Bordoni, Italy); Paolo Romano (INESC-ID, Portugal); Michael Symonds (ATOS, Netherlands); Wolfgang Ziegler (Fraunhofer Institute SCAI, Germany)

Table of Contents

Index of Figures

1 *Introduction*

Cloud computing is essentially changing the way services are built, provided and consumed. As a paradigm building on a set of combined technologies, it enables service provision through the commoditization of IT assets and on-demand usage patterns. Nowadays, cloud computing refers to a computing paradigm whose foundation is the delivery of services and ICT assets [2], often denoted as XaaS (Everything as a Service). The term refers to an increased number of cloud-based resources and services provided over the Internet, with the most common examples, following the SPI model [3], Software (SaaS), Platform (PaaS) and Infrastructure (IaaS) as a service.

As the aforementioned cloud service model matures and becomes ubiquitous, it raises the possibility of improving the way services are provisioned and managed, thus allowing providers to address the (diverse) needs of consumers. In this context, Service Level Agreements (SLAs) emerge as a key aspect, since they serve as the foundation for the expected quality level of the service between the consumer and the provider. Nevertheless, the *diversity of the proposed SLAs* by providers (with marginal overlaps), has led to multiple different definitions of cloud SLAs. Furthermore, misconceptions exist on what is (if there is) the difference between SLAs and contract, what is the borderline, what are the terms included in each one of these documents and if and how are these linked. We provide the following definitions according to ITIL [4]:

A **Service Level Agreement** (SLA) is a formal, negotiated document that defines (or attempts to define) in quantitative (and perhaps qualitative) terms the service being offered to a Customer. Any metrics included in a SLA should be capable of being measured on a regular basis and the SLA should record by whom.

A **Contract** is a legally binding agreement between two or more parties. Contracts are subject to specific legal interpretations.

An alternative definition going a bit away from the pure process oriented ITIL one has been provided by the TM Forum [5]: “A Service Level Agreement (SLA) is a formal negotiated agreement between two parties. It is a contract that exists between the Service Provider (SP) and the Customer. It is designed to create a common understanding about Quality of Service

(QoS), priorities, responsibilities, etc. SLAs can cover many aspects of the relationship between the Customer and the SP, such as performance of services, customer care, billing, service provisioning, etc. However, although a SLA can cover such aspects, agreement on the level of service is the primary purpose of a SLA".

Based on the definitions, this report focuses on SLAs as negotiated "agreements" between different parties / entities. As "agreements", SLAs encapsulate a set of different aspects regarding the services provisioning. These refer to the agreed *Quality of Service* (QoS) – captured through different terms, the *Service Level Objectives* (SLOs), the *responsibilities* and *obligations* of the parties, as well as the *penalties* in cases of non-compliance to the agreed terms. SLAs may be re-negotiated in case service requirements change or if there is an inability to deliver the service based on the initially agreed requirements. Given that neither a core SLA specification nor a core contract template exists for cloud-based services, additional details regarding the contents of these documents are not provided in this report. However, the importance of capturing the corresponding terms and providing a clear differentiation between SLAs and contracts, led us to include it amongst the recommendations (further described in Section 4 of this report).

1.1 Motivation

Service Level Agreements play a central role in the service lifecycle, since by capturing service expectations and entities responsibilities they *drive both engineering decisions* at conception level (during for example service design) and *operational decisions* (during for example service usage and delivery). SLAs enable participating entities to agree on what services will be offered, how will the services be delivered and who will be responsible for execution, completion, potential failures and privacy aspects.

Nevertheless, SLAs are agreements limited to description of expectations and responsibilities. As emphasized in [6]: "An SLA cannot guarantee that you will get the service it describes, any more than a warranty can guarantee that your car will never break down. In particular, an SLA cannot make a good service out of a bad one. At the same time, an SLA can mitigate the risk of choosing a bad service". The latter highlights the need for supporting *tools and mechanisms used during different phases of the SLA lifecycle*, such as monitoring of service execution adherence to the agreed terms and enforcement through triggering of actions to support emerging requirements. The main goal of such frameworks is to ensure that the service is delivered according to specific quality levels (as set by the corresponding QoS attributes). The specific need has been raised by various stakeholders in the cloud ecosystem: Google [7] places *SLAs and mechanisms to enforce them amongst the main challenges*, while another cloud provider, CloudOne, emphasizes that [8]: "Much good work has been completed on *SLAs and the entire business model around the cloud*, but much remains"; Forrester research analysts mention that *SLAs are crucial when sending critical data offsite* [9], while Accenture research analysts also set *management and supervision of SLAs* amongst the main challenges in cloud computing [10]; the requirement for *expression of granular needs in SLAs* has been highlighted by a standards expert at VMWare [11]; with one of the main stakeholders, the users (through an advocacy group) [12] raising the fact that "SLAs are weighted heavily in the provider's favor, leading to the *vendor's liability being limited*. The burden is usually more likely on the consumer to recognize breaches of the SLA, notify their service provider and request a credit".

In this context it is important to provide an overview of SLA-related research outcomes that address the requirements stated by providers, research analysts, standards experts and users. These research outcomes may also be exploited as baseline technologies for the realization of a set of recommendations included in the current report.

1.2 Scope and Purpose

The purpose of this document is to serve as a starting point for the exploitation of research results stemming from European and National projects. To this end, the report identifies and delivers short descriptions of the main SLA-related contribution of each project. What is more, a set of recommendations is provided to address the requirements of different entities in the cloud ecosystem. The recommendations aim at facilitating wider adoption of cloud solutions and enable providers to offer a wider set of services through approaches that enable the provision of QoS guarantees (as required for example in future internet and mission critical applications) and facilitate efficient collaborations amongst providers. The content regarding the research outcomes has been compiled following a working group meeting (Workshop on “Cloud Computing SLAs in FP7 - Exploitation of Research Results”) that was organized and hosted by the EC in Brussels, 27 May 2013.

This report targets not only the research and academic community, but also the European ICT industry and decision makers (including the Cloud Select Industry Group on SLAs, the ETSI Cloud Standards Coordination and the European Cloud Partnership).

The rest of the document is structured as follows: Section 2 introduces the main actors and phases in the SLA lifecycle in order to set the scene and enable mapping of the research outcomes to the overall picture. Section 3 contains the main research contributions of each project, while Section 4 provides the recommendations. Finally, Section 5 concludes the report.

2 *Service Level Agreements Landscape*

This chapter provides an overview of the SLA landscape, introducing various stakeholders and actors engaged in an SLA lifecycle, as well as an SLA lifecycle metamodel. The metamodel doesn't reflect a specific architectural approach and is by no means exhaustive in terms of processes and components. The aim of the metamodel is to depict the main concepts, structures and processes of the SLA lifecycle in order to enable the mapping of EU projects outcomes to the overall picture.

2.1 Stakeholders and Actors

Before introducing the various stakeholders and actors, a brief use case is described in order to clarify the different roles in the SLA lifecycle. A museum would like to offer to the visitors a service for delivering information regarding the exhibits while being in the museum. A software house enterprise has developed such an application, which is being offered as a cloud service. To develop and deploy the application, a platform provider has provided to the software house a framework for developing the application, as well as a framework for obtaining the licenses - required by the application developer while modelling the application (using for example Matlab). Finally, the museum application is being deployed on a cloud infrastructure.

2.1.1 Service Customer

Within the SLA lifecycle, the service customer refers to an entity that *obtains a service* and therefore signs an SLA with the corresponding service provider. There has to be noted, that customers may or may not be end users. In the aforementioned use case, the service customer would be the museum and not the visitors.

The main requirements of a service customer in the SLA lifecycle are *high-level application-related requirements* (e.g. delivery time of exhibits information in less than 10 seconds for 100 simultaneous requests from visitors). The goal of a service customer is to provide a service to end users with a specific level of quality.

2.1.2 Service Developer

The service developer actor refers to the application developer. While her goal is to develop a service, within the SLA lifecycle, she provides fundamental information regarding the service since she is the only *actor with application-specific knowledge*. The information refers to potential *dependencies* (in the case of a composite service that consists of atomic service components) as well as *performance / behaviour characteristics* of the application. In the exemplar use case, she is an employee of the software house enterprise while she is also using a framework that requires licenses (e.g. Matlab) for analysing the performance of the developed application.

2.1.3 Service / Platform / Infrastructure Provider

The ***service provider*** aims at *offering a service to the customer*. The development of the service, or its adaptation for cloud environments, is performed by service developers employed by the service provider. In the SLA lifecycle, the service provider will be the entity signing the SLA (including high-level terms) with the customer. However, the service provider may also sign an *SLA with a platform provider* to obtain / use a platform for developing the application or exploiting additional frameworks (e.g. license management). Moreover, the service provider may sign an *SLA with an infrastructure provider* to deploy the application. In the museum use case, the service provider is the software house enterprise.

The ***platform provider*** aims at *offering a platform* for the development of the service towards the service provider. In the SLA lifecycle, its role may be central if a service provider is not deploying the application on an infrastructure provider but only deals with the platform provider (thus the latter signs an SLA with the infrastructure provider). These *SLAs include low-level terms* (e.g. resource parameters).

The ***infrastructure provider*** aims at *offering an infrastructure* for the deployment and execution of the services. In the SLA lifecycle, it signs *SLAs including low-level terms* with service or platform providers.

These providers may use discovery or monitoring mechanisms to discover lower level potential providers (e.g. service provider discovering platform providers) and monitor the terms included in signed SLAs (e.g. infrastructure provider monitoring resource usage). The providers may also use additional frameworks (e.g. for business modelling) in order to optimize their offerings according to different criteria (e.g. pricing or business models). Additional information regarding the use of these mechanisms and frameworks is provided in the next section.

2.2 SLA Lifecycle Metamodel

This section introduces a metamodel (depicted in Figure 1) that captures the main phases, structures, processes and entities interactions in the SLA lifecycle. The goal of each phase, the participating actors and their role, the potential dependencies as well as the outcomes of each phase are described in the following paragraphs.

2.2.1 Service Use

Service use reflects the *usage of the cloud service* by a service customer. As already described in Section 2.1.1 the service customer may not be the end user. However, the aim of this phase is to obtain the service and thus an SLA may be signed between the customer and the service provider. The SLA includes high-level attributes related to the service / application.

2.2.2 Service Modelling

The service modelling process aims at *providing additional information* with respect to the service that will be deployed in a cloud infrastructure. As the only actor having the required knowledge for the service, the developer is using a set of frameworks in order to design, model and analyse the service. Service design may be extended to include potential dependencies between service components of an application (in the case of a composite service), elasticity rules for the application or / and performance and behaviour hints that are required to guarantee the offered level of quality (e.g. increasing number of users by a factor of 1000 in a multi-tier web application requires the usage of 3 times the deployed application servers and 2 replicas of the deployed database).

The outcome of the process is captured in an *artefact / document* (usually in a structured XML format), which includes all the parameters affecting the service execution, usage and delivery. This artefact is named in some cases Blueprint or Manifest.

2.2.3 SLA Template Definition

The SLA template definition process aims at *generating and refining the SLA templates*. All providers (i.e. service, platform and infrastructure) analyse their business objectives through a business modelling process (that may use business and pricing models simulation frameworks) in order to optimize their offerings. Furthermore, the service provider uses as a basis the blueprint / manifest of the service and refines the SLA templates (in terms of attributes values) following business modelling outcomes, while the service provider may also include additional attributes in the SLA templates reflecting for example the use of licenses. Thus, an SLA template may include the outcomes of one or more service blueprints / manifests.

The outcome of this phase is an *SLA template* that will be published by the providers in order to be negotiated and signed by the participating entities.

2.2.4 SLA Instantiation and Management

The goal of this phase is to instantiate an SLA (i.e. electronically signed agreement). The main process refers to the *SLA negotiation*, which may be extended with mechanisms for dynamic negotiation between different entities as well as with mechanisms for automatic re-negotiation during runtime. Moreover, discovery is used to identify providers for specific services (based on the service parameters captured in the service blueprint / manifest). Mapping / translation refers to a process of analysing the high-level application-related attributes and mapping them to low-level resource parameters (e.g. transmission of 24 frames per second maps to network links of 13MB/s). Besides such functional parameters, non-functional parameters (e.g. redundancy, security, etc) may also be mapped / translated.

The outcome of this phase is a signed *SLA between the participating entities* that includes low-level (resource-related) attributes.

2.2.5 SLA Enforcement

The SLA enforcement phase aims at *ensuring* that the *quality* parameters (agreed in signed SLAs) are retained. All providers exploit monitoring mechanisms to obtain both infrastructure and application monitoring data, while adaptable approaches focus on adjusting the monitoring time intervals or the monitoring metrics based on the collected information during runtime. Evaluation tools are exploited to analyse the monitoring data and trigger corrective actions using SLA violation detection mechanisms, some of which enable proactive violation detection.

2.2.6 SLA Conclusion

During the SLA conclusion phase, signed *SLAs are terminated successfully* (service delivery concluded or SLA validity period is over) or as *violated agreements*. The providers use accounting and billing mechanisms in order to provide the required information to the customers. If the SLAs have been violated, the corresponding compensations / penalties are calculated during the resolution process. Furthermore, in the case of multi-provider environments (e.g. cloud federations or composite services) resolution includes revenue sharing for the engaged providers.

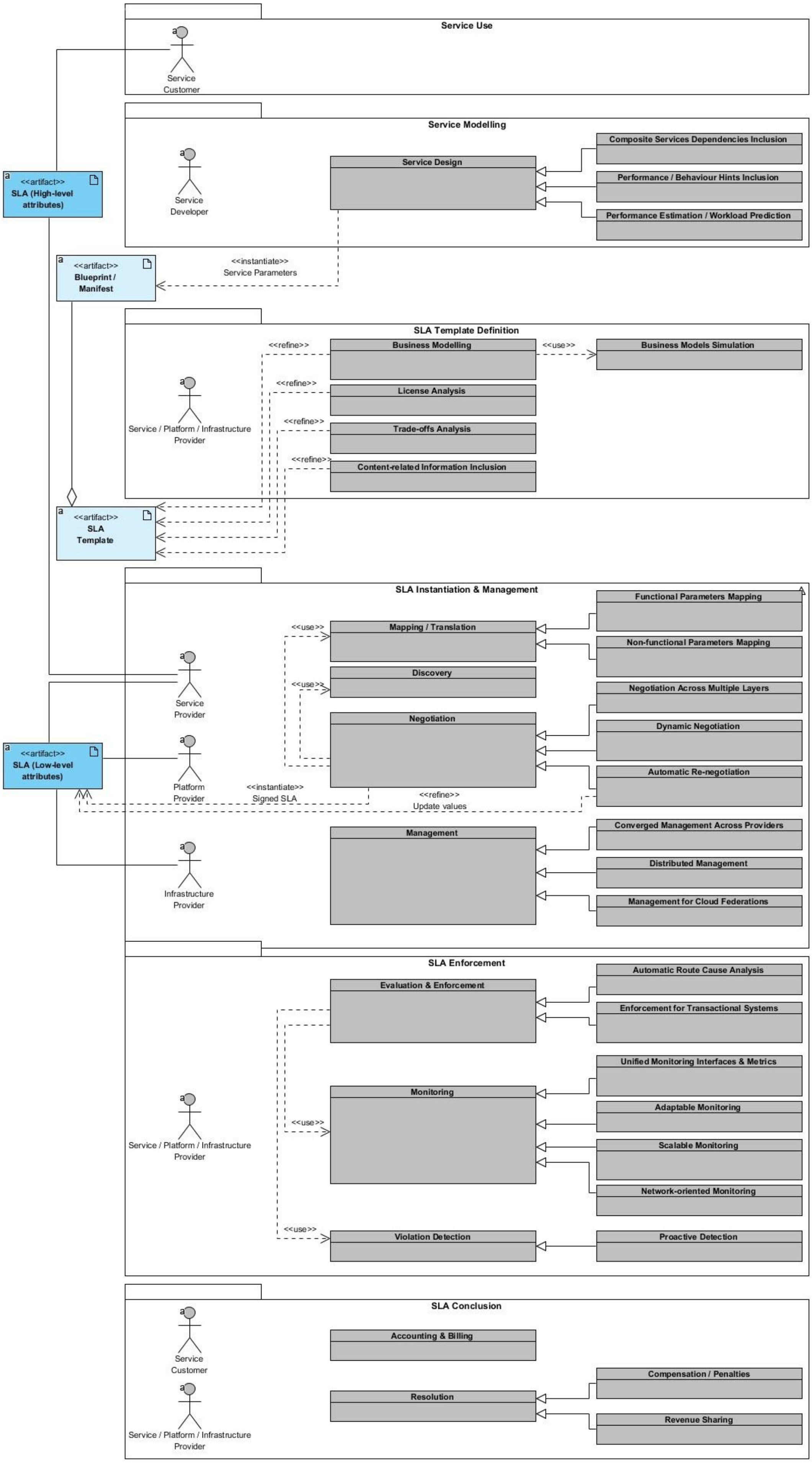

Figure 1: SLA Lifecycle Metamodel

3 Research Results

This section provides a brief overview of research projects (mainly European but also including some National projects) that have delivered SLA-related outcomes. These outcomes cover different and complementary aspects in the SLA lifecycle (e.g. specifications modelling, holistic management, cloud federations SLAs, real-time and storage clouds SLAs, SLA enforcement supporting mechanisms - such as scalability and QoS monitoring, etc).

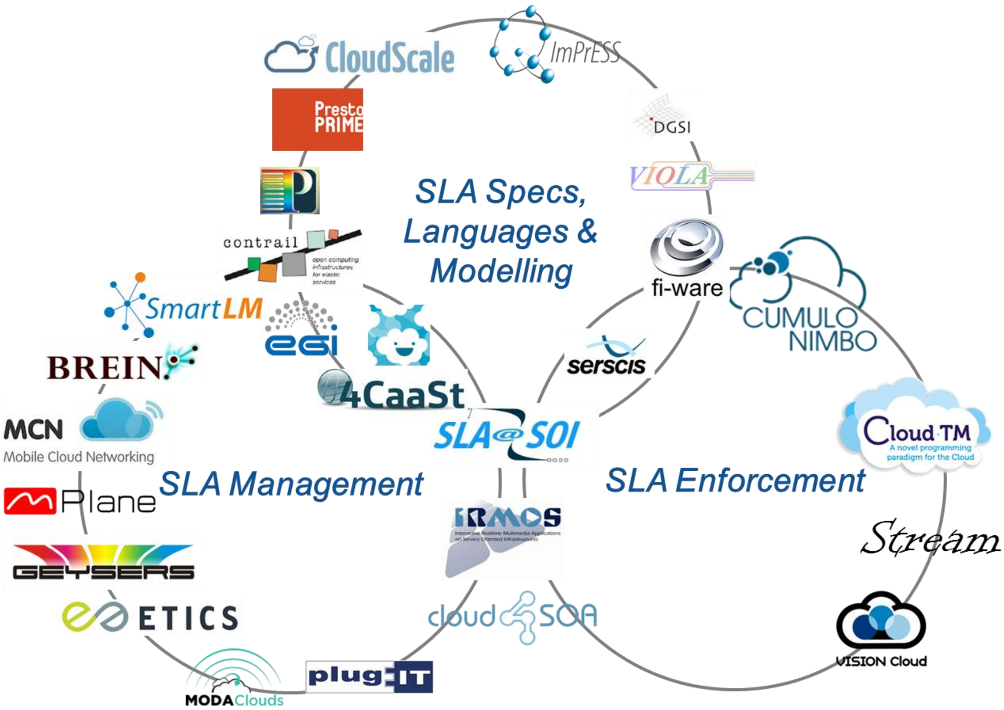

Figure 2: Projects outcomes addressing different and complementary SLA research areas

For each project, the main SLA-related outcomes are listed, while Section 3.24 provides the mapping of these outcomes to the SLA Metamodel introduced in Section 2 of this report.

3.1 4CaaSt

The 4CaaSt project [13] aims to create a PaaS Cloud platform [14], [15] which supports the optimized and elastic hosting of Internet-scale multi-tier applications. 4CaaSt embeds features that ease programming of rich applications and enable the creation of a business ecosystem where applications from different providers can be tailored to different users, mashed up and traded together.

3.1.1 Blueprint Concept

The approaches in 4CaaSt are based on the introduced concept of “products”, which refer to service offerings - atomic or composite ones - of any type (i.e. “X-as-a-Service”). In the case of composite services, what is of major importance is the *definition of the dependencies* between the atomic services. To this end, 4CaaSt has developed a description language capturing the service dependencies within and across the cloud layers, resulting to a descriptive document - the so called “blueprint” [16]. Besides the aforementioned dependencies, the description language enables the definition of provisioning and *management rules* with respect to elasticity and multi-tenancy, as well as the inclusion of “hints” from the application developer in order to map high-level application terms into low-level resource parameters.

What is more, the blueprint encompasses information with respect both to the technical requirements of a product (e.g. through specific KPIs), and to the *business aspects / terms* of such a service offering. The latter is a unique contribution from 4CaaSt, since the use of the eMartkerplace (described in the next section) allows for the optimum identification and selection of the technical terms that should be attached to a service offering through an SLA. Taking into consideration that there is a great degree of flexibility in the application and technical terms, as defined by the application developers (e.g. range of values in a specific parameter), business criteria and simulation aim at *identifying the optimum terms and the corresponding values* for these terms.

3.1.2 eMarketplace

The project has implemented an eMarketplace framework [17] that deals with the *business and pricing aspects* of service offerings [18]. It enables trading of any type of cloud services, including *composite services* that consist of atomic services offered by different providers. Furthermore, the eMarketplace is enriched with a business model simulation tool supporting the service providers during the identification and definition of complex pricing and business models. Through its *business resolution* feature [19], it exploits the experience of end users and customers and proposes business offering which effectively cover the needs of each particular request from a pool of technically valid solutions. Based on the above, the eMarketplace could be considered as a supporting environment during the definition of SLA templates.

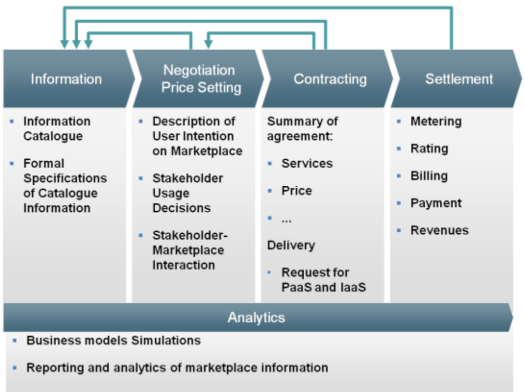

Figure 3: 4CaaSt eMarketplace

3.1.3 Elasticity Management

The mapping process between *high-level application terms to low-level resource parameters* has been enriched by 4CaaSt in order to cover aspects of *elasticity for composite / complex service offerings* [20]. The latter highlights the need for elasticity management considering that the composite applications consist of different atomic services, which may require for different provisioning policies (i.e. elasticity management) either based on their technical and business requirements or based on their interdependencies. Based on the above, 4CaaSt elasticity management is considered as an essential mechanism for SLA enforcement given that future applications are composite ones.

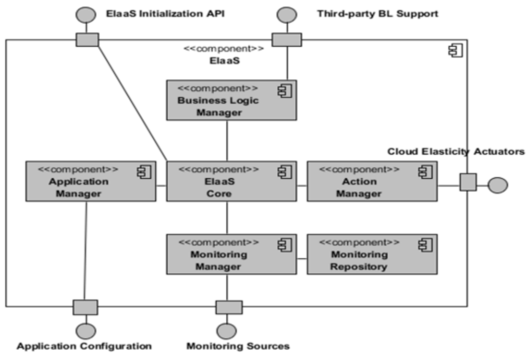

Figure 4: 4CaaSt Elasticity management

3.2 Cloud4SOA

The project [21] empowers a multi-cloud paradigm at PaaS level, providing an interoperable framework for PaaS developers. The system supports Cloud-based application developers with multiplatform matchmaking, management, unified application and cloud monitoring and migration. It interconnects heterogeneous PaaS offerings across different providers that share the same technology through the concept of adapter that provides a REST-based API for any-platform access.

3.2.1 Unified Monitoring Interface and Metrics

Cloud4SOA has identified the challenge that exists with respect to provide a *unified platform-independent mechanism to monitor* the health and performance of business-critical applications hosted on multiple cloud environments in order to ensure that their performance consistently meets expectations defined by the SLA. In fact, different providers use different metrics and deliver the data by implementing specific APIs. To address this challenge, the project has developed *unified interfaces* that overlook all customers' deployments at once, thus allowing customers to compare and evaluate different deployments. This could be performed externally through the REST API or internally by Platform Components.

Furthermore, a *set of unified metrics* (across PaaS providers) has been selected to monitor the application execution and usage. These are both *application-level metrics* (defined through a library embedded in the source code of the application) and *infrastructure-level metrics* (using the interface of the provider). Currently the following set of metrics and the corresponding APIs have been developed: application / database response time, cloud response time, web container response time, application status, memory usage and CPU usage.

3.2.2 Dynamic SLA Negotiation and Enforcement

Support for on-demand based business models is amongst the requirements of cloud providers. To this end, dynamicity needs to be embedded in the SLA lifecycle in order to support business dynamics and changing customer needs (e.g. redefine specific parameters). Cloud4SOA has developed a framework enabling *dynamic SLA negotiation* and tools that enable PaaS providers to analyse their offerings and performance and adapt the SLAs accordingly. The framework allows providers and customers to *negotiate flexibly between standard and customized SLAs*, while supporting business dynamics through *business-performance related SLA metrics* being monitored and analysed.

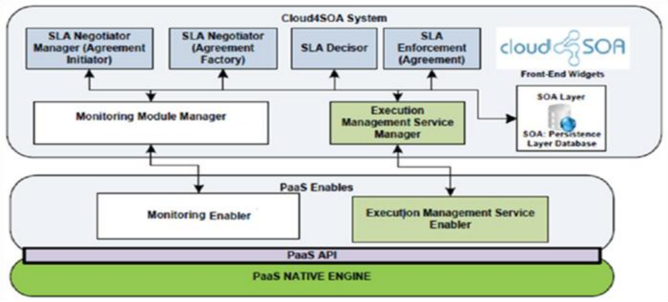

Figure 5: Cloud4SOA SLA management architecture

Cloud4SOA provides a RESTful implementation of the WS-Agreement standard. On top of the implementation the Cloud4SOA governance layer offers three main functionalities that enable users *negotiate and enforce SLA*, as well as *recover from SLA violations*, through (i) Agreement Negotiation, which allows the automatic negotiations on behalf of PaaS providers, based on the semantic description of offerings and the QoS requirements specified by application developers; (ii) Agreement Enforcement, to supervise that all the agreements reached in a SLA are respected (i.e. measurements are within the thresholds established in SLA for QoS metrics); and (iii) Violation recovery. Whenever the execution of the business application does not satisfy the SLA (i.e. breaches of the agreement occurs), the most appropriate recovery action (e.g. warning messages, stop or migration of the application) is suggested based on the policies defined by the software developer.

3.3 CloudScale

The project [22] aims at supporting scalable service engineering. In this context, mechanisms are developed to support service providers in analysing, predicting and resolving scalability issues in cloud environments [23]. CloudScale among other things focus on scalability aspects (i.e. changing needs for infrastructure resources needed during runtime) and their incorporation in SLAs (i.e. quality requirements / attributes for scalability).

3.3.1 Scalability Specification

CloudScale will develop the ScaleDL (Scalability Description Language) which will characterise the scalability requirements of a service. ScaleDL (harmonised with MARTE - Modeling and Analysis of Real-time and Embedded Systems [24]) will be especially targeted at analysing the scalability of composed cloud services. ScaleDL allows specification of all the relevant information about the usage, the software layer, deployment, and cost in order to enable scalability analyses of services.

3.3.2 Automatic Root Cause Analysis

CloudScale is developing mechanisms to *identify causes of potential SLA violations*. When the services do not scale as expected, root causes of the scalability problems based on sources are identified. This analysis is done *based on the source code* for the service. To find out what to do with this scalability problem, a scalability model may be extracted from the same source code. Based on this scalability model a what-if analysis can be performed to find good ways of resolving the scalability issue, for example by using a different cloud provider, or by changing the implementation of the source code. If no viable solution is found, the scalability requirement specified in the SLA may have to be relaxed.

3.4 Cloud-TM

Cloud-TM [25] develops a data-centric PaaS layered on top of a self-optimizing, highly scalable distributed Transactional Memory platform. Cloud-TM allows for reducing the development and operational costs of cloud-based applications in a twofold way: i) hiding complexity by providing programmes with intuitive abstractions that encapsulate innovative data management protocols designed from scratch to meet the requirements of large-scale elastic cloud platforms; ii) via pervasive self-tuning strategies that automate the resource provisioning process [26], [27] and transparently reconfigure the data management mechanisms (e.g. consistency protocols [28], [29], [30], data placement [31], replication degree [32], [33]) based on user-specified QoS/cost constraints [34].

3.4.1 Performance Estimation and Workload Prediction

The provision of both the initially required resources to services (during deployment) and the additional resources (during runtime), require for mechanisms that deliver *performance estimates* [24], [27], [31], [33] and *workload predictions* [34] in order to identify the optimum resources and deployment patterns. Such mechanisms have been developed by Cloud-TM, enabling the prediction of applications' performance when deployed over transactional platforms of different scale as well as the workload prediction of the transactional application independently from the scale of the system, the capacity of the platform (e.g. CPU speed), the data management scheme and the algorithm used by the transactional data platform on which the application is deployed.

3.4.2 SLA Definition and Enforcement in Transactional Data Stores

The project has developed an innovative approach for managing and enforcing SLAs when dealing with transactional cloud data stores. The approach is realized through a framework enabling *self-optimization and self-tuning of the infrastructure resources* based on different QoS metrics [34].

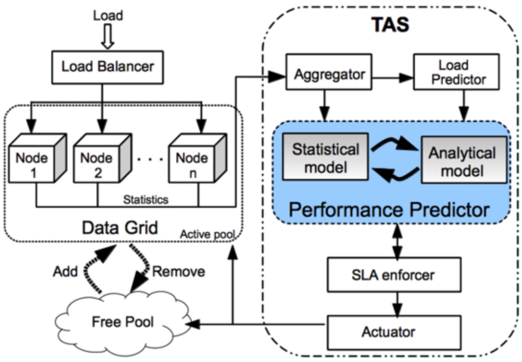

Figure 6: Cloud-TM Transactional Auto Scaler

It triggers in an automated way elastic scaling while ensuring consistency through adaptive data placement schemes [31]. A unique aspect of the Cloud-TM platform consists in its ability to continuously *self-tune its data management protocols* [27], [30] during service usage in order to enforce SLAs. The overall approach allows for overcoming issues related to contention (due to the inclusion of additional resources) both on the logical layer (through the corresponding data management) and on the physical layer (through resource management based on dynamic resource provisioning).

Cloud-TM leverages on the SLA@SOI framework to define SLOs between the Cloud-TM platform's provider and service developer. In particular, Cloud-TM defines custom SLA templates that allow for *negotiating domain-specific QoS levels*, such as constraints on the response time and abort rate of different transaction profiles.

3.5 CONTRAIL

The main objective of CONTRAIL [35] is to offer elastic PaaS services over a federation of IaaS Clouds, while dealing with pertinent issues related to QoS, SLA management, security, interoperability and scalability. In the CONTRAIL vision, small Cloud providers can join forces into a Cloud Federation to stand the competition of bigger players and raise at a worldwide level the competitiveness of the European Cloud market [36].

3.5.1 SLA Specification

To express QoS guarantees CONTRAIL *adopted the SLA(T) model* proposed by the project SLA@SOI (described in Section 3.19.1) and extended it to *use a standard OVF descriptor to specify virtual resources*. VirtualSystems represent classes of Virtual Machines, SharedDisks represent external storage and all the elements can be connected in complex layered architectures through VLANs identified in the NetworkSections. To monitor / enforce SLA terms it must be known what the expressed guarantees are referring to and there should be a link between *SLA terms (guarantees)* and *OVF items (resources)*. Contrail extended the SLA@SOI syntax to create such a link. To allow for scalability SLAs in CONTRAIL define the quality but not the amount of resources (except when advance reservation is used). Automatic scaling can be implemented by actions specified in the SLA (Guaranteed Actions) that ask for more resources when warning thresholds are violated (proactive SLA violation detection).

3.5.2 Quality Model

The project has developed an innovative quality model for capturing different parameters of interest for customers and providers. Within the quality model, *terms* have been classified to: *unobservable*, *observable*, *enforceable* as well as to: *static* or *dynamic* (regarding their evolution in time). The quality model has been used to develop an SLA specification that reflects either *generic SLAs* (i.e. parameters applicable to any resource) or *specific SLAs* (i.e. parameters applicable to specific OVF resources). Besides QoS terms and advance reservation, the SLA specification includes the so called *Quality of Protection (QoP) terms*, such as data locality, protection, replication, etc, and it may also be linked with different pricing models for generating automatic quotations.

3.5.3 Multi-level SLA Interaction Model

CONTRAIL focuses on cloud federations and proposes a model, based on *automated SLA offer generation*, in which the user negotiates a *SLA with the federation* and the federation looks for the best way to satisfy it by negotiating SLAs with one or more providers (on behalf of the user). SLAs can be linked to capture interactions in multi-provider environments. Regarding SLAs for applications spanning multiple providers, an innovative scheme for *SLA splitting* has been proposed, which allows for service-, resource-, or performance-based SLA splitting and revenue sharing / compensation provision.

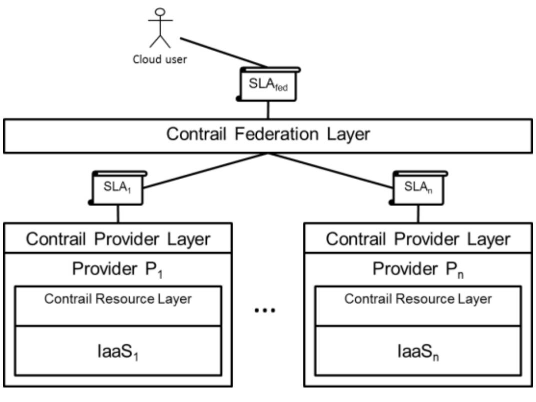

Figure 7: CONTRAIL SLA interaction model

3.5.4 SLA Management for Cloud Federations

Enhanced mechanisms in different phases of the SLA lifecycle have been developed by CONTRAIL to support SLA management for cloud federations. Regarding negotiation, a system has been implemented (based on the SLA@SOI framework) to realize the *federated negotiation with multiple providers* and the *selection of the optimum SLA offer* according to user criteria. In this case the CONTRAIL system acts as a cloud broker, realizing the Service Arbitrage model described by NIST in its cloud computing reference architecture [1]. During service execution / usage, CONTRAIL will allow for application distribution over multiple providers (thus enabling the execution of composite applications), while cross-*provider enforcement strategies* will be exploited to minimize SLA violations.

3.6 CumuloNimbo

CumuloNimbo [37] has developed a PaaS solution that provides high scalability without sacrificing data consistency and ease of programming. The transactional management system can be integrated with any data management system (databases, NoSQL data stores, SQL engines) and software stack (e.g. Java EE, LAMP, etc.).

3.6.1 SLA Enforcement for Transactional Systems

Given that one of the hardest questions in SLA enforcement is how to deal with an increase of the load in the system, CumuloNimbo has developed a solution for facilitating scalability and elasticity for transactional systems [38], [39]. Emphasis is put on how cloud data stores can accommodate full coherence, which requires *ACID* (Atomicity, Consistency, Isolation, Durability) transactions. The latter is of major importance for *SLA enforcement* since specific parameters related to the load of the system can only be addressed through scalability and elasticity but the goal is to retain coherence. To this direction, the project has developed mechanisms to deal with the *scalability* (and thus SLA enforcement) of *transactional systems* when exploiting cloud infrastructures.

3.7 EGI Federated Clouds Infrastructure

EGI [40] as a federation of European Resource Infrastructure Providers is working towards providing a federated IaaS Cloud infrastructure for Research Communities accessing and consuming the provided federated Cloud Computing and Cloud Storage resources.

EGI will develop SLA templates for easy and perhaps automated instantiation by small research communities that cannot afford spending effort on negotiating customised SLAs. These *pre-populated SLAs* will be supported by a framework of *Operational Level Agreements (OLAs)* that EGI will put in place with its resource providers and resource infrastructure providers, reflecting the federated nature of EGI. Complementing the OLA framework, underpinning contracts with Technology Providers and Service Providers will ensure service continuity on the technical level. In the beginning, EGI will offer SLAs to Research Communities that focus on quantifiable technical availability and reliability of the included services (i.e. Cloud Computing and Cloud Storage). With further maturing of the EGI federated Clouds infrastructure to include new and/or more matured services, *new SLA service level targets* will be gradually included covering both quantifiable and qualifiable service level targets, for example metrics around service performance (bandwidth, response times, etc.), privacy and data protection, confidentiality, data provenance, retrieval, data retention times, preservation guarantees, etc.

3.7.1 Service Catalogue in a Federated Environment

Through dedicated consultancy as a client partner within the FedSM project [41], an EGI Service Catalogue [42] was developed and published, which refactored the EGI-InSPIRE activities based on the ITIL framework, the *de facto standard for operating computer centres*, to organise the services being provided from the organisation viewpoint regardless of the project structure. As an open ICT ecosystem that relies on contributions from a wide range of organisations, such as third party technology providers and product teams for the required software or innovation, clearly defining what services are being offered and by whom will allow for the most appropriate and effective agreement to be established.

3.7.2 Federated Service Management

The federated nature of EGI requires an unusual approach to service management. In line with the EGI service catalogue, the services covered by an SLA with a research community will altogether be delivered by EGI.eu, resource infrastructure providers and resource providers. To ensure and formalise the service delivery, EGI employs an *OLA framework* as a mechanism to integrate resource providers into the pan-European EGI production infrastructure while ensuring interoperation of operational services, QoS, and to enforce a common set of policies and procedures. Consequently EGI OLA framework incorporates three types of OLAs: (i) The *Resource Centre OLA* [43], which defines the relationship between a local Resource Centre (RC) and the respective (often national) Resource infrastructure Provider (RP), (ii) The *Resource Infrastructure Provider OLA* [44], which defines the relationship between the Resource infrastructure Provider, its affiliated Resource Centers, and EGI.eu, and (iii) The *EGI.eu OLA* [45], which defines the global services EGI.eu provides in collaboration with its EGI partners to the Resource Infrastructure Providers.

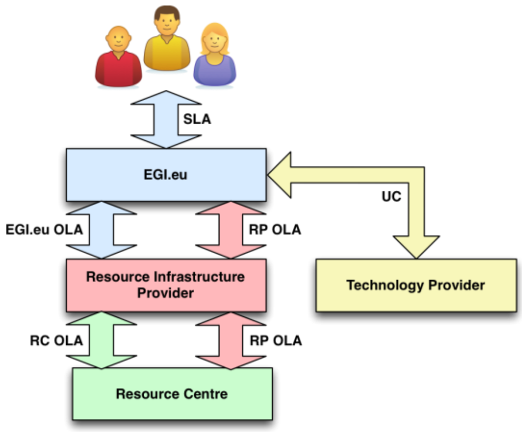

Figure 8: EGI service management framework

3.8 ETICS

ETICS [46] has delivered new network control, management and service plane technologies for the automated end-to-end QoS-enabled service delivery across Network Service Providers allowing for a fair distribution of revenue shares among all the actors of the service delivery value-chain.

3.8.1 SLAs for Composite Services

The project has considered the case of e*nd-to-end, QoS-enabled application services* resulting from the *composition of atomic services* being offered by different providers (e.g. application / content and network providers) in application and network domains. To support the provision of such composite services, a *hierarchy of SLAs* has been defined in reference to the different composition layers. (i.e. SLAs for inter-carrier services). At the atomic service layer, the interconnections between the network providers, between application and network provider and between end-user and network provider are characterized by *static SLAs*, while the intra-domain network services offered by the each network provider follow a *dynamic and* *per-service* paradigm. The composition of the SLAs related to the network intra-domain services and interconnections, results in the SLA for the end-to-end, inter-carrier network service that, in turn, can be further aggregated with the SLA for the atomic application service [47]. The final resulting SLA on top of this hierarchy will deal with the end-to-end, QoS-enabled and network-guaranteed application service. Depending on the service chain, the SLAs for composite services consider as providers either network providers or application providers and as customers either application providers or end users. The latter highlights the fact that composition always follows a provider – customer scheme but the customer in some cases may be another provider.

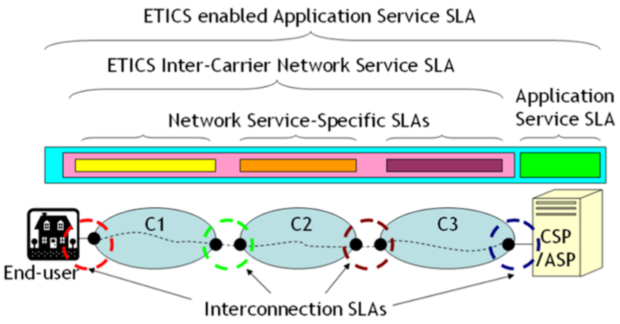

Figure 9: ETICS SLAs for inter-carrier services

Furthermore, the project has contributed towards the identification and realization of different *SLA composition paradigms*. SLA composition may be *centralized* (i.e. a unique entity such as an independent broker or origin domain acts as mediator and manages the SLA with all the domains) or *distributed* (i.e. consecutive SLA establishments on each provider-customer pair following either a cascade model - from origin to destination, or a reverse cascade model - from destination to origin [48]).

3.8.2 Business-enhanced SLA Template

Besides the technical aspects being captured in SLAs, ETICS has proposed an approach for the flexible *integration of business aspects* in the SLA lifecycle [49]. To this end, an SLA template has been developed which is flexible in terms of different *business or charging models*, while meeting general requirements on domain confidentiality and technology heterogeneity. The main components of the SLA template refer to the entities (i.e. customer, providers, or brokers) identification, the service description (i.e. technology agnostic description of service attributes), the business aspects (i.e. price, administrative / legal details, procedures for handling service modification / violation / termination cases) and the technical aspects (i.e. QoS parameters).

3.9 GEYSERS

The project [51] has delivered mechanisms for seamless and coordinated provisioning of networking and IT resources, end-to-end service delivery to overcome limitations of network domain segmentation, business models analysis through a business framework and composition of logical infrastructures following the partitioning of infrastructure resources.

3.9.1 Converged SLA Management for Composed Virtual Infrastructures

Composed virtual infrastructures [50] aim at enabling dynamic *service provisioning on top of network and IT resources*, encompassing several layers and resource types while dealing with the constraints and dependencies of the resources and the services as well as the dependencies between application deployment and usage [52]. GEYSERS considers the *autonomy of physical and virtual providers* as well as virtual operators, and their respective management domains (independent control of policies and operational objectives). The converged SLA management framework proposed and implemented in GEYSERS [53] aims to *handle the* *dependencies between physical and virtual resources* in both network and IT domains, and allow for cross-layer handling of events and alerts that may affect the service provision on top of the virtual infrastructure and their SLA lifecycle. The converged SLA management framework [54], [55] implements *different strategies*: *bottom-up* (i.e. initiated by the lower-layer physical infrastructure providers), *top-down or "truly on demand"* (i.e. initiated by customers and service consumers), *mixed* (i.e. combined message exchanges to reach mutually-agreed SLA).

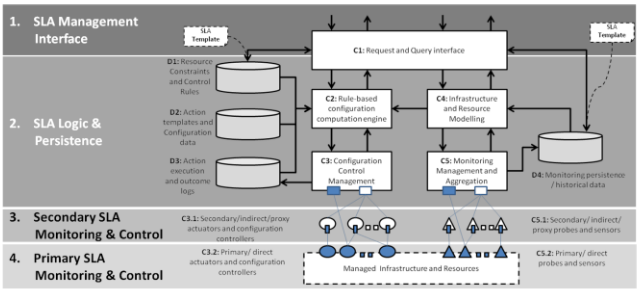

Figure 10: GEYSERS SLA management

3.10 *Helix Nebula*

The project [56] aims at establishing a multi-tenant, multi-provider cloud infrastructure, while identifying and adopting policies for trust, security and privacy, and introducing a governance structure and the related potential funding schemes.

3.10.1 Common Catalogue of Services

The project has identified the need for a common catalogue of services that will include *common / basic information from all providers* as a basis, and will be *linked with specific catalogues* of each provider with additional details regarding the offered services. The information in the catalogue refers to *service attributes* (e.g. availability) that are measurable and capable to describe the *business-relevant attributes* of what is being delivered. Besides, *service classes* will be included in order to describe what kind of attributes each kind of resource needs to describe (e.g. storage service class describes I/O speed or capacity), as well as resource groups to identify what is being described with each attribute (e.g. 99% availability for server provisioning interface as accessible via a specific defined URI).

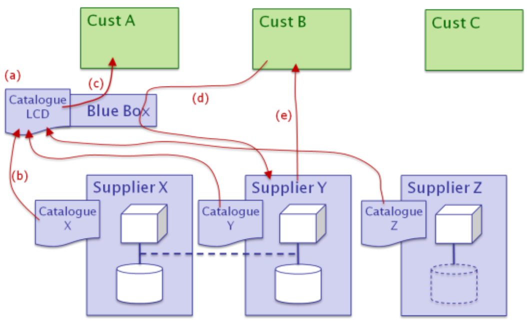

Figure 11: Helix Nebula common catalogue of services

3.11 *IRMOS*

The project [57] developed cloud solutions that allow the adoption of interactive real-time applications, enabling their rich set of attributes (from time-constrained operation to dynamic service control and adaptation) and their efficient integration into cloud infrastructures [58], [59].

3.11.1 SLAs at Different Levels

Since there may be different actors in the SLA lifecycle, IRMOS has introduced *two different types of SLAs* to capture the diverse requirements and the different abstraction level of these requirements [60], [61], [62]. This is the main reason why there should exist different one-to-one agreements between the actors. To this end, IRMOS has proposed two types of SLAs, namely application and technical. The *application SLA* is used by the customer to express her parameters in high-level application terms towards service providers, while the *technical SLA* is used for agreements between for example platform and infrastructure providers, and includes low-level resource parameters.

3.11.2 Dynamic SLA Re-negotiation

Most existing SLA management procedures consider the negotiation to be a process that takes place before the execution phase. Once the negotiation has been produced the service is monitored against the corresponding SLA established and in case there is a violation (or potential violation), then several actions (reflected in the SLA in most cases) are performed. However, sometimes the causes and origin of the violations could be addressed by establishing again a process of negotiation. This is called *SLA re-negotiation* and may be triggered either by the user (e.g. change in application parameters that affect the QoS), by one of the providers (e.g. detection of potential SLA violation) or by the application (e.g. scalability rules) [64]. SLAs can be *updated during runtime* to reserve additional resources following a user request or a corrective decision from the platform in order to maintain the requested QoS [65]. A prerequisite in cases of renegotiation is always the availability of the additional resources in the infrastructure layer; otherwise the re-negotiation of the SLA may be rejected [66].

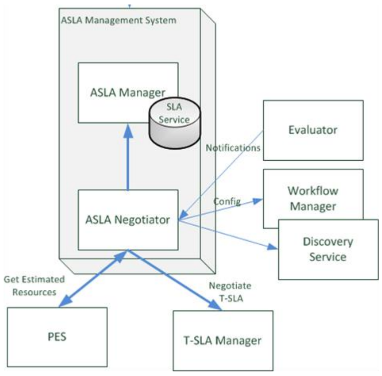

Figure 12: IRMOS SLA management

Another important aspect in the SLA management within IRMOS is the *automatic way of negotiation and re-negotiation*, thus performed without human intervention but based on policies defined by the actors involved in the negotiation.

3.11.3 Adaptable Monitoring and Evaluation

As a basis for real-time application execution, a *monitoring and evaluation* framework has been developed that collects information from both *application* (high-level performance metrics) and *infrastructure levels* (low-level resource utilization metrics) and evaluates the monitoring data against expected QoS to support runtime decision making [67], [68]. The monitoring framework follows a *hierarchical architectural approach* (i.e. monitoring instances also reside in the VMs hosting the deployed services in order to obtain application-related monitoring data), while being *adaptable in terms of monitoring time intervals* (based on the collected monitoring information and the corresponding SLA terms) to minimize the footprint on the system and network overheads when propagating monitoring information.

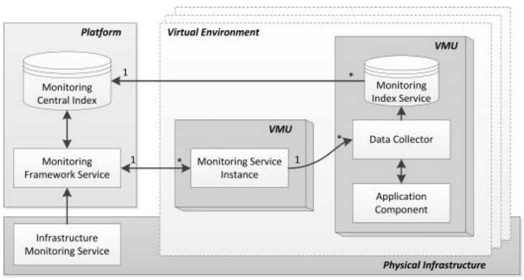

Figure 13: IRMOS Adaptable monitoring framework

3.11.4 Mapping High-level to Low-level Attributes

The *high-level application terms* included in application SLAs (or captured in service blueprints / manifests) need to be mapped to the *low-level resource estimates* in order to enable service execution according to these terms that define a specific quality level. The process is achieved with mapping frameworks that translate these high-level application QoS requirements (like resolution of the video, application end time etc) into low-level resource parameters that are required in order to meet the end user constraints [69].

IRMOS has developed a mapping mechanism that bases translation on an (Artificial Neural Network) ANN-based rule / model, which depicts the relationships between the service characteristics (as inputs), the different hardware configurations and the resulting QoS levels [70].

3.12 MCN

MCN (Mobile Cloud Networking) [71] will develop a fully cloud-based mobile communication and application platform, by delivering a system of mobile network enhanced with decentralised computing and smart storage offered as one atomic service with on-demand, elastic and pay-as-you-go characteristics.

3.12.1 Distributed SLA Management

The main goal of the proposed distributed SLA management framework is to support *SLAs for composite services* by enabling *combined and joint management* of the multiple SLAs for atomic services. While a top-down SLA propagation approach is proposed for the dynamic provisioning of on-demand services, SLA management (i.e. monitoring and enforcement through continuous validation) will be performed in a distributed way through *lightweight SLA agents* that will interact with the monitoring service.

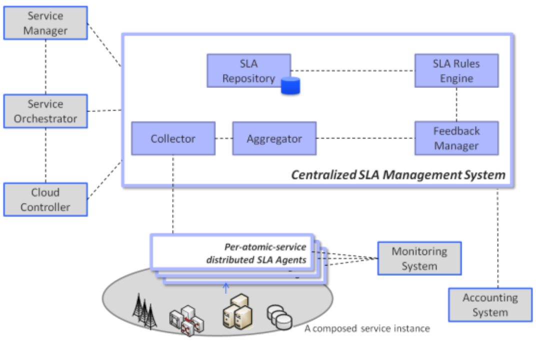

Figure 14: MCN SLA management

These refer to agents that will be deployed with the cloud services (being part of the services) per resource in order to facilitate the required management decisions.

3.13 MODAClouds

MODAClouds [72] will provide methods, a decision support system and an open source IDE and run-time environment for the high-level design, early prototyping, semi-automatic code generation, and automatic deployment of applications on multi-Clouds with guaranteed quality of services.

3.13.1 Unified Monitoring

MODAClouds is developing an approach for enabling *unified monitoring across different cloud layers* (i.e. IaaS and PaaS) in order to support runtime decisions and provide QoS guarantees based on the definition of QoS constraints (hard and soft constraints).

3.13.2 Runtime Re-negotiation

SLA enforcement in MODAClouds is based on *triggers for adaptation* in case of violations during runtime. To this end, automatic triggering of corrective actions is linked to automatic re-negotiation of SLAs for the respective QoS terms.

3.14 mPlane

The project aims at developing an intelligent measurement plane for the Internet in order to collect and analyse measurements in large scale networks.

3.14.1 Network Monitoring for SLAs

mPlane has developed an approach for *defining SLAs according to the OSI layer* (i.e. layer 1-2 to verify the SLA between the ISP and the user, layer 4 to capture the user requirements, and layer 7 to capture the user experience) and monitoring the delivery of network services according to these SLAs. The SLA measurement definition has been performed according to different accesses (i.e. xDSL, FTTx, and 3G-4G) and aims at collecting monitoring data for network resources (i.e. data transmission speed, packet delay, loss ratio and unsuccessful transmission ratio). Moreover, analysis of the users' experience (Quality of Experience - QoE) is performed through a Mean Opinion Score (MOS) for objective and subjective evaluation.

3.15 OPTIMIS

The project [73] aims at enabling organizations to automatically externalize services and applications to trustworthy and auditable cloud providers, while optimizing the complete lifecycle of service engineering, provision, operation, delivery and use.

3.15.1 Service Manifest

The service manifest [74] can be considered to be a *term language*, enabling the description of the requirements of the service provider for an infrastructure service provisioning process. It captures both the *functional* and *non-functional parameters* of the service and allows the specification of the (atomic) components of an application (e.g. a web-application for example may require a web server, an application server and a database server to run). For each component, the corresponding (potentially different) requirements / parameters are captured along with the constraints for each requirement, and the KPIs that will be monitored, while affinity and anti-affinity rules can also be specified per component.

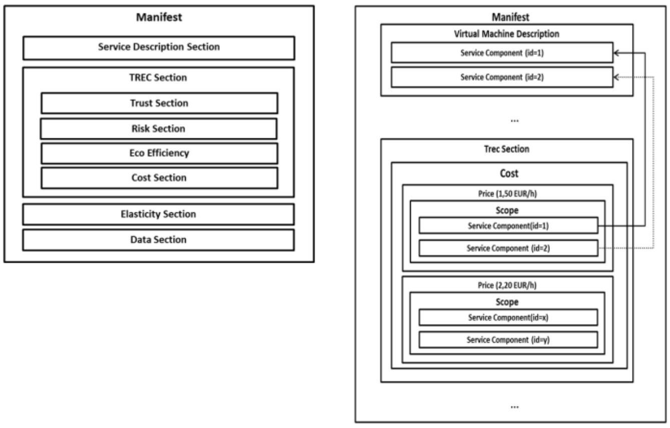

Figure 15: OPTIMIS service manifest

The developed service manifest by OPTIMIS consists of *different elements*: the *common core service manifest*, the *service provider extensions*, and the *infrastructure provider extensions*. With respect to infrastructure services (multiple can be described in one document), the manifest may describe VM images (OVF) and may also include OVF definitions of data location constraints, data protection, and elasticity (derived from the service manifest of the RESERVOIR project). Furthermore, the manifest may include *legal terms* such as *Intellectual Property Rights (IPRs)*, standard *contractual clauses* or *binding corporate rules* [75], [76]. To this end, OPTIMIS has highlighted how IPR categories can be exploited during automated SLA negotiation [77]. Besides the manifest, OPTIMIS has developed an API to develop, import and export the service manifest, refine service and infrastructure providers' extensions, and split it if needed since multiple services may be described in a single document.

3.15.2 Automated SLA Negotiation

SLA negotiation is a process that may be undertaken by various roles and includes a number of specialized terms. In OPTIMIS, the different deployment and runtime configuration scenarios include different cases such as *private, bursting, federated and multi-cloud deployment*. During a bursting the internal provider negotiates with a public cloud provider in order to acquire resources for a load peak. In the case of the multi-cloud, an intermediate entity, the cloud broker, undertakes the role to find resources possibly expanding to different clouds in order to meet the specific requirements posed by the service customer/owner. These requirements may span across a variety of factors, such as service or *provider risk, trust, ecological or cost levels* (TREC) [79], [80], *legal requirements* (when dealing with personal data) or simple *non-admission of the entire service* by a provider due to lack of resources. Finally, in the federated scenario, the infrastructure provider may (during deployment or runtime) *split the service manifest* (e.g. in deployment if an admission controller specifies that the internal resources are not sufficient for the current or future needs of the service) in order to keep one part internally and use a different provider (transparently to the service provider level) for the non-admitted part. The negotiation framework extends WS-Agreement for multi-round negotiations and enables the interplay between legal and flexible service provisioning based on the legal terms included in the SLAs.

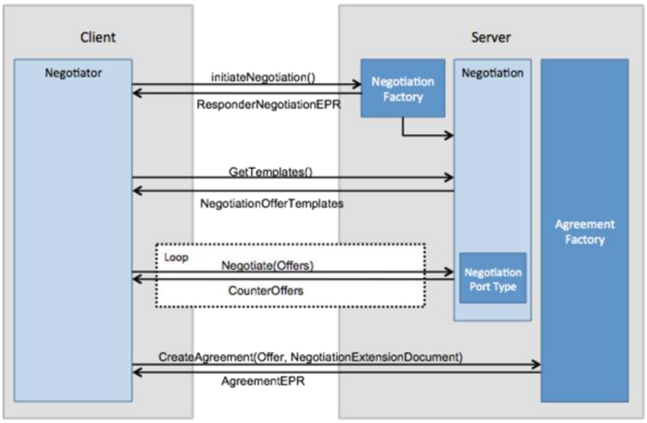

Figure 16: OPTIMIS SLA negotiation

3.16 PrestoPRIME

The PrestoPRIME project [81] developed a service management infrastructure for the long-term preservation of audio-visual digital media objects, programmes and collections. The preservation of digital audio-visual assets is performed by a "service provider", whether this service provider is the same organisation as the producer and consumer, an out-sourced operation but on the same premises, completely out-sourced or even standalone. In this context, the interactions of the preservation service with producers and consumers are defined and managed through service level agreements (SLAs).

3.16.1 SLA Specification for Preservation Services (risk of data loss)

A framework for gathering SLA terms for Preservation Services was developed. The framework included 21 capabilities (e.g. ingestion, delivery, validation, demux, fast preview), 12 features of interest and 15 metrics (e.g. availability of services, storage occupation, SIP ingestion time, DIP conformance), 12 quality of service terms (e.g. set threshold on the SIP ingestion time), 4 constraints (e.g. maximum number of simultaneous users), 6 pricing terms (e.g. yearly subscription charge and a data movement charge), and 7 penalty terms (e.g. payable when file integrity is lost).

To support a system that maintains the required quality of service, the SLAs and the monitoring data are used by the service provider in the capacity management process. Capacity management systems range from “we’ve got another customer: buy some more tapes”, through back of the envelope estimations, spreadsheets and semi-automated models to automatic decision support services. A variety of techniques are supported. Automatic monitoring, reporting and capacity management in complex IT systems are achieved by SLAs understood by the system itself.

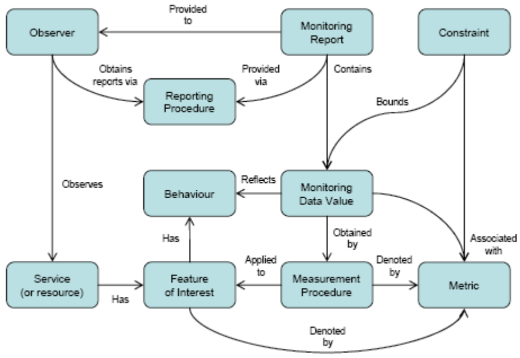

Figure 17: PrestoPRIME Specification for preservation services

3.17 Q-ImPrESS

The project [82] has developed a method for quality-driven software development and evolution, where the consequences of design decisions and system resource changes on performance, reliability and maintainability can be foreseen through quality impact analysis and simulation.

3.17.1 QoS-oriented SLA Specification

Focusing on how different QoS / SLA parameters can be captured, Q-ImPrESS has developed a *metamodel* (namely Service Architecture Meta Model – SAMM [83]) that allows the *definition of service attributes*. *Parameterised definitions* (e.g. data volume, configuration, execution environment) are feasible in SAMM. *Parametric dependencies* can be included in the model, since service architectures may include more than one services (as composite services) and face varying usage contexts. The dependencies are exploited to link different SLAs and incorporate the corresponding relationship structure among service SLAs. Based on analysis results calculated from the model, SLA estimations can be generated.

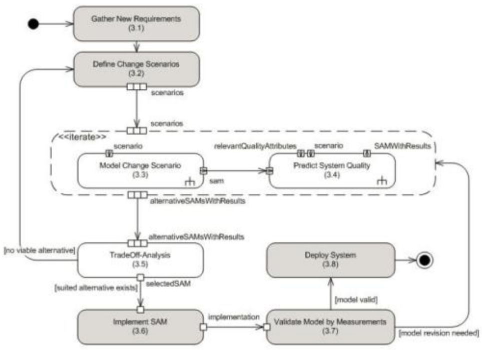

Figure 18: Q-ImPrESS SLA specification process

3.17.2 Trade-off Analysis and SLA Prediction

One of the challenges in cloud environments refers to the way different application characteristics (in the case of Q-ImPrESS expressed in SAMM) and providers’ policies affect the QoS level of the service and the resource provisioning decisions [84]. To address this challenge, the project has developed a trade-off analysis framework that concludes on the *effect of different QoS attributes*, while considering *different states* (i.e. target and as-is) during the service lifecycle [85]. Analysis also aims at *estimating performance metrics*, *accounting for propagation effects* across systems, and *assessing the risk for potential SLA violations* in order to propose design alternatives [86]. The outcome of these frameworks is an SLA prediction in terms of balanced SLA quality dimensions (as reflected in different attributes / parameters).

3.18 SERSCIS

SERSCIS [87] developed an adaptive service-oriented infrastructure for creating, monitoring and managing secure, resilient and highly available information systems underpinning critical infrastructures. The infrastructure allowed information systems to survive faults, mismanagement and cyber-attack, and automatically adapt to dynamically changing requirements arising from the direct impact from natural events, accidents and malicious attacks. SERSCIS used a service-oriented architecture to make interconnected ICT systems more manageable, allowing dynamic adaptation to manage changing situations, and counter the risk amplification effect of interconnectedness.

3.18.1 System Dependability

To control the resulting services SERSCIS provided tools and ontologies for modelling critical infrastructure, including ICT and non-ICT components, in order to capture their requirements, behaviour and compositional nature. System dependability metrics and agreements, and dynamic governance mechanisms were defined to model the behaviour of systems. System composition mechanisms, allowed for the dynamic discovery and interconnection of component services whilst semantic decision support tools provide situational awareness of system status and threats to autonomic components and human actors.

3.18.2 Layers of Decision Making in Federated Interconnected Systems

The architecture was developed for governance of federated and aggregated infrastructures by independent organisations. The architecture consists of a set of high-level enablers organised into three layers: Application, Management, and Decision Support. The Application Layer provides access to application services, where application is a used as a general term for a broad range of service-based assets such as enterprise applications, service-oriented workflows, computation, storage, networks, and sensors. The Management Layer provides autonomic and predictive management of application resourcing policies through assessment of SLA commitments against available in-house and supplier resources, all expressed as SLA terms. The Decision Support Layer provides operational decision support tools and analytics to service providers using QoS metrics and Key Performance Indicators. Using these tools, providers can model and analyse service configurations and risks, and adapt management policies at runtime to achieve desired performance levels and overall service governance.

3.19 SLA@SOI

Dependable cloud computing through SLAs has been the main objective of the project [88]. The developed open-source SLA@SOI framework addresses the complete service lifecycle through autonomous negotiation, provisioning, monitoring and adaptation of SLAs, while also dealing with the entire service stack, from business aspects through to the physical infrastructure. Driven by four use cases, the project demonstrated the correlation of SLA KPIs with business objectives measurable by business metrics.

3.19.1 Service Description

A *model, namely SLA(T),* for the description of both functional and non-functional characteristics of a service has been developed by SLA@SOI. The model is based on *vocabularies* (e.g. for QoS metrics or constraints) and implemented as an *abstract syntax* that

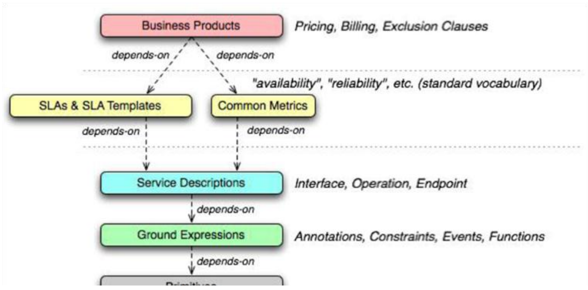

Figure 19: SLA@SOI SLA(T) model

can be instantiated, in whole or in part, by an appropriate concrete syntactic format (e.g. XML, OWL, or human-readable formats), thus being *language and technology independent*. The developed model follows a *hierarchical approach*, being applicable to SLA templates (forming a generic customisable base) and SLAs (having the same basic structure but being non-customisable).

3.19.2 SLA Negotiation across Multiple Layers

The aim of the developed SLA negotiations framework is to enable *negotiations across multiple tiers: business, software, and infrastructure*. To this end, SLA@SOI implemented a framework that enables *different protocols to be injected* so as to facilitate the interaction between the different layers and entities. The framework consists of a *domain-agnostic protocol engine* and a negotiation protocol. The engine executes the negotiation protocol, providing stateful interaction between the customer and the provider. The negotiation protocol enables the implementation of *custom interaction behaviours* and has been encoded as declarative styled rules in order to make it maintainable, readable and machine interpretable. Since the protocol will be used for specific interaction, it may include domain specific content. The SLA model is described in [89].

3.19.3 Scalable SLA-driven Monitoring

As a fundamental mechanism for the provision of QoS guarantees (i.e. SLA enforcement), which may also require service (re)provisioning during runtime, SLA@SOI has developed a *three-layered dynamically configurable monitoring framework*. The upper layer manages the overall monitoring operation by *identifying the monitorable metrics* of the SLAs (i.e. what can be monitored), *selecting monitoring components as sensing elements*, and configuring them in order to identify the optimum sources of events and monitoring data. The middle layer (namely low-level monitoring layer) performs *translation of high level SLAs to operational monitoring specifications* acceptable by specific reasoners (aka monitors), passes operational monitoring specifications to reasoners and receives data from them, while it also maintains the monitoring data. The lower layer (namely sensing and adjustment layer) *captures the events through reasoners* – may be either intrusive (i.e. instrumented into services) or non-intrusive (i.e. run in parallel with the system checking if the events captured from it satisfy the SLA). Reasoners are also able to “understand” the terms included in the SLAs and implement monitoring rules based on abstract syntax trees. The key project results including the SLA Architecture and SLA model are summarised in [90].

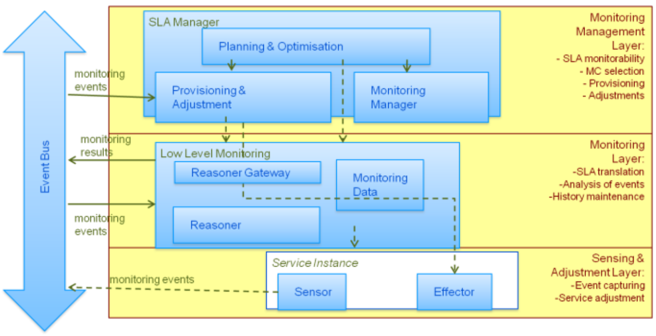

Figure 20: SLA@SOI monitoring architecture

3.19.4 *Interoperability through Open Standards*

Early in the project, SLA@SOI engaged with FP7 RESERVOIR [91] to examine in detail the interactions between service and infrastructure clouds. Considering choice and interoperability across IaaS providers, an immediate observation was the need for, and lack of, an open interface specification to expose relevant details of infrastructure offerings. Service differentiation helps cloud service providers to present options that align with the needs and obligations of service consumers. These needs and obligations can be automatically referenced in an SLA negotiation. A cross-project group initiated and drove the Open Cloud Computing Interface (OCCI) Working Group [92], supported by the Open Grid Foundation (OGF) [93]. While initially proposed for a remote management API for IaaS and PaaS based services, OCCI now presents a protocol and API for Management of Cloud Service Resources. OCCI has upwards of 15 implementations and has been endorsed by NIST [94], SIENA [95], the UK Cabinet Office [96], the German Federal Ministry of Economics and Technology [97] among others.

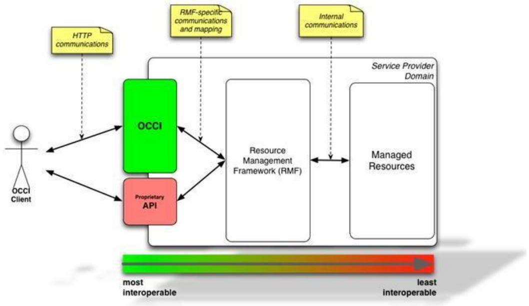

Figure 21: Positioning OCCI

3.20 Stream

Stream [98] architected and developed a system able to process data / event streams in a distributed fashion. By enabling query parallelization and scalability of query operators, thousands of cores can be aggregated to correlate and aggregate millions of events per second.

3.20.1 *Scalable and Efficient Monitoring*

One of the issues with SLA monitoring in large systems (e.g. large data centers) is scalability, given that the amount of monitoring data and the processing needs follow an exponential growth. Stream has developed an approach to *parallelize queries for continuous data streams*, such as monitoring data streams, enabling its scalability to large data stream volumes [99]. The latter is of major importance for cloud environments, since reports during runtime in large scale deployments require the collection and analysis of big amounts of monitoring data. The developed mechanisms are *applicable to cloud monitoring frameworks* since the non-intrusive elasticity will enables them to adapt based on the incoming load [100].

3.21 VISION Cloud

The goal of VISION Cloud [101] is to introduce a powerful ICT infrastructure for reliable and effective delivery of data-intensive storage services, facilitating the convergence of ICT, media and telecommunications. This infrastructure will support the setup and deployment of data and storage services on demand, at competitive costs, across disparate administrative domains, while providing QoS and security guarantees.

3.21.1 *Content-related Terms in SLAs*

The proposed SLA specifications / schemas developed in VISION Cloud project are enriched versions of the traditional ones, since apart from including typical SLA terms they also include *content terms of the data objects* that will be associated with this SLA [102]. The latter enables clouds to provide the users with content-centric services. The content is linked with *performance estimates*, *decisions for moving computation close to storage*, *pricing models* etc, thus allowing for data intensive services of high performance (e.g. quicker search and retrieval of the objects or high performance video streaming speed). Some examples of content terms are telecommunication, media, healthcare, enterprise. Hierarchy of content terms exists. For instance an article for daily news inherits the content term media. Furthermore, specific actions can be executed depending on the SLA content related term, such as storage at specific data centers, execution of compression or format transformation of an object.

3.21.2 Proactive SLA Violation Detection

VISION Cloud has developed an efficient and scalable monitoring framework that is *adaptable to the number of clusters and the nodes per cluster* [103], [104]. It follows a *hierarchical architecture* in order to aggregate monitoring information both at cloud and at cluster levels. The information is being propagated to an event management component that generates events in order to detect and handle error conditions or performance degradations and trigger corrective actions.

What is more, the aforementioned monitoring and event management framework focuses on *proactive SLA violation detection* through the enhanced analysis of monitoring data. This analysis aims at the identification of potential *relationships between the different metrics* being monitored in order to conclude to dependencies that may affect the evolution of the metrics during runtime. Moreover, the analysis aims at *discovering repetitive patterns in the monitoring information* that may provide indications with respect to SLA violations based on the historical data and their evolution in time.

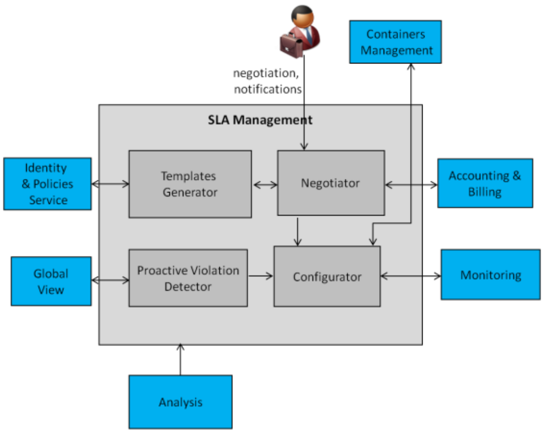

Figure 22: VISION Cloud SLA management

3.22 Additional European projects

This section provides an overview of the research outcomes of various projects, namely plugIT, PHOSPHORUS, SmartLM, CoreGRID and BREIN. Although these projects have not focused their research on clouds, they have delivered specific outcomes that are relevant to SLA topics discussed in this report. These outcomes are briefly described in the following sections.

3.22.1 Term Languages

Both SmartLM and PHOSPHORUS projects have developed *term languages for WS-Agreement* used during the SLA negotiation process. PHOSPHORUS has developed a language for advance reservation of optical network links, which emphasizes on how quality (through the corresponding parameters) can be guaranteed through reservation of resources following a successful negotiation process.

SmartLM has introduced the concept of *software licenses in SLAs* and thus the developed term language enriches existing ones through terms associated with the software license usage during the provision of a service.

3.22.2 Recommendation System

An SLA recommendation system has been developed by plugIT in order to provide an *ordered list of SLA templates* that fulfil the users' requirements. Service (in terms of the so called "project") descriptions are provided by an application owner / businessman while an IT provider proposes IT infrastructure and SLA descriptions including a set of terms. The recommendation system explores semantic annotations SLAs in order to rank the SLA templates and thus provide decision support. Furthermore, it also considers the heterogeneity of resources during the mapping process.

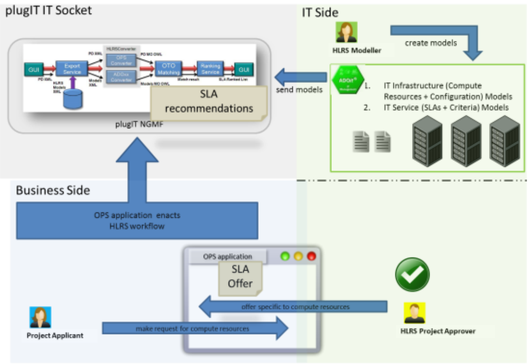

Figure 23: plugIT Recommendation system

3.22.3 SLA Negotiation with WS-Agreement

CoreGRID, an EU-funded project that concluded in 2008, has delivered outcomes in the area of Grid computing with respect to knowledge and data management, programming models, resource management and scheduling, monitoring services and architectural approaches for scalability, dependability and adaptability.

Regarding SLAs, the project has contributed to the *finalisation of the WS-Agreement standard* in the Open Grid Forum (OGF) and the initial discussions on an extended negotiation capability.

3.22.4 Semantic Annotation in SLA templates

BREIN, an EU-funded project that concluded in 2010, has delivered approaches to enable business participants to easily and effectively use Grid technologies for their respective business needs. A business-centric model has been used as a basis to extend "dynamic virtual organisations" and enhance Grid environments with methods from artificial intelligence, intelligent systems, semantic web etc.

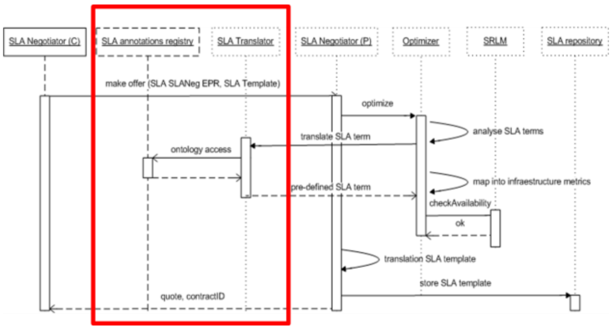

Figure 24: BREIN semantic annotation in SLAs

Thus, BREIN proposed a specification for semantic annotations in SLA files called *Semantic Annotations for Service Level Agreement (SA-SLA)* [105], [106], which is based on the Semantic Annotations for Web Service Description Language (SA-WSDL). SA-SLA provided a standard description format extending the current WS-Agreement specification with semantic annotations in order to provide to WS-Agreement and WSLA elements the domain vocabulary it lacks.

3.23 National projects

This section provides an overview of the SLA-related research outcomes of various national projects, namely VIOLA, DGSI, SLA4D-Grid, and MisuraInternet.

3.23.1 WS-Agreement for Advance Reservation of Network Resources

VIOLA is a German project that similar to PHOSPHORUS on a European level aims at delivering the necessary technology to provide bandwidth on demand in the German Research Network. Along with bandwidth on demand co-allocation of network and computational resources to support distributed computing and visualisation of the results have been developed by the project. Both *co-allocation of resources and bandwidth on demand* are negotiated using WS-Agreement.

3.23.2 Term Language for Resources Advance Reservation

DGSI (D-Grid Scheduler Interoperability) is a German project that aims at providing interoperability for the different Grid-schedulers of the D-Grid communities to support job and resource delegation across the resources maintained and managed by the different communities. The interoperability was achieved through WS-Agreement and WS-Agreement implementations in all D-Grid Grid level schedulers.

One of the main outcomes of the project is a *term language (integration of the JSDL specification of OGF*) to describe computational resources. The language allows for both specifying requirements of a requesting scheduler and offering of providing schedulers, which can be accessed through a registry. DGSI supports activity delegation (job submission with defined QoS to other D-Grid sites) and temporary delegation of resources from a providing scheduler to a requesting scheduler. The latter is a cloud-like approach to rent external resources for a defined time with defined QoS [107].

3.23.3 SLA Negotiation

Within the frame of the SLA4D-Grid project an SLA infrastructure was developed being usable for projects being part of the D-Grid (German Grid). Hence, partners were integrated into the SLA4D-Grid developments and further, the development of the architecture was iteratively performed based on feedback from the D-Grid community. In addition, to ensure the efficiency of the developed solution a monitoring approach for performing a continuously monitoring of the compliance with the negotiated SLAs was implemented. In summary, the outcome of the SLA4D-Grid project was an implemented SLA Management System and a monitoring solution, both applicable for the D-Grid community. Regarding SLAs, the project has developed an *optimally adapted SLA layer* of the SLA4D-Grid service negotiation and orchestration [108]. The project also contributed to the WS-Agreement Negotiation specification of the Open Grid Forum and provided an implementation for D-Grid through an adapted version of WSAG4J, a Java implementation of WS-Agreement and WS-Agreement Negotiation. For evaluating the SLAs regarding fulfilment or violation the SLA layer relies on D-Mon the monitoring infrastructure of D-Grid.

3.23.4 Network QoS Monitoring

MisuraInternet is an Italian project that aims at collecting *QoS monitoring information on the network level* and analysing the potential impact on SLA control.

Regarding SLAs, the project has developed a mechanism for ISP measurements through specific metrics (as identified in the ETSI Guide - EG 202 057-4): data transmission speed (achieved separately for downloading and uploading specific test files), packet delay, packet loss ratio and unsuccessful data transmission ratio.

3.24 Conclusions – Mapping to the SLA Metamodel

An overview of the research outcomes of the European and National projects with respect to different phases of the SLA lifecycle is cited in Figure 25.

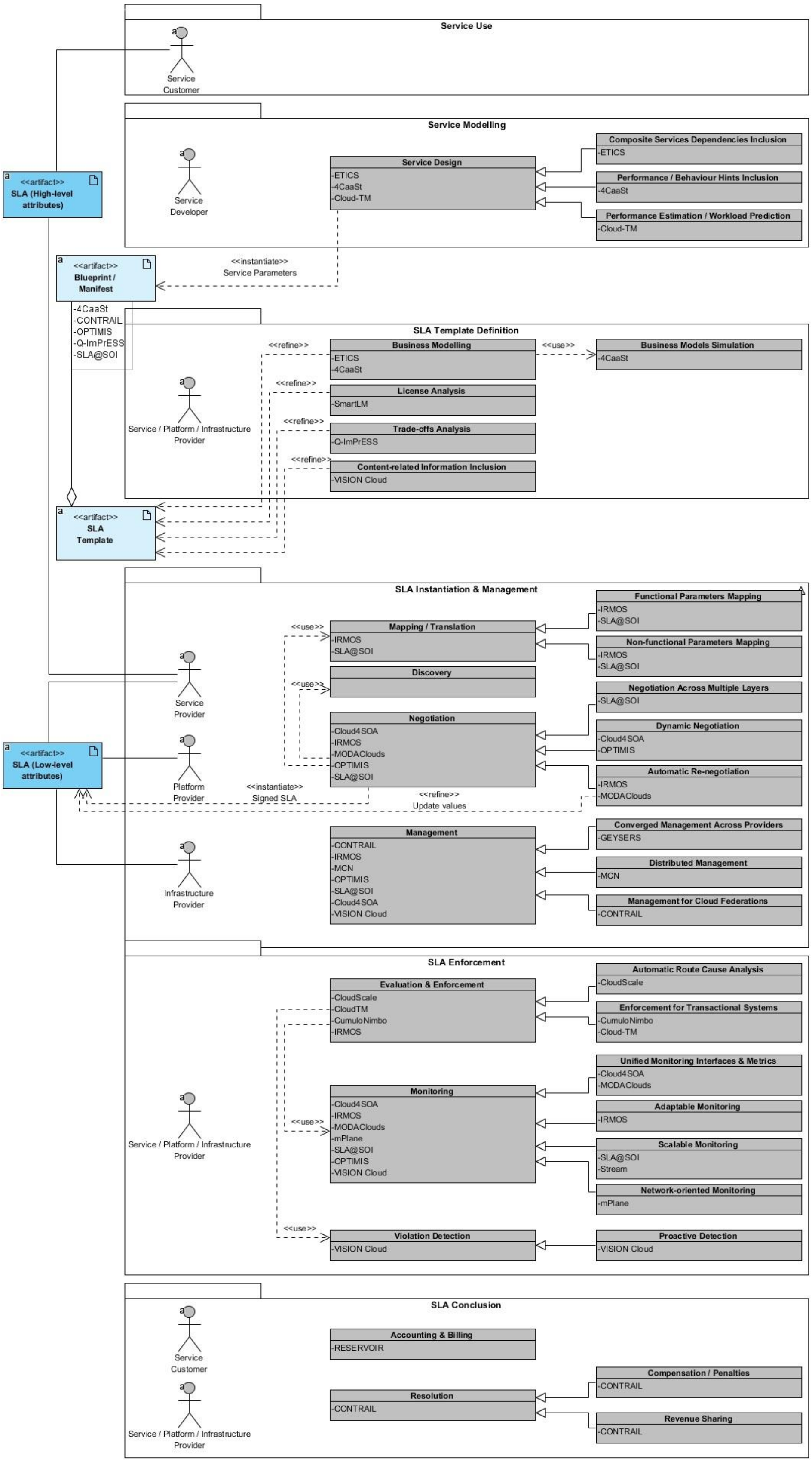

Figure 25: Projects contributions mapped to the SLA metamodel

4 *Recommendations*

The goal of this section is to provide a set of recommendations to the on-going policy work on SLAs of the Cloud Select Industry Group (SIG). Recommendations do not aim at identifying new potential research fields or shortcomings of existing approaches, but focus on the exploitation of the SLA-research outcomes stemming from European and National research projects. To this end, each recommendation includes references to the corresponding sections of this report that shortly describe the related research outcomes of the respective projects.

4.1 Background and Considerations

The European Commission's strategy “Unleashing the potential of cloud computing in Europe” has highlighted a set of key actions, one of which refers to the development of model “safe and fair” contract terms and conditions. To this end, DG ConNECT has launched specific working groups for the implementation of the overall strategy and the identified key actions. A working group (namely “*Cloud Select Industry Group on Service Level Agreements*”) includes representatives from industry and cloud computing stakeholders, aiming to support the EC actions and on-going policy work on cloud SLAs, and provide positioning input to the *European Cloud Partnership*. The initial activities of the working group focused on the compilation of a preliminary list of attributes that should be included in SLAs, raised the need for definitions, classification and descriptors of different metrics, parameters and KPIs, as well as for efficient SLA monitoring approaches. Furthermore, the recommendations included in the current report and specifically the recommendation focusing on standards may be considered by the ETSI CSC (Cloud Standards Coordination) working group, which is an initiative aiming to define what type of standards are required to ensure smooth deployment of Cloud technologies in the European Union, with an emphasis on security and privacy, interoperability, data portability, and SLAs.

While this section of the report focuses on recommendations towards the aforementioned working group, some considerations need to be taken into account by the working group members. These mainly refer to: (i) the prioritization of the recommendations, and (ii) the exploitation of the research project outcomes. With respect to prioritization, the report presents the recommendations in a sorted prioritized order. However, prioritization may also be performed based on the desired outcome (Section 4.3 proposes also an outcome-based classification of the recommendations). Regarding exploitation of the research project outcomes, there has to be noted that in some cases (e.g. SLA monitoring) more than one projects have developed similar approaches. Depending on their focus there are pros and cons which have not been evaluated in the framework of this report. Nevertheless, the common ground of these approaches can be considered as a baseline.

4.2 Recommendations

This section provides a set of recommendations (in a tabular format) addressing different areas in the SLA lifecycle. For each recommendation, a brief description is provided along with the main goal of the recommendation and potential variations. Proposed steps aim at providing a path for the implementation of the recommendation, while the potential contributions section highlights research project outcomes that can be exploited towards the recommendation implementation (links to the specific sections that detail each outcome are embedded in the tables).

4.2.1 Develop a Core SLA Specification and Differentiate SLAs and Contracts

Develop a core SLA specification and differentiate SLAs and contracts	
Recommendation - R1	Clearly separate domains and characteristics of contracts and SLAs by developing one core SLA specification that includes basic terms as core elements, and which meets the following criteria: 1. The terms are *common* for the offered services and independent from the provider 2. The *meaning* of the terms is concise and clear for the users. Terms should be *objective* (not open to more than one interpretations) and *attainable* (terms beyond the control of either party should not be included) 3. The vocabulary allows for the expression of the terms in a *precise* and *well-defined* way, reflecting a specific service quality definition and related actions (e.g. scalability) 4. The vocabulary allows for the *classification of the terms* and the KPIs into main classes (e.g. unobservable, observable, enforceable, mandatory or optional, numeric, %, etc) 5. *Logical expressions* description should also be feasible to enable dynamic negotiation of quality attribute trade-offs 6. Besides functional, *non-functional attributes* should be defined in SLAs, since they may influence the successful establishment of a relationship and the complete SLA lifecycle 7. The specification is captured through a *structured representation* (e.g. in XML format) 8. The specification is *easily extendable* to integrate new concepts and requirements
Goal	Overcome the great variability in the SLA terms and provide the basis for SLA management, reporting and enforcement. A core specification should allow for the identification of expectations and the establishment of performance indicators.
Variations / comments	The core SLA specification can be *extended* (not altered) with additional terms for *specific domains* (e.g. telecommunication, healthcare, media) or *application areas* (e.g. video streaming, transactional systems, content syndications). The additional terms should also be specific for each domain or application

Develop a core SLA specification and differentiate SLAs and contracts	
	area (e.g. for specific resources it could be OVF-based).
Proposed steps	1. Classify *services into main categories* (e.g. storage, processing) 2. Analyse the *service offerings* from different providers for the aforementioned categories in order to conclude to the *attributes per category* 3. Identify the *common set* of terms as well as the *additional* terms (i.e. domain- or application- specific terms) 4. Develop *concrete descriptions* for each term and link it with specific metrics / KPIs to clarify the objective of the term 5. Provide a *structured specification*
Potential contributions	✓ 4CaaSt Project (Section 3.1.1): Blueprint Concept ✓ BREIN Project (Section 3.22.4): Semantic Annotation in SLA templates ✓ CloudScale (Section 3.3.1): Scalability Specification ✓ Cloud-TM Project (Section 3.4.2): SLA Definition and Enforcement in Transactional Data Stores ✓ CONTRAIL Project (Sections 3.5.1 and 3.5.2): SLA Specification and Quality Model ✓ EGI Project (Section 3.7.1): Service Catalogue in a Federated Environment ✓ IRMOS Project (Section 3.11.1): SLAs at Different Levels ✓ OPTIMIS Project (Section 3.15.1): Service Manifest ✓ Q-ImPrESS Project (Section 3.17.1): QoS-oriented SLA Specification ✓ PrestoPRIME Project (Section 3.16.1): SLA Specification for Preservation Services (risk of data loss) ✓ SLA@SOI Project (Section 3.19.1): Service Description ✓ VISION Cloud (Section 3.21.1): Content-related Terms in SLAs

4.2.2 Support Composite and Complex Services

Support composite and complex services	
Recommendation - R2	Composite services provision in cloud environments requires SLA support in the following areas: 1. *SLA specifications* capturing the dependencies and interactions between the services. The dependencies should be parametric and express the overall service context (e.g. data movements, relationships between providers, orchestration rules) 2. Convergence in *SLA management* to handle dependencies (i.e. joint management) while retaining the autonomy in resource management for each provider
Goal	Provide suitable SLAs for composite services in a *multi-provider environment* (including the case of *cloud federations*), since services are increasingly becoming composite, consisting of atomic services that may either be offered *within a cloud layer* (e.g. object and block storage service) or *across cloud layers* (e.g. monitoring service from a third party provider and storage from a cloud provider).
Variations / comments	Enhanced SLA specification and management approaches should take into consideration that *composition* may be

Support composite and complex services	
	performed either *centralized* (i.e. an entity managing the composition and the corresponding service offerings) or *distributed* (i.e. achieved through consecutive SLA establishments). SLA specifications in *cross-domain scenarios* should either include the *common terms* (limiting however end-to-end quality provision to these terms) or be implemented through *links between SLAs* (i.e. one SLA for each domain with enriched specification to include links to the SLAs of other domains), as a protocol to enable interaction between different layers and entities.
Proposed steps	1. Extend *core SLA specification* (Recommendation R1) to include links to other SLAs 2. Adopt existing or extend *SLA management mechanisms* (through interfaces) to handle dependencies of composite services
Potential contributions	✓ 4CaaST Project (Section 3.1.1): Blueprint Concept ✓ CONTRAIL Project (Section 3.5.2 and 3.5.3): Quality Model and Multi-level SLA Interaction Model ✓ ETICS Project (Section 3.8.1): SLAs for Composite Services ✓ OPTIMIS Project (Section 3.15.1): Service Manifest ✓ MCN Project (Section 3.12.1): Distributed SLA Management ✓ Q-ImPrESS Project (Section 3.17.1): QoS-oriented SLA Specification

4.2.3 *Encapsulate Legal Terms and Separate Responsibilities and Obligations*

Encapsulate legal terms and separate responsibilities and obligations	
Recommendation - R3	Introduce legal terms in the SLAs and identify in a clear and precise way the responsibilities and obligations of all involved entities, as well as their boundaries and limits. To this end, SLA specifications should: 1. Cover *legal aspects*, especially with respect to the complete *data lifecycle* in cloud environments (i.e. ingest / collection, storage, processing, replication, distribution, removal) 2. Capture terms, responsibilities and obligations through a *legally valid SLA vocabulary* that will include specific attributes (e.g. Quality of Protection - QoP) 3. Capture *exclusion terms* besides clauses (e.g. violation penalty amount or time period of claims) 4. Allow for the definition of terms (e.g. data location) that can be *observable and enforceable* (e.g. through data placement mechanisms) to guarantee legal conformance 5. Clearly *define Intellectual Property Rights (IPR)* of created information and according ownership 6. Enable the *standardization of IPR categories* so that automated SLA negotiation may include them
Goal	Effective SLA that includes legal terms and *acknowledges the responsibilities and obligations of all participating entities*, i.e. both providers and customers to avoid potential disputes. The latter will allow service providers to minimize the customers' concerns regarding the service delivery and quality, while

Encapsulate legal terms and separate responsibilities and obligations	
	managing expectations by taking into account the customers responsibilities (e.g. reasonable notice of planned changes or requirements).
Variations / comments	Legal aspects, responsibilities and obligations are fundamental in *multi-cloud environments*, *federations* or *composite services* provision. Boundaries and limits should be clearly defined to minimize potential transfer of liability.
Proposed steps	1. Identify *legal terms* that can be included in an SLA 2. Identify *processes* (e.g. data placement or replication) that can be affected by the legal constraints 3. Extend core *SLA specification* (Recommendation R1) to include legal terms 4. Adopt existing or extend *SLA management mechanisms* (through interfaces) to monitor and enforce legal requirements
Potential contributions	✓ CONTRAIL Project (Section 3.5.2): Quality Model ✓ OPTIMIS Project (Section 3.15.1): Service Manifest

4.2.4 Provide Accurate Runtime Monitoring and Reporting

Provide accurate runtime monitoring and reporting	
Recommendation - R4	Aggregate and publish monitoring information to customers taking into consideration that: 1. The *required format* should be on the level of service attributes, thus capturing both application-related high-level monitoring information and low-level resource data 2. *On-time delivery* is of major importance for cloud environments that aim at facilitating real-time and interactive applications 3. The *responsibility* of providing accurate monitoring information should be either on the service provider side or on a Trusted Third Party (TTP) 4. *Accurate and trustable* reports are required since auditing is based on monitoring data 5. *Unified metrics* across providers would ease the aggregation of monitoring data and contribute towards runtime reporting (mapping or translation would not be needed) 6. The *latency* of the monitoring mechanisms and the *footprint* on the infrastructure and the application should not affect the runtime aspects.
Goal	Deliver monitoring information with respect to service / application and resource *usage and delivery*, as well as reports for the *SLA terms*, potential *violations* and *actions* taken (e.g. increase of resources to meet a specific application requirement) or foreseen (e.g. SLA violation and payment of penalty).
Variations / comments	*Monitoring configuration* is critical since the latency and the associated overhead may be reflected to the service delivery. Configuration refers to monitoring deployments (e.g. monitoring agents in each VM to obtain application-specific information) and / or monitoring time intervals (adaptable based on the collected monitoring information). The great *amount of monitoring data* in large deployments may

Provide accurate runtime monitoring and reporting	
	cause inefficient analysis. Scalable and elastic approaches should be considered.
Proposed steps	1. Identify *monitorable / observable attributes* and common metrics between providers 2. Enhance providers' *monitoring mechanisms* to provide the required information or provide interfaces to TTPs for delivering monitoring services 3. Propose *adaptable monitoring frameworks* and *elastic approaches* for obtaining and aggregating the monitoring data
Potential contributions	✓ Cloud4SOA (Section 3.2.1): Unified Monitoring Interface and Metrics ✓ IRMOS Project (Section 3.11.3): Adaptable Monitoring and Evaluation ✓ MODAClouds Project (Section 3.13.1): Unified Monitoring ✓ mPlane Project (Section 3.14.1): Network Monitoring for SLAs ✓ SLA@SOI Project (Section 3.19.3): Scalable SLA-driven Monitoring ✓ Stream Project (Section 3.20.1): Scalable and Efficient Monitoring

4.2.5 Support Runtime Adaptability and Dynamic SLA (Re-)Negotiation

Support runtime adaptability and dynamic SLA (re-)negotiation	
Recommendation - R5	Service and infrastructure providers should support runtime adaptability, which is reflected to the following: 1. *SLA specifications* should allow for the expression of ranges in various terms (associated with the corresponding costs) 2. SLAs should be able to evolve (e.g. reflecting on-demand resource provisioning) *during the service delivery / application execution* based on the monitoring information and the evaluation process that may trigger corrective actions 3. Evolvement of SLAs (i.e. values of attributes) should be feasible not only *within a cloud layer* but also in *cross-layer* scenarios 4. *SLA (re-)negotiation should be transparent* to the customer regarding service delivery
Goal	*Guarantee quality of service* at runtime and support the elasticity and scalability features of cloud environments.
Variations / comments	In the case of *multi-provider environments* (e.g. cloud federations), service providers should deploy SLA management mechanisms supporting automated SLA (re-negotiation) at runtime. Quality can only be guaranteed if the additional resources / services (that may be utilized in the case of a re-negotiation) specified in the initial SLAs are *reserved in advance*, otherwise they may be utilized when requested.
Proposed steps	1. Ensure that terms in SLA specification can be associated with *ranging / floating* values 2. Propose *automatic re-negotiation mechanisms* for SLA management frameworks

Support runtime adaptability and dynamic SLA (re-)negotiation	
Potential contributions	✓ Cloud4SOA (Section 3.2.2): Dynamic SLA Negotiation and Enforcement ✓ IRMOS Project (Section 3.11.2): Dynamic SLA Re-negotiation ✓ MODAClouds Project (Section 3.13.2): Runtime Re-negotiation ✓ OPTIMIS Project (Section 3.15.2): Automated SLA Negotiation ✓ SLA@SOI Project (Section 3.19.2): SLA Negotiation across Multiple Layers

4.2.6 Certify Providers and Enhance SLA Enforcement for Mission-critical Applications

Certify providers and enhance SLA enforcement for mission-critical applications	
Recommendation - R6	Service providers *liability* should be *certified for specific properties* / attributes, thus allowing their exploitation for mission-critical or legally-demanding applications that pose explicit requirements. Service providers should also enhance SLA enforcement with the following key aspects: 1. *Proactive SLA violation detection* based on workload prediction and performance forecasting 2. *SLA violation avoidance* by independent certification of SLA compliance capabilities of providers 3. *Automatic root cause analysis* through models for parameters analysis and evaluation
Goal	Consider SLAs as the means for providers to establish their credibility, attract or retain customers since they will be used as a mechanism for service differentiation. Allow service providers to detect violations proactively (not reactively based on monitoring data) and thus enforce SLAs, while performing root cause analysis to minimize potential future violations. Introduce providers certification will enable *SLA-based risk estimation and assessment* and thus allow for the execution of mission-critical applications. Automate current offline bureaucratic processes for provider certification (e.g. with regard to Binding Corporate Rules compliance to EU data management regulatory framework).
Variations / comments	Certification may be performed by a third party in the role of an "*insurance company*". However, in that case an agreement (potentially an SLA) should be signed between the third party and the service provider. The SLA may either follow the cloud SLAs schemas and be managed by the frameworks or (most likely) refer to an "offline" contract.
Proposed steps	1. Identify an entity that will act as the *certification authority* 2. Identify *properties per provider that can be certified* and provide certifications 3. Include *certifications* as additional information in service registries and SLA repositories (thus customers can search based on these criteria) 4. Propose *proactive SLA detection and automatic root cause*

Certify providers and enhance SLA enforcement for mission-critical applications	
	analysis mechanisms 5. Identify *concrete technical steps* that should be performed by government agencies *to automate the process*.
Potential contributions	✓ 4CaaSt Project (Section 3.1.3): Elasticity Management ✓ CloudScale (Section 3.3.2): Automatic Root Cause Analysis ✓ Cloud-TM Project (Section 3.4.1 and 3.4.2): Performance Estimation and Workload Prediction and SLA Definition and Enforcement in Transactional Data Stores ✓ CumuloNimbo Project (Section 3.6.1): SLA Enforcement for Transactional Systems ✓ Q-ImPreSS Project (Section 3.17.2): Trade-off Analysis and SLA Prediction ✓ VISION Cloud Project (Section 3.21.2): Proactive SLA Violation Detection

4.2.7 Consider Business Models and Objectives

Consider business models and objectives	
Recommendation - R7	Enable *dynamic provision of SLA templates* following the goal of the providers to cultivate business value based on existing and past offerings, user requirements and market conditions. Dynamicity in SLA templates refers to the *attributes' values* that can either be *altered* or expressed in a *layered* format (i.e. multi-layered offers). Multi-layered offers aim at high-utilization of resources by exploiting current resource utilization information as well as historical information regarding the customers (i.e. "loyalties" programmes), or on trade-offs between parameters such as eco-efficiency, cost and risk.
Goal	Allow business decision making to be reflected to SLAs in a dynamic and automated way. *High-level business objectives* (e.g. different pricing models or rewards) and criteria are mapped through business simulation frameworks to low-level resource parameters, thus encompassing the business logic in the SLAs.
Variations / comments	Business resolution and revenue sharing in *multi-provider environments* can be addressed independently from the specific recommendation.
Proposed steps	1. Identify *business terms* that may consist as input to mapping / business simulation frameworks 2. Identify *pricing models* that can be linked with specific business terms and service offerings 3. Propose *frameworks* exploiting information regarding business terms and pricing models to provide recommendations regarding the attributed in SLA templates
Potential contributions	✓ 4CaaSt Project (Section 3.1.2): eMarketplace ✓ EGI Project (Section 3.7.2): Federated Service Management ✓ ETICS Project (Section 3.8.2): Business-enhanced SLA Template ✓ OPTIMIS Project (Section 3.15.1): Service Manifest ✓ plugIT Project (Section 3.22.2): Recommendation System ✓ SLA@SOI Project (Section 3.19.1): Service Description

4.2.8 Invest in User-oriented SLAs

Invest in user-oriented SLAs	
Recommendation - R8	Users either as customers or even as cloud providers (in multi-providers environments) should be treated as first-class citizens in SLAs. Therefore: 1. *Outcome-based SLA specifications* should also be developed. These could be SLA specifications embracing other SLAs; however their main deference is that they capture in a single statement the service outcome and hide all details related to the application parameters and the low-level infrastructure details 2. User-oriented and experience-oriented SLAs should include *clear criteria for the success and the failure* with respect to the delivery of the aforementioned outcome 3. *Simplicity* should be the main goal of such SLAs.
Goal	Understand the needs of customers and providers and capture these needs with *simple*, *clear* SLAs that focus on the *outcome*.
Variations / comments	Derive the effective end users' QoE from individual SLAs.
Proposed steps	1. Develop *guidelines for ensuring simplicity* in SLA specifications 2. Propose an *SLA specification* targeting outcome-based SLAs

4.2.9 Adopt a Reference Baseline Solution for SLA Management

Adopt a reference baseline solution for SLA Management	
Recommendation - R9	A *domain agnostic*, *broadly accepted* SLA management framework should be adopted as a basis. The framework should be extendable with *additional components* (e.g. data placement mechanism considering legal aspects) based on the specific needs of the providers or the application domains.
Goal	Minimize *development efforts* since several SLA management frameworks have been developed and evaluated in different cases and application scenarios.
Variations / comments	The reference solution should support potentially different (i.e. domain-specific) *protocols and languages*.
Proposed steps	1. Identify the *core functionalities* of an SLA management framework 2. Identify and evaluate *candidate frameworks* 3. Exploit one as the *core / baseline framework* 4. Propose *additional components per domain* that can be integrated in the baseline framework
Potential contributions	✓ CONTRAIL Project (Section 3.5.4): SLA Management for Cloud Federations ✓ GEYSERS Project (Section 3.9.1): Converged SLA Management for Composed Virtual Infrastructures ✓ MCN Project (Section 3.12.1): Distributed SLA Management

4.2.10 Develop Standards

Develop standards	
Recommendation - R10	Develop *domain agnostic standards* for different elements and parts of the SLA lifecycle. The main identified elements refer to the following (proposed exclusions are based on the maturity level of the listed elements): 1. Core *SLA specification* 2. Extended SLA specification for *composite services* 3. *SLA monitoring* mechanisms (excluding event evaluation and management) 4. *SLA management* frameworks with core functionalities (excluding SLA re-negotiation)
Goal	Minimize development efforts ("*re-inventing the wheel*" cases) by standardising specific outcomes since existing ones (e.g. WS-Agreement) are widely being adopted and used.
Variations / comments	There are on-going efforts on cloud SLA standards at different places, for instance at OGF, TMF, and others [109], [110]. A detailed analysis of the Cloud SLA standards landscape is currently under development by the ETSI Cloud Stands Coordination [111], supported by the EC through the EC Cloud Strategy. The final report is expected in Q3/2013. Projects should target to contribute to existing efforts or the use of existing specifications.
Proposed steps	1. Identify *core elements* that can be standardised (e.g. SLA specification) 2. *Evaluate standard adoption* through its integration in primary cloud application domains (e.g. data analytics)

4.2.11 Introduce an H2020 Initiative to Support the Work of the SLA Research Group

Introduce an H2020 initiative to support the work of the SLA research group	
Recommendation - R11	Support an initiative in the framework of Horizon 2020 that will focus on: 1. Developing and setting up an *SLA Reference Model* 2. *Evaluating research outcomes* addressing specific SLA aspects through *quantitative and qualitative comparison* 3. Concluding on *research outcomes* that can be exploited for the *realization of the SLA Reference Model* 4. Proposing specific outcomes for *standardisation* 5. Developing *recommendations* towards various bodies and stakeholders (e.g. EC, policy groups, cloud providers, standardisation bodies, user groups, etc)
Goal	Support the work of the SLA research group towards the implementation of the current recommendations and future identified ones considering research results, requirements from stakeholders, cloud landscape and emerging standards.
Proposed steps	1. Identify *main contributors* and *driving organisations* for the initiative as well as potential *ad-hoc on-demand contributors for specific topics* 2. Identify *main work items and target outcomes* of the initiative

4.3 Classification of Recommendations

This section provides a classification of the proposed recommendations based on different properties. These refer to:

- ✓ *Envisioned impact*, providing a classification based on the expected impact of the implementation of the recommendation (e.g. wider adoption of cloud solutions, broader service offerings, improved interoperability, minimized overlapping efforts, optimized service deployment and operation, increased competitiveness, enhanced trade-offs between cost and performance, automated certification process, increased market pool of cloud computing to non-technical users, etc).
- ✓ *Target groups*, providing a classification based on the target groups being addressed by each recommendation.
- ✓ *Reference in the SLA lifecycle*, providing a classification based on the processes of the lifecycle that are being linked with each recommendation.

4.3.1 Envisioned Impact

The classification based on the envisioned impact emphasizes on wider aspects and not on specific technical or business aspects. In this context, a classification is proposed based on the envisioned impact in *business, technical and user dimensions*. The goal of the proposed specification is to allow stakeholders to prioritise the recommendations based on the dimension they consider of major importance.

Mapping to these dimensions is depicted in the following figure while a *"quantitative" evaluation* of the envisioned impact is provided in Figure 27. As depicted in the figures, developing standards is amongst the "must do" recommendation that affects all dimensions to a great extent. The core SLA specification is also of major importance, while even though many recommendations appear in the same level of importance (from a "quantitative" point of view), they target different dimensions (for example inclusion of legal terms has a limited technical impact but a greater user-related impact, while certification of providers has also limited technical impact but great business impact).

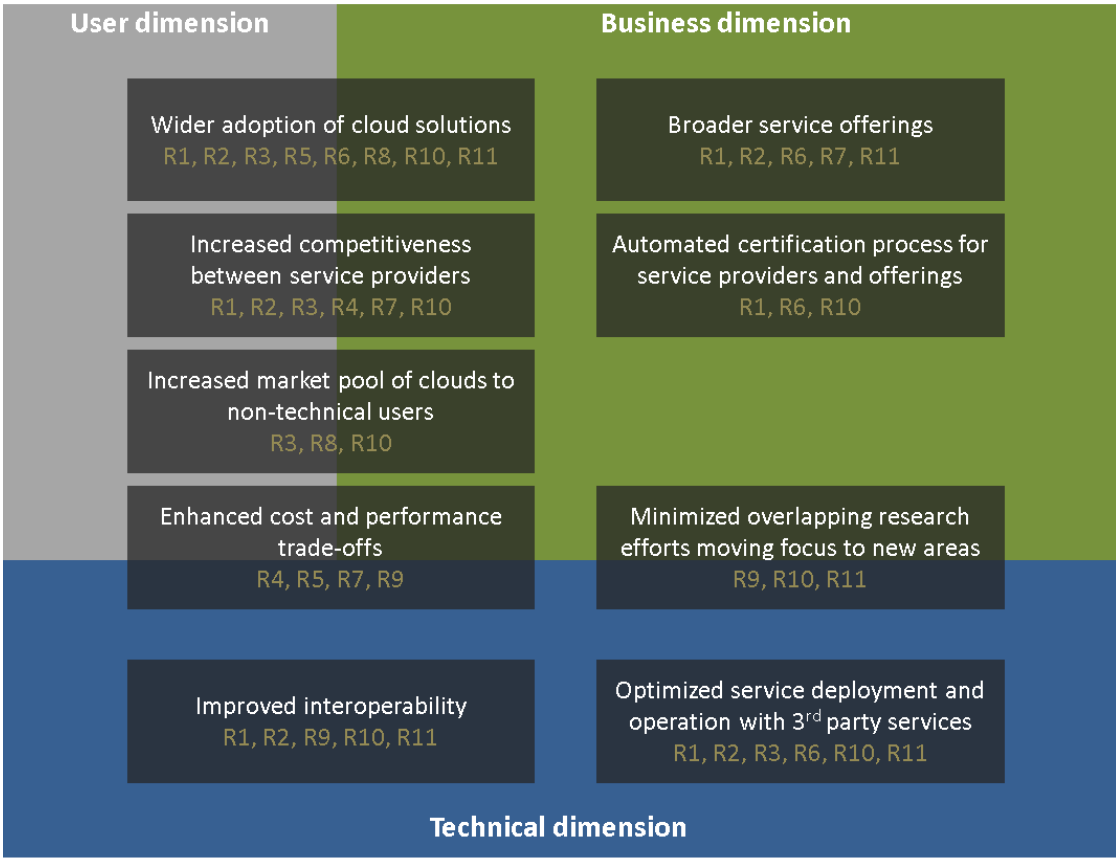

Figure 26: Recommendations across user, business and technical dimensions

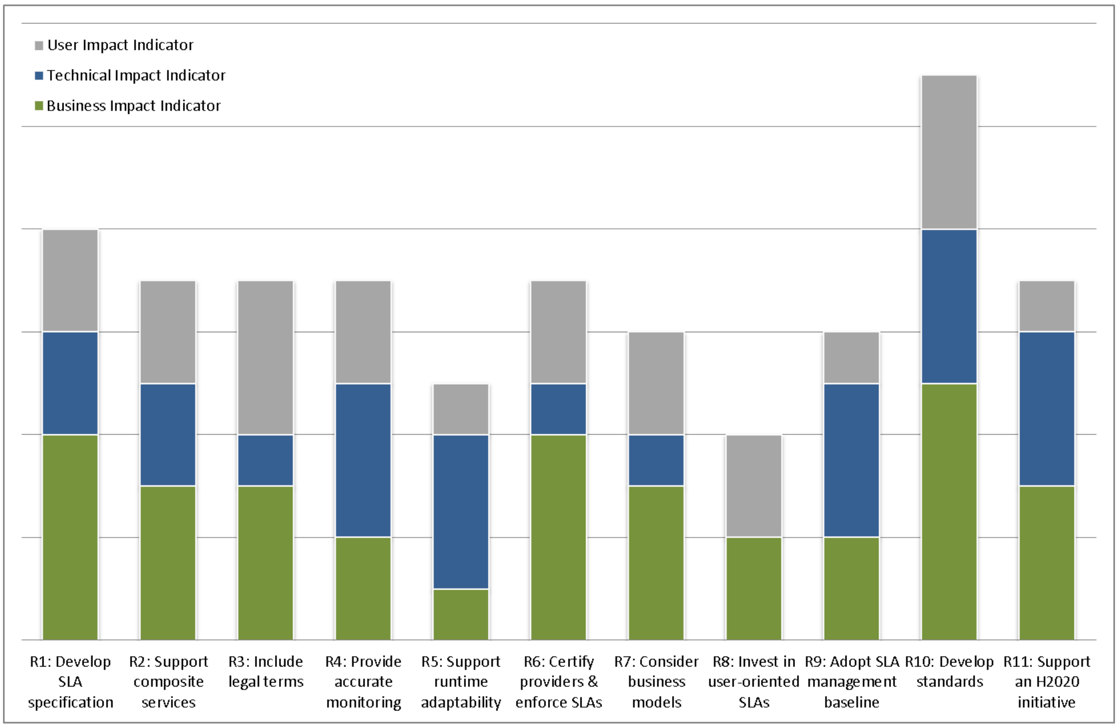

Figure 27: Quantitative evaluation of recommendations (user, business and technical dimensions)

4.3.2 Target Groups

Based on the main groups of users engaged in the SLA lifecycle, the recommendations are classified per target group as depicted in the following figure. As shown in the figure, most of the recommendations target *service customers* and *cloud providers*. The latter is expected given that SLAs have a limited influence in the service design process, and thus towards service developers.

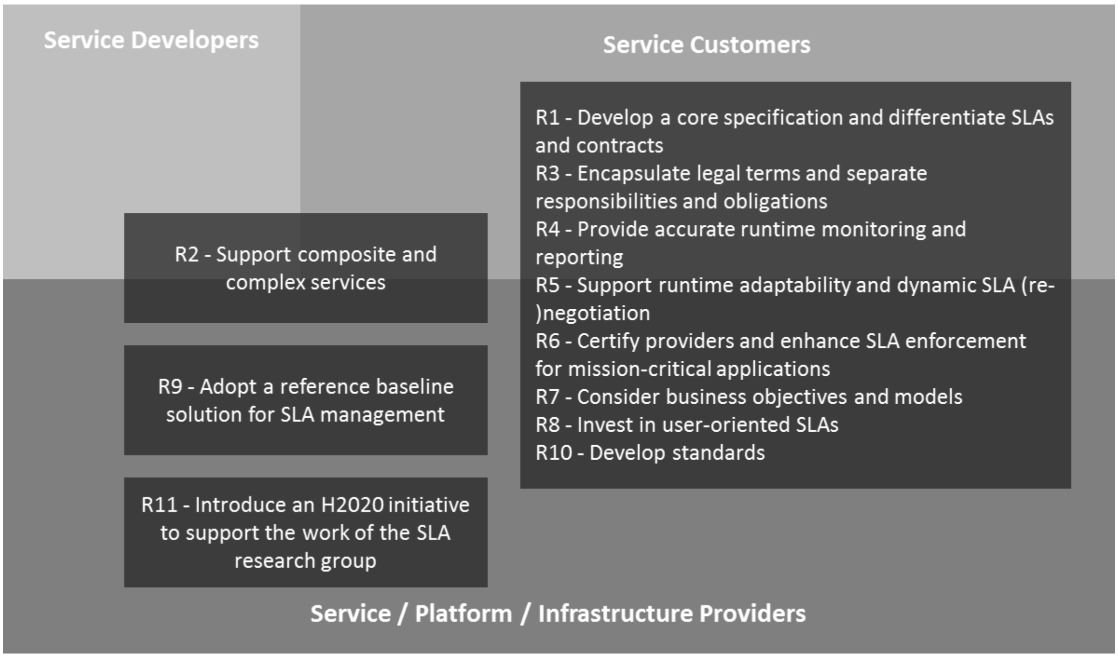

Figure 28: Classification of recommendations for different stakeholders / actors

4.3.3 Reference in the SLA lifecycle

Based on the SLA metamodel (described in Section 2), the recommendations address different phases of the SLA lifecycle as depicted in the following figure. The goal of the proposed specification is to allow stakeholders to prioritise the recommendations based on the SLA lifecycle phase they consider of major importance.

The figure shows that recommendations address different phases, while some recommendations may be considered more valuable comparing to other recommendations, since they target more than one phases.

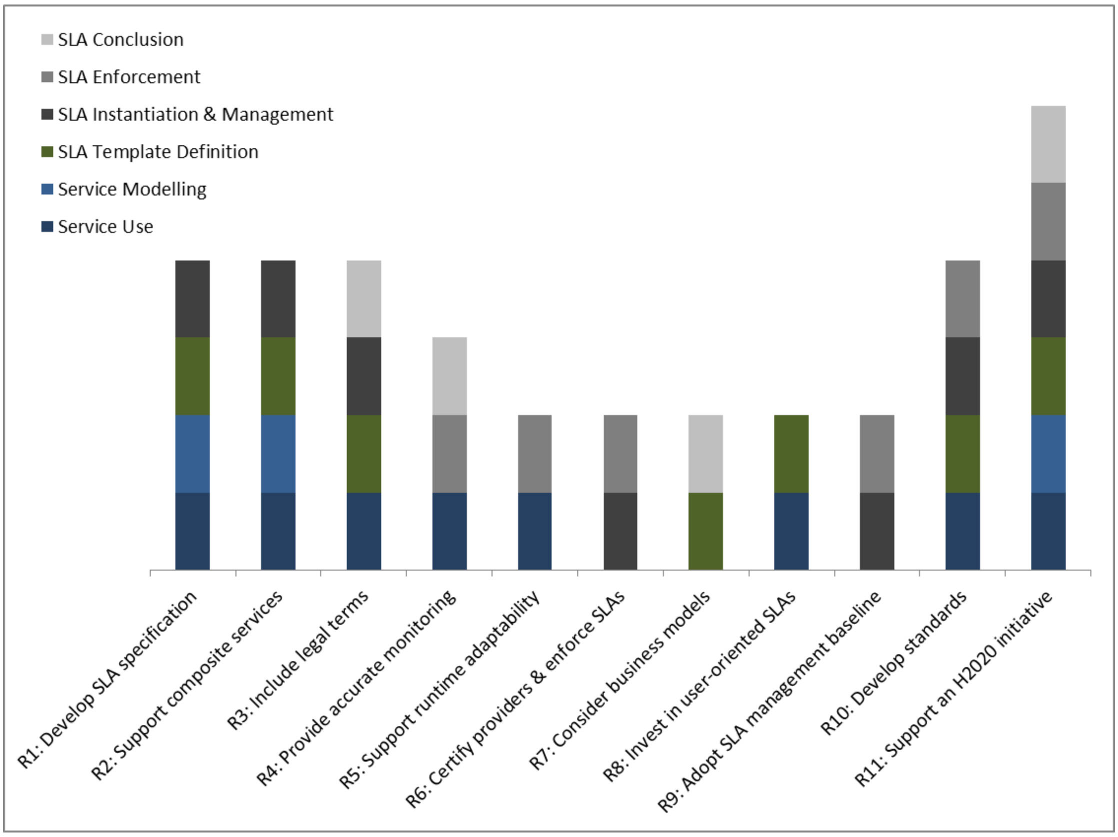

Figure 29: Classification of recommendations across the phases of the SLA lifecycle

5 *Conclusions*

In the cloud ecosystem, a world of multi-stakeholder information and services provisioning, Service Level Agreements are increasingly becoming the key criterion for service selection. Users are now demanding agreements with clear attainable terms, services with guaranteed quality levels, offerings that meet specific legal and protection terms, accurate reporting on the service usage, runtime adaptation for evolving requirements. Conversely, new providers consider SLAs a driving force for entering the cloud market as their certification for the offered services and the means to establish their credibility.

Innovative research outcomes from European projects go beyond what is possible and what is provided today. These outcomes are Europe’s competitive advantage. Therefore, the recommendations provided in this report (not only technological but also covering legal, economic and standardisation areas) can certainly be based on and exploit European research projects’ results as a starting point towards their realization.

Annex 1: Glossary of Acronyms

Acronym	**Definition**
ACID	Atomicity, Consistency, Isolation, Durability
ANN	Artificial Neural Network
ETICS	Economics and Technologies for Inter-Carrier Services
FP	Framework Programme
GEYSERS	Generalized Architecture for Dynamic Infrastructure Services
IaaS	Infrastructure as a Service
IPR	Intellectual Property Rights
IRMOS	Interactive Real-time Multimedia Applications on Service Oriented Infrastructures
ISP	Internet Service Provider
JSDL	Job Submission Description Language
KPI	Key Performance Indicator
MARTE	Modeling and Analysis of Real-time and Embedded Systems
MCN	Mobile Cloud Networking
MODAClouds	Model-driven Approach for Design and Execution of Applications on Multiple Clouds
MOS	Mean Opinion Score
mPlane	An Intelligent Measurement Plane for Future Network and Application Management
NIST	National Institute of Standards and Technology
OCCI	Open Cloud Computing Interface
OGF	Open Grid Forum
OLA	Operational Level Agreement
OPTIMIS	Optimized Infrastructure Services
OVF	Open Virtualization Format
OWL	Web Ontology Language
PaaS	Platform as a Service
Q-ImPrESS	Quality Impact Prediction for Evolving Service-oriented Software
QoE	Quality of Experience
QoP	Quality of Protection
QoS	Quality of Service
SaaS	Software as a Service
SAMM	Service Architecture Meta Model
SA-SLA	Semantic Annotations for Service Level Agreement
SA-WSDL	Semantic Annotations for Web Service Description Language
SIENA	Standards and Interoperability for eInfrastructure Implementation Initiative
SIG	Select Industry Group
SLA	Service Level Agreements
SLA@SOI	SLA@SOI
SLO	Service Level Objective
Stream	Scalable Autonomic Streaming Middleware for Real-time Processing of Massive Data Flows
TREC	Trust, Risk, Ecological, Cost

Acronym	Definition
TTP	Trusted Third Parties
URI	Uniform Resource Identifier
WSAG4J	WS-Agreement for Java
WSLA	Web Service Level Agreement

References

[1] NIST Cloud Computing Reference Architecture, http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505

[2] A. Lenk, M. Klems, J. Nimis, S. Tai, T. Sandholm, "What's inside the Cloud? An architectural map of the Cloud landscape", ICSE Workshop on Software Engineering Challenges of Cloud Computing, Washington, DC, USA, 23-31, 2009

[3] P. Mell, T. Grance, "NIST Definition of Cloud Computing", Sp. Publication 800-145, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, 2011

[4] Information Technology Infrastructure Library (ITIL), http://www.itil-officialsite.com/

[5] TM Forum, "SLA Management Handbook: Volume 2 Concepts and Principles", Release 2.5, TeleManagement Forum, GB 917-2, 2005

[6] P. Allen, "Service Orientation: Winning Strategies and Best Practices", Cambridge Press, 2006

[7] M. Abd-El-Malek, Google: "Challenges in Cloud Storage", 2011

[8] J. McDonald, CloudOne CEO, "The Challenge with Cloud Service Level Agreement Standards", http://blog.cloud-council.org/2013/04/the-challenge-with-cloud-service-level-agreement-standards.html

[9] Forrester Research, "Business Users Are Not Ready For Cloud Storage", 2010

[10] Accenture, "Meeting the Challenges of Cloud Computing", 2011

[11] W. Bumpus, VMWare Director of Standards, "Cloud SLAs: What You Should Be Asking", 2012

[12] Cloud Standards Customer Council: "Cloud SLAs fall short", 2013

[13] 4CaaSt Project, http://4caast.morfeo-project.org/

[14] S. Garcia-Gomez, M. Jimenez-Ganan, Y. Taher, C. Momm, F. Junker, J. Biro, A. Menychtas, V. Andrikopoulos, S. Strauch. "Challenges for the comprehensive management of Cloud Services in a PaaS framework." Scalable Computing: Practice and Experience, 2012

[15] J. L. Vazquez-Poletti, R. Moreno-Vozmediano, I. M. Llorente, E. Oliveros, S. Ortega, M. Jimenez, J. Soriano, A. Menychtas, "Reducing Time to Market with the Platform as a Service Cloud of the Future", in European Research Activities in Cloud Computing, Cambridge Publishing, 2011

[16] M. Papazoglou, W. van den Heuvel, "Blueprinting the Cloud," IEEE Internet Computing, 2011

[17] A. Menychtas, S. Garcia Gomez, A. Giessmann, A. Gatzioura, K. Stanoevska, J. Vogel, V. Moulos, "A Marketplace Framework for Trading Cloud-based Services", 8th International Workshop on the Economics and Business of Grids, Clouds, Systems, and Services (GECON), Paphos, Cyprus, 2011

[18] A. Gatzioura, A. Menychtas, V. Moulos and T. Varvarigou, "Incorporating Business Intelligence in Cloud Marketplaces", International Workshop on Clouds for Business and Business for Clouds (C4BB4C), Madrid, Spain, 2012

[19] A. Menychtas, A. Gatzioura, T. Varvarigou, "A Business Resolution Engine for Cloud Marketplaces", 3rd IEEE International Conference on Cloud Computing (CloudCom), Athens, Greece, 2011

[20] P. Kranas, V. Anagnostopoulos, A. Menychtas, T. Varvarigou, "ElaaS: An innovative Elasticity as a Service framework for dynamic management across the cloud stack layers", 6th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS), Palermo, Italy, 2012

[21] Cloud4SOA Project, http://www.cloud4soa.eu/

[22] CloudScale Project, http://www.cloudscale-project.eu/

[23] G. Brataas, E. Stav, S. Lehrig, S. Becker, G. Kopcak, D. Huljenic, "CloudScale: scalability management for cloud systems", 4th ACM/SPEC International Conference on Performance Engineering, New York, USA, 2013

[24] Modeling and Analysis of Real-time and Embedded Systems - MARTE, http://www.omgmarte.org/

[25] Cloud-TM Project, http://www.cloudtm.eu/

[26] D. Didona, P. Felber, D. Harmanci, P. Romano, J. Schenker, "Identifying the Optimal Level of Parallelism in Transactional Memory Systems", The International Conference on Networked Systems, Best Paper Award, 2013

[27] D. Didona, Paolo Romano, S. Peluso, F. Quaglia, "Transactional Auto Scaler: Elastic Scaling of In-Memory Transactional Data Grids", 9th International Conference on Autonomic Computing (ICAC 2012), San Jose, CA, USA, 2012

[28] M. Couceiro, P. Ruivo, Paolo Romano, L. Rodrigues, "Chasing the Optimum in Replicated In-memory Transactional Platforms via Protocol Adaptation", 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2013

[29] P. Romano, M. Leonetti, "Self-tuning Batching in Total Order Broadcast Protocols via Analytical Modelling and Reinforcement Learning", IEEE International Conference on Computing, Networking and Communications, Network Algorithm & Performance Evaluation Symposium (ICNC'12), 2012

[30] M. Couceiro, P. Romano, L. Rodrigues, "PolyCert: Polymorphic Self-Optimizing Replication for In-Memory Transactional Grids", ACM/IFIP/USENIX 12th International Middleware Conference (Middleware), 2011

[31] J. Paiva, P. Ruivo, P. Romano, L. Rodrigues, "AutoPlacer: scalable self-tuning data placement in distributed key-value stores", The 10th International Conference on Autonomic Computing (ICAC 2013), San Jose, CA, USA, 2013

[32] P. Di Sanzo, F. Antonacci, B. Ciciani, R. Palmieri, A. Pellegrini, S. Peluso, F. Quaglia, D. Rughetti, R. Vitali, "A Framework for High Performance Simulation of Transactional Data Grid Platforms", 6th

International ICST Conference on Simulation Tools and Techniques (SIMUTools), Cannes, French Riviera, 2013

[33] P. Di Sanzo, D. Rughetti, B. Ciciani, F.Quaglia, “Auto-tuning of Cloud-based In-memory Transactional Data Grids via Machine Learning”, 2nd IEEE International Symposium on Network Cloud Computing and Applications (NCCA), London, UK, IEEE Computer Society Press, 2012

[34] D. Didona, P. Di Sanzo, R. Palmieri, S. Peluso, F. Quaglia, P. Romano, “Automated Workload Characterization in Cloud-based Transactional Data Grids”, 17th IEEE Workshop on Dependable Parallel, Distributed and Network-Centric Systems (DPDNS), 2012

[35] CONTRAL Project, http://contrail-project.eu/

[36] R. Cascella, L. Blasi, Y. Jegou, M. Coppola, C. Morin, “Contrail: Distributed Application Deployment under SLA in Federated Heterogeneous Clouds”, Springer, Lecture Notes in Computer Science, 2013

[37] CumuloNimbo Project, http://www.cumulonimbo.eu/

[38] R. Jimenez-Peris, M. Patiño-Martinez, K. Magoutis, A. Bilas, I. Brondino, “CumuloNimbo: A Highly-Scalable Transaction Processing Platform as a Service”, ERCIM News 89, Special Issue on Big Data, 2012

[39] F. Perez-Sorrosal, R. Jimenez-Peris, M. Patiño-Martinez, B. Kemme, “Elastic SI-Cache: Consistent and Scalable Caching in Multi-Tier Architectures”, VLDB Journal, 2011

[40] European Grid Infrastructure, http://www.egi.eu/

[41] FedSM Project, http://fedsm.eu/

[42] EGI Service Catalogue, http://www.egi.eu/services

[43] Resource Centre Operational Level Agreement, https://documents.egi.eu/document/31

[44] Resource Infrastructure Provider Operational Level Agreement, https://documents.egi.eu/document/463

[45] EGI.eu Operational Level Agreement, https://documents.egi.eu/document/1093

[46] ETICS Project, https://www.ict-etics.eu/

[47] ETICS project, Deliverable D4.1, “End-to-end service specification template”, https://bscw.ict-etics.eu/pub/bscw.cgi/d19910/D4.1%20End-to-End%20service%20specification%20template.pdf

[48] H. Pouyllau, G. Carofiglio, “Inter-carrier SLA negotiation using Q-Learning”, Telecommunication Systems Journal Special issue on “Socio-economic Issues of Next Generation Networks”, 2011

[49] G. Carrozzo, N. Ciulli, P. Donadio, A. Cimmino, “The Path Computation Element for the Network Service and Business Plane – Computation of route offers and price modelling for inter-carrier services”, 17th European Conference on Network and Optical Communications (NOC 2012)

[50] A. Jamakovic, T.M. Bohnert, G. Karagiannis, “Mobile Cloud Networking: Mobile Network, Compute, and Storage as One Service On-Demand”, in “The Future Internet – Future Internet Assembly 2013: Validated Results and New Horizons” – Lecture Notes in Computer Science Volume 7858, 2013

[51] GEYSERS Project, http://www.geysers.eu/

[52] GEYSERS project, Deliverable D2.6, “Refined GEYSERS architecture, interface specification and service provisioning workflow”, http://www.geysers.eu/images/stories/D2.6-final.pdf

[53] P.Robinson, A.F. Antonescu, F. Anhalt, J. Aznar, E. Escalona, J.A. Garcia Espin, L.M. Contreras Murillo, P. Vicat Blanc, “SLA management for composite infrastructure as a Service”, whitepaper, http://www.geysers.eu/images/stories/GEYSERS_White_Paper_-_SLA_Management_For_Composite_Infrastructure_As_A_Service.pdf

[54] A.F. Antonescu, P. Robinson, L.M. Contreras-Murillo, J. Aznar, S. Soudan, F. Anhalt, et al (2012). “Towards Cross Stratum SLA Management with the GEYSERS Architecture”. In ISPA 2012

[55] A.F. Antonescu, M. Thoma, P. Robinson, “Service Level Management Convergence for Future Network Enterprise Platforms”, FNMS 2012

[56] Helix Nebula Project, http://www.helix-nebula.eu/

[57] IRMOS Project, http://www.irmosproject.eu/

[58] D. Kyriazis, A. Menychtas, G. Kousiouris, K. Oberle, T. Voith, M. Boniface, E. Oliveros, T. Cucinotta, S. Berger, “A Real-time Service Oriented Infrastructure”, International Conference on Real-Time and Embedded Systems (RTES), Singapore, 2010

[59] A. Menychtas, D. Kyriazis, S. Gogouvitis, K. Oberle, T. Voith, G. Galizo, S. Berger, E. Oliveros, M. Boniface, “A cloud platform for real-time interactive applications “, 1st International Conference on Cloud Computing and Services Science (CLOSER), Noordwijkerhout, The Netherlands, 2011

[60] G. Gallizo, R. Kübert, G. Katsaros, K. Oberle, K. Satzke, S. V. Gogouvitis, E. Oliveros, “A Service Level Agreement Management Framework for Real-time Applications in Cloud Computing Environments”, CloudComp Conference, 2010

[61] G. Gallizo, R. Kuebert, K. Oberle, A. Menychtas, K. Konstanteli, “Service Level Agreements in Virtualized Service Platforms”, eChallenges2009, Istanbul, 2009

[62] R. Kübert, G. Gallizo, T. Polychniatis, T. Varvarigou, E. Oliveros, S. C Phillips, K. Oberle, “Chapter: Service Level Agreements for real-time Service Oriented Infrastructures”, IGI Global Book: Achieving Real-Time in Distributed Computing: From Grids to Clouds, 2012

[63] T. Voith, K. Oberle, M. Stein, E. Oliveros, G. Gallizo, R. Kübert, “A Path Supervision Framework – a key for service monitoring in Infrastructures as a Service (IaaS) Platform”, 36th Euromicro Conference on Software Engineering and Advances Applications (SEAA), 2010

[64] R. Kübert, G. Gallizo, K. Oberle, E. Oliveros , “Enhancing the SLA Framework of a Virtualized Service Platform by dynamic re-negotiation”, eChallenges2010, Warsaw, Poland, 2010

[65] T. Voith, K. Oberle, M. Stein, “Quality of Service provisioning for distributed data center inter-connectivity enabled by network virtualization”, Future Generation Computer Systems, Elsevier , 2011

[66] T. Cucinotta, F. Checconi, G. Kousiouris, D. Kyriazis, T. Varvarigou, A. Mazzetti, Z. Zlatev, J. Papay, M. Boniface, S. Berger, D. Lamp, T. Voith, M. Stein, “Virtualized e-Learning with Real-Time Guarantees on the IRMOS Platform”, IEEE International Conference on Service-Oriented Computing and Applications, SOCA2010, Perth, Australia, 2010

[67] G. Katsaros, G. Kousiouris, S. Gogouvitis, D. Kyriazis, A. Menychtas, T. Varvarigou, “A Self-adaptive hierarchical monitoring mechanism for Clouds”, Elsevier Journal of Systems and Software, 2012

[68] E. Oliveros, T. Cucinotta, S. C Phillips, X. Yang, S. Middleton, T. Voith, “Chapter: Monitoring and Metering on the Cloud”, IGI Global Book: Achieving Real-Time in Distributed Computing: From Grids to Clouds, 2012

[69] G. Kousiouris, D. Kyriazis, S. Gogouvitis, G. Katsaros, T. Varvarigou, “A Service-Oriented Framework for GNU Octave-Based Performance Prediction”, 7th IEEE International Conference on Services Computing (SCC), Miami, USA, 2010

[70] G. Kousiouris, D. Kyriazis, S. Gogouvitis, A. Menychtas, K. Konstanteli, T. Varvarigou, “Translation of application-level terms to resource-level attributes across the Cloud stack layers”, IEEE Symposium on Computers and Communications (ISCC), 2011

[71] MCN Project, https://www.mobile-cloud-networking.eu/site/

[72] MODAClouds Project, http://www.modaclouds.eu/

[73] OPTIMIS Project, http://www.optimis-project.eu/

[74] OPTIMIS Service Manifest, Scientific Report, http://www.optimis-project.eu/sites/default/files/content-files/document/service-manifest-scientific-report.pdf

[75] H. Rasheed, A. Rumpl, O. Wäldrich, W. Ziegler, “A standards-based approach for negotiating service QoS with cloud infrastructure providers” eChallenges2012, Lisbon, Portugal, 2012

[76] J. Tordsson, K. Djemame, D. Espling, G. Katsaros, W. Ziegler, O. Wäldrich, K. Konstanteli, A. Sajjad, M. Rajarajan, G. Gallizo, S. Nair, “Towards holistic cloud management”, European research activities in cloud computing. Newcastle: Cambridge Scholars Publishing, 2012

[77] G. Kousiouris, G. Vafiadis, M. Corrales, “A Cloud Provider Description Schema for Meeting Legal Requirements in Cloud Federation Scenarios”, Conference on e-Business, e-Services, and e-Society (I3E), Athens, Greece, 2013

[78] C. Cacciari, D. Mallmannb, C. Zsigri, F. D’Andria, B. Hagemeier, A. Rumpl, W. Ziegler, J. Martrat, “SLA-based management of software licenses as web service resources in distributed computing infrastructures”, Elsevier Future Generation Computer Systems, 2012

[79] W.Ziegler, “SLAs for energy-efficient data centres: The standards-based approach of the OPTIMIS project”, International Workshop on Energy-Efficient Data Centers (E2DC), Madrid 2012

[80] A. Lawrence, K. Djemame, O. Wäldrich, W. Ziegler, C. Zsigri, “Using service level agreements for optimising cloud infrastructure”, International Conference ServiceWave , Ghent, 2010

[81] PrestoPRIME Project, http://www.prestoprime.org/

[82] Q-ImPrESS Project, http://www.prestoprime.org/

[83] S. Becker, L. Bulej, T. Bureš, P. Hnetynka, L. Kapová, J. Kofron, H. Koziolek, J. Kraft, R. Mirandola, J. Stammel, G. Tamburrelli, M. Trifu, Service Architecture Meta-Model (SAMM), http://www.q-impress.eu/wordpress/wp-content/uploads/2009/05/d21-service_architecture_meta-model.pdf

[84] R. Calinescu, L. Grunske, M. Kwiatkowska, R. Mirandola, G. Tamburrelli, “Dynamic QoS Management and Optimisation in Service-Based Systems”, IEEE Transactions on Software Engineering 37(3), 2011

[85] S. Becker, M. Trifu, R. Reussner, “Towards Supporting Evolution of Service-Oriented Architectures through Quality Impact Prediction”, Proceedings of the First International ARAMIS Workshop, L'Aquila, Italy, July, 2008

[86] H. Koziolek, B. Schlich, C. Bilich, R. Weiss, S. Becker, K. Krogmann, M. Trifu, R. Mirandola, A. Martens, “An Industrial Case Study on Quality Impact Prediction for Evolving Service-Oriented Software”, 33rd ACM/IEEE International Conference on Software Engineering (ICSE), Software Engineering in Practice Track, ACM, 2011

[87] SERSCIS Project, http://www.serscis.eu/

[88] SLA@SOI Project, http://sla-at-soi.eu/

[89] K. Kearney, F. Torelli, C. Kotsokalis, “SLA*: An Abstract Syntax for Service Level Agreements”, Grid Computing (GRID) 2010, 11th IEEE/ACM International Conference on Grid Computing, Brussels, Belgium, 2010.

[90] W. Theilmann, J. Lambea, F. Brosch, S. Guinea, P. Chronz, F. Torelli, J. Kennedy, M. Nolan, G. Zacco, G. Spanoudakis, M. Stopar, G. Armellin, “SLA@SOI Final Report”, September 2011.

[91] RESERVOIR Project, http://www.reservoir-fp7.eu/

[92] Open Cloud Computing Interface, http://www.occi-wg.org/about/

[93] Open Grid Forum, http://www.gridforum.org/

[94] National Institute of Standards and Technology, http://www.nist.gov/index.html

[95] SIENA Project, http://www.sienainitiative.eu/

[96] UK Cabinet Office, https://www.gov.uk/government/organisations/cabinet-office

[97] German Federal Ministry of Economics and Technology, “The Standardisation Environment for Cloud Computing”, http://www.bmwi.de/English/Redaktion/Pdf/normungs-und-standardisierungsumfeld-von-cloud-computing,property=pdf,bereich=bmwi2012,sprache=en,rwb=true.pdf

[98] Stream Project, http://www.streamproject.eu/

[99] V. Gulisano, R. Jimenez-Peris, M. Patiño-Martínez, P. Valduriez, “A Large Scale Data Streaming System”, 30th IEEE Int. Conf. on Distributed Systems (ICDCS), Genoa, Italy, 2010

[100] L. Coppolino, D. De Mari, L. Romano, V. Vianello, "SLA compliance monitoring through semantic processing", Grid Computing (GRID), 2010

[101] VISION Cloud Project, http://www.visioncloud.eu/

[102] N. Mavrogeorgi, S. Gogouvitis, A. Voulodimos, G. Katsaros, S. Koutsoutos, D. Kyriazis, T. Varvarigou, E. Kolodner, “Content Based SLAs in Cloud Computing Environments”, IEEE International Conference on Cloud Computing (CLOUD), 2012

[103] S. Gogouvitis, V. Alexandrou, N. Mavrogeorgi, S. Koutsoutos, D. Kyriazis, T. Varvarigou, “A Monitoring Mechanism for Storage Clouds”, 2nd International Conference on Cloud and Green Computing (CGC), 2012

[104] A. Voulodimos, D. Kyriazis, S. Gogouvitis, A. Doulamis, D. Kosmopoulos, T. Varvarigou, “QoS-oriented Service Management in clouds for large scale industrial activity recognition”, IEEE International Conference of Soft Computing and Pattern Recognition (SoCPaR), 2011

[105] B. Koller, H. Munoz Frutos, G. Laria G, “Service Level Agreements in BREIN”, Springer Grids and Service-Oriented Architectures for Service Level Agreements. Springer, 2010

[106] H. Munoz Frutos, I. Kotsiopoulos, A. Micsik, B. Koller, J. Mora, “Flexible SLA Negotiation Using Semantic Annotations”, Service-Oriented Computing, ICSOC / ServiceWave, 2009

[107] G. Birkenheuer, A. Brinkmann, M. Högqvist, A. Papaspyrou, B. Schott, D. Sommerfeld, W. Ziegler, “Infrastructure federation through virtualized delegation of resources and services: DGSI: Adding interoperability to DCI meta schedulers”, Journal of Grid Computing, 2011

[108] R. Kübert, A. Tenschert, O. Wäldrich, W. Ziegler, D. Battré, “A Service Level Agreement Layer for the D-Grid Infrastructure”, eChallenges2010, Warsaw, Poland, 2010

[109] ETSI GRID12_17, "Grid and Cloud computing Technology: Interoperability and Standardisation for the Telecoms Industry"

[110] ETSI TR102997, "Technical Report: Initial analysis of standardisation requirements for cloud services"

[111] ETSI Cloud Stands Coordination, http://csc.etsi.org