University of Pavia
Executive Doctor of Philosophy
Ph.D. School in Electronics, Electrical and Computer Engineering

A Lean Information Service Architecture for Service Level Management in Multi-Cloud environments

Supervisor:
Prof. Gianmario Motta - University of Pavia

Executive Ph.D. candidate:
Nicola Sfondrini
To myself,

to not forget the sacrifices I made
and to remind me to never give up

“Twenty years from now you will be more disappointed by the things you didn’t do than by the ones you did do. So, throw off the bowlines, sail away from safe harbor. Catch the trade winds in your sails. Explore. Dream. Discover”

Mark Twain
Acknowledgments

After three long years of research, the list of people to thank is very long, so I decided to mention only people that allow me to reach this aim.

Firstly, I would like to express my sincere gratitude to my supervisor, Prof. Gianmario Motta, for the precious support and the constant motivation he provided me during the last eight years.

My loving heartfelt thanks goes to my family, my parents and my brother, for all the support they showed me throughout my career.

Finally, a special thanks to Lisa, the new and wonderful part of my life, to make me feel emotions that I thought possible only in dreams. I decided to dedicate the name of my framework to her, hoping it could be as special as she is to me.
# Summary

Acknowledgments ........................................................................................................ 5
Figure index ................................................................................................................ 11
Table index .................................................................................................................. 15
Abstract ....................................................................................................................... 17

1 Introduction ............................................................................................................... 19
   1.1 Cloud Computing models .................................................................................. 19
   1.2 Cloud Computing evolution phases .................................................................... 21
   1.3 Multi-Cloud model architectures ....................................................................... 22
   1.4 Research approach ............................................................................................ 26
      1.4.1 Problem formulation and scope ................................................................. 26
      1.4.2 Methodology .............................................................................................. 27
      1.4.3 Thesis structure ......................................................................................... 28

2 Systematic literature review .................................................................................... 31
   2.1 A method for Systematic Literature Review (SLR) ........................................... 31
      2.1.1 Introduction ................................................................................................... 31
      2.1.2 Question formulation .................................................................................. 32
      2.1.3 Sources and studies selection ...................................................................... 33
      2.1.4 Information extraction and results summarization ...................................... 33
   2.2 Research trends .................................................................................................. 34
   2.3 Literature summarization ................................................................................... 37
      2.3.1 SLA parameters .......................................................................................... 37
      2.3.2 SLA specification languages ....................................................................... 40
      2.3.3 Main SLA-based resource allocation techniques ........................................ 46
      2.3.4 SLA management frameworks ................................................................... 49
   2.4 Conclusions and validation ............................................................................... 68

3 Enterprise survey .................................................................................................... 71
   3.1 Study methodology ............................................................................................ 71
      3.1.1 Survey purpose ............................................................................................ 71
   3.2 Survey findings .................................................................................................. 74
3.2.1 Cloud model adoption ................................................................. 77
3.2.2 Cloud penetration and workload distribution .................................. 79
3.2.3 Main benefits and constraints ....................................................... 83
3.2.4 Approach to the Service Level Management ................................... 84
3.3 Conclusions and validation ................................................................... 87

4 LISA - Lean Information Service Architecture ........................................... 89
4.1 Cloud service and SLA management lifecycle ....................................... 89
4.2 QoS metrics ....................................................................................... 90
4.3 LISA mapping on OASIS TOSCA ...................................................... 92
4.4 LISA architecture and process flows .................................................... 94
  4.4.1 Knowledge base ............................................................................ 95
  4.4.2 Cloud service catalogue ............................................................... 97
  4.4.3 Service negotiation agent ............................................................. 99
  4.4.4 Advanced monitoring engine ....................................................... 101
  4.4.5 Autonomic resource allocation engine .......................................... 102
  4.4.6 Reporting agent .......................................................................... 104
  4.5 LISA SLA lifecycle coverage evaluation .......................................... 106

5 LISA framework implementation and test .................................................. 107
5.1 Main resource management techniques ................................................. 107
5.2 Experimental environment configuration .............................................. 108
  5.2.1 Multi-Cloud environment configuration ....................................... 109
  5.2.2 LISA deployment configuration ................................................... 110
5.3 Test cases and experimental results ..................................................... 113
  5.3.1 Network performance analysis on on-premise OpenStack ................ 113
  5.3.2 Activity tracker in hybrid Cloud configuration ............................... 117
  5.3.3 Converged data platform in Multi-Cloud configuration .................... 121
5.4 Conclusions and validation ................................................................... 126

6 Conclusions ............................................................................................ 127
  6.1 Key insights ..................................................................................... 127
  6.2 Answers to the research questions .................................................... 128
  6.3 Research limitations ......................................................................... 130
6.4 Future directions .................................................................................................................. 131
7 IEEE and ACM publications .................................................................................................. 133
  7.1 Publications 2015 - onward ......................................................................................... 133
  7.2 Publications 2011 - 2014 .............................................................................................. 134
  7.3 New researches - Accepted and to be published ......................................................... 136
8 Annex .................................................................................................................................. 137
  8.1 Cloudify orchestrator overview .................................................................................... 137
  8.2 LISA framework - Software design ............................................................................. 139
Bibliography ............................................................................................................................ 141
Figure index

Figure 1 - Cloud Computing service models ................................................................. 20
Figure 2 - Size of the Public IaaS market worldwide (millions $) ................................ 21
Figure 3 - Worldwide Cloud IT infrastructure market forecast by development type .... 22
Figure 4 - Cloud adoption stages of growth .................................................................. 23
Figure 5 - Redundant Multi-Cloud architecture .............................................................. 24
Figure 6 - Composite Multi-Cloud architecture ............................................................... 25
Figure 7 - Redundant and composite Multi-Cloud architecture ........................................ 25
Figure 8 - Research Methodology .................................................................................. 27
Figure 9 - Research flowchart ....................................................................................... 28
Figure 10 - Review Methods ......................................................................................... 31
Figure 11 - Literature review filtering process ................................................................. 34
Figure 12 - Researches distribution over the years ......................................................... 35
Figure 13 - Researches distribution by region ................................................................. 35
Figure 14 - Researches distribution by country ............................................................... 36
Figure 15 - Researches distribution by topic .................................................................. 36
Figure 16 - WSLA meta-model ..................................................................................... 40
Figure 17 - WS-Agreement meta-model ......................................................................... 42
Figure 18 - CSLA meta-model ...................................................................................... 42
Figure 19 - SLA* meta-model ....................................................................................... 43
Figure 20 - rSLA meta-model ....................................................................................... 44
Figure 21 - IBM MAPE-K architecture .......................................................................... 48
Figure 22 - SLA management lifecycle .......................................................................... 49
Figure 23 - 4CaaSt SLA management process ............................................................... 51
Figure 24 - Cloud4SOA SLA management architecture ................................................ 52
Figure 25 - SLA@SOI monitoring architecture ............................................................... 54
Figure 26 - CloudScale process flow ............................................................................. 55
Figure 27 - Cloud-TM platform architecture .................................................................. 56
Figure 28 - ETICS SLAs establishment model ............................................................... 57
Figure 29 - Contrail SLA management framework ......................................................... 58
Figure 30 - EGI service management framework ........................................................... 60
Figure 31 - IRMOS SLA management framework ......................................................... 61
Figure 32 - Q-ImPrESS architecture ........................................................................... 62
Figure 33 - DeSVi architecture ................................................................................. 64
Figure 34 - CSLAM architecture and process flow ...................................................... 65
Figure 35 - C@H system architecture ........................................................................ 67
Figure 36 - Survey approach ...................................................................................... 72
Figure 37 - Respondents by country .......................................................................... 75
Figure 38 - Respondents by industry ......................................................................... 75
Figure 39 - Cloud investment by region ..................................................................... 76
Figure 40 - Cloud investment by industry ................................................................. 77
Figure 41 - Private and Public Cloud models ............................................................. 77
Figure 42 - Multi-Cloud model adoption by industry .................................................. 78
Figure 43 - Multi-Cloud model adoption by region .................................................... 79
Figure 44 - Cloud penetration rate ............................................................................ 80
Figure 45 - Cloud penetration rate by industry .......................................................... 80
Figure 46 - Workload in public Cloud by region ......................................................... 81
Figure 47 - Workload in public Cloud by industry ....................................................... 82
Figure 48 - Production workload in Public Cloud by industry ................................... 82
Figure 49 - Main benefits of Public Cloud adoption (average values) ....................... 83
Figure 50 - Main constraints to Public Cloud adoption (average values) ................. 84
Figure 51 - Monitoring tool (a) and CMBD (b) integration ....................................... 84
Figure 52 - Customer portal and CASB adoption ....................................................... 85
Figure 53 - SLM tools adoption by investment rate .................................................... 86
Figure 54 - IT management framework adoption ....................................................... 86
Figure 55 - LiSA Multi-Cloud logo ............................................................................ 89
Figure 56 - LiSA activities mapping on ITIL phases .................................................. 89
Figure 57 - TOSCA service template ....................................................................... 93
Figure 58 - TOSCA application requirements mapping ............................................ 93
Figure 59 - LiSA architecture .................................................................................... 94
Figure 60 - Knowledge base high-level structure ..................................................... 95
Figure 61 - Cloud service catalogue - Process flow .................................................... 97
Figure 62 - Cloud service configuration - Technical view
Figure 63 - Cloud service configuration - Business view
Figure 64 - Private Cloud service negotiation - Process flow
Figure 65 - Public Cloud service negotiation - Process flow
Figure 66 - Autonomic resource allocation engine - SLA violation process flow
Figure 67 - Autonomic resource allocation engine - Scalability process flow
Figure 68 - Reporting agent - Process flow
Figure 69 - Reporting agent - Control panel view
Figure 70 - Reporting agent - Cloud spending view
Figure 71 - MAPE-K adaptation control loop - LISA framework view
Figure 72 - Application generations mapping on Cloud solutions
Figure 73 - Experimental environment network topology
Figure 74 - Experimental Cloud environment configuration
Figure 75 - Network performance analysis - Scenario 1 execution time (seconds)
Figure 76 - Network performance analysis - Scenario 1 workload cost (€)
Figure 77 - Network performance analysis - Scenario 2 execution time (seconds)
Figure 78 - Network performance analysis - Scenario 2 workload cost (€)
Figure 79 - Activity tracker in hybrid Cloud - Network configuration
Figure 80 - Activity tracker in hybrid Cloud - Applications configuration
Figure 81 - Users access frequency to the experimental environment
Figure 82 - Activity tracker in hybrid Cloud - Scenario 1
Figure 83 - Activity tracker in hybrid Cloud - Scenario 2
Figure 84 - Converged data platform in Multi-Cloud - Network configuration
Figure 85 - Converged data platform in Multi-Cloud - Applications configuration
Figure 86 - Converged data platform - Scenario 1 execution time (seconds)
Figure 87 - Converged data platform - Scenario 1 execution cost (€)
Figure 88 - Converged data platform - Scenario 2 execution time (seconds)
Figure 89 - Converged data platform - Scenario 2 execution cost (€)
Figure 90 - Cloudify orchestration engine
Figure 91 - Cloudify open-source components
Figure 92 - Cloud service selection - Sequence diagram
Figure 93 - Billing calculation - Sequence diagram .................................................. 139
Figure 94 - Knowledge base ER diagram (partial view) ............................................. 140
Table index

Table 1 - SLA specification languages comparison ................................................................. 45
Table 2 - 4CaaS SLA lifecycle coverage .................................................................................. 51
Table 3 - 4CaaS Mixed-Cloud models’ coverage ................................................................. 51
Table 4 - Cloud4SOA SLA lifecycle coverage ...................................................................... 53
Table 5 - Cloud4SOA Mixed-Cloud models’ coverage .......................................................... 53
Table 6 - SLA@SOI SLA lifecycle coverage ................................................................. 54
Table 7 - SLA@SOI Mixed-Cloud models’ coverage .......................................................... 54
Table 8 - CloudScale SLA lifecycle coverage ....................................................................... 55
Table 9 - CloudScale Mixed-Cloud models’ coverage .......................................................... 55
Table 10 - Cloud-TM SLA lifecycle coverage ....................................................................... 57
Table 11 - Cloud-TM Mixed-Cloud models’ coverage .......................................................... 57
Table 12 - ETICS SLA lifecycle coverage ............................................................................. 58
Table 13 - ETICS Mixed-Cloud models’ coverage ................................................................. 58
Table 14 - CONTRAIL SLA lifecycle coverage .................................................................... 59
Table 15 - Contrail Mixed-Cloud models’ coverage ............................................................... 59
Table 16 - EGI SLA lifecycle coverage .................................................................................. 60
Table 17 - EGI Mixed-Cloud models’ coverage ................................................................. 60
Table 18 - IRMOS SLA lifecycle coverage ............................................................................. 62
Table 19 - IRMOS Mixed-Cloud models’ coverage ............................................................... 62
Table 20 - Q-ImPrEES SLA lifecycle coverage ...................................................................... 63
Table 21 - Q-ImPrEES Mixed-Cloud models’ coverage ........................................................ 63
Table 22 - DeSVi SLA lifecycle coverage ................................................................................ 65
Table 23 - DeSVi Mixed-Cloud models’ coverage ................................................................. 65
Table 24 - CSLAM SLA lifecycle coverage ............................................................................ 66
Table 25 - CSLAM Mixed-Cloud models’ coverage ............................................................... 66
Table 26 - C@H SLA lifecycle coverage ................................................................................ 68
Table 27 - C@H Mixed-Cloud models’ coverage ................................................................. 68
Table 28 - SLA lifecycle coverage - Overall comparison ....................................................... 69
Table 29 - Mixed-Cloud models’ coverage - Overall comparison ....................................... 70
Table 30 - Survey sections and parameters ........................................................................... 74
Table 31 - Cloud maturity level ................................................................. 76
Table 32 - Cloud penetration level ............................................................. 79
Table 33 - Monitoring algorithm - SLA violation ........................................ 101
Table 34 - Monitoring algorithm - Resource scalability ............................ 102
Table 35 - Plan and execute algorithm - SLA violation .............................. 103
Table 36 - Table 37 - Plan and execute algorithm - Resource scalability .... 104
Table 37 - LISA SLA lifecycle coverage .................................................... 106
Table 38 - LISA Mixed-Cloud models’ coverage ....................................... 106
Table 39 - Resources management techniques comparison .......................... 108
Table 40 - OpenStack virtual instances configuration .................................. 112
Table 41 - Network performance analysis on OpenStack - Scenario 1 .......... 114
Table 42 - Network performance analysis on OpenStack - Scenario 2 ......... 116
Table 43 - Activity tracker analysis - Scenario 1 result .............................. 119
Table 44 - Activity tracker analysis - Scenario 2 result .............................. 120
Table 45 - Result comparison - Scenario 1 vs scenario 2 ........................... 121
Table 46 - Converged data platform - Scenario 1 result (marketing campaign) 123
Table 47 - Converged data platform - Scenario 2 result (churn prevention campaign) 125
Abstract

Cloud Computing recently emerged as a disruptive technology for managing IT services over the Internet, evolving through different phases from Grid Computing, Utility Computing and Software-as-a-Service (SaaS). Despite an initial skepticism, international companies are now widely migrating the IT workload on private and public Cloud solutions to optimize the geographical coverage and launch new digital services. However, the dynamism of the systems and the creation of hybrid and Multi-Cloud environments introduce additional complexity in managing the Quality of Services (QoS) and require the definition of standard Service Level Agreements (SLAs) and structured approaches.

The aim of the research is to evaluate the maturity of the current SLA driven techniques and define a meta-model to cover the entire SLA management lifecycle. To deeply analyze the state of the art, the research evaluated the available Cloud SLA management best practices through a structured literature review and identified the main business needs through a survey on several international companies. The analysis clearly highlights a gap in the SLA management in hybrid and Multi-Cloud environments that is confining the control of the QoS to the use of the Cloud Service Providers (CSPs) monitoring system. Moreover, the main IT management frameworks (i.e. ITIL and COBIT) are still based on the possibility to directly interact with the IT systems and, for this reason, not conceived for Cloud Computing.

Starting from these findings, the research defined a SLA-aware framework, called LISA, to avoid SLA violations and support both consumers and providers across the entire SLA management lifecycle in Multi-Cloud environments. The performance of the framework has been tested in the Innovation lab of a global telco operator, deploying services on private Cloud solutions and public Cloud providers. The experimental results validate the efficiency of LISA in preventing SLA violations and service performance degradations, optimizing the QoS and the degree of control on service’ components deployed across multiple public Cloud providers.
1 Introduction
The digitalization of the businesses and the way in which technology supports is significantly changed during the last years. Moreover, the creation of new disruptive business models and the rise of innovative technologies deeply affected the strategy of multinational companies, forcing them to compete against digital native companies (e.g. Netflix, Spotify, Uber, Airbnb, etc.). In this scenario, Cloud Computing represents the leading actor that changed the information technology, transforming it from a tailored product to a commodity.

1.1 Cloud Computing models
Cloud computing allows to flexible procure, deliver and scale on-demand IT resources, according to the workload changes [1]. Cloud model can be declined towards five essential characteristics, three main delivery models and five deployment models [2].

**Essential characteristics:**
- **On-demand self-service:** a consumer can configure and use computing resources without human interaction with the Cloud provider. Cloud services rely on automation systems and standard configurations to enable the self-service logic;
- **Broad network access:** Cloud resources are available over Internet through standard network interfaces to allow the consumption from different devices (e.g. workstations, laptops, tablet and smartphone);
- **Resource pooling:** Cloud resources are dynamically allocated and deallocated to serve different consumers, using a multi-tenant model;
- **Rapid elasticity:** Cloud resources are automatically provisioned and released according to the actual demand. In this way, consumers have the possibility to virtually allocate infinite resource during demand peaks and release them once they are no longer required;
- **Measured service:** Cloud services are automatically metered through monitoring and reporting systems to provide transparency for both the provider and end user.
Delivery models:

- **Software-as-a-Service (SaaS):** the capability provided is a business application running on a Cloud infrastructure and accessible from various devices through web browser or a thin client interface. In this model, consumers can manage only the application configuration without having control on the underlying virtual infrastructure;

- **Platform-as-a-Service (PaaS):** the capability provided is an advanced development platform to facilitate the creation of applications based on programming languages, services, and tools supported by the Cloud provider;

- **Infrastructure-as-a-Service (IaaS):** the capability provided is a virtual instance where consumers can deploy and run arbitrary operating systems and applications. In this model, consumers can directly manage virtual machines, ensuring a better control on the service performance (Figure 1).

![Essential characteristics](image)

- **Essential characteristics**
  - Broad Network Access
  - Rapid Elasticity
  - Measured Services
  - On-Demand Self-Service
  - Resource Pooling

![Delivery models](image)

- **Delivery models**
  - Software as a Service
  - Platform as a Service
  - Infrastructure as a Service

![Deployment models](image)

- **Deployment models**
  - Public
  - Private
  - Hybrid
  - Community
  - Multi-Cloud

**Figure 1 - Cloud Computing service models**

Deployment models:

- **Private Cloud:** Cloud capabilities are provisioned, on-premise or off-premise, for exclusive use by multiple business units of a single company;

- **Public Cloud:** Cloud capabilities are provisioned for open use, by companies or end-users, and directly managed by the Cloud provider in its own data centers;
• **Hybrid Cloud**: Cloud capabilities are composed by distinct Cloud solutions deployed at the same time on both private and public Cloud environments. This approach offers the benefits of multiple deployment models and allows data and application portability;

• **Community Cloud**: Cloud capabilities are shared across companies and organization linked together by partnership or a common mission. Cloud infrastructure may be owned and managed by one of the entity of the community or an external provider;

• **Multi-Cloud**: Cloud services are deployed simultaneously on different public Cloud providers to enable geo-redundancy or the use of microservices to create a scalable business application [3].

### 1.2 Cloud Computing evolution phases

Despite of an initial skepticism that confined the use of Cloud solutions only for not critical services, enterprises are now migrating a significant part of their production workload, considering public Cloud providers as a natural extension of the private data centers.

For this reason, the adoption of Cloud services will constantly raise towards the years, reaching on a global scale about 83,5 million of dollars only for the IaaS segment [4] (Figure 2).

![Figure 2 - Size of the Public IaaS market worldwide (millions $)](image)

New Cloud providers entered in the market arena toward the last five years, changing the rule of the game and starting a battle based on price reduction and technological
innovation. This wind of change is clearly reflected in the strategy of multinational companies that planned a massive migration on both private and public Cloud environments. Market analysts (e.g. Gartner, IDC, Forrester, etc.) predict a constant growth of public Cloud services adoption reaching 33% of the workload by 2021 and allowing the disposal of private data center facilities (Figure 3). This approach was already adopted by big international companies like AT&T, Verizon and ENEL that streamlined their data centers footprint thanks to a massive service portfolio simplification and the adoption of public IaaS solutions.

Hence, traditional IT is constantly decreasing, and it will cover less than 50% of the technology footprint by 2021.

1.3 Multi-Cloud model architectures

International companies faced the Cloud adoption trough different journeys, according to the readiness of the IT estate and the flexibility of the ICT department. Companies with a solid data center footprint, started their journey with the adoption of private Cloud solutions as natural evolution of the virtualization. This approach allowed to deeply test the Cloud solutions from the implementation phase until the management of the operations, facilitating the acquisition of Cloud oriented skills. On one hand, telecommunication and media companies followed this journey, enabling themselves to act as private Cloud providers for operating companies and partner markets. On the other hand, small enterprise and companies widely based on Unix systems, started their journey from the adoption of public IaaS to develop new applications and evaluate the
impact of the migration to x86 based systems. In addition, financial institutions and energy companies followed this journey setting the scene for a massive ICT department transformation. The confluence to a hybrid Cloud model, represented the third step of both the journeys allowing the simultaneous deployment of business services on private Cloud and public Cloud to decouple back-end and front-end or enable the geo-resilience configuration (i.e. active-passive or active-active setup).

Nowadays, a new trend is catching-on to exploit multiple public Cloud services, to facilitate a wider geographical coverage and reduce the risk of vendor lock-in. To prove the natural evolution of the different Cloud strategies, they have been mapped on the stages of growth model [5]. The growing technological and organizational maturity leads international companies to adopt Cloud solutions for mission-critical services and then to globally scale using multiple public Cloud providers (Figure 4).

For this reason, the Multi-Cloud model, based on the simultaneous use of different IaaS and PaaS providers, will become a standard approach for 80% of multinational companies by 2020, up from less than 10% in 2015 [6, 7]. This model requires specific Cloud-native architectures to enable the move of applications among different locations and to run on multiple providers at the same time (Figure 5).
In the redundant Multi-Cloud architecture, multiple instances of a unique Cloud-native application are replicated between two different Cloud providers to enable resiliency. In case of failure on one instance, workload can be redirected to the redundant version to ensure the business continuity. This architecture is widely adopted to deploy mission-critical applications in public Cloud, ensuring high performance and high availability. Nevertheless, it introduces new challenges across the service lifecycle:

- The application and its configurations require to be simultaneously implemented across multiple and heterogeneous environments
- The performance of the instances, deployed across different public Cloud providers, needs to be tracked through a centralized monitoring tool to ensure the proper degree of control on the Quality of Services (QoS)
- The data consistency requires frequent updates and the replications across both environments

In the composite Multi-Cloud architecture, different components of a single application are deployed across multiple Cloud provider. This architecture is adopted to enable the communication via APIs among different SaaS solutions and the distributed deployed of microservices (Figure 6). This approach requires the use of Cloud ready and Cloud native applications, developed according to the microservices logic, to fully support scalability and the dynamic allocation of computing resources in different locations. All the deployed nodes of the Cloud providers are active to enable the execution of the business application, requiring a low-latency broadband connectivity and a centralized network management.
This configuration requires an advanced application packaging and introduces specific challenges:

- The application implementation requires a complex configuration to ensure the proper management of the components deployed across different providers.
- The management of the application requires a strict control on different capabilities and the configuration of multiple APIs.
- The monitoring of the overall application performance requires a real-time control on each microservice and component.

The redundant and composite Multi-Cloud approach combines two configurations in a single solution architecture (Figure 7).

In this scenario, the application components are distributed and duplicated, for
resiliency purpose, on two different Cloud providers. This architecture is the most advanced and complex configuration of the Multi-Cloud model and it is still not widely explored by multinational companies since it requires a great Cloud maturity to be adopted on a broad scale. The Multi-Cloud model raises a set of opportunities that will strongly affect the Cloud market toward the next years, moving the focus of the IT operations from the virtualized infrastructure to the application level. Software developers will transform themselves into SaaS providers, enabling a native communication among different services (via APIs) and supporting their customization according to the customers’ needs. This model will drive also a significant growth in hybrid Cloud adoption supported by the demand of DevOps oriented solutions (e.g. Docker, Kubernetes, etc.).

1.4 Research approach

The adoption of a Multi-Cloud model ensures to gain the best from each Cloud provider and allows a wider geographical coverage, supporting the creation of a global Cloud strategy. Moreover, splitting the services among different providers strongly mitigate the risk of vendor lock-in and facilitates the implementation of a structured IT resilience and disaster recovery strategy. However, the spread of services among different Cloud providers and different regions introduce additional complexity in the management of the services and on the control of the related performance.

1.4.1 Problem formulation and scope

Multi-Cloud management requires a high level of expertise to properly track and ensure the QoS and it requires the adoption of tailored network configuration, monitoring tools and flexible processes. The main IT frameworks (e.g. ITIL, COBIT, ISO 27001, etc.) are still consolidating their guidelines to ensure an optimized governance on Cloud services but a Multi-Cloud model requires additional controls still not properly defined as best-practices.

Service Level Management is the most impacted step of the service life-cycle in a Multi-Cloud model since the users are losing control on the services due to the geographical spread of both applications and data. Moreover, the tracking of the services performance requires a structured real-time monitoring approach linked to different
Cloud providers, data center regions and technologies.

According to our preliminary analysis the Multi-Cloud approach ensures several benefits but hides critical points of attentions to be addressed to ensure a successful adoption. The aim of the research is to deeply analyze the Multi-Cloud model, identify the impact on the Service Level Management and define a structured framework to facilitate the management across the lifecycle of services deployed in Multi-Cloud environments. The research relies on the following three questions:

- **Question 1**: Is the future of Cloud services adoption widely based on a Multi-Cloud approach?
- **Question 2**: What are the impacts on the services lifecycle caused by a Multi-Cloud model adoption and which are the available best-practices and frameworks to address them?
- **Question 3**: Is Service Level Management strongly affected by the adoption of a Multi-Cloud model?

### 1.4.2 Methodology

The research is based on a structured approach to ensure scientific value and avoid misjudgments during the design of the Multi-Cloud SLM framework and the development of the Cloud Broker.

The methodology adopted to develop the Ph.D. research relies on the four classical scientific components: thesis, theory, application (prototype) and demonstration (Figure 8).
Following these logical steps, the research guarantees a clear problem formulation and a precise analysis of the state of art to clearly position the Multi-Cloud SLM framework among a huge amount of studies.

Moreover, the initial assumptions and the effectiveness of the framework are validated during quality gates to ensure a constant control on the research quality and the full adherence to the research questions and scope (Figure 9).

- **Quality gate 1**: this gate validates the problem formulation through a structured systematic literature review to check the importance of the research topic and the correctness of the initial assumption. This quality gate ensures the creation of a research based on a topic of interest but still not properly addressed by the available studies;

- **Quality gate 2**: this gate validates the importance of the research topic and identifies the lack of a suitable solutions to address SLM in Multi-Cloud environments. This quality gate ensures the creation of business-oriented viewpoint through a survey executed on a set of multinational companies;

- **Quality gate 3**: this gate validates the effectiveness of the Cloud broker, developed according to the Multi-Cloud SLM framework guidelines. The proof of concept, performed at the innovation lab of a global telco operator, analyses the effectiveness of the Cloud broker toward the E2E SLM lifecycle of Cloud services.

**1.4.3 Thesis structure**

The methodology is reflected also in the index of chapters to easily guide the reader through the different steps of the research.

The thesis rages over different academic and business-oriented viewpoints to ensure the creation of a structured analysis on Service Level Management in Multi-Cloud
environments.

- **Chapter 1**: this section aims to highlight the Cloud Computing evolution and to specify the research questions, the scope and the methodologies adopted to deep dive the Service Level Management in Multi-Cloud environments;

- **Chapter 2**: this section aims to analyze the state of the art through a systematic literature review and highlight the main research trends and significant frameworks defined to address the Service Level Management in Multi-Cloud environments;

- **Chapter 3**: this section aims to provide the enterprise viewpoint through a survey based on direct interviews to CIOs and IT managers of multinational companies. The survey is focused on benefits, constraints and strategies adopted in Multi-Cloud environments;

- **Chapter 4**: this section aims to provide a detailed view on the Multi-Cloud SLA framework and highlight the functionalities and the logics behind each single building block. The Multi-Cloud SLA framework is then compared to the main best practices and other frameworks to prove the wider coverage and the better effectiveness in managing Service Level Management in Multi-Cloud environments;

- **Chapter 5**: this section aims to prove the effectiveness of the Multi-Cloud SLM framework, testing it as module of the Cloud orchestrator in a global telco company. The framework is tested on virtual instances deployed on different public and private IaaS solutions to analyze the Quality of Services (QoS) and detect eventual SLA breached;

- **Chapter 6**: this last section presents general comments on the thesis and provide insights on new ideas and future developments.
2 Systematic literature review

To properly analyze the current state of art on Cloud Computing studies and validate the research questions, a systematic literature review was performed on highly cited papers published on top ranked conferences and journals. The analysis considered researches published after 2012 to provide the academic view on the topic in continuity with a previous study focused on the 2008-2011 period [8]. The study aimed to investigate the main research trends on Cloud Computing and provide a structured view on the main adoption models and the related approaches to manage the QoS.

2.1 A method for Systematic Literature Review (SLR)

2.1.1 Introduction

A systematic literature review is an “explicitly formulated, reproducible and up-to-date summary” that includes and extends the statistical results of a meta-analysis methodology. As opposed to a narrative review, a methodological study based on a broad topic, the SLR relies on specific research questions and a structured approach to select and classify academic papers [9] (Figure 10).

The SLR was structured according to a robust approach based on five main steps to avoid biases and errors.

1. **Question formulation**: this step aims to define the research questions and properly determinate the scope of the analysis

2. **Sources selection**: this step aims to identify the sources and the selection criteria used to identify relevant researches
3. **Information extraction**: this step aims to categorize the selected papers according to specific research areas and to identify relevant contributions from each of them.

4. **Results summarization**: this step aims to summarize the information from the selected paper to create a consolidated view.

5. **Results interpretation**: this step aims to extract key insights from the selected paper to identify best practices and future trends.

### 2.1.2 Question formulation

The questions formulation step defines the objectives and the scope of the review through the following parameters:

- **Issue**: this parameter defines the target of the systematic literature review and clarifies the research context.

  The review was focused on the adoption of public and mixed Cloud models (i.e. hybrid and Multi-Cloud) to analyze the main impacts on the SLM activities and identify best practices adopted to ensure the proper QoS.

- **Question**: this parameter clearly defines the aim and the scope of the analysis.

  The review was structured across three research questions.

    - **Question 1**: what are the SLA parameters to be monitored to ensure the proper QoS in public Cloud environments?
    - **Question 2**: what are the main architectures to support SLM in public Cloud?
    - **Question 3**: what are the impacts on SLM activities in a Multi-Cloud model?

- **Keywords**: this parameter sets a list of terms to be used in the citations repository to search relevant studies.

  The review considered different keywords split in two different areas:

    - Cloud models: public Cloud, hybrid Cloud, Multi-Cloud
    - Service level management: Cloud QoS, Cloud SLA, Cloud SLM, Cloud resource allocation, Cloud SLA architecture

- **Control**: this parameter highlights existing researches used as starting point for
the review and the time frame considered for the analysis.

The review scouted relevant studies published after 2012 to provide a comparison with a previous study published in 2011 [8].

- **Effect**: this parameter clearly defines the benefits expected by the study.

The review aimed to provide a clear view on the SLM architectures and best practices and to identify constraints driven by the adoption of a Multi-Cloud model.

### 2.1.3 Sources and studies selection

The definition of the selection criteria is a crucial step to ensure the achievement of the goals initially set for the systematic literature review. The first phase is the definition of the criteria to select the search engines to be used to scout the literature. To ensure scientific relevance, the review considered only global citations repositories of peer-reviewed literature:

- IEEE Xplore digital library (ieeexplore.ieee.org)
- ISI Web of Knowledge (webofknowledge.com)
- ACM Digital Library (libraries.acm.org)
- Scopus Database (info.scopus.com)

Likewise, the second phase is focused on the definition of criteria used to filter the researches retrieved by the citation repositories. To reduce the number of researches and to ensure the identification of the main relevant ones, the review considered a structured set of criteria:

- Full papers and work-in-progress papers only (no short papers)
- Pragmatic researches only, no theoretical papers (i.e. case studies, surveys, simulations, literature reviews and position papers)
- Relevant studies with a good citation rate toward the years

### 2.1.4 Information extraction and results summarization

These two steps represent the effective execution of the literature view and are focused on the categorization of the researches and the extraction of key insights from them. To ensure the creation of a clear and structured analysis, the review was based on
categorization framework able to cover all the areas of interest:

- **SLA parameters**: this domain contains researches focused on the contract negotiation between Cloud consumers and CSPs to highlight the main SLA parameters and the available approaches to track them against the actual QoS;

- **SLA specification languages**: this domain contains researches focused on the definition of SLA meta-models and SLA templates to simplify the creation of a digital version of the contract;

- **SLA-based resource allocation techniques**: this domain contains researches focused on SLA-oriented architectures and the resources allocation methodologies available to simplify the management of the QoS;

- **SLA management frameworks**: this domain contains researches focused on SLA management framework to measure their ability to cover the entire SLA lifecycle.

In addition to the summary of the current literature, the review performed a statistical analysis (i.e. paper distribution per year, region, category, etc.) to provide a clear view on the research trends and to highlight the most active countries and the most cited researches.

### 2.2 Research trends

The execution of the literature review clearly highlighted that SLA management in Cloud environments is an important research topic. The initial scouting retrieved 182 significant researches, but the filters applied during the studies selection phase reduced the number of researches to 51 (Figure 11). Even if, many SLA specification languages and SLA management frameworks were discussed on different papers, the literature review considered only the original researches.

![Figure 11 - Literature review filtering process](image)

To prove the value of the sample, the literature review calculated the average citation
and the weighted-citation value towards the years. It achieves in average 113 average citations and more than 20 citations per year. The distribution of the researches over the years shows a peak in 2012 and 2013, concurrently with the adoption of structured Cloud strategy by several international companies (Figure 12). To cope the need of a better control on Cloud services, high-tech companies directly financed researches and facilitated the creation of test environments in their laboratories.

![Figure 12 - Researches distribution over the years](image)

In the same way, the researches distribution by region highlights a wide majority of papers from European research centers. This result is validated by the large number of researches focused on Cloud SLA management directly supported by the European Union (EU) to facilitate the creation of best practices. Likewise, Asia, America and Oceania research centers created a significant number of studies to facilitate the creation of hybrid and Multi-Cloud ecosystem and to ensure the compliance with local regulations about cyber-security and data privacy (Figure 13).

![Figure 13 - Researches distribution by region](image)

Furthermore, a structured view on the researches distribution by country highlights a good distribution of the eurozone nations and the leadership of Australia that can be considered one of the most active research pole on Cloud Computing thanks to the Cloud
Computing and Distributed Systems (CLOUDS) laboratory (Figure 14).

Finally, the view on the selected categorization framework, shows the lack of significant literature reviews on SLA management in Cloud environments [10-13] and the presence of specification languages recognized as standards to automate the contract negotiation phase (i.e. WSLA, WS-Agreement, SLA* and CSLA). SLA parameters are widely analyzed but most of the studies provide only a list of significant values to be tracked for each Cloud delivery model without any guidelines on how to measure them against the actual performance. Most of the studies (70%) identified during the literature review are focused on SLA-based allocation techniques and SLA management frameworks, proving the importance of defining automated approaches to track the service performance and ensure the agreed QoS (Figure 15).
2.3 Literature summarization

The literature review surveyed top ranked studies focused on SLA management in Cloud environments and analyzed the available solutions to evaluate the maturity of the research topic and to highlight potential areas of improvement. To simplify the extraction of key insights from the selected studies, the literature summarization follows the primary domains defined in the classification framework.

2.3.1 SLA parameters

The adoption of public Cloud solutions requires the negotiation of a contract to protect both consumers and providers over the entire service lifecycle. The SLA parameters and the related threshold values must be precisely specified in the contract to facilitate the tracking of the service performance. The literature review highlighted a set of standard parameters applicable across the entire Cloud stack (i.e. IaaS, PaaS, SaaS)[14, 15]:

- **Availability**: probability that Cloud service is “up and running” and correctly reachable by the users during the agreed time slots. To ensure the high availability (i.e. 99,999%) Cloud providers offer structured redundancy configuration and geo-resilience features;

- **Response time**: availability (in hours) of the service desk to address the resolution of an incident to the right department. This parameter is crucial to speed-up the corrective actions to fix the issue;

- **Resolution time**: time elapsed between the incident detection and the actual resolution. Since this parameter is strictly related to the service availability, Cloud providers offer different levels of support;

- **Security**: system hardening features and data encryption methodologies offered to protect business services from undesired access. To avoid data sniffing and “man in the middle” attacks, Cloud providers are offering dedicated and secured virtual network;

- **Scalability**: ability to increase the amount of available resources according to the actual needs. This parameter is crucial to ensure the business continuity during unexpected demand peaks. However, since Cloud systems are based on a pay-per-use approach, this value must be defined considering the trade-off between
the business impact caused by a service interruption and the cost of the additional resources;

- **Data management**: data management policies to ensure the compliance with international and local regulations. This parameter includes the data location and the retention policy to allow auditing activities on data after the service termination;

- **Penalty model**: penalties calculation formula based on the duration of the SLA breach. This parameter is very important since the Cloud providers recognize a refund based on the value of the Cloud service and not on the business damage generated by the incident.

To ensure the proper degree of control on the QoS, Cloud providers offer a wide range of parameters tailored for each level of the Cloud stack. These parameters are often specified in the contract as plain text, easily interpretable by human but non-readable for standard monitoring tools. Moreover, the Multi-Cloud model increases the complexity since it requires the management of several contracts and the matching between different parameters [16]. To overcome this gap, it is necessary to implement language processing techniques and normalize the parameters among the different providers [17].

### 2.3.1.1 IaaS parameters

IaaS model ensures a high degree of control on the Cloud services and allows consumers to select the best hardware configuration to satisfy the business requirement. This model can be considered as an advanced hosting solution since the consumers have a direct control on the entire IT stack and all the SLA parameters are focused on the performance of the virtual infrastructure [18].

- **CPU speed**: number of virtual CPU cores and related clock rate. This parameter is strictly related to the throughput of the virtual instance;

- **Memory performance**: RAM size, latency and frequency assigned to each virtual instance;

- **Storage space**: net storage size and read/write speed (I/O performance). Any kind of redundancy or backup space are transparent for the consumer and paid
according to the selected support level;

- **Service scalability**: maximum amount of additional virtual resources added to the service in case of demand peak. This parameter can be declined as vertical scalability, to increase resources on the current virtual instance, or horizontal scalability to clone an additional instance of the service;

- **Boot time**: maximum time required to power-on or restart the virtual instance. This parameter is important when the service is used during a specific time frame and then switched-off to avoid unnecessary costs;

- **Bandwidth**: download/upload speed and average network latency. This parameter is key for high-computing and streaming services.

### 2.3.1.2 PaaS parameters

PaaS is the less popular Cloud delivery model and it provides an integrated platform to ensure to developers all the features needed to develop, test, deliver and up-date Cloud-oriented applications [19].

- **Flexibility**: ability to support different programming languages and availability of integrated engines and runtime interpreters;

- **Integration**: availability of APIs to communicate with external services and tools to simplify the development and the execution phases (e.g. ITSM tools, patterns repositories, etc.);

- **Portability**: ability to execute the application on multiple PaaS solutions to avoid vendor lock-in.

### 2.3.1.3 SaaS parameters

SaaS model enables the use of web-based applications provided according to the pay-per-use subscription logic. This model is on the top of the Cloud stack and allows a limited control on the service, confining the consumer’s activities only to the users creation and configuration [20]:

- **Upgrade frequency**: time elapsed between two different releases of the SaaS solution. This parameter includes the update frequency to measure the reactivity of the provider to release patches to fix bugs or release new minor features;

- **Customizability**: possibility to customize or add new features to the standard
solution. This parameter facilitates the creation of tailored solutions to better cope the business requirements;

- **Integration**: availability of APIs to communicate with external services and repositories (e.g. Microsoft Office suite, ticketing systems, etc.).

### 2.3.2 SLA specification languages

The negotiation phase is a crucial step of the SLA lifecycle since it defines all the parameters to be included in the contract and the threshold values to be monitored during the execution phase. To simplify this activity, researchers developed different meta-models based on structured SLA templates and standardized mark-up languages [11]. Furthermore, the use of a standardized taxonomy and the creation of XML-based contracts, strongly facilitate the creation of Multi-Cloud environments.

#### 2.3.2.1 Web Service Level Agreement (WSLA)

Web Service Level Agreement (WSLA) defines a standardized language to describe and monitor SLA parameters for web-based and Cloud services [21]. The meta-model includes structured templates to facilitate the negotiation phase, allowing the translation of both business and technical level objectives into SLAs. WSLA covers the entire SLA management lifecycle and allows the definition of new parameters to ensure the tracking of the service performance.

The meta-model relies on three main conceptual blocks (Figure 16) [22]:

![Figure 16 - WSLA meta-model](image-url)
• **Parties:** this block describes the signatory parties involved in the contract and all the external parties involved in the provisioning and management of the service,

• **Service description:** this block contains all the business and technical parameters required to instantiate the service and track the related performance during the execution phase. The service description is based on an XML schema to facilitate the interpretation of the SLAs by monitoring systems;

• **Obligations:** this block describes the corrective actions to be taken in case of SLA breaches and the penalties model to calculate the refund to be included in the next billing cycle.

Since the latest WSLA release was published by IBM in 2003, the meta-model is now integrated and evolved into WS-Agreement.

### 2.3.2.2 Web Service Agreement (WS-Agreement)

WS-Agreement is a SLA specification language defined by the Open Grid Forum to automatize the negotiation phase [23]. The meta-model automatically generates a set of possible agreements based on the requirements highlighted during the service modelling phase. Furthermore, WS-Agreement supports the update of SLA parameters through the creation of a XML-based version of the contract to be digitally signed-off when both consumer and Cloud provider agreed on the service performance required during the execution phase.

The meta-model relies on three main conceptual blocks (Figure 17):

• **Agreement schema:** this block contains the offer schema used by the agreement initiator to define service requirements and map them on structured SLA parameters;

• **Agreement template schema:** this block contains pre-compiled digital contracts used by the agreement responder to promote acceptable offers according to the initial requirements;

• **Ports type:** this block contains communication patterns used during the negotiation phase to accept or reject the offer and to track the QoS during the execution phase.
2.3.2.3 Cloud Service Level Agreement (CSLA)

CSLA is a specification language tailored for Cloud services and facilitate the measurement of the QoS in scalable and dynamic environments [24]. The meta-model allows the description of SLA parameters in any language across the entire Cloud stack according to three main blocks (Figure 18):

- **Template**: this block contains all the SLA parameters, the guarantees of the Cloud provider in case of SLA breaches and the details about the billing cycle and
termination clauses

- **Parties**: this block describes the signatory parties involved in the contract and all the supporting parties involved in the provisioning and management of the service;

- **Validity**: this block contains parameters related to the contract duration.

### 2.3.2.4 SLA*

SLA* is a domain-independent syntax defined by the SLA@SOi project to create SLA template readable by negotiation engines and monitoring tools [25, 26]. SLA* allows the formalisation of contractual parameters in any language overcoming the restriction caused by the use of markup languages (e.g. XML and JSON).

A SLA* template contains five main blocks (Figure 19):

- **SLA attributes**: this block contains parameters related to the contract duration and the details for the contract termination

- **Parties**: this block describes the signatory parties involved in the contract and all the supporting parties involved in the provisioning and management of the service. This parameter is crucial in Multi-Cloud environment to properly define roles and responsibilities during the entire service lifecycle;

- **Service description**: this block contains all the business requirement to be translated as performance metrics to facilitate the monitoring of the QoS;

- **Variable declaration**: this block contains all the SLA parameters agreed during the negotiation phase and the performance metrics to be tracked during the
execution phase;

- **Terms of the agreement**: this block contains the mapping between the SLA parameters, the related corrective actions and the obligations of the Cloud provider in case of SLA breaches.

### 2.3.2.5 rSLA

rSLA is a specification language defined on Ruby DSL to simplify the declaration of SLA and SLO parameters and enable the use of them through the main configuration management tools (e.g. Chef, Puppet, etc.) [27]. Moreover, the rSLA language allows the users to specify the notification criteria on different event-condition-action rules and enable the definition of composite metrics.

The rSLA language is part of a framework defined on three main components (Figure 20):

- **SLA language**: this component provides the taxonomy to specify SLA and SLO parameters among the different actors involved in the provisioning of Cloud services;
- **rSLA service**: this component interprets the parameters specified in the contract
and implements the behaviors (e.g. notification, implementation of corrective actions, etc.) agreed during the negotiation phase.

- **Xlets**: this component offers a standard API-based interface to track the service performance during the execution phase. Xlets provides two different interfaces:
  - **Runtime interface**: it specifies features and functionalities offered by the application instance (e.g. monitoring, reporting, etc.)
  - **Configuration interface**: it contains tenant-specific information to enable multi-tenancy and application customizations

### 2.3.2.6 SLA specification languages comparison

The analyzed SLA specification languages are linked together, and they are based on a common set of metrics. However, a structured comparison on four different parameters highlights a different degree of maturity (Table 1):

- **Multi-party**: ability to manage additional actors during the negotiation and monitoring phases (in addition to consumer and Cloud provider);
- **Cloud ready**: ability to natively consider Cloud-oriented metrics and to track performance of services deployed on Cloud environments;
- **Business metrics**: ability to manage (as plain text) business requirements as part of the contract;
- **Technical metrics**: ability to translate business requirements in technical parameters readable by monitoring tools;
- **Flexible syntax**: ability to define digital contracts in any kind of languages overcoming the limits of using mark-up languages only.

<table>
<thead>
<tr>
<th></th>
<th>Multi-party</th>
<th>Cloud ready</th>
<th>Business metrics</th>
<th>Technical metrics</th>
<th>Flexible syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSLA</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>WS-Agreement</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>SLA*</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CSLA</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>rSLA</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1 - SLA specification languages comparison

The comparison shows that all the SLA specification languages can manage multiple
parties and define technical parameters in the digital contracts. WSLA cannot cover most of the parameters and can be considered as a deprecated language. WS-agreement and SLA* offer additional features but these meta-models require the definition of tailored SLA parameters to properly manage Cloud services; contrariwise CSLA and rSLA can be considered as the most innovative Cloud-oriented SLA specification languages.

### 2.3.3 Main SLA-based resource allocation techniques

The dynamic allocation of virtual IT resources ensured by scalable Cloud solutions can be used to address specific business and technical needs like cost-effectiveness, compliance with the agreed service performance and other specific objectives according to the adopted delivery model [28, 29]:

- **Workload prediction**: this objective was deeply studied to ensure the dynamic allocation of additional virtual resources to anticipate the demand peaks and enable the business continuity. All the main resource allocation approaches for workload prediction rely on artificial intelligence and advanced analytics techniques to analyze the past workload trends and identify variables and conditions occurred before any demand peaks [30-32];

- **SLA violation detection**: this objective is strictly related to the workload prediction and it extends the use of advanced analytics techniques to the monitoring of virtual instances and physical assets to prevent system faults [33, 34]. The allocation techniques defined to prevent SLA violation are largely adopter by Cloud providers to grant the compliance with the SLAs agreed in the contract;

- **Energy efficiency**: this objective is focused on the energy consumption reduction to execute Cloud service [35-37]. Contrariwise to the workload prediction, this objective is focused on the reduction of physical assets powered-on to support the service execution. For this reason, the energy efficiency can be reached only through an advanced capacity management and the fast disposal of physical IT resources (mainly server and storage);

- **Cost-effectiveness**: this objective is focused on the net profit maximization for Cloud providers and it can be reached thanks to overbooking techniques and
geographical applications distribution [38-40]

Many SLA-based resource allocation techniques were developed over the last years to maximize the Cloud services’ efficiency. Despite the literature propose a wide range of solutions, they can be summarized on four main approaches:

- **Policy-based approach**: this approach relies on a set of conditions used to trigger specific actions. Since policies can be successfully implemented in environments with limited adaption scenarios this approach is not suggested for complex Cloud deployments. In fact, the dynamic negotiation of additional SLA parameters can introduce a new policy that can generate conflicts with the existing ones [41, 42];

- **Heuristic-based approach**: this approach relies on the definition of heuristics able to cover the scenarios the system will have to manage during the execution phase [43, 44]. Heuristics can define both simple policies (i.e. yes or no decision) and multilevel logics to properly manage complex scenarios. This approach is limited by the ability of Cloud designers to forecast most of the scenarios to avoid unexpected behaviors and generate SLA breaches. To overcome this limitation, Cloud providers adopted advanced analytics tool to predict uncovered scenarios and support the design of ad-hoc heuristics;

- **Optimization approach**: this approach relies on machine learning and proactive fault-tolerance techniques to predict and react to SLA breaches [45, 46]. The optimization approach is largely used to predict system failures and demand peaks but, similarly to the heuristic-based approach, it requires the definition of advanced analytics models during the service design phase;

- **Economic-based approach**: this approach relies on economic principles like auctions, Nash equilibrium and financial options to ensure cost-effectiveness of and compliance with the agreed SLA parameters [47]. The economic-based approach ensures an easy implementation of a SLA-based resource allocation model, but it is strongly affected by the limited number of SLA parameters manageable in parallel.

All these SLA-based resource allocation techniques rely on the MAPE-K control loop, an architectural blueprint defined by IBM for autonomic computing. The MAPE-K architecture is composed by five blocks [48] (Figure 21):
- **Knowledge**: this block contains the standard data shared among the other blocks and it is mainly divided in four parts [49].
  - **Managed system knowledge**: it provides relevant information on the resources status, capacity and performance;
  - **Environment knowledge**: it provides information on the context in which the resources are situated;
  - **Concern knowledge**: it provides the details on the required resources to ensure the QoS;
  - **Adaptation knowledge**: it provides the information about the corrective actions executed to reach the system stability;
- **Monitor**: this block collects information from the managed resources to track their capacity, the actual performance and particular system’ behaviors that requires additional analysis;
- **Analyze**: this block analyses the reasons of significant behaviors provided by the monitor module. In case of corrective actions, it sends a change request to the plan module;
- **Plan**: this module defines the corrective actions required to ensure the QoS. This module can rely on simple commands or structured workflow, according to the complexity of the environment in scope;
- **Execute**: this module executes the corrective actions required to ensure the stability of the system and it updates the knowledge module accordingly.

![IBM MAPE-K architecture](image)

Figure 21 - IBM MAPE-K architecture
2.3.4 SLA management frameworks

The lack of direct control on the deployed virtual resources and the difficult monitoring of the QoS, highlighted as main constraints in hybrid Cloud and Multi-Cloud models, pushed the researchers to develop several SLA oriented frameworks [13]. The frameworks analyze different phases of the SLA lifecycle to address the main criticalities according to a structured set of guidelines. The entire SLA management life cycle can be summarized in six main phases, used during the review to evaluate the selected SLA oriented frameworks (Figure 22)[12].

- **Service modelling:** this phase aims to identify business and technical requirements of the service to be deployed in public Cloud environment and define the related parameters to be mapped within the SLA (e.g. availability, incident resolution time, etc.)

![Figure 22 - SLA management lifecycle](image)

- **SLA negotiation:** this phase translates the requirements in SLA parameters and initiates the negotiation between consumer and CSP. The negotiation process includes also monetary aspects to complete the contract sign-off. This phase can be strongly automated using SLA specification languages like WSLA and WS-agreement

- **SLA establishment:** this phase drives the implementation of the Cloud service according to the selected SLA parameters (e.g. redundancy, scalability). The main CSPs are reluctant in defining tailored SLA parameters and they prefer to offer predefined service classes (e.g. bronze, silver and gold)

- **SLA monitoring:** this phase is in charge to track in real-time the performances of the service and map them against the agreed SLA parameters. The monitoring considers both functional and non-functional requirements to ensure the proper
control on the QoS

- **SLA violation alert**: this phase is triggered in case of incident and it measures the duration of the SLA breach to calculate the business impacts and formalize the penalties to be included in the next billing cycle

- **SLA reporting**: this phase periodically generates a SLA report to provide a clear view on the performance supplied by the CSP during a specific time-frame. SLA reporting period should be defined during the negotiation phase to ensure transparency and facilitate auditing activities

- **SLA termination**: this phase manages the termination of the contract after the completion of the service or in case of serious violations caused by any of the parties

To deeply analyze the literature, the review ranked the main SLA oriented frameworks according to the overall quality, the coverage of the entire SLA management lifecycle and the ability to track services deployed on hybrid and Multi-Cloud environments.

### 2.3.3.1 4CaaSt project

The 4CaaSt project [50, 51] defined a platform to facilitate the creation of a Cloud ecosystem composed by applications from different CSP and ensure the tracking of the related performance. The project relies on two main components (Figure 23):

- **4CaaSt Marketplace**: this component enables the “self-service” provisioning of services acting as a centralized point of delivery for the Cloud demand. Furthermore, the marketplace facilitates the creation of composite services through the connection of micro-services offered simultaneously by different providers. To facilitate the negotiation phase, the marketplace includes a business simulation module in charge of mapping technical requirements on a structured set of SLA parameters

- **4CaaSt platform**: this component tracks the QoS to identify incidents and SLA breaches. The 4CaaSt elasticity management module provides an essential SLA enforcement engine and it measures the performance of each micro-service to identify bottlenecks or service interruptions caused by a specific Cloud provider
In summary, the 4CaaSt project represents an advanced SLA-oriented framework able to cover the entire SLA management lifecycle (Table 2) and to manage services deployed on hybrid and Multi-Cloud environments (Table 3). The only gap highlighted during the analysis of the framework is the lack of a business-oriented viewpoint to guide the Cloud consumers in defining the best service configuration according to the business requirements only (no technical skills required).

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partially covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
</tbody>
</table>

Table 2 - 4CaaSt SLA lifecycle coverage

<table>
<thead>
<tr>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
</tbody>
</table>

Table 3 - 4CaaSt Mixed-Cloud models’ coverage
2.3.3.2 Cloud4SOA

The Cloud4SOA project[52] developed a Multi-Cloud platform to negotiate technical features and track the performance of applications deployed on multiple public Cloud environments. The Cloud4SOA framework offers three main functionalities across the SLA lifecycle (Figure 24):

- **SLA negotiation**: this feature relies on WS-agreement standard to allow providers and customers to dynamically negotiate the best SLA parameters according to the services’ requirements

- **SLA enforcement**: this feature is based on a unified interface to provide a consolidated overview of the performance of the different CSPs. This interface is connected to the Cloud services via REST API and provides a set of predefined parameters, at application and infrastructure level, to simplify the comparison among the deployments.

- **Recovery from SLA violation**: this feature suggests the best service recovery approach in case of SLA breach and facilitates the calculation of penalties to be included in the next billing cycle. To simplify the identification of the best recovery strategy, the application should contain a set of policies defined during the design and development phases

In summary, the Cloud4SOA framework provides an advanced engine to automatically recover the services in case of incident, but it is not able to cover the entire SLA management lifecycle (Figure 24).
The framework represents one of the most innovative approach to negotiate, deploy and manage PaaS and IaaS solutions running on Multi-Cloud environments (Table 5).

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partially covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 4 - Cloud4SOA SLA lifecycle coverage

<table>
<thead>
<tr>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
</tbody>
</table>

Table 5 - Cloud4SOA Mixed-Cloud models’ coverage

2.3.3.3 SLA@SOI

The SLA@SOI project [53] defined a set of guidelines to manage the SLA across the service lifecycle and the entire Cloud stack from the business applications to the underlying infrastructure. The framework relies on four main features (Figure 25):

- **Service description**: this module describes both functional and non-functional requirements through different syntactic formats (i.e. XML or human language) and creates a SLA template based on a hierarchical approach to highlights the dependencies among service components and the related performance.

- **SLA negotiation**: this module relies on a domain-agnostic engine to facilitate the interaction between Cloud consumer and CSP during the definition of the SLA parameters to be included in the contract.

- **Scalable SLA monitoring**: this module relies on three layers and provides a dynamic and high-configurable monitoring system. The “SLA manager layer” identifies the components to be tracked to ensure a clear visibility on the QoS. The “low-level monitoring layer” defines operational SLAs and specifications for each component and enables the “sensing and adjustment layer” to track in real-time the service performance.

- **Interoperability standards**: this feature was developed as part of the FP7 project and defined the Open Cloud Computing Interface (OCCI), a RESTful protocol to facilitate the execution of management activities towards the entire Cloud stack.
In summary, SLA@SOI framework manages the SLA lifecycle across the autonomous negotiation of the contract, the provisioning of the service and the monitoring and refinement of the SLA parameters during the operational activities (Table 6). However, the framework does not support the SLA termination phase and the self-configuration of services based on business parameters. Moreover, even if the framework was largely considered as starting point to develop other SLA-oriented frameworks, it does not specify guideline to manage services deployed in a Multi-Cloud environment (Table 7).

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partially covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 6 - SLA@SOI SLA lifecycle coverage

<table>
<thead>
<tr>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 7 - SLA@SOI Mixed-Cloud models’ coverage

2.3.3.4 CloudScale

The CloudScale project [54] defined guidelines to support the CSPs in analyzing, predicting and managing scalability requests in public Cloud environments. The framework is focused on scalability aspects and their translation in SLA parameters following two main processes (Figure 26):
- **Scalability specification**: this phase relies on the Scalability Description Language (ScaleDL), a tailored specification language to identify the value of each parameter required to enable the service scalability (e.g. usage, capacity, cost, business criticality, etc.)

- **Automatic root cause analysis**: this module tracks the service performance and identifies causes of SLA violations caused by scalability issues. In case of incident or service degradation, the module analyses the source code of the application to identify the root cause and define the proper resolution activities (based on a what-if analysis)

In summary, the CloudScale project is strictly focused on the definition of SLA parameters and incident resolution approaches to enable the service scalability. For this reason, the coverage of the SLA management lifecycle is limited to the operational activities (Table 8). Furthermore, the framework relies on a set of guidelines applicable for public and hybrid Cloud, but it does not specify a tailored approach for services deployed in Multi-Cloud environments (Table 9).
2.3.3.5 Cloud-TM

The Cloud-TM project [55] developed a data-centric PaaS solution to speed-up the application development phase and reduce the operational costs. The platform provides data management protocols tailored for Cloud environments and simplifies the delivery automation of SaaS and IaaS solutions. The framework relies on two main blocks (Figure 27):

- **Data platform**: this module supports the development phase facilitating the storage of data across distributed nodes and providing access to standard patterns to easily create load balancing, scheduling and thread synchronization features. The core of the module relies on Red Hat’s Infinispan, an in-memory data grid, customized to ensure high-availability of data.

- **Autonomic manager**: this module manages the scalability of the deployed Cloud solution through the dynamic reconfiguration of data distribution and the reconfiguration of the service. The module contains a SLA negotiation and specification engine to manage service requirements and budget constraints.

![Figure 27 - Cloud-TM platform architecture](image_url)

In summary, the Cloud-TM framework provides a high-performance platform to speed-up the creation of Cloud oriented applications and simplify the SLA negotiation and establishment for IaaS and SaaS solutions (Table 10). The use of API to natively interconnect different Cloud providers enable the management of services distributed on Mixed-Cloud environments (Table 11).
### 2.3.3.6 ETICS

The ETICS project [56] defined a network control and management technology to automate the service delivery across multiple providers and ensure the proper QoS. Furthermore, the framework facilitates the calculation of a revenue sharing model among the different actors involved in the Cloud service supply chain.

To support the provisioning of Multi-Cloud solutions, the framework defined a set of hierarchical SLA parameters among network providers, applications providers and Cloud consumers. The SLA negotiation phase can be supported by a central entity (i.e. an independent broker) or distributed through a cascade model on each actor involved in the end-to-end lifecycle of the Cloud services (Figure 28).

Moreover, to extend the coverage on the SLA lifecycle, the ETICS project defined a SLA template to easily combine business and technical aspects across the different actors.

![ETICS SLAs establishment model](image)

In summary, the ETICS project is focused on the SLA management across the Cloud value chain.
chain through the definition of service plane technologies and tailored SLA templates applicable on different Cloud providers (Table 12, Table 13).

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 12 - ETICS SLA lifecycle coverage

<table>
<thead>
<tr>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
</tbody>
</table>

Table 13 - ETICS Mixed-Cloud models’ coverage

2.3.3.7 CONTRAIL

The Contrail project [57] designed an elastic PaaS solution to coordinate SLA management and interoperability among different IaaS providers. The platform includes a federation layer to allow the collaboration among small Cloud providers and enlarge their services portfolio. The framework is based on three features (Figure 29):

- **SLA specification**: this feature extends the SLA model proposed by the SLA@SOI project thanks to the adoption of the Open Virtualization Format (OVF) to represent pool of resources available across different providers as a unique virtualized layer

- **Quality model**: this feature includes a set of terms to classify business services and infrastructure components as unobservable, observable and enforceable according to their nature and the requirements captured during the service modelling phase. Furthermore, to ensure the scalability of the services and track their evolution towards the time, the model differentiates between static and dynamic resources
• **SLA interaction model**: this feature allows the consumers to negotiate the best Cloud solution available within the federation combining services from different Cloud providers. Moreover, to simplify the execution of an application on multiple Cloud providers, the model includes a schema to split the SLA to identify the proper revenue sharing agreement and the responsibilities in case of incidents or services’ degradations.

In summary, the framework implements a federation layer to act as a unique interface for Cloud consumers to facilitate the creation of Multi-Cloud environments and the tracking of the QoS of applications developed according to the microservices logic (Table 14, Table 15).

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Partially covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 14 - CONTRAIL SLA lifecycle coverage

<table>
<thead>
<tr>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
</tbody>
</table>

Table 15 - Contrail Mixed-Cloud models’ coverage

**2.3.3.8 EGI federated Cloud**

The EGI project [58] is a European initiative launched to create a low-cost federated Cloud infrastructure to facilitate the creation of test environment by the research communities. The framework simplifies the SLA negotiation and establishment phases thanks to SLA templates prepopulated with parameters and features supported by the affiliated Cloud providers. The EGI framework relies on two main features to properly manage the SLA lifecycle (Figure 30):

- **Service catalogue**: the EGI project defined a standard Cloud service catalogue categorized according to the guidelines defined by the IT Infrastructure Library (ITIL), the most adopted IT management framework. The items included in the catalogue were defined to cover most of the demand and avoid exceptions and ad-hoc configurations.

- **Federated service management**: the EGI project developed a platform (EGI.eu) to act as a federation layer to filter the services distributed by different Cloud
providers and implement a SLA and OLA framework. The federation layer simplifies the negotiation with the Cloud consumer establishing standard SLA templates based on normalized SLA parameters offered by affiliated Cloud providers. Moreover, to properly track the QoS, the framework establishes also OLA parameters between the CSPs and the EGI foundation.

In summary, the EGI federated Cloud project provides an ecosystem to deliver cost-effective IaaS solutions based on standard configurations and a predefined set of SLA and OLA parameters (Table 16). This project strongly simplified the access to distributed high-computing and high-memory solutions to test machine-to-machine and advanced analytics algorithms (Table 17)

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Partially covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 16 - EGI SLA lifecycle coverage

<table>
<thead>
<tr>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 17 - EGI Mixed-Cloud models’ coverage
2.3.3.9 IRMOS

The IRMOS project [59] developed a framework to simplify the adoption of high-computing and real-time applications through the definition of application and technical SLA parameters. The application SLA can be established between a Cloud consumer and a Cloud service provider while the technical SLA is established among different providers (e.g. infrastructure provider and platform provider).

The framework is based on two main modules (Figure 31):

- **Dynamic SLA Re-negotiation**: this module enables a dynamic negotiation of SLA parameters before and during the execution phase. The re-negotiation can be triggered by the customer to consider additional service requirement, by the Cloud provider to review the original configuration selected by the consumer or by the monitoring system to suggest different scalability rules. To speed-up the negotiation phase, the IRMOS framework developed a SLA engine to automatically review and update the SLA parameters without human intervention.

- **Adaptable monitoring**: this module collects in real-time information from the application and infrastructure layers to compare the actual performance of the services with the expected QoS. To detect SLA breaches caused by performance degradation, the monitoring system translates high-level application...
requirements (e.g. video quality, user experience, etc.) into low-level technical parameters (e.g. response time, availability, etc.)

In summary, the IRMOS framework enables a dynamic SLA negotiation approach to adapt the QoS according to the actual business needs and ensure the best service configuration during the execution phase (Table 18). The framework can easily collect information from hybrid Cloud environments, but it is not able to track service components distributed across different Cloud providers (Table 19).

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 18 - IRMOS SLA lifecycle coverage

<table>
<thead>
<tr>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 19 - IRMOS Mixed-Cloud models’ coverage

2.3.3.10 Q-ImPrESS

The Q-ImPrESS project [60] defined a quality-driven approach to simplify the Cloud service development. The Q-ImPrESS framework developed a service architecture metamodel to simplify the definition of performance indicators and parametric dependencies among services deployed on different Cloud providers (Figure 32).

![Figure 32 - Q-ImPrESS architecture](image-url)
To dynamically track and adapt the SLAs during the service lifecycle, the framework developed a trade-off analysis engine to predict incidents and service’ degradations according to previous monitoring logs. The engine is based on algorithms to perform advanced analytic calculation and real-time analysis to execute preventive activities to ensure the agreed QoS.

In summary, the Q-ImPrEES framework is focused on the SLA negotiation and establishment phases to ensure the QoS of custom applications deployed on public Cloud environments (Table 20). The monitoring engine can be easily deployed on a private virtual solution and interconnected with the control panel of a public Cloud provider but it is not able to track the performance of different service components deployed across several Cloud providers (Table 21).

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 20 - Q-ImPrEES SLA lifecycle coverage

<table>
<thead>
<tr>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 21 - Q-ImPrEES Mixed-Cloud models’ coverage

2.3.3.11 DeSVi

The Detecting SLA Violation infrastructure (DeSVi) [33] defined an advanced monitoring system to track the performance of Cloud services through the real-time measurement of low-level metrics (e.g. availability, downtime period, etc.) and consumer oriented parameters (e.g. response time). The DeSVi architecture relies on the LoM2HiS framework\(^1\) to manage the SLA establishment and monitoring phases and prevent future SLA violation. The LoM2HiS framework [61] is based on two different monitoring systems to track low-level metrics (host monitor) and compare the actual virtual machines performances against the agreed SLA parameters (run-time monitor).

The DeSVi architecture can be summarized towards four main steps (Figure 33)

---

\(^1\) LoM2HiS: Low Level Metrics to High level SLAs
• **Step 1**: the Cloud consumer configures a virtual infrastructure request and starts the negotiation with the Cloud provider to agree on the SLA parameters to match the business needs;

• **Step 2**: the application deployer allocates the virtual instances required to cope the required configuration;

• **Step 3**: the VM deployer and configurator installs the selected operating system (e.g. Windows, Linux, etc.) and configures the virtual machine with the required parameters;

• **Step 4**: the host monitoring tracks the service performance and match the actual QoS with the agreed SLA parameters. In case of incidents or service degradations the VM deployer and configurator can redirect the workload on the redundant pool of resources (if required by the initial configuration).

![DeSVi architecture](image_url)

In summary, the DeSVi architecture simplifies the monitoring of the services’ performance and enables the detection of future SLA violations thanks to the predictive analyses provided by the LoM2HiS framework. Despite the DeSVi architecture covers
most of the SLA lifecycle it is not able to control distributed virtual resources in hybrid Cloud or Multi-Cloud configurations (Table 22, Table 23)

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 22 - DeSVi SLA lifecycle coverage

<table>
<thead>
<tr>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 23 - DeSVi Mixed-Cloud models’ coverage

### 2.3.3.12 CSLAM

The CSLAM framework [62] defined a structured flow to manage the entire SLA lifecycle in Cloud environments. The framework relies on WSLA language to define infrastructure and application performance SLA parameters. The process flow can be summarized in four main steps (Figure 34):

- **Step 1:** in this step, the Cloud consumer starts the negotiation phase sending the business and technical requirements to the Cloud provider. The negotiation iteratively reviews the SLA parameters and generates a digital version of the contract (XML). In case of multiple providers, the negotiation generates one XML file for each provider and a master contract with the high-level SLA parameters split according to the value ensured by the different providers (i.e. the overall...
availability threshold is declined in a sub-parameter to highlight the value granted by each provider);

- **Step 2**: in this step, the digital version of the contract is stored in a dedicated module to provide the parameters to track during the monitoring phase;

- **Step 3**: in this step, the monitoring engine tracks the actual performance of the virtual instances and measures them against the SLA parameters. This activity controls the QoS since the SLA termination or the shutdown of the service;

- **Step 4**: in this step, activated in case of SLA breach, the SLA manager module identifies the source of the fault and defines the best approach to handle it to ensure the business continuity.

In summary, the CSLAM framework enrich the WSLA negotiation language with additional details to speed-up the creation of the contract between Cloud consumers and Cloud provider. Furthermore, a dedicated module enables the automation of the SLA establishment and monitoring phase (Table 24) and the creation of sub-parameters to track the performance of each provider strongly simplify the management of services deployed in hybrid and Multi-Cloud environments (Table 25).

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 24 - CSLAM SLA lifecycle coverage

<table>
<thead>
<tr>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
</tbody>
</table>

Table 25 - CSLAM Mixed-Cloud models’ coverage

**2.3.3.13 C@H system**

The C@H system [63] defined a Cloud brokering architecture to combine resources and services from different private and public Cloud providers. The architecture relies on a virtual provider, namely C@H Cloud provider, that acts as a single interface for Cloud consumers to simplify the SLA management across the entire service lifecycle.

The process flow defined to properly manage the resource aggregation involves three main actors:

- **C@H consumer**: the Cloud consumer configures the required services on the
Cloud broker and starts the negotiation phase with the C@H Cloud providers combining virtual resources from different providers;

- **C@H admin**: the administrator deploys, configures and manages the C@H Cloud provider and selects the IaaS solutions to be activated to satisfy the service requirements;

- **C@H resource owner**: the resource owner can be both a public Cloud provider and the solution owner of a private Cloud service. The resource owner can be additionally categorised as public contributor (i.e. a public Cloud provider) and volunteer (i.e. community of companies or research centres that share their IT resources).

The C@H system architecture relies on different modules and units used to build up the Cloud broker solution (Figure 35):

- **Resource abstraction**: this module implements an abstraction layer to combine infrastructural resources from different providers into a single virtual service;

- **SLA management**: this module facilitates the negotiation of SLA parameters and tracks the actual QoS through the real-time monitoring of the IaaS performance. This module relies on an advance analytic engine to predict incidents and service degradations starting from past logs and reports;

- **SLA resource management**: this module is the core of the C@H architecture. It manages the SLA establishment phase and the activation/de-activation of the virtual resources to satisfy the agreed SLA parameters;
• **Front-end**: this module acts as a unique interface for all the Cloud requests for both C@H users and C@H admin.

In summary, the C@H architecture implements a Cloud broker to aggregate Cloud services from different public and private Cloud providers. The modular architecture supports most of the SLA lifecycle phases and ensure the proper control on hybrid and Multi-Cloud environments (Table 26, Table 27). However, despite the architecture offers a broker as unique entry point for Cloud demand, it does not support the consumers during the service modelling phase.

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 26 - C@H SLA lifecycle coverage

<table>
<thead>
<tr>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
</tbody>
</table>

Table 27 - C@H Mixed-Cloud models’ coverage

### 2.4 Conclusions and validation

The literature review highlighted that SLA management in Cloud environments is still an important research topic. Many government agencies directly managed researches to define standard and best practices to facilitate the adoption of Cloud solutions. Moreover, new studies were recently launched to address cyber security and data privacy issues to ensure the compliance with local and international regulations (e.g. the GDPR in Europe, the Cybersecurity Information Sharing Act in US and the Personal Information Security Specification in China).

Despite the huge amount of studies on SLA management in Cloud environment, just few of them provides clear guidelines and innovative solutions to track and ensure the QoS. Several studies focused the research on the identification of SLA parameters to track the QoS and ensure the proper degree of control on the Cloud services. Most of the studies analyzed the entire IT stack and the IaaS parameters, proving that the control on the underlying virtual infrastructure is crucial to monitor the services’ performance and detect incidents or service degradations.
A clear gap is highlighted by the wide adoption of tailored SLA templates instead of using standard protocols like WSLA and WS-agreement to simplify the creation of hybrid and Multi-Cloud environments. Likewise, the available resource allocation approaches are not able to cover both technical and business parameters, slowing down the adoption of Cloud solutions for business-critical services.

However, the main gap in the literature is the lack of frameworks able to cover the entire SLA lifecycle. Only 4CaaSt, Cloud4SOA and SLA@SOI have a good coverage across the different SLA management activities but they are not fully able to address the service modelling phase and the SLA termination phases (Table 28). Moreover, despite several frameworks are based on SLA@SOI guidelines, the research focus is on the monitoring phase empowerment instead of a wide SLA lifecycle coverage.

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>4CaaSt</td>
<td>Partially covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
<tr>
<td>Cloud4SOA</td>
<td>Partially covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>SLA@SOI</td>
<td>Partially covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>CloudScale</td>
<td>Partially covered</td>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>Cloud-TM</td>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>ETICS</td>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>CONTRAIL</td>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Partially covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>EGI</td>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Partially covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>IRMOS</td>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>Q-ImPrEES</td>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>DeSVi</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>CSLAM</td>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>C@H system</td>
<td>Not covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 28 - SLA lifecycle coverage - Overall comparison

However, a better result is highlighted by the evaluation of the frameworks against their
ability to manage services deployed in hybrid Cloud and in different configurations of the Multi-Cloud model. This result confirms that the adoption of a Multi-Cloud approach will be a growing trend over the next years and validates the research questions. The combination of the comparisons highlights 4CaaSt and Cloud4SOA as the most advanced SLA management frameworks. Surprisingly, despite SLA@SOI is largely considered as de-facto standard and the starting point for several studies, the framework does not provide guidelines for managing services deployed in Multi-Cloud environments (Table 29).

<table>
<thead>
<tr>
<th></th>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>4CaaSt</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
<tr>
<td>Cloud4SOA</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
<tr>
<td>SLA@SOI</td>
<td>Fully covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>CloudScale</td>
<td>Fully covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>Cloud-TM</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
<tr>
<td>ETICS</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
<tr>
<td>CONTRAIL</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
<tr>
<td>EGI</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>IRMOS</td>
<td>Fully covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>Q-ImPrEES</td>
<td>Fully covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>DeSVi</td>
<td>Not covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>CSLAM</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
<tr>
<td>C@H system</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
</tbody>
</table>

Table 29 - Mixed-Cloud models’ coverage - Overall comparison
3 Enterprise survey

Multinational companies are continuously investing on Cloud technologies to streamline IT processes, substantially reduce the time to market of new services and strongly simplify the test of new technologies like Internet of Things (IoT), Machine learning and Big Data. The Cloud strategy of multinational companies evolved towards the years, moving the IT workload from their own data centers to public Cloud providers across the world. The current trends in public Cloud adoption are driving the evolution of technologies and best practices to cope the business demand. For this reason, to complete the analysis of the state of art on Cloud Computing and provide the current enterprise view, a survey was executed on Cloud related key decision makers of over sixty multinational companies. The survey assessed both business and technical aspects of the Cloud strategies to provide a structured landscape of benefits, constraints, and opportunities towards different regions and industries. The results of the analysis were then compared against the key insights extracted by a previous study performed in 2015 to highlight the evolution of the different Cloud strategies [64].

3.1 Study methodology

The literature highlights a structured set of core building blocks to properly drive a quantitative and qualitative analysis on a specific sample [65].

- **Survey purpose**: questions the survey tries to give answer
- **Target population**: description of the survey sample
- **Sample sources**: sources used to profile the sample
- **Data collection**: survey methodology adopted to collect data
- **Data analysis and key insights extraction**: instruments and methodologies used to analyze the data

3.1.1 Survey purpose

The main purpose of the analysis was to investigate the Cloud model adopted by multinational companies and understand the future trends included in the short-term and long-term Cloud strategy. According to the adopted Cloud model, the survey aims then to deep dive the main benefits and constraints related to the usage of private and
public Cloud solutions and analyze best practices and frameworks implemented to optimize the Service Level Management activities.

3.1.2 Target population and sample source

The analysis was focused on multinational companies with a structured vision on Cloud adoption. Thanks to the business network of Business Integration Partners (Bip), 62 multinational companies were identified as candidate for the survey. The sample was selected to ensure a good distribution across different regions and industries and to analyze different Cloud strategies. To prove the value of the sample the overall revenue generated in 2017 was calculated. It achieves more than 4 trillion USD, a valuable amount considering that the top 10 companies ranked by Fortune generated 2 trillion USD. To cover both business and IT details, the analysis involved directly the CIOs, to properly define the Cloud vision, and Cloud architects, to deepen technical aspects (Figure 36).

![Survey approach](image)

Figure 36 - Survey approach

3.1.3 Data collection

The methodologies to collect data and investigate a sample can be summarized in three main approaches:

- **Survey**: this approach is based on a pre-defined structured set of questions (e.g. single and multiple choice, ratings, etc.) to drive the collection of data. A survey is more effectuated when carried-out face-to-face since the interviewer can address eventual questions, but it can be also completed over an on-line form;

- **Interview and focus group**: these approaches are based on a set of open questions or discussion points and they are more effective for qualitative
researches as they facilitate the analysis of people’s opinions and attitudes;

- **Observation**: this approach can be considered as a flipped method compared to the previous ones since the information are extracted by the observer during a period of analysis on the sample behaviors.

The proposed analysis was based on a mixed approach to exploit the benefits provided both by the survey and the interview. The proposed questions were based on proven best practices [66]:

- **Wording of survey questions**: the language used for the questions formulation was clear for the members of the sample and appropriate to the selected topic. No acronyms were used without a clear definition and each question was additionally supported by a structured explanation and useful examples;

- **Questions sequencing**: question sequencing followed a logical order to guide the respondents during the collection of the data and facilitate the extraction of technical details;

- **Appearance**: the survey layout was based on the Bip corporate template and it was used for the entire duration of the interviews. The survey was additionally supported by a non-disclosure agreement (NDA) to ensure the proper degree of privacy to the respondents.

According to these best practices the survey was structured on three main macro-sections and thirteen questions (Table 30).

The first section aimed to profile the different companies according to the location of the head quarter, the industry and the current investments on Cloud solutions. This section was crucial to highlight how the geo-distribution of the main public Cloud providers and the enact of new laws on data privacy can strongly affect the Cloud strategy. The second section was focused on the IT workload distribution across the different Cloud models and the evaluation of the related benefits and constraints. This section aimed to highlight the move of the workload from standard IT and private Cloud solutions to the public Cloud ones, laying the foundation of the Multi-Cloud model. The last section of the survey was defined to deepen tools, approaches and best practices adopted to optimize security and SLM in public Cloud.
### Table 30 - Survey sections and parameters

<table>
<thead>
<tr>
<th>Sections</th>
<th>Parameters</th>
<th>Values and description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Company profile</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Country and region</td>
<td>Head quarter location</td>
<td></td>
</tr>
<tr>
<td>Industry</td>
<td>Standard Industrial Classification (SIC)</td>
<td></td>
</tr>
<tr>
<td>IT and Cloud investments</td>
<td>% of IT investment allocated for Cloud</td>
<td></td>
</tr>
<tr>
<td><strong>Cloud strategy</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current and future Public Cloud model</td>
<td>Usage of a single or multiple IaaS CSPs</td>
<td></td>
</tr>
<tr>
<td>Current and future Private Cloud model</td>
<td>% services running on Cloud (virtual instances) on the entire IT estate</td>
<td></td>
</tr>
<tr>
<td>Cloud penetration</td>
<td>% services running on Cloud on the entire IT estate</td>
<td></td>
</tr>
<tr>
<td>Current and future workload on Public Cloud</td>
<td>% of services running in Public Cloud on the entire Cloud workload</td>
<td></td>
</tr>
<tr>
<td>Current and future workload on Private Cloud</td>
<td>% of services running in Private Cloud on the entire Cloud workload</td>
<td></td>
</tr>
<tr>
<td>Benefits and constraints</td>
<td>Evaluation of the main 5 benefits and constraints on a scale from 1 to 5</td>
<td></td>
</tr>
<tr>
<td><strong>SLM approach</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adoption of CASB (Cloud Access Security Broker) and customer portal</td>
<td>Adoption of a Cloud Access Security Broker and implementation of a web portal to enable the Cloud services delivery</td>
<td></td>
</tr>
<tr>
<td>Monitoring tool integration with CSPs</td>
<td>Integration of the internal monitoring system with the services deployed in Public Cloud</td>
<td></td>
</tr>
<tr>
<td>CMDB integration with CSPs</td>
<td>Virtual instances deployed on Public Cloud tracked in the CMDB</td>
<td></td>
</tr>
<tr>
<td>IT management framework</td>
<td>Preferred framework to manage the E2E service lifecycle</td>
<td></td>
</tr>
</tbody>
</table>

### 3.2 Survey findings

The survey analyzed 62 multinational companies from eleven countries and three regions: Europe, America and Asia (Figure 37). Since the sample is composed by multinational companies with offices, branches and SOHOs distributed across the world, the head quarter location was used to simplify the clustering of the respondents. European companies represent 70% of the sample due to the nature of the business network of Bip and the USA is well represented with ten companies, all included in the Top 500 Fortune. Asian companies represent a small percentage of the sample and a more structured analysis on this area will be managed in future studies.
The analysis of the sample across 8 different industries shows how Cloud computing is playing a significant role in telecommunication companies, where business directly relies on technology and financial institutions, where technology is crucial for real time transaction and secure data management (Figure 38). In a similar way, energy companies are exploiting Cloud capabilities to optimize their operations and streamline business critical processes (e.g. smart metering, drilling, geological mapping, etc.).

The profile based on the number of employees, adopted in many recent researches, is not significant in the analyzed sample that includes only multinational companies. Moreover, the number of employees is not closely related to the size of the IT department or the Cloud model adopted. To properly analyze the sample, the percentage of IT budget allocated to private Cloud and public Cloud projects was selected as primary indicator. According to this approach, the companies were profiled on three different maturity stages related to the spending on Cloud solutions (Table 31).
### Table 31 - Cloud maturity level

<table>
<thead>
<tr>
<th>Maturity level</th>
<th>% of IT budget allocated for Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud beginner</td>
<td>Less than 30%</td>
</tr>
<tr>
<td>Cloud explorer</td>
<td>From 31% to 60%</td>
</tr>
<tr>
<td>Cloud focused</td>
<td>More than 61%</td>
</tr>
</tbody>
</table>

The split by region highlights the large investments done by American companies, while most Asian and European companies are still allocating the majority of IT budget to “standard” IT platforms (Figure 39).

![Figure 39 - Cloud investment by region](image)

The split by industry shows the huge investments done by telco companies which are exploiting their own data centers to directly act as Cloud providers and, moreover, are widely investing in public Cloud services to test new technologies. Energy companies are strongly investing in Cloud solutions to consolidate their data centers and reduce the number of facilities and the amount of legacy systems. An important part of their budget is also allocated to public Cloud to support distributed systems like smart metering services and move the computing power as close as possible to their power stations and points of distribution. In the analyzed sample, media companies are exploring Cloud capabilities to reduce their technological gap from the big on-demand players (e.g. Netflix, Prime Video, etc.) and the time to market for launching new services. Retail, chemical and automotive companies still have small IT departments, often outsourced, and 100% of our sample is still allocating less than 30% of the budget on Cloud solutions (Figure 40). Moreover, companies with small IT departments are still widely facing technical constraints in re-platforming vertical industry solutions and custom systems.
3.2.1 Cloud model adoption

All the surveyed organizations defined a short and long-term Cloud strategy to support the evolution of their business needs. Private Cloud represented the entry point to this new technology as a natural evolution from the legacy virtual farm. The wide supply of private Cloud solutions and the different application generations in the service portfolio of the companies led the evolution towards a multi private cloud model (Figure 41).
trend will grow in the next future, reaching 76% over the next five years. This trend is mainly driven by the low percentage of Cloud ready and Cloud native applications available in the current service portfolios. Different application generations require different effort in re-platforming or upgrading the services to a Cloud oriented version. For this reason, most of the companies invested on different private Cloud capabilities to create a target platform for each application generation. On the other hand, Public Cloud was often confined to test and development environments and non-sensitive data. Hence, even if only 18% of the companies adopted a Multi-Cloud model (i.e. adoption of services from different C providers), the growing request of geo-distributed services led to a turning point. The adoption of a Multi-Cloud model will be an obvious trend over the next years to cope new business requirements and enable geo-resilience and disaster recovery solutions. The split by industry confirms the Multi-Cloud as a growing trend (Figure 42).

![Figure 42 - Multi-Cloud model adoption by industry](image)

All retail, media companies and financial institutions included in the sample already defined the roadmap to adopt a Multi-Cloud model over the next years. The large ICT departments of telecommunication, media companies and financial institutions have
already adopted a Multi-Cloud model to migrate production environments and test new technologies. This approach is strongly supported by the high maturity of the enterprise architecture (EA) of these companies that is fully based on SOA principle and microservices logic. The split by region highlights that top CSPs are not operating on a global scale, thus creating some constraints to Asian business. So far, UAE (and Middle East area in general) can rely only on Alibaba as global CSPs since regulation and low latency requirements inhibit the migration of services outside the country borders. By contrast, in Europe and America, a wide range of CSPs offer Cloud services and facilitate the rapid adoption of a Multi-Cloud model (Figure 43).

3.2.2 Cloud penetration and workload distribution

To provide a clear view on the current Cloud adoption the survey analyzed the percentage of IT workload on Cloud platforms. To facilitate the consolidation of the data extracted from the configuration management database, the analysis considered three different Cloud penetration levels (Table 32):

<table>
<thead>
<tr>
<th>Cloud penetration level</th>
<th>Current Cloud workload</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Less than 20%</td>
</tr>
<tr>
<td>Medium</td>
<td>From 21% to 40%</td>
</tr>
<tr>
<td>High</td>
<td>More than 41%</td>
</tr>
</tbody>
</table>

Table 32 - Cloud penetration level

The analysis of Cloud penetration level shows a slow migration of critical services and sensitive data. Most companies migrated less than 40% of their workload, due to infrastructural constraints and to high costs related to the re-platforming of applications...
to Cloud ready and Cloud Native generations, as required to fully exploit Cloud benefits [67] (Figure 44).

The split by industry shows a close relationship between the importance of the IT for the business of the company and the Cloud penetration. In fact, analysis results are like the ones on the investment rate. Retail, automotive and chemical industries migrated less than 20% of their workload on Cloud due to a fragmented IT infrastructure, stemming from partnerships and acquisitions and to the large presence of legacy and RISC environments. Similarly, financial institutions confined the migration on the Cloud to a limited set of applications, because their business is still strongly based on RISC containers and high-end servers that hardly can migrate on Cloud due the low maturity of the available solutions (Figure 45).
Some energy companies exploited a massive migration of the workload on private and public Cloud to speed-up the data center consolidation projects and reduce the number of facilities, but most of the them limited the migration to the not critical services. Telecommunication companies, supported by huge investments, migrated most of their services on Cloud enabling a full orchestrated and flexible IT infrastructure ready to host innovative services for both the internal use and go-to-market proposition.

The focus on public Cloud workload shows a growing trend but it also confirms a slow adoption in Asia. The absence of global CSPs in Asia impacts the Cloud strategy and slows down the adoption of public Cloud services. So far, only Alibaba Cloud is considered the only CSP able to support an international Cloud strategy. Conversely, European and American companies are supported by several global Cloud providers, allowing the creation of geo-distributed services (Figure 46).

![Figure 46 - Workload in public Cloud by region](image)

The split by industry highlights a wide adoption of public Cloud services by the industries with a limited budget allocated to the Cloud migration. Retail, automotive and chemical industries will exploit the low initial investment to migrate existing services and test new applications. In the same way, the big ICT departments of telecommunication, media and energy companies will increase the workload in Public Cloud to streamline IT processes and support peaks of demand (Figure 47). This trend raises a financial impact since the “as-a-service” model massively increases the amount of operating expenses (OpEx) in their income statement. Moreover, the business case for the public Cloud adoption clearly highlights that multinational companies should introduce capacity optimization tools to manage demand peaks and avoid waste of IT resources and the related increase of costs.

81
To properly analyze the migration to public Cloud, the survey considered the split between production and non-production workload (e.g. test and development). The percentage of production workload migrated to public Cloud is a good indicator of Cloud maturity (Figure 48).

Figure 47 - Workload in public Cloud by industry

Telecommunications and media companies migrated about 25% of their production workload to public Cloud.
workload to public Cloud and this indicator will reach 30% in forthcoming years thanks to IoT and streaming services. Retail industry will be characterized by a huge adoption of public Cloud services, doubling the current percentage of production workload. Conversely, high tech, chemical, automotive companies and financial institutions will strongly limit the migration of production workload to public Cloud due to the business criticality of the data managed (e.g. patents).

### 3.2.3 Main benefits and constraints

To evaluate the perception of the business impact provided by the adoption of public Cloud services, the survey evaluated benefits and constraints on a Likert scale from 1 to 5 [68]. Most companies selected the faster time-to-market as the main benefits ensured by public Cloud. CPSs enable the users to configure their infrastructure by selecting standard and fully compatible components, thus autonomously deploying virtual servers in few minutes (Figure 49). Similarly, the sample deeply appreciated the scalability and the availability of the virtual infrastructure which enable to serve the demand peaks. Interestingly, economic and organizational benefits got a low rating, probably because they require a high maturity level to be positively perceived [69]. In fact, the public Cloud business case is negative after about 3 years of constant utilization but can be strongly positive by adopting an effective disposal strategy and turning-off virtual resources when not used. Likewise, the adoption of public Cloud services can reduce manual tasks and increase IT staff efficiency, but it requires a structured re-organization of ICT department.

![Figure 49 - Main benefits of Public Cloud adoption (average values)](image)

The assessment of the constraints shows, interestingly, that the three highest rates are related to SLM. Despite the maturity achieved by Cloud solutions and the security certifications obtained by the main CSPs, security still represents the main barrier to
adopt public Cloud services for critical applications and sensitive data. Likewise, the migration of data protected by local regulations and the difficulty of auditing public Cloud environments strongly reduce the percentage of production workload and limit the number of suitable CSPs. Another important constraint is the lack of a direct control on deployed services, since tracking QoS in public Cloud environments is a complex task and requires huge integration investments. Economic and organizational constraints obtained a low score since CSPs can ensure high transparency and skills on Cloud capabilities are available on the market (Figure 50).

![Figure 50 - Main constraints to Public Cloud adoption (average values)](image)

### 3.2.4 Approach to the Service Level Management

The survey evaluated the SLM approach towards a set of different parameters. The first part of the analysis was focused on the on-boarding of useful information through the integration of the internal monitoring tools with the one provided by the CSPs and the tracking of the deployed virtual services in the CMDB (Figure 51).

![Figure 51 - Monitoring tool (a) and CMBD (b) integration](image)

On one hand, about half of the companies still controls the service performance only through the monitoring tools available on the administration panel of the CSPs. This
decision limits the detection of the actual QoS or any SLA breaches. On the other hand, 77% of companies have integrated Public Cloud services in the CMDB to track deployed virtual servers and avoid shadow IT [64]. The survey analyzed also the adoption of brokering systems for autonomously deploying virtual servers and implementing structured security policies. 37% of companies, mainly telecommunication companies and financial institutions, already adopted a Cloud Access Security Broker (CASB), thus proving the criticality of managing security in Public Cloud. By contrast, only 19% of companies implemented a customer portal to enable their own business lines to autonomously deploy virtual servers. This result proves that ICT departments are still centralizing governance, also for better tracking the QoS delivered to their business customers (Figure 52).

![Customer portal and CASB adoption](image)

To highlight the perception of the importance of these tools, the survey compared the adoption status with the Cloud investment rate. The integration of the public Cloud services in the CMDB is a clear priority for the companies to track the full IT estate and the related average usage. Similarly, the control of the QoS through the internal monitoring tool is crucial to cross-check the performance of the services guaranteed by the CSPs. For this reason, even if the current adoption is strongly limited, most of the companies planned for 2018 the integration between their monitoring system and the one provided by the CSPs.

Although the adoption of a CASB strongly empowers the security of services, it also limits the companies’ budget to the Cloud strategy. In a similar way, most of the companies consider the customer portal as a low priority due to the high degree of automation required and the potential lack of control on the budget consumption (Figure 53).
The last part of the survey was focused on the adoption of IT management frameworks to optimize the degree of control on the virtual infrastructure and the services running on top of it [70, 71]. Most companies control Cloud services through ITIL v3 and COBIT v4, frameworks still based on the possibility to directly interact with the IT systems and, for this reason, not conceived for public Cloud. Only 37% of companies adopted ISO 20000:9 to specify requirements for the CSP and to track the QoS against them. ISO 20000 is currently the only recognized standard able to provide a set of guidelines and use cases to map the business requirements in SLA parameters to be included in the contract. The fragmentated scenario shows that most companies are still managing Cloud platforms though a framework conceived for an in-house infrastructure. However, both ITIL and COBIT have defined a new release to better support services deployed on public Cloud [72] (Figure 54).


3.3 Conclusions and validation

All the surveyed organization already defined a short-term and long-term Cloud strategy to support the business requirements and streamline the IT tasks during the whole service lifecycle. Telecommunication, energy companies and financial institutions invested a significant part of their budget on Cloud services to test innovative technologies (e.g. Big Data, IoT, M2M and on-demand services) [73]. The growing Cloud supply on the market and the high maturity of the main CSPs are pushing companies toward a Multi-Cloud strategy. The growing request of geo-distributed services and the willingness to avoid vendor lock-in have convinced the companies in selecting multiple public Cloud providers to address different business requirements. Telecommunication, media companies and financial institutions will strongly invest on a Multi-Cloud strategy to provide services as close as possible to their customers [74, 75]. However, regulatory constraints and the lack of global Cloud providers are strong limits for companies operating in Middle East. The analysis of the Cloud adoption shows that Cloud has a constant growth trend but it is still affected by the barriers identified in a similar research executed in 2015 [76]. Security, regulatory compliance and the difficult control of the QoS are still confining Public Cloud to non-business critical applications and non-sensitive data. To address these critical points, multinational companies are investing in the implementation of systems to facilitate the SLM in Multi-Cloud environment, proving the importance of the research topic. Moreover, a Cloud oriented version of the main IT management frameworks will be released in 2018 to facilitate the service level management activities across the entire services lifecycle.
4 LISA - Lean Information Service Architecture

The key insights extracted by the systematic literature review and the results highlighted by the survey clearly show a gap in the current Cloud management techniques. From a consumer viewpoint, the dynamism of the systems’ requirements and the distribution of computing resources across different locations strongly affect the ability to effectively track the QoS. Moreover, from a provider viewpoint, the compliance with multiple SLA parameters requires a QoS-oriented monitoring on each Cloud component [77]. To overcome these problems, the research developed a SLA-aware Lean Information Service Architecture (LISA) able to manage Cloud systems in unpredictable conditions and to support the consumers during the entire services lifecycle (Figure 55).

4.1 Cloud service and SLA management lifecycle

LISA Multi-Cloud framework aims to support Cloud consumers during the entire service lifecycle, acting as an advanced ecosystem and unique entry point for Cloud services configuration, deployment and management. To ensure the service lifecycle coverage, the activities supported by LISA Multi-Cloud were mapped on the main ITIL phases (Figure 56).

**Figure 56 - LISA activities mapping on ITIL phases**

**Service design**

This step aims to define the features and the logics to be included in the Cloud services (e.g. infrastructure configuration, security policies, expected QoS, etc.). LISA framework
includes a self-service online catalogue to address both technical and business requirements and proposes the best Cloud providers and the main SLA parameters to be included in the contract. After the service selection, the framework supports the customers during the negotiation phase, creating a digital version of the contract.

**Service transition**
This step aims to control the service implementation on the selected Cloud solutions. The framework uses open access APIs to automate the delivery of the services on private and public Cloud environments and can be easily interconnected with the configuration management database (CMDB) to ensure the proper degree of control on the virtual IT estate.

**Service operation**
This step aims to manage the services during the execution phase to ensure the QoS. The framework provides an advanced SLA-aware resource allocation technique to optimize the service performance and ensure cost-effectiveness. Moreover, the analysis of log generated by the monitoring tools are used to identify critical behaviors and predict failures or service degradation.

**Service improvement**
This step aims to execute corrective actions to optimize the services configuration and dispose them in case termination requests. The framework offers a centralized management console to change the virtual machines configuration, shutdown the services and delete them. To optimize the cost-effectiveness the management system offers the possibility to schedule activities and automate these processes.

### 4.2 QoS metrics
LISA framework relies on a set of QoS metrics to properly track the performance of services deployed in Multi-Cloud environments.

The list of different SLAs are mapped on the following array, where \( n \) represents the total number of parameters:

\[
\text{List of SLA} = < s_1, s_2, s_3, \ldots, s_n >
\]

The SLA parameters managed by the framework are mainly focused on the business continuity and they are natively offered in the basic support of all the major Cloud providers.
Mean Time Between Failure (MTBF) = \frac{\text{Total service uptime}}{\# \text{ breakdowns}}

Mean Time To Repair (MTTR) = \frac{\text{Total service downtime}}{\# \text{ breakdowns}}

Availability = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}

Reliability = \text{MTBF} + \text{MTTR}

To properly track the QoS, the framework considers as system breakdowns, both incidents and service degradations:

Incident rate (FR) = \sum_{i=1}^{n} \left( \frac{\text{Incident} (s_i)}{n} \right)

Service degradation rate (SD) = \sum_{i=1}^{n} \left( \frac{\text{Service degradation} (s_i)}{n} \right)

Since an incident or a service degradation do not automatically imply a SLA breach, the framework assigns \( s = 0 \) in case of SLA satisfaction or \( s = 1 \) in case of violation. In addition, to calculate the SLA violation rate, the framework assigns a weighted parameter for each SLA \( (w_i) \):

\text{SLA violation rate (SVR)} = \text{FR} \times \sum_{i=1}^{n} (w_i)

To evaluate the service performance and identify incidents and services degradations, the framework considers the execution time as the difference between the workload completion time \( (WC_i) \) and the workload request time \( (WR_i) \). To cover all the potential scenarios, the execution time formula considers also the time to restart the node \( (\Delta t_i) \) in case of failure:

\text{Workload execution time (E_i)} = \sum_{i=1}^{n} \left( \frac{WC_i - WR_i}{n} \right) + \Delta t_i

The execution time is used to identify workloads with an overall completion time higher than the expected deadline \( (Ed_i) \):
Workload deadline \((Wd_i)\) = \(\sum_{i=1}^{n}\left(\frac{Ed_i}{Et_i} - 1\right)\)

According to the SLA violation rate and the execution time formula, the framework calculates the SLA cost:

\[
\text{SLA cost (SC}_i\text{)} = \text{SVR} \times \sum_{i=1}^{n}(\text{Et}_i)
\]

Finally, the framework calculates the overall execution cost and eventual penalties to evaluate the effectiveness of the workload:

\[
\text{Workload cost} = \text{Et}_i \times \text{resource price}
\]

\[
\text{Penalty cost} = \sum_{i=1}^{n}(\text{PC}_i)
\]

The overall Cloud service cost is calculated from a consumer viewpoint, considering penalties as positive values since they will be refunded in the next billing cycle:

\[
\text{Cloud service cost} = \text{Workload cost} - \text{Penalty cost}
\]

### 4.3 LISA mapping on OASIS TOSCA

LISA framework ensures the QoS management across the entire IT stack, managing the SLA parameters both at infrastructure and application levels. To ensure the efficient management of Multi-Cloud environments and facilitate the creation of interoperable services, LISA framework adopted the Topology and Orchestration Specification for Cloud Applications (TOSCA) [78, 79]. This open access specification language relies on XML 1.0 and provides extension mechanism to include additional domain-specific information. OASIS TOSCA specification language describes Cloud service components and their relationship using templates and plans to facilitate the orchestration processes. The templates describe the structure and the features of a Cloud service while plans define the processes to start, stop and manage the Cloud service during the execution phase. Moreover, the topology includes node templates and relationship templates to design the service architecture as a tree graph. All the nodes of the graph define the properties and the operations available to manage the related Cloud component (Figure 57). According to this logic, TOSCA can be used during the application design phase to describe the relation among Dockers, virtual instances and service components,
facilitating the portability on different Cloud platforms.

Moreover, TOSCA enables the decoupling of the application requirements from the features offered by the underlying virtual infrastructure capabilities. The automatic matching and optimization layer, included in TOSCA orchestration, configures the connectivity with different Cloud providers, enabling the creation of Multi-Cloud environments and reducing the risk of vendor lock-in (Figure 58).
4.4 LISA architecture and process flows

LISA framework is composed by six building blocks interconnected among themselves to create a unique Cloud management engine and cover the entire Cloud service lifecycle. The architecture is based on customized components (colored in green) defined by previous studies, including new features to overcome the gaps highlighted by the literature review (Figure 59) [80-82]. Moreover, LISA includes brand new Cloud-oriented components (colored in blue) to ensure an optimized SLA management and ensure the novelty of the research. Thanks to the adoption of OASIS approved standards, LISA can be easily integrated with TOSCA oriented orchestration tools and connected with operations systems through APIs [83].

LISA includes a Cloud service catalogues to facilitate the selection of services from different providers and support the configuration of Cloud solutions according to both technical and business requirements. In case of lack of configuration able to satisfy the service requirements, Cloud consumer can start a negotiation phase to create a tailored Cloud solution. The service negotiation agent facilitates the communication between Cloud consumer and Cloud providers, supporting the different phases of the deal till the definition of a final agreement (i.e. request approval or rejection) [84]. After the
configuration selection, the catalogue registers the Cloud service in the CMDB and triggers the orchestrator to deliver the resources required to execute it. During the execution phase the service is monitored in real-time to ensure the proper control on the QoS and measure the resources consumption. In case of incident or service degradation, the advanced monitoring system triggers the autonomic resource allocation engine to analyze the service behavior, plan a corrective action and execute it, ensuring the business continuity and the compliance with the agreed SLAs. The resource allocation engine can provision additional resources to manage peak of demand and release them once the workload decreases to ensure cost-effectiveness. The details on the service performance and the resources consumption are stored in the knowledge base to facilitate data analysis activities. Finally, the reporting agent can be configured to create synthetic dashboards and calculate the amount of the next invoice to ensure transparency to Cloud consumers.

4.4.1 Knowledge base

LISA framework continuously interacts with a centralized knowledge base to store and retrieve information required to manage Cloud services across the entire lifecycle. The knowledge base is composed mainly by three different data clusters: contractual information, deployment information and aggregated information (Figure 60).

![Knowledge base high-level structure](image)

Contractual details are stored in the knowledge base after the service selection to enable an effective service management and the correct execution of the other LISA framework components:

- **Service configuration**: this cluster contains information about Cloud services in
terms of virtual instance configuration (i.e. CPU, RAM and Storage), operating system, data base and additional features. Moreover, this cluster contains information about the active nodes, the redundancy configuration and the scalability approach selected to ensure the business continuity;

- **Service cost**: this cluster contains information on the unit cost of the service and the pricing model selected during the configuration phase (i.e. on-demand or reserved service);

- **QoS metrics**: this cluster contains information on the selected performance parameters to be measured by the service monitoring engine;

- **SLA parameters**: this cluster contains the SLAs agreed during the negotiation phase, acting as minimum threshold to measure the performance of the Cloud provider.

Deployment details are updated according to the information registered by the monitoring tools to facilitate the proper control on the service components:

- **Service location**: this cluster contains information on the service nodes deployed on the different Cloud regions, enabling the geo-distribution of the workload;

- **Resource consumption**: this cluster contains information about the actual amount of resources used during a specific time-frame. This value is used to estimate the amount of the next invoice to ensure transparency to the Cloud consumers;

- **Service performance**: this cluster contains information registered by the monitoring tool about the actual QoS, enabling a real-time comparison with the SLA parameters;

- **Incident and downtime**: this cluster contains information on events with a negative effect on the service performance. The monitoring system updates this cluster in case of service performance below the agreed SLA values, measuring the duration of the performance issue to facilitate the calculation of the penalties.

Additional details are stored in the knowledge base by other LISA framework components, aggregating the information to create business-oriented documents.

- **KPI reporting**: this cluster contains information about the service usage and
related performance to facilitate the creation of reports and real-time dashboards;

- **Workload trends:** this cluster contains information on the workload trends and the frequency of peaks of demand to enable the preventive allocation of additional resources to ensure the business continuity;

- **Penalty and refund:** this cluster contains information on the penalties cumulated by the Cloud provider during a specific time-frame, measuring the impact of SLA breaches and the duration of performance degradations. The information stored in this cluster is used by the reporting agent to calculate the refund expected in the next billing cycle;

- **Spending and invoice:** this cluster contains information on the expected spending related to the usage of Cloud services during a specific time-frame. The overall cost is then decreased by the value of penalties cumulated by the Cloud provider during the same period.

### 4.4.2 Cloud service catalogue

The Cloud service catalogue acts as a single point of contact to configure, deliver and manage Cloud services. The catalogue is based on the Universal Description, Discovery, and Integration (UDDI) standard [85], enabling the creation of a XML-based registry to store the Cloud services and specify the related features (Figure 61).

![Figure 61 - Cloud service catalogue - Process flow](image-url)
The Cloud service catalogue simplifies the design phase, providing both technical and business-oriented approaches to identify the best configuration to satisfy the service requirements. The technical view offers a set of pre-approved configurations defined to cover most of the Cloud demand, limiting the exception requests (Figure 62). This view allows technical users to configure the service toward the entire IT stack, selecting the virtual instance configuration (i.e. CPU, RAM and storage), the operating system and the SLA parameters clustered in different service class (i.e. basic, business standard, business premium and mission critical). Moreover, the user can select the data center regions and the related network configuration to deploy geo-redundant solutions and enable the creation of Multi-Cloud environments.

![Cloud Service Catalogue](image)

**Figure 62 - Cloud service configuration - Technical view**

The business view supports users with no technical skills during the configuration of the Cloud service, automatically proposing the best configuration according to the business requirements (Figure 63). The mapping between the service requirements and the available virtual instance configurations is based on the maximum number of concurrent users and the use cases defined for the business service (e.g. high computing, high memory, low network latency, etc.). Moreover, to facilitate the definition of the best service set-up, the users can consult a list of certified applications releases already linked...
to specific configurations. Finally, to ensure compliance with security policies and international regulations, the configuration system requires additional information on the data confidentiality (e.g. financial data, citizen data, etc.) and the business criticality of the service to estimate the impact in case of incidents or service degradations.

Figure 63 - Cloud service configuration - Business view

### 4.4.3 Service negotiation agent

In case of lack of configurations able to satisfy the service requirements, Cloud consumers can directly request an exception through the service catalogue. This activity is supported by the service negotiation agent, allowing the users to request tailored configurations on both private and public Cloud. The private Cloud service negotiation follows a standard approval process and requires both technical and financial feasibility (Figure 64).

In case of approval the configuration is deployed as exception and registered in the CMDB to analyze the frequency of this request and evaluate its insertion in the catalogue as standard configuration. The public Cloud service negotiation requires a more structured process and involves different stakeholders across the organization. The required configuration is submitted to different Cloud providers both as plain text and XML file to speed-up the feasibility evaluation.
The negotiation phase relies on an iterative process based on five different status (Figure 65):

- **Request accepted**: one or more Cloud providers accept the exception request and submit a proposal to the Cloud consumer;
- **Request rejected**: all the Cloud providers refuse the creation of the requested configuration;
- **Configuration approved**: the service catalogue administrator approves one or more proposals received from the Cloud providers;
- **Configuration rejected**: the service catalogue administrator rejects all the proposals received from the Cloud providers. The administrator can ask for a configuration update, highlighting the reason of the rejection;
- **Service selection**: Cloud consumer selects one of the approved configurations and accepts the contract proposed by the Cloud provider.

It is important to consider that only few Cloud providers accept the definition of tailored Cloud services, limiting the negotiation to the price discount.
4.4.4 Advanced monitoring engine

The advanced monitoring engine aims to track the actual performance of Cloud services, analyzing in real-time the information captured by sensors and probes. To enable an efficient runtime monitoring, a QoS agent is installed on each service node [86]. Moreover, thanks to a direct API connection, the monitoring engine analyses the QoS of public Cloud services with no impact on their performance. According to the QoS parameters managed by LISA framework, the monitoring engine evaluates the services performance to identify SLA violations (i.e. incidents and service degradations) and workload trends. These details are communicated to the autonomic resource allocation engine to plan and execute corrective actions to ensure the business continuity and the compliance with the agreed SLA parameters.

The advanced monitoring engine is based on active clustering algorithms [87] and configured with tailored parameters to analyze the QoS and the resource capacity during the workloads execution ($W_A$). In case of QoS below the agreed SLA values ($SLA_{TH}$), the monitoring engine generates an alert and triggers the autonomic resource allocation engine (Table 33).

<table>
<thead>
<tr>
<th>Start</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workload Queue: $W_Q = &lt; W_1, W_2, W_3, ..., W_n &gt;$</td>
</tr>
<tr>
<td>Add Workloads: $W_A = &lt; W_1, W_2, W_3, ..., W_o &gt;$ where $o \leq n$</td>
</tr>
<tr>
<td>Allocate resources to workloads based on QoS requirements for all workloads ($W_A$), identify SLA violation</td>
</tr>
<tr>
<td>if ([QoS &gt; $SLA_{TH}$] = ‘TRUE’) then</td>
</tr>
<tr>
<td>Continue execution</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>Generate alert</td>
</tr>
<tr>
<td>end if</td>
</tr>
<tr>
<td>end for</td>
</tr>
</tbody>
</table>

Table 33 - Monitoring algorithm - SLA violation

Likewise, in case of resource consumption (RC) above the defined threshold ($RC_{TH}$), the monitoring system generates an alert and triggers the autonomic resource allocation engine to allocation additional resources (Table 34).
Start  
Workload Queue: $W_Q = < W_1, W_2, W_3, \ldots, W_n >$

Add Workloads: $W_A = < W_1, W_2, W_3, \ldots, W_o >$ where $o \leq n$

Allocate resources to workloads based on QoS requirements for all workloads ($W_A$), identify Resource overloading

if ([RC < RC$_{MAX \ TH}$] = = 'TRUE') then
if ([RC > RC$_{MIN \ TH}$] = = 'TRUE') then
Continue execution
else
Generate alert
end if
end if
end for

Table 34 - Monitoring algorithm - Resource scalability

The capacity threshold can be set by Cloud consumers during the design phase and modified according to the workload trends registered during the execution phase.

### 4.4.5 Autonomic resource allocation engine

The autonomic resource allocation engine is the core of LISA framework and manages the analyze, plan and execution phases. This component is activated by the monitoring system through the generation of alerts and analyses the status of all the active nodes to identify the reason of the incident and the impact on the Cloud service (Figure 66)

In case of SLA breach the system generates an alert and registers the violation in the knowledge base, adding a time-stamp to the event to track the duration of the SLA violation and allow the calculation of penalties (Table 35).
for all node \([\text{Node}_1]\)
if \([\text{QoS} < \text{SLA}_{TH}] = \text{‘TRUE’}\)
do
Set status \([\text{Node}_1] = \text{Impacted}\)
if \([\text{RC} > \text{RC}_{TH}] = \text{‘TRUE’}\) then
Add new node \([\text{QoS} > \text{SLA}_{TH}]\)
else
Add It into Execution Time
Redirect workload and restart the Node \([\text{Node}_1]\)
if \([\text{Node}_1] = \text{‘RESTARTED’}\) then
Check Node status
if Node status \([\text{Node}_1] = \text{‘ACTIVE’}\)
Remove the node and Generate Alert [Node is declared as Dead]
end if
end if
end if
end if
end for

Table 35 - Plan and execute algorithm - SLA violation

To reduce the impact on the business continuity, the resource allocation engine analyses the percentage of resources used by the current workload \((\text{RC})\) to evaluate the scalability on the same virtual machine (i.e. vertical scalability) or clone the instance (i.e. horizontal scalability). In case of resource consumption below the threshold, the engine redirects the workload on a different node and restarts the one affected by performance issue (Figure 67).

![Figure 67 - Autonomic resource allocation engine - Scalability process flow](image)

Once the node is restarted, the monitoring system check the status of the node to ensure
the service stability and remove it in case of additional performance issue. Moreover, to ensure the cost-effectiveness of the Cloud services, the autonomic resource allocation engine dynamically allocates additional resources in case of peak of demand and releases them once the workload decreased (Table 36).

```plaintext
for all node [Node_i]
    if ([RC > RC_{MAX_TH}] = = ‘TRUE’) then
        Generate alert and Add new Node (QoS > SLA_{TH})
    else
        Generate alert and remove Node (QoS > SLA_{TH})
    end if
end if
```

Table 36 - Table 37 - Plan and execute algorithm - Resource scalability

### 4.4.6 Reporting agent

The reporting agent facilitates the creation of synthetic dashboards to ensure visibility on the service performance. The agent elaborates information stored by other LISA framework components to create reports and estimate the cost to be paid in the next billing cycle (Figure 68).

The reports can be easily customized by Cloud consumers, selecting different parameters to calculate technical and business-oriented KPIs. In addition, Cloud consumers can control and manage the status of the virtual instances directly from the control panel.
included in the service catalogue (Figure 69).

Figure 69 - Reporting agent - Control panel view

The invoice calculation is a useful feature to ensure the proper transparency to Cloud consumers, allowing the cross-check with the invoice sent by the Cloud provider. Moreover, the forecasting model facilitates the allocation of budget to support the Cloud service execution and a structured financial planning during the whole fiscal year (Figure 70).

Figure 70 - Reporting agent - Cloud spending view
4.5 LISA SLA lifecycle coverage evaluation

To prove the novelty proposed by LISA framework, the research mapped it against the model used to evaluate the existing frameworks extracted by the literature review. LISA ensures the complete coverage of the activities required towards the SLA management lifecycle, acting as a centralized and automated Cloud management system (Table 37).

<table>
<thead>
<tr>
<th>Service modelling</th>
<th>SLA negotiation</th>
<th>SLA establishment</th>
<th>SLA monitoring</th>
<th>SLA reporting</th>
<th>SLA termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
</tbody>
</table>

Table 37 - LISA SLA lifecycle coverage

The adoption of open access and standard APIs facilitates the integration with the existing operations tools (i.e. CMDB, monitoring and ticketing systems, etc.) and enables the complete control on virtual instances deployed across different private and public Cloud providers (Table 38).

<table>
<thead>
<tr>
<th>Hybrid Cloud</th>
<th>Redundant Multi-Cloud</th>
<th>Composite Multi-Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully covered</td>
<td>Fully covered</td>
<td>Fully covered</td>
</tr>
</tbody>
</table>

Table 38 - LISA Mixed-Cloud models’ coverage

Moreover, LISA aims to act as a unique ecosystem to configure, deploy and manage private and public Cloud services, extending the features of the framework beyond the standard MAPE-K control loop coverage (Figure 71). In fact, the framework provides an on-line service catalogue to configure Cloud services, negotiate tailored SLA parameters and offers an accounting agent to calculate the expected value on the next invoice, facilitating the cross-check with the amount exported by the Cloud providers.

Figure 71 - MAPE-K adaptation control loop - LISA framework view
5 LISA framework implementation and test

The framework was developed as a Python module of Cloudify orchestrator and implemented in the Innovation Lab of a global telecommunication company. In addition to the autonomic resource allocation engine, the experimental implementation included also an online customer portal to enable the self-service configuration and the management of the deployed virtual instances (i.e. power-on, power-off, delete, clone, migrate, etc.).

5.1 Main resource management techniques

The performances of LISA framework were compared to best in class resource allocation techniques to prove its effectiveness and highlight the innovative features included. The literature review identified four main SLA-aware allocation techniques to be used as benchmark on the selected QoS metrics:

- **Partial Utility-driven Resource Scheduling (PURS):** the framework enables an elastic SLA and pricing negotiation to exchange resources among virtual instances [88, 89]. Furthermore, to ensure cost-effectiveness the framework manages the overbooking of physical resources till an acceptable level of performance degradation;

- **SLA-aware Autonomic Management of Cloud Resources (STAR):** the framework is mainly focused on the reduction of the SLA violation rate to ensure QoS in public and private Cloud environments [90]. The framework manages the resources allocation at two different levels (i.e. global and local) to ensure the proper degree of control on the Cloud service components. STAR represents the last evolution of two previous frameworks mainly focused on energy efficiency (SOCCER and EARTH) [91, 92] and cyber security attacks prevention (CHOPPER) [93];

- **QoS based Resource Provisioning and Scheduling (QRPS):** the framework aims to optimize the resource provisioning through a set of workload patterns. The framework relies on the k-means clustering algorithm to assign a weighted value for each workload and allocate virtual resources to ensure the QoS [94];

- **Autonomic Resource Contention Scheduling (ARCS):** the framework relies on a
structured algorithm to reduce contention of resources shared among different and simultaneous workloads. The ARCS architecture is composed by four components to manage each virtual component from the queuing of jobs till the related resource assignment [95].

The comparison among the selected resource management techniques highlights that self-optimization and incidents detection are basic features, covered by the entire sample. In the same way, the comparison shows a clear gap on the service degradations detection, only LISA correctly manages this feature (Table 39).

<table>
<thead>
<tr>
<th></th>
<th>Self configuration</th>
<th>Self optimization</th>
<th>Incidents detection</th>
<th>Degradations detection</th>
<th>Self healing</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCS</td>
<td>Not covered</td>
<td>Covered</td>
<td>Covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>QPRS</td>
<td>Not covered</td>
<td>Covered</td>
<td>Covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>PURS</td>
<td>Not covered</td>
<td>Covered</td>
<td>Covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>STAR</td>
<td>Not covered</td>
<td>Covered</td>
<td>Covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>CHOPPER</td>
<td>Covered</td>
<td>Covered</td>
<td>Covered</td>
<td>Not covered</td>
<td>Covered</td>
</tr>
<tr>
<td>SOCCER</td>
<td>Not covered</td>
<td>Covered</td>
<td>Covered</td>
<td>Not covered</td>
<td>Not covered</td>
</tr>
<tr>
<td>LISA</td>
<td>Covered</td>
<td>Covered</td>
<td>Covered</td>
<td>Covered</td>
<td>Covered</td>
</tr>
</tbody>
</table>

Table 39 - Resources management techniques comparison

5.2 Experimental environment configuration

The experimental configuration relied on a multiple deployment across three private data centers (Milan, Düsseldorf and Dublin) and two AWS and Azure regions (i.e. Germany and Ireland). The private Cloud deployments adopted two different solutions to test LISA framework in different environments. The experiment considered OpenStack as preferred solution for high-computing uses cases to exploit the scalability of the platform [96] and the Kubernetes as preferred solution for custom applications and web sites to enable continuous delivery and continuous deployment approach [97].

LISA framework was deployed as Python module of Cloudify orchestrator [98] and connected through APIs to Cloud solutions and existing IT management systems. All the existing resources allocation techniques used as benchmark were imported as configuration parameters of the Cloud orchestrator.
5.2.1 Multi-Cloud environment configuration

The experimental configuration selected only Cloud oriented applications to ensure full compatibility with the selected Cloud solutions [99, 100]. These application generations provide important features to properly measure the effectiveness of LISA framework and enable the control on the full IT stack (Figure 72):

- **Autoscaling**: the application identifies in real-time the amount of virtual resources available to execute the workload and ensure the QoS;
- **Self-healing**: the application is designed to tolerate components failure and auto-repair itself in case of incident;
- **Full automation**: the application maintenance and upgrade can be easily automated through scheduled activities managed by the Cloud orchestrator.

![Figure 72 - Application generations mapping on Cloud solutions](image)

To simplify the communication among different Cloud solutions and enable the creation of a hybrid and Multi-Cloud environment, the experimental environment adopted the available Software Defined Network (SDN) configuration [101]. The network configuration included three different connectivity layers (Figure 73):

- **SDN core**: this network layer enables the network virtualization and a centralized network management in each private data center;
- **SDN backbone**: this network layer enables the connectivity among private data centers and facilitates the service migration or replication across different region;
- **SDN external Cloud**: this network enables the secured and dedicated
connectivity between private data centers and public Cloud providers (i.e. AWS Direct Connect and Microsoft Express Route). This network, based on VPN technologies, is also used to interconnect different Cloud providers and enable the creation of a Multi-Cloud environment.

![Diagram of experimental environment network topology]

**Figure 73 - Experimental environment network topology**

### 5.2.2 LISA deployment configuration

The experimental Cloud environment configuration is composed by three different layers to ensure the proper degree of control on each component (Figure 74).

The presentation layer consists in a web portal to request and manage public and private Cloud solutions. To ensure the best user experience, the web portal enables the users’ identification through LDAP protocol and considers three different classes of users:

- **Requestor**: this profile is assigned to users enabled to request the deployment of Cloud solutions. The users included in this cluster belong both to business and technical departments to enable the self-service deployment and IT consultancy services;

- **Approver**: this profile is assigned to users enabled to approve the spending for Cloud service deployment. The users included in this cluster are top managers of different business and IT departments;
- **Administrator**: this profile is assigned to users enabled to create, manage and delete other accounts. The users included in this cluster belong to the Cloud Centre of Excellence and they oversee the entire users’ lifecycle.

The orchestration layer represents the core of the Cloud management system and enables a structured control on each Cloud component. The entire layer relies on Cloudify orchestrator, deployed in each data center to ensure low network:

- **Service catalogue**: this component offers a set of pre-approved virtual instances configurations in terms of vCPU, RAM, storage and operating system. The requestor can configure a business service and deploy it on the selected location and Cloud solution without any kind of human interaction. The standard configurations are selected to cover most of the demand, avoiding exception requests and the related architectural design. A standard amount of gigabyte is assigned to all the configurations to ensure the execution of the operating system and additional space can be required through the portal on NAS and object storage (Table 40). The algorithm embedded in LISA framework allows the configuration of Cloud services through business specifications, without
requiring any technical skills and enable the self-service logic;

<table>
<thead>
<tr>
<th>Configuration</th>
<th>vCPU</th>
<th>vRAM</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1.Small</td>
<td>1</td>
<td>2</td>
<td>70</td>
</tr>
<tr>
<td>G1.Small</td>
<td>1</td>
<td>4</td>
<td>70</td>
</tr>
<tr>
<td>C1.Medium</td>
<td>2</td>
<td>4</td>
<td>70</td>
</tr>
<tr>
<td>G1.Medium</td>
<td>2</td>
<td>8</td>
<td>70</td>
</tr>
<tr>
<td>M1.Medium</td>
<td>2</td>
<td>16</td>
<td>70</td>
</tr>
<tr>
<td>C1.Large</td>
<td>4</td>
<td>8</td>
<td>70</td>
</tr>
<tr>
<td>G1.Large</td>
<td>8</td>
<td>16</td>
<td>70</td>
</tr>
<tr>
<td>M1.Large</td>
<td>8</td>
<td>32</td>
<td>70</td>
</tr>
<tr>
<td>C1.Xlarge</td>
<td>16</td>
<td>32</td>
<td>70</td>
</tr>
</tbody>
</table>

Table 40 - OpenStack virtual instances configuration

- **Monitoring and billing**: this component relies on the Cloudify monitoring system to track the actual QoS and identify incidents and service degradations. The values registered by the monitoring system are used by LISA module to detect SLA violations and calculate the penalties. According to the actual resources consumption and the overall amount of penalties, Cloud Health module calculates the real time cost of each Cloud service and generates a forecast of cost to be paid in the next billing cycle;

- **Legal and security**: this component is not involved in LISA deployment and it is focused on the enforcement of security policies and the prevention of cyber security attacks;

- **Process automation**: this component is not involved in LISA deployment and it focused on the automation of recursive activities to ensure the service security and the business continuity (e.g. patching, OS upgrade, etc.);

- **Resource allocation**: this component relies on LISA algorithms to allocate and release resources according to the actual workloads requirements. The module exploits the APIs of Cloudify to directly interact with private and public Cloud solutions and ensure the proper degree of control on the virtual instances;

- **Maintenance**: this component is not involved in LISA deployment and it focused on the customizations and upgrades of the business services.

The foundation layer is composed by private and public solutions in scope of the test and the related network configurations. To ensure high-speed and low-latency
connectivity, the services running on public Cloud are connected with the on-premise deployment through dedicated connectivity (i.e. AWS Direct Connect and Azure Express Route).

5.3 Test cases and experimental results
To validate the effectiveness of LISA framework, the test defined three different test cases executed on private, hybrid and Multi-Cloud configurations. The test measured the performance of the resource allocation techniques according to three policies:

- **Time-based**: this policy requires that the workload is completed within a specific time frame;
- **Cost-based**: this policy requires that the workload is completed using a maximum amount of virtual resources to avoid cost increase;
- **Availability-based**: this policy requires the continuous availability of the business services during a specific time frame.

5.3.1 Network performance analysis on on-premise OpenStack
The first test case considered the analysis of a network performance log file extracted from the NetPerform service. The network analysis application was installed on virtual machines deployed on OpenStack and tested with different resources allocation techniques against time-based and cost-base policies.

The algorithms of each resource allocation technique were configured in the Cloud orchestrator deployed on a dedicated virtual machine (i.e. operations management instance). The test case included two different workload scenarios and considered the Cloud hosting pricing model provided by the telecommunication companies as B2B offering.

**Scenario 1**
This scenario evaluated the performance of the resource allocation techniques to correctly complete a workload according the following configuration:

- **Workload type**: Analysis of a network performance log file
- **File dimension**: 20,000 rows (373 MB)
- **Total VMs available**: 2
- **VM configuration**: G1.Large (8 vCPU, 16 GB vRAM) - Linux Red Hat
- **Maximum workload duration**: 500 seconds
- **Maximum workload cost**: 0.85 €

The experimental results show an exceeding execution cost for PURS, QRPS and ARCS techniques. Moreover, QRPS completed the workload after the expected deadline, positioning itself as the worst resource allocation technique. STAR provided good results on both QoS metrics, ensuring 3.5% time saving and 6.8% cost saving compared to PURS. However, LISA framework provided better results on both the QoS metrics, positioning itself as the most effective resource allocation technique for this scenario (Table 41).

<table>
<thead>
<tr>
<th>Resource allocation technique</th>
<th>Workload execution time (sec)</th>
<th>SLA breach (time)</th>
<th>Workload execution cost (€)</th>
<th>SLA breach (cost)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PURS</td>
<td>485</td>
<td>No</td>
<td>0.88</td>
<td>Yes</td>
</tr>
<tr>
<td>QRPS</td>
<td>503</td>
<td>Yes</td>
<td>0.98</td>
<td>Yes</td>
</tr>
<tr>
<td>ARCS</td>
<td>496</td>
<td>No</td>
<td>0.96</td>
<td>Yes</td>
</tr>
<tr>
<td>STAR (cost-based)</td>
<td>473</td>
<td>No</td>
<td>0.76</td>
<td>No</td>
</tr>
<tr>
<td>STAR (time-based)</td>
<td>468</td>
<td>No</td>
<td>0.82</td>
<td>No</td>
</tr>
<tr>
<td>LISA (cost-based)</td>
<td>466</td>
<td>No</td>
<td>0.71</td>
<td>No</td>
</tr>
<tr>
<td>LISA (time-based)</td>
<td>453</td>
<td>No</td>
<td>0.75</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 41 - Network performance analysis on OpenStack - Scenario 1

LISA cost-based ensured 2.5% time saving compared to STAR cost-based and LISA time-based ensured 4.3% time saving compared to STAR time-based (Figure 75)

In the same way, LISA cost-based ensured 7.6% cost saving compared to STAR cost-based and LISA time-based ensured 9.6% cost saving compared to STAR time-based (Figure 76).
Scenario 2

This scenario evaluated the performance of the resource allocation techniques to correctly complete three parallel workloads according the following configuration:

- **Workload type**: Analysis of three network performance log files in parallel
- **Files dimension**:
  - File 1: 20,000 rows (373 MB)
  - File 2: 22,000 rows (412 MB)
  - File 3: 25,000 rows (467 MB)
- **Total VMs available**: 6
- **VM configuration**: C1-Large (4 vCPU, 8 GB vRAM) - Linux Red Hat
- **Maximum workload duration**: 1700 seconds
- **Maximum workload cost**: 3,75 €

The experimental results show an important improvement in terms of SLA fulfilment for almost all the resource allocation techniques, since only PURS caused a SLA breach on the execution cost due to an ineffective horizontal scaling. This result is linked to the higher availability of virtual machines and the related possibility to parallelize the tasks across different nodes. ARCS gained a position thanks to the possibility to share resources across different workloads to reduce both cost and execution time. Like the results highlighted in scenario 1, LISA framework provided better results on both the QoS metrics (Table 42).
<table>
<thead>
<tr>
<th>Resource allocation technique</th>
<th>Workload execution time (sec)</th>
<th>SLA breach (time)</th>
<th>Workload execution cost (£)</th>
<th>SLA breach (cost)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PURS</td>
<td>1627</td>
<td>No</td>
<td>3.47</td>
<td>No</td>
</tr>
<tr>
<td>QRPS</td>
<td>1659</td>
<td>No</td>
<td>3.78</td>
<td>Yes</td>
</tr>
<tr>
<td>ARCS</td>
<td>1621</td>
<td>No</td>
<td>3.42</td>
<td>No</td>
</tr>
<tr>
<td>STAR (cost-based)</td>
<td>1591</td>
<td>No</td>
<td>3.19</td>
<td>No</td>
</tr>
<tr>
<td>STAR (time-based)</td>
<td>1568</td>
<td>No</td>
<td>3.34</td>
<td>No</td>
</tr>
<tr>
<td>LISA (cost-based)</td>
<td>1583</td>
<td>No</td>
<td>3.11</td>
<td>No</td>
</tr>
<tr>
<td>LISA (time-based)</td>
<td>1559</td>
<td>No</td>
<td>3.24</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 42 - Network performance analysis on OpenStack - Scenario 2

In this scenario, LISA is still the best resource allocation technique but gap from STAR is strongly reduced compared to the one highlighted in scenario 1. LISA cost-based ensured 0.5% time saving compared to STAR cost-based and LISA time-based ensured 0.6% time saving compared to STAR time-based (Figure 77).

![Figure 77 - Network performance analysis - Scenario 2 execution time (seconds)](image)

In the same way, LISA cost-based ensured 2.4% cost saving compared to STAR cost-based and LISA time-based ensured 3.0% cost saving compared to STAR time-based.

![Figure 78 - Network performance analysis - Scenario 2 workload cost (£)](image)
5.3.2 Activity tracker in hybrid Cloud configuration

The second test case considered a stress test on an Agile activity tracker system executed on a hybrid Cloud environment between the private data centers in Dublin and the North Region of Azure Cloud. The two sites were connected through a VPN tunnel, enabling the communication between the 10.11.0.0/16 subnet on Azure and the 20.2.0.0/16 subnet on the on-premise data center (Figure 79). The algorithms of each resource allocation technique were configured in the Cloud orchestrator deployed on a dedicated virtual machine running on OpenStack (i.e. operations management instance).

The experimental environment was composed by a private deployment of OpenStack, a public Azure Cloud and a container cluster deployed with Kubernetes. The activities tracking tool, namely Jira, was installed on OpenStack and the collaboration software of the same suite, namely Confluence, was installed on Kubernetes. To test the performance of the hybrid Cloud deployment, the applications shared the same Maria DB installed on Microsoft Azure (Figure 80).

The experiment evaluated the performance of the resource allocation techniques during the execution of two different test plans, measuring both incidents (i.e. service unavailability) and service degradations.
• **Workload type**: 2 test plans
  - Creation of a new event in Confluence
  - Creation of a new task in Jira

• **Total private VMs available**: 4 (2 VMs for Jira and 2 VMs for Confluence)

• **Private VM configuration**: C1-Large (4 vCPU, 8 GB vRAM) - Linux Red Hat

• **Total Azure VMs available**: 2 VMs

• **Public VM configuration**: A4 v2 (4 vCPU, 8 GB vRAM) - Linux Red Hat

• **Incident**: response-time > 10 seconds (web page in time-out)

• **Service degradation**: response-time > 3 seconds

The test case included two different workload scenarios to test the effectiveness of the resource allocation techniques under different conditions. Both the tests considered the same number of users but with a different frequency of access to systems (Figure 81).

**Scenario 1**

This scenario evaluated the performance of the resource allocation techniques to correctly complete a loop of 1.250.000 activities (2500 users in loop for 500 time). The access to the platform is based on 4 steps of 500 users executed every 5 minutes:

• **Number of threads (users)**: 2500 (1250 on Jira and 1250 on Confluence)

• **Initial number of threads (users)**: 500 (250 on Jira and 250 on Confluence)

• **Threads increase frequency**: 500 additional users every 5 minutes

• **Ramp-up period**: 2100 seconds (35 minutes)

• **Loop count**: 500

The experimental results highlight a first critical peak of load after 10 minutes of test
when 1500 users already accessed the system. PURS and QRPS generated the first service degradations around this timeframe and caused a long series of slow responses, strongly affecting the users’ experience. ARCS ensured better performance, generating the first service degradation after 12 minutes but it generated more than 10% of slow responses. STAR and LISA confirmed the results highlighted in the first test case, confirming themselves as the best allocation techniques and generating the first service degradation after 15 minutes of test when 2000 users already accessed the system. LISA framework ensured the best performance, generating the first service degradation after 29 seconds the one generated by STAR and provided a more effective service execution generating 0,44% service degradations less than STAR (Table 43).

<table>
<thead>
<tr>
<th>Resource allocation technique</th>
<th>1st incident occurrence (sec)</th>
<th>Total % incidents</th>
<th>1st degradation occurrence (sec)</th>
<th>Total % degradations</th>
</tr>
</thead>
<tbody>
<tr>
<td>PURS</td>
<td>948</td>
<td>1,72%</td>
<td>664</td>
<td>12,48%</td>
</tr>
<tr>
<td>QRPS</td>
<td>987</td>
<td>1,46%</td>
<td>641</td>
<td>13,72%</td>
</tr>
<tr>
<td>ARCS</td>
<td>1056</td>
<td>0,91%</td>
<td>728</td>
<td>10,39%</td>
</tr>
<tr>
<td>STAR</td>
<td>-</td>
<td>-</td>
<td>983</td>
<td>8,73%</td>
</tr>
<tr>
<td>LISA</td>
<td>-</td>
<td>-</td>
<td>1012</td>
<td>8,29%</td>
</tr>
</tbody>
</table>

Table 43 - Activity tracker analysis - Scenario 1 result

The experimental results highlight a second critical peak of load after 15 minutes of test when 2000 users already accessed the system. PURS and QRPS generated the first incident around this timeframe and caused a series of time-out errors before correctly allocating the workload. ARCS ensured better performance, generating the first incident after 17 minutes and providing a more stable service with only 0,91% of time-out errors. STAR and LISA ensured a high-available service execution, completing the stress test without generating incidents (Figure 82).
Scenario 2
This scenario evaluated the performance of the resource allocation techniques to correctly complete a loop of 1,250,000 activities (2,500 users in loop for 500 time). The access to the platform is based on 9 different steps of 250 users each and executed every 3 minutes:

- **Number of threads (users):** 2,500 (1,250 on Jira and 1,250 on Confluence)
- **Initial number of threads (users):** 250 (125 on Jira and 125 on Confluence)
- **Threads increase frequency:** 250 additional users every 3 minutes
- **Ramp-up period:** 2,100 seconds (35 minutes)
- **Loop count:** 500

Like the results provided by the first scenario, the second test plan highlighted a critical peak of load after 15 minutes when 1,500 users already accessed the system. In fact, PURS and QRPS generated the first service degradation during this time frame. A second critical peak of load affected the performance of ARCS after 18 minutes when 1,750 users already accessed the system and caused a series of slow responses provided by the system. Despite a better resistance in generating the first service degradation, ARCS caused an overall percentage of slow responses like the one caused by PURS and QRPS (Table 44).

<table>
<thead>
<tr>
<th>Resource allocation technique</th>
<th>1st incident occurrence (sec)</th>
<th>Total % incidents</th>
<th>1st degradation occurrence (sec)</th>
<th>Total % degradations</th>
</tr>
</thead>
<tbody>
<tr>
<td>PURS</td>
<td>1293</td>
<td>0,68%</td>
<td>934</td>
<td>5,69%</td>
</tr>
<tr>
<td>QRPS</td>
<td>1337</td>
<td>0,71%</td>
<td>951</td>
<td>5,62%</td>
</tr>
<tr>
<td>ARCS</td>
<td>1489</td>
<td>0,52%</td>
<td>1092</td>
<td>5,34%</td>
</tr>
<tr>
<td>STAR</td>
<td>-</td>
<td>-</td>
<td>1313</td>
<td>3,28%</td>
</tr>
<tr>
<td>LISA</td>
<td>-</td>
<td>-</td>
<td>1471</td>
<td>2,73%</td>
</tr>
</tbody>
</table>

Table 44 - Activity tracker analysis - Scenario 2 result

A third critical peak of load affected the performance of STAR framework after 21 minutes, when 2,000 users already accessed the system. This peak of load affected also PURS and QRPS since they caused a set of time-out errors, affecting the availability of the system (Figure 83). LISA confirmed itself as the best resource allocation technique since generated the first service degradation after 24 minutes, when 2,250 users already accessed the system. In the same time-frame, ARCS generated a long series of time-out errors, reaching 0,52% of requests not properly managed.
This scenario provided an important improvement in the performance of all the resource allocation techniques, proving that frequent and small peaks of load are easier to manage than the rare and huge ones. Comparing the two scenarios, PURS and QRPS reached more than 50% reduction on both incidents and service degradations while ARCS limited the improvement respectively to 42% and 48%. STAR and LISA frameworks ensured the best result also according to the improvement metric, reaching respectively to 62% and 67% (Table 45).

<table>
<thead>
<tr>
<th>Resource allocation technique</th>
<th>Incident reduction (scenario 1 vs scenario 2)</th>
<th>Degradation reduction (scenario 1 vs scenario 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PURS</td>
<td>53,42%</td>
<td>58,53%</td>
</tr>
<tr>
<td>QRPS</td>
<td>58,72%</td>
<td>54,97%</td>
</tr>
<tr>
<td>ARCS</td>
<td>42,86%</td>
<td>48,60%</td>
</tr>
<tr>
<td>STAR</td>
<td>-</td>
<td>62,43%</td>
</tr>
<tr>
<td>LISA</td>
<td>-</td>
<td>67,07%</td>
</tr>
</tbody>
</table>

Table 45 - Result comparison - Scenario 1 vs scenario 2

5.3.3 Converged data platform in Multi-Cloud configuration

The third test case considered a stress test on a converged data platform executed on a Multi-Cloud environment created on AWS and Azure and connected to the private data centers in Dublin. All the sites were connected through a VPN tunnel to ensure a dedicated communication and low latency (Figure 84).

The experimental environment was composed by a private deployment of OpenStack and two active nodes on Azure and AWS to ensure high-availability. The converged data platform, namely MapR, and custom marketing campaign application were installed in cluster on both Azure and AWS.
The data visualization system, namely Pentaho, was installed on the private deployment of OpenStack to generate synthetic dashboards (Figure 85). The data warehouse on customers profiling was uploaded on both Azure and AWS to ensure data consistency.

The experiment evaluated the completion time of the resource allocation techniques during the execution of two different direct marketing campaigns:

- **Workload type**: 2 direct marketing campaigns
- **Total private VMs available**: 1 VM for Pentaho platform
- **Private VM configuration**: G1-Medium (2 vCPU, 8 GB vRAM) - Linux Red Hat
- **Total Azure VMs available**: 2 VMs - F4 v2 (4 vCPU, 8 GB vRAM) - Linux Red Hat
- **VMs execution cost**: $0.345/hour on Azure and $0.23/hour on AWS
- **Total AWS VMs available**: 2 VMs - c5.xlarge (4 vCPU, 8 GB vRAM) - Linux Red Hat
**Scenario 1**

This scenario evaluated the performance of the resource allocation techniques to identify new prospects in scope for a direct marketing campaign through social networks (i.e. Facebook, Twitter and LinkedIn). The converged data platform analyzed a data set populated with 1 million of dummy users with unaggregated information extracted by different sources.

- **Workload type**: Direct marketing campaign
- **Data set dimension**: 1 million users (5.71 GB)
- **Maximum workload duration**: 2700 seconds (45 minutes)
- **Maximum workload cost**: 0.52 $

The experimental results show an exceeding execution cost and time for both PURS and QRPS. ARCS and STAR completed the customers profiling activity within the expected time and cost, proving their effectiveness in highly distributed environments. Considering the execution time, LISA framework provided better results on all the QoS metrics, ensuring 1% effectiveness compared to STAR and positioning itself as the most effective resource allocation technique for this scenario (Table 46).

<table>
<thead>
<tr>
<th>Resource allocation technique</th>
<th>Workload execution time (sec)</th>
<th>SLA breach (time)</th>
<th>Workload execution cost ($)</th>
<th>SLA breach (cost)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PURS</td>
<td>2723</td>
<td>Yes</td>
<td>0.63</td>
<td>Yes</td>
</tr>
<tr>
<td>QRPS</td>
<td>2781</td>
<td>Yes</td>
<td>0.57</td>
<td>Yes</td>
</tr>
<tr>
<td>ARCS</td>
<td>2653</td>
<td>No</td>
<td>0.49</td>
<td>No</td>
</tr>
<tr>
<td>STAR (cost-based)</td>
<td>2612</td>
<td>No</td>
<td>0.43</td>
<td>No</td>
</tr>
<tr>
<td>STAR (time-based)</td>
<td>2561</td>
<td>No</td>
<td>0.47</td>
<td>No</td>
</tr>
<tr>
<td>LISA (cost-based)</td>
<td>2597</td>
<td>No</td>
<td>0.41</td>
<td>No</td>
</tr>
<tr>
<td>LISA (time-based)</td>
<td>2534</td>
<td>No</td>
<td>0.44</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 46 - Converged data platform - Scenario 1 result (marketing campaign)

The test clearly highlighted a gap between QRPS and PURS and all the other resource allocation techniques, reaching till 9% between the best (i.e. LISA) and the worst (i.e. QRPS) (Figure 86). However, a more critical gap is highlighted by the execution cost metric in which LISA ensured 45% cost saving compared to PURS. LISA cost-based ensured 4.6% cost saving compared to STAR cost-based and LISA time-based ensured 6.4% cost saving compared to STAR time.
The improvement is considerably bigger than the one in the first test case, proving the effectiveness of LISA framework in a Multi-Cloud scenario (Figure 87).

Scenario 2

This scenario evaluated the performance of the resource allocation techniques to execute a churn prevention campaign on 1 million of existing users (dummy users).

- **Workload type:** Churn prevention campaign
- **Data set dimension:** 1 million users (3.36 GB)
- **Maximum workload duration:** 2100 seconds (35 minutes)
- **Maximum workload cost:** 0.45 $

The experimental results show how a structured and smaller file is easier to manage compared to the unaggregated one analyzed in scenario 1. However, also in this case, PURS and QRPS exceeded the execution cost and breached the SLA (Table 47). ARCS completed the activity within the thresholds of time and cost, proving its effectiveness in hybrid and Multi-Cloud environments.
<table>
<thead>
<tr>
<th>Resource allocation technique</th>
<th>Workload execution time (sec)</th>
<th>SLA breach (time)</th>
<th>Workload execution cost ($)</th>
<th>SLA breach (cost)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PURS</td>
<td>2034</td>
<td>No</td>
<td>0,52</td>
<td>Yes</td>
</tr>
<tr>
<td>QRPS</td>
<td>2052</td>
<td>No</td>
<td>0,49</td>
<td>Yes</td>
</tr>
<tr>
<td>ARCS</td>
<td>1987</td>
<td>No</td>
<td>0,44</td>
<td>No</td>
</tr>
<tr>
<td>STAR (cost-based)</td>
<td>1948</td>
<td>No</td>
<td>0,36</td>
<td>No</td>
</tr>
<tr>
<td>STAR (time-based)</td>
<td>1911</td>
<td>No</td>
<td>0,41</td>
<td>No</td>
</tr>
<tr>
<td>LISA (cost-based)</td>
<td>1934</td>
<td>No</td>
<td>0,36</td>
<td>No</td>
</tr>
<tr>
<td>LISA (time-based)</td>
<td>1896</td>
<td>No</td>
<td>0,39</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 47 - Converged data platform - Scenario 2 result (churn prevention campaign)

This scenario confirmed LISA and STAR as the best resource allocation techniques. LISA cost-based ensured 0,7% time saving compared to STAR cost-based and LISA time-based ensured 0,8% time saving compared to STAR time-based (Figure 88).

In the same way, LISA time-based ensured 4,8% cost saving compared to STAR time-based but, surprisingly, the cost-based versions of LISA and STAR obtained the same results (Figure 89).
5.4 Conclusions and validation

The effectiveness of LISA framework was evaluated on an experimental environment, created in the innovation lab of a global telecommunication company. LISA framework was compared against the main resource allocation techniques (i.e. PURS, QRPS, ARCS and STAR) and evaluated on three different Cloud deployments and test cases. The tests considered different QoS parameters, calculating the SLA violation rate, the execution time and the execution cost. All the test cases and the related scenarios, confirmed LISA as the most effective resource allocation technique and highlighted the ability of the proposed approach in improving the scalability of Cloud-based services.
6 Conclusions

This research aimed to analyze the Multi-Cloud trend and identify the impacts on the SLA management lifecycle. To provide clear answers to the research questions, the research deeply analyzed the state of the art through a structured literature review and evaluated the main Cloud business trends through a survey on several multinational companies. In addition, the research presented a SLA-aware autonomic framework to ensure the QoS in redundant Multi-Cloud environments. The effectiveness of the framework in reducing the SLA violation rate was tested in private and public Cloud environments proving better performance compared to existing resource management techniques.

6.1 Key insights

The high maturity of Cloud technologies and the availability of innovative solutions are driving the digital strategy of multinational companies, facilitating the migration of IT workload on public and hybrid Cloud environments. Moreover, the lack of Cloud providers able to cover a global strategy is forcing multinational companies in selecting different Cloud providers and embracing a Multi-Cloud approach [102]. The adoption of Multi-Cloud environments ensures several benefits, but it introduces, at the same time, some important challenges in tracking the services performance and ensuring the compliance with the agreed SLAs.

From an academic viewpoint, the growing number of researches focused on SLA management in Cloud environments, often supported by government institutions, highlights the importance of this topic. Several studies proposed different SLA-aware frameworks and autonomic resources allocation techniques without a structured approach to cover the entire SLA management lifecycle and tailored features to ensure the tracking of services performance in Multi-Cloud environments.

From a business viewpoint, different industries approached the Cloud adoption with diverse strategies, but they are now converging to the adoption of hybrid and Multi-Cloud solutions. Most of the surveyed companies highlighted the service level management as a key point of attention to ensure a successful Multi-Cloud model adoption. Moreover, the main IT management frameworks (i.e. ITIL and COBIT) still relies
on the possibility to directly interact with the IT systems and, for this reason, not conceived for hybrid and Multi-Cloud solutions [103].

These insights clearly indicate the need of a centralized ecosystem to control the services performance and ensure the compliance with the agreed SLA. Moreover, the main market analysts predicts a wide adoption of Cloud brokering system to mitigate this gaps and facilitate the implementation of a Multi-Cloud strategy [104].

According to this trend, the research defined a SLA-aware framework, named LISA, able to cover all the SLA management activities and reduce the SLA violation rate. The framework relies on OASIS TOSCA best practices and takes advantage from open-source languages and technologies (e.g. XML, WS-Agreement, OCCI, etc.) to ensure compatibility with Cloud providers and IT management tools. LISA aims to act as unique entry point for Cloud management activities, providing several features to cover both technological and business requirements.

To prove the effectiveness of the framework, it was implemented as a Python module of the Cloud orchestrator and tested in the Innovation Lab of a global telecommunication company. The experiment analyzed the behavior of services executed on different private Cloud solutions, deployed across Milan, Düsseldorf and London, and two regions of Amazon Web Service and Microsoft Azure (i.e. Germany and Ireland). In addition, to ensure a scientific approach and evaluate the effectiveness of the framework in different scenarios, the test considered different kind of services and workloads (e.g. web sites, backup service, productivity applications, etc.). The results of the test show that LISA framework ensures better performance in terms of SLA violation rate, execution time and execution cost compared to existing SLA-oriented resource allocation techniques (i.e. PURS, QRPS, ARCS and STAR. Moreover, acting as a unique interface for Cloud management activities, LISA provides a better user experience and enables an IT standardization and a faster delivery process.

6.2 Answers to the research questions

The research addresses all the questions defined through a complete analysis of the state of the art:

- **Question 1**: Is the future of Cloud services adoption widely based on a Multi-
Cloud approach?

- **Answer 1:** the SLR clearly highlighted a growing interest on the Multi-Cloud model and the main opportunities and challenges provided by its adoption. Likewise, the enterprise survey showed how the high maturity of the current Cloud solutions and the need of a wider geographical coverage are supporting the adoption of different public Cloud providers. For this reason, Multi-Cloud can be considered as a natural evolution of the Cloud strategy of international enterprises, representing the last step of the Cloud adoption journey;

- **Question 2:** what are the impacts on the services lifecycle caused by a Multi-Cloud model adoption and which are the available best-practices and frameworks to address them?
  - **Answer 2:** the Multi-Cloud model implies a series of challenges directly related to the heterogeneity of the environments and the distribution of the tenants on different Cloud platform. The SLR highlighted some critical gaps in the main SLA-oriented frameworks, still not fully able to track the performance and manage the scalability of services deployed across multiple public Cloud providers. Likewise, the enterprise survey showed how the main IT management frameworks (e.g. ITIL, COBIT, etc.) are still not mature enough to support the adoption of a Multi-Cloud model. New Cloud-oriented versions of the main IT management frameworks are still under definition and they will be released in the second quarter of 2019. For this reason, to ensure a successful adoption of the Multi-Cloud model it is required the creation of a tailored Cloud-oriented ecosystem base on standard APIs to enable the proper visibility on the entire service lifecycle and facilitate the communication across the different Cloud solutions;

- **Question 3:** Is Service Level Management strongly affected by the adoption of a Multi-Cloud model?
  - **Answer 3:** Service Level Management and Security are the most impacted areas in case of adoption of a Multi-Cloud model. The simultaneous execution of an application across different public Cloud providers
complicates the monitoring of the actual QoS and the clear detection of SLA breaches. For this reason, to ensure a secure and structured adoption of the Multi-Cloud model it is required a detailed negotiation with the selected Cloud providers and the implementation of a robust monitoring system.

The research highlighted different points of attention to enable the adoption of a Multi-Cloud model and provided guidelines fully in line with the “Manifesto for Future Generation Cloud Computing”, proving its scientific value and the novelty of LISA framework [105].

6.3 Research limitations

Despite the proven scientific value of the research, it presents some limitations that can be solved in future studies [106].

- **Literature review limitation**: the systematic literature review analyzed a wide set of academic researches, clearly highlighting a gap in the current SLA management techniques. However, the literature review analyzed only academic researches, not considering proprietary techniques adopted by suppliers of Cloud management systems;

- **Enterprise survey limitation**: the survey deeply analyzed the Cloud strategies and the adoption trends of important multinational companies, most of them included in the Top 500 Fortune. However, due to the structured approach adopted, the sample is limited to 62 companies. Moreover, the percentage of Asian companies is very small (12%) and confined only to UAE, Turkey and Japan;

- **LISA framework limitation**: the framework relies on OASIS TOSCA technologies and was developed as a module of Cloudify orchestrator, using Python programming language. The experiment setup analyzed both the performance of services running on different private and public Cloud solutions. However, due to the lack of a Cloud native applications developed according the microservices logic, the test considered only the redundant Multi-Cloud configuration.
6.4 Future directions

Thanks to the structured approach adopted for the analysis of the state of the art and the modular design of the framework, the research provides several guidelines and ideas for future studies.

- **Literature review**: the systematic literature review can be replicated in two years to identify the best practices selected to support the adoption of a Multi-Cloud strategy;

- **Enterprise survey**: the survey can be extended to Chinese and Indian companies to analyze some of the most important Cloud markets. Moreover, these companies could provide interesting insights about the SLA management lifecycle ensured by Alibaba Cloud and Tata communications;

- **LISA integration for IoT environments**: the framework can be extended to ensure the QoS of highly distributed services deployed on IoT, M2M and Edge computing solutions [107];

- **LISA SLA parameters extension**: the parameters currently managed by LISA framework are limited to execution time, cost, availability, reliability, latency and SLA violation. Additional parameters can be included to drive the resources allocation to ensure energy efficiency and detect cyber-attacks [108];

- **LISA integration with CASB**: the framework can be empowered with APIs to enable the dialogue with Cloud Access Security Brokers (CASB) to ensure visibility on the service performance and include additional security management features;

- **LISA integration with machine learning system**: the framework can be extended with the integration of a machine learning system to improve the horizontal scalability feature and enable the predictive maintenance of physical components
7 IEEE and ACM publications

The scientific value of this thesis is supported by a series of high-ranked conference proceedings widely cited towards the last eight years (citations extracted from Scopus Index on December 2018).

7.1 Publications 2015 - onward

Five conference proceedings were published during the executive Ph.D. (from 2015 to 2018).

- **Title**: Public Cloud Adoption in Multinational Companies - A Survey
  - **Year**: 2018
  - **Conference**: IEEE International Conference on Services Computing (SCC)
  - **Ranking**: B1 (Qualis) - A (ERA)
  - **Authors**: Nicola Sfondrini, Antonella Longo, Gianmario Motta
  - **ISBN**: 978-1-5386-7250-1
  - **Citations**: -

- **Title**: SLA-aware broker for Public Cloud
  - **Year**: 2017
  - **Conference**: IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)
  - **Ranking**: A2 (Qualis) - B (ERA)
  - **Authors**: Nicola Sfondrini; Gianmario Motta
  - **ISBN**: 978-1-5386-2704-4
  - **Citations**: -

- **Title**: SLM-as-a-Service-a conceptual framework
  - **Year**: 2017
  - **Conference**: IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)
  - **Authors**: Nicola Sfondrini; Gianmario Motta
  - **ISBN**: 978-1-5090-4499-3
  - **Citations**: 2
• **Title:** Service level agreement (SLA) in Public Cloud environments: a survey on the current enterprises adoption  
  **Year:** 2015  
  **Conference:** 5th International Conference on Information Science and Technology (ICIST)  
  **Authors:** Nicola Sfondrini; Gianmario Motta; Linlin You  
  **ISBN:** 978-1-4799-7489-4  
  **Citations:** 10

• **Title:** SLM as a Third-Party Service in Cloud Environment: A Reference Framework  
  **Year:** 2015  
  **Conference:** IEEE International Conference on Services Computing (SCC)  
  **Ranking:** B1 (Qualis) - A (ERA)  
  **Authors:** Linlin You; Gianmario Motta; Nicola Sfondrini  
  **ISBN:** 978-1-4673-7281-7  
  **Citations:** 3

### 7.2 Publications 2011 - 2014

Five conference proceedings were published as refinement and empowerment of the research developed for the Master of Science dissertation (from 2011 to 2014).

• **Title:** Service Level Management (SLM) in Cloud Computing - Third Party SLM Framework  
  **Year:** 2014  
  **Conference:** IEEE 23rd International WETICE Conference  
  **Ranking:** B1 (Qualis) - B (ERA)  
  **Authors:** Gianmario Motta; Linlin You; Nicola Sfondrini; Daniele Sacco; Tianyi Ma  
  **ISBN:** 978-1-4799-4249-7  
  **Citations:** 9
• **Title:** Cloud Computing: The Issue of Service Quality: An Overview of Cloud Service Level Management Architectures  
  **Year:** 2013  
  **Conference:** 5th International Conference on Service Science and Innovation (ICSSI)  
  **Authors:** Gianmario Motta; Linlin You; Daniele Sacco; Nicola Sfondrini  
  **ISBN:** 978-0-7695-4985-9  
  **Citations:** 7

• **Title:** Cloud Computing: A Business and Economical Perspective  
  **Year:** 2012  
  **Conference:** International Joint Conference on Service Sciences (IJCSS)  
  **Authors:** Gianmario Motta; Nicola Sfondrini; Daniele Sacco  
  **ISBN:** 978-1-4673-1992-8  
  **Citations:** 8

• **Title:** Cloud Computing: An Architectural and Technological Overview  
  **Year:** 2012  
  **Conference:** International Joint Conference on Service Sciences (IJCSS)  
  **Authors:** Gianmario Motta; Nicola Sfondrini; Daniele Sacco  
  **ISBN:** 978-1-4673-1992-8  
  **Citations:** 12

• **Title:** Research studies on cloud computing: A systematic literature review  
  **Year:** 2011  
  **Conference:** International Business Information Management Association Conference (IBIMA)  
  **Ranking:** B4 (Qualis) - B (ERA)  
  **Authors:** Gianmario Motta; Nicola Sfondrini  
  **ISBN:** 978-0-9821-4896-9  
  **Citations:** 3
7.3 New researches - Accepted and to be published

Two research papers were submitted and currently under review:

- **Title**: LISA: An efficient Lean Service Architecture for SLA management in Multi-Cloud environments  
  **Year**: 2019  
  **Conference**: IEEE International Conference on Services Computing (SCC)  
  **Ranking**: B1 (Qualis) - A (ERA)  
  **Authors**: Nicola Sfondrini, Antonella Longo, Gianmario Motta

- **Title**: A Systematic Literature Review on Service Level Management in Multi-Cloud environments  
  **Year**: 2019  
  **Conference**: IEEE 27th International WETICE Conference  
  **Ranking**: B1 (Qualis) - B (ERA)  
  **Authors**: Nicola Sfondrini, Antonella Longo, Gianmario Motta
8 Annex

This section aims to provide technical details useful to better understand the design of LISA framework and the selection of Cloudify as orchestrator for the implementation phase.

8.1 Cloudify orchestrator overview

Cloudify\(^2\) is an open source Cloud orchestration platform, designed to fully automate the lifecycle of applications and network services in Multi-Cloud environments (Figure 90). Cloudify uses a declarative approach based on TOSCA framework, allowing users to define the desired service level for each application deployments through a simple DSL. The embedded monitoring engine can be configured with additional technical and business parameters to fully automate the scalability and the restore of the services to maintain the desired SLAs.

![Figure 90 - Cloudify orchestration engine](https://cloudify.co)

The flexible architecture of Cloudify ensures a set of benefits:

- **Open access source-code**: the open source license allows to easily customize the orchestrator, adding new features and connecting it to different Cloud solutions (Figure 91);
- **API Pluggability**: the open access APIs enables the native communication with external tools (e.g. reporting, patching, monitoring, etc.) to ensure the creation of a centralized Cloud ecosystem;
- **Visual workflow engine**: the workflow can be visually designed using the available building blocks or creating new blocks to satisfy technical requirements. Moreover, developers can model their topology using YAML and automatically deploy it to the selected Cloud solutions;

---

\(^2\) [https://cloudify.co](https://cloudify.co)
- **Full Cloud stack coverage**: Cloudify ensures the orchestration of both IT and network components, enabling the creation of SDN-based and NVF-based environments. This feature is crucial to enable the orchestration of Cloud servicers deployed in a Multi-Cloud environment;

- **Security by design**: the orchestrator is developed considering the main security risks to ensure the robustness of the entire Cloud environment.

Figure 91 - Cloudify open-source components
8.2 LISA framework - Software design

Due to the business nature of the research, the code of the developed framework is under non-disclosure agreement (NDA). For this reason, the research developed a set of sequence diagrams (Figure 92, Figure 93) and ER diagrams (Figure 94) to simplify the understanding of the LISA architecture.

Figure 92 - Cloud service selection - Sequence diagram

Figure 93 - Billing calculation - Sequence diagram
Figure 94 - Knowledge base ER diagram (partial view)
Bibliography


[61] V. C. Emeakaroha, I. Brandic, M. Maurer, and S. Dustdar, "Cloud resource


S. Green, E. Pearson, V. Gkatzidou, and F. O. Perrin, "A community-centred design approach for accessible rich internet applications (ARIA)," in *Proceedings of the 26th annual bcs interaction specialist group conference on people and computers*, 2012, pp. 89-98.


M. Randles, D. Lamb, and A. Taleb-Bendiab, "A comparative study into distributed load balancing algorithms for cloud computing," in *Advanced Information*


