Contribution to the uptake of Cloud Computing solutions: Design of a cloud services intermediator to foster an ecosystem of trusted, interoperable and legal compliant cloud services. Application to multi-cloud aware software.

THESIS

Contribución a la estimulación del uso de soluciones Cloud Computing: Diseño de un intermediador de servicios Cloud para fomentar el uso de ecosistemas distribuidos digitales confiables, interoperables y de acuerdo a la legalidad. Aplicación en entornos multi-cloud.

Electronics and Telecommunications Engineering Doctorate

at the

UNIVERSITY OF THE BASQUE COUNTRY

eman ta zabal zazu

Universidad del País Vasco

Euskal Herriko Unibertsitatea

Bilbao, 2022

Presentada por: JUNCAL ALONSO IBARRA

Dirigida por: MAIDER HUARTE ARRAYAGO

Acknowledgements // Eskerrak // Agradecimientos

Apart from my effort, this PhD thesis has been successfully completed thanks to the encouragement and guidelines of many people. I take this opportunity to express my gratitude to them through these lines.

First and foremost, I would like to express my gratitude to my family for their unconditional support, love and encouragement. To my husband Borja thank you for your constant belief in me regardless of the challenge, to my daughters and son, thank you for inspiring me with your endless vitality, to my parents, Adelaida and Rafael who raised me with a love of science and supported me in all my pursuits and to my brothers Iñigo and Mikel thank you for all those life moments that we have shared. I would also like to thank my friends and other family members for their unconditional support and trust throughout my studies.

I am also thankful to my PhD director, Dr. Maider Huarte Arrayago from Universidad del País Vasco /Euskal Herriko Unibertsitatea for providing me the freedom to undertake my research objectives and for accompany me through my PhD endeavour.

I would like to take this opportunity to thank TECNALIA to let me complete this research work and specially to my colleagues Iñaki Etxaniz, Gorka Benguria, María José López and Marisa Escalante. My special thanks to Dr. Leire Orue-Echevarría for her mentoring, energy, commitment, and continuous support.

Furthermore, I would like to thank the funding received from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731533 DECIDE project which gave me the opportunity to start the present research line as well as allow me to grow as technical manager.

The research leading to these results has also received funding from the same programme under grant agreement No 101000162 PIACERE project, where I am the project coordinator and No 783221 AFARCLOUD project where I collaborated in the Environment Characterization platform Work package. Within these projects I am continuing the research subjects of this Thesis and I would like to thank the opportunity granted as well.

Lastly, I would like to acknowledge all the members of DECIDE and PIACERE projects consortia for their valuable help in my research.

ESKERRIK ASKO BIHOTZEZ

Table of Contents

List of Tables

List of Figures

Terms and Abbreviations

Acronym	Explanation
ACSmI	Advanced Cloud Service meta-Intermediator
API	Application Programming Interface
CP	Cloud Provider
CPU	Central Computing Unit
CS	Cloud Service
CSB	Cloud Service Broker
CSLA	Cloud Service Level Agreement
ESB	Enterprise Service Bus
GDPR	General Data Protection Regulation
HPC	High Performance Computing
IaC	Infrastructure as Code
IaaS	Infrastructure as a Service
IoT	Internet of things
NFR	Non-Functional Requirement
PaaS	Platform as a Service
QoS	Quality of Service
RoI	Return of Investment
REST	Representational State Transfer
RQ	Research Question
SaaS	Software as a Service
SDLC	Service Development Life Cycle
SLA	Service Level Agreement
SLO	Service Level Objective
SOA	Service Oriented Architecture
SOLC	Service Operation Life Cycle
VM	Virtual Machine

Executive summary

The evolution of cloud computing has changed the way in which Cloud Service Providers offer their services and how Cloud Customers consume them, moving towards the use of multiple cloud services, in what it is called multi-cloud. Indeed, multi-cloud, understood as the serial or simultaneous use of services from diverse providers to execute an application, is getting more and more interest as microservices-based software applications are increasingly getting popular fostering flexibility for the developers to build applications for these referenced distributed execution environments. The expansion of IoT, edge and the Cloud Continuum has increased the complexity of the scene, increasing the types of services, their characteristics, and features.

Considering this situation, intermediate layers to govern the complex ecosystem of cloud services are becoming more and more relevant as the next generation of Cloud environments emerges following the Cloud Continuum paradigm.

The increasing complexity and heterogeneity of cloud computing tasks have reached to the situation where a single cloud environment is frequently inadequate. A multi-cloud, or federated cloud, environment is urgently needed to provide users with required computing resources and services and to serve wide-ranging geographical areas effectively.

However, several technological and scientific challenges are still open with respect to the federation and brokerage of cloud services, being the most relevant ones the (automatic) governance of diverse (in nature) and heterogenous cloud services, the provider agnostic continuous monitoring and assessment of cloud services SLAs and compound SLAs, the interoperability and data portability across different cloud services and the provision of legislation and standardization aware cloud services.

The aim of the work presented in this thesis is to facilitate the developers and operators of multi-cloud-based applications the discovery and management of the different Cloud Services including the support fostering their re-use and combination, for assembling a network of interoperable, legally compliant, quality assessed (against SLAs) single and composite cloud services. A key contribution of the thesis is the design and development of a continuous cloud services mediation enabler framework: ACSmI (Advanced Cloud Services meta-Intermediator). ACSmI provides means to assess continuous real time verification of the cloud services non-functional properties fulfilment and legislation compliance enforcement. ACSmI also provides a cloud services store where developers can easily access centrally negotiated deals of compliant and accredited services.

In addition, the research work characterizes the multi cloud native applications, identifies a set of research challenges related to the development and operation of these applications and proposes the extended DevOps novel concept. The Extended DevOps concept intends to solve some of the challenges of multi-cloud applications design, development, deployment, and adaptation, providing a novel and extended DevOps approach for the fully adaptation of the current DevOps practices to the multi-cloud paradigm.

To evaluate the proposed cloud services intermediator and the associated Extended DevOps concept four testing cases were set up, three of them industrial and a fourth academic one. Using a quantitative and qualitative evaluation process, we demonstrate the feasibility of the proposed cloud services meta-intermediator to support the deployment and operation of multi cloud native applications of different nature and with different deployment needs. The methods and tools proposed by ACSmI introduced significant savings in the efficient discovery, registration, management, contracting, monitoring and portability of cloud services so that these services can be published and accessible to be optimally used. We have also demonstrated that legal aspects can be and should be treated as non-functional requirements proposing an approach to incorporate and assess relevant aspects concerning regulations and legislation.

Finally, we have also shown that the “Extended DevOps” concept makes it possible to adapt the SDLC and SOLC practices to the specific needs of multi cloud native applications. In particular, activities prior to the development and after the operation phases are of special relevance to this.

Laburpena

“Cloud Computing“ (Hodei Konputazioa) kontzeptuak izan duen bilakaerak azken urteetan aldatu egin du hornitzaileek beren zerbitzuak eskaintzeko modua eta baita Hodeiko bezeroek kontsumitzeko modua ere. Aldaketa hau Hodei zerbitzu anitz bakarraren ordez erabiltzean gauzatzen da, multiHodei deritzon horretan. Izan ere, multiHodei kontzeptua, aplikazio bat exekutatzeko hainbat Hodei zerbitzu aldi berean erabiltzea hain zuzen ere, gero eta interesgarriagoa da, mikro-zerbitzuetan oinarritutako software aplikazioak ezagunagoak bait dira. Hau dela eta, aplikazio informatikoen egileek malgutasun handiagoa daukate exekuzio-ingurune banatu hauetarako aplikazioak eraikitzeko. Gauzen Internet, Edge Computing eta Cloud Continuum-aren hedapenak eszenatokiaren konplexutasuna areagotu du, zerbitzu motak, eta haien ezaugarriak handituz.

Egoera horren aurrean, Hodei zerbitzuen ekosistema konplexua gobernatzeko teknologia bitartekarien erabilera gero eta garrantzitsuagoa da. Hodei Anitzeko edo Hodei Federatuaren inguruneak garatu behar dira erabiltzaileei beharrezko baliabide informatiko eta zerbitzuak eskaintzeko.

Hala ere, hainbat erronka teknologiko eta zientifiko aurkitu daitezke Hodei Federatuen testuinguruan. Hodei zerbitzu federazioari dagokionez, garrantzitsuenak hauek dira: Hodei zerbitzu anitzen eta heterogeneoen kudeaketa (automatikoa), zerbitzu-maila-akordioak betetzen direnaren etengabeko ebaluazioa, elkarreragingarritasuna, datuen eramangarritasuna Hodei-zerbitzu ezberdinen artean eta Hodei-zerbitzuak indarrean dagoen legediaren eta estandarren arabera hornitzea.

Tesi honetan aurkezten den ikerketa-lanaren helburua Hodei Anitz aplikazioen egileei eta operadoreei informatika-zerbitzu desberdinak ezagutu eta kudeatzea erraztea, haien erabilpena eta konbinazioa lagunduz, elkarreragin garri diren zerbitzuen sare bat sortzeko. Tesi honen ekarpenetako bat ACSmI (Advanced Cloud Services meta-Intermediator) izeneko soluzioaren diseinua eta garapena da. ACSmI-k zerbitzu-mailako akordioak betetzen diren ebaluatzeko aukera ematen du, legeria barne.

Horrez gain, ikerketa-lan honek hodei anitzeko aplikazio natiboen karakterizazioa eta aplikazio mota honetarako bereziki diseinatutako "Extended DevOps" kontzeptua proposatzen ditu. "Extended DevOps" kontzeptuak diseinu, garapen eta inplementazioko zenbait arazo konpontzea du helburu. DevOps ikuspegi berri eta hedatua eskainiz, egungo DevOps praktikak Hodei anitzeko paradigmara egokitzeko.

Proposatutako ACSmI eta lotutako DevOps kontzeptu hedatuaren bideragarritasuna ebaluatzeko, lau test kasu ezarri dira, horietako hiru industrialak eta laugarrena akademikoa izanik. Ebaluazio-prozesu kuantitatibo eta kualitatibo baten bidez, ACSmI-ren bideragarritasuna frogatzen dugu hodei anitzeko aplikazio natiboen hedapena eta funtzionamendua hobetzeko. Kontuan izan behar da enpirikoki frogatu dela proposatutako metodo eta tresnek aurrezpen handia ekarri dutela Hodei zerbitzuen aurkikuntzan, erregistroan, kudeaketan, kontratazioan, monitorizazioan eta elkarreragingarritasunean. Era berean, alderdi juridikoak betekizun ez-funtzional gisa tratatu daitezkeela erakutsi dugu, araudi eta legediari dagozkion alderdi garrantzitsuak txertatzeko eta ebaluatzeko irtenbide bat proposatuz.

Azkenik, "Extended DevOps" kontzeptuak softwarearen garapena eta eragiketa-zikloaren praktikak Hodei anitzeko aplikazio natiboen behar zehatzetara egokitzea ahalbidetzen duela ere erakutsi dugu.

Resumen Ejecutivo

La evolución de la Computación en la Nube ha cambiado la forma en que los proveedores ofrecen sus servicios asi como la manera en la que los clientes de la Nube los consumen. Este cambio se materializa en el uso de múltiples servicios en la Nube en vez de uno solo, en lo que se denomina multiNube. En efecto, el concepto de multiNube, entendido como el uso en serie o simultáneo de servicios de diversos proveedores para ejecutar una aplicación está cobrando cada vez más interés a medida que las aplicaciones de software basadas en microservicios son cada vez más populares. Este hecho fomenta la flexibilidad para que los desarrolladores construyan aplicaciones para estos entornos de ejecución distribuida referenciados. La expansión de la Internet de las cosas, Computación en el Edge y el Cloud Continuum ha incrementado la complejidad del escenario, aumentando los tipos de servicios, sus características y prestaciones.

A la luz de esta situación, el uso de intermediadores para gobernar el complejo ecosistema de servicios en la Nube se vuelve cada vez más relevante.

La creciente complejidad y diversidad de las tareas de Computación en la Nube han llegado al punto en que un solo entorno de Nube es cada vez más inadecuado y menos útil. Es por eso que es necesario dar soporte a los entornos de Nube múltiple o Nube federada para proporcionar a los usuarios los recursos y servicios informáticos requeridos de manera efectiva.

Sin embargo, existen varios desafíos tecnológicos y científicos con respecto a la federación y la intermediación de servicios en la Nube, siendo los más relevantes la gestión (automática) de servicios en la Nube diversos (en naturaleza) y heterogéneos, su monitorización en tiempo real, la evaluación continua del cumplimiento de los acuerdos de nivel de servicio contratados, la interoperabilidad y la portabilidad de datos entre diferentes servicios en la Nube y la provisión de servicios en la Nube acordes a la legislación y los estándares vigentes.

El objetivo del trabajo de investigación presentado en esta tesis es facilitar a los desarrolladores y operadores de aplicaciones desplegadas en múltiples Nubes el descubrimiento y la gestión de los diferentes servicios de Computación, soportando su reutilización y combinación, para generar una red de servicios interoperables, que cumplen con las leyes y cuyos acuerdos de nivel de servicio pueden ser evaluados de manera continua. Una de las contribuciones de esta tesis es el diseño y desarrollo de un bróker de servicios de Computación llamado ACSmI (Advanced Cloud Services meta-Intermediator). ACSmI permite evaluar el cumplimiento de los acuerdos de nivel de servicio incluyendo la legislación. ACSmI también proporciona una capa de abstracción intermedia para los servicios de Computación donde los desarrolladores pueden acceder fácilmente a un catálogo de servicios acreditados y compatibles con los requisitos no funcionales establecidos.

Además, este trabajo de investigación propone la caracterización de las aplicaciones nativas multiNube y el concepto de “DevOps extendido” especialmente pensado para este tipo de aplicaciones. El concepto “DevOps extendido” pretende resolver algunos de los problemas actuales del diseño, desarrollo, implementación y adaptación de aplicaciones multiNube, proporcionando un enfoque DevOps novedoso y extendido para la adaptación de las prácticas actuales de DevOps al paradigma multiNube.

Para evaluar la viabilidad de la solución propuesta y el concepto de DevOps extendido asociado, se han establecido cuatro casos de prueba, tres de ellos industriales y un cuarto académico. Mediante un proceso de evaluación cuantitativa y cualitativa, demostramos la viabilidad del ACSmI para mejorar la implementación y operación de aplicaciones nativas multiNube. Cabe destacar que se ha demostrado de manera empírica que los métodos y herramientas propuestos introdujeron ahorros significativos en el descubrimiento, registro, gestión, contratación, monitorización e interoperabilidad de los servicios de Computación. También hemos demostrado que los aspectos legales pueden y deben ser tratados como un requisito no funcional proponiendo una solución para incorporar y evaluar aspectos relevantes en materia de normativa y legislación.

Finalmente, también hemos demostrado que el concepto “Extended DevOps” permite adaptar las prácticas del ciclo de vida de desarrollo y operación de software a las necesidades específicas de las

aplicaciones nativas multiNube. En particular, las actividades previas al desarrollo y posteriores a las fases de operación son de especial relevancia en este contexto.

1. Introduction

To adress the digital trasnforamtion and the evolution from product to service economy changes in the companies' operating environment need to be adressed: they need to become service providers from product providers changing their role in the value chain and markets.

To promote this change, the IT infrastructure of the companies needs to be flexible. Cloud services enable partially this flexibility, but also create dependencies mainly to external partners of the company. In the past years the paradigm shift from PC-centric computing to cloud computing has led into the emergence of several Cloud Services and Providers. Initially, Cloud Services were offered as third-party computational capacities but nowadays the offer has become more and more functional diverse, context-specific, and technology-driven. Following this transition, Cloud Services users' consumption of such services has evolved too, from one single Cloud Service type, offered by one provider to the usage of multiple Cloud Services, in what is called a Multi-Cloud approach.

While developing and deploying a web site, an organisation can decide to build it in a dedicated internal server, build it as an instance in a shared resource, build it in a dedicated external server, or even build it as an instance in that external server. The decision on using one, another, or several approaches simultaneously is driven by certain evaluation criteria (e.g. profitability, reliability, performance, security, legal or even ecological aspects). Cloud providers themselves may fail too, so for the greatest measure of protection possible, an enterprise may wish to embark upon a multi-cloud strategy.

Multi-cloud is defined as the serial or simultaneous use of services from diverse providers to execute an application [1]. Hybrid Cloud is also used, mainly at business level. Gartner [2] defines hybrid cloud as *"the coordinated use of cloud services across isolation and provider boundaries among public, private and community service providers, or between internal and external cloud services. Several scenarios demonstrate these serial or simultaneous interactions among hybrid heterogeneous private and public clouds and across all cloud layers (IaaS/PaaS/SaaS)"*. Therefore, Multi-Cloud is getting more and more interest as microservices-based software applications are increasingly getting popular fostering flexibility for the developers to build applications for distributed complex environments.

Microservice architectures have evolved from the Service-Oriented Architecture (SOA) concept [3], improving issues and challenges in terms of applications build and deployment when the size of the application becomes large and distributed. Microservices architectures provide isolated, loosely coupled unit of development that works on a single concern. This independency makes them the best candidate to profit from the advantages of heterogeneous Hybrid Cloud scenarios. Each application component (or microservice) can be deployed independently, considering its specific deployment needs or desired Non-Functional Requirements (NFR) such as location, cost, performance, etc. Especially for critical business applications and applications where the fulfilment of the Service Level Agreements (SLA)s is crucial, the possibility of selecting different types of services with different characteristics and Service Level Objectives (SLOs) optimized for each component incorporates relevant benefits as no NFR needs to be favoured at the expense of other. This new paradigm, where a single application is deployed over an ecosystem of heterogeneous and distributed cloud resources encompasses new needs in terms of management, governance, monitoring, or SLA assessment.

While the research community has focused the effort to advance in the way the software applications can be de-coupled with the aforementioned micro-services based architectures and the enablers that support the operation of this kind of newly distributed applications (i.e. containerization, server-less computing, etc.), a few research initiatives have been concentrated on the management of federated, interoperable, self-monitored, and legal compliant ecosystem of cloud services from a holistic point of view, not only focusing on a single step of the Cloud Service Lifecycle, like for example the operation and deployment of the physical resources [4,5].

There are several multi-cloud solutions available for solving specific problems, but to date, little attention has been paid to distributing the cloud risk (i.e. vendor lock-in, unavailability) and managing multiple clouds from a single technology platform. Working with many Cloud Service Providers (CSPs) means managing multiple relationships. Most enterprises are already negotiating multiple contracts with multiple CSPs and multiple contracts mean multiple service level agreements, multiple payments, multiple passwords, multiple data streams, and multiple providers to check up on. That leads to questions about how to make those services work together, or how to unify all the efforts so as to optimize effectiveness and efficiency.

In addition, intermediate layers to govern the complex ecosystem of Cloud Services are becoming more and more relevant as the next generation of Cloud environments emerges following the Cloud Continuum paradigm. The new Computing Continuum can be defined as a heterogeneous environment based on the decentralization and federation of diverse computing entities and resource typologies [6]. Besides the traditional Cloud Computing services, Fog and Edge Computing services are to be part of this Cloud Continuum. To this end, the support of Multi-Cloud and computing federation models are required so that diverse, decentralized and autonomic management and hybrid computing models can be implemented. The brokerage of Multi-Cloud services can provide flexible means for assembling heterogeneous Cloud-based elements so that compute workloads and requirements are deployed across multiple Cloud environments to provide an optimal delivery model. This approach can be then extended to support the Cloud Computing Continuum.

That leads to questions about how to make those heterogenous services work together, or how to unify all the efforts so maximum effectiveness and efficiency can be obtained out of the existing computing services. This is when a Cloud Service Broker (CSB) comes into play. The CSB goal is to make the service more specific to a company or to a concrete software component as part of a whole application, or to integrate or aggregate services, to enhance their security, or to to implement approaches which add a significant layer of value (i.e., capabilities) to the original Cloud services being offered. Consumers can leverage solutions offered by CSBs that allow organizations to focus on other business needs instead [7].

Existing cloud services shall be made available dynamically, broadly, and cross border, so that software providers can re(use) and combine cloud services, assembling a dynamic and re-configurable network of interoperable, legal compliant, quality assessed (against SLAs) single and composite cloud services. To this extent, a viable intermediator and federator of Cloud Services Brokers [8] can make it less expensive, easier, safer (also in legal terms), interoperable and more productive for companies to discover, aggregate, consume and extend Cloud services, particularly when they span multiple, diverse Cloud services providers in different EU Member States.

The conceptual architecture of this intermediator, ACSmI (Advanced Cloud Services meta-Intermediator) conceptual architecture is shown in Figure 1. ACSmI tries to re-conciliate both the Cloud Provider perspective and the operator of the multi-cloud application perspective providing a set of supporting functionalities during the Cloud Service lifecycle, for Cloud Service Initialization, Cloud Service Operation and Cloud service Termination. These are further discussed in chapter 7.

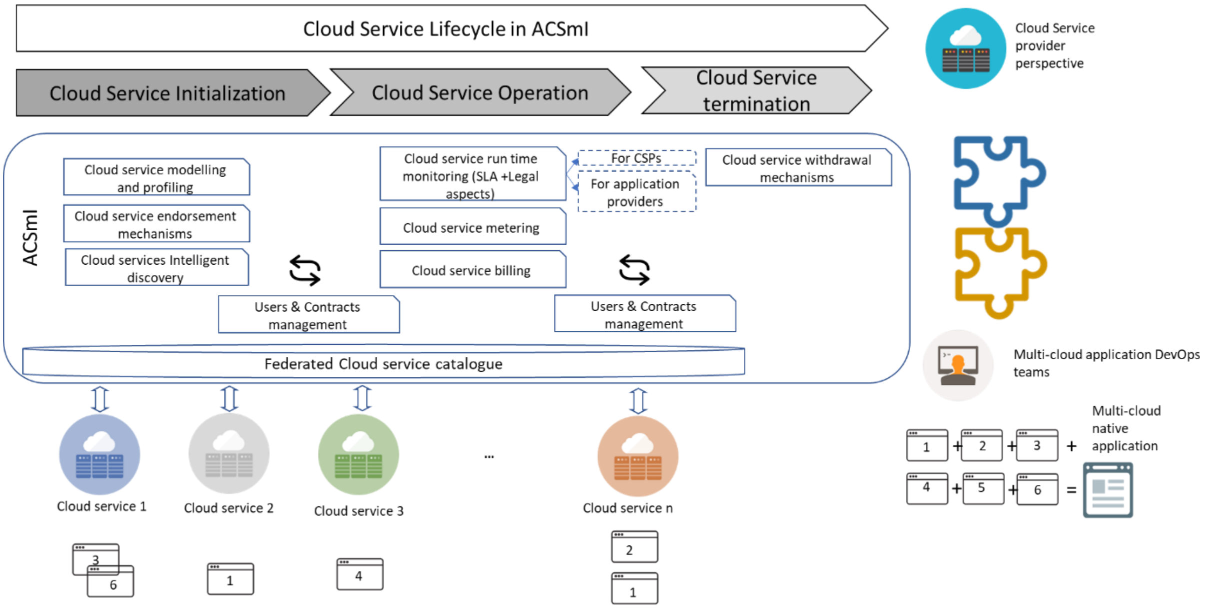

Cloud Service Lifecycle in ACSmI
Cloud Service Initialization
Cloud Service Operation
Cloud Service termination
ACSmI
Cloud service modelling and profiling
Cloud service endorsement mechanisms
Cloud services Intelligent discovery
Users & Contracts management
Cloud service run time monitoring (SLA +Legal aspects)
Cloud service metering
Cloud service billing
For CSPs
For application providers
Cloud service withdrawal mechanisms
Users & Contracts management
Federated Cloud service catalogue
Cloud service 1
Cloud service 2
Cloud service 3
...
Cloud service n
Cloud Service provider perspective
Multi-cloud application DevOps teams
Multi-cloud native application
1 + 2 + 3 +
4 + 5 + 6 =

Figure 1. *Motivation for the proposed research work. Source: Author's own contribution*[8].

With so much activity implementing front-end and back-end applications in public, private and hybrid clouds, complexity has grown at every level (business, application, transaction and regulatory).

To generate meaningful results, it is envisioned that enterprises need to address key challenges in the next years [9]:

1. **Governance**: Ensuring that services deployed in the cloud are protected is critical. Sharing can create leaks that cannot be tolerated. Promoting solid governance programs in place will protect enterprises and their data.
2. **Risk tolerance**: Every enterprise should assess their tolerance for pitfalls such as lost data and application outages. As Information as a Service and Integration as a Service evolve, organizations will reduce the risks.
3. **Regulations**: Lobbying for regulations and standards are predicted to be a key step to ensure cloud integration.
4. **Cross border interoperability**: The resulting service intermediator shall support intelligent discovery, context-aware service management and fluid service integration, assuring data portability in such a federated ecosystem, while guaranteeing proper identity propagation with service-specific granularity level of information.
5. **Matching customer requirements** with cloud service specifications: security, legislation awareness and other non-functional requirements need to be fulfilled when using any cross-border service deployed in an heterogenous cloud environment. This implies that the selected service offerings must match with every functional and non-functional requirement coming from the customers.
6. **Legislation compliance**, defining means of assuring service compliance with legislation of EU countries: a service is legislation aware when the services are constrained by legal requirements, such as data privacy, data protection, data security and data location. Moreover, a big challenge in this concern is to provide the required means for assuring legislation compliance and update in a heterogeneous environment of composite services form different countries.
7. **Cloud service SLA assessment and monitoring**: monitor and control the diverse properties of utilized services, composite or stand-alone, at real-time, while also being able to provide

all the critical information for the appropriate reactions, when necessary, especially when SLA conditions are not fulfilled (e.g., elasticity, data localisation).

8. **Seamless change of provider**: enable to seamlessly change the service provider with automatic management of all the dependencies to avoid vendor lock-in and decrease the time needed to situations causing outage of the service.

A viable intermediator and federator of cloud services [7] can make it less expensive, easier, safer (also in legal terms), interoperable and more productive for companies to discover, aggregate, consume and extend cloud services, particularly when they span multiple, diverse cloud services providers in different EU Member States.

This memory has been divided into 11 different chapters: Introduction, Background to concepts and technologies, Research aims and objectives, Research Methodology, Analysis of current practice and technology, Results, System validation: Qualitative and quantitative viability, Conclusions and future work, References, and supplementary information.

"Chapter 1- Introduction" presents the work and justifies of the motivation for this research work.

"Chapter 2- Background to concepts and Technologies" introduces the main concepts and technologies that serve as the baseline body of knowledge for this Thesis.

"Chapter 3" presents the main goal pursued with this research work and the breakdown of this main goal into four secondary goals that will aid to achieve the main objective.

"Chapter 4" introduces the research and validation methodology that has been used to implement the proposed investigation.

"Chapter 5" details the research questions and related hypothesis that guided this thesis. It also provides an overview of the Thesis output and scientific contributions achieved.

"Chapter 6" analyses the Cloud computing against the existing cloud services intermediators solutions. This chapter also presents a review of the current state of practice with respect to relevant aspects for this Thesis. These include modelling and designing of multi cloud native applications runtime monitoring of cloud services, regulatory framework and certification of cloud services and legislation related to cloud services provision.

"Chapter 7" explains the two main scientific contributions of this thesis. On the one hand, it presents the characterization of the multi cloud native application concept and the analysis of its specific needs in the SDLC and SOLC, deriving on a proposition for an extended DevOps philosophy. On the other hand, it describes the implementation of ACSmI (Advanced Cloud Services meta-Intermediator), the proposed cloud service meta intermediator, including the analysis of Cloud Services lifecycle, the functional requirements and main functionalities and the technical design and architecture.

"Chapter 8" presents the quantitative and the qualitative evaluation of the ACSmI. The quantitative evaluation measured the performance and accuracy of ACSmI discovery. The qualitative evaluation was carried out by a group of DevOps experts who were part of the requirements analysis process.

"Chapter 9" concludes the thesis by summarizing the major outcomes of this research. The chapter also points to the different directions in which this research can be continued.

"Chapter 10" lists the references in the Thesis.

"Appendixes" (Appendix A and Appendix B) provide supplementary information of the ACSmI technical implementation, i.e., the sequence diagrams of the Cloud service lifecycle in ACSmI, and screenshots of the main ACSmI features.

2. Background to concepts and Technologies

2.1 Cloud Computing

2.1.1 Cloud Computing fundamentals

According to NIST [10], Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

The NIST definition of cloud computing defines three delivery models, as shown in Figure 2:

- **Software as a Service (SaaS)**: The consumer uses an application, but does not control the operating system, hardware or network infrastructure on which it is running. Examples: Salesforce.com, Gmail, Google Apps, GotoMeeting, Run My Process.
- **Platform as a Service (PaaS)**: The consumer uses a hosting environment for their applications. The consumer controls the applications that run in the environment (and possibly has some control over the hosting environment), but does not control the operating system, hardware or network infrastructure on which they are running. The platform is typically an application framework. Examples: Google App Engine, Microsoft Azure, Salesforce.com.
- **Infrastructure as a Service (IaaS)**: The consumer uses "fundamental computing resources" such as processing power, storage, networking components or middleware. The consumer can control the operating system, storage, deployed applications and depending on the specific provider the consumer may control networking components such as firewalls and load balancers, even select the characteristics of the virtual image. Examples: Amazon EC2, Mozy, Nirvana.

Cloud computing refers to both hardware and systems software in the data centres that provide services and the applications delivered as services over the Internet. Those services have long been referred to as Software as a Service (SaaS). In fact, SaaS is a model of software deployment where an application is hosted as a service outside of the customer’s site and delivered to customers across the Internet.

A successful SaaS application, unlike any other traditional application, is built as a single instance but multitenant and shared among multiple customers on a common infrastructure (hardware and software). These architectural considerations added to other key functional, scalability, security and support requirements are not addressed in traditional software development, and thus need to be considered while adapting traditional applications architecture to SaaS.

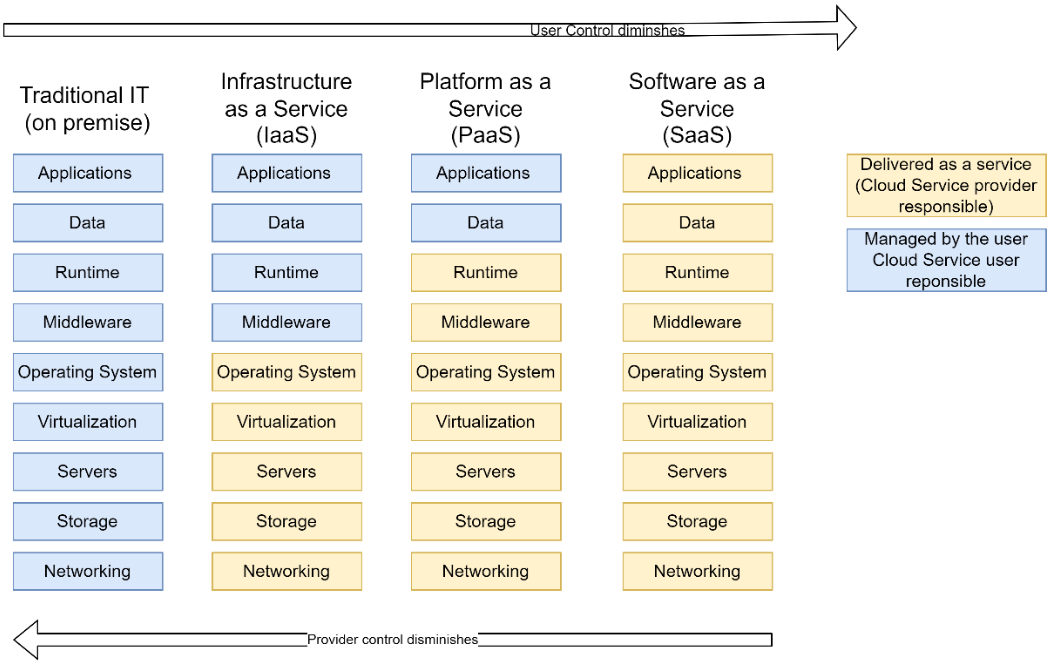
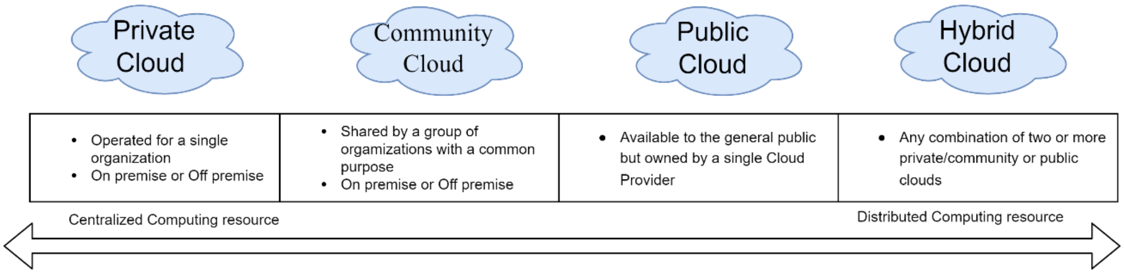

User Control diminshes
Traditional IT (on premise)
Infrastructure as a Service (IaaS)
Platform as a Service (PaaS)
Software as a Service (SaaS)
Applications
Data
Runtime
Middleware
Operating System
Virtualization
Servers
Storage
Networking
Delivered as a service (Cloud Service provider responsible)
Managed by the user Cloud Service user reponsible
Provider control disminishes

Figure 2. *Cloud Computing models. Source: Author's own contribution inspired from* [11].

2.1.2 Cloud Computing deployment models evolution

There are four deployment models (Figure 3) for Cloud services, also according to NIST [12]; these are private, community, public, and hybrid. A deployment model indicates the attributes associated with Cloud services especially the access attributes.

- **Private Cloud Services**: The Cloud infrastructure is used and operated by the same organization. This is a highly trusted and secure model as in most cases the user infrastructure is based locally within the organization. The disadvantages of this model include lack of elasticity, i.e., increasing or decreasing the size of the Cloud on-demand.
- **Community Cloud Services**: The Cloud infrastructure is shared and operated by a group of organizations, with all supporting policy, security and operations (i.e., data centers).
- **Public Cloud Services**: The Cloud infrastructure is available to the public or business for use. This is owned by a large organization and is the most common form of Cloud deployment. Large organizations such as Amazon, Microsoft and Google offer this form of Cloud.
- **Hybrid Cloud Services**: The Cloud infrastructure is a combination of two or more types of Clouds. This model requires the sub models to be bound by standard set of communication rules. An example of this would be a community Cloud working with a public Cloud to handle untimely surge in resource demand.


Private Cloud
Community Cloud
Public Cloud
Hybrid Cloud
Centralized Computing resource
Distributed Computing resource

- Operated for a single organization
- On premise or Off premise

- Shared by a group of organizations with a common purpose
- On premise or Off premise

- Available to the general public but owned by a single Cloud Provider

- Any combination of two or more private/community or public clouds

Figure 3. *Cloud Computing deployment models. Source: Author's own contribution.*

The different Cloud Computing models also describe the evolution from centralized resources to distributed resources.

2.1.3 Cloud Federation and management

The progress of Cloud Computing and related models (IaaS, PaaS and SaaS), has enabled to transition from "an era in which underlying computing resources were both scarce and expensive to an era in which the same resources started to be cheap and abundant" [13] .

Today, public cloud providers have permitted the commoditization of computing services in a promise of infinite cloud resources. In this ecosystem of cloud offerings, new alternative models, such as the federation of Cloud services start to emerge with the objective of supporting interoperability and portability between different providers and avoiding the vendor lock-in.

The topic of Federated Cloud has been an area extensively studied in the Cloud research community. Federated Cloud solutions and approaches such as Hybrid Cloud and Multi-clouds are today gaining momentum both in commercial and community set-ups. Examples of these are Hybrid Cloud Solutions in the market, from vendors such as VMware and more recently AWS Outposts, IBM/RedHat OpenShift or Google Anthos among others [14].

Hybrid Cloud is one of the simplest models of the Federated Cloud, where workloads can be switched from one execution environment to another (typically form private to public clouds).

In more advanced federated cloud scenarios, a cloud provider is able to access additional capacity from other providers when needed. It can also offer other added value services on top of the classical ones such as aggregation, enhanced elasticity, resilience, and fault tolerance. From a user perspective, part of the heterogeneity remains transparent to the final consumer while other (such as the QoS monitoring) is still managed by each of the individual service providers.

The federated cloud scenario is often related to community cloud set-ups. The notable examples in the eScience community in Europe is represented by the European Open Science Cloud (EOSC) [15].

Federated Cloud can also provide advantages to cloud providers that own multiple cloud sites (for instance, in diverse geographical regions) with the purpose of easing the management of the diverse cloud islands and balancing workload among them. But more significant is the benefits that the federated model can provide to groups of providers to offer communally services to final users, including or not a mediation broker. This is the model envisioned by GAIA-X [16], that aims to create a network of European federated high-performance, competitive, secure and trustworthy Cloud services, involving all the potential (small and medium) European Cloud services providers.

GAIA-X is a European initiative which is a very first proof of concept for the design of the European cloud. GAIA-X stands for a federated high-performance, competitive, secure and trustworthy data and Cloud infrastructure for Europe. It pursues to provide a data infrastructure as a federated technical infrastructure consisting of components and services that make it possible to access data and to store, exchange and use it according to predefined rules. The main goals they want to achieve with such an initiative are: data sovereignty, reducing dependencies from non-European (Cloud) providers, broader adoption of Cloud Computing by the (European) SMEs, creation of an open, digital ecosystem to help European companies and business models to scale up competitively worldwide.

Project GAIA-X aims to connect centralised and decentralised infrastructures (in particular cloud services and edge services) into a homogeneous, user-friendly system. Therefore, GAIA-X establishes an ecosystem in which data is made available and shared in a trustworthy environment. The users always retain sovereignty over their data through this federated system that links many cloud services providers and users together. This ecosystem also takes into account various preferences regarding security aspects, latency times and the breadth of application; it supplies tailor-made solutions and enables the use of various cloud providers.

Such data-infrastructure ecosystem (Figure 4) consists of joined interconnected data and infrastructure ecosystems, aggregated as 'Federators' by the concept of a GAIA-X Federation, and individually orchestrated and operated by a set of 'Federation Services' [17]:

- The data ecosystem supports data spaces based on data exchange with agreed rules. Data and service offerings can be transparently shared across different industry players in different sectors. These so-called Federations promote the creation of new advanced services thanks to the increased amount of data and information that a Federation provides with regard to any individual Participant.
- In the infrastructure ecosystem, the offerings from different providers are interoperable, interconnected, and compliant to certain rules. Providers can collaborate linking with each other, to further develop and expand their offerings in terms of catalogues, performances, regions, and critical mass.

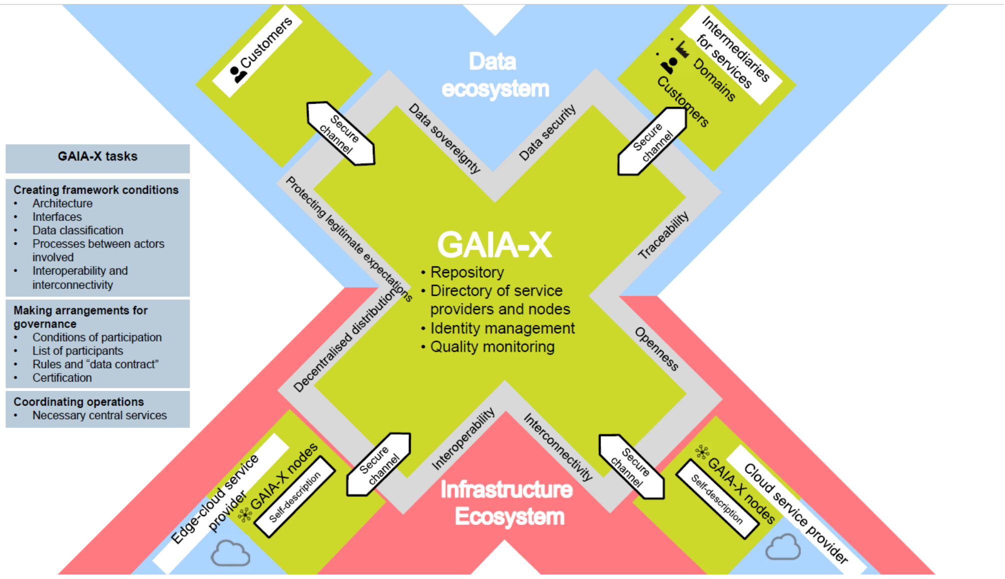


Figure 4. *Gaia X conceptual architecture (source* [18]*).*

Some of the envisioned benefits of the Cloud Federated model are [14] :

For the providers:

- Benefit from the long tail business model: while each CSP may have several big clients for whom the infrastructure is set up, smaller companies may benefit from the unused, but available, infrastructural resources, providing the CSP with smaller, but continuous, income.

- Continuous innovation to remain competitive.

For the consumers:

- More offerings to select based on their own requirements, for instance, cost, availability, a certain 'legal' level.
- If the federation follows a rigorous approach to endorse services in the catalogue of federated services, the consumer can feel that their data and applications are more secure.
- Optimized management of cloud services.

Nevertheless, the realization of a fully federated Cloud vision is still an unsolved challenge as several additional aspects need to be developed : collective provisioning, contracting, metering and billing; across cloud privacy, security and identity management; fine grained cloud service agnostic QoS and composable Cloud Service Level agreements; secure mechanisms for data sharing; consideration of diversity and heterogeneity of resources at all levels (compute, data and network); cloud service portability (at resource level and at application level) as well as, adoption of existing Cloud standards and regulations.

Some of the barriers for federation include [14]:

- Federation is based on voluntary basis
- Different levels of coupling in federations
- Lack of trust because the on-boarding process is not open, and it is not sure that the services on-board follow existing accreditations and have a compliance check
- Interoperability and portability aspects among different services are not ensured

Only with the complete development of these novel capabilities, the Cloud Federated model could be fully adopted, and the industry will take advantage of all the upgrades promised though this model, raising new business opportunities both for existing and new cloud stakeholders. A sustainable Federated Cloud model will allow enterprises and public sector to materialise the Cloud promise of a complete hybrid provisioning and soon the realization of the Cloud Continuum with the incorporation of infrastructural elements of a new nature (i.e., network elements, sensors, etc.) across the full Cloud Continuum, embracing set-ups for large data storage, High Performance Computing intelligent analytics and Edge for IoT.

Indeed, the movement towards hyper distribution of the computing continuum will also involve the progress on advanced federated cloud and edge computing solutions and related techniques such as : self-organisation, self-management and self-healing across many and heterogeneous resources present in all kinds of IoT Edge devices, micro edge data centres, private enterprise clouds, federated cloud models and large Cloud set-ups. Orchestration, services brokerage and workloads distribution problems in this context will need the development of novel management tools, programming models and approaches [14].

2.2 Cloud Continuum

Cloud Computing evolution in the last decade and its transformation into a service utility has promoted a wide adoption by the industry for applications in general to store and process data [19]. With the expansion of the IoT paradigm, the need of computational and storage services is expected to grow in the next years as well as the quantity of data generated at the edge of the network. Up to now, cloud computing has been effective to compute a store in a "as a service" approach software applications. Nevertheless, it may not be appropriate to manage the endless data from IoT devices and fulfil all the application requirements. To this extent, some of the limitations of the traditional Cloud Paradigm specially applies to applications that need real time, low latency or those giving support to critical infrastructures. The centralized nature of the traditional cloud services poses some limitations as communication and data transfers need to move over multiple leaps, introducing delays and increasing the consumption of network bandwidth at the edge and in the core networks. The edge has increased its computing capacity with the hardware evolution of personal devices, and it has been proposed to run applications and store data, bringing a new player: Edge Computing.

As a result, new approaches that effectively leverage distributed computational and storage infrastructure and services are necessary. These approaches must seamlessly combine resources and services at the edge (edge computing), in the core (cloud computing), and along the data path (fog computing) as needed, through the Cloud continuum (Figure 5).

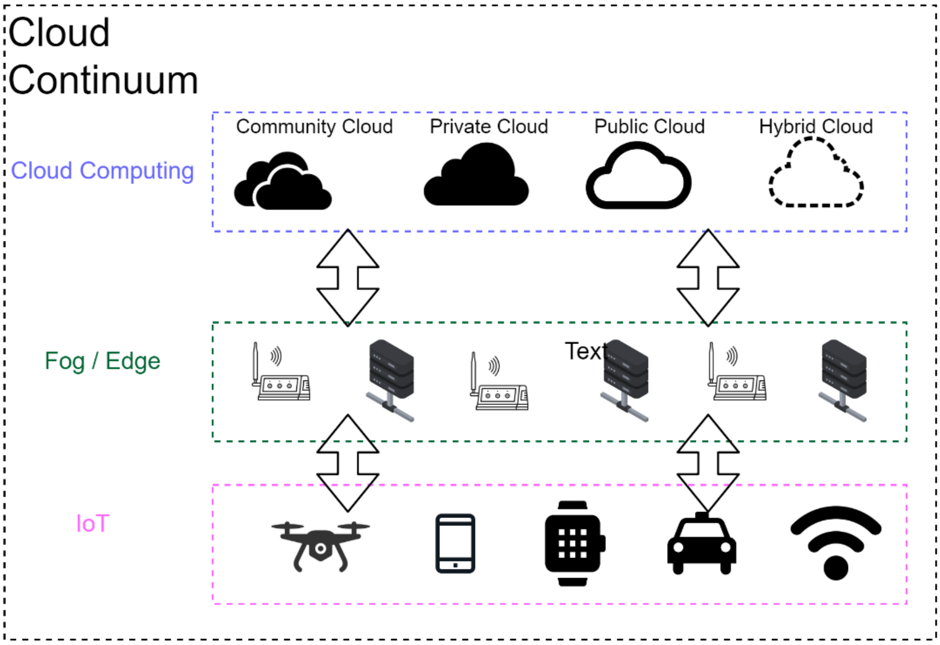


Figure 5. *Cloud Continuum paradigm. Source: Adapted from* [20].

Thus, for application developers and operators to fully embrace this new paradigm, specific and in-depth knowledge on the plethora of underlying techniques and technologies is needed. Furthermore, the operation of such heterogeneous computing environments poses complex tasks for the operators of the applications as they must configure, plan, prepare and execute them, facing new challenges in all the stages of the operation phase of the application.

Fog computing enables pervasive access to a continuum of computing resources. The model is composed of physical and/or virtual nodes located between the cloud services and the end-devices [21].

Usually, these nodes are context aware and are organized in groups or clusters both vertically (supporting isolation) or horizontally (supporting federation).

Edge computing [22] is the network layer encompassing the devices and their users, providing local computing capability on a sensor, metering, or some other devices that are network accessible. This peripheral layer is also often referred to as IoT network. Fog computing is often erroneously confused with edge computing, but there are key differences [14] between the two concepts. Fog computing runs applications in a multi-layer architecture that decouples and meshes the hardware and software functions, allowing for dynamic reconfigurations for different applications while performing intelligent computing and transmission services. Edge computing runs specific applications in a fixed logic location and provides a direct transmission service. Fog computing is hierarchical, whereas edge computing tends to be limited to a small number of peripheral devices. Moreover, in addition to computation, and networking, fog computing also addresses storage, control, and data-processing acceleration. It is rapidly growing [23] and represents the major trend in the distributed computing area. Edge computing [24] provides novel forms of computing acquiring computing power and data resources at the Edge of the network. It accelerates the evolution of traditional Cloud computing environments to decentralised environments supporting IoT approaches. The alternation of decentralized models and traditional ones is fostering Cloud Federation to advance upon this objective [22].

At the same time Edge computing is evolving in many different areas being IoT scenarios one of the most prominent one. Even in this simple scenario, many challenges remain in relation to the optimal workload encapsulation, service placement, latency, networking, security, and privacy. To address these, IoT deployments need to be combined with other technologies such as AI (Artificial Intelligence) and federated machine learning to extract and manage huge volumes of generated data generated data at the Edge of the network.

2.3 (multi-) Cloud native applications

A Cloud Native Application is a piece of software which has been specially implemented for virtualized environments or cloud computing platforms [25] [26]. Therefore, these applications should make optimal use of the services provided by virtualized infrastructures.

In general, cloud native applications have these characteristics [25]:

- They can execute different parts of the application simultaneously in different resources from diffrent locations composed of a large number of parallel processing units.
- Full and optimal use of cloud resources through the usage of application programming interfaces (API) and other methods to simplify management tasks and efficiently use most of the available resources (computing, memory and storage).

The cloud-native approach is based on developing and running applications that take full advantage of the (multi) cloud computing service model. The importance lies in how the applications are built and deployed, rather than where they are executed. In this way, developers can leverage virtually unlimited computing resources as needed. Companies that develop and deploy cloud-native applications can bring new products to market faster and meet customer demands faster.

In this situation, the need platforms that automate and integrate concepts such as DevOps, continuous delivery, microservices and containers is a requirement for IT intensive companies.

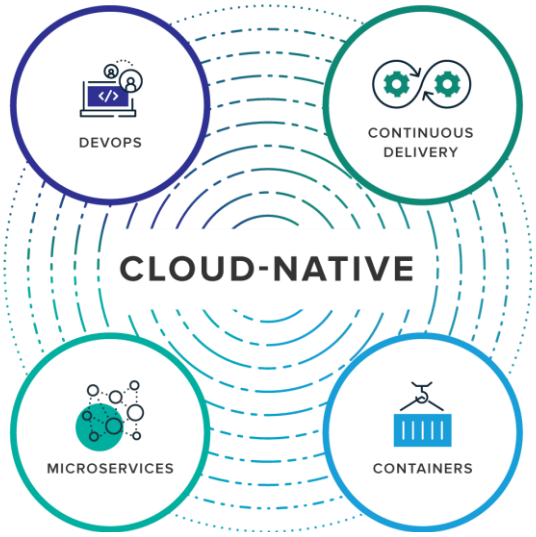


Figure 6. *Fundamentals of Cloud Native applications (source* [27]*).*

Moving one step forward, multi-cloud native applications are gaining great interest in the software industry. The value proposition offered by multi cloud native applications to the industry is well known, enabling the implementation of complex IoT solutions with better performance and empowering organizations to distribute their workloads across multiple cloud environments with optimized RoI[1]. Every resource is optimized for each application component, functional and non-functional needs such a low latency, real time, high processing requirements, superior security, or autonomy (less vendor lock-in), among others. Thus, companies conclude that multi-cloud accelerates innovation, enhances data agility, and reduces costs. Some examples where the application of multi-cloud native applications is gaining popularity are entertainment and Media (i.e. Netflix), energy sector, autonomous driving or e-health [28]

Another sign of the importance of the multi-cloud paradigm for the software industry is the appearance of new standards and industrial frameworks which aim to set up the multi-cloud concepts and practices. As example, the ISO/IEC JTC 1/SC "Cloud Computing and Distributed Platforms" [29], which is still under development, and which provides an overview and foundational concepts for cloud computing involving multiple cloud service providers (CSPs). Again, these draft points out several benefits for the industry while adopting multi cloud-based solutions. These include more flexibility, higher availability rates, resource-based optimization, better fault tolerance, decreased latency, less costs or enhanced privacy.

After outlining the concepts of cloud native applications, we summarized in [27] the following list presenting the advantages of embracing the paradigm of multi-cloud applications..

- *"Minimise the risk of widespread data loss and keep at a minimum downtime. Cloud platforms are very complex computing environments that inherently have multiple points of failure: hardware, software or infrastructure. Due to the use of two or more cloud services, it is possible to ensure SLAs agreed with customers.*
- *"Improve the overall performance of organisations. Multi-cloud strategies promote the use of open source and standardised technologies, avoiding vendor lock-in with proprietary solutions. The diversity of multi-cloud ecosystems facilitates compliance with the requirements of a wider range of partners and customers"*
- *"Impact directly on customer satisfaction. The speed with which a particular website loads its content is strongly linked to the satisfaction of end users. Websites with faster page loads usually have more frequent and longer visits, so search engine rankings are also affected. A multi-cloud deployment can help significantly reduce the load time of an organisation's web applications.*
- *Optimise the traffic of customers and partners. In addition to providing the redundancy required to efficiently reduce fault tolerance, in a multi-cloud environment it is also possible to route traffic based on the type of client, the content to be distributed, and the nature of the application. For instance, some CSPs offer servers and networks best suited to handle large numbers of requests requiring small data transfers, while other CSPs have a portfolio which performs best for smaller numbers of requests with larger data transfers"*

Just as some of the advantages of multi-cloud applications have just been presented, the challenges of this forward-looking paradigm are explained in [27]:

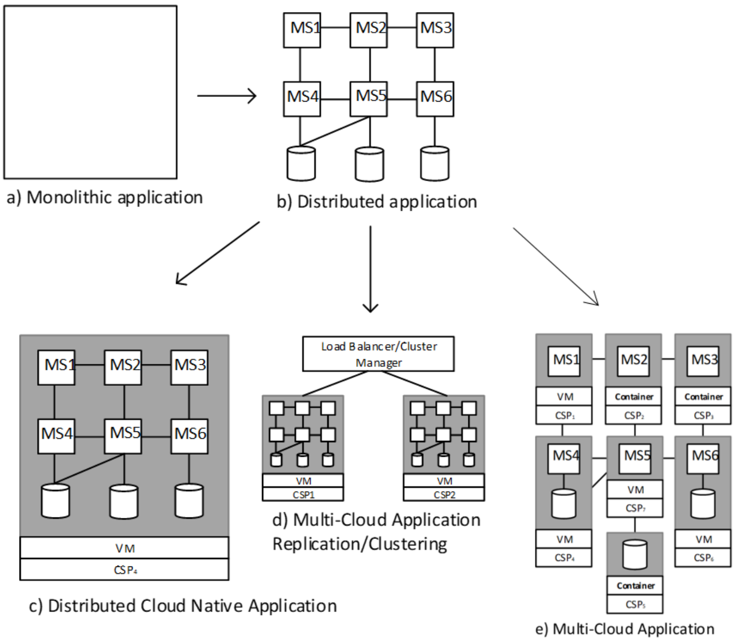
- *"Limited interoperability. There is currently no complete interoperability between different cloud providers. This forces developers to use workarounds to successfully deploy applications on different platforms and clouds.*
- *Greater complexity. This is the biggest challenge of multi-cloud applications. Developers and administrators have to deal with different interfaces, technologies and services. There are currently neither standardised terminologies nor methodologies across cloud providers.*
- *More workload. Implementing a multi-cloud environment brings a greater workload for developers and DevOps teams. It first takes longer to select the right services to use from each provider. They*

[1] ROI: Return Of Investment

also have to learn how to integrate their applications with the different infrastructures and APIs available. In fact, it is sometimes necessary to maintain specific source code versions for each provider. Once applications are in production, it is more complex for DevOps engineers to manage and maintain their performance across multiple clouds.

- *Difficulty to estimate costs. The flow of application data into or out of the infrastructure of each cloud provider generates costs which are difficult to calculate accurately. It is required to conduct an in-depth review of the pricing structure of each service used to come up with an approximation of the overall cost."*

Figure 7 depicts the evolution in software development and deployment architectures.



a) Monolithic application

b) Distributed application

c) Distributed Cloud Native Application

d) Multi-Cloud Application Replication/Clustering

e) Multi-Cloud Application

Figure 7. *Evolution in software development and deployment architectures. Source* [30].

This figure shows the big difference between native cloud applications and native multi-cloud architectures and developments. It starts by showing a monolithic application (a), (b) represents a distributed microservices-based application in the traditional sense. The traditional distributed application can be deployed in the cloud through different approaches c), d) or e). C) shows a cloud native application based on microservices, d) shows a distributed application replicated or scaled across two CSPs, and finally e) shows the architecture and deployment of a multi-cloud application.

2.4 Legal challenges and relevant standards

Software companies can choose from deploying their applications in one single private cloud to a more complex approach where a combination of different service offerings, from the same or different CSPs, is used for the deployment of the application. The decision whether to use one approach or another, or an hybrid approach, is driven by certain evaluation criteria (e.g. profitability, reliability, performance, security, legal or even ecological aspects).

A big challenge in this concern is to develop the methods and interfaces for assuring legislation compliance and facilitate legislation change propagation in a legislation heterogeneous environment. The impact and importance of the legislation has become even more relevant with the EU's General Data

Protection Regulation (GDPR) having gained full application in May 2018 [31] . This aspect acquires even more importance when considering heterogeneous environments involving multiple cloud providers and all the subsequent implications (multiple contracts management, multiple SLAs management, etc.).

A multi-cloud native application needs to take into consideration many legally relevant aspects of the underlying cloud services during its development and operation. Such aspects might relate to data location, data protection requirements, security level, accountability and control, confidentiality, and contractual terms such as liability and exit clauses. They will depend on the contractual terms and SLAs offered by the cloud Service Providers concerned. All of these relate to typical legal challenges in a Cloud environment [32–34], but become even more important as the Cloud resources (and CSPs) are being multiplied, resulting in multiple contracts being managed for a single application.

The impact of these aspects on the software development and operation lifecycle is such that they function as a legal NFR for the multi-cloud native application as a whole. Non-functional requirements are requirements set during the development of an application, which are maintained throughout its operation lifecycle. The requirement has the effect of enabling the inclusion of certain cloud services or justifying the exclusion of certain other services, not based on any functional technical necessities, but purely based on compliance with the legal aspect at issue.

Data location for example can be a relevant legal requirement for part or the whole of an application. While the EU's proposed Regulation for a free flow of non-personal data [35] aims to remove all legal obstacles in Member State law which prevent full use of the cloud by restricting data location to the own territory, nothing prevents the owner of an application to introduce location requirements in its multi-cloud application nonetheless. Moreover, with regards to applications containing personal data location restrictions may equally persist. The software development and operation lifecycle has to accommodate this requirement throughout the lifetime of the application, no matter how many times the application is re-deployed with different CSP services, with different configurations, being used to run the application as a whole.

Similarly, other legal aspects will be translated into non-functional requirements for the whole of an application, aimed at pre-selecting suitable cloud services.

Another prime example of a legal aspect that needs to be accommodated is data protection. Under the EU's new GDPR [31], applications processing personal data have to follow the principles of data protection by design and default, which entails that every application should be designed and developed with respect for data protection principles and should have default settings ensuring the greatest level of privacy (Article 25 GDPR [31]). Let it be noted that nearly any application will be processing personal data in one form or another. The GDPR will therefore apply in most cases which means that practically all software developers will need to make this exercise for themselves, ensuring that the application satisfies this principle. But data protection will also have an impact on the level of the underlying cloud services of a multi-cloud native application. After all, according to the GDPR, the application as a whole has to be secure, and may only be provided using the services of CSPs that provide sufficient safeguards in relation to data protection and are bound by a specific data processing agreement containing all legally required clauses (Article 28 GDPR).

However, unlike data location, which although not always easy to verify, is clear and precise as a requirement (e.g., in/outside European Economic Area, in any specific country, ...), the requirements imposed by the GDPR are rather open to interpretation to start with. Article 28 GDPR requires that the application developer should only use CSPs in deploying the application, providing enough safeguards to implement appropriate measures to support the GDPR requirements. Moreover, the data processing agreement with such CSPs should obligate the CSP to take all the measures to ensure a level of security appropriate to the risk. In other words, the GDPR requires an application developer to assess the sensitivity of the application and the data processed therein, and to determine the corresponding level in terms of security (both technical and organizational), required for the application.

These data protection requirements quite closely relate to security requirements in general, which are of legal relevance, at least in terms of liability scenarios. Equally important are the measures in place in relation to accountability and control, confidentiality, access management etc. Although these are all largely covered by the GDPR's requirements imposed on CSPs by virtue of their role as processors under that legislation, they retain a standing of their own as legally relevant.

Another legal aspect that may be useful for application developer to set relates to liability clauses and exit (penalty) clauses. Application developers could for example well prefer to only use CSPs that do not have penalty clauses in their contracts, obliging the developer to have a continued relationship with the CSP for a certain period of time or a given volume. Large CSPs typically do not have such clauses in any case but smaller CSPs, often providing bespoke services might have these requirements.

However, generally, the challenge for any software development framework in relation to the legal aspects of the underlying cloud services, is to assess the complex legal reality of the cloud service's SLA, the security level at the CSP, etc. and to reduce it to machine-readable non-functional requirements. It presupposes the definition of easy-to-answer questions or the definition of simple metrics through which one can capture the legal essence of this complex situation. This must be understandable for both the CSP and the application developer, so that a match can be made between the expectations of the developer and the services offered by the CSP.

2.5 Cloud SLAs (CSLAs)

A CSLA (cloud service-level agreement) is an agreement between a cloud service provider and a customer that ensures a minimum level of service is maintained that includes cloud service level objectives (SLOs) and cloud service qualitative objectives (SLQOs) for the covered cloud service [36] .

CSLAs are usually described unilaterally by the service providers and thus, cloud providers publish the functional and quality aspects of their services by means of SLAs described in natural language. However, the SLAs could also be agreed between the interested parties in dynamic or adaptable scenarios.

From the perspective of cloud providers, the delivery of cloud services is not a trivial task since it usually requires provisioning an approximated number of the actual required cloud resources to satisfy consumer's demand. An accurate estimation of such resources would lower the resources provisioning cost. However, such estimation is an error prone task that depends on: 1) the number of users, 2) their fluctuating demand, and 3) the Quality of Service (QoS) offered.

On the other hand, cloud services consumers need to trust the metrics and information provided by the cloud providers with respect to the assessment of the contracted SLOs. Usually, cloud vendors provide their own tools to monitor and assess the accomplishment of such contracts. This situation becomes even more complex when a single user needs to assess the CSLA of several cloud services used to deploy different components of the same application. This is crucial when addressing multi-cloud applications, for which the composed Multi-Cloud SLA (MCSLA) [37] is based on the composition of the underlying Cloud services SLAs' on which the different components are deployed. The MCSLA can act as the contract between the end-users and the developer of the multi-cloud native application, and it needs to be assessed at runtime. The fulfilment of such MCSLA depends on the individual cloud services contracted and their own CSLAs.

An MCSLA must therefore act as an aggregator of all terms defined in the various SLAs (Figure 8).

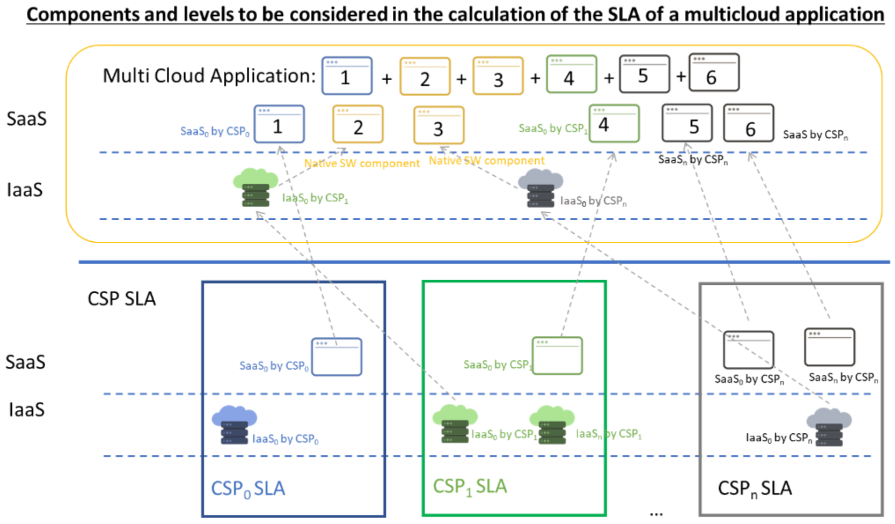


Figure 8. *Conceptual Idea – Make up of an MCSLA. Source: Author's own contribution.*

3. Research aims and objectives

3.1 Main goal

The main goal of this Thesis can be stated as follows:

Research the means, drivers, risks and barriers of multi-cloud native applications development and operation as well as implement a solution for the ***re-use and combination of*** **cloud services,** ***for assembling a network of interoperable, legal compliant, quality assessed (against SLAs) single and composite cloud services for multi-cloud aware applications*** *deployment and operation*.

To achieve this goal we aim to research, analyse, design, and develop an Advance Cloud Service Intermediator (ACSmI) that supports the discovery, aggregation, and consumption of cloud service functionalities for multi-cloud applications. This aim can be broken down into:

1. Definition and implementation of the multi-cloud aware applications concept
2. Analysis and provision of mechanisms to discover and select a combination of cloud services specific for multi-cloud aware applications
3. Research and provision mechanisms to assess continuous real time verification of the cloud services non-functional properties fulfilment (Composite CSLA) including legal aspects
4. Study and development of means for seamless change of Cloud service provider enhancing the portability and interoperability of multi-cloud aware applications.

3.2 Specific goals

3.2.1 Definition and characterization of the multi-cloud native applications

The objective is to analyse, describe and characterize multi-cloud native applications that will ease the design, development, optimization and deployment of multi-cloud native applications.

This characterization can be made through the understanding and analysis of the software engineering practices covering the whole application lifecycle. From development to operation and runtime. **The proposed characterization of the multi-cloud native applications will allow the improvement of design, development and operation of distributed applications over heterogeneous cloud resources whose components are prepared to be optimally deployed on different cloud service providers (CSPs) and still, they all work in an integrated way and transparently for the end-user.**

In this work we consider a multi-cloud native application as a distributed application over heterogeneous cloud resources whose components are deployed on different CSPs and still, they all work in an integrated way and transparently for the end-user. There are several reasons for deploying an application in a multi-cloud architecture, the most important ones being: non-compliance of the CSPs to the agreed SLAs, avoidance of vendor lock-in, increasing reliability or improving other QoS concerns such as increasing performance or security, and finally, reducing costs. The application types that would benefit the most from such a multi-cloud approach are on the one hand, those that are critical to the business and that need to respond efficiently to the user’s needs in terms of performance, reliability, and security and on the other hand, complex applications whose components need to be distributed over different cloud providers due to their specific needs and requirements. Examples of these applications include: Network Management in extended multi-country scenarios with differentiated cloud layers, online videogames, Public Administrations online services, and travel agencies or ticket agencies (e.g. Ticketmaster [38]). However, any application offered as SaaS can benefit from a multi-cloud architecture. Currently, this is solved by deploying the same application on several cloud providers following a master-slave or active-passive approach. This, however, also poses several risks, since the synchronization of all the data is critical for a correct functioning of the application if no data loss is wanted. **The multi-cloud approach presented in this work tries to minimize synchronization risks and guarantee the fulfilment of the application providers' requirements, which can range from maintaining a constant cost structure to a certain response time, security issues or a certain performance level.**

Activities carried out:

- **Multi-cloud native applications' lifecycle study**: Analysis and study of current practices for the development and operation of multi-cloud native applications. The whole lifecycle of the application will be studied form the design to the operation and maintenance with focus on the activities to be performed for each of the steps considering the peculiarities of multi-cloud native applications. Current approaches in the context of software engineering and operation such as DevOps will be analysed and extended if needed.
- **Multi-cloud native applications characterization and analysis**: Characterization of the "multi-cloud" native applications in comparison to traditional legacy applications or even just "cloud-native" applications. This characterization should address the "multi-cloud" native application from different perspectives such as technologies, standards, processes, etc. used in the literature for their architectural definition, characterization and actual design and development. To this end, the study of different techniques, processes and technologies proposed in the literature to realize the concept of "multi-cloud" by design at application level is envisioned.
- **Current research gaps and challenges for the design, development and operation of the "multi-cloud" applications**. Study of the challenges that are currently faced by both the developers and the operators of "multi-cloud native" applications. The focus has been on the identification of the critical aspects experienced by the researchers and the practitioners over the application lifecycle, Software Development LifeCycle (SDLC) and Software Operation LifeCycle (SOLC). This ends up with the identification of problems and proposed solutions in the context of multi-cloud native application development and operation and at the same time derive unsolved issues that could divert into new research trends for further work.

3.2.2 Analyse and provide mechanisms to discover and select a combination of cloud services

The objective **is to provide means for the discovery, registration and management of cloud service providers and offerings into the Advanced Cloud Service meta-Intermediator**, so that these services can be published and accessible to be used. ACSmI will aggregate and intermediate not only resources provisioning services (HaaS) but also DBaaS, DPaaS. The goal is to provide **advance methods for intelligent Cloud service discovery based on a set of specific requirements set up by the end-user.**

Activities carried out:

- **Analysis and extension of machine-readable means to describe cloud service offerings**. Current initiatives and standards have been analysed and these have been mapped to the requirements of cloud services with the objective of providing novel language terms for describing services. These novel terms also consider attributes such as e.g.; location, legal level, performance or availability.

- **Research and provision of advanced cloud service exposure mechanisms to provide all the relevant metadata regarding the available intermediated services offerings and the necessary interfaces for registration, update and information retrieval**. The objective is to finally provide a cloud service catalogue, with both atomic and composite services (composition of atomic services) coming from public, private and federated service catalogues. By means of this facility and enabled by service's commutability (based on the service desciprtion), the service catalogue enables a multi-service packaging and a service pre-composition engine.

3.2.3 Research and provide mechanisms to assess continuous real time verification of the cloud services non-functional properties fulfilment (Composite CSLA) including legal aspects

The objective is to analyse and establish the necessary mechanisms to carry out the real time monitoring of the cloud services non-functional properties fulfilment in three layers, at physical, virtual infrastructure and application level, taking into consideration certain types of services and parameters such as legal and security ones, with the main aim of assessing theoretical SLA and QoS (e.g. performance) values, as provided by the CSP with respect to the ones resulting from these monitoring activities. This information will be, on one side useful to the users, but also internally so to detect violations in the agreed QoS.

The proposed solution includes means for monitoring and assessing that the aggregated and intermediated cloud offerings fulfil the corresponding SLA terms and conditions, including legislation and accreditation issues, security aspects and propagation of changes.

As a novel aspect, the dynamic accreditation of legal solutions (contractual and policy framework) has been investigated. The goal is to legally accredit services registered prior to the operation phase, so that the user interested in operating one specific service registered in the ACSmI knows the degree of fulfilment of the legal criteria. When the service is in operation, the ACSmI checks the validity of the accreditations according to the results of continuous assessment.

Activities carried out:

- **Analysis of existing CSLA of various CSPs and development** of a machine-readable format for the representation of CSLA.
- **Research and propose an approach to compose CSLAs of various CSPs to generate composed CSLAs** for multi-cloud native applications. For that, the ACSmI provides means for implementing the composition of CSLA and the definition of a concrete SLO, at development time and if feasible, at operation time with associated qualitative and quantitative metrics.
- **Analyse legal characteristics of the Cloud Service providers** and means to support their characterization and classification to a certain legal level.

4. Research aims and objectives

To be able to execute research, there are many research methods and data collection techniques available to follow in order to achieve the research findings. However, the selection of research method depends on problem in hand.

Therefore, this research adopts a well-known Design Science Research (DSR) methodology [39] which is a system of principles, practices and procedures applied to a specific branch of knowledge to produce and present high-quality research artifacts or outcomes [39]. The DSR aims to provide verifiable contribution through the development and evaluation of an artifact. The artifact development may involve the review of existing theories and knowledge with a view to develop a rigor solution or artifact for the intended purpose and audiences. The developed artifact is then evaluated using the DSR evaluation criteria.

In this work, we followed the six-steps of the DSR process (see Figure 9). The DSR process steps are: Problem identification, solution objective, design and development, demonstration and testing, evaluation, and communication.

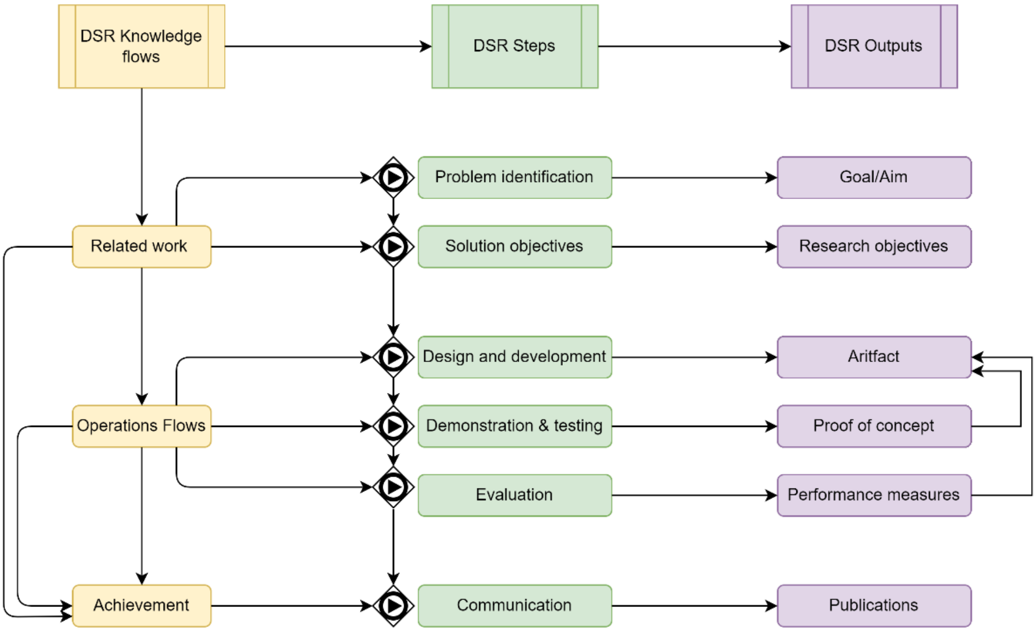


Figure 9. *DSR method process. Source: Adapted from* [39] **.**

The first step identifies the research problems based on the research gaps highlighted in chapter 1 of this Thesis. Based on the literature review and related work (chapter 5), it has been found that there is a need of a viable intermediator and federator of cloud services broker that can make it less expensive, easier, safer (also in legal terms), interoperable and more productive for companies to discover, aggregate, consume and extend cloud services specially in the context of multi-cloud native applications. Hence, the main research question is: It is possible to demonstrate that the proposed ACSmI can contribute to the creation of an ecosystem of trusted, interoperable, and legal compliant cloud services fostering the uptake of cloud computing, with a special focus on multi-cloud aware applications that have specific non-functional requirements and needs?

The output of this step ends up with the definition of the main goal of this Thesis as described in Chapter 3, section 3.1.

The second step of DSR further investigates the identified problem and provides motivation and aim of the research. The main aim of this research is to develop a DRA (Design Research Artifact) to enable the proposed ACSmI. As a result, the output of this step constitutes the set of specific goals defined in section 3.2.

The third step focuses on developing contextual, conceptual, logical and physical models of DRA for ACSmI:

1. DRA Contextual Model
2. DRA Conceptual Model
3. DRA Logical Model
4. DRA Physical Model

All these models as part of the technical ACSmI technical design described in section 7.3. Besides, during the implementation of the third step and as a result of the preliminary contextual and conceptual models of the ACSmI, the extended DevOps concept was proposed and developed (the other main contribution of this Thesis).

The fourth step focuses on demonstration and testing of DSR artifact (DRA). The demonstration and testing is composed of the activities to implement the first Proof Of Concept including the testing activities. This step has been iteratively performed across 3 versions of ACSmI. The testing results have been derived into new/modified requirements for ACSmI.

The DRA evaluation step covers the qualitative and quantitative validation of the proposed DRA and is based on the evaluation framework proposed in [40] . It includes the evaluation of a none instantiated artifact (such as the design or model), the evaluation of the artifact in artificial settings (in this case using the "Sock Shop application") and the naturalistic evaluation of the artifact as explained in section 7. Quantitative evaluation involves measurement and analysis of variables between methods. This is used for evaluating similar systems for efficiency and accuracy. This evaluation process is objective and based on figures obtained. Qualitative evaluation is the set of activates used for observing the behaviour of a system. This is undertaken by a group of experts in the field. This method of evaluation is subjective as it is based on the group's observations. In the context of this research a group of DevOps team members were available for performing this evaluation.

Finally, the DRS process finishes with the communication step which includes the dissemination and communication activities to spread the different results and achievements of the different phases of the process of this research work.

5. Research questions and hypothesis

The major research question and hypothesis that this thesis pursues is presented below:

RQ- How can an adequate cloud broker intermediator for multi cloud native applications be realized to re-use and combine cloud services, for assembling a network of interoperable, legal compliant, quality assessed (against SLAs) single and composite cloud services for multi-cloud aware applications deployment and operation?

H- It is possible to demonstrate that the proposed ACSmI can contribute to the creation of an ecosystem of trusted, interoperable and legal compliant cloud services fostering the uptake of cloud computing, with a special focus on multi-cloud aware applications that have specific non-functional requirements and needs

This hypothesis can be broken down into the following ones:

- *RQ1- Is there a common understanding of the term multi-cloud from the application perspective?*
  H1- It is possible to define the multi-cloud concept from the application SDCL and SOCL perspective and demonstrate its validity in real use case scenarios.

  Multi-cloud concept as referred to applications that can dynamically distribute their components (pieces of the application, snippets of code) over different cloud services and still hold the functional, business and non-functional properties (NFP) declared in their SLAs. The real use cases where to demonstrate the validity will be High availability (energy management), eHealth (Clinical Research Platforms) and Network management.
  These use cases have been selected bearing in mind the requirements and needs of those type of software applications that need to be compliant and aware of relevant legislation, and need to fulfil different non-funtional properties (i.e., cost, performance, availability). Along with the multi-cloud concept, the Multi Cloud Service Level agreement (MCSLA) shall be defined.
- *RQ2- How can be discovered, benchmarked and selected the best combination of cloud services based a set of specific non-functional requirements elicited by the end-user?*
  H2- It is possible to discover, benchmark and select the best combination of Cloud Services based a set of specific non-functional requirements elicited by the end-user.

  This hypothesis tries to prove the possibility of intelligent discovery of Cloud Services following a resource-centric approach, searching always for the best opportunistic choices while fulfilling the requirements set by the user. These requirements shall include non-functional properties, such as cost or availability and also total or partial compliance with respect to relevant legislation.
- *RQ3- How can be assessed and monitored the combination of the running cloud services against their CSLAs and current legislation to avoid violations of the contracted features?*
  H3- It is possible to assess and monitor the fulfilment of non-functional requirements of contracted cloud services against composed CSLAs and legislation and react to the violation of these requirements.

  In order to remain sustainable, a cloud-based application cannot stop its operation and it is expected that it is self-adaptive with respect to the new topology needed to fulfil the users' requirements at all times. That is why the dynamic monitoring of NFRs as set by the user or potential SLA violations must be assessed and monitored. These NFRs will at the end be part of the composed MCSLAs, the service level agreement that the multi cloud application will offer to end-users.

6. Analysis of current practice and technology

This section analyses the state of the art and the state of practice which has set up the basis for the definition of the research questions and hypothesis that the proposed research work aims to solve. As depicted in Figure 10, these are used in the later chapters for identifying existing challenges and deriving requirements to architect the envisaged system and building its prototype. It also helps with gathering the technical requirements for building the prototype of the envisaged system.

The section is sub divided into five sections where the first section gives an overview of the research problem. The next four sections describe the current state of practice and research for four relevant topics to this research, namely: 1) cloud services brokers, 2) multi-cloud applications design and operation, 3) runtime monitoring of cloud services and 4) regulatory framework.

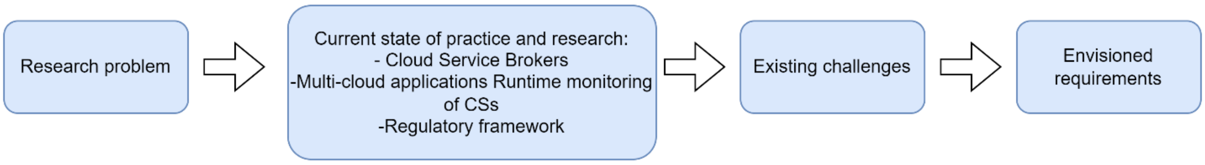


Figure 10. *The requirements analysis process. Source: Author's own contribution.*

6.1 Research problem overview and context

Enabling the complex ecosystem of distributed Cloud Services to support the Cloud Continuum still lays down into several challenges for both users and providers [8].

Authors in [41] and [42] discussed key challenges faced by the users in moving their data/services to Cloud platforms including:

- Choosing the right provider specially for concrete needs such as regulation compliance.
- Service management, including the ability to discover, contract and operate them.
- Security and Privacy issues.
- Trustworthiness of CSPs.
- Dealing with vendor lock-in.
- Support and reliability related to liability of the cloud provider in case of SLA or QoS breaches.

From the provider's perspective, there are many challenges to be addressed being the most relevant ones [43] :

- Understanding the market, the competitors in the domain, the user preferences for various features such as security and trust requirements.
- Adapting to the market: Current Cloud platforms follow a fixed price per resource for their products and services with some small exceptions like Amazon spot pricing [44] therefore more dynamic pricing strategies are required to attract more customers.

Considering the scientific community, researchers [45] [46] [47] [48] extracted similar conclusions from the analysis of existing Cloud Services intermediation approaches: customer assistance is not addressed as the user perspective of the Cloud Brokerage is not tackled; Complex services bidding is still a challenge limited to very few restricted characteristics of the services mainly related with the low level interoperability of the resources and not expanding to other high level layers such as the common monitoring ability or multi-contracting capability; Standards or common languages for services descriptions are missing, and the majority of works rely on simulations for verifying their solutions.

Another relevant stakeholder in the area is the European Future Cloud research cluster [49]. It provides a forum for discussion and collaboration for research and innovation initiatives that address next generation Cloud Computing challenges and issues, including diverse forms of distributed computing (Cloud, Multi-Cloud, Edge, Fog, Ad-hoc and Mobile computing. They produced a revision of research areas and challenges in the scope of Cloud computing, where they identified 36 research challenges organized into 13 research areas (see Figure 11):

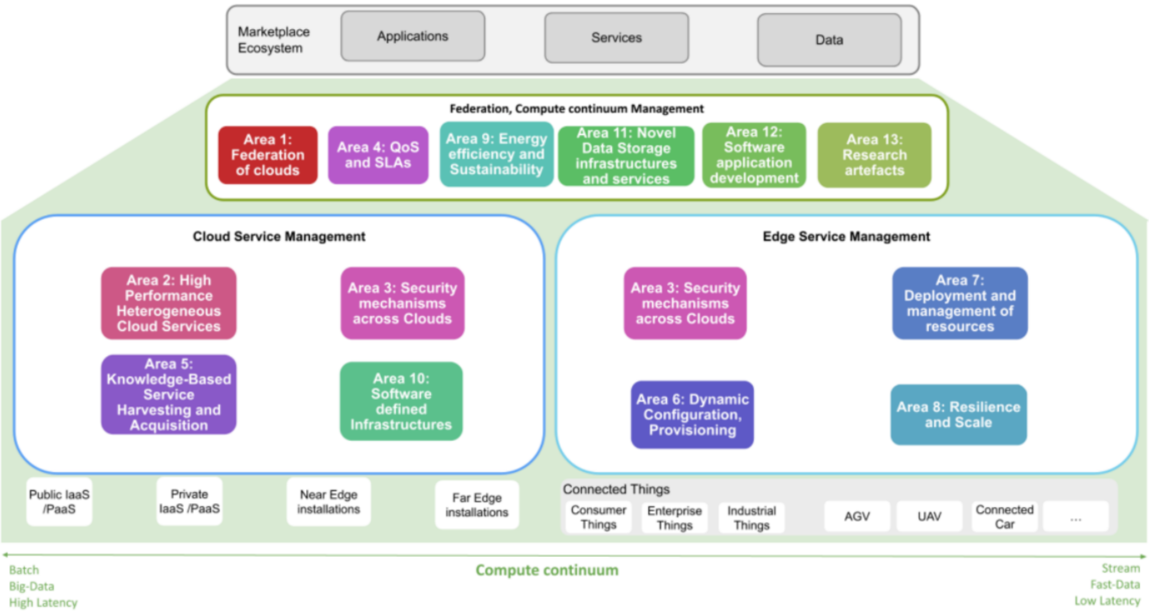

Marketplace Ecosystem
Applications
Services
Data
Federation, Compute continuum Management
Area 1: Federation of clouds
Area 4: QoS and SLAs
Area 9: Energy efficiency and Sustainability
Area 11: Novel Data Storage infrastructures and services
Area 12: Software application development
Area 13: Research artefacts
Cloud Service Management
Area 2: High Performance Heterogeneous Cloud Services
Area 3: Security mechanisms across Clouds
Area 5: Knowledge-Based Service Harvesting and Acquisition
Area 10: Software defined Infrastructures
Edge Service Management
Area 3: Security mechanisms across Clouds
Area 7: Deployment and management of resources
Area 6: Dynamic Configuration, Provisioning
Area 8: Resilience and Scale
Public IaaS /PaaS
Private IaaS /PaaS
Near Edge installations
Far Edge installations
Connected Things
Consumer Things
Enterprise Things
Industrial Things
AGV
UAV
Connected Car
...
Batch
Big-Data
High Latency
Compute continuum
Stream
Fast-Data
Low Latency

Figure 11. *Mapping of Future Cluster Research Areas to Architecture levels. Source: Future Cloud Cluster* [49].

From these, we identified several research challenges that will be addressed during the proposed research work:

- Research Area 1: Federation of clouds, Facilitate Cloud and Edge interoperability and portability, taking into account data privacy, security and applicable legislation.
- Research Area 4: QoS and SLAs, Enforcement of quality of service across cloud models
- Research Area 7: Deployment and management of resources: in a decentralised, autonomous way Autonomic resource management
- Research Area 12: Software application development for the computing continuum, Software engineering, design and programmability of applications in Cloud continuum
- Research Area 13: Research artefacts for the computing continuum, Researchers access to tools, data and infrastructures

The analysis of the presented situation enabled us [8] to group the needs and propose the challenges (CH) that organizations using multi-Cloud Computing will address in the next years:

1. **Governance (CH1)**: Ensuring that services deployed in the cloud are protected is critical. Sharing can create leaks that cannot be tolerated. Promoting solid governance programs in place will protect enterprises and their data.
2. **Risk tolerance (CH2)**: Every enterprise should assess their tolerance for pitfalls such as lost data and application outages. As Information as a Service and Integration as a Service evolve, organizations will reduce the risks.
3. **Regulations (CH3)**: Lobbying for regulations and standards are predicted to be a key step to ensure cloud integration.

4. **Cross border interoperability (CH4)**: The resulting service intermediator shall support intelligent discovery, context-aware service management and fluid service integration, assuring data portability in such a federated ecosystem, while guaranteeing proper identity propagation with service-specific granularity level of information.
5. **Matching customer requirements** with cloud service specifications **(CH5)**: security, legislation awareness and other non-functional requirements need to be fulfilled when using any cross-border service deployed in an heterogenous cloud environment. This implies that the selected service offerings must match with every functional and non-functional requirement coming from the customers.
6. **Legislation compliance (CH6)**, defining means of assuring service compliance with legislation of EU countries: a service is legislation aware when the services are constrained by legal requirements, such as data privacy, data protection, data security and data location. Moreover, a big challenge in this concern is to provide the required means for assuring legislation compliance and update in a heterogeneous environment of composite services form different countries.
7. **Cloud service SLA assessment and monitoring (CH7)**: monitor and control the diverse properties of utilized services, composite or stand-alone, at real-time, while also being able to provide all the critical information for the appropriate reactions, when necessary, especially when SLA conditions are not fulfilled (e.g., elasticity, data localisation).
8. **Seamless change of provider (CH8)**: enable to seamlessly change the service provider with automatic management of all the dependencies to avoid vendor lock-in and decrease the time needed to situations causing outage of the service.

The identification of such challenges has aided on the characterization of the state-of-the-art analysis. In this sense, section 6.2 provides the results of the analysis of the support of Cloud Services Brokers solutions to the identified challenges. These challenges have also guided us on the selection of the main research themes to be studied, giving as a result the analysis presented in sections 6.2, 6.3, 6.4 and 6.5.

6.2 Current practice in Cloud Service Brokers

In this section an analysis of the presented Cloud Broker Solutions will be shown, including their description with their functional scope, type (Commercial CB solutions/Open-Source CB solutions/ EU funded projects results-based solutions, Government Cloud Marketplaces) and the current functionalities covered. This section provides a summary of the main finding in the analysis of the Cloud Services Brokers. These findings were published in [8], in [47] and in [50].

Trying to cover existing solutions with different maturity in terms of technology readiness the analysis has included Cloud providers such as Amazon WS [51], HP [52] and IBM [53], as well as other big players such as Cisco or Oracle. At the same time, both commercial solution providers (such as Appcara AppStack [54] and Jamcracker Service Delivery Network [55]) and Open Source initiatives (Ubuntu Juju [56]) which are developing solutions that enable the creation of customized cloud marketplaces have been studied. In the European research community and relevant for this work we have found Helix Nebula Marketplace [57], composed by a combination of public and private organizations addressing European legal and regulatory requirements.

In another segment, Governmental Cloud marketplaces continue to grow in number and influence. Gov.apps [58] in the US was the first one to appear, but soon others summed up to this trend: UK with Digital Marketplace [59] (previously CloudStore offered under G-Cloud) and other on-going initiatives in Australia and New Zealand [60]. Both US (Gov.apps) and UK (Digital Marketplace) Cloud marketplaces are operated from Government institutions: GSA (US General Services Administration) and UK

Government Procurement Service as part of the G-Cloud Programme. For instance, US GSA supports the contracting of negotiated prices reducing the operational costs of US Government agencies using it. At European level, efforts are finally being invested in a European Federated Cloud through the GAIA-X project [18], in collaboration with German and French governments, which first proof of concept for the design of the European cloud are set to be ready towards the end of 2021[2].

Different solutions have been evaluated, from the more stable ones (already commercialized in the market) to the most innovative ones (provided as research project results). More concrete, 19 commercial solutions, 2 open-source products, 2 public administration solutions and 9 solutions coming from research projects have been analysed. This analysis was published in [50] by the author of this Thesis document.

Table 1. *Analysis of current solutions included in the document* [50].

Number	Solution name	Solution type	Solution's functional scope
#1	AWS Marketplace	Commercial Cloud Services Market place	It is one of the most used Cloud services Marketplace, including developer tools, software infrastructure, business software, and Desktop applications. It supports different technologies, such as Ubuntu, Microsoft, or RedHat among others.
#2	HPE Helion	Commercial Cloud Services Market place	HPE Helion manages cloud and IT services from in-house and external providers. This platform enables service governance hybrid complex clouds and provides functionalities for IT spending related to the contracted Cloud services, service performance, operational configurations and security features.
#3	IBM	Commercial Cloud Services Market place	IBM offers a marketplace with several cloud services. These cloud services are classified in four different types: Application, Business Process, Infrastructure and Platform.
#4	Appcara AppStack	Commercial Cloud Services Market place	AppStack allows users to deploy and manage multi-tiered applications and also to migrate existing applications from virtual or physical servers, into a selected set of cloud resources. It provides an "App Marketplace" populated with more than 60 applications of different categories, i.e., CMS, Business Intelligence, Database, and others.
#5	Jamcracker Service Delivery Network	Commercial Cloud Services Market place	The Jamcracker Platform provides brokerage and management of cloud services, including added value services such as risk and policy compliance management. The Platform offers flexibility and scalability, with a multi-tiered, multi-tenant architecture, RESTful APIs and integration frameworks [55].
#6	Cloud Broker	Commercial Cloud Service Broker	CloudBroker GmbH is a Swiss company providing services in the IT domain and the product the CloudBroker Platform. It is a middleware and application store for compute intensive applications in the cloud.
#7	IBM Cloud Brokerage	Commercial Cloud Service Broker	IBM Cloud Brokerage products enable organizations to procure and provision software and computing resources from multiple suppliers and cloud service providers through this intermediate layer. It provides a single UI to access and compare different cloud services.

[2] Gaia-X has not been included in the study as in the moment where this document was written the first prototypes were not implemented.

Number	Solution name	Solution type	Solution's functional scope
#8	Compatible one	Commercial Cloud Service Broker	CompatibleOne is an open-source cloud-aware software platform which enables organizations to operate Cloud Services Brokerage by making fully interoperable any cloud resources such as IaaS including Amazon, Azure, CloudSigma, and others.
#9	Cloud More	Commercial Cloud Service Broker	Cloudmore provides a way to procure, deploy and consume IT services. Cloudmore enables customizable IT automation, distributed control and relevant cost reconciliations that lower operational overhead and increase business agility.
#10	Activeeon	Commercial Cloud Service Broker	Activeeon is an open-source solution. The company focuses on consulting and project business over the technical solution. Activeeon is focused on French companies.
#11	Nimbix	Commercial Cloud Service Broker	The solution offered includes pure high-performance computing (HPC) cloud built for volume, speed and simplicity. The solution allows users to build, compute and visualize processes. However, there is no transparency on supported underlying infrastructure provided.
#12	Compute	Commercial Cloud Service Broker	The solution offers complete stack for cloud management in a modular way. Additional characteristics are listed below: • Remote visualization. • Automated workflows. • HPC data stager. • Set of ready to use applications. • Own supercomputing center. However, there is no transparency on supported infrastructure provided. Also, the pricing is not transparent.
#13	CycleComputing	Commercial Cloud Service Broker	Cycle computing offers full stack for cloud management. However, it is not transparent in terms of architecture, available software and use cases. Additional characteristics are listed below: • CycleServer is a single management and submission interface for all grid activities. • Support for Condor (with GridEngine, Torque and Hadoop coming soon). However, there is no transparency on supported infrastructure provided.
#14	Nice	Commercial Cloud Service Broker	NICE empowers Grid & Cloud infrastructures by increasing usability and user-friendliness, without sacrificing flexibility and control. It provides portal and visualization capabilities for running applications on cloud and grid infrastructures. Provides Universal and flexible access to grid infrastructure.

Number	Solution name	Solution type	Solution's functional scope
#15	UberCloud	Commercial Cloud Service Broker	A marketplace initiative. UberCloud provides users instant access to CAE software deployed onto various clouds. Multiple resource providers such as Amazon are represented on the Marketplace together with the scientific and engineering solutions. Additional characteristics are listed below: • Marketplace. • Proven by 155 conducted experiments. Allows users to access multiple apps deployed onto multiple clouds.
#16	Fortissimo	Commercial Cloud Service Broker	A marketplace initiative. The initiative is providing different applications running on HPC cloud infrastructure. One-stop-shop availability greatly simplifies access to advanced simulation, particularly to SME. Fortissimo includes over 50 experiments. It has 45 core partners including manufacturing companies, application developers, domain experts, IT solution providers and HPC cloud service providers from 14 countries. Hardware, expertise, applications, visualization and tools easily available and affordable on a pay-per-use. The Fortissimo project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 609029. The Fortissimo 2 project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 680481.
#17	Extreme Factory	Commercial Cloud Service Broker	Extreme factory is a Bull and Atos company subsidiary. It has 4 configurations available: shared public, reserved public, private on-site, private hosted. Extreme factory offers computing portal with work environment customized for a user. Additional characteristics are listed below: • Access to supercomputer for any size of business. • Focused on Bullx^{TM} Super Computers. • Large number of European HPC specialists. Remote visualizer (XRV).
#18	Rescale	Commercial Cloud Service Broker	Rescale is an American based company supported by Amazon and series of US investors. It has large number of applications available and large number of reference success stories. Additional characteristics are listed below: • Series of industry partnerships. • Supporting Cloud Engine. • Number of applications. However, Rescale supports only Amazon infrastructure. In addition, it is US-focused.

Number	Solution name	Solution type	Solution's functional scope
#19	Computenext	Commercial Cloud Service Broker	ComputeNext is an award-winning Cloud Marketplace Platform provider based in Redmond, WA. They empower consumers, vendors, and distributors of cloud services to connect and transact in a transparent and near real-time service delivery model. Specialties include Cloud Computing, Cloud Brokerage, Cloud Marketplace Platform, Federation, IaaS SaaS Marketplace, and Cloud Kiosk.
#20	Juju	Open-Source Services Market place	Juju (formerly Ensemble) is an open-source service orchestration management tool developed by Canonical Ltd., the company behind Ubuntu. It provides cloud services and servers where software can be deployed in an easy way through the code-based scripts called *charms* following IaC principles.
#21	Helix Nebula Marketplace Broker	Combination of public and private organizations	HNX broker has been used since 2014 (when the first version was released). It provides a self-register and self-provisioning engine, and intermediates services form different IaaS cloud providers such as ATOS, and Canopy among others.
#22	Gov.apps	Government Cloud Marketplaces	Apps.gov is a marketplace serving 4.2 million federal employees in the US. It offers several types of services. It is a Cloud services catalogue with information about the service and the links for using, contracting, or accessing it.
#23	UK Digital Marketplace	Government Cloud Marketplaces	UK Digital Marketplace acts more as a catalogue of offerings than as an online actual marketplace, given that the process of getting services for a public sector organisation already considers selection of offers after multiple vendor discussions.
#24	FI WARE platform	EU funded projects results-based solution	Open cloud-based infrastructure for cost-effective creation and delivery of Future Internet applications and services developed by FI WARE project.
#25	ARTIST	EU funded projects results-based solutions	ARTIST has developed a benchmarking tool to select the most suitable cloud provider as part of its migration tool-suite. It provides the functionality to compare different cloud services (IaaS) against pre-defined non-functional requirements
#26	CELAR	EU funded projects results-based solutions	CELAR system for elastic provisioning of resources in cloud computing platforms. It provides automatic, multi-grained resource allocation for cloud applications. This enables the commitment of just the right number of resources based on application demand, performance, and requirements.
#27	Mosaic	EU funded projects results-based solutions	Abstraction layer for IaaS. It offers Cloud developers, maintainers, and users to specify the service requirements in terms of a Cloud ontology and communicate them to the platform via the provided API providing the best-fitting Cloud services to their actual needs and outsource computations.

Number	Solution name	Solution type	Solution's functional scope
#28	Strategic	EU funded projects results-based solutions	Service Broker at infrastructure level based on the OPTIMIS Toolkit (https://digital-strategy.ec.europa.eu/en/news/optimis). The STRATEGIC Service Store acts as a marketplace for developers that want to publish their applications for eGovernment. Continuous monitoring and security are inherent features of the platform.
#29	Broker@Cloud	EU funded projects results-based solutions	The objective of Broker@Cloud (https://sites.google.com/site/brokeratcloud/) is to help enterprises deal with the overwhelming complexity of consuming large numbers of cloud services from diverse providers, future enterprise cloud service delivery platforms will need to implement a wide array of sophisticated brokerage-enabling capabilities, which will give rise to services that go far beyond anything currently offered by today's cloud intermediaries.
#30	BEACON	EU funded projects results-based solutions	This project (https://cordis.europa.eu/project/id/644048) defines and implements a federated cloud network framework that enables the provision of federated cloud infrastructures, with special emphasis on intercloud networking and security issues, to support the automated deployment of applications and services across different clouds and data-centers.
#31	Stormclouds	EU funded projects results-based solutions	STORM CLOUDS (http://storm-clouds.eu/) aims at deeply exploring how the needed shift by Public Authorities to a cloud-based paradigm in service provisioning should be addressed, mainly from the point of view of the end-users, and taking full advantage of edge ICT. The purpose of STORM CLOUDS is to define useful guidelines on how to address the process in order to accelerate it, for Public Authorities and policy makers. These guidelines will be prepared based on direct experimentation in at least 4 European cities, creating a set of relevant use cases and best practices.
#32	Cloud28+	EU funded projects results-based solutions	Cloud28+ (https://cloud28plus.com/) is a federation of European Cloud Service Providers, Resellers, Independent Software Vendors (ISVs), and government entities coming together for information exchange, business development and providing an online Service Hub of trusted cloud services and enterprise apps. Cloud28+ allows comparison of available services vs. customer requirements. The Cloud28+ community currently includes more than 400 members, with a catalogue of more than 1400 services.

6.2.1 Overview of the existing solutions

In this section a matrix with the coverage of the different current solutions to the challenges encountered for the brokerage of Cloud Services is presented. From the solutions presented before, in this sub-section we have selected a group of them to perform a deeper analysis. In this case the selection criteria has been based on availability of the solution/information related to the solution and relevance to the research work. To this end, we have finally evaluated 23 cloud brokerage solutions: 10 commercial solutions, 2 open source, 2 government solutions, 9 solutions based on EU funded research projects.

The colours and headers of the table have the following legend:

🟩	Totally Covered
🟥	Not Covered
🟨	Partially Covered

CHA1	Governance
CHA2	Risk tolerance
CHA3	Regulations
CHA4	Cross border interoperability
CHA5	Matching customer requirements with cloud service specifications
CHA6	Legislation compliance
CHA7	Cloud Service SLA assessment and monitoring
CHA8	Seamless change of provider

Table 2. *Matrix of the support to the challenges by existing brokerage solutions.*

Solution name	CHA1	CHA2	CHA3	CHA3	CHA4	CHA5	CHA6	CHA7	CHA8
Amazon WS	🟨	🟨	🟥	🟨	🟥	🟥	🟨	🟨	🟨
HP	🟥	🟩	🟨	🟨	🟥	🟥	🟨	🟨	🟨
IBM	🟨	🟨	🟥	🟩	🟥	🟥	🟨	🟩	🟨
Appcara AppStack	🟨	🟩	🟨	🟨	🟥	🟥	🟩	🟩	🟨
Jamcracker Service Delivery Network	🟨	🟩	🟨	🟩	🟥	🟥	🟨	🟨	🟩
Juju	🟥	🟨	🟨	🟩	🟥	🟥	🟩	🟨	🟨
Helix Nebula Marketplace	🟨	🟨	🟨	🟥	🟥	🟥	🟨	🟨	🟨
Embotic	🟥	🟨	🟨	🟨	🟥	🟥	🟨	🟨	🟩
Gov.apps	🟨	🟨	🟥	🟥	🟥	🟥	🟥	🟨	🟥
UK Digital Marketplace	🟥	🟩	🟨	🟥	🟥	🟥	🟨	🟨	🟩
Cloudmore	🟨	🟨	🟨	🟨	🟥	🟥	🟨	🟨	🟨
ComputeNext	🟥	🟩	🟥	🟩	🟥	🟨	🟥	🟨	🟥
CloudBroker NO	🟥	🟨	🟩	🟥	🟥	🟥	🟩	🟥	🟨
Intercloud	🟨	🟨	🟨	🟩	🟥	🟥	🟨	🟨	🟨
FI WARE platform	🟨	🟨	🟨	🟨	🟥	🟥	🟨	🟨	🟨
ARTIST	🟥	🟩	🟥	🟩	🟥	🟨	🟥	🟨	🟥
CELAR	🟨	🟨	🟨	🟨	🟥	🟥	🟨	🟨	🟨
mOSAIC	🟥	🟨	🟩	🟥	🟥	🟥	🟩	🟥	🟨
Strategic	🟨	🟨	🟨	🟩	🟥	🟥	🟨	🟨	🟨
Broker@Cloud	🟨	🟨	🟨	🟨	🟥	🟥	🟨	🟨	🟨
BEACON	🟨	🟨	🟨	🟨	🟥	🟥	🟩	🟩	🟩
Stormclouds	🟨	🟨	🟨	🟥	🟥	🟥	🟥	🟥	🟥
Cloud28+	🟩	🟨	🟨	🟩	🟨	🟥	🟨	🟨	🟥

6.2.2 Analysis of key functionalities in Cloud Service Brokers

In this section we present the analysis of the Cloud Broker Solutions with the objective of identifying the key functionalities that the proposed solution ACSmI should cover in order to fulfil the gaps that current solutions do not cover.

To carry out this study the challenges have been translated into Key Functionalities (KF) for the analysed existing solutions to fulfil (Table 3):

Table 3. *Relationship between ACSmI key functionalities and detected challenges for Cloud Services Brokerage.*

Key feature	Related challenge
KF1-Mechanisms to authorize and manage different roles and profiles	CH1, CH6
KF2-Services endorsement with complete information	CH5, CH6
KF3-Information about the status services for the CSPs shall be available	CH7
KF4-Intelligent discovery (including ranking) of services based on NFRs selected	CH4, CH5
KF5-Contracting and billing functionalities for different providers	CH1
KF6-Deployment mechanisms	CH4, CH8
KF7-NFRs monitoring	CH2, CH3, CH4, CH6

During the state-of-the-art phase of the current research an analysis of the most relevant existing Cloud Broker Solutions has been performed. Different nature solutions have been evaluated: Commercial CB solutions/Open-Source CB solutions/ EU funded projects results-based solutions, Government Cloud Marketplaces. In the following pictures the coverage of the different solutions with respect to the key features described in table 3 is presented. (0- Not Known, 1-Not covered, 2-Partially covered, 3-Fully covered):

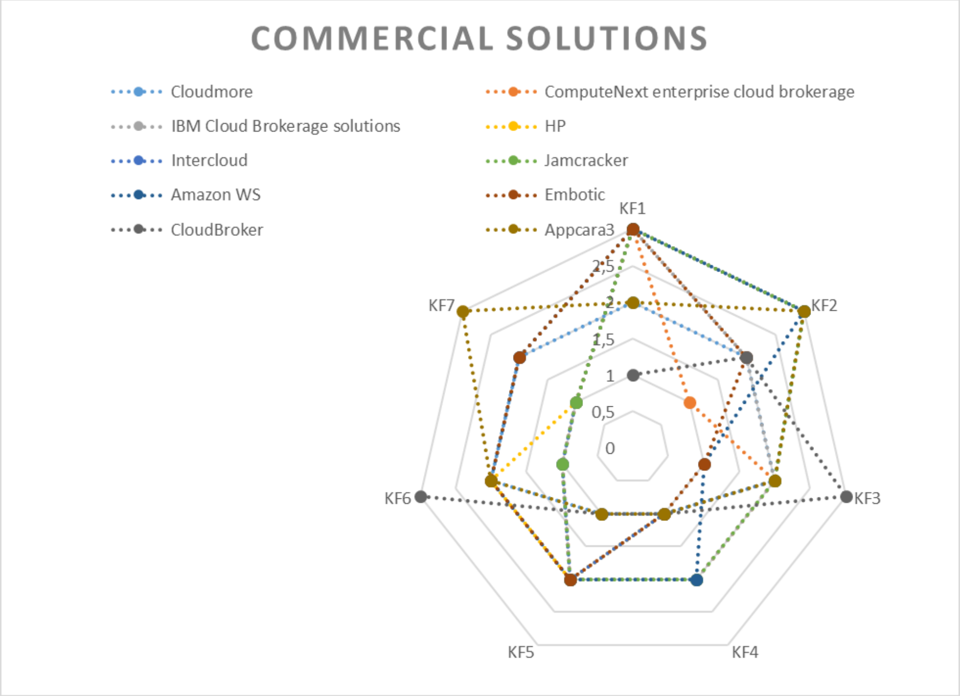


Figure 12. *Analysed commercial solutions coverage of the key features. Source: Author's own contribution.*

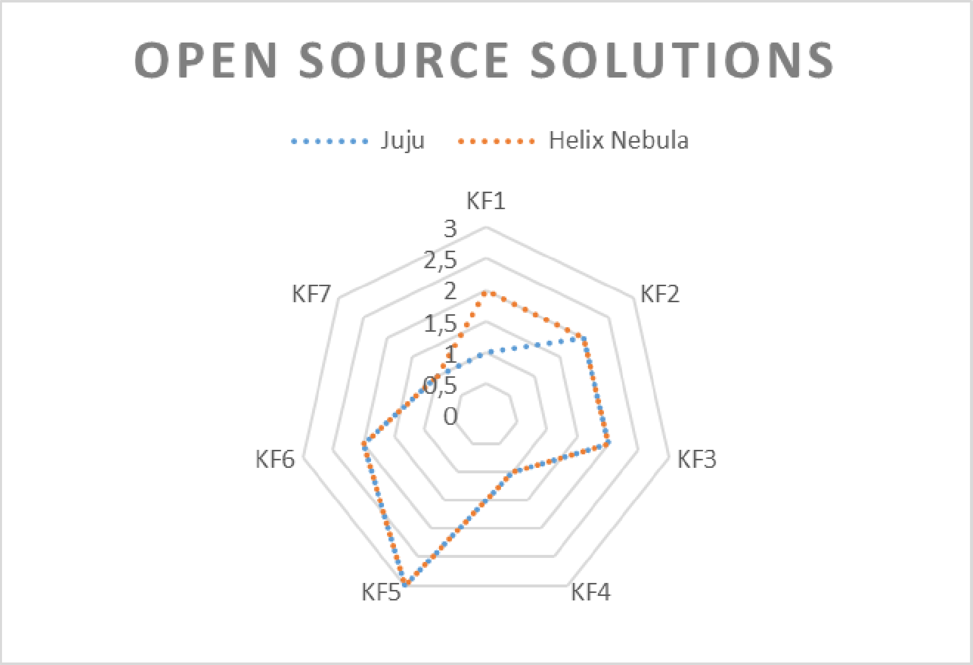


Figure 13. *Analysed open-source solutions coverage of the key features. Source: Author's own contribution.*

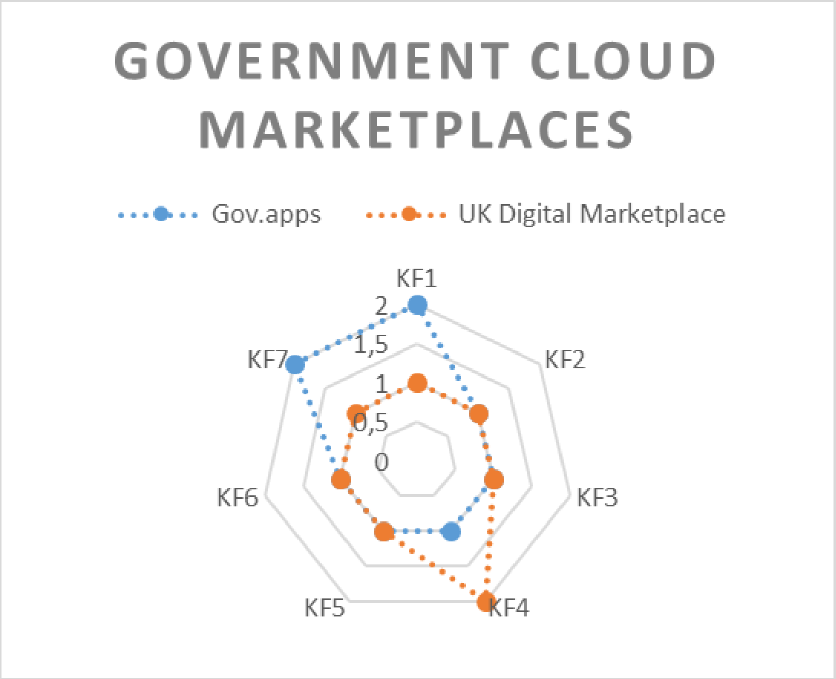


Figure 14. *Government Cloud Marketplaces coverage of the key features. Source: Author's own contribution.*

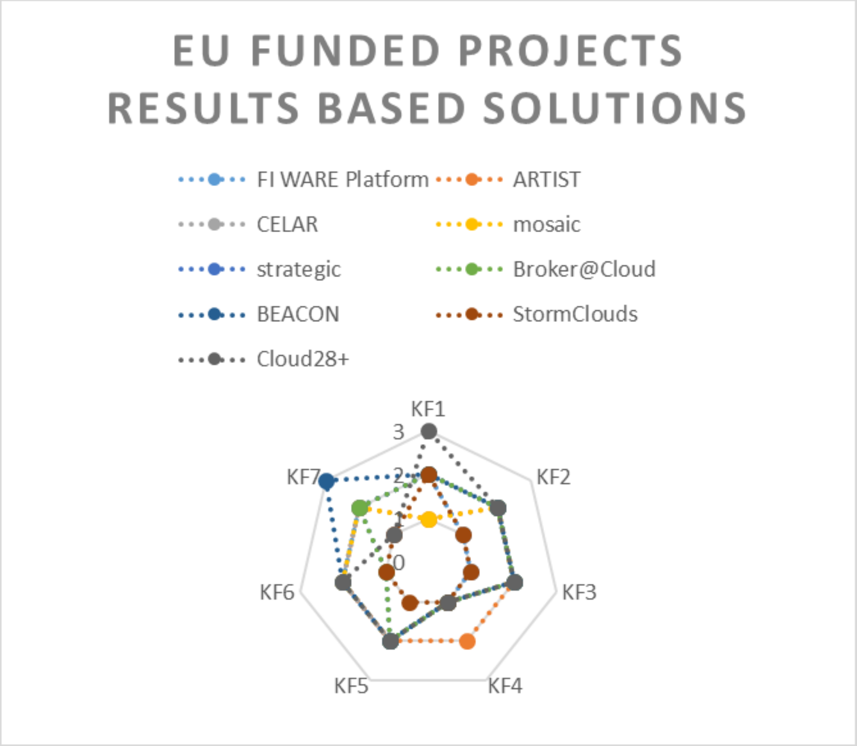
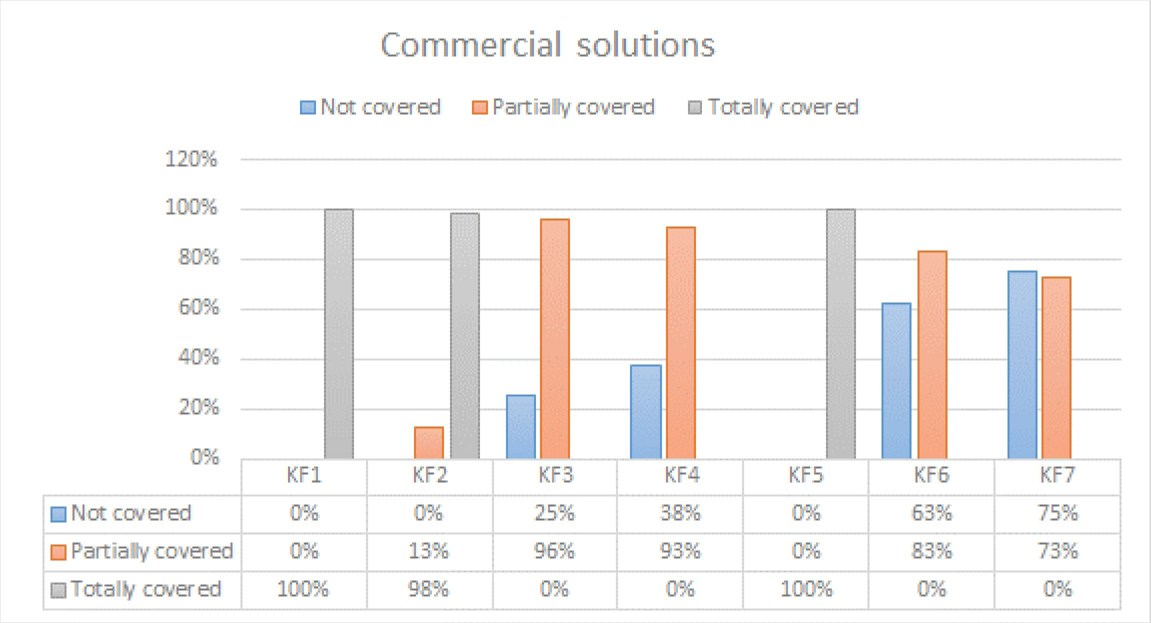


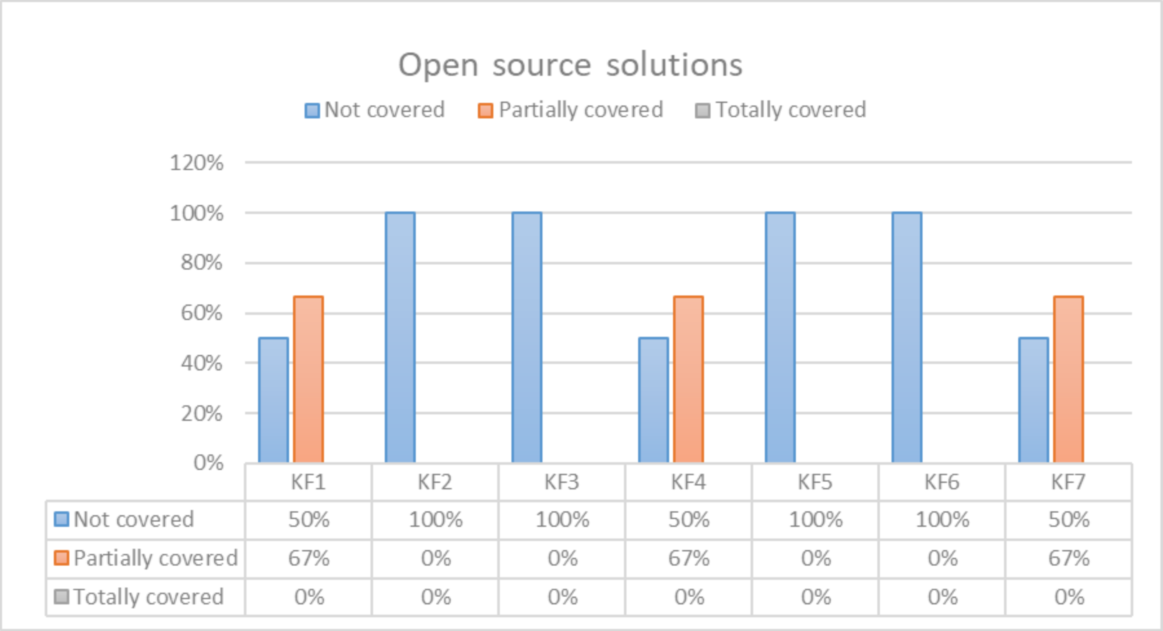
Figure 15. *EU funded projects results-based solutions coverage of the key features. Source: Author's own contribution.*

As shown, different solutions have been evaluated, from the more stable ones (already commercialized in the market) to the most innovative ones (provided as research project results).



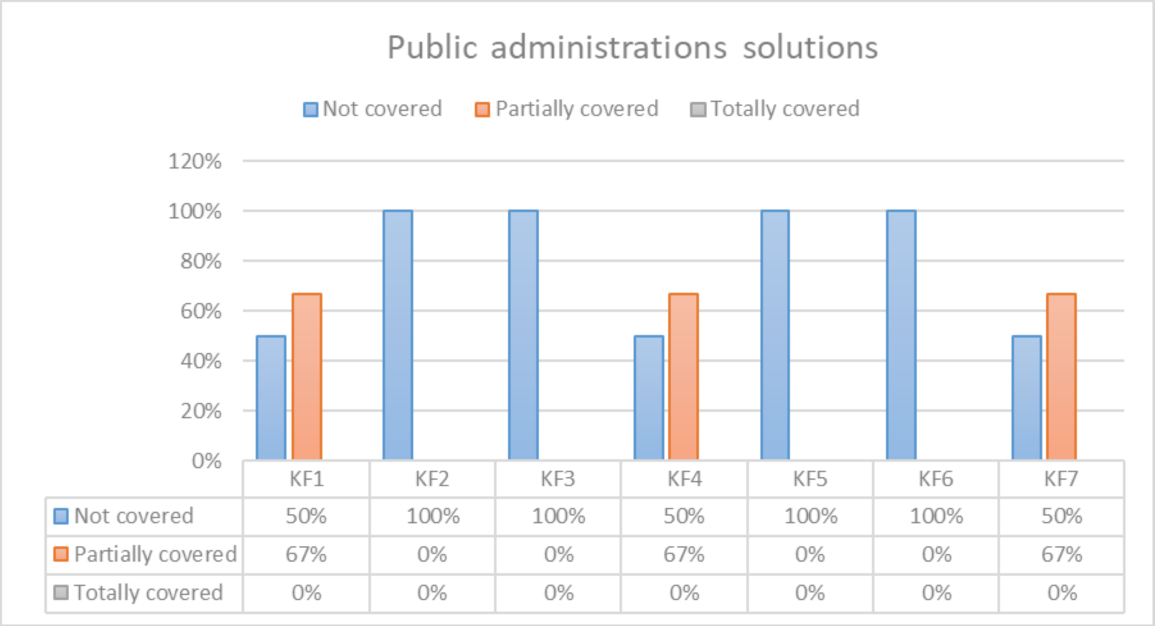
	KF1	KF2	KF3	KF4	KF5	KF6	KF7
Not covered	0%	0%	25%	38%	0%	63%	75%
Partially covered	0%	13%	96%	93%	0%	83%	73%
Totally covered	100%	98%	0%	0%	100%	0%	0%

Figure 16. *Coverage percentage with respect to the key functionalities of commercial solutions for Cloud Service Brokering. Source: Author's own contribution.*



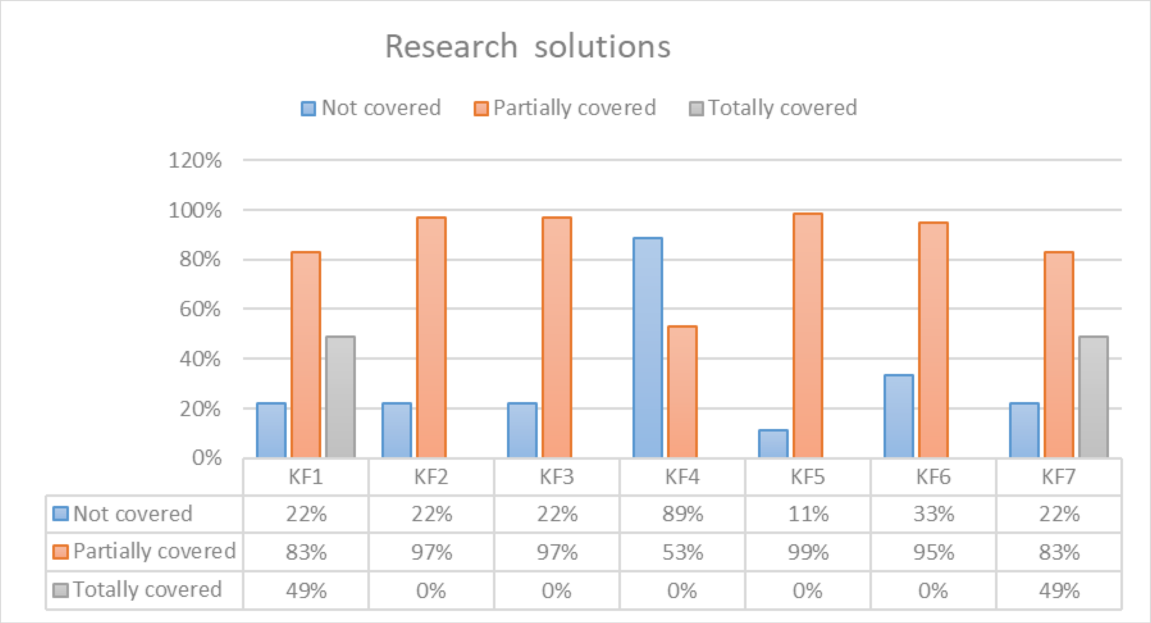
	KF1	KF2	KF3	KF4	KF5	KF6	KF7
Not covered	50%	100%	100%	50%	100%	100%	50%
Partially covered	67%	0%	0%	67%	0%	0%	67%
Totally covered	0%	0%	0%	0%	0%	0%	0%

Figure 17. *Coverage percentage with respect to the key functionalities of open-source frameworks and solutions for Cloud Service Brokering. Source: Author's own contribution.*



	KF1	KF2	KF3	KF4	KF5	KF6	KF7
Not covered	50%	100%	100%	50%	100%	100%	50%
Partially covered	67%	0%	0%	67%	0%	0%	67%
Totally covered	0%	0%	0%	0%	0%	0%	0%

Figure 18. *Coverage percentage with respect to the key functionalities of solutions from the public administration for Cloud Service Brokering. Source: Author's own contribution.*



	KF1	KF2	KF3	KF4	KF5	KF6	KF7
Not covered	22%	22%	22%	89%	11%	33%	22%
Partially covered	83%	97%	97%	53%	99%	95%	83%
Totally covered	49%	0%	0%	0%	0%	0%	49%

Figure 19. *Coverage percentage with respect to the key functionalities of research projects-based outcomes for Cloud Service Brokering. Source: Author's own contribution.*

In Figures 16, 17, 18 and 19 an analysis of their coverage with respect to the key features described in table 3 is presented.

From this analysis the following conclusions can be extracted:

- The mechanisms for the governance of the services are covered by almost all the analyzed solutions.
- The features related with the intelligent discovery and the assessment of the SLA are not covered in the majority of the solutions.
- Most of the commercial solutions do not cover or only cover partially the majority of the key features identified. Indeed a few of them cover the functionalities related to automatic multi-cloud deployment and NFRs monitoring.
- EU funded projects address the majority of the challenges identified (except the intelligent discovery) but they do not offer a complete solution as they are not focused on multi cloud applications and their needs.
- None of the existing solutions cover all the identified challenges/features that are relevant for multi-cloud scenarios.
- The proposed solution ACSmI can contribute to relevant modules of the European GAIA-X initiative for the federated catalogue of services, continuous monitoring, certification and accreditation of CSs.

Besides, an additional analysis can be done to classify the encountered gaps into gaps partially supported or not supported by the tools in the market.

Functionalities not supported by current solutions

These functionalities can be grouped into areas related to security, monitoring and assessment, and legal aspects.

1. Security:

- Data encryption.
- Custom access through API.
- Backup and archiving of cloud consumers information.

2. Monitoring and assessment:
    - CSP SLA monitoring.
    - Information mechanism when an SLA violation or a cloud service outage is detected minimising the use of non-operational clouds.
    - Optimal services recommender.
3. Legal aspects:
    - Transparent management of legal compliance.

Functionalities partially supported by current solutions

In this case, the functionalities can be grouped into the following thematic areas. Billing and metering, cloud service registry and security.

1. Billing:
    - Service usage-based charging.
    - Provision of periodical invoices and billing details.
    - Metering of usage reports per user
2. Cloud Service Registry:
    - Cloud service registry with multiple resources in a common format and with comparable descriptions available to be consumed. The existing solutions of cloud repositories will be extended to contain various parameters available for ACSmI services, e.g., cloud availability, legal aspects and so on
3. Security:
    - Communication layer security
    - Authentication and policy and roles management

6.3 Current practices in modelling and designing dynamic and re-configurable multi-cloud applications

In order to analyse the current state of the art of multi-cloud native applications we performed a Systematic Literature Review study which was sent to Journal of Network and Computer Applications (JNCA) for publication.

The primary motivation of this analysis was to advance on a common definition for multi-cloud native applications setting up the basis for the architectural characterization of these applications and analysing the challenges faced during the lifecycle of these by the DevOps teams and reported by both the industry and the academia. Multi cloud applications can benefit the most of a solution like ACSmI and therefore we include as part of this research work the characterization of these applications.

To have a clear picture of the current situation of the multi-cloud native applications this section provides an overview of the results analysis of the research of the art we performed in the aforementioned SLR. This systematic literature review (SLR) about the origin and characterization of multi cloud native applications had special focus on the application perspective and the challenges that the developers and operators of such applications face during the entire lifecycle of the application, from its design and conceptualization to the runtime and operation phases, going through the implementation and the deployment phases and proposed the following research objectives:

- Objective 1: Meaning and characterization of the multi-cloud concept from the application perspective.
- Objective 2: Definition of the characteristics of a "multi-cloud native" application.
- Objective 3: Analysis of research trends and existing challenges in multi-cloud by design, development and operation

We retrieved more than 900 peer-reviewed papers from journals and the top software engineering conferences in which authors published research related to the multi-cloud applications concept. We filtered these papers, following the specific inclusion and exclusion criteria, to finally obtain 82 primary studies on which we have focused our analysis.

In this section we summarize the results on the analysis of these 82 primary studies and the challenges as unsolved issues identified derived from this work with respect to the characterization of the term multi-cloud and the challenges in the development and operation of multi-cloud applications. Other part of the work done in paper submitted to Journal of Network and Computer Applications (JNCA) has not been included in the current Thesis document.

6.3.1 Discovered challenges and unsolved issues

Meaning of the term "multi-cloud"

When analysing the terms used for defining the term multi-cloud several synonyms or related words are used by the authors: multi-cloud, hybrid cloud, cloud federation, etc. such as those represented in the word map in Figure 20. This compilation of terms is usually used with similar meanings but some slightly different but relevant aspects. Depending on the terms used the way the cloud services are used and operated changes.

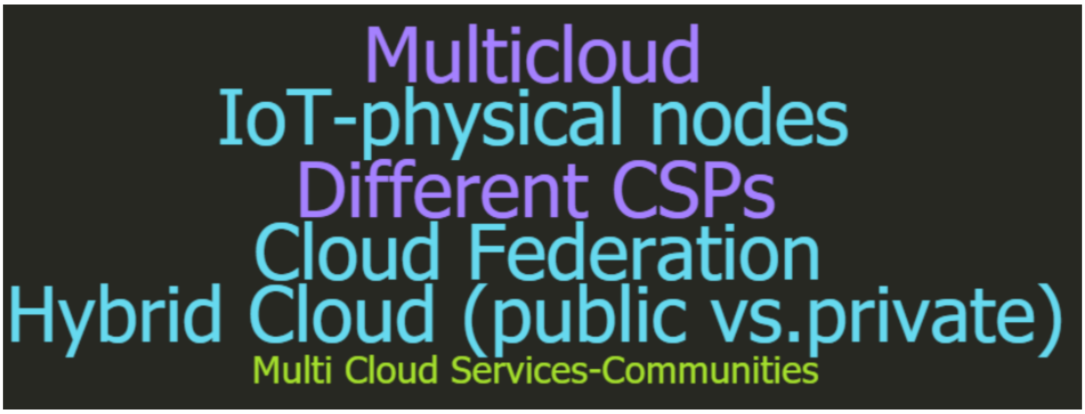

Multicloud
IoT-physical nodes
Different CSPs
Cloud Federation
Hybrid Cloud (public vs.private)
Multi Cloud Services-Communities

Figure 20. *Word map created from the terms collected from the definition of multi-cloud in the primary studies.*

The term "multi-cloud" from the infrastructure layer perspective is used to specify different aspects of the cloud service, from the ownership of the Cloud Service model (i.e., public vs. private) to the specification of the nature of the infrastructural element (i.e., IoT devices).

These are the main conclusions we obtained from the analysis:

- Most of the authors (more than the 40% of the publications analysed) did not specify the actual meaning of the term in the context of the work presented. Most of the ones which provided such definition defined multi cloud as cloud services that were provided by different cloud services providers (28%) with or without of third-party services or any other intermediate layer providing federation mechanisms between these services [61, 62].
- The second most relevant categorization for multi-cloud infrastructure was the focus on the ownership of the cloud infrastructural elements/services. In this sense, several studies used the term multi-cloud to refer to the combination of in house services, owned by a concrete company (the same as the one deploying the application there) and public services with more than one user (Public Cloud services).
- A number of the studies, referred to Federated Cloud Services, that is, different Cloud services coming from different or same providers but that were already interoperable in the sense that an intermediate layer already provided the capabilities for the usage of these services despite their peculiarities considering that they are provided by different Cloud Providers.
- Also, IoT is considered by several authors as a multi cloud implementation
- Recent publications used broad meaning definition for multi-cloud, such as resources from different cloud providers; aggregation of resources by a third-party broker; hybrid cloud architectures and added other novel concepts such as Cloud Continuum or Osmotic computing [63].
- There was a concept of service communities where the cloud services are categorized based on their functionality and the concept of multi-cloud where the cloud services are grouped based on who owns such services. The combination of both approaches results in the multi-cloud services communities' concept [64].

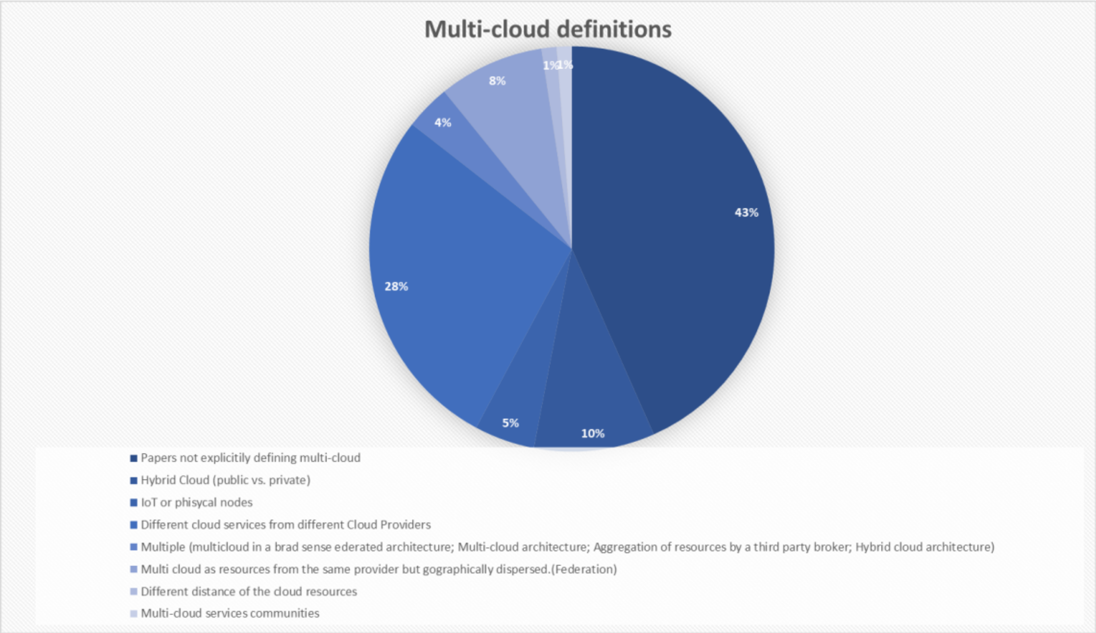


Figure 21. *Classification of research studies based on their multi-cloud definition.*

As a result, we can confirm there is a common but unconscious understanding across the analysed studies of the multi-cloud term.

Existing challenges in the development and operation of "multi-cloud" native applications

Following, we have identified the challenges faced by the DevOps teams during the lifecycle of a multi cloud native application. To do this, we extracted the challenges identified in the design and development phase of the application, in the deployment and execution phase and in the operation phase of multi-cloud applications. For each of these three main phases, categories of the most mentioned challenges were identified (Table 4) and are discussed next.

Table 4. *Identified challenges in the SDLC/SOLC of multi-cloud applications.*

Multi-cloud application SDLC/SOLC phase	Main challenges categories
Design and development	Application components NFRs compliance and resource level matching.
	Software components partitioning
	Cloud agnostic application architectural models
	Lack of Cloud security standards
Deployment	Cloud services heterogeneity
	DevOps practices specific to multi-cloud
Operation	Dynamic re-adaptation
	Communication layer
	Lack of cloud standards
	Maintenance & evolution
	Cloud federation
	Risk management and security

These are the main conclusions we achieved through the analysis of the identified research studies:

- Design and development**:**
  - Difficulties in the specification of the cloud resources characteristics to meet the QoS requirements of each component from the application (from location or user preferences to resources needs) [65], [66].
  - Need of characterization of new functional requirements such as bandwidth on demand or application-specific routing needs to ne now considered [67] or well-known non-functional requirements that need to be addressed from the multi cloud point of view (i.e. security) [68].
  - Software components need to be context aware partitioned, so that they are aware of the characteristics of cloud resources where they can be deployed. NFRs optimization (cost, latency, performance), components synchronization, and data consistency are the challenges identified here [69–71]. Furthermore, the utilization of more than one target deployment resource requires cloud agnostic design of software systems [72] and specific programming models and application architectures so that essential characteristics like scalability can be acquired at application level [73].

- Deployment
  - Traditional DevOps principles are not directly applicable to multi-cloud applications and therefore need to be reconsidered taking into account the specific needs of these kind of applications [72, 74–76] . New activities need to be considered in the DevOps loop, such as deployment configuration optimization, multi-cloud deployment, and creation and monitoring of multi-cloud SLAs.

    - Interoperability through all layers of the federation model, including SLA management and accounting [70], and automatically contract negotiation [77].
- Runtime
    - New solution proposals are needed for automatic re-adaptation and self-healing mechanisms [62, 72, 78, 79] in case of quality degradation, high cost or service unavailability [78].
    - New methodologies are needed to efficaciously and continuously audit cloud services, allowing enhanced monitoring of heterogeneous resources at different levels, from the low level monitoring where different APIs need to be used based on the CSP to the high level multi-cloud application SLA where the different metrics need to be combined to get the composed SLA [80, 81].
    - Migration between cloud providers is still a challenge due to the proprietary nature of the technologies used and the need of seamless portability of both stateless and stateful components [82, 83].
    - Communication layer and the dependency on the network of these distributed multi-cloud applications is also an aspect to be considered [67, 84]. The communication system becomes more complex when switching to a Cloud-to-Edge scenario and balance between performance and security needs to be managed.
    - Security at all levels and during all the application lifecycle needs to be specially considered. The constant flow of data in Public and hybrid Cloud brings uncertainty regarding the various data protection legislations [73] . At the same time, cloud consumers need to address certain outsourcing risks coming along with the adoption of cloud services, such as concerning the risk of shadow-IT [85], loss of control and transparency, security and business continuity [86].

From this state of the art and proactive analysis of the solutions we can derive that most of the challenges identified have their origin in the heterogeneity of the current cloud services. Indeed, this heterogeneity is getting more and more relevant due to the incorporation of new infrastructural elements with the advent of the Cloud Continuum. Many of these aspects, and especially those derived from the heterogeneity of the current cloud services could be improved with the application of recognized of Cloud Standards. But existing efforts for cloud standardization are still in their first stages. Moreover, cloud-vendors themselves are reluctant to unification approaches as they prefer to keep their competitive edge and diversity to attract customers [69, 70].

6.4 Current approaches for the runtime monitoring of Cloud Services

Novel techniques for monitoring of clous resources have been made needed as new decentralized paradigms arise in Cloud Computing, namely: cloud computing, fog computing or more recently the Cloud Continuum.

In the cloud, each microservice (or component of a software application) can be deployed in a different resource, even in a different cloud, attending its specific needs or non-functional requirements (NFR) such as location, cost, performance, etc., making the multi-cloud scenario especially adequate for the deployment of microservices-based applications. This new archetype, where an application is deployed in distributed cloud resources, would not be possible without novel developments in governance, SLA management and monitoring.

In fact, one of the challenges in the area of cloud services federation, among others already discussed like the data portability or the lack of applicability of standards and legislation, is the monitoring and assessment of cloud services SLAs [87].

Specific monitoring of Quality of Service (QoS) and SLA verification of cloud services enables additional functionalities, as optimal service selection [88], real time capacity estimation [89] or future failures prediction [90]. There exist several tools such as Nagios[3], Ganglia[4], Zabbix[5], Checkmk[6], Prometheus[7] Grafana[8], Cacti[9], OpenNMS[10], Icinga[11], NetData[12], among others, that allow monitoring low-level metrics of computing resources in general, but automation on the configuration and calculation of complex metrics to assess CSLAs is still missing, especially when addressing multi-cloud environments [47]. Some attempts [91] have been performed to address the elasticity and scalability inherent to Cloud deployments but with no focus on the monitoring of specific metrics to properly assess actual CSLAs contractually relevant.

Accurate and reliable monitoring of Cloud services is an important source to verify trust and to adjust trust. If the monitoring is conducted by a Cloud Service Broker, then the belief in the results of monitoring is dependent on the trust in that broker with respect to objective and professional monitoring [92]. Thus, the implementation of independent, trustable, and reliable monitoring mechanism is a relevant feature.

In [93], the authors describe different architectures to approach the monitoring of cloud resources depending on the scenario and on the user of the monitoring information , the cloud service consumer (c) or the cloud service provider (p): i) *Extended and Adaptive Internal Monitoring Architecture* (IMA) (c+p), ii) *Concentrated External Monitoring Arquitecture* (EMA) (c+p), iii) *Extended and Adaptative IMA, Traditional IMA* (c) y iv) *Traditional IMA* (p) y *Concentrated IMA* (p+c), v) *Traditional IMA.* They also group these approaches in two main groups: internal monitoring architectures and external monitoring architectures.

Similarly, authors in [87], provide a taxonomy over which they perform a comparative analysis on the monitoring solutions for cloud environments. The following characteristics are compared: monitoring perspectives, monitoring proposal, type of cloud computing model, monitoring architecture, communication model, monitoring platform cost overrun, license and scalability. In the same vein, there are several studies proposing different monitoring architectures in line with the proposed taxonomies [94, 95].

Other solutions propose mechanisms to assess the performance of specific cloud services such as cloud-based storage services [96]. In these cases, the metrics considered are specific to the problem, and difficult to be utilised in other contexts. In a similar way, authors in [97] describe a method for the automatic extraction of monitoring metrics directly from the SLA, in cloud environments.

Thus, it is confirmed that there are still several challenges to be solved at cloud monitoring level [87]. After this overview of the current state of the art the main open issues in the context of runtime monitoring of cloud services are highlighted following:

- More research in research area of cloud-native monitoring is needed with special focus on the dynamic monitoring of cloud resources and their underlying metrics.
- Advanced and intelligent monitoring solutions need to be proposed for complex cloud-based systems. Significant effort needs to be allocated to the investigation of solutions to manage and store the monitoring data.

[3] https://www.nagios.org/

[4] http://ganglia.sourceforge.net/

[5] https://www.zabbix.com/

[6] https://checkmk.com/

[7] https://prometheus.io/

[8] https://grafana.com/

[9] https://www.cacti.net/

[10] https://www.opennms.com/

[11] https://icinga.com/

[12] https://www.netdata.cloud/

- Real time visualization of SLA management for users is another open issue that becomes even more relevant in multi cloud environments when the user consumes several cloud services simultaneously for a single application.

6.5 Regulatory framework, certification, and standards

6.5.1 Relevant standards

The European Telecommunication Standardization Institute (ETSI) has launched the initiative Cloud Standards Coordination (CSC) [98] (on behalf of the European Commission) to coordinate all the standards on Cloud Computing that are being produced in the last years. In this respect there are several sources available analysing and comparing these standards in the Cloud Computing area [50].

- The cloud-standards.org website [99].
- The CloudWatch II project [100].
- StandICT project [101] is Coordination and Support Action funded by the European Commission with the central goal to support European and Associated states presence in the international ICT standardisation scene. It provides the ICT standardization observatory as database for ICT in general and cloud computing in particular relevant standards.

In tables 5, 6, 7 and 8 we provide an overview of standards on Cloud Computing which could be relevant for this research work. This analysis is based on the one published in the DECIDE public deliverable in [102]. Following this approach, we have assessed the relevance of standards according to the following criteria:

- Nature: Only standards approved by a Standard Development Organization (SDO) have been considered.
- Status: Published standards only considered.
- Topics:
    - General purposes standards on Cloud concepts and vocabulary.
    - Service level agreements and service level monitoring.
    - Cloud interoperability and Cloud federation including Cloud service discovery and aggregation.
    - Security related aspects.

The following list aims to provide an overview on the most relevant standards to be considered as a basis for this research. This list was performed under the DECIDE project and included in the D5.1 document [50] and in [102].

Table 5. *Standards related to Cloud concepts and vocabulary. Source:* [50] [102].

Standard	SDO	Summary	Relevance
ISO/IEC 17788:2014 Information technology -- Cloud computing -- Overview and vocabulary [103]	ISO/IEC ITU-T	Provides an overview of cloud computing along with a set of terms and definitions. It is a terminology foundation for cloud computing standards. The standard is applicable to all types of organizations (e.g., commercial enterprises, government agencies, not-for-profit organizations).	These standards provide an internationally agreed terminology for cloud computing. Terms and definition defined in this standard shall be used throughout ACSmI.

ISO/IEC 17789:2014: Information technology -- Cloud computing -- Reference architecture [104]	ISO/IEC ITU-T	The standard specifies the cloud computing reference architecture (CCRA). The reference architecture includes the cloud computing roles, cloud computing activities, and the cloud computing functional components and their relationships.	

Table 6. *Standards related to Service Levels and Service Level Monitoring* [50] [102].

Standard	SDO	Summary	Relevance
ISO/IEC 19086-1:2016 Information technology -- Cloud computing -- Service level agreement (SLA) framework -- Part 1: Overview and concepts [36]	ISO/IEC	ISO/IEC 19086-1:2016 seeks to establish a set of common cloud SLA building blocks (concepts, terms, definitions, contexts) that can be used to create cloud SLAs. It specifies an overview of cloud SLAs, an identification of the relationship between the cloud service agreement and the cloud SLA, concepts that can be used to build cloud SLAs, and terms commonly used in cloud SLAs.	The standard provides a comprehensive overview on Cloud service level and quality objectives and is thus a suitable reference for the metrics to be considered for the development of ACSmI.
ISO/CD 19086-2: Information technology -- Cloud computing -- Service level agreement (SLA) framework -- Part 2: Metric Model [105]	ISO/IEC	The standard provides a formal definition of the term SLA metric.	The standard defines a model for metrics for Cloud service levels that has a machine-readable representation. Such a model is needed to develop a scheme for service discover and composition and to derive associated monitoring objectives.
Web Services Agreement (WS-Agreement) [106]	OGF	Describes Web Services Agreement Specification (WS-Agreement), a Web Services protocol for establishing agreement between two parties, such as between a service provider and consumer, using an extensible XML language for specifying the nature of the agreement, and agreement templates to facilitate discovery of compatible agreement parties. The specification consists of three parts which may be used in a compound manner: a schema for specifying an agreement, a schema for specifying an agreement template, and a set of port types and operations for managing agreement life-cycle, including creation, expiration, and monitoring of agreement states.	The standards describes schema for service level agreements alternative to the one given in [105].

Table 7. *Standards related to Cloud Security* [50] [102].

Standard	SDO	Summary	Relevance
ISO/IEC 27001:2013 Information technology — Security techniques — Information security management systems — Requirements [107]	ISO/IEC	The standard formally specifies an Information Security Management System (ISMS), a suite of activities concerning the management of information risks (called 'information security risks' in the standard). The ISMS ensures that the security arrangements are fine-tuned to keep pace with changes to the security threats, vulnerabilities, and business impacts.	The ISO/IEC 27000-series is the most influential set of standards on security for information and communication systems. Of particular relevance are [108] and [109] addressing cloud computing and the protection of personally identifiable information, respectively.
ISO/IEC 27002:2013 Information technology — Security techniques — Code of practice for information security controls [110]	ISO/IEC	The standard is a popular, internationally recognized standard of good practice for information security.	
ISO/IEC 27017:2015 / ITU-T X.1631 — Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services [109]	ISO/IEC	This standard provides guidance on the information security aspects of cloud computing, recommending and assisting with the implementation of cloud-specific information security controls supplementing the guidance in ISO/IEC 27002.	
ISO/IEC 27018:2014 — Information technology — Security techniques — Code of practice for protection of Personally Identifiable Information (PII) in public clouds acting as PII processors [109]	ISO/IEC	The standard is a reference for selecting PII protection controls within the process of implementing a cloud computing information security management system based on ISO/IEC 27001, or as a guidance document for organizations for implementing commonly accepted PII protection controls.	
Privacy Level Agreement Outline for the Sale of Cloud Services in the European Union [111]	CSA	The Privacy Level Agreements (PLAs) are intended to be used as an appendix to Cloud Services Agreements to describe the level of privacy protection that the cloud service provider will maintain.	The standard provides a complementary view on cloud security that considers the General Data Protection Regulation of the EC.
Cloud Controls Matrix [112]	CSA	The standard is specifically designed to provide fundamental security principles to guide cloud vendors in assessing the overall security risk of a cloud provider. The Cloud Controls Matric (CCM) provides a framework that gives detailed understanding of security concepts.	Comprehensive table of controls to ensure/enhance security of cloud services. Complementary to the ISO/IEC 27000-series.

Table 8. *Standards related to Cloud Interoperability and Federation* [50].

Standard	SDO	Summary	Relevance
Open Cloud Computing Interface (OCCI) [113]	OGF	OCCI is a Protocol and API for all kinds of management tasks with a strong focus on integration, portability, interoperability and innovation while offering a high degree of extensibility.	Most influential standard for the creation and management of interoperable and federated cloud services supported by a large variety of implementations.
Cloud Application Management for Platforms (CAMP) [114]	OASIS	CAMP provides a common basis for developing multi-cloud management tools as well as offering cloud providers and consumers a REST-based approach to application management. CAMP advances an interoperable protocol that cloud implementers can use to package and deploy their applications. It provides a common development vocabulary and API that can work across multiple clouds without excessive adaptation and is compatible with PaaS-aware and PaaS-unaware application development environments, both offline and in the cloud.	These standards provide several approaches to ensure interoperability and portability in distributed or federated cloud systems. Although they are not directly relevant for ACSmI since they are not dealing with SLAs, a general knowledge on how interoperability and portability can be implemented in a cloud system is needed to ensure a consistent approach within the project.
Cloud Infrastructure Management Interface (CIMI) [115]	DMTF	This specification standardizes interactions between cloud environments to achieve interoperable cloud infrastructure management between service providers and their consumers and developers, enabling users to manage their cloud infrastructure use easily and without complexity.	
Topology and Orchestration Specification for Cloud Applications (TOSCA) [116]	OASIS	The standard enhances the portability of cloud applications and services providing a machine-readable language to describe the relationships between components, requirements, and capabilities. TOSCA enables the interoperable description of application and infrastructure cloud services, the relationships between parts of the service, and the operational behaviour of these services.	
Cloud Data Management Interface (CDMI) [117]	SNIA	The standard defines the functional interface that applications will use to create, retrieve, update and delete data elements from the Cloud. As part of this interface the client will be able to discover the	

Standard	SDO	Summary	Relevance
		capabilities of the cloud storage offering and use this interface to manage containers and the data that is placed in them. In addition, metadata can be set on containers and their contained data elements through this interface.	
European Cloud Certification Schema (EUCS) [118]	EUCS	EUCS scheme (European Cybersecurity Certification Scheme for Cloud Services), looks into the certification of the cybersecurity of cloud services. It was created by an Ad Hoc Working Group (AHWG) to work on the preparation of the candidate scheme on cloud services, as part of the European Cybersecurity Certification Framework. The first version was published on December 2020.	The standard focuses on the Cybersecurity certification of the Cloud Services.
IEEE 2302-2021 [119]	IEEE	This new standard was developed by the Intercloud Working Group within the IEEE Computer Society's Cloud Computing Standards Committee, in collaboration with the National Institute of Standards and Technology (NIST). Based on the cloud federation roadmap outlined in NIST Special Publication 500-332: The NIST Cloud Federation Reference Architecture.	Even this standard is of high relevance the first publication was in March 2022 when the research work of this Thesis was completed.

6.5.2 Regulatory framework and legislation compliance

The impact and importance of the legislation in the provision and usage of Cloud Services has become even more relevant with the EU's General Data Protection Regulation (further: GDPR) having gained full application in May 2018 [31]. This aspect acquires even more importance when considering heterogeneous environments involving multiple cloud providers and all the subsequent implications (multiple contracts management, multiple SLAs management, etc.).

The impact of legal aspects of the cloud services underlying a multi-cloud native application has an impact on the Software Development and Operation Lifecycle by necessitating their introduction as non-functional requirements when developing the application and maintaining these requirements throughout the application's lifecycle. This presupposes a legal assessment of the cloud services, in order to be able to include or exclude cloud services based on the legal non-functional requirements set by the application developer.

When talking about cloud service brokers, legal compliance can relate to contractual aspects such as conclusion and termination of contracts, data location, data protection, data portability, interoperability or rules blocking the free flow of non-personal data etc. While many of these aspects might not directly affect the functioning of the service, they may be very relevant for the user of the service. Therefore, they should be considered prior to the selection and launch of the specific set of cloud services used for a particular application as NFRs on the application level and a broker solution should show their fulfilment, playing a facilitating role for the selection by the cloud user of the services that fit both the functional requirements of the envisioned processing or application and the legal requirements/preferences of his or her organization.

Data location is discussed next as an example. While this might be hard to confirm in practice (i.e., where is the data really), it is easy to measure as a legal assessment of a given cloud service. The location of the data is promised by the CSP and mentioned in the CSP's offering for the given service in sufficiently specific terms. As a legally relevant non-functional requirement it is easy to compile from the (contractual) information provided by the CSP or available online.

Other legal aspects, however, are harder to define from the available information i.e., appropriate technical and organizational measures required by the GDPR. This is much harder to define precisely, based on the information a CSP offers. The same goes for security level of an offered service. In general, security aspects are much harder to be determined in terms of what the application developer wants/requires and how this should be compared against what the CSPs offer.

It is here that alternative indications such as adherence to standards and/or codes of conduct, but mostly the instrument of certification, plays a role, as a substitute for a more precise determination of the legal aspect in question. Rather than defining a concrete security level for instance, it could be defined by the certifications a CSP or the offered service has. The application developer can derive from this a certain level of compliance with the legal requirement, e.g., of security. Moreover, certification has the added benefit that an external body has actually checked whether the CSP has the controls declared to be present as part of the certification in place actually.

Therefore, it seems a good idea to leverage the work done in relation to certification and adherence to certain standards or codes, by including this as part of legal non-functional requirement, enabling the application developer to make a better assessment of the extent to which a CSP or its service fulfils a given legal non-functional requirement and whether it is therefore suited to be used as part of the deployment of the application concerned. The challenge however is how to deal with the plethora of existing certifications currently in existence and the lack of any single comparison mechanism.

As explained above, relevant legal aspects may relate to contractual aspects such as conclusion and termination of contracts, data location, data protection, data portability, interoperability or rules blocking the free flow of non-personal data etc. While many of these aspects might not directly affect the functioning of the service, they may be very relevant for the user of the service. Therefore, they should be considered prior to the selection and launch of the specific set of cloud services used for a particular application as NFRs on the application level.

Some of these relevant legislations are: regulations such as GDPR article 28, article 20 (data processing and data portability) [31], or the upcoming Free Flow of Data, but also on existing certification schemes such as the ones introduced in the previous section.

Other aspects, such as the adherence to EU Code of Conduct for cloud providers [120] or the adherence to the self-regulation of portability and interoperability, is also required to be inserted. This last aspect is very important, since it is already observed in Article 6 of the proposal for a regulation of Free Flow of Data [121] which states that "*the Commission shall encourage service providers and professional users to develop and implement codes of conduct detailing the information on data porting conditions (including technical and operational requirements) that providers should make available to their professional users in a sufficiently detailed, clear and transparent manner before a contract is concluded*". This means that CSPs must offer technical solutions for users to be able to port their applications and data in an easier way, using a standard and interoperable format.

Table 9. *Relationship of relevant legislation to be considered.*

Legislation	Brief description
General Data Protection Regulation (GDPR) (Regulation (EU) 2016/679) [31]	This regulation [31] is a regulation by which the European Commission intends to strengthen and unify data protection for individuals within the European Union (EU). The objectives of the GDPR are to give citizens back the control of their personal data and to simplify the regulatory environment for international business by unifying the regulation within the EU. When the GDPR takes effect (May 2018 after a two-year transition period) it will replace the data protection directive (officially Directive 95/46/EC) from 1995.
Code of Conduct for cloud services providers [122]	Code of Conduct (Code)'s objective is to make it simpler for cloud customers to analyse whether cloud services are suitable for the processing of personal data. It will increase trust and will encourage a high default level of data protection in the European cloud computing market and their stakeholders: SMEs, users and providers, and public administrations. The Code is a voluntary instrument, allowing a CSP to evaluate and demonstrate its adherence to the Code's requirements, through self-evaluation and self-declaration of compliance, or by relying on third-party certification. Any CSP may sign up to the Code, irrespective of where personal data is stored and processed.
ISO/IEC 27018 Information technology — Security techniques — Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors	ISO 27018 [109] is part of this risk scenario and introduces a number of measures, procedures and controls through which the cloud service providers guarantee compliance with the European Directive governing the processing of personal data, reassuring potential buyers of the ability to always control the processes personal data undergo within the cloud provider's systems in complete transparency.

6.6 Summary

The research work performed for this thesis arises at a time when cloud federation and brokerage solutions are starting to be defined, designed, and adopted. At policy level, the need of specific solutions for cloud services Federation is also considered as a key research objective. To this respect, the European Data Strategy (European Commission, 2020) outlines a strategy for policy measures and investments to enable the data economy in Europe. Among the different problems identified the adoption of cloud is mentioned, both from the consumer and provider side. These problems include compliance with data protection regulation, multi-cloud interoperability, data portability, and the lack of a European cloud and data infrastructure. One of the four pillars upon which the data strategy is built is the proposal to create a "*a cloud services marketplace for EU users from the private and public sector […] by Q4 2022*" where "*potential users (in particular the public sector and SMEs) [are] in the position to select cloud processing, software and platform service offerings that comply with a number of requirements in areas like data protection, security, data portability, energy efficiency and market practice*".

Unsolved scientific challenges remain in the state-of-the-art solutions for multi cloud systems in different aspects.

First, multi cloud native applications need to be defined and characterized and specific techniques and processes are required for the Development and Operation of multi cloud native applications. These types of applications can benefit the most of cloud federated solutions. Therefore, their characterization specially with respect to the SDLC and SOLC is crucial so that multi cloud paradigm can be fully embraced by the software industry. Existing DevOps solutions are focused on traditional applications and the specificities of multi-cloud native applications from the DevOps perspective have not been thoroughly addressed yet. While multi cloud native applications are designed, developed, deployed, and operated following the DevOps philosophy, the challenges that DevOps teams face are more complex compared to a "normal" application DevOps process. For example: the need for the identification of multi cloud based architectural patterns in the design phase, the incorporation of the functional and non-functional requirements of the application from the design to the deployment phase to choose the most suitable cloud service for each application component, the self-adaptive mechanisms at operation phase, etc.

Second, current cloud brokerage frameworks do not fully support the (automatic) governance of diverse (in nature) and heterogenous cloud services. Even some of them address to some extent heterogeneity of cloud services, the mechanisms incorporated to describe, contract or monitor the services is technology specific or lacks the flexibility to manage cloud services from different "types" of cloud services providers (i.e. big cloud service providers, small cloud services providers). Therefore, as shown in the next chapter the work proposed in this thesis advances in the provision of different and heterogeneous discovery and contracting mechanisms both for the big players and the small European cloud service providers (CSPs), facilitating the governance and co-living of both types of CSPs.

Third, ACSmI monitoring provides an implementation of the ISO/IEC 19086-1 standard to be able to monitor metrics in accordance with the CSLA established for a set of concrete NFR (availability, performance, location and cost). This standard defines a set of common cloud SLA building blocks (concepts, terms, definitions and contexts) that can be used to create Cloud Service Level Agreements (CSLAs). With this proposal, our work contributes to enhance the utilisation and comparison cloud services from different cloud providers. Existing tools such as Nagios or Ganglia provides means to monitor low level metrics of computing resources in general, but still automation on the configuration and calculation of complex metrics to assess CSLAs is still missing, especially when addressing multi-cloud environments. With the objective of being able to compare and combine the SLA from different CSPs, the work presented in this thesis proposes a framework where the metrics are defined and expressed by the cloud services intermediator and are compared to the SLOs provided by the providers.

Forth, legal aspects and regulation compliance is gaining relevance due to the EU's General Data Protection Regulation having gained full application in May 2018. This aspect acquires even more importance when considering heterogeneous environments involving multiple cloud providers and all the subsequent implications (multiple contracts management, multiple SLAs management, etc.). Thus, the legal aspects of the cloud services underlying a multi-cloud native application have an impact on the Software Development and Operation Lifecycle by necessitating their introduction as non-functional requirements when developing the application and maintaining these requirements throughout the application's lifecycle. This presupposes a legal assessment of the cloud services, in order to be able to include or exclude cloud services based on the legal non-functional requirements set by the application developer.

Last but not least, ACSmI proposes a catalogue of cloud resources through an intermediate data abstraction layer where all the elements of the Services can be compared and benchmarked. While some approaches have been proposed using semantics or other modelling languages, ACSmI proposes a pragmatic approach where the data model has been designed to address the needs of the Cloud Service providers and consumers and to provide novel attributes which are not considered in other approaches (i.e., legal aspects, cost). Furthermore, the approach followed allows the easy enlargement of the service classes to new type of services (i.e., edge nodes services in the case of IoT based systems). The rationale behind this design is based on the assumption that ACSmI will integrate both general purpose Cloud vendors such as Amazon as well as small niche oriented ones (i.e. Aimes https://www.aimes.uk/

specialized on the Health sector). This contributes to the creation of a federated catalogue of cloud services which is one of the objectives of the European Commission with the GAIA-X project. The catalogue created through the work presented in this Thesis has been already presented as a candidate to implemented de GAIA-X federated catalogue [17].

7. DevOps concept for multi cloud native applications and ACSmI

7.1 Introduction

This section describes the two major research outputs and related contributions to the uptake of Cloud Computing solutions brought by this Thesis:

1. Extended DevOps concept for multi cloud native applications
2. ACSmI -Advanced Cloud Service meta-Intermediator

Figure 22 represents how each of the outcomes contributes to the different stages of the DevOps workflow.

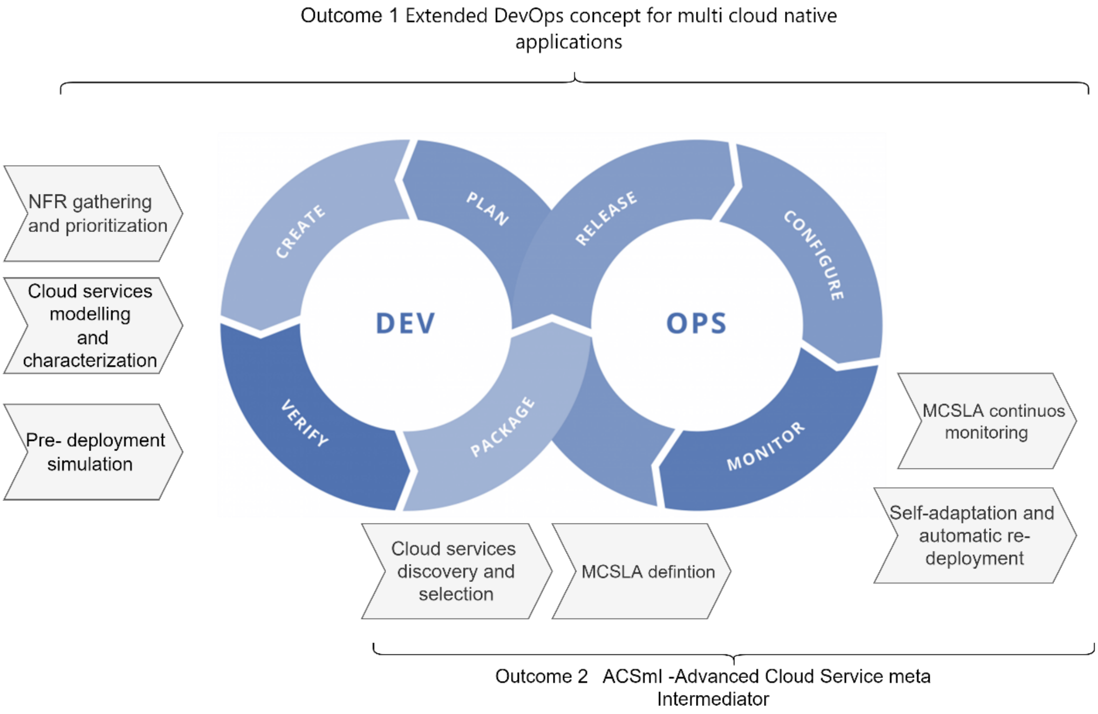

Outcome 1 Extended DevOps concept for multi cloud native applications
NFR gathering and prioritization
Cloud services modelling and characterization
Pre- deployment simulation
CREATE
PLAN
VERIFY
PACKAGE
DEV
RELEASE
CONFIGURE
MONITOR
OPS
MCSLA continuos monitoring
Self-adaptation and automatic re-deployment
Cloud services discovery and selection
MCSLA defintion
Outcome 2 ACSmI -Advanced Cloud Service meta Intermediator

Figure 22. *Thesis outputs in the DevOps workflow of multi cloud applications. Source: Author's own contribution.*

The "Extended DevOps concept" spans over the whole workflow and indeed extends it on both its axes: namely Dev and Ops, to address multi-cloud software needs. This novel "*Extended DevOps*" concept includes new phases both in the Development part and in the Operations part of the "traditional DevOps" which currently do not exist (such as pre-deployment simulation) and proposes ACSmI , the cloud service meta-intermediator as one of the main supporting tools for the operation phase of this methodology.

The next subsections provide a detailed description of each of these outcomes produced in the Thesis work and in Section 7 the validation carried out for each of the parts is fully described.

7.2 Multi Cloud Native applications: architectural characterization for implementation, and deployment optimization

In this section we describe the main contributions achieved during the research work to fulfil one of the objectives proposed: *Definition and characterization of the multi-cloud native applications*. Multi cloud native applications have been analysed form their lifecycle point of view, SDLC and SOLC and in the next sections the two contributions related to this objective and collected during the research work are described in detail:

a) Definition of multi-cloud native application concept and related research challenges
b) Extended DevOps workflow for multi-cloud applications

7.2.1 Definition and investigation on the multi cloud application concept and related future research trends

The term "cloud native application" has not been used as such in the research community. Nevertheless, many of the studies characterizing multi-cloud environments also provided information about the types and the characteristics of the applications deployed in top of such multi-cloud environments (30% of the analysed works).

From the analysis of these studies, we found that multi-cloud native applications were structured into three main categories:

a) *Replicated multi-cloud applications* [70, 72, 123, 124]: Applications which are deployed in multiple clouds not simultaneously. These applications make serial usage of services migrating from one Cloud to another. They are not componentized applications. On the contrary these applications are specially built to run in different clouds and to switch from one cloud to another as a whole. Main reasons to use this model are cost reductions, backups, emergencies, contract ending, etc.
b) *Distributed multi-cloud applications* [65, 67, 74, 75, 83, 125, 126, 126–134]: Multi-cloud applications with subcomponents. Each component of the application is designed to be deployed on a different resource form the same or different provider. The usage of simultaneous multiple providers within a single application is driven by situations when a cloud provider does not provide the whole functionalities required by the application. These applications can exploit those cloud services which better suit their requirements cost, security etc.
c) *Multi-cloud applications covering the serial usage of cloud services (replicated multi-cloud applications) and the simultaneous usage of cloud services (distributed multi-cloud applications)* [135, 136].

While replicated applications do not need to coexist with more than one cloud environment, distributed multi-cloud applications are simultaneously being run into heterogenous and diverse cloud services (in terms of location, management systems, technology, interfaces, etc). This is the kind of applications that we try to characterize, understand and analyse the underlying challenges as the main application target of this Thesis.

As a result of this analysis, we can confirm that the topic of multi-cloud appears to be highly relevant for the researchers and practitioners. Software community has contributed heavily to its body of knowledge from the birth of cloud computing and the results of the study we performed (see section 6.3.1) demonstrate the great significance of multi-cloud to the academic and industry community. However, with regard to the clear understanding and characterization of the term is still unachieved. Multi-cloud is used to describe un-like and heterogeneous concepts especially tackled to the infrastructural layer, ranging from the ownership of the elements to the relationship of this elements (i.e., federation) addressing many other relevant concepts. This lack of a common understanding of the term is even more relevant when the "multi-cloud" is referred to the potential of the application to be deployed on a multi-cloud environment. While "cloud native applications" have been described and characterize in the literature, we haven't found any work trying to characterize and understand the specific case of "multi-cloud native applications" and their particularities with respect to their design, development, and operation. In general, although the topic appears to be promising, research on multi-cloud native applications seems to be still in its infancy, providing a range of new opportunities for researchers.

We have shown clear evidence that scientific contributions in the literature compiles a "cloud" of terms that are usually used with similar meanings but some slightly different but usually relevant aspects that impact the way those cloud services are used and operated.

Related future research trends

As a result of the research work, we have identified a set of research challenges which impact the multi cloud native applications lifecycle. The research to date has focus on factors such as resource allocation, cloud federation, virtualization, and modelling for the cloud. More concrete opportunities for future research include:

- **Characterization of the Cloud Continuum and its relationship with Cloud Osmotic paradigm** [63] is an emerging research topic that has raised from the characterization of complex environment where "every element" at infrastructural layer can be considered as part of such continuum. Several approaches define and address the term multi-cloud from a restricted perspective where the adjective "multi" is in fact defining only two or three different "types" of cloud elements. The rapid development of emerging Cloud, Edge, Fog and Internet of Things (IoT) into the Cloud Continuum has become the management of service heterogeneity even more complex, as well as the service provisioning based on the classification and allocation of suitable computational resources. **Osmotic computing** [137] is a new paradigm that allows the service migrations leveraging Functions-as-a-Service (FaaS) and the concept of hybrid architectural style combining both microservices and serverless architectures. Current approaches like the one described still focus a lot on the "infrastructural" side of the problem (i.e., resource allocation, migration) while leaving aside the implications at architectural level. Special attention to **data portability, and stateful components migration at runtime** are identified as areas for future research.
- Although the usage of loosely coupled architectures based on microservices and implemented through containers is emphasized in many primary studies, we discovered that **current design application patterns for multi-cloud native applications are underdeveloped**. In [72] the focus is on portability between different given CSPs but no generic software design patterns are provided. Besides the need of technology and CSP agnostic patterns a clear gap appears in solutions which focus on the **architectural aspects of a multi-cloud application** rather than on the deployment and portability of cloud applications. Again, the focus needs to be shifted form the infrastructural layer to the application architectural layer. In this sense, further investigation is needed on new approaches for stateful, and stateless application components design and partition, lightweight design profiles of software components to be deployed on the edge, or reference architecture models for multi-cloud native applications.
- The **application of the DevOps principles to the life-cycle of multi-cloud applications** presents a clear opportunity for future research as different activities inside the DevOps cycle need to be adapted to the multi cloud context. The application components NFRs compliance and resource level matching still poses several challenges as claimed by the majority of the studies. The heterogeneity of models for the characterization of the different infrastructural elements is one of the key research areas. While several standards like OASIS CAMP [114], OASIS TOSCA [116] or CIMI [115] try to provide stable and common interfaces to describe the topology

of such cloud services, the multi-cloud notion has not yet been incorporated. In this sense, the description and characterization of the whole cloud continuum including, at the same time, traditional cloud resources, IoT elements and edge components need to be incorporated to the standards. Furthermore, the incorporation of these new infrastructural elements on the board game opens the type of NFR to be considered to a new level. Here the definition of the novel NFR related to the network, communications, security (i.e., especially data sharing) or even legal plays a key role that needs to be incorporated to the requirements definition phase.

- Similarly, the **context aware design of multi-cloud applications architecture** is another open topic that still poses several challenges for the research community. NFRs optimization (cost, latency, performance), components synchronization, data consistency and cloud agnostic design of software systems are the challenges identified here [72, 73, 78, 83, 86, 138].

  Once the NFRs from both sides (multi cloud application and infrastructural elements available) are clearly described and characterized, they need to be matched in the sense that the best combination of infrastructural elements needs to be selected for each application component and for the complete application as a whole. In the analysed studies techniques such as optimization or benchmarking have been commonly used but they are usually restricted to performance and workload. In the new context of Cloud continuum other approaches like lightweight benchmarking and multi-objective optimization need to be addressed.

- To address the challenges of the operation of multi-cloud native applications, the **federated model** of such heterogenous services is to be proposed to leverage its benefits [70, 74, 77, 125, 136, 139]. Nevertheless, even if federation of cloud services is getting mature, it needs to be expanded to address the whole Cloud Continuum. Communication layer for instance, and the dependency on the network of these distributed multi-cloud applications is also an aspect that needs to be further considered and investigated in the federated model [67, 84]. New, lightweight elements such as sensors, edge nodes or IoT gateways included in the Cloud Continuum need to be considered in such federated model. Thus, new models for the characterization of such elements need to be researched. These models include the description of such elements at different levels of abstractions but also the incorporation of such elements into the management of business workflows, such as SLA management and accounting[70], and automatically contract negotiation [77]. Specially SLA management and accounting derives the necessity of the incorporation of new methods and techniques for the monitoring of new elements, lightweight ones, and networking elements.
- **Re-configuration and self-healing of the multi-cloud native applications** at runtime is also attracting more and more interest in the research community. In this context, current solutions are domain or technology specific or only focused on single steps of the re-configuration process. Other aspects, such as networking issues are still not covered in a sufficient way. Furthermore, self-healing mechanism for the applications and executions environments covering the whole self-healing process from the discovery, and configuration of the resources, for the network preparation and deployment of all software layers are also missing. This process becomes even more complex when addressing not only the portability of the computational components (stateless components) but also the portability of data, or stateful components. One of the enablers for such portability can be adopted from the application point of view, through the adoption of containers-based technologies, such as Docker [140]. But containerisation per se does not solve the portability problem. As we stated in [19] "*when porting components between two cloud providers, data need to be moved and kept synchronized (at three different levels—blocks, files, or transactions), and most container-based platforms do not handle this. Manual configuration is still needed, and stateless components are decoupled from the data that have to be stored in a database or other type of storage. Consequently, data portability introduces new requirements added to the already time consuming, error prone and mainly manual activity of porting application components over different infrastructural elements: 1) establishing the right networking conditions, so that data can be accessed from the required*

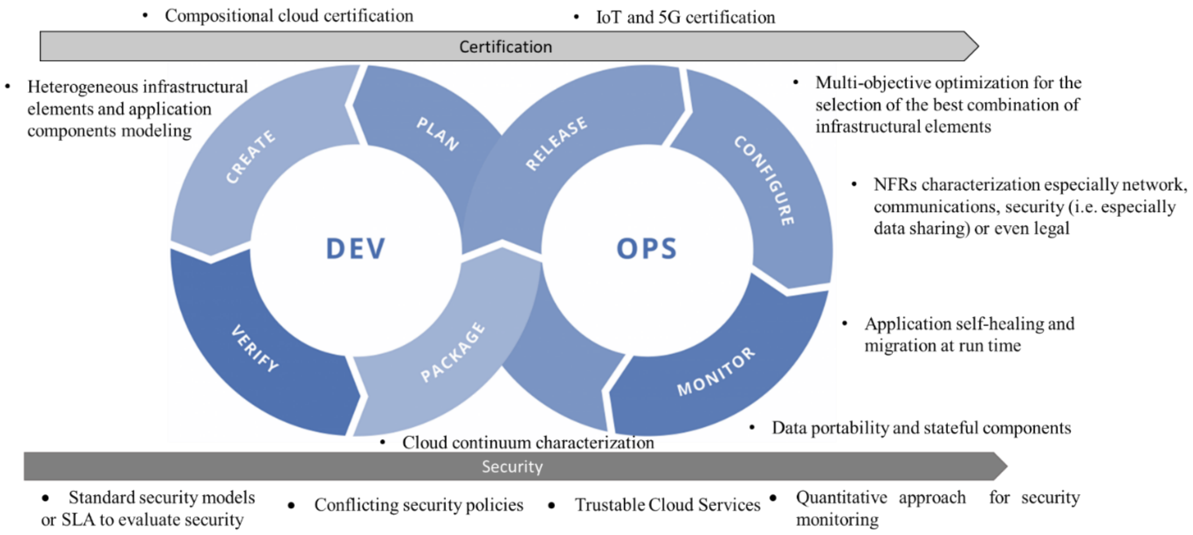
microservice with the required (network) conditions; 2) handling persistent data storage, during a redeployment; 3) data Base automatic configuration so that it can be re-deployed without manual intervention".

- When talking about multi-cloud, **new security challenges** arise but also new opportunities to protect data and services and guarantee confidentiality, integrity, and availability. The adoption of multi-cloud applications has significantly facilitated the usage of file storage for users, trying to overcome some of the common security issues in cloud architectures as loss of availability, loss and corruption of data, loss of privacy, and vendor lock-in. Despite this, specific security issues have been analysed and proper countermeasures have been presented to propose unified approaches, to access control policies and frameworks, to preserve data confidentiality and privacy preserving techniques. In order to improve the trust in these services, it is required not only to ensure adequate transparency and security awareness in multi-cloud environments but also to push for the adoption of standard security models, or SLA to evaluate security, and to develop frameworks to provide, or to assess and monitor security with a quantitative approach. As an example, cloud users may have concerns about what CSPs intend to do with their (potentially confidential) data, and therefore technologies that realize the proper countermeasures to address this issue have been proposed [141]. Nevertheless, regulations and policies need to be put in place so that the consumers of the cloud service are sure that they are consuming secure and trustable resources. Customers of IaaS clouds are responsible for all aspects of their application security and should take the necessary steps to protect their application to address application-level threats in a multi-tenant and hostile Internet environment, and these include the underlying (multi -)cloud services [142].
- Many of these aspects, and especially those derived from the heterogeneity of the current cloud services could be improved or even solved with **the application of recognized of Cloud Standards**. But existing efforts for cloud standardization are still in their first stages and we are not expecting to have a mature solution soon. There exists a big fragmentation in the domain of existing certification schemes, not only in what the controls cover, but also in the conformity assessment methods. Moreover, cloud-vendors themselves are reluctant to unification approaches as they prefer to keep their competitive edge and diversity to attract customers [69,70]. To address this issue the European Commission through ENISA, the European Union Agency for Cybersecurity, has created the EUCS (European Cloud Services Scheme) which would be the European cloud certification framework. The implementation of the measures and requirements derived from such a framework arises new research challenges that will drive the research in the area of Cloud Certification, especially with respect to the following topics: Compositional cloud certification and combination with IoT and 5G certification.

Table 10. *Overview of research trends identified.*

Research topic	Research trends and opportunities
Multi-cloud concept	• Cloud continuum characterization and understanding • Incorporation of new paradigms to the multi-cloud concept: Fog Computing, Osmotic Computing
Multi-cloud by design [RQ3]	• Architectural patterns for multi-cloud native applications • Means and methods to model at high level of abstractions (platform/technology independent) heterogeneous infrastructural elements and application components. • Linking the applications models to the infrastructural models through NFRs characterization especially network, communications, security (i.e., especially data sharing) or even legal
DevOps for multi-cloud	• Lightweight benchmarking and multi-objective optimization for the selection of the best combination of infrastructural elements. • Federation models for the Cloud Continuum, including IoT and networks elements to the traditional Cloud Services. • Application self-healing and migration at runtime, with special focus on data portability and stateful components
Multi-cloud security	• Standard security models or SLA to evaluate security • Conflicting security policies • Frameworks to provide, assess and monitor security with a quantitative approach • Trustable Cloud Services
Multi-cloud certification	• Compositional cloud certification and combination with IoT and 5G certification

Figure 23 provides an overview of the main challenges identified in the Development and Operation of multi-cloud applications.



• Compositional cloud certification
• IoT and 5G certification
Certification
• Heterogeneous infrastructural elements and application components modeling
• Multi-objective optimization for the selection of the best combination of infrastructural elements
• NFRs characterization especially network, communications, security (i.e. especially data sharing) or even legal
CREATE
PLAN
RELEASE
CONFIGURE
DEV
OPS
VERIFY
PACKAGE
MONITOR
• Application self-healing and migration at run time
• Data portability and stateful components
• Cloud continuum characterization
Security
• Standard security models or SLA to evaluate security
• Conflicting security policies
• Trustable Cloud Services
• Quantitative approach for security monitoring

Figure 23. *Overview of the identified research trends in the DevOps lifecycle of multi cloud native applications.*

7.2.2 Extended DevOps lifecycle for multi-cloud native applications

DevOps Principles

DevOps is often considered a methodology or standard but in reality it was conceived as a culture or a set of principles or recommendations grouped into the following main four pillars described in [143] and reported by us in [144]:

1. *"Culture: above all, DevOps means a cultural shift. It is not just a set of tools and practices, it is about establishing priorities and expectations, and how those priorities and expectations are pursued.*
2. *Automation: this is a key concept for DevOps. Everything that can be automated must be automated. Not having to worry about common and repetitive way frees time to dedicate to higher level work.*
3. *Measurement: feedback is an important part of agile and lean practices. Feedback is obtained by measuring, and with a DevOps-oriented mind, everything that moves in production should be measured. These measurements should be then shared with the widest possible audience.*
4. *Sharing: nowadays, organizations are complex and software teams are formed by people with different skills and specialized knowledge. These people must work together in order to be efficient, which is more easily achieved by sharing. As mentioned above, defining metrics, and exposing them to everyone can be greatly beneficial for the organization."*

From these pillars, the DevOps principles can be broken down [144]:

1. *"Small but frequent updates: Small, frequent and more incremental updates allowing organizations to innovate faster. Smaller updates also reduce the risk on each implementation as the encountered errors are smaller and therefore, they are quickly solved. In general, organizations that follow a DevOps model implement updates much more frequently than those that follow a traditional model.*
2. *Use microservices architecture: A microservices architecture divides big and complex systems into smaller and independent projects. This architecture reduces the costs associated to update applications. When a service is assigned to small and agile team, organizations can move forward quickly. The communication between microservices is usually through programming interfaces (APIs) and they encapsulate agile and scalable individual applications that perform a specific business functionality. Microservices are cloud-native in nature, and they are usually operated through containers, which make them more scalable and portable [145]. Some differences with SOA are that microservices use API layers and lightweight protocols such as http or REST, while SOA solution communicates through an ESB using multiple message protocols. Furthermore, SOA is focused on maximizing the application service reusability (being monolithic in nature) while microservices are more focused on decoupling (distributed by nature).*
3. *Continuous integration: Continuous integration is a software development practice that consists in periodically combining the changes in code in a central repository, after which versions and automatic tests are executed. Using this approach, DevOps teams can improve software quality, find, and fix errors faster and reduce the validation and new releases time.*
4. *Continuous delivery: With continuous delivery, changes in code are created, tested and prepared automatically, to be delivered to the production phase. Continuous integration is extended with the implementation of all changes in code on a test and/or production environment. Following a continuous delivery approach every available artefact has been subject to a standardized testing process.*
5. *Continuous deployment: The main goal of continuous deployment is to allow quick, frequent, and reliable automatic deployment of production-ready code. Continuous deployment references production deployments.*

6. *Infrastructure automation (Infrastructure as code). Infrastructure as code is a practice by which infrastructure is provisioned and managed using code and software development techniques, such as version control and continuous integration. The infrastructure is created, configured and manage through code avoiding manual errors. DevOps teams can interact with infrastructure with code-based tools and treat said infrastructure similarly to how they treat application code. As they are defined by code, both infrastructure and servers can be implemented quickly with standardized patterns, updated with the latest revisions and versions, or duplicated in a repeatable way.*
7. *Monitoring, use of registries, issue tracking: Organizations monitor metrics and logs to study how applications and infrastructure performance affects the experience that the final user has with their product. By gathering, categorizing and analysing the data generated by applications and infrastructure, organizations can understand how changes and updates affect users, which provides information about the root cause of unexpected problems. Active monitoring is becoming increasingly important since services must be available 24/7 as update frequency increases. Alert creation and real-time data analysis also help organizations to monitor their services in a proactive way.*
8. *Communication and cooperation: The increase in communication and cooperation within an organization is one of the key cultural aspects in DevOps. The use of DevOps tools and the automation of the software delivery process promotes cooperation since it physically gathers the workflows and responsibilities of the DevOps teams. Furthermore, these teams stablish solid cultural rules that revolve around sharing information and facilitating communication, by means of chat applications, project and issue tracking systems and wikis. This approach fosters the communication between development and operation teams, and even among other teams, such as marketing and sales. This allows all departments of the organization to better align with projects and goals.*
9. *Planning: Teams that plan common goals have a better comprehension of the dependencies, can see bottlenecks before they appear and can overcome priority conflicts. Regardless of whether a tool oriented to "blackboard" technologies, such as Kanban, is used or not, what matters the most is to leverage the living assets, not static plans. Special attention should be paid to divide objects into manageable tasks that can be solved quickly. Static plans that are updated weekly and distributed by mail are not the best option. A better alternative would be to use shared planning tools, which allow to easily see each team member's progress in real time, and to hold conversations between different teams in a collaborative way.*
10. *Security: In a DevOps environment, attention to security is vital. Infrastructure and company assets must be protected, and, when problems arise, they must be addressed quickly and effectively [146]."*

Extended DevOps Principles

This research work presents proposes an extension of the discussed traditional DevOps principles with the objective of covering the complete SDLC and SOLC of multi-cloud native applications.

The aim is customized and extend the traditional DevOps activities to cover steps that are specially needed in the context of multi-cloud software development and operation life cycle. With this in mind, the proposed "Extended DevOps" complements the aforementioned principles with the following activities:

1. Continuous architecting and profiling: Due to the complex nature of multi-cloud execution environemnts, the design and architecture of multi-cloud native applications needs to be considered from the early stages in the SDLC. "Extended DevOps" includes (and supports) this principle so that continuous delivery and continuous integration of the application are enriched with the incroporation of best practices and design principles for multi-cloid applications.

2. Continuous pre-deployment: In a multi-cloud scenario, the selection of the resources/services where to deploy the different components is a complex process as Functional and non-functional requirements of each component need to be considered and aligned with existing resources available. This is the reason why "Extended DevOps" includes this "pre-deployment" activity to be considered in the cycle. We propose the incorporation of means to analyze alternative cloud deployment options along with their impact in performance of the application with respect to the selected NFRs (e.g., security, performance, cost) and therefore in the multi-cloud application SLA (MCSLA). This activity is supporting the developers and operators in the selection of the best cloud deployment alternatives.
3. Continuous proactive self-healing and reconfiguration: The classical DevOps principles include the continuous integration and continuous deployment where the integration of changes/updates in the software are supported. "Extended DevOps" proposes to advance on this direction including a new activity in the SOLC: the continuous proactive self-healing and reconfiguration. It comprises two main activities: the assessment of the multi-cloud application SLA (and indirectly the SLAs of the underlying cloud resources) and the semi-automatic self-healing and reconfiguration into the new execution environment.

Overall approach

This section describes the first outcome of the Thesis as per Figure 22, i.e., the Extended DevOps concept for multi-cloud native applications.

Both developments and operators are the main responsible in designing, developing, and provisioning multi-cloud- applications. To support both in the whole process, from design to operation, a DevOps approach is needed so that both teams work together aligned. However, existing DevOps solutions are focused on tailor-made, single-purpose industrial solutions and no generalized, multi-cloud-oriented DevOps framework exists to date[144] .

To address this situation new generation DevOps framework for software developers and operators are needed which support the following requirements:

- Development and operation of multi-cloud native applications whit specific NFR requirements.
- Methods and means to design, develop, and dynamically (re-)deploy multi-cloud native applications
- Enabling an ecosystem of reliable, interoperable, monitored and legally aware cloud services

Figure 24 illustrates the overall approach and the activities of the proposed Extended DevOps workflow which are detailed in the next subsection.

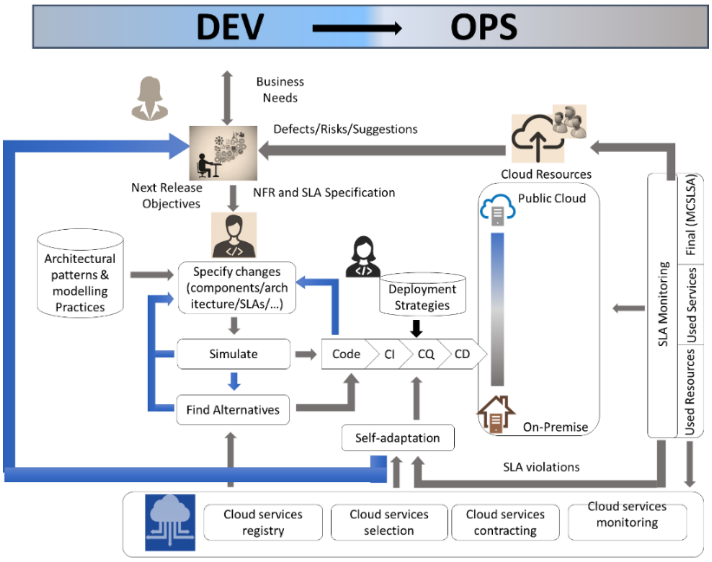


Figure 24. *DevOps workflow for multi cloud applications. Source: Authors' own contribution.*

The Figure represents the activities that the developers and operators of multi cloud applications should need to perform during their lifecycle. This workflow is based on the traditional DevOps workflow but with the incorporation of new activities that are needed to cover the specific needs of multi-cloud applications, deriving in one of the contributions of this thesis: the novel approach for DevOps, targeting multi-cloud native applications, named “Extended DevOps“ approach.

Based on the presented analysis of current practices and existing solutions with respect to the software engineering methodologies and approaches covering both the SDLC and SOLC of multi-cloud native applications, this thesis proposes to extend the “traditional” DevOps approach on both its axes: namely Dev and Ops, to address multi-cloud software needs [74].

This novel “*Extended Devops*“ concept include new phases both in the Development part and in the Operations part of the “traditional DevOps” which currently do not exist (such as pre-deployment simulation). The details of the activities in Figure 24 are formally described in next subsection.

Proposed Extended DevOps workflow

As introduced, DevOps is a philosophy, a conceptual framework for integrating development and operations of Information Systems into a unique team. In the past few years DevOps practices have been adopted as part of the Agile transformation in IT organizations. Adopting continuous integration principles in their software delivery lifecycle and shortening the distance between the Dev and Ops teams has improved the efficiency of Service Development Lifecycle (SDLC) and Service Operation Lifecycle (SOLC).

However, the current complexity of the software applications (i.e., native multi-cloud applications), pushed by the increasing range of offers for personalized computing resources available as cloud services has made visible “traditional” DevOps shortcomings. These weaknesses are mainly related with activities that the current DevOps practices do not cover, or only cover partially.

Table 11. *Activities or practices that are relevant in the SDLC and SOLC of multi-cloud native applications. Extended from* [144].

Phase	Sub-phase	Multi-cloud applications key activities
Development phase	Design	Support for NFR specification
Development phase	Design	Architectural design for distributed applications over different clouds
Development phase	Pre-deployment	Application profiling/classification and selection of the best available cloud resource for each software element.
Development phase	Pre-deployment	Theoretical deployment generation
Development phase	Pre-deployment	Deployment Simulation
Development phase	Pre-deployment	Cloud services discovery
Development phase	Design/Pre-deployment	(MC)SLA definition
Operation phase	Deployment preparation	Manage different CSPs connectors
Operation phase	Deployment preparation	Cloud services contracting automation
Operation phase	Application deployment	Automatic Deployment
Operation phase	Application deployment	(Deployment) Configuration management
Operation phase	Application Monitoring	Application MCSLA monitoring
Operation phase	Application Monitoring	NFR monitoring
Operation phase	Application Monitoring	CSP monitoring
Operation phase	Application Adaptation	Handle violations
Operation phase	Application Adaptation	Application re-deployment and adaptation
Operation phase	Deployment preparation	Current deployment configuration and history
Operation phase	Application Monitoring	Monitor the usage of the services

The proposed Extended DevOps approach for multi-cloud applications incorporates these activities in the traditional DevOps workflow with the objective of adapting the traditional DevOps philosophy to the specific case and needs of the applications distributed over different cloud resources (multi-cloud applications) [147].

The complete DevOps activities proposed for the Extended DevOps workflow involves both Development and Operation activities of the multi-cloud application lifecycle as shown in Figure 25. In the approach traditional activities such as CD or CI have been maintained (blue boxes) while new ones such as pre-deployment or self-adaptation of the application have been incorporated (grey boxes).

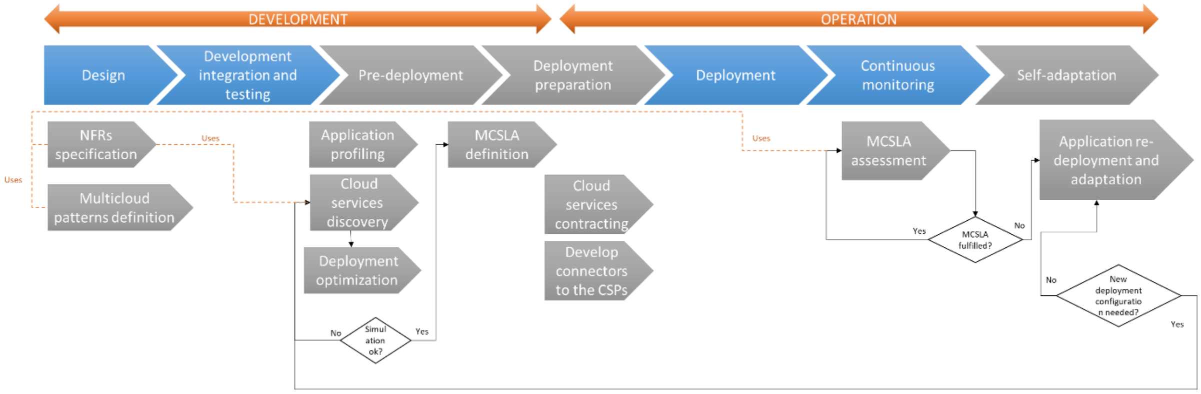


Figure 25. *Activities of the proposed Extended DevOps workflow* for multi-cloud applications. Source: Author's own contribution.

The whole Extended DevOps workflow proposed extends the two axis, development, and operation of the traditional DevOps workflow. In the Development axis, the proposed extension includes the creation of two new phases i.e., Pre-deployment and deployment preparation and the incorporation of new activities in the design phase. Similarly, Operation phase is proposed to be extended with the Self-adaptation phase and the inclusion of a new activity into the continuous monitoring phase. It is to remark that almost all of the activities and phases proposed in the extension (except the definition of multi-cloud patterns) are related to the (pre)/ (re) deployment phases. This is logical because the new needs of multi-cloud applications are precisely related to the fact that they run in a more complex environment.

The extended phases and activities are described in the following:

- Design phase:
  - NFRs specification: The DevOps extended workflow starts from the design of the multi-cloud native application that can be affected by the run-time situation of the heterogeneous and variable (due to external conditions) multi-cloud-based environment. In this situation, developers need to set up quantitative (i.e. performance, availability, response time, lag, cost, throughput) and qualitative (i.e. security, location, cost, legal terms) that the application must comply with. In the case of regular applications, NFRs are considered mainly in the deployment phase but when considering multi-cloud applications this sub-phase takes relevance as NFRs need to be specified not only for the application itself but also per component. This sub-phase is significant as it will be a relevant input for sub-sequent sub-phases such as the application of architectural multi-cloud patterns and the optimization of the deployment configuration.

    Thus, the selected NFRs will be considered during the whole Extended DevOps process:

    1. In the design phase, to adapt the architectural design so that the selected NFRs are fulfilled.
    2. In the pre-deployment phase to optimize the deployment configuration based on the selected NFRs.
    3. In the continuous operation phase to monitor and ensure that the selected NFRs are being fulfilled at run-time.
  - Multi-cloud architectural patterns definition: Multi-cloud applications developers need to prepare the application for a multi-cloud deployment scenario by applying and creating a set of (multi-)cloud patterns. An architectural pattern provides a general, reusable solution to a commonly occurring problem in the development or deployment of software components [30]. In the context of the proposed Extended DevOps framework, we consider two types of patterns, those that provide a description or template for solving a particular problem and are independent of the implementation

details (Cloud Provider Independent Patterns (CPIP)) and those that target specific implementation techniques or rely on the use of specific software components (Cloud Provider Specific Patterns (CPSP)). With the former (CPIP) the formulation of high-level solutions that cover a broad problem space is possible, whereas with the latter (CPSP) tailored solutions are provided, for instance to optimize an application in a very specific context. Apart from this classification, the proposed patterns can be categorized as follows [30] :

1. Fundamental Patterns: They follow the DevOps principles adopted and reflected by the workflow and the best practices for the componentization and de-coupling of applications. Examples can be "Distributed Application" and "Loose Coupling" [30].
2. Optimization Patterns: Patterns to improve the applications NFRs by taking adequate actions at application code level. Optimization patterns can be of multi-cloud nature, or related to specific NFRs, such as elasticity. i.e., the usage of the cloud persistence layer instead of implementing it as part of the application.
3. Development Patterns: These patterns provide the developer with best practices for building multi-cloud applications. The "Stateless" [30] pattern is one example.
4. Deployment Patterns: They with the optimal management of deployment configuration. Deployment patterns are also relevant for the further application of re-adaptability and re-deployment principles for multi-cloud environments.

- Pre-deployment phase:
    - Application profiling: This sub-phase consists in classifying the components (in this case microservices) composing the multi-cloud application. The main objective of this activity is to link the characteristics of those microservices with the group of available Cloud Services features. This activity considers ¡ the core of the application and other elements that the application makes use of. i.e., data bases, processing clusters, communications, etc. For the representation and processing of profiling information existing technologies such as CloudML and OpenTosca can be used.
    - Cloud services discovery: Cloud services discovery covers the identification of the set of the cloud services where the application will be deployed based on NFRs like availability, performance, location, legal level, or cost. In this sub-phase the developers need to deal with the multiple cloud services description available be able to discover services from a set of services available, assuring that the best combination for the user is met and keeping the integrity and security of the overall multi-cloud application. All the CSPs and related cloud services which are suitable for each of the application components need to be considered. To support this sub-phase the modelling of both the CSPs and cloud services is recommended.
    - Deployment optimization: Once the classification of the components is made, and the NFRs gathered, the developer must perform a process where he /she can obtain an optimized schema for the deployment from all the solutions available. The combination of the different optimized possibilities of deployment, will consider the theoretical and individual deployment possibilities for each microservice and the list of cloud services that suit them. In this case, the usage of optimization algorithms such as genetic algorithms, Harmony search, or Dandelion codes could be used to support the achievement the best combination of cloud services that fulfils the established NFRs. In this sub-phase the developers need perform the required changes in the application structure/schema/code to achieve the required configuration deployment.
    - MCSLA definition: Following the proposed Extended DevOps framework, when a multi-cloud native application is ready for deployment its MCSLA has to be defined. The multi-

cloud native application composite SLA (MCSLA) is defined as the accumulation of several SLAs from different CSPs [37]. It has two main objectives; first it is the contract between the end-users and the developer of the multi-cloud native application, second it can be used for monitoring purposes during the continuous monitoring phase, assessing the MCSLA at runtime. A cloud SLA is typically composed of several Service Qualitative Objectives (SQO) and Service Level Objectives (SLO) as defined in ISO/IEC 19086-1. The SLOs and SQOs represent, the non-functional requirements of an application and its underlying infrastructure. The developer needs to manage all the possible SLOs and SQOs, with single and aggregated values. Once a developer has set an MCSLA and communicated it to the end-users it will be assessed at runtime to check if it is being accomplished.

- Deployment preparation phase:
    - Cloud services contracting: Once the deployment schema is selected and the MCSLA is agreed, the process of contracting the services starts. Again, due to the fact that the Extended DevOps workflow is customized for the multi-cloud scenario the application owner needs to establish the different contracts for the corresponding (combination of) cloud services (accomplishing the required MCSLAs). This sub-phase can imply the management of different contract frameworks, methods, and practices.
    - Develop connectors to the CSPs: With the objective of deploying the multi-cloud native application the DevOps team members need to develop and execute the configuration files that will result in the provisioning and configuration of the cloud resources from different providers indicated in the optimized deployment schema. Thus, the required connectors and APIs need to be implemented so that the continuous deployment phase of the DevOps workflow can take place. In the case of multi cloud applications the incorporation of tools supporting the automatization of the deployment phase is crucial to increase the efficiency of the deployment process. Tasks that can benefit from this automatization are:
        1. Provisioning of the needed cloud resources from different providers
        2. Configuration of the provisioned infrastructure
        3. Generation of the IaC for the deployment of container-based applications based on the configuration indicated in the deployment optimization sub-phase
- Continuous monitoring phase:
    - MCSLA assessment: At runtime, the application owner needs to continuously monitor and assess the fulfilment of the established NFRs and MCSLA. This includes the collection of monitoring data from both the application and the cloud providers hosting it, and the assessment of these metrics against the contracted SLOs at different levels (application owner- application users, application owner-cloud services providers). The continuous analysis of the collected metrics will support the identification of possible violations of the MCSLA and the underlying CSLAs in a timely manner, allowing the peed up of countermeasures implementation. This sub-phase covers the detection of any violation according to the SLOs of the NFRs and MCSLA, which will trigger the application re-deployment and adaptation sub-phase.
- Self-adaptation phase:
    - Application adaptation and redeployment: It covers the needed activities for the implementation of the needed corrective activities once a violation of the MCSLA occurs. In the case of a violation of the MCSLA the DevOps team needs to intervene and decide how the adaptation should proceed: a reboot is needed, new cloud resources are needed, the application code needs to be reviewed (i.e., bugs). Redeploying an application may involve stopping it, un-deploying the current configuration and deploying the new one. Application adaptation using redeployment for each SLA

violation may impact business continuity. Techniques to avoid such an impact have been studied and have been reported in [27]. Some of the studied techniques apply to the application and can be described as development patterns (i.e., a microservices architecture, stateless containers, replication), whereas other techniques involve the re-deployment procedure itself (i.e., Blue-Green deployment).

The concept proposed and shown in Figure 26 intends to solve some of the challenges of multi-cloud applications design, development, deployment, and adaptation, providing a novel and extended DevOps approach addressing the identified challenges for the fully adaptation of the current DevOps practices to the multi-cloud paradigm. The objective of the proposed extended DevOps concept is to address totally or partially the encountered challenges [147]:

- Applications need to be responsive to hybrid/multi-cloud model scenario.
- Means shall be provided to manage and assess cloud deployment alternatives to better support cloud resources selection and discovery.
- Existing cloud services shall be made available dynamically, broadly, and cross border.
- The definition and assessment of complex MCSLA needs to be supported.
- Working conditions of the multi-cloud applications need to be monitored and assessed to verify their fulfilment.
- Mechanisms shall be produced to ease the adaptation of the multi-cloud application at run-time, so that the NFR requirements are met.

The proposed extended DevOps concept can be enhanced through the usage of supporting tools or frameworks for the automation of the activities envisioned. With the outcome 2 of this Thesis work, "ACSmI-Advanced Cloud Services meta-Intermediator" we propose a tool to support the automatization of the pre-deployment, deployment preparation and self-adaptation phases and MCSLA assessment sub-phase.

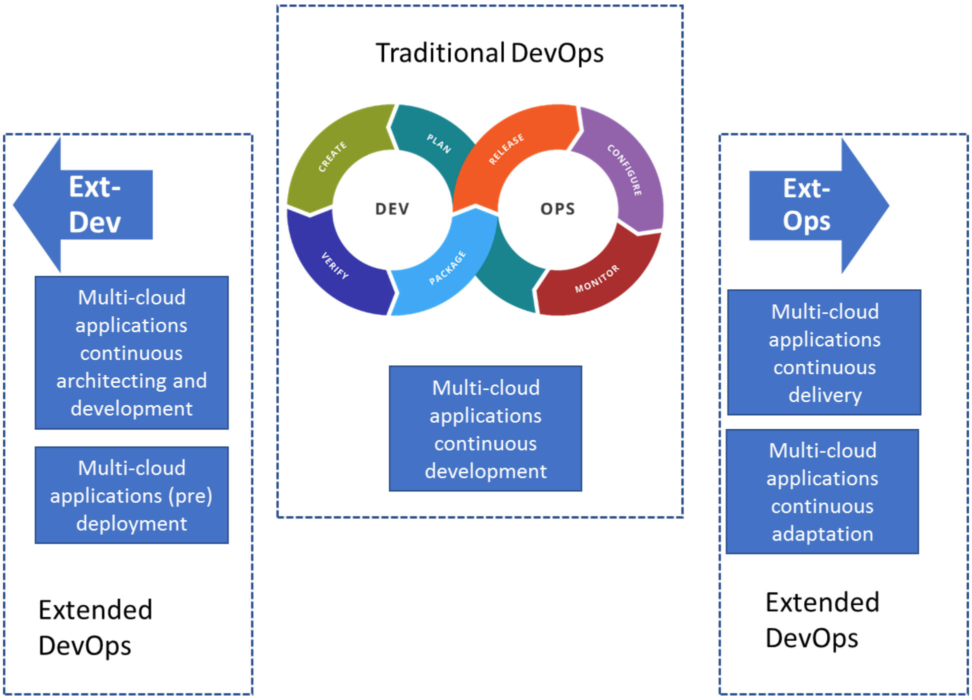

Traditional DevOps
CREATE
PLAN
RELEASE
CONFIGURE
DEV
OPS
VERIFY
PACKAGE
MONITOR
Multi-cloud applications continuous development
Ext-Dev
Multi-cloud applications continuous architecting and development
Multi-cloud applications (pre) deployment
Extended DevOps
Ext-Ops
Multi-cloud applications continuous delivery
Multi-cloud applications continuous adaptation
Extended DevOps

Figure 26. *Extended DevOps concept in a nutshell. Author's own contribution.*

7.3 ACSmI: Continuous cloud services mediation enabler framework

In this section we describe the second main contribution achieved during the research work to fulfil three of the objectives proposed in the Thesis work:

- *Analyse and provide mechanisms to discover and select a combination of cloud services specific for multi-cloud aware applications*
- *Research and provide mechanisms to assess continuous real time verification of the cloud services non-functional properties fulfilment (Composite CSLA) including legal aspects*
- *Study and develop means for seamless change of Cloud service provider enhancing the portability and interoperability of multi-cloud aware applications*

In the next sub-section ACSmI (R2) -Advanced Cloud Service meta-Intermediator, the contribution related to this objective and collected during the research work is described in detail.

7.3.1 Cloud Service Life Cycle analysis for Cloud intermediation

ACSmI can be described and understood considering the different phases that a Cloud Service will go through during its lifecycle (see Figure 27). In this respect, while some standardizations efforts are centred on the application lifecycle (i.e. TOSCA) or in the Cloud SLA lifecycle [148]) up to our knowledge no work has been published centred on the Cloud Services lifecycle. In the case of the intermediation of Cloud Services it is relevant to be aware of the different phases that a Cloud Service passes through from the point of view of the Cloud Service Broker. These phases have been defined considering the different actors that has a role in the ACSmI. They are based on the roles defined by NIST in the Cloud Computing Standards Roadmap [10].

With this deep analysis of the different states that a Cloud Services may go through we intended to acquire a deep knowledge on how and when in the lifecycle of a Cloud Service is identified a gap which may be supported.

ACSmI actors

For the correct understanding of the ACSmI workflow and the lifecycle of the Cloud Service we have analysed the different actors that we envision in such a multi-cloud environment.

Each actor is an entity (a person or an organization) that participates in a transaction or process and/or performs tasks in ACSmI. The main roles defined by NIST are: Cloud consumer, cloud provider, cloud auditor, cloud broker, and cloud carrier. In this section the roles and responsibilities defined in [10] have been customised for ACSmI considering only the relevant ones.

Table 12. *ACSmI actors.*

Cloud services consumer
The cloud services consumer is the final user of the service from the ACSmI. The cloud consumer browses the service registry from the ACSmI, requests the appropriate service based on specific needs, sets up service contracts with the ACSmI, and uses the contracted service. In the context of ACSmI as a supporting tool for the Extended DevOps workflow the Cloud consumer can be: 1. Developer: develops the multi-cloud native application, as well as new functionalities that arise during development, based on the technologies that the client demanded. 2. Operator: ensures the proper working of all machines and systems used in the project, which will guarantee that the platform is working as intended.
Cloud services provider / CSP

From the NIST definition [149], a cloud provider "*can be a person, an organization, or an entity responsible for making a service available to cloud consumers. A cloud provider builds the requested software/platform/ infrastructure services, manages the technical infrastructure required for providing the services, provisions the services at agreed-upon service levels, and protects the security and privacy of the services*". This actor could have two main roles in the ACSmI: 1. Account Owner is the maximum responsible of the account and the relationship between the CSP and the ACSmI 2. Sub-users of the CSP. These are users created by the CSP account owner and they have assigned (through roles) their allowed actions in the ACSmI.
Cloud Broker / ACSmI
Cloud Broker can be understood as a third party that manages the use, provision, and delivery of cloud services, and provides additional added services to cloud consumers such as the integration of cloud services form different providers. The NIST [149] considers that "*Cloud Brokers provide a single point of entry for managing multiple cloud services. The key defining feature that distinguishes a Cloud Broker from a Cloud Service Provider is the ability to provide a single consistent interface to multiple differing providers, whether the interface is for business or technical purposes*". This actor could have three main roles in the ACSmI: 1. CB Administrator. He/she is the responsible of administrating the ACSmI, creating the sub-users and assigning the roles. 2. CB Operator. He/she is the responsible of operating the ACSmI: validation of the user; launching back up processes and so on. 3. Law expert. He/she is in charge of modelling the legislation and doing and monitoring activity of the changes on the European legislation.

Based on these definitions the following table shows in which processes are involved each role and which activity should carry out.

Table 13. *Cloud services consumer activities.*

Cloud Services Consumer	
Process	**Activities**
Intelligent Discovery	Specify the requirements for the service Select the service/aggregation of services to be contracted
Service Contracting	Select and request the contracting of the selected services. Provide the additional information for contracting Accept the conditions of the contract
User management in the ACSmI	Create the Account owner providing the information. Create Sub-user providing the information. If Account owner validate the cancellation or modification of the sub user information If Account owner assign the Roles and policies to each sub-user
Security Management	If Account owner assign the Roles and policies to each sub-user Provide the credentials to work with ACSmI.
Service Withdrawal	If possible, select a service to substitute the one that is going to be de-activated.
Service Contract termination	Request for a service termination. Confirm the service termination.

Table 14. *CSP Activities.*

Cloud Provider /CSP	
Process	**Activities**
Endorse a cloud service into the broker	Request to register a Cloud service in the ACSmI Provide the required information, including the one related to the legislation
Service Contracting	Accept the contract with the ACSmI
CSLA Provision	Provide the CSLA for the service in a machine-readable format (if possible)
User management in the ACSmI	Create the Account owner providing the information. Create Sub-user providing the information. If Account owner, validate the cancellation or modification of the sub user information If Account owner, assign the Roles and policies to each sub-user
Security Management	If Account owner, assign the Roles and policies to each sub-user Provide the credentials to work with the ACSmI.
Legislation compliance update due to changes in the legislation	Provide information about legislation compliance to certificate the service in the registry
Data Migration/portability	Provide the connectors for facilitating the migration
Service Withdrawal	Terminate the contract following the defined process
Service Contract termination	Terminate the contract following the defined process

Table 15. *CB Activities.*

Cloud Broker	
Process	**Activities**
Endorse a cloud service into the broker	Collect the information of the services Check if the service complies with the legislation Update the service registry Inform to the CSP
Intelligent Discovery	Look for a service with the specified requirements Prioritize the list of the services Present the results to the Cloud Service Consumer
Intelligent Discovery	Collect the request of searching for services Look for services according to the requests Present the results of the discovery to the Cloud services consumer. Inform about the selected service to the external Cloud services consumer
Federated Service Contracting	Request for the appropriate information Contracts the service to the CSP Send the contract to the Cloud services consumer Start the operation of the service Inform about the operation of the services Notify the accounting module a new contract exists
CSLA Provision	Look for the CSP CSLAs in a machine-readable format If it does not exist, create the CSP CSLA in a machine-readable format or request it to the CSP.

Cloud Broker	
Process	**Activities**
	Create the ACSmI CSLA based on the CSP CSLA and on the non- functional characteristics Update the service registry with the ACSmI CSLA.
User management in the CB	Create the Account owner providing the information. Create Sub-user providing the information. If Account owner, validate the cancellation or modification of the sub user information If Account owner, assign the Roles and policies to each sub-user Update the user registry
Security Management	Create the policy, assign the functions and select the context Cancel or modify of the policies Create the role, assign the policies Cancel or modify of the roles Provide the information about the roles to one or more users and assign the corresponding roles. Provide the information about the policies to one or more users and assign the corresponding policies. Update the user registry. Check if the credentials are ok and notify it they are incorrect Provide the token for future actions Look for the allowed actions for the user (token)
Creation of the aggregation services	Create the composition script ConFigure the script Launch the CSLA provision process Update the service registry.
CSLA Service Monitoring	Gather the metrics of the CSLA Ask for the metrics to the metering process Compute the values of the metrics for the CSLA terms Assess the fulfilment of the CSLA terms and detect if there is any non-compliance Register the non-compliance.
Legislation compliance update due to changes in the legislation	Law expert monitors if the legislation changes Make a gap-analysis Look for the services that are affected by the changes Send a notification to the CSP
Data Migration/portability	Discover and contract the new services Migrate the data using the appropriate connectors Terminate the contract Notify the affected parts
Service Metering	Create, delete and modify the metering mechanism Create the filter and enquire some metrics/values from the registry
Billing user	Obtain the billable concepts Ask for the information to the Service metering and calculate the information according to the billable concepts Generate the bill
CSP Costs estimation	Ask for the information to the Service metering and calculate the billable values for the CSP. Notify the accounting department the estimation of the costs

Cloud Broker	
Process	**Activities**
Service Withdrawal	Check if the service to be withdrawal has a contract Notify the affected part the withdrawal of the service Launch the withdrawal protection if it is contracted Manage the portability if a new service is selected If no service is selected, notify the time that the services is going to be active. Terminate the contract with the CSP
Service Contract termination	Gather the request for termination Inform about the implications of terminating the contract to all affected parts generating a contract termination report. Terminate the contract with the CSP Update the service contract registry.

In the next sub-section in Figure 27 related activities have been grouped into processes and phases.

Cloud Service LifeCycle in ACSmI

In the ACSmI the cloud service will pass through the following phases and processes during its lifecycle [9]:

- Service initialization, including cloud service endorsement into the broker, intelligent discovery of services, service contracting, CSLA provision, Users management, Security Management.
- Service operation, including CSLA monitoring, legislation compliance assessment, data migration/portability between cloud services, service metering and billing, and costs estimation.
- Service termination: including service withdrawal and service contract termination

In the following picture, the graphical representation of the ACSmI value chain, including its phases, processes and results is depicted.

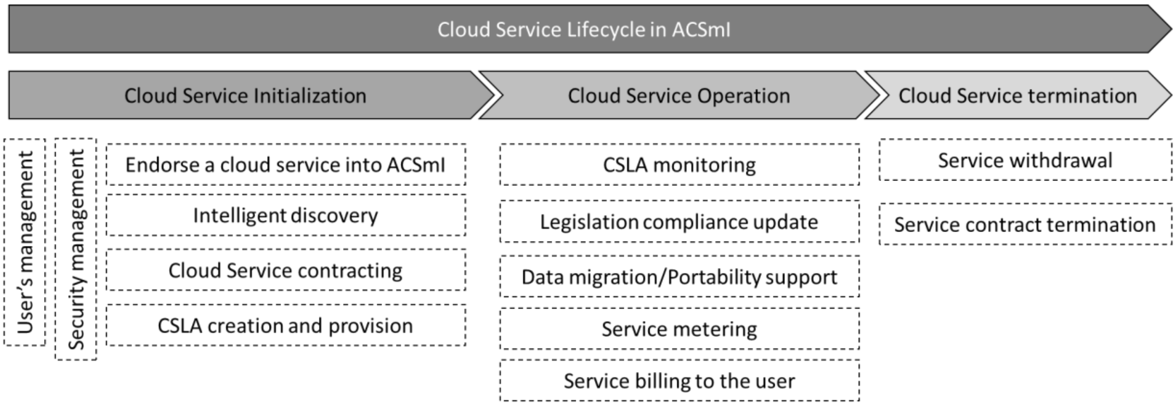

Cloud Service Lifecycle in ACSmI
Cloud Service Initialization
Cloud Service Operation
Cloud Service termination
User's management
Security management
Endorse a cloud service into ACSmI
Intelligent discovery
Cloud Service contracting
CSLA creation and provision
CSLA monitoring
Legislation compliance update
Data migration/Portability support
Service metering
Service billing to the user
Service withdrawal
Service contract termination

Figure 27. *Cloud Service Lifecycle in ACSmI. Source: Author's own contribution* [8].

In order to perform the corresponding design, the behavioural analysis of the processes presented was performed. The detailed analysis of these processes and the related sequence diagrams are provided as supplementary information Appendix A.

7.3.2 ACSmI functional requirements

As explained in section 5, the functional requirements for ACSmI have been elicited from the analysis of the current needs for cloud services intermediation and existing solutions and their lacks, needs collected form the test cases or other sources such as the knowledge of the developers of the solution. The functional requirements have been grouped into the main functional blocks for ACSmI.

These requirements have been documented following the same template and have been reported in the as part of the documentation of the different releases of ACSmI in DECIDE project [50] [102]:

CSPs service discovery and management

Req. ID	DIS01
Req. Short Title	Endorse a cloud service into the ACSmI.
Req. Description	CSPs or the ACSmI administrator register(s) one of its services the service registry. The description of each service includes the different terms defined by the CSPs for their services. This will allow the discovery of all the services from the registry in a common way.
Phase/sub-phase of the Extended DevOps workflow	Operation phase/Pre- deployment preparation
Supported Functionality of the Extended DevOps workflow	Create and update the service registry into the ACSmI
Source	Challenges analysis
Priority[13]	High

Req. ID	DIS02
Req. Short Title	Specify a set of (non-)functional requirements for the services.
Req. Description	The NFRs of the multi-cloud application shall be informed to ACSmI so that optimal services from the service registry are selected. Through this specification, services in the registry can be compared and prioritized according to the expressed requirements.
Phase/sub phase of the Extended DevOps workflow	Operation phase/ Pre-deployment
Supported Functionality of the Extended DevOps workflow	Cloud services discovery
Source	Challenges analysis
Priority	High

Req. ID	DIS03
Req. Short Title	Discovery of services
Req. Description	The objective is to provide a list of services from the services registry that fulfil (totally or partially) the requirements specified by the ACSmI operator. These requirements are specified as explained in DIS02.
Phase/sub phase of the Extended DevOps workflow	Operation phase/ Pre-deployment

[13] The priority was set up in by the developers of ACSmI and the test cases and was used to prioritize the implementation of the different features in the different versions of ACSmI

Supported Functionality of the Extended DevOps workflow	Cloud services discovery
Source	Challenges analysis
Priority	High

Req. ID	DIS04
Req. Short Title	Automatic Discovery of services
Req. Description	The objective is to provide a list of services from other ACSmIs that fulfil (totally or partially) the requirements specified by the Cloud Consumer (multi-cloud application). These requirements are specified in the DIS01.
Phase/sub phase of the Extended DevOps workflow	Operation phase/ Pre-deployment
Supported Functionality of the Extended DevOps workflow	Cloud service discovery
Source	Challenges analysis
Priority	Low

Req. ID	DIS05
Req. Short Title	Benchmark of services
Req. Description	The discovered services (DIS03, DIS04) shall be prioritized based on the level of fulfilment of the NFRs expressed by the ACSmI operator. As a result ACSmI will provide a sorted list with the degree of fulfilment.
Phase/sub phase of the Extended DevOps workflow	Operation phase/ Pre-deployment
Supported Functionality of the Extended DevOps workflow	Cloud service discovery
Supported Functionality of the Extended Extended DevOps workflow	Challenges analysis
Priority	High

Req. ID	DIS06
Req. Short Title	User management.
Req. Description	The objective is to provide means to create, read, update, and delete (CRUD) objects in the users´ registry. And assigning them different types of roles: CSP, multi-cloud application operator, multi-cloud application owner, ACSmI administrator and ACSmI operator. This requirement is related to the Roles management (SEC01) and Security Policy management (SEC02).
Phase/sub phase of the Extended DevOps workflow	Operation phase/ Pre-deployment
Supported Functionality of the Extended DevOps workflow	Cloud services discovery/ Could services monitoring/ Cloud services contracting.
Source	Challenges analysis
Priority	High

Req. ID	DIS07
Req. Short Title	Service registry management

Req. Description	The objective is to provide means to create, read, update, and delete the services registry objects and related information such as which multi-cloud application is using the service, SLAs violations, or legal compliance information. The service registry management shall be aware of the SLA violations occurred as well as non-legislation compliance (MON06 & LEG01) in order to have the most up to date information in the registry.
Phase/sub phase of the Extended DevOps workflow	Operation phase/ Pre-deployment
Supported Functionality of the Extended DevOps workflow	Cloud service discovery
Source	Challenges analysis
Priority	High

Req. ID	DIS08
Req. Short Title	User Interface personalization
Req. Description	The ACSmI GUI shall be personalized based on the role of the user. Only the allowed information and permissions to perform certain tasks will be showed to each user.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation
Supported Functionality of the Extended DevOps workflow	Dashboard management
Source	Challenges analysis
Priority	High

Req. ID	DIS09
Req. Short Title	Service withdrawal
Req. Description	The objective is to remove a service from the service registry when needed. In the removal process multi cloud applications using those services need to be informed.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation Operation phase/Application monitoring
Supported Functionality of the Extended DevOps workflow	Cloud service contracting CSP Monitoring
Source	Challenges analysis
Priority	Medium

Dynamic monitoring of CSPs SLAs

Req. ID	MON01
Req. Short Title	Define the firewall port (Standard open ports)
Req. Description	The objective is to define the needed default firewall policy to be established before every deployment.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation

Supported Functionality of the Extended DevOps workflow	Cloud service contracting
Source	Other
Priority	Medium

Req. ID	MON02
Req. Short Title	Define the monitoring method (Push or pull).
Req. Description	The objective is to offer the "push" and "pull" monitoring methods based on the different needs. • "Push Monitoring": No additional facilities or agents are required. As it does not need additional software installation, the monitoring activities will not impact the performance. Only metrics provided by the CSP can be considered. • "Pull Monitoring": Installation of different types of software / agents on the cloud server where the application is deployed to monitor specific metrics. This method allows monitoring any aspect/parameter/process of both the application and the Cloud Server independent from the Cloud Provider.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation
Supported Functionality of the Extended DevOps workflow	CSP monitoring
Source	Other
Priority	Medium

Req. ID	MON03
Req. Short Title	Define the monitoring parameters
Req. Description	The objective of this requirement is to align the different SLA terms and NFRs, to the parameters to be monitored by ACSmI.
Phase/sub phase of the Extended DevOps workflow	Operation phase/ Application Monitoring
Supported Functionality of the DevOps framework	CSP monitoring
Source	Other
Priority	High

Req. ID	MON04
Req. Short Title	Manage the list of parameters to be monitored
Req. Description	Based on the SLA contracted and the selected NFRs, the list of parameters to be monitored shall be selected from the generic list of parameters (MON03)
Supported Functionality of the Extended Extended DevOps workflow	Operation phase/Application Monitoring
Supported Functionality of the Extended DevOps workflow	CSP monitoring

Source	Challenges analysis
Priority	High

Req. ID	MON05
Req. Short Title	Check MCSLA from the multi cloud application
Req. Description	The objective is to gain access to the composite MCSLA created by the developer of the multi-cloud application in order to parse the parameters to be monitored.
Phase/sub phase of the Extended DevOps workflow	Operation/Application Monitoring
Supported Functionality of the Extended DevOps workflow	CSP Monitoring
Source	Challenges analysis
Priority	High

Req. ID	MON06
Req. Short Title	Alert of an SLA violation
Req. Description	If a SLA parameter is not fulfilled ACSml needs to inform the user.
Phase/sub phase of the Extended DevOps workflow	Operation/Application Monitoring
Supported Functionality of the Extended DevOps workflow	CSP Monitoring
Source	Challenges analysis
Priority	High

Req. ID	MON07
Req. Short Title	Get monitored values for a given parameter
Req. Description	The objective of this requirement is to provide the ACSml user with the actual values monitored for a certain cloud service.
Phase/sub phase of the Extended DevOps workflow	Operation/Application Monitoring
Supported Functionality of the Extended DevOps workflow	CSP Monitoring
Source	Challenges analysis
Priority	High

Req. ID	MON08
Req. Short Title	Assess the CSP´s SLA
Req. Description	Implementation of techniques to assess the SLA.
Phase/sub phase of the Extended DevOps workflow	Operation/Application Monitoring
Supported Functionality of the Extended DevOps workflow	CSP Monitoring
Source	Challenges analysis
Priority	High

Req. ID	MON09
Req. Short Title	Get log of violations
Req. Description	All violations shall be logged and informed to the users. The log shall hold the following parameters and values: • CSP Id/info • Violated parameters • Value of violated parameters • Time and date of parameters • Application id • Other data • The log should be read only, hashed and signed by ACSmI
Phase/sub phase of the Extended DevOps workflow	Operation/Application Monitoring
Supported Functionality of the Extended DevOps workflow	CSP Monitoring
Source	Challenges analysis
Priority	High

Req. ID	MON10
Req. Short Title	Define the monitoring parameters
Req. Description	The monitoring component of the ACSmI shall include the definitions on how the monitoring of the parameters is performed.
Phase/sub phase of the Extended DevOps workflow	Operation/Application Monitoring
Supported Functionality of the Extended DevOps workflow	CSP Monitoring
Source	Challenges analysis
Priority	High

Security management

Req. ID	SEC01
Req. Short Title	Roles management
Req. Description	The objective is to provide means to create, delete and modify roles in the ACSmI to be assigned to the users (DIS06). The main roles envisioned are: CSP, multi-cloud application operator, multi-cloud application owner, ACSmI operator and ACSmI administrator.
Phase/sub phase of the Extended DevOps workflow	Operation phase/ Pre-deployment
Supported Functionality of the Extended DevOps workflow	Cloud service Discovery Cloud service contracting CSP Monitoring
Source	Challenges analysis
Priority	High

Req. ID	SEC02
Req. Short Title	Security Policy management
Req. Description	The objective is to provide means to create, delete, and modify policies in the ACSmI to be assigned to the roles.

Phase/sub phase of the Extended DevOps workflow	Operation phase/ Pre-deployment
Supported Functionality of the Extended DevOps workflow	Cloud service discovery Cloud service contracting CSP Monitoring
Source	Challenges analysis
Priority	High

Req. ID	SEC03
Req. Short Title	Authentication & Authorization
Req. Description	The objective is to authenticate a user based on the user credentials as well as to provide access to allowed actions considering its role.
Phase/sub phase of the Extended DevOps workflow	Operation phase/ Pre-deployment
Supported Functionality of Extended DevOps workflow	Cloud service discovery Cloud service contracting CSP Monitoring
Source	Challenges analysis
Priority	High

Req. ID	SEC04
Req. Short Title	Communication layer security
Req. Description	Communication layer security using SSL transport layer encryption both between the client and the platform and between the platform and the cloud infrastructures.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Pre-deployment
Supported Functionality of the Extended DevOps workflow	Cloud service discovery Cloud service contracting CSP Monitoring
Source	Challenges analysis
Priority	High

Req. ID	SEC05
Req. Short Title	Data encryption
Req. Description	Users shall be able to store their data encrypted in a cloud storage service. This feature will be optional: a user can select either to encrypt the data stored or to leave them unencrypted.
Phase/sub phase of the DevOps framework	Operation phase/Pre-deployment
Supported Functionality of the DevOps framework	Cloud service Discovery Cloud service contracting CSP Monitoring
Source	Challenges analysis
Priority	High

Req. ID	SEC06
Req. Short Title	Secure API access in ACSmI

Req. Description	The objective of this requirement is to allow ACSmI users to setup the configuration for their account.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Pre-deployment
Supported Functionality of the Extended DevOps workflow	Cloud service discovery Cloud service contracting CSP Monitoring
Source	Challenges analysis
Priority	Medium

Req. ID	SEC07
Req. Short Title	Client data backup and archiving
Req. Description	This feature will allow to backup and archive ACSmI users´ information so that in case of need or emergency, they could be easily recovered. This will ensure ACSmI´s data integrity and safety.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation
Supported Functionality of the Extended DevOps workflow	ACSmI set-up
Source	Challenges analysis
Priority	Low

Legislation compliance and monitoring

Req. ID	LEG01
Req. Short Title	Legal level
Req. Description	ACSmI shall be able to show what the legal level is of the cloud service in question. This legal level is defined once the assessment of all the legally relevant aspects is done. It will consider terms that regulate the termination of a service, e.g. data format on exit, data portability, etc.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation
Supported Functionality of the Extended DevOps workflow	ACSmI set-up
Source	Challenges analysis
Priority	High

Business modelling implementation

Req. ID	BUS01
Req. Short Title	Monitor and control the service status.
Req. Description	The objective is to check the service status via the ACSmI (e.g. if the service is operational or not).
Phase/sub phase of the Extended DevOps workflow	Operation/Application Monitoring
Supported Functionality of the Extended DevOps workflow	CSP Monitoring
Source	Challenges analysis
Priority	High

Req. ID	BUS02
Req. Short Title	Implement the procedures to get access to a service
Req. Description	ACSmI shall provide the multi-cloud application operator with details of how the access can be obtained. It is (often) impossible to get instant access to some resources. The CSP may request detailed information from the multi-cloud application operator. After the CSP checks the information and decides that the multi-cloud application operator can be allowed to the service, the multi-cloud application operator gets appropriate access. An example of such provider is HLRS (High-Performance Computing Center in Stuttgart [6]): HLRS provides extremely powerful infrastructure for its users, however, requires a specific procedure to be completed before getting a cloud account.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation
Supported Functionality of the Extended DevOps workflow	Application Monitoring
Source	Operation phase/Deployment preparation
Priority	Cloud services contracting

Req. ID	BUS03
Req. Short Title	Charge a user in the background for service usage.
Req. Description	Each user shall be charged for service usage if there are specific prices for this service. To ensure this, a reasonable billing mechanism shall be available. It shall be possible to charge user in a background while the service is being used.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation
Supported Functionality of the Extended DevOps workflow	Cloud services contracting
Source	Challenges analysis
Priority	High

Req. ID	BUS04
Req. Short Title	Provide a user with usage reports.
Req. Description	Since the user is charged on actual service consumption basis, detailed reports related to the resources consumed shall be provided to the user.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation
Supported Functionality of the Extended DevOps workflow	Cloud services contracting
Source	Challenges analysis
Priority	High

Req. ID	BUS05
Req. Short Title	Provide a user with periodical invoices.
Req. Description	The objective is to enable a regular invoicing. Since a user is charged for service consumption, it would be very convenient to bill the user on periodical basis.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation
Supported Functionality of the Extended DevOps workflow	Cloud services contracting
Source	Challenges analysis
Priority	High

Req. ID	BUS06
Req. Short Title	Provide a user with billing details.
Req. Description	The user will be provided with detailed reports related to the resources consumed and costs related to this consumption.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation
Supported Functionality of the Extended DevOps workflow	Cloud services contracting
Source	Challenges analysis
Priority	High

Req. ID	BUS07
Req. Short Title	Contract a cloud service in the ACSmI
Req. Description	This requirement shall allow contracting a service or services in the ACSmI for a certain multi-cloud application owner.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation
Supported Functionality of the Extended DevOps workflow	Cloud services contracting
Source	Challenges analysis
Priority	High

Req. ID	BUS08
Req. Short Title	Contract a service directly with the CSP
Req. Description	This requirement shall allow developer to contract a service or services directly with the CSP. ACSmI will require the information for the contracted services (SLAs) to be included in the registry and to be monitored.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation
Supported Functionality of the Extended DevOps workflow	Cloud services contracting
Source	Challenges analysis
Priority	High

Req. ID	BUS09
Req. Short Title	Manage connectors
Req. Description	This requirement shall generate the APIs required to contract the services and monitor them in different CSPs. This requirement is closely related to the BUS02 requirement.
Phase/sub phase of the Extended DevOps workflow	Operation phase/Deployment preparation
Supported Functionality of the Extended DevOps workflow	Cloud service contracting Cloud service monitoring
Source	Challenges analysis
Priority	High

7.3.3 Overview of ACSmI

ACSmI provides a cloud services marketplace where developers can easily access homogenous and centrally negotiated deals of legally compliant services. It also provides mechanisms to assess continuous real time verification of the contracted cloud services non-functional properties' fulfilment as well as provides legislation compliance enforcement support.

In Figure 28 the main functionalities of the ACSmI are detailed:

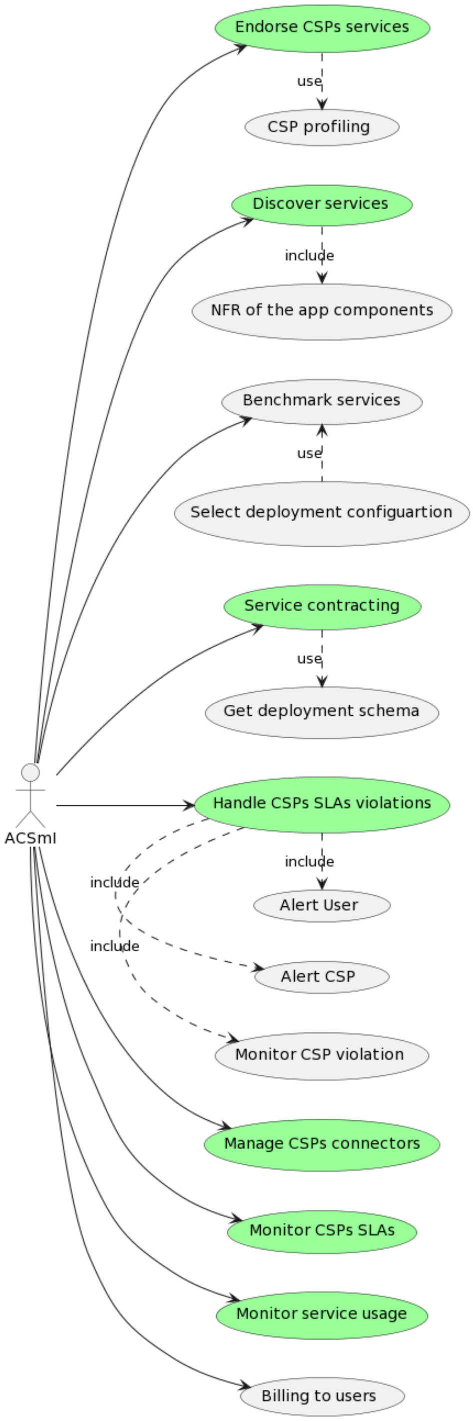


Figure 28. *Main functionalities of ACSmI. Source: Author's own contribution.*

1. Endorse a cloud service into the ACSmI: The register process of services in ACSmI covers the relevant terms in a homogeneous way that allow the discovery of the services from the services registry.
2. Discover and benchmark services: ACSmI user selects the NFR to be fulfilled by the services. ACSmI discovers them from the registry and prioritizes these services in terms of the percentage

of NFRs fulfilment (including legal aspects). This short list is provided to the user with all the relevant information.

3. Contract services: ACSmI provides the mechanism to contract the different services in the registry. Depending on the type of services and the CSP providing such service, the contract is managed following one of the next approaches: 1) ACSmI facilitates contracting services directly by the user to the provider and 2) ACSmI manages the contract itself with the provider and the user. Therefore, ACSmI manages two types of contracts, the contracts with the CSP and the contracts with the users of the services intermediated by the ACSmI.
4. Manage CSPs: ACSmI supports different connectors to facilitate the contracting of the services and to monitor their usage.
5. Monitor NFR CSPs and manage the violation alerts: ACSmI assesses the SLA (NFRs) fulfilment of the service offered by the CSPs to detect any anomaly at run time. If a violation is detected, alerts to the user and the CSP are sent.
6. Monitor the use and bill the user. This functionality calculates the costs made by the user for the use of the ACSmI services, and to provide the corresponding invoice.

To be able to support these activities along the different phases of the Cloud Service lifecycle, Service Initialization, Service Operation and Service termination, the proposed ACSmI conceptual architecture is shown in Figure 29. There are four main components in charge of the implementation of the core functions defined above from the Cloud Service Lifecycle point of view.

The high-level architecture of the ACSmI is presented next.

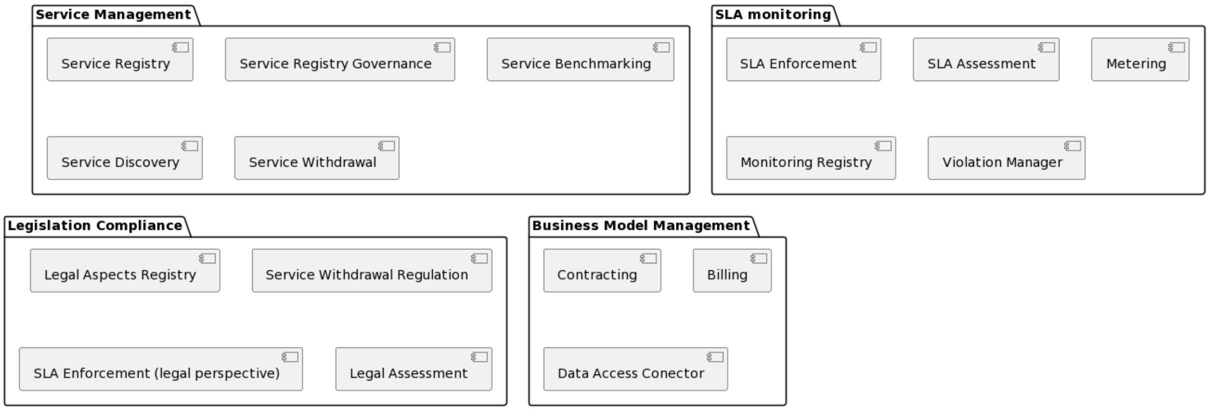


Figure 29. *ACSmI high level conceptual architecture. Source* [102].

ACSmI tries to reconcile both the CSP perspective and the operator of the multi-cloud application perspective providing a set of supporting functionalities during the Cloud Service lifecycle: Cloud Service Initialization, Cloud Service Operation and Cloud service Termination

In ACSmI, four main conceptual components oversee the implementation of the core functions:

1. Service Management oversees the execution and management and control all the activities of the Cloud Service lifecycle in ACSmI, such as: cloud services endorsement, intelligent discovery, or service operation.
2. Cloud Service SLA monitoring implements the monitoring functionalities: 1) SLA terms collection and selection of the metric/parameters associated to each term, 2) collected metrics storage, 3) continuous assessment of the SLA fulfilment of the contracted services 4) information to the user and to the CSP if an anomaly occurs.
3. Legislation Compliance is responsible of the assessment of the information collected from the CSPs with respect to the requirements set by the applicable legislation, as requested by the user

when defining the NFRs. It also ensures the propagation of the changes in the legislation through all the services inside the service registry with the corresponding assessment and also manages the termination of the contracts e.g., data format on exit, data portability, security measures etc.

4. Business Model management which executes and manages all the operations related to Service Contracts in ACSmI and to the financial operations with the different users.

These main conceptual components set up the main contributions of this Thesis work with respect to the outcome 2. The detail description of these conceptual components is provided in the next sub-sections.

CSPs service discovery and management

ACSmI Discovery component, supports the intelligent discovery and the endorsement and withdrawal of services in the service registry.

One of the major contributions of ACSmI is to provide means to support the Cloud Service Lifecycle both from the CSP perspective and from the Cloud Consumer perspective. Thus, ACSmI discovery supports the activities from the service initialization phase, endorsement of the different cloud services by the CSPs, and intelligent discovery from the Cloud Consumers.

An example of the Cloud Service discovery process dialog is shown in Figure 30. Here the different interactions in ACSmI for a Cloud Service Discovery are shown.

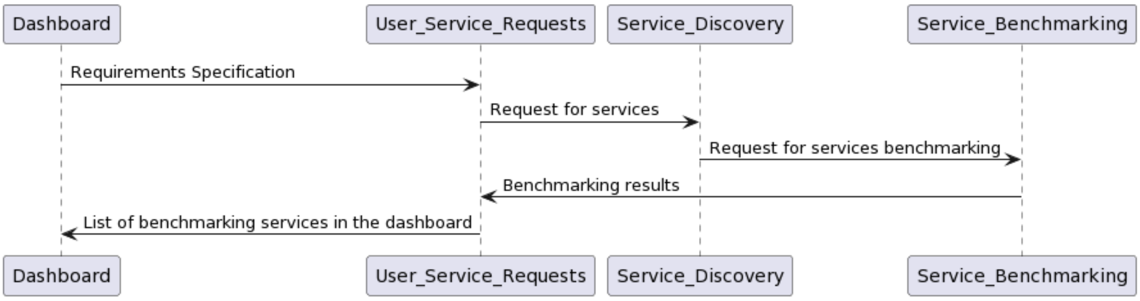


Figure 30. *Intelligent discovery process Sequence diagram. Source* [102].

The discovery and endorsement of the services are performed based on common attributes for each service type. Figure 31 shows the entity model of the ACSmI discovery component with the service attributes of each service type “Storage”, “Database”, and “Virtual machine”[102].

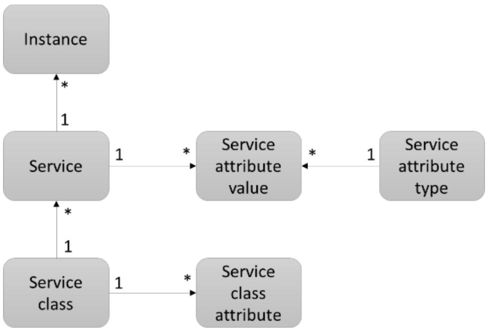


Figure 31. *ACSmI Discovery Service model. Source* [102].

The approach followed allows the easy enlargement of the service classes to new type of services (i.e., edge nodes services in the case of IoT based systems). The rationale behind this design is based on the assumption that ACSmI will integrate both general purpose Cloud vendors such as Amazon as well as small niche oriented ones (i.e. Aimes https://www.aimes.uk/ specialized on the Health sector). Table 16 shows the relationship between the service type and the attributes of each service.

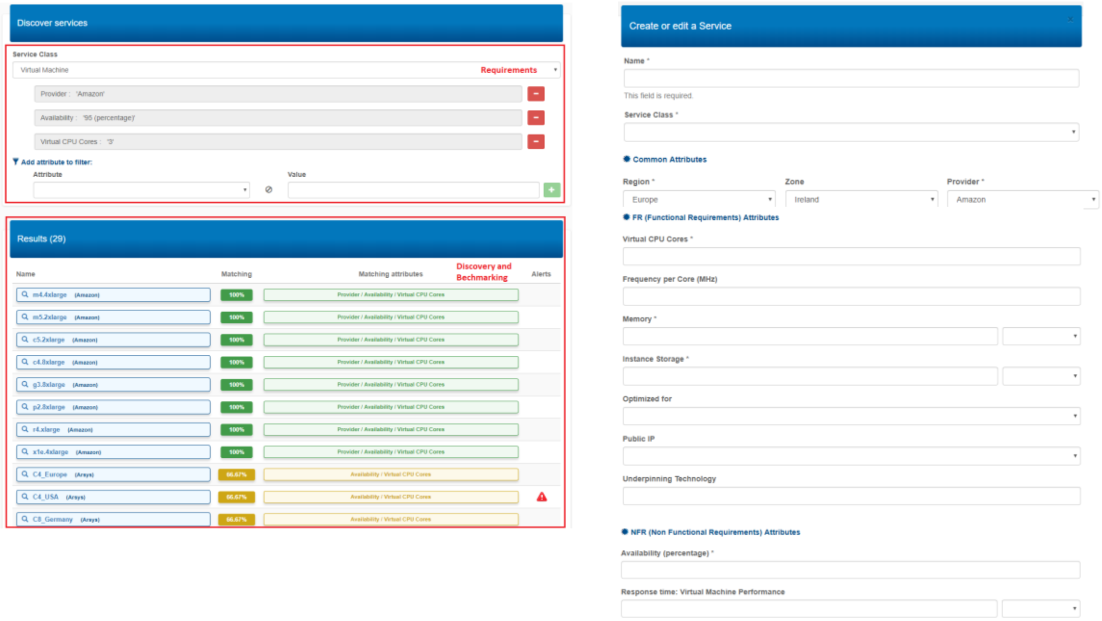
Table 16. *Attributes for each cloud service class in the ACSmI Discovery.*

Storage	DataBase	Virtual Machine
Region	Region	Region
Zone	Zone	Zone
Provider	Provider	Provider
Storage type	Database type	Virtual CPU cores
Storage subtype	Database technology	Frequency per core
Storage capacity	Data transfer IN	Memory
Storage data redundancy	Data transfer OUT	Instance storage
Availability	Virtual CPU cores	Optimized for
Request – Response time: Storage Performance	Database storage capacity	Public IP
Legal certifications/ accreditations	Availability	Underpinning technology
Cost/Currency	Transaction Unit (DTU): Database performance	Availability
	Legal certifications/ accreditations	Response time: Virtual Machine Performance
	Cost/Currency	Legal Level/accreditations
		Cost/Currency

The motivation for the selection of these attributes is twofold, 1) To comply requirements elicited from the test-cases used for the initial validation of the solution 2) To be able to validate the main functionalities of ACSmI related not only to the discovery of those services but also to the continuous monitoring (i.e., non -functional requirements such as performance, availability) or legal awareness (i.e., accreditations of the Cloud Service). Through this intermediate data abstraction layer all the elements of the Services can be compared and benchmarked. While some approaches have been proposed using semantics or other modelling languages, ACSmI proposes a pragmatic approach where the data model has been designed to address the needs of the Use Cases and to provide novel attributes which are not considered in other approaches.

In Figure 32, it can be seen how the different attributes describing the cloud services are used by the CSPs, through the endorsement functionality ("Endorse a service" tab in Figure 32) and for discovering and benchmarking the available services for the Cloud Consumers through the "Discover Services" functionality in the UI which provides the result of the compliant discovered services and their benchmarking result with respect to the selected attributes ("Results" in Figure 8).

Like the service classes, attributes can be extended to include new services properties enabling new functionalities (i.e. internal resource assignment in the CSP [150]). Any CSP that wants to endorse their cloud services in the service registry is required to provide this information (Figure 32).



Discover services

Service Class: Virtual Machine

Requirements

Provider : 'Amazon'

Availability : '95 (percentage)'

Virtual CPU Cores : '3'

Add attribute to filter:

Attribute

Value

Results (29)

Discovery and Bechmarking

Name	Matching	Matching attributes	Alerts
m4.4xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
m5.2xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
c5.2xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
c4.8xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
g3.8xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
p2.8xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
r4.xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
x1e.4xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
C4_Europe (Arsys)	66.67%	Availability / Virtual CPU Cores	
C4_USA (Arsys)	66.67%	Availability / Virtual CPU Cores	⚠
C8_Germany (Arsys)	66.67%	Availability / Virtual CPU Cores	

Create or edit a Service

Name *

This field is required.

Service Class *

Common Attributes

Region *: Europe

Zone: Ireland

Provider *: Amazon

FR (Functional Requirements) Attributes

Virtual CPU Cores *

Frequency per Core (MHz)

Memory *

Instance Storage *

Optimized for

Public IP

Underpinning Technology

NFR (Non Functional Requirements) Attributes

Availability (percentage) *

Response time: Virtual Machine Performance

Figure 32. *ACSmI discovery UI for the CSPs (Cloud Service Endorsement) and Cloud Consumers (Cloud Service Discovery). Source: Author's own contribution .*

Dynamic monitoring of CSPs SLAs violations

ACSmI Monitoring assesses the fulfilment of the SLAs for each Cloud Service contracted. Accurate monitoring of QoS and SLA verification of cloud services enables additional functionalities (i.e. service selection [88] or real time capacity estimation [89]) verifying and increasing the trust. If this monitoring is performed by a CSB, then the reliability of the results of monitoring is dependent on the trust in that broker with respect to objective monitoring [92] Existing tools such as Nagios or Ganglia provide means to monitor low level metrics of computing resources in general, but still automation on the configuration and calculation of complex metrics to assess CSLAs is still missing, especially when addressing multi-cloud environments. Some attempts [91] have been performed to address the elasticity and scalability inherent to Cloud deployments but with no focus on the monitoring of specific metrics to properly assess actual CSLAs contractually relevant. ACSmI monitoring assesses the SLAs (referred as non-functional properties) of the services offered by the CSPs to detect any violation of the SLAs. If a violation is detected, an alert to the CSP will be sent. In ACSmI, the NFRs to be assessed are performance, availability, location and cost. These have been selected considering the needs and preferences of the test cases used for the validation of the solution. Nevertheless, as with the other modules the design has been made to be extendable with new non-functional requirements. For the current implementation only virtual machines (IaaS) have been considered.

For each of the selected Non-Functional Requirement, related metrics to be assessed have been defined. With the objective of being able to compare and combine the SLA from different CSPs, the metrics are defined and expressed by ACSmI and compared to the SLOs provided by the providers. ACSmI supports ISO/IEC 19086-1:2016 standard for the SLOs and the metrics definition. This standard defines a common cloud SLA building blocks (concepts, terms, definitions, and contexts) that can be used to create Cloud Service Level Agreements (CSLAs). In the ACSmI prototype, the authors have used the MCSLA core library [37] which up to our knowledge is the only practical implementation of the standard. The instantiation of such library for the usage of ACSmI monitoring has implied the definition of the corresponding parameters such as Metric, Expression, Parameter, Remedy, ViolationTriggerRule, and UnderlyingMetricRef, which are terms required by the standard.

The definition of a compound SLA for a multi-cloud application is another issue that needs to be tackled by ACSmI. The former library supports the definition of different aggregation patterns for composed Service Level Objectives. This is crucial when addressing multi-cloud applications, for which the composed Multi-Cloud SLA (MCSLA) is based on the composition of the underlying Cloud services SLAs' on which the different components are deployed. The MCSLA can act as the contract between the end-users and the developer of the multi-cloud native application, and it needs to be assessed at runtime. The fulfillment of such MCSLA depends on the individual cloud services contracted and their own CSLAs.

Figure 33 shows a general configuration of a Cloud Service Broker business environment. ACSmI is in this cloud service broker and acts as is an intermediator between multiple CSPs and a Cloud Service Client (in this case the multi cloud application owner and the ACSmI user). Therefore, when the user requested a service, ACSmI as an intermediator is in SLA relationships among both a CSP and a ACSmI user. For instance, if a service requested by a user is brokered to CSP C (i.e., CSP C provides a service to the user via ACSmI), CSP C and ACSmI need to form an SLA (CSP C, ACSmI); ACSmI and the user need to form a SLA (ACSmI, user). According to business models of a ACSmI, the agreement in SLA (ACSmI, user) can be identical to SLA (CSP, ACSmI,). If a ACSmI, business model is just forwarding a service from a CSP to a user, the ACSmI, usually forwards an SLA, which the CSP guarantees, to the user so that the CSP and user form SLA (CSP, user). Otherwise, a CSB, in this case ACSmI provides value-added services (e.g., customized service, integrated service, and so on) to CSCs, the CSB needs to form and manage SLA (CSB, CSC).

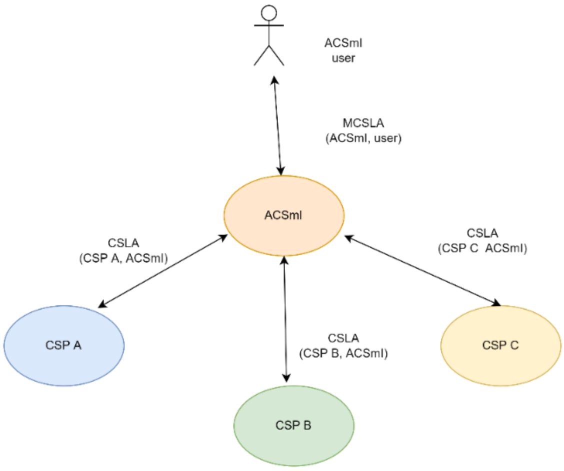


Figure 33. *Business environment in a Cloud Service Broker approach. Source: Adapted from* [37].

In the case of ACSmI and in the context of multi-cloud native applications, ACSmI needs to create and manage its own MCSLA with the user, aggregating the CSLAs of the different cloud services for a given application and a given user.

An MCSLA must therefore act as an aggregator of all terms defined in the various SLAs, as shown in the next formula with Availability:

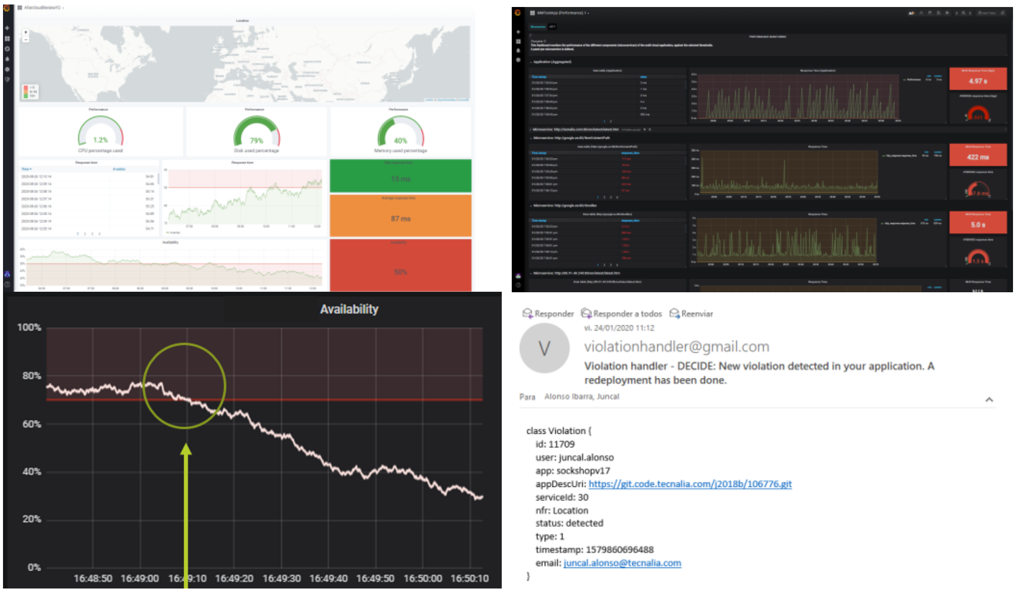
$MultiCloudAvailability\ (term_1,\ldots,term_n) = 100\% - (\sum (100\% - term_i))$

ACSmI combines push and pull monitoring (internal and external approach for monitoring VMs) for cloud resources. Technically this implies the incorporation of preconfigured agents to be installed in the corresponding virtual machines, in what it is called "Extended Internal Adaptive Architecture" [93]. As the definition of the metrics differ from one CSPs to others and being far of having a common approach, the set of metrics to assess the selected NFRs have been specifically defined for ACSmI. This strategy allows to compare, compose, and assess the values in a consistent way for different CSPs. On the contrary, it enforces the CSPs to provide specific metrics for the SLOs defined by ACSmI. In order to support the most standardized metrics, the guidelines defined in ISO/IEC 19086-1 and in the literature [151] and practitioners (i.e. CSPs) have been adopted for the metrics selected:

- Availability:
  Availability = MTBF/(MTBF+MTTR)
  MRBF and MTTR are calculated based on other discrete metrics using different techniques (i.e. responses to the ping command).
- Performance: For the performance the usage of CPU, memory and disk is measured. For the current prototype, ACSmI monitoring is initialized to the 80% of the values in each category (i.e. 80% disk usage) but different thresholds can be configured ad-hoc through the ACSmI monitoring API.
- Location: It determines where a cloud resource is located, geo-locating its IP address from the Service registry. The information that relates the IP addresses with their real time locations is taken from MaxMind GeoIP2 Java API with the free GeoLite2 database.
- Cost: It determines the current cost that a CSP is reporting on a certain resource. The actual incurred cost in a certain period of time (configurable) per Cloud Service is monitored through the ACSmI billing API (described in section 7.3.4).

Thus, ACSmI monitoring provides an implementation of the ISO/IEC 19086-1 standard to be able to monitor metrics in accordance with the CSLA established. Up to our knowledge, ACSmI monitoring is the first implementation for the mentioned standard for a set of concrete NFR (Availability, Performance, Location and Cost).

Figure 34 and Figure 35 show examples of the implemented ACSmI monitoring component GUI. In Figure 34, in the upper side, different monitoring dashboards of cloud services non-functional requirements are shown. In the lower side, the detail of a specific violation of a NFR (in this case availability) is shown. When this occurs ACSmI monitoring generates the corresponding email for the operator where the details of the violation are included. Figure 35 shows the different violations affecting a Cloud Service during its lifecycle in ACSmI. These violations can be then used to assess the convenience or not of selecting this specific Cloud Service in future deployments.



vi. 24/01/2020 11:12

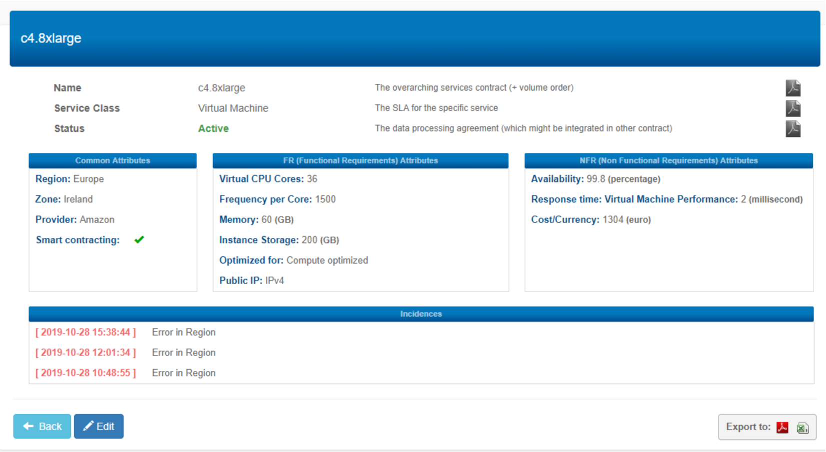
violationhandler@gmail.com

Violation handler - DECIDE: New violation detected in your application. A redeployment has been done.

Para Alonso Ibarra, Juncal

```
class Violation {
  id: 11709
  user: juncal.alonso
  app: sockshopv17
  appDescUri: https://git.code.tecnalia.com/j2018b/106776.git
  serviceId: 30
  nfr: Location
  status: detected
  type: 1
  timestamp: 1579860696488
  email: juncal.alonso@tecnalia.com
}
```

Figure 34. *ACSmI Cloud Resource Monitoring Grafana dashboards and detail of the email received after a location NFR violationSource: Author's own contribution.*



c4.8xlarge

Name c4.8xlarge

Service Class Virtual Machine

Status Active

The overarching services contract (+ volume order)

The SLA for the specific service

The data processing agreement (which might be integrated in other contract)

Common Attributes

Region: Europe

Zone: Ireland

Provider: Amazon

Smart contracting: ✓

FR (Functional Requirements) Attributes

Virtual CPU Cores: 36

Frequency per Core: 1500

Memory: 60 (GB)

Instance Storage: 200 (GB)

Optimized for: Compute optimized

Public IP: IPv4

NFR (Non Functional Requirements) Attributes

Availability: 99.8 (percentage)

Response time: Virtual Machine Performance: 2 (millisecond)

Cost/Currency: 1304 (euro)

Incidences

[2019-10-28 15:38:44] Error in Region

[2019-10-28 12:01:34] Error in Region

[2019-10-28 10:48:55] Error in Region

Figure 35. *Detail of location violations registered in an ACSmI Cloud Service.*

Legislation compliance and monitoring

ACSmI legal contributes to the incorporation of legal related aspects in the Cloud Service Lifecycle. Such aspects might relate to data location, data protection requirements, security level, accountability and control, confidentiality, and contractual terms such as liability and exit clauses. The main technical challenge to be solved, from this conceptual approach, as for any software development framework in relation to the legal aspects of the underlying cloud services, is to assess the complex legal reality of the cloud service's SLA, the security level at the CSP, etc. and to translate them to machine-readable non-functional requirements. For that, ACSmI legal provides the means to characterize the compliance of each regulation by the cloud service, through a questionnaire designed to be able to extract the relevant information from which the legal assurance level of each service is determined. It sets up the machine-readable legal characterization of the cloud services, following a self-assessment approach. At the operation phase, ACSmI legal, is also in charge of ensuring that any changes in the applicable legislation are propagated and that all the cloud service offerings of the ACSmI registry are reassessed when this happens. In the termination phase of the cloud service, the component ensures the terms that regulate the termination of that service, interacting actively with the Business Model Management component to exchange information on the contracts and on possible penalties that need to be considered when the billing process is launched.

The assessment of the legal compliance is based on information to be provided by the CSP, namely:

- The service contract applicable to the service.
- The SLA applicable to the service.
- The Data processing agreement governing the service.
- Any other contract also governing the service, if any.

For the determination of the legal level, only legal controls which information can be assessed "*in concreto*" are considered. Therefore, two types of information are considered: the contractual documents provided by the CSPs when endorsing their service into ACSmI, and other pieces of legal related information that CSPs might be required to provide in addition to those contracts, i.e. by answering a limited list of questions to obtain legal information which is relevant but not (typically) provided in a contract (e.g. the presence of a DPO at the CSP) [152]:

- 8 controls related to the 8 yes/no questions asked to the CSP. They are the "simple controls" and can either be present (✓) or not present (×).

- 26 controls managed by the legal expert through an assessment of the contracts. These “layered controls” ensure a minimum level of legal protection and/or safeguards is/are present. If any of these layered controls are present, no legal level will be assigned, and the service cannot be endorsed into ACSmI. Based on the number/characteristics of the layered controls found the service is classified in one of the following categories: basic legal safeguards present (★), substantial legal safeguards (★★) or strong legal safeguards (★★★).

This leads to a matrix (Table 17) where the relationship of the controls and the legal level is related.

Table 17. *Excerpt of the controls and the related ACSmI legal level.*

Control	**Legal level tier 3 (basic legal safeguards)**	**Legal level tier 2 (substantial legal safeguards)**	**Legal level tier 1 (strong legal safeguards)**
		Simple controls	
Valid company registration	✓	✓	✓
DPO contanct	✓	✓	✓
Representative	✓	✓	✓
Data transfer mechanisms	✓	✓	✓
Data Processing agreement	✓	✓	✓
ISO 27001 or equivalent	×	✓	✓
		Layered controls	
Assessment of Alternative Dispute Resolution mechanisms	★	★	★★
Termination options of CSPs counterparties	★	★	★★
Liability coverage	★	★	★★
Force majeure coverage	★	★	★★

Based on the answers and the analysis by a legal expert of CSP contracts, a legal level will be assigned to the Cloud service being endorsed.

The conceptual approach for the implementation the legal legislation assessment in ACSmI has been carried out with the collaboration of the lawyer Pieter Gryffroy. In [152] all the legal controls used in ACSmI are extensively explained.

Business modelling implementation

ACSmI Business model Management component executes and manages the following core functions:

- Contracting: It establishes the contracts with the CSPs to use their resources through the ACSmI and the CSPs APIs.
- Manage CSPs: To deploy the software onto an infrastructure, the user needs to have an access. The prototype enables the user to provide their own credentials for a concrete CSP or to establish a new contract which can be reused.
- Billing: Including features for setting specific rules for contract(s) billing, tracking the usage, charging consumers and providing billing and usage-related reports both to the provider and to the monitoring component.

The current solution is an extension of the CSB platform (http://cloudbroker.com/). The management of all the possible situations and maturity levels with respect to the automation of the contracting of Cloud services poses a big challenge. ACSmI provides the intermediate layer where the heterogeneities of the contractual process are hidden for the user, while several options for contracting are supported for the CSPs (Figure 36).

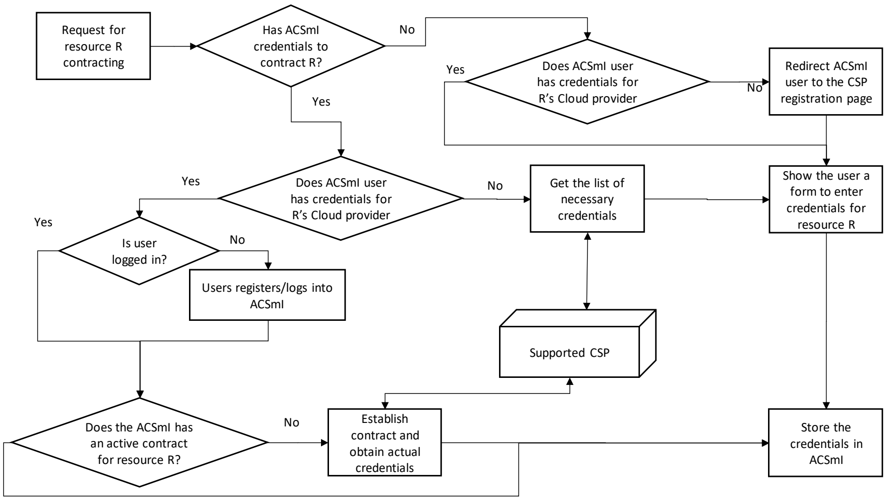

Request for resource R contracting
Has ACSmI credentials to contract R?
No
Yes
Does ACSmI user has credentials for R's Cloud provider
Yes
No
Redirect ACSmI user to the CSP registration page
Does ACSmI user has credentials for R's Cloud provider
Yes
No
Get the list of necessary credentials
Show the user a form to enter credentials for resource R
Yes
Is user logged in?
No
Users registers/logs into ACSmI
Supported CSP
Does the ACSmI has an active contract for resource R?
No
Establish contract and obtain actual credentials
Store the credentials in ACSmI

Figure 36. *Resource contract process in ACSmI. Source: Author's own contribution [8].*

7.3.4 ACSmI technical design and architecture

This section presents the technical design and architecture of the conceptual components conforming one of the main outcomes of this Thesis work. The structural and behavioural representation of the main functional components are described along the next sub-sections.

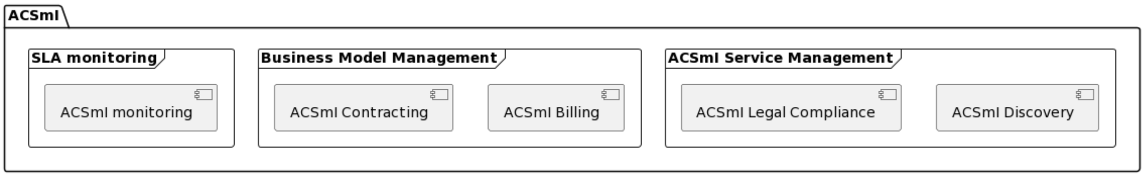


Figure 37. *ACSmI main components diagramme. Source: Author's own contribution.*

ACSmI Service Management

Structural description

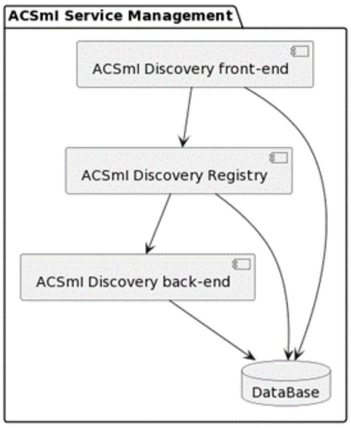


Figure 38. *ACSmI Service Management High-Level Architecture. Source: Author's own contribution.*

The sub-modules included in ACSmI discovery are (for further detail see appendix B where the screenshots of the main features of these modules are included):

1. *Frontend*. This component is responsible of providing the ACSmI GUI. It also manages the different roles defined for ACSmI users (Admin, User/Developer, CSP and Legal expert). The GUI is sensitive to the roles accessing ACSmI and it is automatically personalized depending on the user's role.
2. *Registry*. This component manages the communication between the frontend and the backend.
3. *Backend*. This component oversees 1) implementing the discovery of the services based on the information provided by the user, 2) benchmarking each cloud service including the degree of fulfilment and the attributes that are fulfilled, 3) the endorsement of the services based on the information provided by the CSPs through the frontend and 4) manage the legal compliance for each service.
4. *Database*, which stores the services and all information related to them.

Behavioural description

Figure 39 and 40shows the behavioural description of ACSmI discovery and the interchanged data among the different actors through the next sequence diagrams. Only the most relevant cases have been presented.

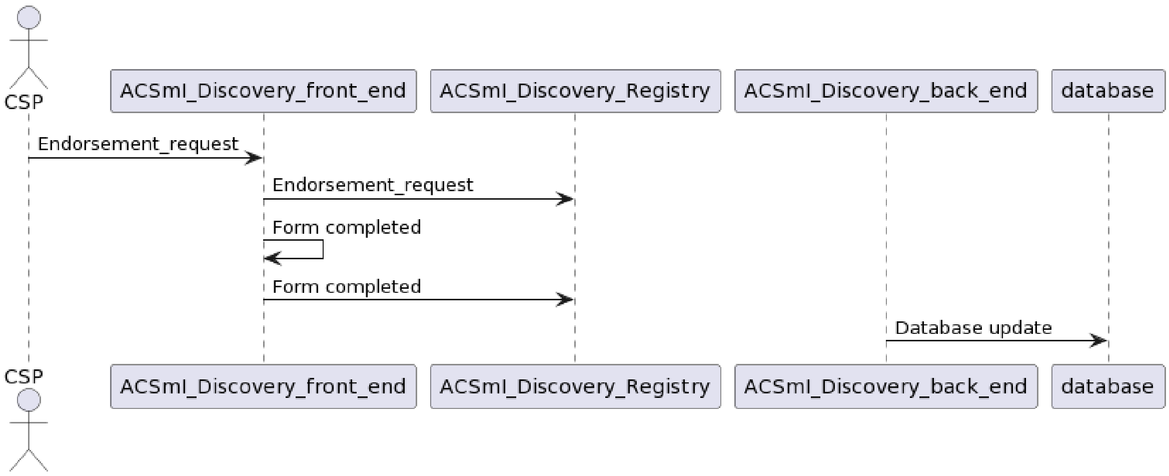


Figure 39. *Endorsement of Cloud Services in ACSmI. Source: Author's own contribution.*

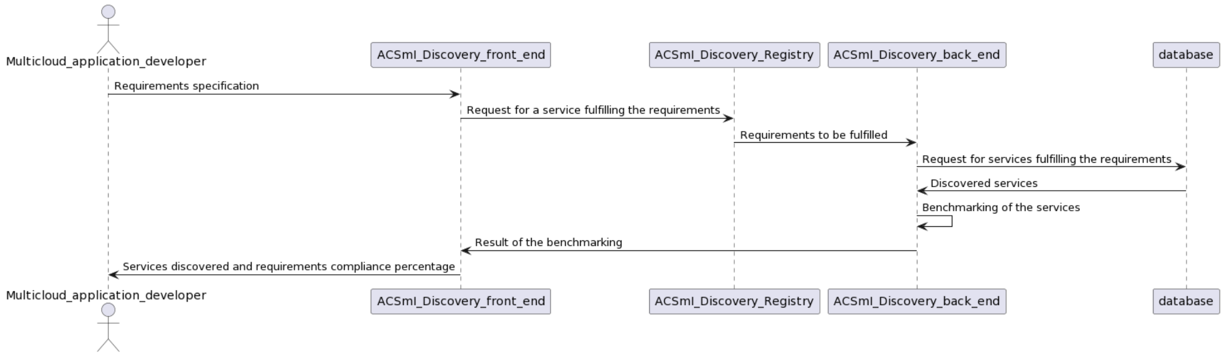

Multicloud_application_developer
ACSmI_Discovery_front_end
ACSmI_Discovery_Registry
ACSmI_Discovery_back_end
database
Requirements specification
Request for a service fulfilling the requirements
Requirements to be fulfilled
Request for services fulfilling the requirements
Discovered services
Benchmarking of the services
Result of the benchmarking
Services discovered and requirements compliance percentage
Multicloud_application_developer
ACSmI_Discovery_front_end
ACSmI_Discovery_Registry
ACSmI_Discovery_back_end
database

Figure 40. *Cloud services discovery in ACSmI. Source: Author's own contribution.*

ACSmI Business model management

Structural description

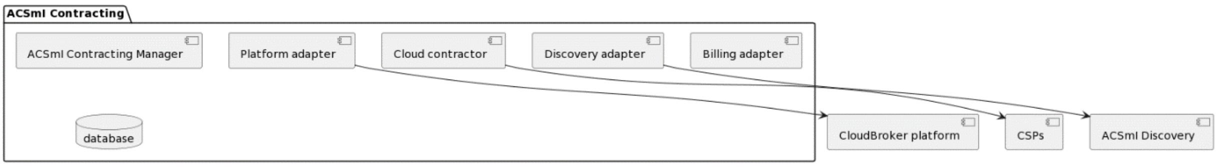


Figure 41. *ACSmI Contracting High-Level Architecture. Source: Author's own contribution.*

The purpose of the different components of the Figure 41:

- *Contracting Manager* manages the different types of contracts supported based on the availability of the user's credentials.
- *Platform Adapter* serves for the communication with the CloudBroker Platform to define the configurations of VMs.
- *Cloud Contractor* allows developers to contract their services directly with the CSPs.
- *Discovery Adapter* serves for the communication with the ACSmI Discovery component to define the service requirements.
- *Billing Adapter* is responsible for setting the specific rules for contracts related activities (i.e., billing, metering, etc.).

This component is realized in ACSmI through the Cloud Broker [153] billing component.

ACSmI SLA Monitoring

Structural description

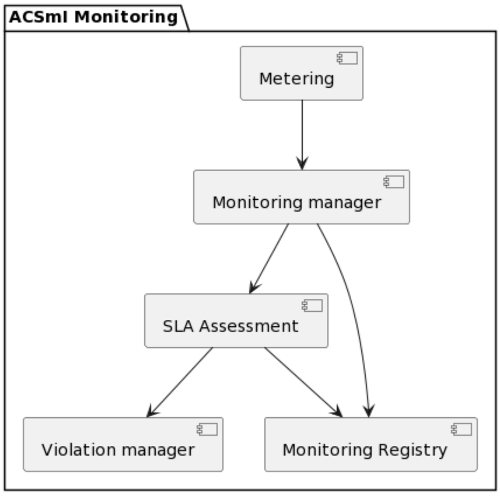


Figure 42. *ACSmI SLA Monitoring High-Level Architecture. Source: Author's own contribution.*

ACSmI monitoring assesses the fulfilment of the CSLA through the following activities:

- *Metering*: This sub-component collects the data from the different cloud services where the application is deployed. This subcomponent gets the raw metrics (explained in section 6.3.3) and stores them in the monitoring registry.
- *Monitoring registry*: This sub-component oversees storing the data collected from the metering sub-component.
- *Monitoring manager*: This component manages all the activities of ACSmI monitoring. It receives the request to start the monitoring of the cloud services form. Then this sub-component configures the different monitoring agents in the metering sub-component and sets up the monitoring registry.
- *SLA Assessment*, this sub-component oversees the aggregation of the different raw metrics to assess the values of the different NFRs with respect to the SLOs.
- Violation's manager: Once the SLA Assessment detects a violation of the SLA CSP, this subcomponent is in charge of carrying out the corresponding activities: 1) inform the user of the cloud service, 2) registry the violation in the ACSmI registry and 3) inform the CSP of the occurred violation.

Behavioral description

The ACSmI SLA Monitoring behavior and the interchanged data among the different actors is shown in the next sequence diagram.

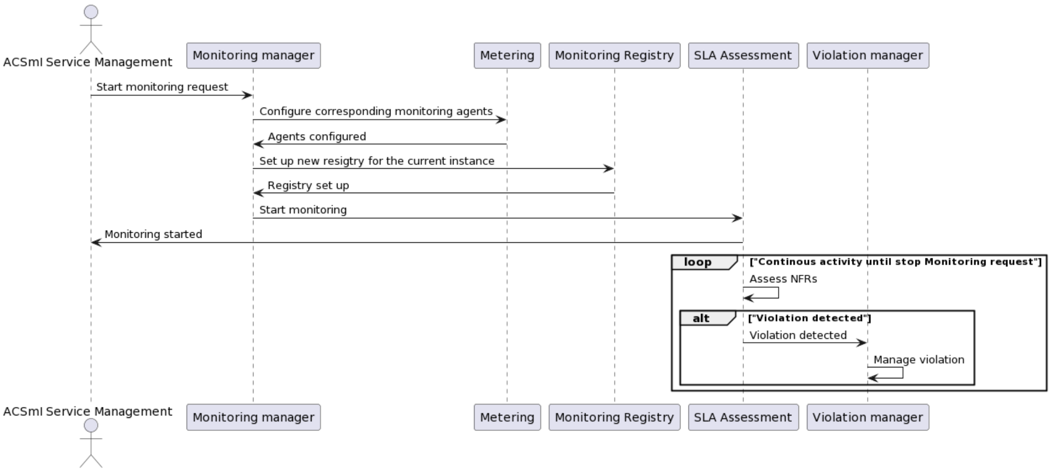


Figure 43. *ACSmI SLA Monitoring sequence diagram. Source: Author's own contribution.*

Technologies used for the implementation and ACSmI APIs

The solution was implemented and deployed for experimentation in TECNALIA premises in Derio (Spain) (the used corresponding technologies are shown in Table 18). The code has been released as open-source code and is available in a gitlab public repository[14].

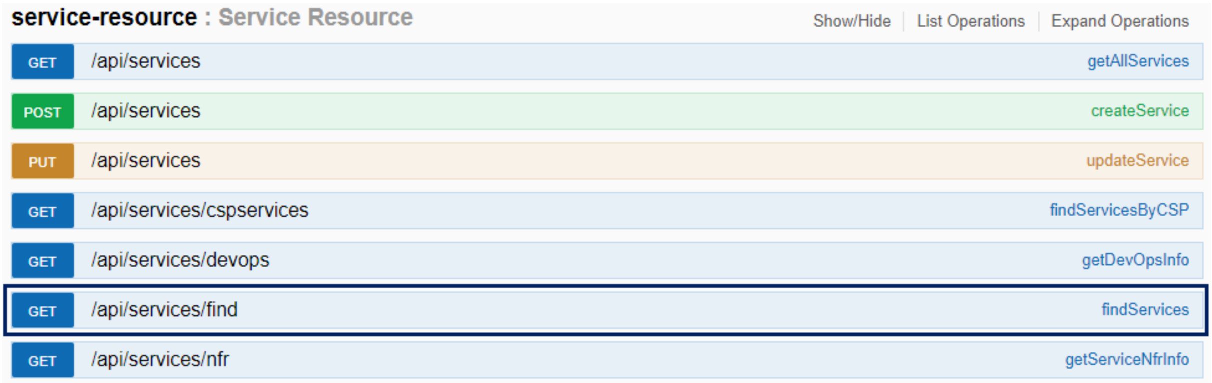
Table 18. *Overview of ACSmI technologies per software component.*

ACSmI component	Baseline technologies	Implementation language
ACSmI service management	JHipster	Java EE
ACSmI Business model management	Cloud Broker Platform	Ruby on Rails
ACSmI SLA monitoring	Telegraf, InfluxDB, MaxMind GeoIP2, MCSLA java library	Java

- ACSmI service management: This prototype has been developed using JHipster [154].

  JHipster allows the use of Swagger [155] to define the communication between the frontend and backend API REST.

[14] https://microservices-demo.github.io/



service-resource : Service Resource

GET	/api/services	getAllServices
POST	/api/services	createService
PUT	/api/services	updateService
GET	/api/services/cspservices	findServicesByCSP
GET	/api/services/devops	getDevOpsInfo
GET	/api/services/find	findServices
GET	/api/services/nfr	getServiceNfrInfo

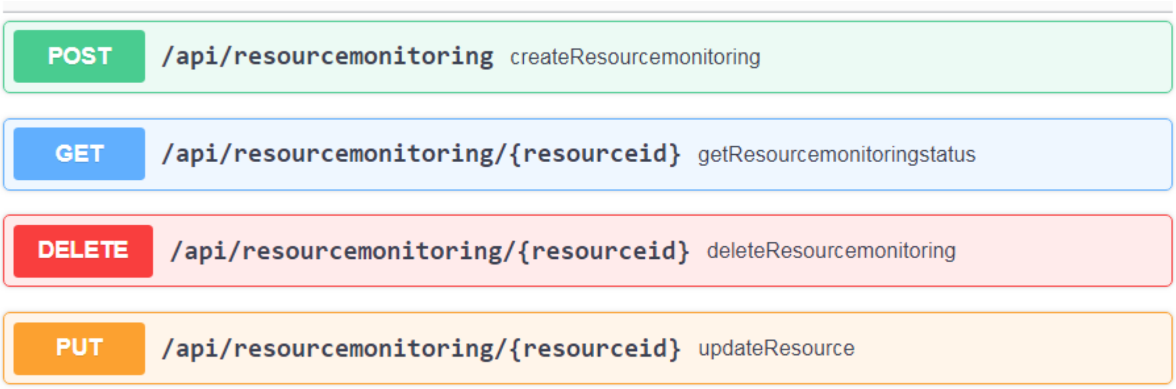
Figure 44. *Example of the API developed for service discovery.*

- ACSmI Business model management: The business model management functional block has been developed based on the CloudBroker platform tool [153]. Technical specifications can be found here [102] .
- ACSmI SLA monitoring: In includes a whole monitoring stack, covering the data collection, the data storage, data aggregation and Cloud service SLA assessment.
- Data collection (Metering subcomponent): For the collection of the different metrics, Telegraf [156] open source technology is used. For the purpose of this final prototype of ACSmI monitoring the five Telegraf plugin (one output and four input) have been configured to acquire the parameters required and indicate where to record them:
    - *Ping* plug-in is used to get the raw metrics that allow to calculate the parameters defined for calculate availability. The raw metric to calculate availability is taken from the field "result_code (int, success = 0, no such host = 1, ping error = 2)". With this metric and the time stamp recorded in the influx DB, ACSmI monitoring can calculate the availability and assess if the SLA is fulfilled or not using the MCSLA core library.
    - In order to assess the NFR of performance, three plugins are used. To obtain the correct values a telegraf container should be deployed in each VM.
        - *mem* plugin. It provides a set of metrics to measure the performance of the VM, the metric used to calculate the performance is "available_percent".
        - *disk* plugin. It provides a set of metrics to measure the performance of the VM, the metric used to calculate the performance is "used_percent".
    - *cpu* plugin. It provides a set of metrics to measure the performance of the VM, the metrics used to calculate the performance is "usage_user" and "usage_system".
    - Output plugins : *Influx DB plugin*. This plugin sends the metrics collected by the agent to a specific Influx DB instance (with a given url).

  Beyond Telegraf, ACSmI monitoring also uses two more mechanisms to collect the information required to assess all the NFR:

    - "*MaxMind GeoIP2*" Java API with the free GeoLite2 database to gather information regarding the location of the VMs.
    - ACSmI billing API to collect the information regarding the cost per each VM.
- Data storage (Monitoring registry sub-component): InfluxDB storage technology has been used to store the runtime metrics in a non-relational data base. It also supports the connection with protocols such Telegraf. The measurements are injected by the Telegraf plugins in the Influx DB database.
- Monitoring manager: It is a Java program that manages the different activities and workflow to be able to start and configure all the elements for the monitoring. It is deployed using a Jetty server.

- Cloud Service SLA Assessment. It is composed of several java classes that compare the SLO introduced by the CSP when endorsing a service (ACSmI Discovery). The current prototype uses the MCSLA library [37] to support this assessment. These java classes are included as part of the monitoring manager.
- Manage violation: this sub-component has been developed as a function of the Monitoring manager. This function alerts to the user and the ACSmI registry that a violation is occurred.



POST /api/resourcemonitoring createResourcemonitoring

GET /api/resourcemonitoring/{resourceid} getResourcemonitoringstatus

DELETE /api/resourcemonitoring/{resourceid} deleteResourcemonitoring

PUT /api/resourcemonitoring/{resourceid} updateResource

Figure 45. *Methods offered by the ACSmI monitoring API for a programmatic monitoring request. Source: Author's own contribution.*

8. Solution validation: Qualitative and quantitative viability

8.1 Introduction

The previous chapter described the two outcomes proposed by this research work: The Extended DevOps concept for multi-cloud native applications and the implementation of cloud service intermediator for multi-cloud native applications solution. This chapter presents the quantitative and qualitative evaluation of the proposed outcomes in four testing cases, three of them industrial and a fourth academic one [157]. Industrial test cases were used to validate the solution and provide insights on the performance of the proposed system through the evaluation (qualitative and quantitative) of the usage of the tools and their impact in the industrial process evaluated. The 4th test case, the academic one, was used to continuously perform the development testing process through the Sock Shop Application [157]. It is a loosely coupled microservices demonstration application which simulates an e-commerce website that sells socks. Available as open-source software it has been developed with the intention of aiding in the demonstration and testing of microservices and cloud native technologies, and with this purpose was used in this research Thesis. It has served as an additional "test case" application to test each stage of the development of the system and ensure the scalability of the solution as it includes 10 microservices while the other test cases utilise much smaller numbers.

The proposed test cases are Multi-Cloud applications with specific Non-Functional Requirements that are deployed in a Multi-Cloud topology using the presented the Extended DevOps methodology supported by ASCmI for the discovery, contracting and runtime monitoring of the cloud services:

- Test case 1 - Clinical Trial Governance Platform:
- Test case 2- Blockchain-based energy trading platform.
- Test case 3 - Cloud Service provider incidence tracking
- Test case 4 - Multi-cloud micro-services-based application (Sock Shop).

An overview of the main characteristics of each test case is included to ease the readability and under stability of the document.

The validation of ACSmI was performed along with the validation of the DECIDE project outcomes [158], being the presented ACSmI one of the outcomes of the project.

8.2 Validation objectives and methodology

The main aim of the validation process is to assess if the solution proposed (Extended DevOps methodology supported by ACSmI) is being correctly developed towards fulfilling the objectives described in Chapter 3, that is:

- Demonstrate that it is possible to characterize the multi cloud native applications concept from their SDLC and SOLC perspective and to enable the customization of the DevOps approach to the needs of the multi cloud native applications.
- Prove that it is possible to improve the discovery and selection of NFR characterized cloud services though the provision and implementation mechanisms to discover and select a combination of cloud services specific for multi-cloud aware applications and assess
- Demonstrate that it is possible to assess continuously real time verification of the cloud services non-functional properties fulfilment (Composite CSLA) including legal aspects

The evaluation of the solution has proceeded systematically in a two-fold manner. On the one hand, a quantitative approach has been taken so to examine ACSmI (particularly regarding the number and intensity of functionalities being implemented and the business-related Key Performance Indicators (KPIs) that are measured without ACSmI and with ACSmI to observe evolution).

On the other hand, a qualitative approach has been taken by gathering and analysing statistical metrics obtained with Questionnaires (on both functional, and non-functional requirements), and with feedback from the owners for each test case (for the applications of the extended DevOps methodology). These qualitative metrics yield valuable insight on the subjective perceptions of the solution stakeholders regarding its resulting status along several dimensions, by supplying concrete, specific statistical measurements on the aforementioned subjective perceptions. Thanks to this qualitative approach, it is possible to evaluate the usability, utility, completeness, desirable functionality, documentation, and other facets of the solution that are difficult to measure only with the quantitative approach.

In summary:

- Quantitative Evaluation (ACSmI):
  - Number of requirements and intensity of functionalities being tested (Test case 1, test case 2, test case 3 and test case 4).
  - Business oriented KPIs (Test case 1, test case 2, test case 3 and test case 4).
- Qualitative Evaluation (ACSmI and Extended DevOps concept):
  - Statistical metrics (Test case 1, test case 3)
  - Feedback reports from the test cases owners (Test case 1, test case 2, test case 3 and test case 4)

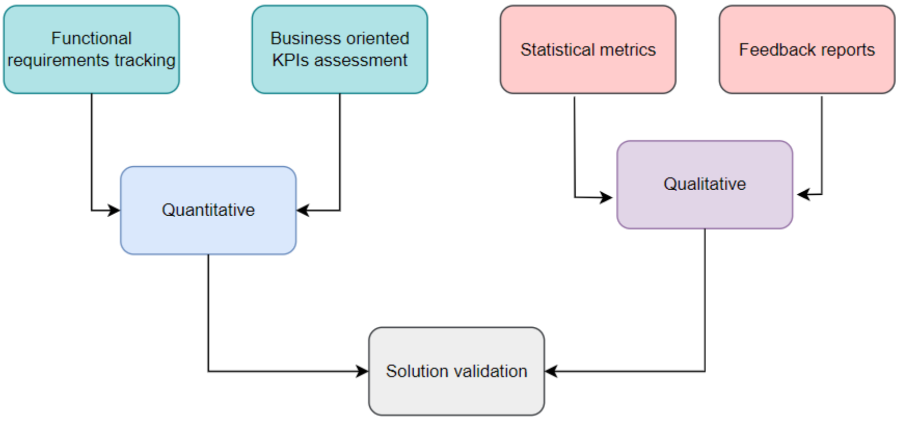
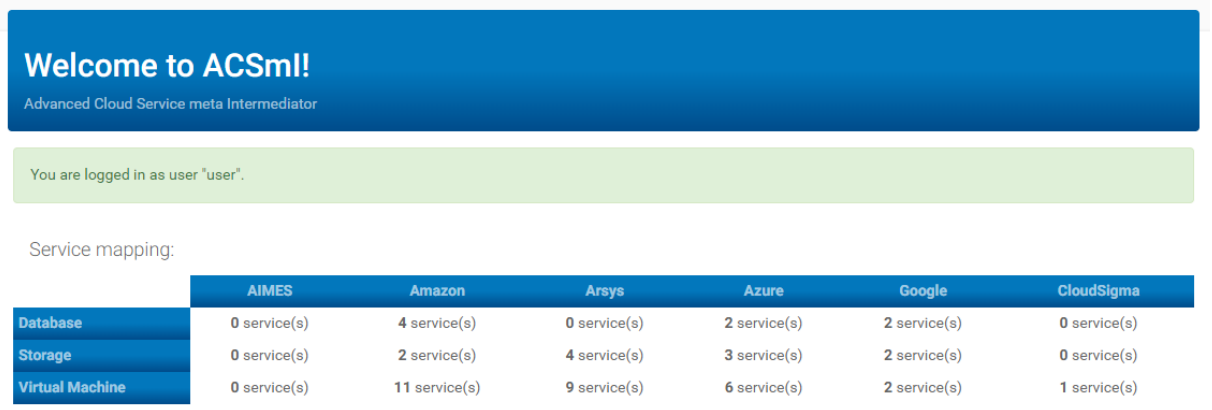


Figure 46. *Proposed solution validation approach. Source: Author's own contribution.*

8.3 Experiments set up

All the test cases used an independent installation of ACSmI. ACSmI was initially configured in the four installations with cloud services from 5 cloud service providers: Amazon (17), Arsys (13), Azure (11), Google (6) and Cloud Sigma (1). All the cloud services where virtual machines or storage services.



Welcome to ACSmI!

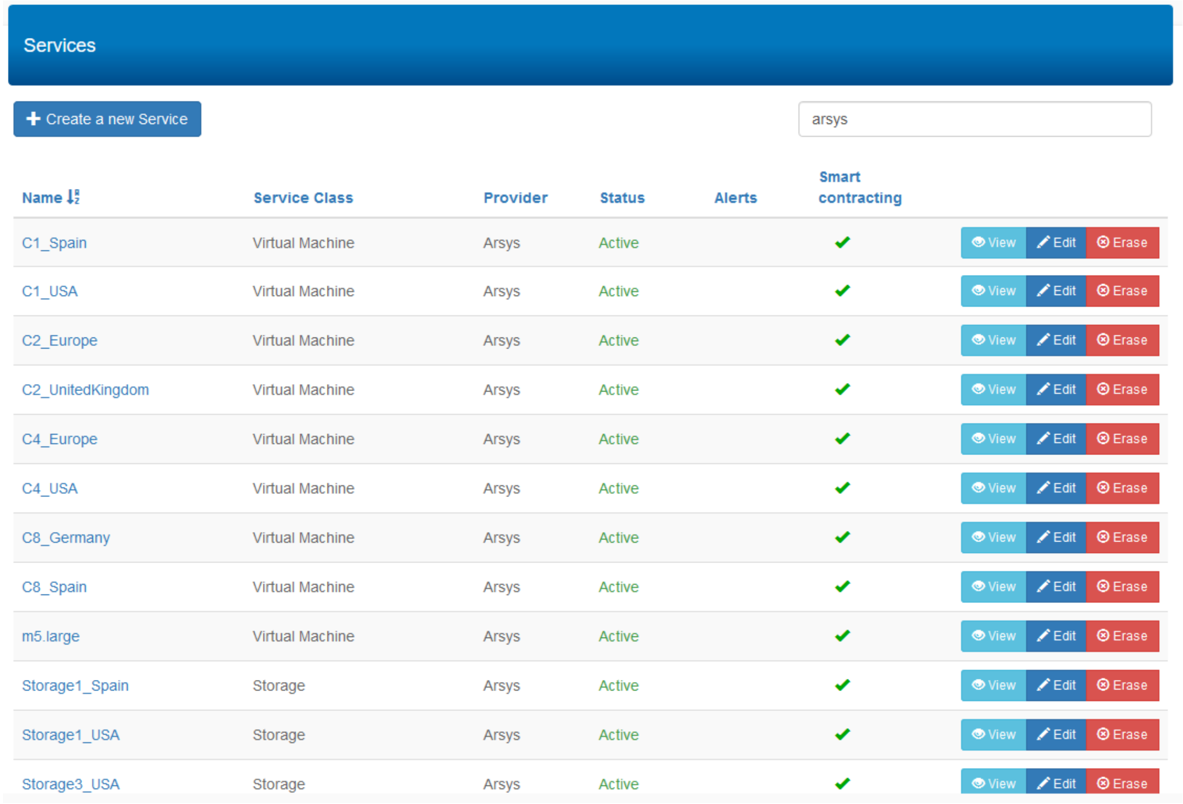
Advanced Cloud Service meta Intermediator

You are logged in as user "user".

Service mapping:

	AIMES	Amazon	Arsys	Azure	Google	CloudSigma
Database	**0** service(s)	**4** service(s)	**0** service(s)	**2** service(s)	**2** service(s)	**0** service(s)
Storage	**0** service(s)	**2** service(s)	**4** service(s)	**3** service(s)	**2** service(s)	**0** service(s)
Virtual Machine	**0** service(s)	**11** service(s)	**9** service(s)	**6** service(s)	**2** service(s)	**1** service(s)

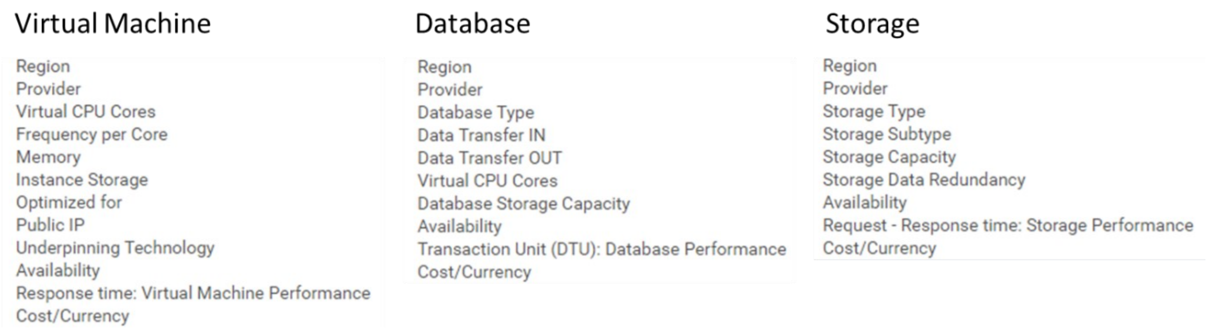
Figure 47. *Cloud services initially loaded in ACSmI. Source: Author's own contribution.*



Services

Name	Service Class	Provider	Status	Alerts	Smart contracting	
C1_Spain	Virtual Machine	Arsys	Active		✓	View Edit Erase
C1_USA	Virtual Machine	Arsys	Active		✓	View Edit Erase
C2_Europe	Virtual Machine	Arsys	Active		✓	View Edit Erase
C2_UnitedKingdom	Virtual Machine	Arsys	Active		✓	View Edit Erase
C4_Europe	Virtual Machine	Arsys	Active		✓	View Edit Erase
C4_USA	Virtual Machine	Arsys	Active		✓	View Edit Erase
C8_Germany	Virtual Machine	Arsys	Active		✓	View Edit Erase
C8_Spain	Virtual Machine	Arsys	Active		✓	View Edit Erase
m5.large	Virtual Machine	Arsys	Active		✓	View Edit Erase
Storage1_Spain	Storage	Arsys	Active		✓	View Edit Erase
Storage1_USA	Storage	Arsys	Active		✓	View Edit Erase
Storage3_USA	Storage	Arsys	Active		✓	View Edit Erase

Figure 48. *A snapshot of the types of services loaded in ACSmI for evaluation purposes. Source: Author's own contribution.*



Virtual Machine

- Region
- Provider
- Virtual CPU Cores
- Frequency per Core
- Memory
- Instance Storage
- Optimized for
- Public IP
- Underpinning Technology
- Availability
- Response time: Virtual Machine Performance
- Cost/Currency

Database

- Region
- Provider
- Database Type
- Data Transfer IN
- Data Transfer OUT
- Virtual CPU Cores
- Database Storage Capacity
- Availability
- Transaction Unit (DTU): Database Performance
- Cost/Currency

Storage

- Region
- Provider
- Storage Type
- Storage Subtype
- Storage Capacity
- Storage Data Redundancy
- Availability
- Request - Response time: Storage Performance
- Cost/Currency

Figure 49. *Services types and information included per service type in ACSmI. Source: Author's own contribution.*

Each of the test cases used its own instantiation of ACSmI to discover, contract and monitor the cloud services needed to deploy their applications with specific non-functional needs:

- Test case 1 - Clinical Trial Governance Platform: The final release of the ACSmI was evaluated using the StreamLine V2 [159]. A tool for academic health science researchers to develop and manage clinical trials. Sensitive Personal Patient Identifiable Information is stored within this tool, and the data belongs to people who live in different countries across the world. One of the key challenges faced in this test cases is accomplishing by national and pan-European legislation about the location of such data. The use of containerized microservices, lends itself the ability to handle a variety of data protection laws, not least by enabling data to reside in its country of origin. Adopting the Multi-Cloud approach to address the legal ramifications of hosting sensitive data however comes with its own series of challenges that has been minimized though the usage of ACSmI. For the validation exercise an architecture of 5 microservices has been used.
- Test case 2- Blockchain-based energy trading platform. The platform brings together energy producers and consumers, allowing the former to make energy offers and the latter to purchase power under the terms of the offer. The energy exchanges are handled by means of smart contracts, so, after both parties have agreed, the purchase will take place automatically. For the validation an instance of 3 microservices of the original system has been utilized.
- Test case 3 - Cloud Service provider incidence tracking. An internal application to coordinate and monitor activities performed in the data centres of the Cloud Service provider. For the validation exercise the architecture tested was formed of 3 microservices.
- Test case 4 - Multi-cloud microservices-based application (Sock Shop): The Sock Shop app is designed to provide as many microservices as possible. All services in the Sock Shop communicate using REST. The Sock Shop app is being built using Spring Boot4, Go kit5 and Node.js6 and is packaged in Docker7 containers. The Sock Shop has served as an additional "use case" application to guarantee the scalability of the solution as it includes 9 microservices while the other use cases include 2 or 3. For validation purposes the application has been deployed into 1 Cloud Service provider (but different Cloud Services of such provider) and two different Cloud Services providers over multiple services.

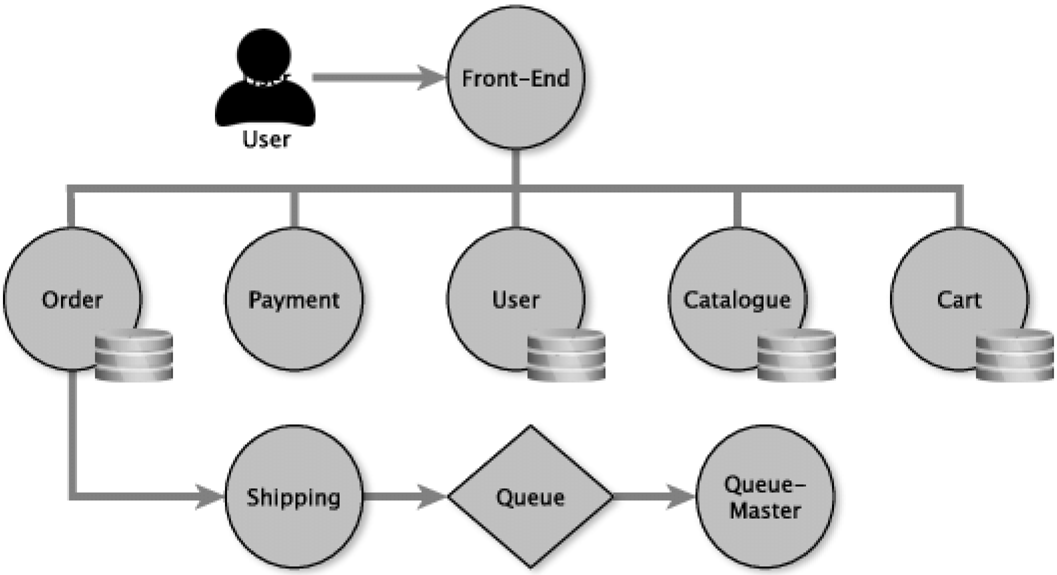


Figure 50. *Architecture of the application deployed in Test case 3 with the support of ACSmI .Source* [157].

With respect to the Extended DevOps concept all the DevOps members of the different Test cases were skilled in DevOps methodologies. The team members followed the proposed Extended DevOps concept to develop and operate their respective applications.

8.4 Quantitative evaluation

8.4.1 Objective

The objective of this experiment was twofold:

- Evaluate the whether the requirements defined for the ACSmI solution have been met and to which extent
- Evaluation of the improvement brought by ACSmI with respect to business related KPIs

8.4.2 Requirement's evaluation tracking

The complete list of functional requirements defined in ACSmI was developed at design time as explained in chapter 7. The description of some requirements has been adapted since then based on the updated knowledge gained during the development of the solution while keeping the overall objective.

The fulfilment of the functionalities described by these requirements were continuously evaluated during the three development iterations of the solution by the developers and in agreement with participants from the 3 industrial test cases owners. In the next table the final status of the requirements is presented:

Table 19. *Requirement's coverage of ACSmI discovery final prototype.*

Req. ID	Status	Requirement coverage by the prototype
DIS01	Satisfied	This prototype allows the endorsement of three different types of service classes: • Virtual machine • DB • Storage The prototype requests the specific information depending on the cloud service type.
DIS02	Satisfied	ACSmI provides a REST enabling the programmatic access to the intelligent discovery functionality. This API allows to look for FR and NFR like availability or certification schemes if these are the requirements of the developer. ACSmI also provides an UI to allow the user to search himself.
DIS03	Satisfied	In this prototype, the discovery functionality has been fulfilled considering the comparison of the units of the terms, or the nature of the term, (i.e., 500 MB= 0,5 GB; or if required availability is 95%, all the services with the availability major than 95% are presented).
DIS04	Satisfied	The functionality has been fulfilled.
DIS05	Satisfied	In this release, the discovered services are shown with a percentage of the fulfilment and a list of the NFRs that are fulfilled.
DIS06	Satisfied	ACSmI takes care the management of the CSPs users.
DIS07	Satisfied	ACSmI registry records information about the SLA violation: Time and type of the violation and allows to upload the required information to be analysed by a legal expert to assign the legal level.
DIS08	Satisfied	ACSmI allows to perform different tasks depending on the role that the user has assigned. At this moment, there are 4 roles identified: Admin, CSP, Developer/operator/user and Legal expert.
DIS09	Satisfied	ACSmI allows to delete resources.

Table 20. *Requirement's coverage of ACSmI monitoring final prototype.*

Req. ID	Status	Requirement coverage by the prototype
MON01	Satisfied	Fully covered.
MON02	Satisfied	Fully covered.
MON03	Satisfied	Fully covered: Availability, Cost, location, performance.
MON04	Satisfied	ACSmI is capable to check the type of the services and the parameters to be monitored.
MON05		Satisfied. ACSmI accesses the JSON application description to gather this information.
MON06	Satisfied	This prototype registers an incidence in the ACSmI registry to enhance the discovery of services without incidences.

Req. ID	Status	Requirement coverage by the prototype
		It also uses an external API (violation handler [160]) to send an email to the operator of the cloud services which SLA is being violated.
MON 07	Satisfied	This is provided by the ACSmI UI.
MON08	Satisfied	This subcomponent takes the information from three different places depending on which NFR is monitored: • InfluxDB data (performance and availability) base and assesses it to check if the metrics obtained fulfil or not the SLA of the services. This subcomponent uses the MCSLA core library to support this assessment. • MaxMind GeoIP2 Java API with the free GeoLite2 database to monitor location. This library checks where is really located the VM and ACSmI monitoring checks if this information is equal to the one in the registry. • ACSmI billing that provides the information of the Cost of the cloud service.
MON09	Satisfied	Once a service cloud SLA violation is detected, ACSmI monitoring informs to ACSmI registry, using the interface provided by ACSmI discovery, to record the information regarding this violation. The information is: CSId, Timestamp, the NFR violated. ACSmI monitoring also informs to the operator. The information provided is the type of violation, the NFR violated, the application that is deployed in the cloud service and the time when the assessment identifies the violation.
MON10	Satisfied	This prototype defines and implements push monitoring using the Telegraf plugin. For VM the plugin to be used is "ping" to measure availability and "mem", "disc" and CPU plugins to measure performance. Although ACSmI monitoring continues using the push method, to allow the measurement the performance of the VM, it is required to deploy a Telegraf configuration file in each VM.

Table 21. *Requirement's coverage of ACSmI legal final prototype.*

Req. ID	Status	Requirement coverage by the prototype
LEG01	Satisfied	This ACSmI Discovery prototype allows when endorsing a service, on one hand to answer a set of questions (called simple controls) by the CSP and on the other hand allow to the CSPs to upload the following documents that will allow the legal expert to assess the legal compliance. These documents are: • The service contract applicable to the service • The SLA applicable to the service • The Data processing agreement governing the service Other functionality implemented is the questionnaire to be answered by the legal expert based on the information available on the uploaded documents, and finally ACSmI

		discovery implements the logic to derive the legal level following the matrix presented in section 6.

Business and security modules were implemented based on existing solutions as explained in section 7.3.3. These existing solutions were selected in order to fulfil all the requirements identified. Therefore, all the requirements describe in section 7.3.2 of "Security Management" and "Business modelling implementation" were satisfied by ACSmI.

Although in the former table only the final version of the requirements evaluation is shown, this process has been executed in a continuous based approach. Three different iterations corresponding to the three versions of the solution prototype have been executed. The complete process followed is shown in Figure 51.

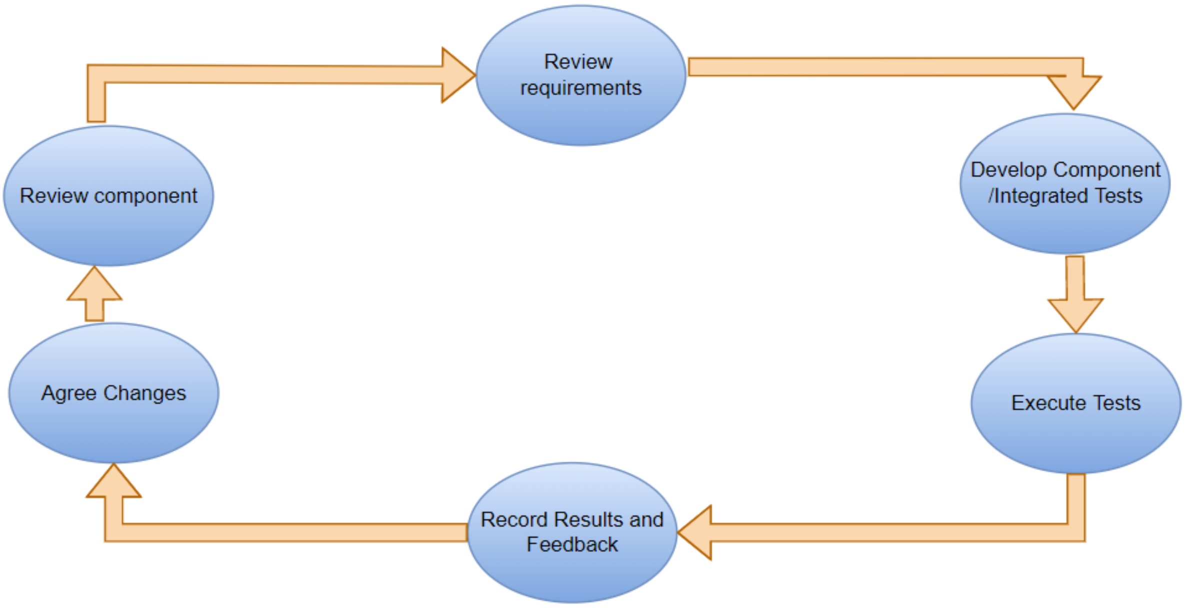

Review requirements
Develop Component /Integrated Tests
Execute Tests
Record Results and Feedback
Agree Changes
Review component

Figure 51. *Continuous requirements evaluation process. Source: Author's own contribution.*

8.4.3 Business oriented evaluation

The business-oriented evaluation has been performed using a developer-centric approach testing. Therefore, the evaluation has been performed following the process that a developer would follow to manage the cloud services in a usual application development process using ACSmI. As part of the developer-centric testing we have included a comparison of the needed tasks and effort (in terms of time consumed) that are needed to support the multi-cloud application with and without the support of ACSmI.

In the next table all the activities that a developer/operator needs to perform are included. In the following section the effort needed to perform these activities with and without ACSmI is detailed:

Table 22. *Activities to be considered in the management of the Cloud services as part of the multi-cloud application lifecycle.*

DevOps phase	Activities to be performed by the DevOps team	ACSmI supporting component
Cloud services discovery	- Discovery of the appropriate cloud services - Endorsement of services (Cloud Service Provider)	ACSmI discovery ACSmI monitoring
Cloud services contracting	- For each of the selected Cloud provider /cloud service: ○ Create a user ○ Create the contract ○ Manage the contract	ACSmI contracting
Monitoring	- Create a (virtual) machine where the monitoring agents can be deployed and executed - Select/create/incorporate agents[15] that can monitor low level metrics for the cloud resources and/or for the microservices. - Gather runtime low level metrics and store them - Process the metrics so that they can be traduced into SLO metrics. - Continuously compare the metrics at microservice level (component level) and at cloud resource level with the SLO at application level and at infrastructural level so that the SLA violations are detected. - Register the violations at cloud resource level so that the related cloud resources are not chosen for the next deployment configuration.	ACSmI monitoring
Billing	- Continuously get the consumed resources and the usage records for each of the cloud service contracted. - Manage the invoices.	ACSmI billing
DevOps lifecycle	- Manage the whole lifecycle of a multi-cloud application	Extended DevOps concept

The testing of the different ACSmI components has been done taking the role of a DevOps team member and trying to accomplish the reported activities (Table 22) for the fourth test cases. For that, an estimation of the effort needed to manually perform the activities corresponding to different processes under the Cloud Service Lifecycle have been reported by the test case owners. Then, the effort needed for the same activity has been recorded using ACSmI. The effort is translated to costs based on the characteristics of each company and project. An example of this process for the Intelligent Discovery is included in Table 23.

[15] In the case of cloud resource monitoring, monitoring functions offered by the cloud providers can be used. This implies a limitation on the metrics/expressions to be monitored as well as the trustworthiness of those data as usually the monitoring capabilities are offered as a black box by the cloud providers

Table 23. *Resources needed to perform the activities included in the Intelligent Service Discovery process, under the Cloud Service Lifecycle.*

Cloud Service Intelligent Discovery process	Broken down activities	Effort needed w/o ACSmI PH: Person/hour PM: Person/month	Effort needed with ACSmI	Saved effort
	Study the existing available cloud services from different providers	2PH	0.5PH	2PH
	Analyse the characteristics of each Cloud Service (SLAs, costs, supported technologies, third party components dependencies, etc.)	8PH	0PH	8PH
	Be aware of run-time information of specific cloud services with respect to SLA violations so that these cloud services can be discarded when selecting the most appropriate ones.	0,5 PM	0PH	60PH

The compilation of obtained results is reported and discussed in the results section. Each of the 4 test cases proposed was set up and executed to assess the benefits reported from the usage of ACSmI with different configurations.

8.4.4 Results

This part of the evaluation has assessed whether the requirements defined for the system have been met. This part has evaluated also, that the proposed system is relevant and incorporates savings in terms of the benefits that it brings to the companies using it.

As a result, 100% of the functional requirements elicited, which had been derived from the analysed challenges have been successfully implemented in the solution.

Table 24. *Overall fulfilment of the functional requirements of ACSmI.*

ACSmI component	Requirement Id	ACSmI version	Requirement status	Priority
ACSmI Discovery	DIS01	Final prototype	Finished	High
	DIS02	Final prototype	Finished	High
	DIS03	Final prototype	Finished	High
	DIS04	Final prototype	Finished	High
	DIS05	Final prototype	Finished	High
	DIS06	Final prototype	Finished	High
	DIS07	Final prototype	Finished	High
	DIS08	Final prototype	Finished	High
	DIS09	Final prototype	Finished	Medium
ACSmI Contracting	BUS02	Final prototype	Finished	High
	BUS07	Final prototype	Finished	High
	BUS08	Final prototype	Finished	High

	BUS09	Final prototype	Finished	High
ACSml Monitoring	MON01	Final prototype	Finished	Low
	MON02	Final prototype	Finished	Low
	MON03	Final prototype	Finished	High
	MON04	Final prototype	Finished	High
	MON05	Final prototype	Finished	High
	MON06	Final prototype	Finished	High
	MON07	Final prototype	Finished	Low
	MON08	Final prototype	Finished	High
	MON09	Final prototype	Finished	High
	MON10	Final prototype	Finished	Low
ACSml Billing	BUS01	Final prototype	Finished	High
	BUS03	Final prototype	Finished	High
	BUS04	Final prototype	Finished	High
	BUS05	Final prototype	Finished	High
	BUS06	Final prototype	Finished	High
ACSml Legal	LEG01	Final prototype	Finished	High
All components	SEC01	Final prototype	Finished	High
	SEC02	Final prototype	Finished	High
	SEC03	Final prototype	Finished	High
	SEC04	Final prototype	Finished	High
	SEC05	Final prototype	Finished	High
	SEC06	Final prototype	Finished	High
	SEC07	Final prototype	Finished	High

On the other hand, the result on the business-oriented evaluation is also very promising. An overview of the results obtained is depicted in Figure 52 where the effort needed to perform all the activities from each of the processes (Cloud Service Discovery, Cloud Service Contracting and Cloud Service Monitoring) is reported in person/hour. In graphic a) the same application is deployed in the same number of CSPs both manually and using ACSmI. In graphic b) (Figure 53) the same application is deployed firstly into one service provider and secondly on two services providers. This exercise allowed us to compare on the one hand the advantages brought by ACSmI when the complexity of the application increases (in terms of microservices number) and on the other hand the benefit provided by ACSmI when, with the same application complexity (the same number of microservices) the number of CSPs increases. The unit to measure the effort is the person hour so that the results from the different companies can be compared independent from their internal structural cost based on their size or industrial sector.

In Figure 52 the colour code is as follows:

- Blue results correspond to the CSP Incidence Tracking application deployed in one CSP (Arsys): Dark blue is for the effort needed to perform each task manually and light blue is for the effort needed to perform each task using ACSmI.
- Purple results correspond to the Clinical Trial Governance Platform deployed in two CSPs (Aimes and Amazon): Light yellow is for the effort needed to perform each task manually and dark yellow is for the effort needed to perform each task using ACSmI.

In these Figure a) two applications are compared, one with two microservices deployed in a single CSP and another one with 4 microservices deployed in two different CSPs. In both cases the developer saved effort when using ACSmI in the Cloud Service Contracting and monitoring processes. The developers reported that the effort saved in the monitoring process (up to the 80 %) is mainly due to the proactive continuous monitoring which allowed them to assure the fulfilment of the CSLA contracted in terms of accomplishment of the selected NFRs (availability, location and performance). When the number of CPs increases the savings apply to the discovery phase too. The developers also reported that analysing the needs of each microservice and each CSP is a time-consuming task, even more when the offer is disaggregated and described in different terms.

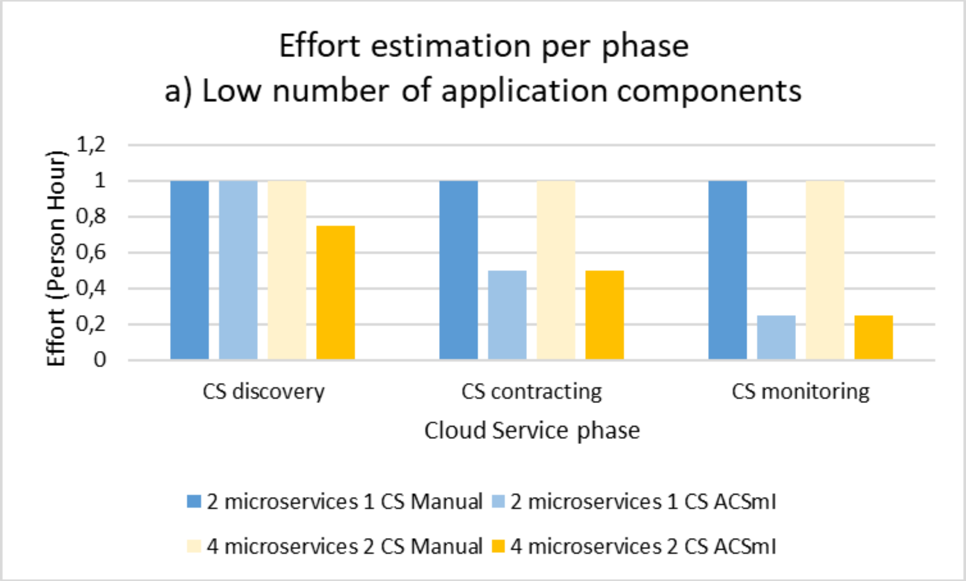


Figure 52. *Effort estimation per phase in the same and in different number of Cloud resources. Source: Author's own contribution.*

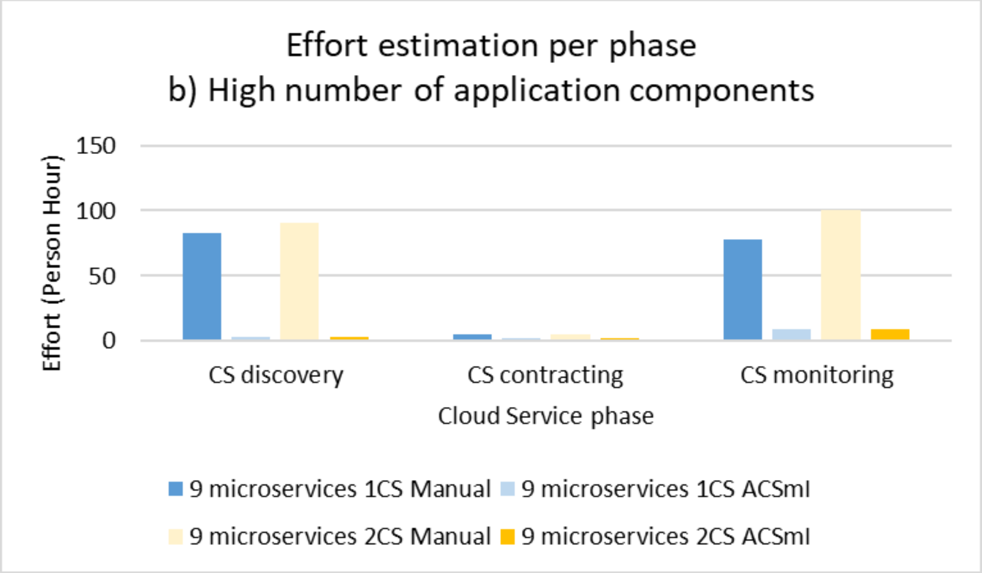


Figure 53. *Effort estimation per phase in the same and in different number of Cloud resources. Source: Author's own contribution.*

In Figure 53 the colour code is as follows:

- Blue results correspond to the Sock Shop deployed in one CSP (Amazon). Dark blue for the effort needed to perform each task manually and light blue depicts the effort needed to perform each task using ACSmI.
- Purple results correspond to the Sock shop deployed in two CSPs (Amazon and Azure). Light yellow for the effort needed to perform each task manually and dark yellow depicts the effort needed to perform each task using ACSmI.

In this case, the savings achieved when using ACSmI are greater. In the three phases the effort saved is relevant, more than 90%. Now, as 9 microservices are composing the application, the analysis of which Cloud Service fits better (Intelligent Service Discovery) requires much more effort than in the previous cases. Subsequently the benefits reported when using ACSmI for this purpose are also greater.

From this, it can be deduced that when the number of CSPs grows, the benefits of using ACSmI increase exponentially.

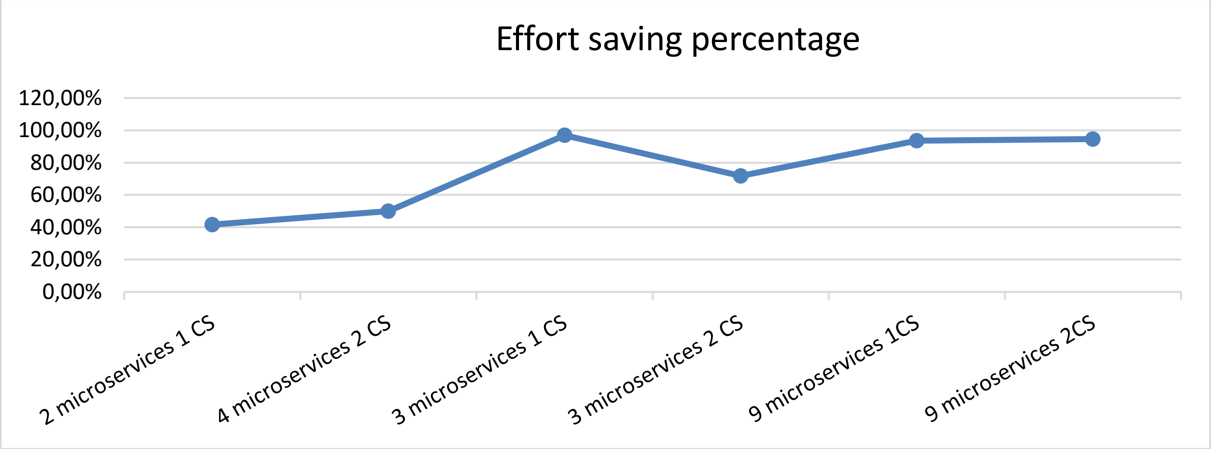


Figure 54. Effort saving percentage. Source: Author's own contribution.

In Figure 54 the different effort saving percentages are shown for the different experiments. The savings produced by ACSmI increase both with the application complexity (in terms of microservices number) and the number of cloud services to be used in each case. Of course, some deviations may exist as the exercises have been made by different DevOps teams.

8.5 Qualitative evaluation

8.5.1 Objective

One objective of the qualitative evaluation of the solution is to gather statistical metrics and feedback through questionnaires from the test cases owners ("users") of the solution. These qualitative metrics will generate valuable information on the subjective perception of the solution stakeholders regarding its resulting status along several dimensions, by supplying concrete, specific statistical measurements on the aforementioned subjective perceptions. Through this qualitative approach, it would be possible to evaluate the usability, utility, completeness, desirable functionality, documentation and other facets of the solution that are difficult to measure only with the quantitative approach.

To perform this qualitative evaluation, we have selected a set of dimensions to be assessed:

- Availability: Availability[16] as the ratio of time a system is functional compared to the total time. It can be calculated as a direct proportion (for example, 9/10 or 0.9) or as a percentage (for example, 90%).
- Efficiency: Efficiency[17] measures the extent to which input is well used for an intended task.
- Usability: Usability[18] is the ease of use and learnability of a human-made object (i.e. a software application).
- Flexibility: It can be defined as the ability of software to change easily from user and system requirements.
- Interoperability: Interoperability[19] is a property of a product to work with other products or systems, present or future, without any restricted access or implementation.
- Reusability: The ability[20] to reuse relies in an essential way on the ability to build larger things from smaller parts and being able to identify commonalities among those parts.

Another objective of the qualitative evaluation is to validate the Extended DevOps concept by the DevOps teams of the test cases. The validation of the Extended DevOps concept included the validation of the new activities proposed:

8.5.2 Extended DevOps concept qualitative evaluation

The qualitative evaluation of the DevOps concept included the incorporation of the Extended DevOps activities in the Development and Operation of the Test case 1 and Test case 3 by their DevOps teams. These new activities were introduced:

- Design phase:
  - NFRs specification.
  - Multi-cloud architectural patterns definition.
- Pre-deployment phase:
  - Application profiling: Profiling of application components.
  - Cloud services discovery.
  - Deployment optimization.
  - MCSLA definition.
- Deployment preparation phase:
  - Cloud services contracting.
  - Develop connectors to the CSPs.
- Continuous monitoring phase:
  - MCSLA assessment.
- Self-adaptation phase:
  - Application adaptation and redeployment.

Some of these activities were supported by tools i.e., ACSmI for service discovery, while others were performed manually.

[16] https://whatis.techtarget.com/definition/Reliability-Availability-and-Serviceability-RAS

[17] https://en.wikipedia.org/wiki/Efficiency

[18] https://en.wikipedia.org/wiki/Usability

[19] http://interoperability-definition.info/en/

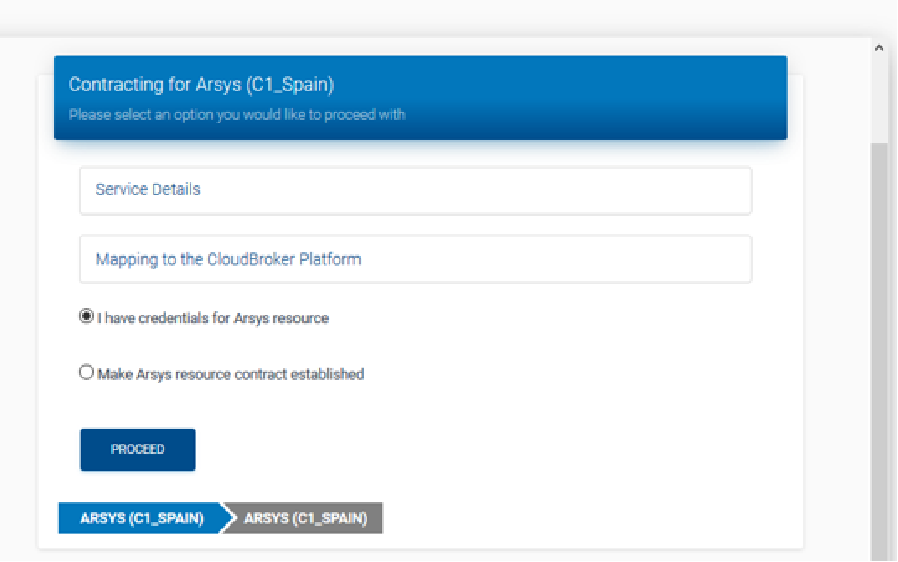
[20] https://en.wikipedia.org/wiki/Reusability

8.5.3 Early user evaluation

ACSmI has been assessed with respect to the qualitative evaluation by two of the three industrial test cases owners. The process followed by the users experimenting with the prototype (Beta tests) has served to acquire the metrics related to the dimensions identified and to acquire feedback, in form of reports where we have acquired the user's perspective of ACSmI. The aim of the evaluation was to identify how the ACSmI would be used and how could it be improved.

In this case, each of the test cases owners has used each own instance of ACSmI (as described in section 7.3) and has assessed one/some of the components of ACSmI depending on their interests in the following way:

- Blockchain-based energy trading platform: ACSmI contracting has been used to contract a set of cloud services (in this case the service "Arsys Cloud builder") selected for the deployment of the application. The test is done assuming using the Arsys credentials from the test case owner.



Contracting for Arsys (C1_Spain)

Please select an option you would like to proceed with

Service Details

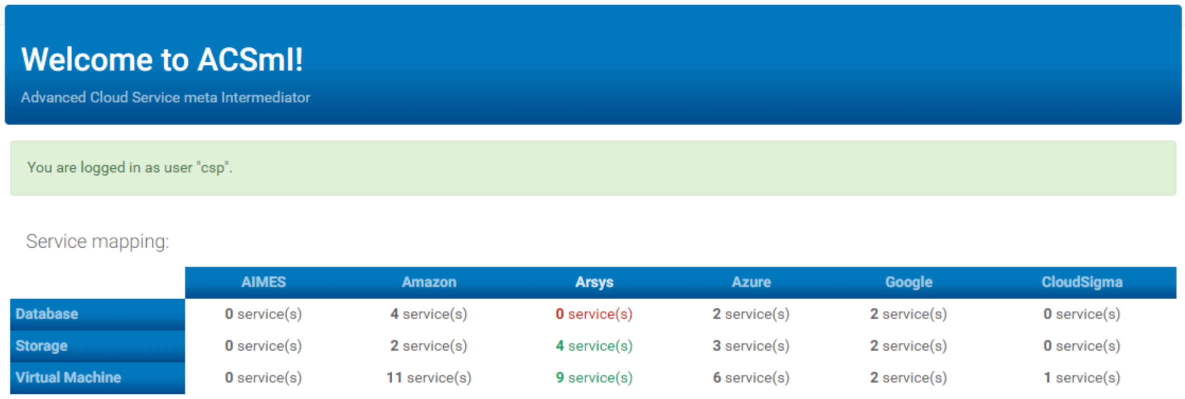
Mapping to the CloudBroker Platform

● I have credentials for Arsys resource

○ Make Arsys resource contract established

Figure 55. *ACSmI Contracting test in energy trading platform. Summary of CSPs.* Source: Author's own contribution.

- Cloud Service provider incidence tracking. In this case ACSmI discovery and contracting were evaluated. ACSmI discovery was tested under two roles perspective, a CSP role and a cloud services user role.



Welcome to ACSmI!

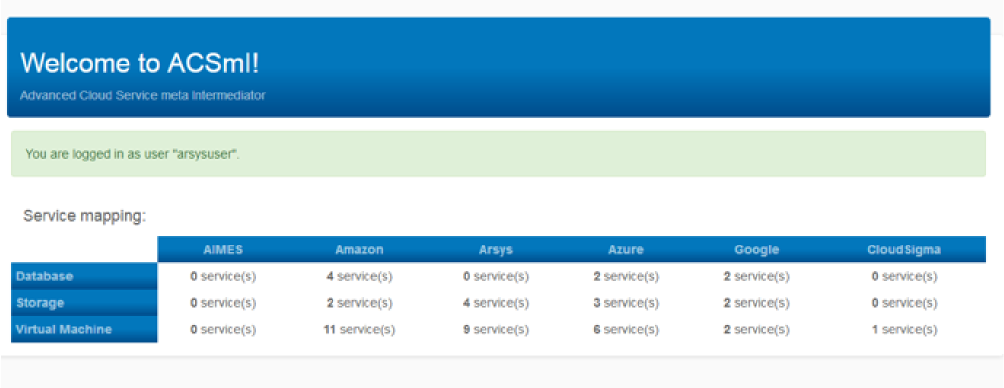
Advanced Cloud Service meta Intermediator

You are logged in as user "csp".

Service mapping:

	AIMES	Amazon	Arsys	Azure	Google	CloudSigma
Database	0 service(s)	4 service(s)	0 service(s)	2 service(s)	2 service(s)	0 service(s)
Storage	0 service(s)	2 service(s)	4 service(s)	3 service(s)	2 service(s)	0 service(s)
Virtual Machine	0 service(s)	11 service(s)	9 service(s)	6 service(s)	2 service(s)	1 service(s)

Figure 56. *ACSmI CSPs view* Source: Author's own contribution.



Welcome to ACSmI!

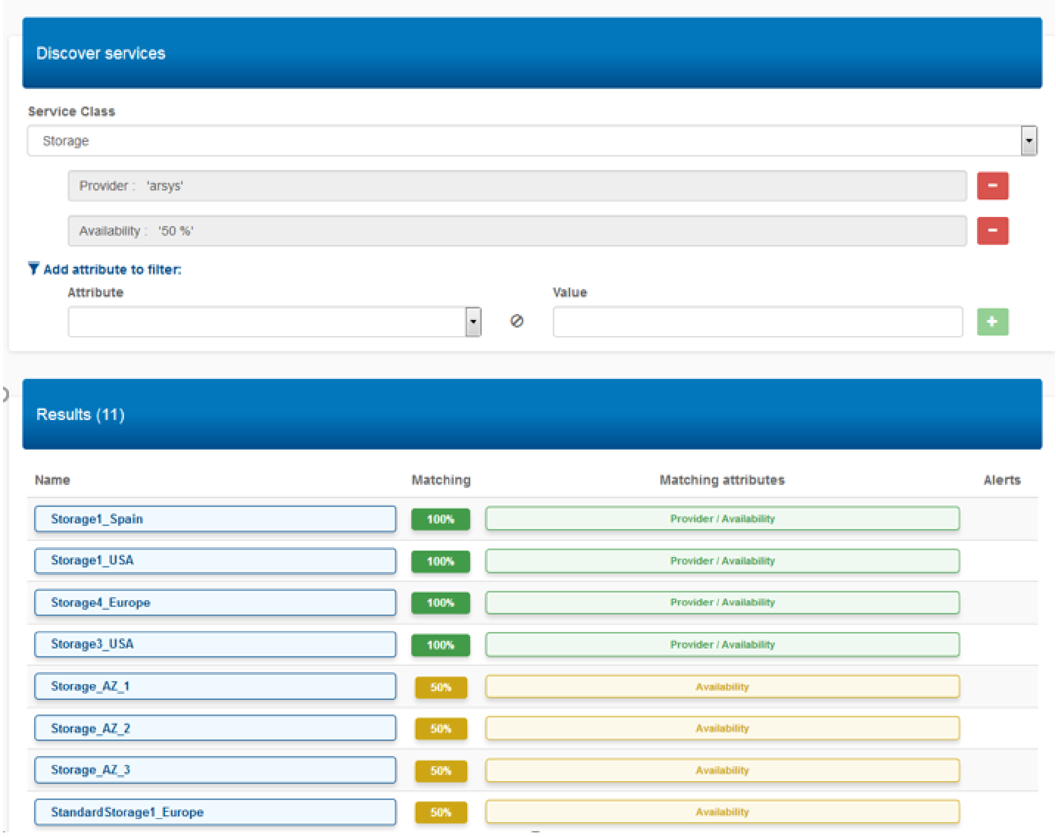
Advanced Cloud Service meta Intermediator

You are logged in as user "arsysuser".

Service mapping:

	AIMES	Amazon	Arsys	Azure	Google	CloudSigma
Database	0 service(s)	4 service(s)	0 service(s)	2 service(s)	2 service(s)	0 service(s)
Storage	0 service(s)	2 service(s)	4 service(s)	3 service(s)	2 service(s)	0 service(s)
Virtual Machine	0 service(s)	11 service(s)	9 service(s)	6 service(s)	2 service(s)	1 service(s)

Figure 57. *Welcome screen as User role.* Source: Author's own contribution.



Discover services

Service Class

Storage

Provider : 'arsys'

Availability : '50 %'

Add attribute to filter:

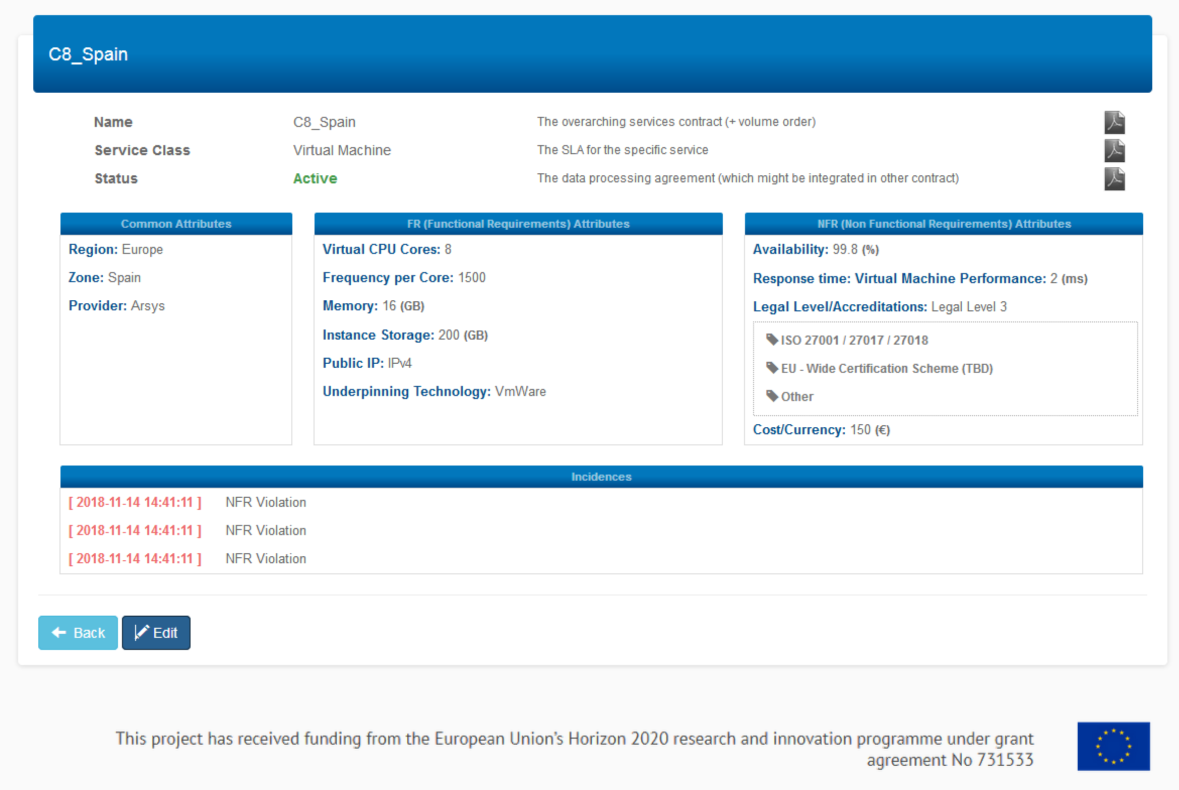
Attribute

Value

Results (11)

Name	Matching	Matching attributes	Alerts
Storage1_Spain	100%	Provider / Availability	
Storage1_USA	100%	Provider / Availability	
Storage4_Europe	100%	Provider / Availability	
Storage3_USA	100%	Provider / Availability	
Storage_AZ_1	50%	Availability	
Storage_AZ_2	50%	Availability	
Storage_AZ_3	50%	Availability	
StandardStorage1_Europe	50%	Availability	

Figure 58. *NFR matching.* Source: Author's own contribution.



C8_Spain

Name	C8_Spain	The overarching services contract (+ volume order)	
Service Class	Virtual Machine	The SLA for the specific service	
Status	**Active**	The data processing agreement (which might be integrated in other contract)	

Common Attributes

Region: Europe

Zone: Spain

Provider: Arsys

FR (Functional Requirements) Attributes

Virtual CPU Cores: 8

Frequency per Core: 1500

Memory: 16 (GB)

Instance Storage: 200 (GB)

Public IP: IPv4

Underpinning Technology: VmWare

NFR (Non Functional Requirements) Attributes

Availability: 99.8 (%)

Response time: Virtual Machine Performance: 2 (ms)

Legal Level/Accreditations: Legal Level 3

- ISO 27001 / 27017 / 27018
- EU - Wide Certification Scheme (TBD)
- Other

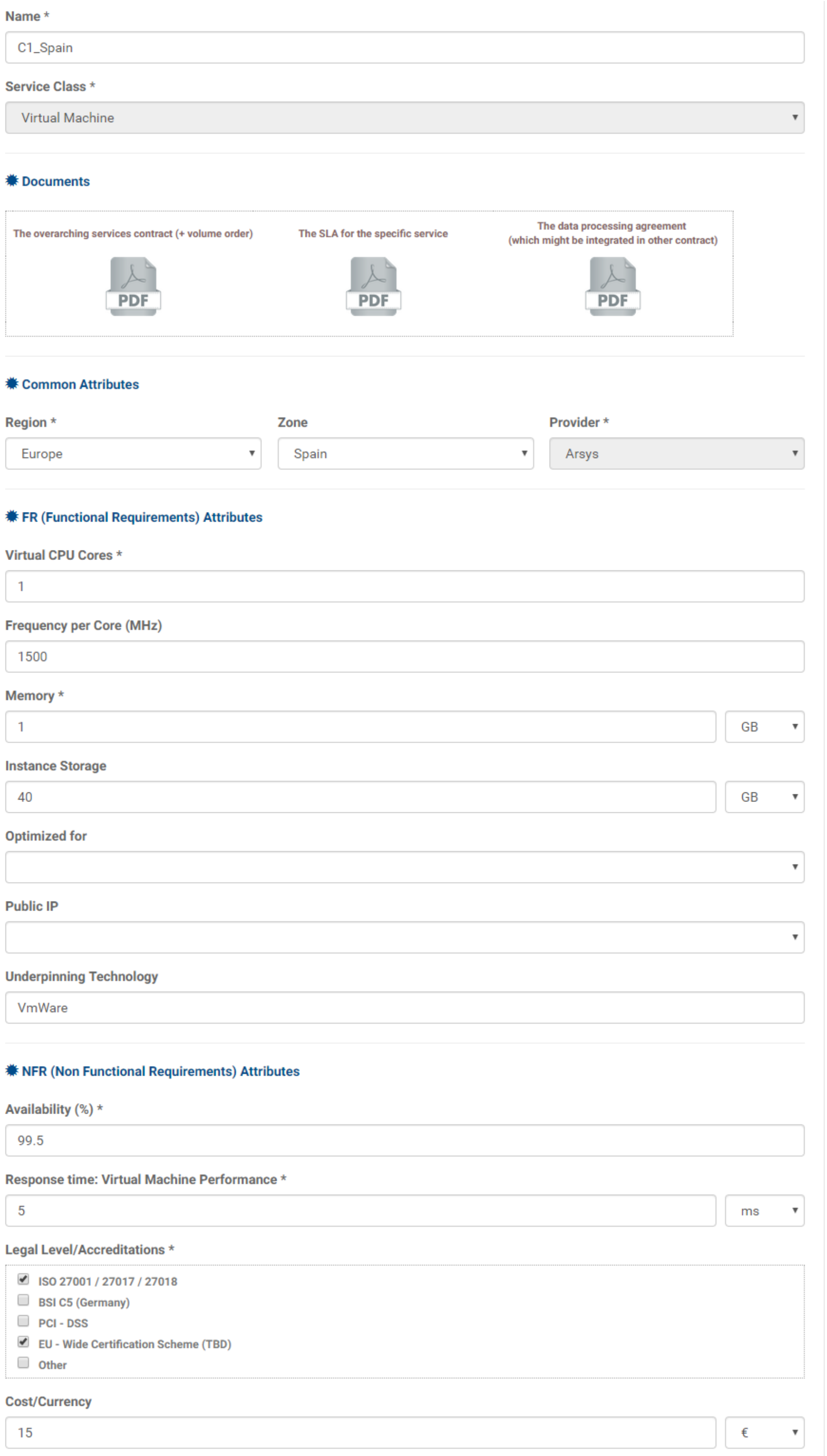
Cost/Currency: 150 (€)

Incidences

[2018-11-14 14:41:11]	NFR Violation
[2018-11-14 14:41:11]	NFR Violation
[2018-11-14 14:41:11]	NFR Violation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731533

Figure 59. *Detail of a service from the "View" screen.* Source: Author's own contribution.



Name *

C1_Spain

Service Class *

Virtual Machine ▾

Documents

The overarching services contract (+ volume order)	The SLA for the specific service	The data processing agreement (which might be integrated in other contract)
PDF	PDF	PDF

Common Attributes

Region *	Zone	Provider *
Europe ▾	Spain ▾	Arsys ▾

FR (Functional Requirements) Attributes

Virtual CPU Cores *

1

Frequency per Core (MHz)

1500

Memory *

1 GB ▾

Instance Storage

40 GB ▾

Optimized for

▾

Public IP

▾

Underpinning Technology

VmWare

NFR (Non Functional Requirements) Attributes

Availability (%) *

99.5

Response time: Virtual Machine Performance *

5 ms ▾

Legal Level/Accreditations *

- ☑ ISO 27001 / 27017 / 27018
- □ BSI C5 (Germany)
- □ PCI - DSS
- ☑ EU - Wide Certification Scheme (TBD)
- □ Other

Cost/Currency

15 € ▾

Figure 60. *ACSmI service endorsement main page.* Source: Author's own contribution.

The early user evaluation has provided two main outcomes:

1. A set of user feedback reports with the impressions and the suggestions for improvements. An overview of the key comments from these user experience reports is provided in the results section.
2. Results about the basement of the selected dimensions: Availability, Efficiency, Usability, Flexibility, Interoperability and Reusability.

8.5.4 Statistical metrics gathering

The proposed qualitative evaluation process encompasses the gathering of metrics that provide insights on the viability of the solution. The evaluation strategy for each dimension and the KPIs against which each dimension is assessed are detailed below.

Availability

In simple terms Availability is the proportion of time a system is in a working condition[21]. Historically, availability is one of the most relevant SLA guarantees offered by cloud providers for their services, but each provider gives its own detailed definition of availability as well as their own calculation strategy, i.e. differences on the total time over which to measure availability, exact definition of "unavailable", or who and how is entitled to measure it.

In ACSmI the Availability dimension is considered under two different perspectives:

- A1: Availability of the system proposed, ACSmI, in each of the installations as described in section 7.3 Experiments set up.
- A2: Availability of the application (test case 1, test case 2 and test case 3) obtained through the usage of ACSmI. In this case applications' MCSLA is impacted by the resources CSLAs which are monitored by ACSmI. Through this metric we can assess if the application is being available and to which extent when using ACSmI.

Table 25. *Template for availability assessments.*

Dimension	Evaluation method	Acceptance criteria
Availability	A1: % of time that the service is running A2: Degree of fulfilment of the application MCSLA related to availability	A1: 99% A2: 99.99%

Efficiency

In general, efficiency is described as the ability to do something or produce something without wasting materials, time, or energy. Usually, efficiency and effective are used indistinctly.

[21] https://en.wikipedia.org/wiki/Availability

Peter Drucker [161] defined the difference between being efficient and being effective:

- Efficiency is the capacity to do things right.
- Effectiveness is the capacity to do the right thing.

Following this approach in the software context, a system effectiveness can be defined as doing the objective effectively, that is, correctly. The efficiency of a system can be defined as using the available resources optimally. Therefore, the evaluation of metrics related to the efficiency of the system can be applied to the software itself or to the task that a software component performs. Similarly, the optimal resources usage of the resources can be both referred to the resources used by the software component itself (i.e. memory, CPU, time, files, connections, databases etc.) or the resources used to perform a task (usually measured as time).

In the context of this thesis, ACSmI's efficiency will be evaluated calculating the saving in resources when using it to discover the cloud services needed to deploy a multi-cloud native application.

Table 26. *Template for efficiency assessment.*

Dimension	Evaluation method	Acceptance criteria
Efficiency	Resources usage optimization (usually time reduction percentage) when performing a task using ACSmI and without using it (manually).	More than 30%-time reduction

Flexibility/Scalability

In this Thesis and following the approach proposed in [162] we have based the evaluation of scalability in the method proposed by the authors in [163].

They state that in general, "*it is expected that if a service scales up ideally then the increase in the number of CPU cores should be matched by proportional increase in the number of requests processes per unit of time*". Ideal scaling is described by the formula below:

$$D' / D = I' / I$$

Where D' – number of requests processes per unit of time after scaling.

D – Number of requests processes per unit of time before scaling.

I' – Number of cores after scaling.

I – Number of cores before scaling.

Authors in [163] also claim that "*real systems are expected to operate below the level of the ideal scaling behaviour and the aim of measuring scalability is to quantify the extent to which the real system behaviour differs from the ideal behaviour*".

Thus, following their approach, if the following ratio is close to 1 the system is close to ideal quality scalability, and if it is close to 0 the quality scalability of the system is much less than ideal:

$$D' / D / I' / I \rightarrow 1$$

Scalable systems have an advantage because they are more adaptable to the changing needs or demands of their users or clients, in other words, they are more flexible. Thus, the ability to scale may define the flexibility ACSmI.

Table 27. *Template for Scalability/Flexibility assessment.*

Dimension	Evaluation method	Acceptance criteria
Scalability	Comparison of the ideal scaling behaviour and the real scaling behaviour.	The real scaling behaviour corresponds to at least 70% of the ideal one.
Flexibility	–	The system should perform both vertical and horizontal scaling.

Interoperability

Interoperability is defined [164] as a property of a system which can seamlessly work with other products or systems. In the context of this thesis this generic definition can apply to more than one area. On one hand, ACSmI has been designed to enable the programmatic communication with other systems that may benefit from the information offered by ACSmI. This communication can be made by different means:

- I1: ACSmI is enabled to write and read from a JSON file information that could be shared from/with other systems. This JSON file called application description and is part of the DECIDE solution [74].
- I2: ACSmI use API calls as an information-sharing mechanism. For these communications, the systems will be considered interoperable if their APIs are readily available and well documented using a standardized language, such as OpenAPI.
- I3: ACSmI needs to interact with or rely on information gained through interaction with different cloud service providers, to obtain information about their services and to contract them. In that sense it will be considered interoperable if it is able to communicate with a minimum of 5 CSPs.

Table 28. *Template for interoperability assessment.*

Dimension	Evaluation method	Acceptance criteria
Interoperability I1	Assessment of the CRUD functionality from the JSON Application description file	Correct parsing of all the information required/provided from/to the JSON application description file
Interoperability I2	Assessment of the provided APIs	API available and well documented
Interoperability I3	Number of CSPs included in the solution	Able to communicate with a minimum of 5 CSPs.

8.5.5 Results

Key responses from the feedback obtained from the DevOps team member when using ACSmI regarding their experience:

Cloud services discovery screen/functionality:

- Some validations and selection mechanisms were missed that could help the user to fine tune the NFR matching as dropdown menu selectors for Providers, Provider, Region and Underpinning Technology for example.
- The search option is appreciated to help the user to find quickly a service when the list is too long.
- The ACSmI tool provides efficiency for evaluating a user's selected NFRs options for each of the service classifications and provides an instant response to illustrate matching based on the user's specified values of each of their NFRs to help evaluate the most effective options when selecting a service solution. This efficiency exists, as without this tool the user would have to search each CSP individually to identify matches and reach a decision for the best solution on the information that was discovered manually [19].
- Cloud Services information includes the Legal Level. It is assigned to each cloud service a legal level (tier 1, 2 or 3, tier 1 being the best), based on an extensive questionnaire and guidelines set guiding the legal interpretation of the existing legal compliance situation of a cloud service. The Cloud Service discovery can be performed also with respect to this cloud legal level which is really a novelty with respect to cloud services characterization.
- The discovery functionality was very useful to discover only the cloud services that fulfilled a set of characteristics. The effort needed for the discovery of the services decreased in more than 50%
- The reusability of cloud service offerings through the ACSmI is notable. The reusability of proven legal services with no additional set-ups is of 80%. The legal discovery part is "unique", different legal levels for the cloud services can be requested based on the simple and layered controls established. Once a Legal Expert assesses the legal level the changes are automatically recorded, and the cloud services can be automatically re-used. This approach requires a legal expert to confirm Cloud Services controls, that's why the reusability is not 100%.

Cloud service endorsement screen/functionality:

- The "view option" of any selected service is complete and the information related to the service can be quickly seen.
- Just from this same window we can edit the properties of the selected service directly, which is comfortable, and we can with a few clicks easily edit the properties of our service and upload the related legal documentation.
- The API section provides very the complete documentation of ACSmI API including code examples.

Cloud service contracting process/functionality:

- The proposed process is completely automatic and intuitive, and the experience is seamless.
- ACSmI contracting decreases the time needed to contract cloud services, by 70%. The contract is done automatically in less than 2 minutes. This is almost the half of the time needed without ACSmI per cloud service.

Cloud services monitoring functionality:

- Fulfilment of the application's MCSLA by 99%. As the MCSLA and the underlying CSLAs of the running cloud services can continuously be monitored and the violations are automatically detected the MCSLA is hardly broken [19].

Test case 1: Clinical Trial Governance Platform

Table 29. *Availability assessment in Test case 1.*

Evaluation method	**Acceptance criteria**	**Level of accomplishment**
Y1: % of time that the service is running Y2: Degree of fulfilment of the application MCSLA related to availability	Y1: 99% Y2: 99.99%	Currently no failure so meeting acceptance criteria Currently 100% availability and fulfilling terms of MCSLA

Table 30. *Flexibility /scalability assessment in Test case 1.*

Dimension	**Evaluation method**	**Acceptance criteria**	**Level of Accomplishment**
Scalability	Comparison of the ideal scaling behavior and the real scaling behavior.	The real scaling behavior corresponds to 70% of the existing model.	As tested real scaling behavior corresponds to 100% of the existing model
Flexibility	–	The system should perform both vertical and horizontal scaling.	The system performs both vertical and horizontal scaling.

Table 31. *Interoperability assessment in Test case 1.*

Acceptance criteria	**Level of accomplishment**
Able to communicate with a minimum of 5 CSPs.	10, Fulfilled. AZURE, Arsys, Amazon, Cloud Sigma, Google and AIMES

Table 32. *Efficiency assessment in Test case 1.*

Evaluation method	**Acceptance criteria**	**Level of accomplishment**
Resources usage optimization (usually time reduction percentage) when performing a task using ACSmI and without using it (manually).	More than 30%-time reduction.	Accepting that the development process itself is likely to be roughly equal, all other aspects exceeded the original time reduction.

Test case 3: Cloud Service provider incidence tracking

Key responses from the DevOps team member regarding his experience with ACSmI:

Table 33. *Availability assessment in Test case 3.*

Evaluation method	Acceptance criteria	Level of accomplishment
Y1: % of time that the service is running Y2: Degree of fulfilment of the application MCSLA related to availability	Y1: 99% Y2: 99.99%	Y1: ACSmI service has been running without a break for the last month of the project. Y2: It has been observed that there is a 100% fulfilment. Thus far no failures within the test environment.

Table 34. *Scalability/Flexibility assessment in Test case 3.*

Dimension	Evaluation method	Acceptance criteria
Scalability	Comparison of the ideal scaling behaviour and the real scaling behaviour.	The real scaling behaviour corresponds to at least 70% of the ideal one.
Flexibility	–	The system performs both vertical and horizontal scaling.

Table 35. *Interoperability assessment in Test case 3.*

Acceptance criteria	Level of accomplishment
Able to communicate with a minimum of 5 CSPs.	10, Fulfilled AZURE, Arsys, Amazon, Cloud Sigma, Google and AIMES

Table 36. *Efficiency assessment in Test case 3.*

Evaluation method	Acceptance criteria	Level of accomplishment
Resources usage optimization (usually time reduction percentage) when performing a task using ACSmI and without using it (manually).	More than 30%-time reduction	In the case of the Test case 3 we found relevant time savings in this part just for the automatisms that ACSmI provides with more than 50% of time saved from the original scenario for these operations. More info in section 8.4.3.

Extended DevOps concept

The validation of the Extended DevOps concept for multi-cloud native applications was mainly focused on the pre-deployment, deployment, and continuous monitoring phase, as these were the activities supported by ACSmI. Nevertheless, the design and adaptation related activities showed to be beneficial to improve the development and business continuity of the multi-cloud application as reported by the DevOps teams.

As it has been shown in section 8.4, 8.5.3, and 8.5.4 all the activities suggested by this Thesis in the Extended DevOps concept have proved to be efficient in the consecution of their objectives. Especially when supported by tools such as ACSmI.

Therefore, as overall result, we can conclude that the Extended DevOps concept has proved to serve in contributing to improve the Development and Operation of multi-cloud native applications since it helped to [19]:

- Discover optimized cloud services: The pre-deployment phase provides efficiency for evaluating a user's potential NFR options for each of the service classifications and provide an instant response to illustrate matching based on the user's specified values of each of their NFRs to help evaluate the most effective options when selecting a service solution. This efficiency exists, as without this tool supported activity the user would have to search each CSP in-dividually to identify matches and reach a decision for the best solution on the information that was discovered manually.
- Increase the business continuity of the multi-cloud applications: The operation and self-healing phase enhances the fulfilment of the application's Service Level Agreement (SLA). As the SLA and the underlying SLAs of the running cloud services can continuously be monitored and the violations are automatically detected the application's SLA is hardly broken.

8.6 Summary

The chapter has presented the evaluation results for the ACSmI and the Extended DevOps concept; this shows that the use of the Extended DevOps concept supported by ACSmI positively impacts the effort needed to discover, contract, and manage multiple cloud services at runtime.

Four quantitative evaluation experiments were carried out to monitor the performance of ACSmI (in terms of time saved to the user of the solution). Three of them included real industrial scenarios and users. The fourth one used the Sock Shop Application which incorporated a more complex application (in terms of components number). The quantitative evaluation also included the assessment of the functional requirements fulfilled by the prototype from the ones elicited at design time.

The qualitative evaluation covered the early user evaluation, by two experts' teams (belonging to the test case 1 and test case 3). As a result, we obtained a set of comments and feedback from potential users of the solution. The qualitative also provided information of the ACSmI performance in terms of relevant non-functional properties: Availability, efficiency, flexibility/scalability, and interoperability.

9. Conclusions and future work

Chapter 9 closes the thesis by summarizing its contributions and their impact in the advancement of multi-Cloud service brokering research. We present in Section 9.1 a summary of the thesis contributions, and in Section 9.2 we discuss open research issues and the possible extensions to address them.

9.1 Summary and research findings

As stated in section 3.1, the main objective of this research work is to find an answer to the following fundamental question:

How can an adequate cloud broker intermediator for multi cloud native applications can be realized to re-use and combine cloud services, for assembling a network of interoperable, legal compliant, quality assessed (against SLAs) single and composite cloud services for multi-cloud aware applications deployment and operation?

Pursuing the answer to the above question, we have achieved several contributions. The main scientific outputs and contributions from this Thesis are the following:

- **Result 1 (R1) - Extended DevOps concept for multi cloud native applications** – Demonstrates H1.
  R1 is an extension to the traditional DevOps philosophy including activities prior to the development and after the operation of the multi cloud application which address the specific needs of these kind of applications. This customized DevOps concept for the multi cloud native applications provides the DevOps teams with a set of guidelines about the efficient implementation of the DevOps principles in a multi cloud environment.
- **Result 2 (R2) - Advanced Cloud Service meta-Intermediator (ACSmI)** - Demonstrates H2-H3
  R2 is a cloud services brokerage solution that implements a cloud federation, supporting the seamless discovery, set up, and contracting of cloud services and the proactive monitoring the QoS of the selected services. ACSmI architecture includes a cloud services description model for describing uniformly the cloud services with special attention to relevant (i.e., performance, availability, cost) and novel (i.e., legislation) non-functional characteristics of such services. It also provides cloud vendor agnostic discovery functionalities for the DevOps teams to select the best combination of cloud services and monitoring capabilities to continuously assess the performance of the contracted services against the specific non-functional properties. ACSmI also sets up the basis for a seamless change of service provider offering the runtime monitoring data and the alerts when a failure happens and exposing the needed interfaces to programmatically create new deployment configurations (i.e., new combination of cloud services) to substitute the failing service/s. This avoids vendor lock-in to the cloud services consumers and guarantees business continuity.
  The experiments show that proposed solution effectively saves up to 75% of the DevOps teams' effort to discover, contract and monitor cloud services [8].

Next, we recapitulate the contributions achieved throughout this thesis, which answer the above question by addressing the three research questions presented in Section 5.1.

ACSmI was designed to answer this research question to hide the complexity of cloud services selection and management and to support dynamically the monitoring of the services to ensure the fulfilment of the SLOs with respect to certain non-functional properties such as location, performance, availability and cost, as part of the SLA. ACSmI is an extensible framework where common, but crucial NFRs, for the cloud can be included both for services discovery and monitoring (i.e., load balancing, scalability, legal awareness). It has been proven, by a set of four experiments, that the use of ACSmI positively impacts the effort needed) to discover, contract and manage multiple cloud services at runtime.

Overall, this work distinguishes itself from existing research achievements with the following unique contributions:

- Characterization of *"multi cloud native applications"* in terms of architectural patterns and design and deployment models.
- Proposition of the novel concept of "Extended DevOps" for multi-cloud native applications, which extends the phases of the DevOps philosophy with specific practices to address the needs of multi cloud native applications.
- Implementation of a technology agnostic, efficient and cost saving utility for cloud service discovery, contracting, proactive continuous monitoring and CSLA
- Inclusion of relevant and up to now not addressed NFRs such as the legislation compliance in the characterization of the cloud services
- Implementation of the ISO/IEC 19086-1 standard to be able to monitor metrics in accordance with the CSLA established for a set of concrete NFR (availability, performance, location and cost).
- Provision of different and heterogeneous discovery and contracting mechanisms both for the big players and the small European cloud service providers (CSPs), facilitating the governance and co-living of both types of CSPs.

Overall, we can conclude that the functionality and performance of the proposed solution have been proven and the objectives proposed in this Thesis work have been achieved:

- We have proved the feasibility of the proposed cloud services meta-intermediator to support the deployment and operation of multi cloud native applications of different nature and with different deployment needs. The methods and tools proposed by ACSmI introduced significant savings in the efficient discovery, registration, management, contracting, monitoring and portability of cloud services so that these services can be published and accessible to be optimally used.
- We have also demonstrated that legal aspects can be and should be treated as a non-functional requirement. The approach proposed by ACSmI legal to incorporate and assess relevant aspects concerning regulations and legislation proposes a significant advance in the creation of trust in cloud services promoting the re-use and combination of these services, assembling a dynamic and re-configurable network of interoperable, legal compliant cloud services.
- Finally, we have also shown that the "Extended DevOps" concept makes it possible to adapt the SDLC and SOLC practices to the specific needs of multi cloud native applications. In particular, activities prior to the development and after the operation phases are of special relevance to this.

The major research hypothesis of the Thesis work has therefore been proved to be true, i.e., it has been demonstrated *H- It is possible to demonstrate that the proposed ACSmI can contribute to the creation of an ecosystem of trusted, interoperable and legal compliant cloud services fostering the uptake of cloud computing, with a special focus on multi-cloud aware applications that have specific non-functional requirements and needs*. Overall, it has been proved that ACSmI effectively saves up to 75% of the DevOps teams' effort to discover, contract and monitor cloud services [8].

In particular, the implications of this hypothesis have been thus demonstrated as follows:

- *H1- It is possible to define the multi-cloud concept from the application SDCL and SOCL perspective and demonstrate its validity in real use case scenarios*.
  The proposed Extended DevOps concept for multi cloud applications has demonstrated that multi-cloud applications have their own specific needs and that the adaptation of traditional

software development and software operation needs can improve the design, development, and operation of such applications.

- *H2- It is possible to discover, benchmark and select the best combination of Cloud Services based a set of specific non-functional requirements elicited by the end-user.*
  The validated ACSmI discovery component as part of the ACSmI solution enables the seamless discovery and benchmarking of cloud services against a set of non-functional requirements relevant in the context of multi-cloud native applications. The benefits provided by the solution increases as the number and needs of the components grows.
- *H3- It is possible to assess and monitor the fulfilment of non-functional requirements of contracted cloud services against composed CSLAs and legislation and react to the violation of these requirements.*
  The verified ACSmI monitoring assesses the compliance of composed Cloud Service Level Agreements in accordance with ISO/IEC 19086-1 monitoring the QoS for each service at runtime thus enabling the users to make informed selection about relevant non-functional requirements of the available cloud services. It also incorporates the legal aspect as a new NFR and proposes the approach to include and assess legal related characteristics through the ACSmI legal component in an efficient and lightweight manner. This approach fosters the creation and utilization of trustworthy cloud services in real industrial scenarios such as e-health or energy as verified through the test cases.

Furthermore, the proposed solution ACSmI has been proved to be interoperable with six different cloud providers in terms of cloud service characterization, contracting and monitoring. Automatic discovery, contracting and monitoring features provided by ACSmI contribute to the automation of the portability between distinct CSPs. ACSmI also provides the possibility of programmatic access to these features through a set of APIs which paves the way for the incorporation of other modules that can trigger the process of service discovery, contracting and bringing up when a failure occurs.

9.2 Dissemination of results

Five (5) journal articles focused on the research presented in this this PhD have been submitted for the dissemination of the results:

J1) Article:

- Title: Understanding the challenges and novel architectural models of Multi-Cloud native applications – A systematic literature review.
- Authors: Juncal Alonso, Leire Orue-Echevarria, Valentina Casola, Ana Isabel Torre, Maider Huarte, Eneko Osaba, Jesús López Lobo.
- Journal: Journal of Network and Computer Applications (JNCA)
- ISSN: 1084-8045
- Impact:
  - Index: JCR-SCIE Impact Factor: 6.281 Year: 2020
  - Category: "COMPUTER SCIENCE, HARDWARE & ARCHITECTURE"
  - Journal position:5 Number of Journals within the category: 53
  - Quartile: Q1
- Status: Submitted.
- Related contribution: Extended DevOps concept for multi cloud native applications.

J2) Article:

- Title: CloudOps: Towards the Operationalization of the Cloud Continuum
- Authors: Juncal Alonso, Leire Orue-Echevarria, Maider Huarte.

- Journal: MDPI Applied Sciences
- ISSN: 2076-3417
- Impact:
  - Index: JCR- SCIE Impact Factor: 2.679 Year: 2020
  - Category: "ENGINEERING, MULTIDISCIPLINARY"
  - Journal position:38 Number of Journals within the category: 90
  - Quartile: Q2
- Status: Published. DOI: https://doi.org/10.3390/app12094347
- Year of publication: 2022
- Related contribution: Extended DevOps concept for multi cloud native applications

J3) Article:

- Title: ACSmI: A solution to address the challenges of Cloud services federation and monitoring towards the Cloud Continuum
- Authors: Juncal Alonso, Maider Huarte, Leire Orue-Echevarria.
- Journal: International Journal of Computational Science and Engineering (IJCSE)
- ISSN: 1742-7185
- Impact:
  - Index: JCR-ESCI Citations: 1245 Year: 2020
  - Category: "COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS"
  - Journal position:121 Number of Journals within the category: 142
  - Quartile: Q4
- Status: In Press
- Related contribution: ACSmI -Advanced Cloud Service meta-Intermediator

J4) Article:

- Title: Embracing IaC through the SecDevOps philosophy: Concepts, challenges, and a reference framework
- Journal: IEEE software
- ISSN: 0740-7459
- Impact:
  - Index: JCR-SCIE Impact Factor: 2.967 Year: 2020
  - Category: "COMPUTER SCIENCE, SOFTWARE ENGINEERING"
  - Journal position:26 Number of Journals within the category: 108
  - Quartile: Q1
- Authors: Juncal Alonso, Radosław Piliszek, Matja Canjkar
- Status: Submitted
- Related contribution: Extended DevOps concept for multi cloud native applications

J5) Article:

- Title: Optimization and prediction techniques for self-healing and self-learning applications in a trustworthy Cloud Continuum
- Authors: Juncal Alonso, Leire Orue-Echevarria, Jesus Lopez Lobo, Eneko Osaba, Josu Diaz de Arcaya, Iñaki Etxaniz.
- Journal: MDPI Information
- ISSN: 2078-2489
- Impact:
  - Index: JCR-ESCI Citations: 2.371 Year: 2020
  - Category: "COMPUTER SCIENCE, INFORMATION SYSTEMS"
  - Journal position:134 Number of Journals within the category: 223
  - Quartile: Q3
- Status: Published. DOI: https://doi.org/10.3390/info12080308

- Year of publication: 2021
- Related contribution: ACSmI -Advanced Cloud Service meta-Intermediator

Nine (9) International conferences publications related with this PhD:

C1) Publication:

- Title: Empowering Services Based Software in the Digital Single Market to Foster an Ecosystem of Trusted, Interoperable and Legally Compliant Cloud-services
- Authors: Juncal Alonso Ibarra, Leire Orue-Echevarria, Marisa Escalante and Gorka Benguria.
- Conference: CLOSER 2016. Proceedings of the International Conference on Cloud Computing and Services Science
- Year: 2016
- Status: Published.
- Related contribution: ACSmI -Advanced Cloud Service meta-Intermediator

C2) Publication:

- Title: Transformational Cloud Government (TCG): Transforming Public Administrations with a Cloud of public services
- Authors: Juncal Alonso Ibarra, Leire Orue-Echevarria, Marisa Escalante and Gorka Benguria.
- Conference: Cloud Forward 2016: From Distributed to Complete Computing. Procedia Computer Science ISSN: 1877-0509
- Year: 2016
- Status: Published. https://dblp.org/rec/conf/closer/IbarraOEB16.html
- Related contribution: ACSmI -Advanced Cloud Service meta-Intermediator

C3) Publication:

- Title: DECIDE: DevOps for Trusted, Portable and Interoperable Multi-Cloud Applications towards the Digital Single Market
- Authors: Juncal Alonso Ibarra, Leire Orue-Echevarria, Marisa Escalante and Gorka Benguria.
- Conference: CLOSER 2017. Proceedings of the International Conference on Cloud Computing and Services Science
- Year: 2017
- Status: Published. https://dl.acm.org/doi/10.5220/0006292403970404
- Related contribution: Extended DevOps concept for multi cloud native applications

C4) Publication:

- Title: Federated Cloud Service Broker (FCSB): An Advanced Cloud Service Intermediator for Public Administrations
- Authors: Juncal Alonso, Leire Orue-Echevarria, Marisa Escalante and Gorka Benguria.
- Conference: CLOSER 2017. Proceedings of the International Conference on Cloud Computing and Services Science
- Year: 2017
- Status: Published. https://www.researchgate.net/publication/317299487_Federated_Cloud_Service_Broker_FCSB_An_Advanced_Cloud_Service_Intermediator_for_Public_Administrations
- Related contribution: ACSmI -Advanced Cloud Service meta Intermediator

C5) Publication:

- Title: Towards Supporting the Extended DevOps Approach through Multicloud Architectural Patterns for Design and Pre-Deployment. A Tool Supported Approach
- Authors: Juncal Alonso, Marisa Escalante, Lena Farid, Maria Jose Lopez, Leire Orue-Echevarria and Simon Dutkowski
- Conference: SECLOUD 2018. Proceedings of the International Conference on Software Engineering for Service and Cloud Computing

- Year: 2018
- Status: Published. https://www.scitepress.org/papers/2018/68560/68560.pdf
- Related contribution: Extended DevOps concept for multi cloud native applications

C6) Publication:

- Title: DECIDE: DevOps for Trusted, Portable and Interoperable Multi-Cloud Applications towards the Digital Single Market
- Authors: Leire Orue-Echevarria, Juncal Alonso, Marisa Escalante, Kyriakos Stefanidis, Lorenzo Blasi.
- Conference: ". PROFES 2019 Proceedings of International Conference on Product-Focused Software Process Improvement
- Year: 2019
- Status: Published.
- Related contribution: Extended DevOps concept for multi cloud native applications

C7) Publication:

- Title: DECIDE: An Extended DevOps Framework for Multi-cloud Applications
- Authors: Juncal Alonso, Kyriakos Stefanidis, Leire Orue-Echevarria, Lorenzo Blasi, Michael Walker, Marisa Escalante, María José López, Simon Dutkowski.
- Conference: CCBDC 2019: Proceedings of the International Conference on Cloud and Big Data Computing
- Year: 2019
- Status: Published. https://link.springer.com/chapter/10.1007/978-3-030-35333-9_45
- Related contribution: Extended DevOps concept for multi cloud native applications

C8) Publication:

- Title: Contribution to the uptake of Cloud Computing solutions: Design of a cloud services intermediator to foster an ecosystem of trusted, interoperable and legal compliant cloud services. Application to multi-cloud aware software
- Authors: Juncal Alonso, Leire Orue-Echevarria, Marisa Escalante.
- Conference: WEBIST 2019. International Conference on Web Information Systems and Technologies.
- Year: 2019
- Status: Published. https://webist.scitevents.org/Abstract.aspx?idEvent=NlDFbzQWVmc=
- Related contribution: ACSmI -Advanced Cloud Service meta Intermediator

C9) Publication:

- Title: PIACERE: Programming trustworthy Infrastructure as Code in a Secure Framework
- Authors: Juncal Alonso, Christophe Joubert, Leire Orue-Echevarria, Matteo Pradella and Daniel Vladušič.
- Conference: SwForum workshop Proceedings
- Year: 2021
- Status: Published. http://ceur-ws.org/Vol-2878/paper2.pdf
- Related contribution: ACSmI -Advanced Cloud Service meta-Intermediator

Table 37. *Summary of publications related to the Thesis*

Id and year	Venue	Impact	Related contribution	Status
J1 (2022)	JNCA	JCR: Q1	Extended DevOps concept for multi cloud native applications	Submitted
J2 (2022)	MDPI Applied Sciences	JCR: Q2		Published
J4 (2022)	IEEE software	JCR: Q1		Submitted
C3 (2017)	CLOSER			Published
C5 (2018)	SECLOUD			Published
C6 (2019)	PROFES			Published
C7 (2019)	CCBDC			Published
J3 (2021)	IJCSE	JCR: Q4	ACSmI -Advanced Cloud Service meta Intermediator	In Press
J5 (2021)	MDPI Information	JCR: Q4		Published
C1 (2016)	CLOSER			Published
C2 (2016)	Cloud Forward			Published
C4 (2017)	CLOSER			Published
C8 (2019)	WEBIST			Published
C9 (2021)	SwForum workshop			Published

9.3 Open research issues

During the development of this research work some new open issues have been identified resulting in some new research trends.

9.3.1 New research trends for the development and operation of multi cloud applications.

Through this thesis we have characterized multi cloud native applications and proposed a novel extended DevOps approach covering phases on the pre-design and on the post operation of the multi-cloud application. During the analysis performed we have also identified future research lines for each of the four themes studied (which have already been validated from some of the publications of the author of this Thesis), 1) characterization of multi-cloud and multi-cloud native applications, 2) multi-cloud by design, 3) DevOps for multi-cloud and 4) secure multi-cloud native applications represent opportunities for future research:

- Characterization of the Cloud Continuum, including federated models for the cloud (1).
- Incorporation of the network elements into the multi cloud paradigm (1).
- Proposal of new approaches for stateful and stateless application components design and partition (2).
- Lightweight design profiles of software components to be deployed on the edge (2).
- Context aware design of multi-cloud applications architecture (2) (3).
- NFRs optimization (cost, latency, performance) for the different components of the multi cloud application (1) (3) (4).
- Component's synchronization, data consistency and cloud agnostic design of software systems (2) (4).
- Re-configuration and self-healing mechanism for the multi-cloud native applications and their components (3).

9.3.2 European Cloud Federation: Gaia X

GAIA-X [17] project stands for a federated high-performance, competitive, secure and trustworthy data and Cloud infrastructure for Europe. It pursues to provide a data infrastructure as a federated technical infrastructure consisting of components and services that make it possible to access data and to store, exchange and use it according to predefined rules. The main goals they want to achieve with such an initiative are: data sovereignty, reducing dependencies from non-European (Cloud) providers, broader adoption of Cloud Computing by the (European) SMEs, creation of an open, digital ecosystem to help European companies and business models to scale up competitively worldwide.

ACSmI can also contribute [8] to relevant elements of the European GAIA-X initiative, specifically to the federated catalogue, continuous monitoring, and certification of services. The inclusion of ACSmI in GAIA-X federated cloud would imply the adequation of the description model of the cloud services in ACSmI to the GAIA-X reference architecture.

9.3.3 ACSmI for the Cloud Continuum

The current IT market is more and more dominated by the "Cloud continuum". The Increasing ubiquity and proliferation of computing and data capabilities have led to a growth of complex computing environments with heterogenous digital resources. This computing continuum consists of different infrastructures that resources and services at the edge (edge computing), in the core (cloud computing), and along the data path (fog computing) as needed.

The heterogeneity of the "computing continuum" is broad and multi-level. In the "*traditional*" cloud, computing resources are typically provisioned through virtualization and containerization [165] with "infinite" resource availability thanks to horizontal scaling. In contrast, in edge computing, computational resources are scarce and must be managed very efficiently due to battery constraints or other limitations. The heterogeneity of the Cloud Continuum is difficult to manage and opinionated decisions on which deployment configuration use based on available resources and their characteristics is a time consuming an error prone activity for DevOps teams.

This Thesis paves the way to further investigate new requirements in an edge-cloud environment, including the characterisation of edge nodes and network services as available resources to be selected, brokered and monitored through the framework [19]. This will require the extension of the taxonomy for the service registry as well as the adaptation of the monitoring mechanisms and techniques:

- Mechanisms to characterize, discover, select and configure of diverse and heterogeneous IT infrastructural components (cloud, edge and fog computing).
- Operation and proactive monitoring of the dynamic and heterogenous Cloud continuum to be aware of new available resources, bottlenecks situations, or network reconfiguration needs.

9.3.4 ACSmI & Cloud Continuous Certification

As a future research trend, the work on the legal aspects could be extended to incorporate these into the monitoring phase so that the legal level is included in the continuous monitoring phase. To this respect the approach can be broaden to the incorporation of compositional certification monitoring feature, assuring that the composition of monitoring metrics from the cloud services fulfils the evidences required by the certification scheme at any time. In this case new monitoring parameters, metrics and the related techniques to acquire and securely store them shall be put in place. This research line can be addressed in the short term as the new European Cybersecurity Certification Scheme for Cloud Services

(EUCS) was launched in 2021 by the European Commission. ACSmI can be extended to support the characterization of the cloud services metrics defined in the EUCS and serve as an implementation of the automatic certification of Cloud Services and Cloud composite services, in what is called compositional cloud certification.

9.3.5 Support to the seamless change of clod service provider

ACSmI has been designed to improve the selection and management of several cloud services specially for applications which components are deployed on several different cloud services at the same time. This can be done in the first stage where the application is deployed for the first time but the support to the change of cloud service provider on the fly (while the application is running) and assuring business continuity is still an open issue. To this respect several future research lines have been identified to improve the support to seamless change of cloud service provider through ACSmI:

- Add failure prediction functionality to ACSmI, based on historic monitored data supporting the automatic reconfiguration of the cloud resources avoiding CSLA compromising situations before they even happen.
- Include self-healing mechanisms for both the infrastructure and the applications to be capable of being reconfigured (infrastructural elements) and re-deployed (software components) given the new environmental conditions
- Investigate on the support to seamless components portability without affecting the service continuity and addressing special needs for data and stateful components migration over different service providers and at run-time.

10. References

[1] D. Petcu, Multi-Cloud: expectations and current approaches, in: Proceedings of the 2013 International Workshop on Multi-Cloud Applications and Federated Clouds - MultiCloud '13, ACM Press, Prague, Czech Republic, 2013: p. 1. https://doi.org/10.1145/2462326.2462328.

[2] P. Raj, A. Raman, The Hybrid Cloud: The Journey Toward Hybrid IT, in: Software-Defined Cloud Centers, Springer International Publishing, Cham, 2018: pp. 91–110. https://doi.org/10.1007/978-3-319-78637-7_5.

[3] N. Niknejad, W. Ismail, I. Ghani, B. Nazari, M. Bahari, A.R.B.C. Hussin, Understanding Service-Oriented Architecture (SOA): A systematic literature review and directions for further investigation, Information Systems. 91 (2020) 101491. https://doi.org/10.1016/j.is.2020.101491.

[4] É. Michon, J. Gossa, S. Genaud, L. Unbekandt, V. Kherbache, Schlouder: A broker for IaaS clouds, Future Generation Computer Systems. Vol. 69 (2017) 11–23. https://doi.org/10.1016/j.future.2016.09.010.

[5] J. Tordsson, R.S. Montero, R. Moreno-Vozmediano, I.M. Llorente, Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers, Future Generation Computer Systems. 28 (2012) 358–367. https://doi.org/10.1016/j.future.2011.07.003.

[6] L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, O. Rana, The Internet of Things, Fog and Cloud continuum: Integration and challenges, Internet of Things. 3–4 (2018) 134–155. https://doi.org/10.1016/j.iot.2018.09.005.

[7] J. Alonso, L. Orue-Echevarria Arrieta, M. Escalante, G. Benguria, G. Echevarria, Federated Cloud Service Broker (FCSB): An Advanced Cloud Service Intermediator for Public Administrations, 2017. https://doi.org/10.5220/0006285003840391.

[8] ACSmI: A solution to address the challenges of Cloud services federation and monitoring towards the Cloud Continuum, IJCSE. In Press (n.d.).

[9] J. Alonso, L. Orue-Echevarria Arrieta, M. Escalante, G. Benguria, G. Echevarria, Federated Cloud Service Broker (FCSB): An Advanced Cloud Service Intermediator for Public Administrations, in: Proceedings of Cloud Computing and Service Science, Springer, Porto, Portugal, 2017: pp. 384–391. https://doi.org/10.5220/0006285003840391.

[10] National Institute of Standards and Technology, Cloud Computing Standards Roadmap Working Group, NIST Cloud Computing Standards Roadmap, National Institute of Standards and Technology, 2013. https://doi.org/10.6028/NIST.SP.500-291r2.

[11] 3 Cloud Computing Service Delivery Models | 2nd Watch, (n.d.). https://www.2ndwatch.com/blog/back-to-the-basics-the-3-cloud-computing-service-delivery-models/ (accessed March 3, 2022).

[12] The NIST definition on Cloud Computing, (n.d.).

[13] A.J. Ferrer, J.M. Marquès, J. Jorba, Towards the Decentralised Cloud: Survey on Approaches and Challenges for Mobile, Ad hoc, and Edge Computing, ACM Comput. Surv. 51 (2019) 1–36. https://doi.org/10.1145/3243929.

[14] FederatedCloud_RA_PP_022021.pdf, (n.d.). http://www.pledger-project.eu/FederatedCloud_RA_PP_022021.pdf (accessed March 3, 2022).

[15] EOSC | EOSC Portal, (n.d.). https://eosc-portal.eu/about/eosc (accessed April 29, 2022).

[16] Project GAIA-X – A Federated Data Infrastructure as the Cradle of a Vibrant European Ecosystem, (n.d.) 56.

[17] DE-CIX Management GmbH, G. Eggers, B. Fondermann, Google Germany, B. Maier, K. Ottradovetz, J. Pfrommer, R. Reinhardt, R. Hannes, A. Schmieg, S. Steinbuß, P. Trinius, A. Weiss, C. Weiss, Wilfling Sabine, GAIA-X: Technical Architecture, Federal Ministry for Economic Affairs and Energy (BMWi), 2020.

[18] 2020_GAIA-X_Presentation_overview.pdf, (n.d.).

[19] J. Alonso, L. Orue-Echevarria, M. Huarte, CloudOps: Towards the Operationalization of the Cloud Continuum: Concepts, Challenges and a Reference Framework, Applied Sciences. 12 (2022) 4347. https://doi.org/10.3390/app12094347.

[20] R. Mahmud, R. Kotagiri, R. Buyya, Fog Computing: A Taxonomy, Survey and Future Directions, in: B. Di Martino, K.-C. Li, L.T. Yang, A. Esposito (Eds.), Internet of Everything, Springer Singapore, Singapore, 2018: pp. 103–130. https://doi.org/10.1007/978-981-10-5861-5_5.

[21] M. Iorga, L. Feldman, R. Barton, M.J. Martin, N. Goren, C. Mahmoudi, Fog computing conceptual model, National Institute of Standards and Technology, Gaithersburg, MD, 2018. https://doi.org/10.6028/NIST.SP.500-325.

[22] M. Satyanarayanan, The Emergence of Edge Computing, Computer. 50 (2017) 30–39. https://doi.org/10.1109/MC.2017.9.

[23] K. Cao, Y. Liu, G. Meng, Q. Sun, An Overview on Edge Computing Research, IEEE Access. 8 (2020) 85714–85728. https://doi.org/10.1109/ACCESS.2020.2991734.

[24] X. Wang, Y. Han, V.C.M. Leung, D. Niyato, X. Yan, X. Chen, Fundamentals of Edge Computing, in: Edge AI, Springer Singapore, Singapore, 2020: pp. 15–32. https://doi.org/10.1007/978-981-15-6186-3_2.

[25] N. Kratzke, P.-C. Quint, Understanding cloud-native applications after 10 years of cloud computing - A systematic mapping study, Journal of Systems and Software. 126 (2017) 1–16. https://doi.org/10.1016/j.jss.2017.01.001.

[26] F. Leymann, U. Breitenbücher, S. Wagner, J. Wettinger, Native Cloud Applications: Why Monolithic Virtualization Is Not Their Foundation, in: M. Helfert, D. Ferguson, V. Méndez Muñoz, J. Cardoso (Eds.), Cloud Computing and Services Science, Springer International Publishing, Cham, 2017: pp. 16–40. https://doi.org/10.1007/978-3-319-62594-2_2.

[27] L. Blasi, D4.1 Initial DECIDE ADAPT Architecture, (n.d.) 77.

[28] How leading industries are driving multi-cloud adoption | ITProPortal, (n.d.). https://www.itproportal.com/features/how-leading-industries-are-driving-multi-cloud-adoption/ (accessed January 1, 2022).

[29] ISO - ISO/IEC JTC 1/SC 38 - Cloud computing and distributed platforms, (n.d.). https://www.iso.org/committee/601355.html (accessed March 3, 2022).

[30] DECIDE, D3.1 Initial architectural patterns for implementation deployment and optimization_v1.0_20171130.pdf, (n.d.). https://www.decide-h2020.eu/sites/decide.drupal.pulsartecnalia.com/files/documents/D3.1%20Initial%20architectural%20patterns%20for%20implementation%20deployment%20and%20optimization_v1.0_20171130.pdf (accessed March 3, 2022).

[31] GDPR.pdf, (n.d.). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L:2016:119:FULL&from=EN (accessed January 17, 2021).

[32] H. CHANG, Data Protection Regulation and Cloud Computing, 2015.

[33] J. Kong, X. Fan, K.P. Chow, Introduction to cloud computing and security issues, in: Chapters, Edward Elgar Publishing, 2015: pp. 8–25. https://ideas.repec.org/h/elg/eechap/15891_1.html (accessed January 17, 2021).

[34] C. Reed, Information in the cloud: ownership, control and accountability, in: Privacy and Legal Issues in Cloud Computing, Edward Elgar Publishing, 2015: pp. 139–159. https://doi.org/10.4337/9781783477074.00014.

[35] S. Pearson, G. Yee, eds., Privacy and Security for Cloud Computing, Springer London, London, 2013. https://doi.org/10.1007/978-1-4471-4189-1.

[36] ISO - ISO/IEC 19086-1:2016 - Information technology — Cloud computing — Service level agreement (SLA) framework — Part 1: Overview and concepts, (n.d.). https://www.iso.org/standard/67545.html (accessed March 4, 2022).

[37] DECIDE, D3.15 Final multi-cloud native application composite CSLA definition, (2019). https://www.decide-

h2020.eu/sites/decide.drupal.pulsartecnalia.com/files/newsletters/D3.15%20Final%20multi-cloud%20native%20application%20composite%20CSLA%20definition%20v1.0_20190531.pdf.

[38] Ticketmaster: Buy Verified Tickets for Concerts, Sports, Theater and Events, (n.d.). https://www.ticketmaster.com/ (accessed May 26, 2022).

[39] K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee, A Design Science Research Methodology for Information Systems Research, Journal of Management Information Systems. 24 (2007) 45–77. https://doi.org/10.2753/MIS0742-1222240302.

[40] J. Venable, J. Pries-Heje, R. Baskerville, A Comprehensive Framework for Evaluation in Design Science Research, in: K. Peffers, M. Rothenberger, B. Kuechler (Eds.), Design Science Research in Information Systems. Advances in Theory and Practice, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012: pp. 423–438. https://doi.org/10.1007/978-3-642-29863-9_31.

[41] S.A. Nizamani, A Quality-aware Cloud Selection Service for Computational Modellers, Published PhD thesis, University of Leeds, 2012.

[42] T.F. Fortiş, V.I. Munteanu, V. Negru, A taxonomic view of cloud computing services, International Journal of Computational Science and Engineering. Vol. 11 (2015) 17–28. https://doi.org/10.1504/IJCSE.2015.071360.

[43] G. Neelakanta, Broker-Mediated Multiple-Cloud Orchestration Mechanisms for Cloud Computing, Published PhD thesis, National University of Singapore, 2012. https://core.ac.uk/download/pdf/48657036.pdf (accessed October 16, 2020).

[44] V. Jurg van, P. Flavia, Programming Amazon EC2, O'Reilly Media, Inc., 2011. https://www.oreilly.com/library/view/programming-amazon-ec2/9781449303617/ (accessed October 16, 2020).

[45] A. Elhabbash, F. Samreen, J. Hadley, Y. Elkhatib, Cloud Brokerage: A Systematic Survey, ACM Computing Surveys. Vol. 51 (2019) 1–28. https://doi.org/10.1145/3274657.

[46] J. Alonso, L. Orue-Echevarria, M. Escalante, G. Benguria, Empowering Services based Software in the Digital Single Market to Foster an Ecosystem of Trusted, Interoperable and Legally Compliant Cloud-Services:, in: Proceedings of the 6th International Conference on Cloud Computing and Services Science, SCITEPRESS - Science and and Technology Publications, Rome, Italy, 2016: pp. 283–288. https://doi.org/10.5220/0005893302830288.

[47] J. Alonso, L. Orue-Echevarria, M. Escalante, Contribution to the uptake of Cloud Computing solutions: Design of a cloud services intermediator to foster an ecosystem of trusted, interoperable and legal compliant cloud services. Application to multi-cloud aware software, in: Proceedings of the 15th International Conference on Web Information Systems and Technologies (WEBIST), Zenodo, Vienna, Austria, 2019. https://doi.org/10.5281/zenodo.3748988.

[48] G.S. Zhou, W. Du, H.C. Lin, X.W. Yan, An approach for public cloud trustworthiness assessment based on users' evaluation and performance indicators, International Journal of Computational Science and Engineering. Vol. 19 (2019) 206–214. https://doi.org/10.1504/IJCSE.2019.100241.

[49] Future Cloud Cluster | Clusters of European Projects on Cloud, (n.d.). https://eucloudclusters.wordpress.com/future-cloud/ (accessed March 4, 2022).

[50] DECIDE, D5.1 ACSml requirements and technical design v3.0_20170531.pdf, n.d. https://www.decide-h2020.eu/sites/decide.drupal.pulsartecnalia.com/files/documents/D5.1%20ACSml%20requirements%20and%20technical%20design%20v3.0_20170531.pdf (accessed June 1, 2022).

[51] AWS Free Tier, (n.d.). https://aws.amazon.com/free/?trk=ps_a134p000003yhdRAAQ&trkCampaign=acq_paid_search_brand&sc_channel=ps&sc_campaign=acquisition_IBERIA&sc_publisher=google&sc_category=core&sc_country=IBERIA&sc_geo=EMEA&sc_outcome=Acquisition&sc_detail=amazon%20aws&sc_content=Amazon%20AWS_e&sc_matchtype=e&sc_segment=455709741594&sc_medium=ACQ-P|PS-GO|Brand|Desktop|SU|AWS|Core|IBERIA|EN|Text&s_kwcid=AL!4422!3!455709741594!e!!g!!amazon%20aws&ef_id=Cj0KCQiA1KiBBhCcARIsAPWqoSrZosqHPhBvN55scw7ZdoXr8_iftyF7ro5vDjZG

o6yBHaRf2RRBJVcaAioxEALw_wcB:G:s&s_kwcid=AL!4422!3!455709741594!e!!g!!amazon%20aws&all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc (accessed February 15, 2021).

[52] Hybrid Cloud Solutions | HPE, (n.d.). https://www.hpe.com/us/en/solutions/cloud.html (accessed February 15, 2021).

[53] Hybrid Cloud Solutions | IBM, (n.d.). https://www.ibm.com/cloud/hybrid (accessed February 15, 2021).

[54] Appcara, (n.d.). http://www.appcara.com/ (accessed February 15, 2021).

[55] Cloud Service Brokerage & Multi Cloud Management | Jamcracker, (n.d.). https://www.jamcracker.com/ (accessed February 15, 2021).

[56] Juju | Operator lifecycle manager for K8s and traditional workloads, (n.d.). https://juju.is/ (accessed February 15, 2021).

[57] Helix Nebula Marketplace Catalogue | Helix Nebula, (n.d.). https://www.helix-nebula.eu/publications/reports/helix-nebula-marketplace-catalogue (accessed February 15, 2021).

[58] Applications - Data.gov, (n.d.). https://data.gov/applications/ (accessed May 26, 2022).

[59] Digital Marketplace, (n.d.). https://www.digitalmarketplace.service.gov.uk/ (accessed May 26, 2022).

[60] Marketplace | NZ Digital government, (n.d.). https://www.digital.govt.nz/products-and-services/products-and-services-a-z/marketplace/ (accessed May 26, 2022).

[61] V. Casola, A. De Benedictis, M. Rak, U. Villano, Security-by-design in multi-cloud applications: An optimization approach (PS51), Information Sciences. 454–455 (2018) 344–362. https://doi.org/10.1016/j.ins.2018.04.081.

[62] A.F. Leite, V. Alves, G.N. Rodrigues, C. Tadonki, C. Eisenbeis, A.C.M.A. de Melo, Dohko: an autonomic system for provision, configuration, and management of inter-cloud environments based on a software product line engineering method (PS68), Cluster Comput. 20 (2017) 1951–1976. https://doi.org/10.1007/s10586-017-0897-1.

[63] A. Buzachis, M. Fazio, A. Celesti, M. Villari, Osmotic Flow Deployment Leveraging FaaS Capabilities (PS69), in: R. Montella, A. Ciaramella, G. Fortino, A. Guerrieri, A. Liotta (Eds.), Internet and Distributed Computing Systems, Springer International Publishing, Cham, 2019: pp. 391–401. https://doi.org/10.1007/978-3-030-34914-1_37.

[64] O.A. Wahab, J. Bentahar, H. Otrok, A. Mourad, Towards Trustworthy Multi-Cloud Services Communities: A Trust-Based Hedonic Coalitional Game (PS52), IEEE Transactions on Services Computing. 11 (2018) 184–201. https://doi.org/10.1109/TSC.2016.2549019.

[65] M. Gao, M. Chen, A. Liu, W.H. Ip, K.L. Yung, Optimization of Microservice Composition Based on Artificial Immune Algorithm Considering Fuzziness and User Preference (PS79), IEEE Access. 8 (2020) 26385–26404. https://doi.org/10.1109/ACCESS.2020.2971379.

[66] B. Somoskői, S. Spahr, E. Rios, O. Ripolles, J. Dominiak, T. Cserveny, P. Bálint, P. Matthews, E. Iturbe, V. Muntés-Mulero, Airline Application Security in the Digital Economy: Tackling Security Challenges for Distributed Applications in Lufthansa Systems (PS46), in: N. Urbach, M. Röglinger (Eds.), Digitalization Cases, Springer International Publishing, Cham, 2019: pp. 35–58. https://doi.org/10.1007/978-3-319-95273-4_3.

[67] J. Jofre, C. Velayos, G. Landi, M. Giertych, A.C. Hume, G. Francis, A. Vico Oton, Federation of the BonFIRE multi-cloud infrastructure with networking facilities (PS38), Computer Networks. 61 (2014) 184–196. https://doi.org/10.1016/j.bjp.2013.11.012.

[68] Hybrid Cloud Placement Algorithm (PS17), in: IEEE Conference Publication, n.d. https://ieeexplore.ieee.org/document/7033655 (accessed March 3, 2021).

[69] I. Elgedawy, SULTAN: A Composite Data Consistency Approach for SaaS Multi-cloud Deployment (PS10), in: 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), 2015: pp. 122–131. https://doi.org/10.1109/UCC.2015.28.

[70] A. Oprescu, A. Antonescu, Y. Demchenko, C. d Laat, ICOMF: Towards a Multi-cloud Ecosystem for Dynamic Resource Composition and Scaling (PS15), in: 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, 2013: pp. 49–55. https://doi.org/10.1109/CloudCom.2013.14.

[71] N. Kaviani, E. Wohlstadter, R. Lea, MANTICORE: A framework for partitioning software services for hybrid cloud (PS5), in: 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings, IEEE, Taipei, Taiwan, 2012: pp. 333–340. https://doi.org/10.1109/CloudCom.2012.6427541.

[72] E.D. Nitto, M.A.A. da Silva, D. Ardagna, G. Casale, C.D. Craciun, N. Ferry, V. Muntes, A. Solberg, Supporting the Development and Operation of Multi-cloud Applications: The MODAClouds Approach (PS16), in: 2013 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, IEEE, Timisoara, Romania, 2013: pp. 417–423. https://doi.org/10.1109/SYNASC.2013.61.

[73] A.J. Ferrer, D.G. Pérez, R.S. González, Multi-cloud Platform-as-a-service Model, Functionalities and Approaches (PS18), Procedia Computer Science. 97 (2016) 63–72. https://doi.org/10.1016/j.procs.2016.08.281.

[74] J. Alonso, K. Stefanidis, L. Orue-Echevarria, L. Blasi, M. Walker, M. Escalante, M.J. López, S. Dutkowski, DECIDE: An Extended DevOps Framework for Multi-cloud Applications (PS33), in: Proceedings of the 2019 3rd International Conference on Cloud and Big Data Computing, ACM, Oxford United Kingdom, 2019: pp. 43–48. https://doi.org/10.1145/3358505.3358522.

[75] A. Brogi, J. Carrasco, J. Cubo, F. D'Andria, E. Di Nitto, M. Guerriero, D. Pérez, E. Pimentel, J. Soldani, SeaClouds: An Open Reference Architecture for Multi-Cloud Governance (PS56), (2016). https://riuma.uma.es/xmlui/handle/10630/12561 (accessed March 3, 2021).

[76] P. Jamshidi, C. Pahl, S. Chinenyeze, X. Liu, Cloud Migration Patterns: A Multi-cloud Service Architecture Perspective (PS27), in: F. Toumani, B. Pernici, D. Grigori, D. Benslimane, J. Mendling, N. Ben Hadj-Alouane, B. Blake, O. Perrin, I. Saleh Moustafa, S. Bhiri (Eds.), Service-Oriented Computing - ICSOC 2014 Workshops, Springer International Publishing, Cham, 2015: pp. 6–19. https://doi.org/10.1007/978-3-319-22885-3_2.

[77] Q. Li, Z. Wang, W. Li, Z. Cao, R. Du, H. Luo, Model-based services convergence and multi-clouds integration (PS39), Computers in Industry. 64 (2013) 813–832. https://doi.org/10.1016/j.compind.2013.05.003.

[78] A. Almeida, F. Dantas, E. Cavalcante, T. Batista, A Branch-and-Bound Algorithm for Autonomic Adaptation of Multi-cloud Applications (PS6), in: 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2014: pp. 315–323. https://doi.org/10.1109/CCGrid.2014.25.

[79] M.M. Alshammari, A.A. Alwan, A. Nordin, I.F. Al-Shaikhli, Disaster recovery in single-cloud and multi-cloud environments: Issues and challenges (PS75), in: 2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS), IEEE, Salmabad, 2017: pp. 1–7. https://doi.org/10.1109/ICETAS.2017.8277868.

[80] S.N.V. Kumar, R. Meenakshi, Securing multi-cloud by auditing (PS42), in: 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS), IEEE, Chennai, India, 2017: pp. 253–258. https://doi.org/10.1109/SSPS.2017.8071601.

[81] C.-M. Chituc, Towards a Methodology for Trade-off Analysis in a Multi-cloud Environment Considering Monitored QoS Metrics and Economic Performance Assessment Results (PS77), in: 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, Vancouver, BC, Canada, 2015: pp. 479–482. https://doi.org/10.1109/CloudCom.2015.87.

[82] R. Yasrab, N. Gu, Multi-cloud PaaS Architecture (MCPA): A Solution to Cloud Lock-In (PS32), in: 2016 3rd International Conference on Information Science and Control Engineering (ICISCE), IEEE, Beijing, China, 2016: pp. 473–477. https://doi.org/10.1109/ICISCE.2016.108.

[83] N. Ferry, F. Chauvel, H. Song, A. Rossini, M. Lushpenko, A. Solberg, CloudMF: Model-Driven Management of Multi-Cloud Applications (PS11), ACM Trans. Internet Technol. 18 (2018) 1–24. https://doi.org/10.1145/3125621.

[84] A. Celesti, M. Fazio, A. Galletta, L. Carnevale, J. Wan, M. Villari, An approach for the secure management of hybrid cloud–edge environments (PS8), Future Generation Computer Systems. 90 (2019) 1–19. https://doi.org/10.1016/j.future.2018.06.043.

[85] A. Kopper, D. Fürstenau, S. Zimmermann, S. Klotz, C. Rentrop, H. Rothe, S. Strahringer, M. Westner, Shadow IT and Business-Managed IT: A Conceptual Framework and Empirical Illustration, International Journal of IT/Business Alignment and Governance. 9 (2018) 53–71. https://doi.org/10.4018/IJITBAG.2018070104.

[86] L. Heilig, E. Lalla-Ruiz, S. Voß, Modeling and solving cloud service purchasing in multi-cloud environments (PS49), Expert Systems with Applications. 147 (2020) 113165. https://doi.org/10.1016/j.eswa.2019.113165.

[87] H.J. Syed, A. Gani, R.W. Ahmad, M.K. Khan, A.I.A. Ahmed, Cloud monitoring: A review, taxonomy, and open research issues, Journal of Network and Computer Applications. 98 (2017) 11–26. https://doi.org/10.1016/j.jnca.2017.08.021.

[88] J. Xu, L. Xiao, Y. Li, M. Huang, Z. Zhuang, T.-H. Weng, W. Liang, NFMF: neural fusion matrix factorisation for QoS prediction in service selection, Connection Science. (2021) 1–16. https://doi.org/10.1080/09540091.2021.1889975.

[89] J. Dantas, E. Araujo, P. Maciel, R. Matos, J. Teixeira, Estimating capacity-oriented availability in cloud systems, IJCSE. 22 (2020) 466–476. https://doi.org/10.1504/IJCSE.2020.109409.

[90] J. Alonso, L. Orue-Echevarria, E. Osaba, J. López Lobo, I. Martinez, J. Diaz de Arcaya, I. Etxaniz, Optimization and Prediction Techniques for Self-Healing and Self-Learning Applications in a Trustworthy Cloud Continuum, Information. 12 (2021) 308. https://doi.org/10.3390/info12080308.

[91] J. Ward, A. Barker, Cloud cover: monitoring large-scale clouds with Varanus, Journal of Cloud Computing. 4 (2015). https://doi.org/10.1186/s13677-015-0041-9.

[92] J. Huang, D.M. Nicol, Trust mechanisms for cloud computing, Journal of Cloud Computing: Advances, Systems and Applications. Vol. 2 (2013). https://doi.org/10.1186/2192-113X-2-9.

[93] J.M. Alcaraz Calero, J. Gutiérrez Aguado, Comparative analysis of architectures for monitoring cloud computing infrastructures, Future Generation Computer Systems. Vol. 47 (2015) pp.16-30. https://doi.org/10.1016/j.future.2014.12.008.

[94] S. Al-Shammari, A. Al-Yasiri, MonSLAR: a middleware for monitoring SLA for RESTFUL services in cloud computing, in: 2015 IEEE 9th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Environments (MESOCA), IEEE, Bremen, 2015: pp. 46–50. https://doi.org/10.1109/MESOCA.2015.7328126.

[95] J. Povedano-Molina, J.M. Lopez-Vega, J.M. Lopez-Soler, A. Corradi, L. Foschini, DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant Clouds, Future Generation Computer Systems. 29 (2013) 2041–2056. https://doi.org/10.1016/j.future.2013.04.022.

[96] V. Chang, G. Wills, A model to compare cloud and non-cloud storage of Big Data, Future Generation Computer Systems. 57 (2016) 56–76. https://doi.org/10.1016/j.future.2015.10.003.

[97] S. Mittal, K.P. Joshi, C. Pearce, A. Joshi, Automatic Extraction of Metrics from SLAs for Cloud Service Management, in: 2016 IEEE International Conference on Cloud Engineering (IC2E), IEEE, Berlin, Germany, 2016: pp. 139–142. https://doi.org/10.1109/IC2E.2016.14.

[98] European Telecommunications Standards Institute, Cloud Standards Coordination - Final report, (2013).

[99] CloudStandards, (n.d.). https://cloud-standards.org/index_title_main_page/ (accessed March 9, 2022).

[100] CloudWATCH, (n.d.). http://www.cloudwatchhub.eu/ (accessed March 9, 2022).

[101] Home | StandICT.eu 2023, (n.d.). https://standict.eu/ (accessed March 9, 2022).

[102] DECIDE, D5.4 Final Advanced Cloud Service meta-Intermediator, (2019). https://www.decide-h2020.eu/sites/decide.drupal.pulsartecnalia.com/files/D5.4%20Final%20%20Advanced%20Cloud%20Service%20meta-intermediator_v1.0_20190531%28annex%29.pdf.

[103] ISO - ISO/IEC 17788:2014 - Information technology — Cloud computing — Overview and vocabulary, (n.d.). https://www.iso.org/standard/60544.html (accessed March 9, 2022).

[104] ISO - ISO/IEC 17789:2014 - Information technology — Cloud computing — Reference architecture, (n.d.). https://www.iso.org/standard/60545.html (accessed March 9, 2022).

[105] ISO - ISO/IEC 19086-2:2018 - Cloud computing — Service level agreement (SLA) framework — Part 2: Metric model, (n.d.). https://www.iso.org/standard/67546.html (accessed March 9, 2022).

[106] O. OGF, Web Services Agreement Specification (WS-Agreement), (n.d.). https://www.ogf.org/documents/GFD.192.pdf (accessed March 9, 2022).

[107] ISO - ISO/IEC 27001:2013 - Information technology — Security techniques — Information security management systems — Requirements, (n.d.). https://www.iso.org/standard/54534.html (accessed March 9, 2022).

[108] ISO/IEC 27017:2015(en), Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services, (n.d.). https://www.iso.org/obp/ui/#iso:std:iso-iec:27017:ed-1:v1:en (accessed March 9, 2022).

[109] ISO - ISO/IEC 27018:2014 - Information technology — Security techniques — Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors, (n.d.). https://www.iso.org/standard/61498.html (accessed March 9, 2022).

[110] ISO - ISO/IEC 27002:2013 - Information technology — Security techniques — Code of practice for information security controls, (n.d.). https://www.iso.org/standard/54533.html (accessed March 9, 2022).

[111] C. CSA, Privacy Level Agreement Outline for the Sale of Cloud Services in the European Union, n.d. https://downloads.cloudsecurityalliance.org/initiatives/pla/Privacy_Level_Agreement_Outline.pdf (accessed March 9, 2022).

[112] CSA, (n.d.). https://cloudsecurityalliance.org/research/cloud-controls-matrix/ (accessed March 9, 2022).

[113] Open Cloud Computing Interface | Open Standard | Open Community, (n.d.). https://occi-wg.org/ (accessed March 9, 2022).

[114] OASIS, OASIS Cloud Application Management for Platforms (CAMP), (n.d.). https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=camp.

[115] Distributed Management Task Force, Inc. (DMTF), Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol, (2012). https://www.dmtf.org/sites/default/files/standards/documents/DSP0263_2.0.0.pdf.

[116] OASIS, OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA), OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA). (n.d.). https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca.

[117] Cloud Data Management Interface (CDMI^{TM}) | SNIA, (n.d.). https://www.snia.org/cdmi (accessed March 9, 2022).

[118] European Union Agency for Cybersecurity, EUCS – Cloud Services Scheme, n.d. https://www.enisa.europa.eu/publications/eucs-cloud-service-scheme (accessed February 25, 2021).

[119] IEEE SA - IEEE 2302-2021, (n.d.). https://standards.ieee.org/ieee/2302/7056/ (accessed April 7, 2022).

[120] Home: EU Cloud CoC, (n.d.). https://eucoc.cloud/en/home (accessed April 29, 2022).

[121] Regulation (EU) 2018/1807 of the European Parliament and of the Council of 14 November 2018 on a framework for the free flow of non-personal data in the European Union (Text with EEA relevance.), 2018. http://data.europa.eu/eli/reg/2018/1807/oj/eng (accessed March 9, 2022).

[122] Cloudopting | CloudOpting Project | Fact Sheet | CIP | CORDIS | European Commission, (n.d.). https://cordis.europa.eu/project/id/621146 (accessed March 9, 2022).

[123] M. Miglierina, G.P. Gibilisco, D. Ardagna, E.D. Nitto, Model based control for multi-cloud applications (PS36), in: 2013 5th International Workshop on Modeling in Software Engineering (MiSE), 2013: pp. 37–43. https://doi.org/10.1109/MiSE.2013.6595294.

[124] N. Chondamrongkul, P. Temdee, Multi-cloud computing platform support with model-driven application runtime framework (PS14), in: 2013 13th International Symposium on Communications and Information Technologies (ISCIT), IEEE, Surat Thani, Thailand, 2013: pp. 715–719. https://doi.org/10.1109/ISCIT.2013.6645946.

[125] G. Tricomi, A. Panarello, G. Merlino, F. Longo, D. Bruneo, A. Puliafito, Orchestrated Multi-Cloud Application Deployment in OpenStack with TOSCA (PS22), in: 2017 IEEE International Conference on Smart Computing (SMARTCOMP), 2017: pp. 1–6. https://doi.org/10.1109/SMARTCOMP.2017.7947027.

[126] J. Guillén, J. Miranda, J.M. Murillo, C. Canal, Developing migratable multicloud applications based on MDE and adaptation techniques (PS35), in: Proceedings of the Second Nordic Symposium on Cloud Computing & Internet Technologies - NordiCloud '13, ACM Press, Oslo, Norway, 2013: pp. 30–37. https://doi.org/10.1145/2513534.2513541.

[127] S. Zhou, G. Chen, G. Huang, J. Shi, T. Kong, Research on multi-authority CP-ABE access control model in multicloud (PS44), China Commun. 17 (2020) 220–233. https://doi.org/10.23919/JCC.2020.08.018.

[128] K. Kritikos, P. Skrzypek, Are Cloud Modelling Languages Ready for Multi-Cloud? (PS66), in: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion - UCC '19 Companion, ACM Press, Auckland, New Zealand, 2019: pp. 51–58. https://doi.org/10.1145/3368235.3368840.

[129] E. Rios, W. Mallouli, M. Rak, V. Casola, A.M. Ortiz, SLA-Driven Monitoring of Multi-cloud Application Components Using the MUSA Framework (PS40), in: 2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW), IEEE, Nara, Japan, 2016: pp. 55–60. https://doi.org/10.1109/ICDCSW.2016.29.

[130] C. Quinton, N. Haderer, R. Rouvoy, L. Duchien, Towards multi-cloud configurations using feature models and ontologies (PS47), in: Proceedings of the 2013 International Workshop on Multi-Cloud Applications and Federated Clouds - MultiCloud '13, ACM Press, Prague, Czech Republic, 2013: p. 21. https://doi.org/10.1145/2462326.2462332.

[131] K. Kritikos, T. Kirkham, B. Kryza, P. Massonet, Towards a security-enhanced PaaS platform for multi-cloud applications (PS50), Future Generation Computer Systems. 67 (2017) 206–226. https://doi.org/10.1016/j.future.2016.10.008.

[132] G. Casale, M. Artač, W.-J. van den Heuvel, A. van Hoorn, P. Jakovits, F. Leymann, M. Long, V. Papanikolaou, D. Presenza, A. Russo, S.N. Srirama, D.A. Tamburri, M. Wurster, L. Zhu, RADON: rational decomposition and orchestration for serverless computing (PS67), SICS Softw.-Inensiv. Cyber-Phys. Syst. 35 (2020) 77–87. https://doi.org/10.1007/s00450-019-00413-w.

[133] K. Kritikos, T. Kirkham, B. Kryza, P. Massonet, Reprint of "Towards a security-enhanced PaaS platform for multi-cloud applications" (PS73), Future Generation Computer Systems. 78 (2018) 155–175. https://doi.org/10.1016/j.future.2016.11.014.

[134] B. He, J. Wang, J. Zhou, L. Li, W. Zhou, L. Zhu, M. Zhai, The Design and Implementation of Multi-Cloud Based Distributed Storage Platform with Random Linear Coding (PS76), in: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, Zhangjiajie, China, 2019: pp. 1233–1240. https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00173.

[135] K. Baby, A. Vysala, Multicloud architecture for augmenting security in clouds (PS48), in: 2015 Global Conference on Communication Technologies (GCCT), IEEE, Thuckalay, Kanya kumari district, India, 2015: pp. 474–478. https://doi.org/10.1109/GCCT.2015.7342707.

[136] R. Patel, D. Dahiya, Aggregation of cloud providers: A review of opportunities and challenges (PS43), in: International Conference on Computing, Communication & Automation, IEEE, Greater Noida, India, 2015: pp. 620–626. https://doi.org/10.1109/CCAA.2015.7148448.

[137] M. Villari, M. Fazio, S. Dustdar, O. Rana, R. Ranjan, Osmotic Computing: A New Paradigm for Edge/Cloud Integration, IEEE Cloud Comput. 3 (2016) 76–83. https://doi.org/10.1109/MCC.2016.124.

[138] M. Ciavotta, D. Ardagna, G.P. Gibilisco, A mixed integer linear programming optimization approach for multi-cloud capacity allocation (PS74), Journal of Systems and Software. 123 (2017) 64–78. https://doi.org/10.1016/j.jss.2016.10.001.

[139] V. Xhagjika, L. Navarro, V. Vlassov, Enhancing Real-Time Applications by Means of Multi-tier Cloud Federations (PS53), in: 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, Vancouver, BC, Canada, 2015: pp. 397–404. https://doi.org/10.1109/CloudCom.2015.69.

[140] Home - Docker, (n.d.). https://www.docker.com/ (accessed April 29, 2022).

[141] L. Yang, A. Humayed, F. Li, A multi-cloud based privacy-preserving data publishing scheme for the internet of things (PS63), in: Proceedings of the 32nd Annual Conference on Computer Security Applications, ACM, Los Angeles California USA, 2016: pp. 30–39. https://doi.org/10.1145/2991079.2991127.

[142] M.R. Movahedisefat, S.M. Reza Farshchi, D. Mohammadpur, Emerging Security Challenges in Cloud Computing, from Infrastructure-Based Security to Proposed Provisioned Cloud Infrastructure (PS9), in: Emerging Trends in ICT Security, Elsevier, 2014: pp. 379–393. https://doi.org/10.1016/B978-0-12-411474-6.00023-2.

[143] Puppet Labs, Get Started with DevOps: A Guide for IT Manager, n.d. https://xact.spiceworks.com/u/gen/Mar2016/get_started_with_devops_guide_for_it_managers_29cd3d57ea49e1d446882483a4ed2271.pdf (accessed March 11, 2022).

[144] DECIDE, D2.1 Requirements Specification_v1.0_20170531.pdf, n.d. https://www.decide-h2020.eu/sites/decide.drupal.pulsartecnalia.com/files/documents/D2.1%20Requirements%20Specification_v1.0_20170531.pdf (accessed June 2, 2022).

[145] Z. Xiao, I. Wijegunaratne, X. Qiang, Reflections on SOA and Microservices, in: 2016 4th International Conference on Enterprise Systems (ES), IEEE, Melbourne, Australia, 2016: pp. 60–67. https://doi.org/10.1109/ES.2016.14.

[146] Exploring the ENTIRE DevOps Toolchain for (Cloud) Teams, (n.d.). https://www.infoq.com/articles/devops-toolchain/ (accessed March 11, 2022).

[147] J. Alonso, M. Escalante, L. Farid, M.J. Lopez, L. Orue-Echevarria, S. Dutkowski, Towards Supporting the Extended DevOps Approach through Multi-cloud Architectural Patterns for Design and Pre-deployment - A Tool Supported Approach:, in: Proceedings of the 13th International Conference on Software Technologies, SCITEPRESS - Science and Technology Publications, Porto, Portugal, 2018: pp. 813–823. https://doi.org/10.5220/0006856008130823.

[148] R. Trapero, N. Suri, A Common Reference Model to describe, promote and support the uptake of SLAs, (2016). https://www.sla-ready.eu/cloud-sla-lifecycle (accessed October 16, 2020).

[149] A. Pereira, R.J. Machado, J.E. Fernandes, J. Teixeira, N. Santos, A. Lima, Using the NIST Reference Model for Refining Logical Architectures, in: B. Murgante, S. Misra, A.M.A.C. Rocha, C. Torre, J.G. Rocha, M.I. Falcão, D. Taniar, B.O. Apduhan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2014, Springer International Publishing, Cham, 2014: pp. 185–199. https://doi.org/10.1007/978-3-319-09156-3_14.

[150] Z. Ruan, D. Yang, Self-organised resource assignment for on-demand services in the cloud platform, IJCSE. 22 (2020) 62–73. https://doi.org/10.1504/IJCSE.2020.107248.

[151] E. Ataie, R. Entezari-Maleki, L. Rashidi, K. Trivedi, A. Movaghar, Hierarchical Stochastic Models for Performance, Availability, and Power Consumption Analysis of IaaS Clouds, IEEE Transactions on Cloud Computing. Vol. 7 (2017) 1039–1056. https://doi.org/10.1109/TCC.2017.2760836.

[152] DECIDE, D5.4 Final Advanced Cloud Service meta-intermediator (Annex), (2019). https://www.decide-h2020.eu/sites/decide.drupal.pulsartecnalia.com/files/D5.4%20Final%20%20Advanced%20Cloud%20Service%20meta-intermediator_v1.0_20190531%28annex%29.pdf (accessed October 16, 2020).

[153] CloudBroker GmbH – Compute-intensive applications in the cloud, (n.d.). http://cloudbroker.com/ (accessed March 11, 2022).

[154] JHipster - Full Stack Platform for the Modern Developer!, (n.d.). https://www.jhipster.tech/ (accessed March 11, 2022).

[155] SwaggerHub | API Design and Documentation with OpenAPI, (n.d.). https://swagger.io/tools/swaggerhub/?&utm_source=aw&utm_medium=ppcg&utm_campaign=SEM_SwaggerHub_PR_EMEA_ENG_EXT_Prospecting&utm_term=swagger&utm_content=511173019632&gclid=CjwKCAjwgr6TBhAGEiwA3aVuIQ2_9p9_SjExeaeq2CujxlqzBFwCmYwZhPeNAhUtr8tBYAkZDMAySBoCwggQAvD_BwE&gclsrc=aw.ds (accessed May 2, 2022).

[156] Telegraf Open Source Server Agent | InfluxDB, (n.d.). https://www.influxdata.com/time-series-platform/telegraf/ (accessed March 11, 2022).

[157] Microservices Demo: Sock Shop, (n.d.). https://microservices-demo.github.io/ (accessed March 11, 2022).

[158] DEvOps for trusted, portable and interoperable Multi-Cloud applications towards the Digital singlE market | DECIDE Project | Fact Sheet | H2020 | CORDIS | European Commission, (n.d.). https://cordis.europa.eu/project/id/731533 (accessed April 29, 2022).

[159] RIPCORD 2 Accepted for Presentation at European Society of Cardiology 2021 – AIMES – Intelligent Data Solutions, (n.d.). https://aimes.uk/ripcord-2-accepted-for-presentation-at-european-society-of-cardiology-2021/ (accessed April 29, 2022).

[160] A. Ibarra, D4.9 Final multi-cloud application monitoring_v1.0, (n.d.) 62.

[161] J.W. Walker, T.P. Bechet, Defining Effectiveness and Efficiency Measures in the Context of Human Resource Strategy, in: R.J. Niehaus, K.F. Price (Eds.), Bottom Line Results from Strategic Human Resource Planning, Springer US, Boston, MA, 1991: pp. 235–245. https://doi.org/10.1007/978-1-4757-9539-4_18.

[162] DECIDE, Deliverable D6.6 Final DECIDE Use Case Evaluation, (n.d.).

[163] A. Al-Said Ahmad, P. Andras, Measuring the Scalability of Cloud-Based Software Services, in: 2018 IEEE World Congress on Services (SERVICES), IEEE, San Francisco, CA, 2018: pp. 5–6. https://doi.org/10.1109/SERVICES.2018.00016.

[164] E. Kajan, F.-D. Dorloff, I. Bedini, eds., Handbook of Research on E-Business Standards and Protocols: Documents, Data and Advanced Web Technologies, IGI Global, 2012. https://doi.org/10.4018/978-1-4666-0146-8.

[165] L. Baresi, D.F. Mendonça, M. Garriga, S. Guinea, G. Quattrocchi, A Unified Model for the Mobile-Edge-Cloud Continuum, ACM Trans. Internet Technol. 19 (2019) 1–21. https://doi.org/10.1145/3226644.

APPENDIX A: Supplementary information

Cloud Service LifeCycle detailed analysis

This section includes the analysis of the different activities and processes of the Cloud Services lifecycle. This analysis has served as a basis for the understanding of the functional needs required by ACSmI. In this analysis Cloud Broker (CB) comparable to ACSmI.

This work was done for the Cloud for Europe project from the European Union's Seventh Framework Programme for research, technological development and demonstration and publised in the corresponding paper "Federated Cloud Service Broker (FCSB): An Advanced Cloud Service Intermediator for Public Administrations" [9].

Cloud Service Initialization

Endorse a cloud service into the broker

These are the main steps for endorsing a service from a CSP into the ACSmI:

Table 38. *Endorse a cloud service into the broker.*

Section	Description
Use Case id	Endorse a cloud service into the broker
Description	The CSP wants to registry one of its service so that the user can access the service through the broker. For that the Broker needs check its compliance against the regulatory framework.
Actors	Cloud provider (CSP), Cloud Broker (CB).
Objective	The Cloud Provider wants to include one of its services in the Cloud Broker's catalogue.
Pre-Conditions	The CSP is already registered in the broker
Process Dialog	1. The CSP requests to register a Cloud Service in the ACSmI through the GUI of the ACSMI console. 2. The required information is gathered from the ACSmI and stored in the service registry. 3. The Regulatory Framework assessment module checks if the service complies with the legislation based on the information provided and stored in the service registry. 4. The Service registry is accordingly updated with the information about the compliance of the legislation. The Regulation compliance assessment returns the result to the Service registry governance and updates the Service Registry. The CSP is informed through the CSP console if the service has been finally included in the ACSmI.

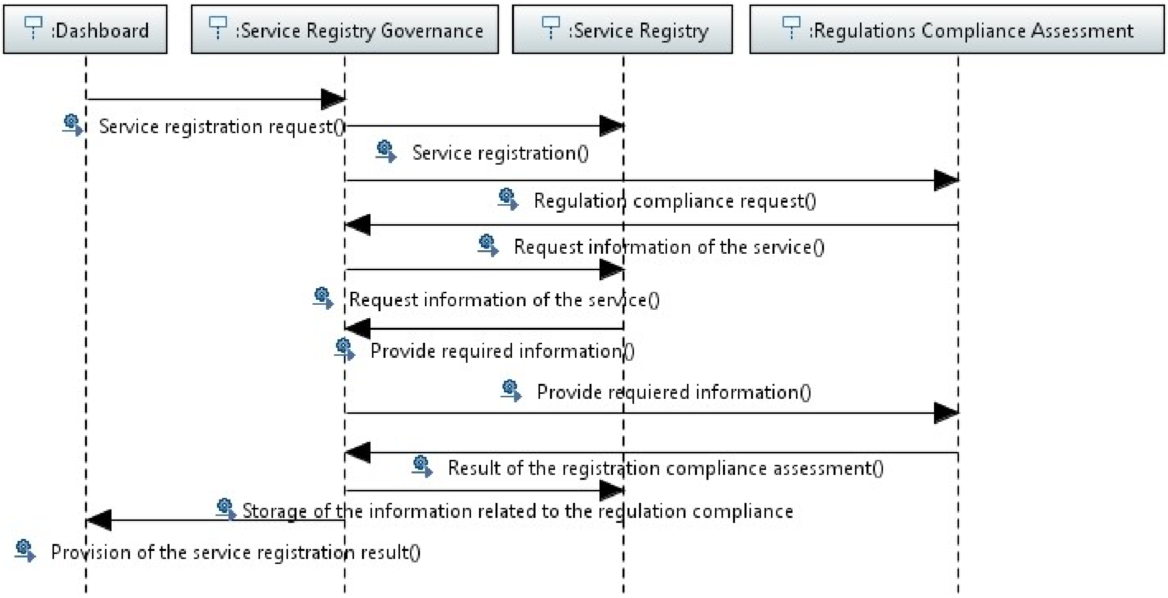

:Dashboard
:Service Registry Governance
:Service Registry
:Regulations Compliance Assessment
Service registration request()
Service registration()
Regulation compliance request()
Request information of the service()
Request information of the service()
Provide required information()
Provide requiered information()
Result of the registration compliance assessment()
Storage of the information related to the regulation compliance
Provision of the service registration result()

Figure 61. *Endorse a cloud service into the broker Sequence diagram.*

Intelligent Discovery Process

Table 39. *Intelligent Discovery Process.*

Section	Description
Use Case id	Intelligent Discovery Process
Description	A Cloud Consumer wants to discover services in the broker which fulfil a set of requirements.
Actors	Cloud provider (CSP), Cloud Broker (CB), Cloud Consumer
Objective	This process will provide a sort list of services in the Broker that fulfils (totally or partially) the requirements specified by the Cloud Consumer/PA.
Pre-Conditions	The Cloud Consumer is already registered in the Broker and is logged.
Process Dialog	1. The user enters the user console and looks for the service he wants specifying the requirements for the service. 2. With the requirements specified by the user, the Service Discovery module looks for the services fulfilling the requirements. 3. The Service Benchmarking module creates a prioritized list with the selected services/aggregation of services. 4. The user console presents the list of services provided by the service benchmarking to the user. 5. The user selects the service/aggregation of services which best suits his interests.

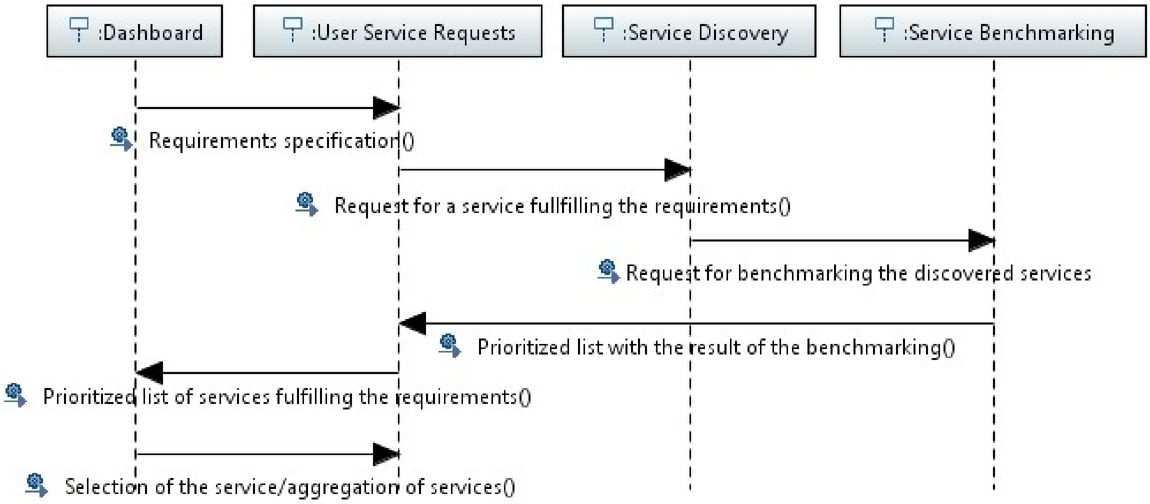

:Dashboard
:User Service Requests
:Service Discovery
:Service Benchmarking
Requirements specification()
Request for a service fullfilling the requirements()
Request for benchmarking the discovered services
Prioritized list with the result of the benchmarking()
Prioritized list of services fulfilling the requirements()
Selection of the service/aggregation of services()

Figure 62. *Intelligent discovery process Sequence diagram.*

Cloud Service contracting

Table 40. *Cloud Service contracting.*

Section	Description
Use Case id	Cloud Service contracting
Description	A Cloud Consumer wants to contract a service in the Broker.
Actors	Cloud provider (CSP), Cloud Broker (CB), Cloud Consumer
Objective	This process will consist of 1.- contract a service in the CB for a certain Cloud Consumer, 2 CB contracts service to the CSP and 3. Start the operation phase of this service.
Pre-Conditions	The Cloud services are already endorsed into the broker. The Cloud Consumer is already logged into the CSB. PA has selected the service/ aggregation of services
Process Dialog	1. The Cloud Consumer asks for contracting the selected services. 2. CB requests s additional required information (payment data/ additional information for the services configuration). 3. CB contracts a new service in the CSP. A connector (specified by the Connector manager) will contract the service if possible. Returns the data of the contract (if yes or no contract) to the CSP. 4. CSP modifies service contract registry. 5. CB contract sends the contract to the Cloud Consumer and he/she accepts the contract and service contract registry is updated. 6. CB starts operating the service launching the Service operation process 7. CB tells the Cloud Consumer that the service is in operation
Variations	The role of the CSP could be an external ACSMI

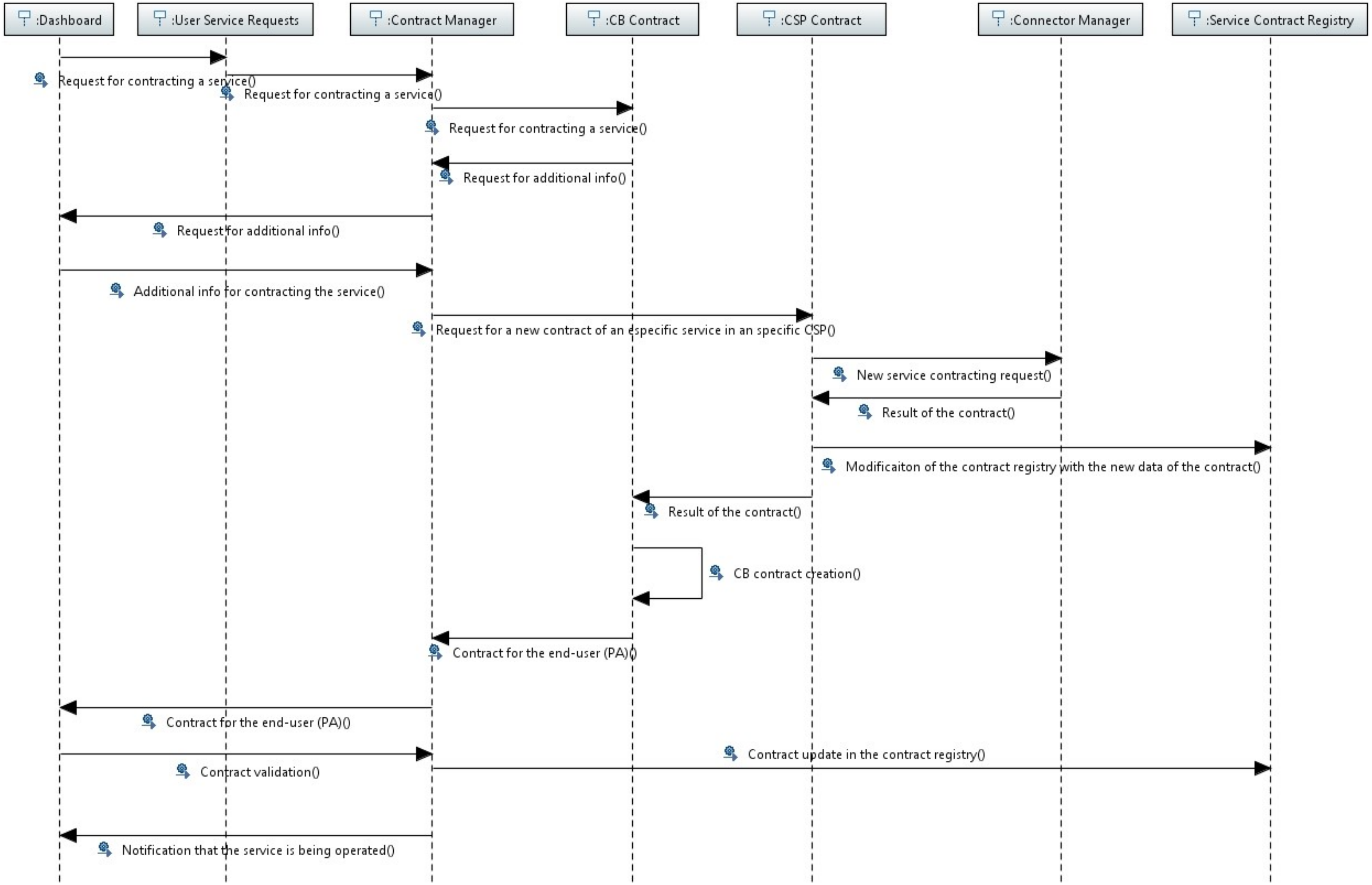

:Dashboard
:User Service Requests
:Contract Manager
:CB Contract
:CSP Contract
:Connector Manager
:Service Contract Registry
Request for contracting a service()
Request for contracting a service()
Request for contracting a service()
Request for additional info()
Request for additional info()
Additional info for contracting the service()
Request for a new contract of an especific service in an specific CSP()
New service contracting request()
Result of the contract()
Modificaiton of the contract registry with the new data of the contract()
Result of the contract()
CB contract creation()
Contract for the end-user (PA)()
Contract for the end-user (PA)()
Contract validation()
Contract update in the contract registry()
Notification that the service is being operated()

Figure 63. *Federated service contracting Sequence diagram*

CSLA provision

Table 41. *CSLA creation.*

Section	Description
Use Case id	CSLA creation
Description	CB provides the CSLA to the user of the service. This CSLA is based will be based on the CSLA of the CSP/s.
Actors	Cloud Broker (CB), Cloud Consumer
Objective	This process will provide the user the final CSLA of the service (or aggregation of services).
Pre-Conditions	The base CSLA (from the CSP) is non-repudiable (signed). The service contracting process has been launched.
Process Dialog	1. CB looks for the CSLA in machine readable format in the Service registry. 2. Based on the CSP CSLA (service registry), The CSLA definition creates the CSLA offered by the CB based on the CSLAs and the non-functional characteristics types. It also considers where the aggregation is done (broker, CP, etc.). 3. The Service management: CSLA definition registers the CB CSLA in the service registry.
Variations	If the CSLA (service registry) is not in machine readable format, CSLA definition module transforms the CSP CSLA into machine readable and updates the Service registry

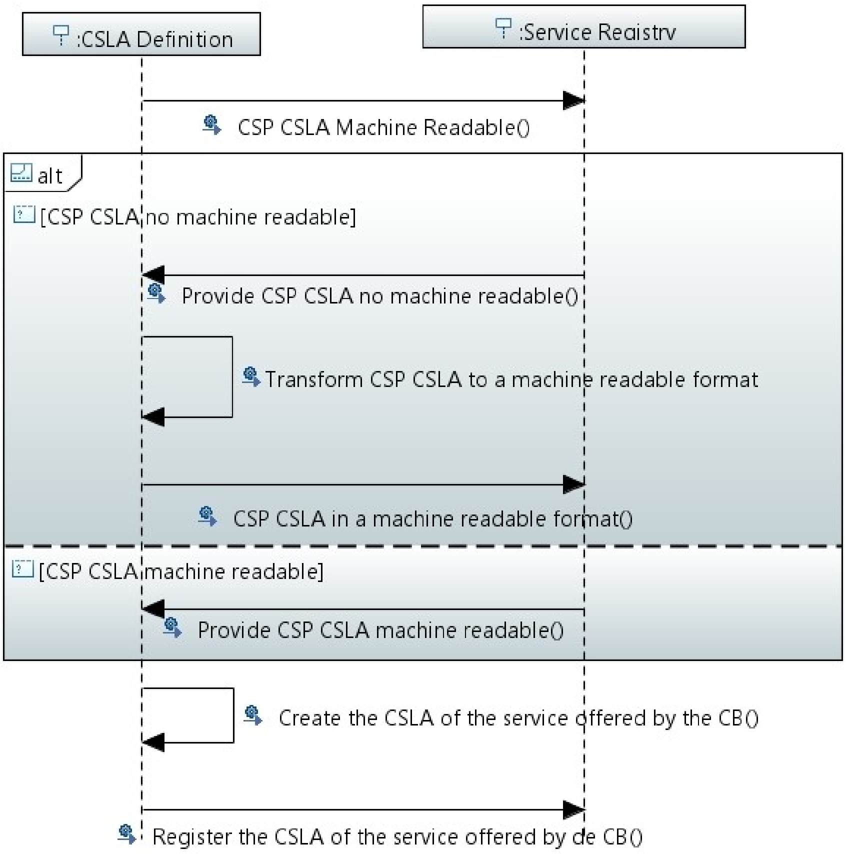

:CSLA Definition
:Service Registry
CSP CSLA Machine Readable()
alt
[CSP CSLA no machine readable]
Provide CSP CSLA no machine readable()
Transform CSP CSLA to a machine readable format
CSP CSLA in a machine readable format()
[CSP CSLA machine readable]
Provide CSP CSLA machine readable()
Create the CSLA of the service offered by the CB()
Register the CSLA of the service offered by de CB()

Figure 64. *CSLA Provision Sequence diagram.*

Cloud Service operation

CSLA Service Monitoring

Table 42. *CB CSLA Service Monitoring.*

Section	Description
Use Case id	CB CSLA Service Monitoring
Description	The CB monitors the CSLA and checks its compliance, so that the Cloud Consumer/PAs is informed.
Actors	Cloud Broker (CB),
Objective	The objective of this process is to monitor the CSLA of the service offered to the Cloud Consumer/PA and detect incompliances of the CSLA during the operation
Pre-Conditions	The service must be in operation
Process Dialog	This is an iterative process. 1. The CSLA Monitoring from the CSLA Management gathers the required information from the metrics of the CSLA. 2. The CSLA Monitoring requests the metrics to the Metering Service Management and computes the information to get the CSLA values for the CSLA terms (based on the gathered basic metrics). 3. The CSLA assessment Service Management: Service operation management: CSLA Assessment) checks the compliance of the agreed CSLO. 4. If any of the terms is not accomplished, the CSLA assessment module detects which CSP is the origin of the not compliance. 5. CB alerts the CSP of the not compliance. 6. The not compliances are registered in the Service registry by CSLA assessment.
Variations	As future extension in case of the CB provides its own services, • If the not compliance is originated in the CB, the Administration Console: notification and alert manager informs the CB through the Administration Console: Broker monitoring and the user through the User Console: notification and alert manager.
Post-Conditions	This process will cause: • Contract termination • Compensations • No action

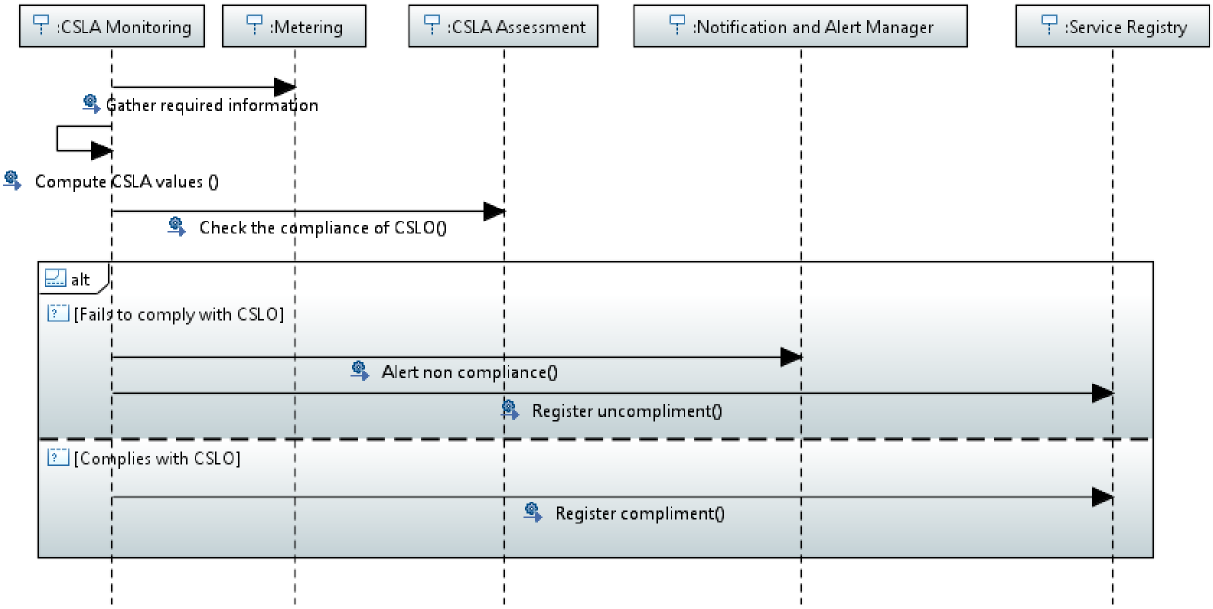


Figure 65. *CB CSLA Service Monitoring Sequence Diagram.*

Data Migration/portability

Table 43. Data Migration/portability.

Section	**Description**
Use Case id	Data Migration/portability
Description	The CB migrates the information/data form one CSP/service to another CSP/service.
Actors	Cloud Broker (CB), Cloud Consumer, Cloud Provider (CP).
Objective	The CB migrates the information from one Cloud Service to another Cloud Service.
Pre-Conditions	The Cloud Consumer wants to change the Cloud Service due to not compliance of the CSLA, legislation or changes in the NFR, or there is a withdrawal of a service
Process Dialog	1. A request for migration of the service 2. Dashboard launches Intelligent Discovery Process. 3. When the new service is in operation, Portability module notifies to the Connector Management the APIs that needs to use (we will try to use the ISO17826 standard-CDMI). 4. Portability module performs the migration from one Cloud Service to another Cloud Service through these connectors. 5. CB notifies the migration to the new service to the PA/Cloud Consumer through the Notification and alert management. 6. Service Contract Termination process is launched.

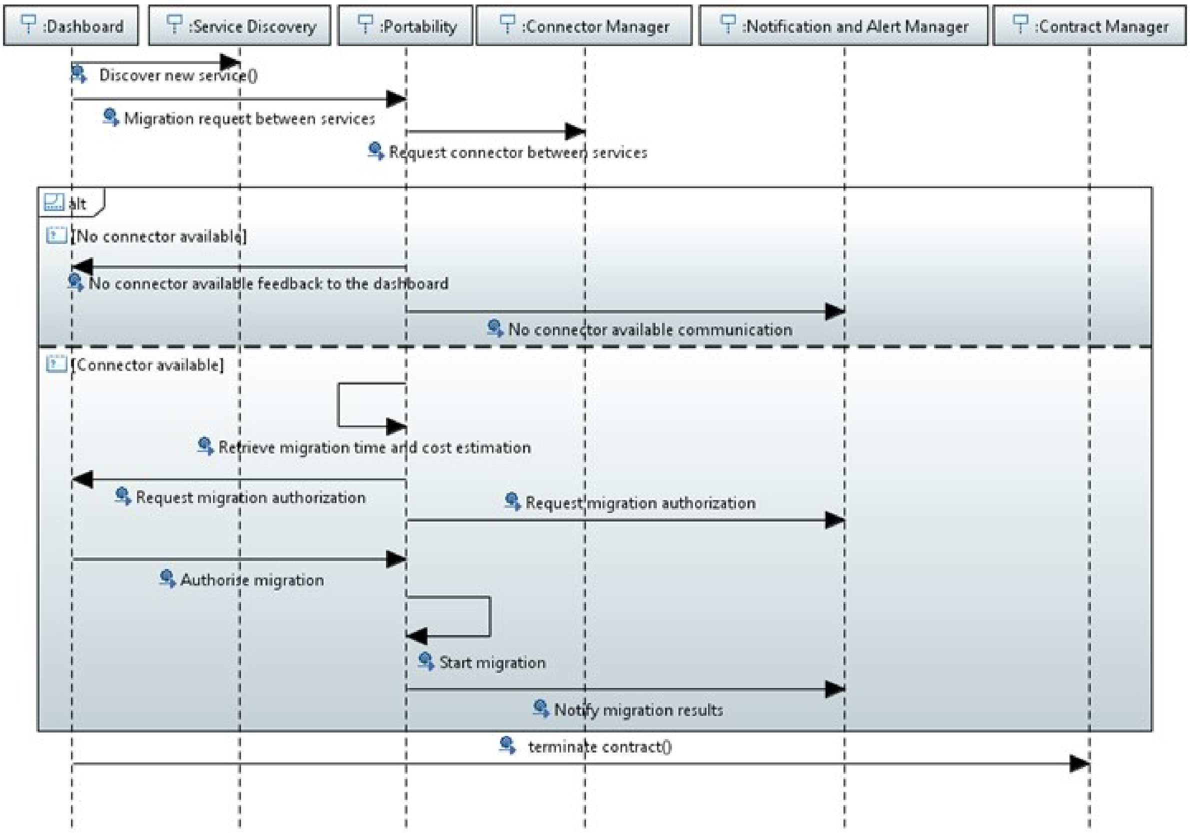

:Dashboard
:Service Discovery
:Portability
:Connector Manager
:Notification and Alert Manager
:Contract Manager
Discover new service()
Migration request between services
Request connector between services
alt
[No connector available]
No connector available feedback to the dashboard
No connector available communication
[Connector available]
Retrieve migration time and cost estimation
Request migration authorization
Request migration authorization
Authorise migration
Start migration
Notify migration results
terminate contract()

Figure 66. *Data migration Sequence diagram.*

Service metering

The objective of these processes is to gather measurements for allowing the broker to:

- Bill the user
- Pay the CSP
- Guarantee the service (CSLA)
- Enhance the service

Measurements enquiry

Table 44. *Measurement's enquiry.*

Section	Description
Use Case id	Measurements enquiry
Description	The metering component wants to recover some measurements from a cloud service.
Actors	Cloud Broker (CB),
Objective	The objective of this process is to recover the measurements from a cloud service.
Pre-Conditions	The metering mechanism must exist and must be active. A requirement to get the measurements must exist.

Section	Description
Process Dialog	1. Metering sets up the filter for the enquiry (CSLA monitoring or Accounting management) 2. The enquiry is made in the registry log 3. The set of the enquired measurements are returned.

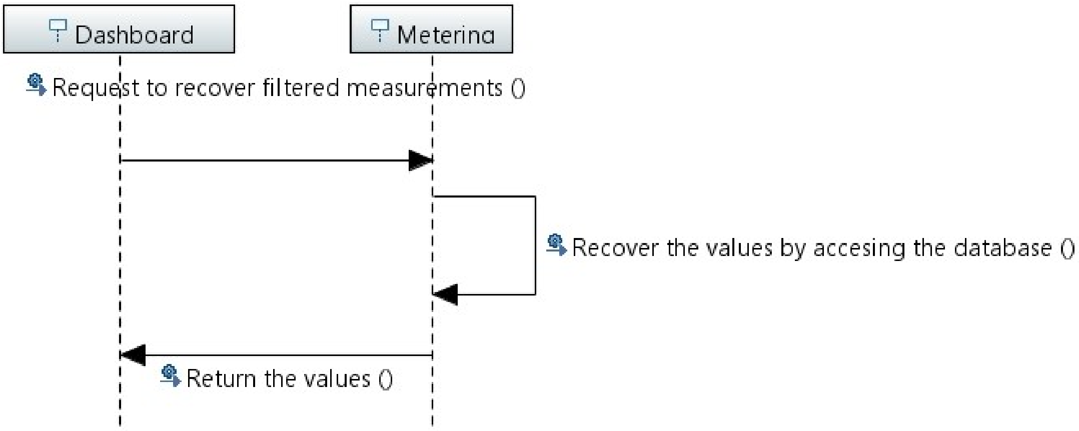

Dashboard
Metering
Request to recover filtered measurements ()
Recover the values by accesing the database ()
Return the values ()

Figure 67. *Measurements Enquiry Sequence diagram.*

Billing the users

Table 45. *Billing user.*

Section	Description
Use Case id	Billing user
Description	The CB calculates the payments to be charged to the Cloud Consumer for the services consumed.
Actors	Cloud Broker (CB), Cloud consumer.
Objective	The objective of this process is to calculate the cost made by the PA for the use of the CB services.
Pre-Conditions	A contract with a concrete CB must exist, and the contract signed stored. Billable concepts exist.
Process Dialog	1. Accounting module gathers the required information from the billable concept of the contract. 2. The Financial management: accounting request the metrics associated to the billable concepts to the Metering Service Management: Service operation management; metering) and computes the information to get the Billable values for the billable concepts (based on the gathered basic metrics). 3. Billing module generates a bill (take into account if there are some potential compensations)

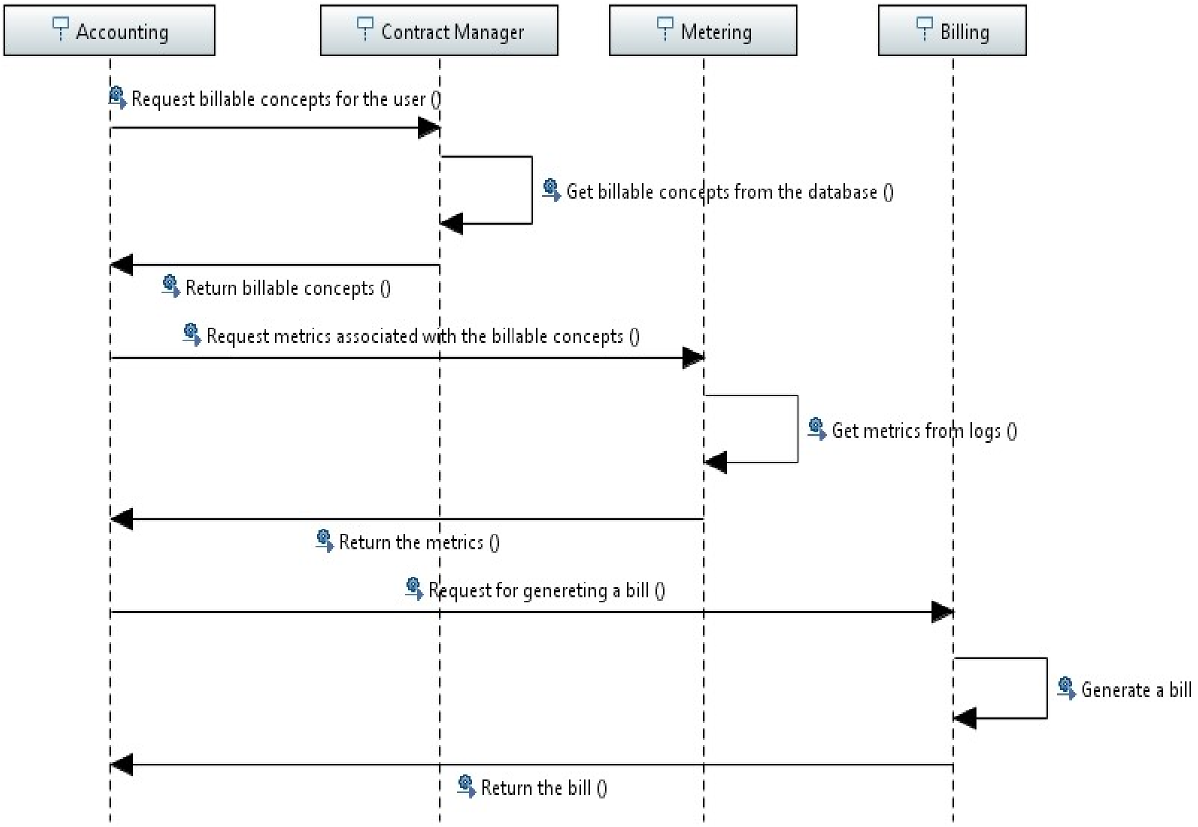

Accounting
Contract Manager
Metering
Billing
Request billable concepts for the user ()
Get billable concepts from the database ()
Return billable concepts ()
Request metrics associated with the billable concepts ()
Get metrics from logs ()
Return the metrics ()
Request for genereting a bill ()
Generate a bill
Return the bill ()

Figure 68. *Billing the users Sequence diagram.*

CP costs estimation

Table 46. *CP costs estimation.*

Section	Description
Use Case id	CP costs estimation
Description	The CB calculates the payments to be done to the CP for the services used/consumed.
Actors	Cloud Broker (CB)
Objective	The objective of this process is to calculate the costs made by the CB for the use of the CP services.
Pre-Conditions	A contract with a concrete CP must exist, so the CP has to be registered in the CB and the contract signed stored. A receipt for the billable services/concepts shall exist.
Process Dialog	1. Accounting module checks the contract., 2. Accounting module asks metrics to the Metering 3. Accounting module computes the information to get the Billable values for the CSP 4. Accounting module notifies to the accounting department.

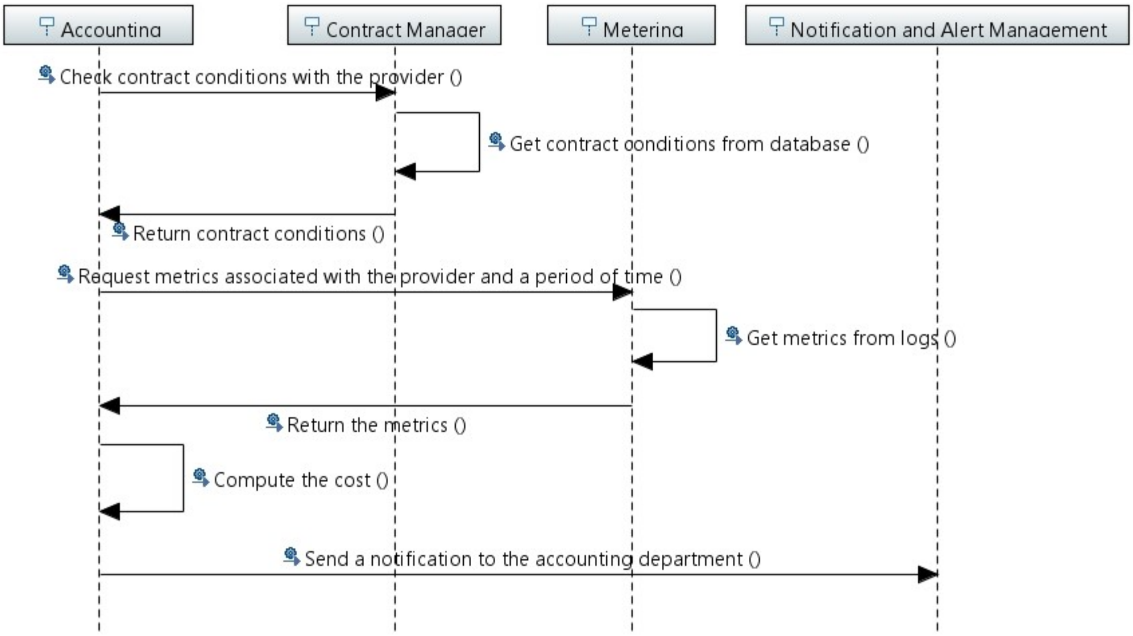

Accounting
Contract Manager
Metering
Notification and Alert Management
Check contract conditions with the provider ()
Get contract conditions from database ()
Return contract conditions ()
Request metrics associated with the provider and a period of time ()
Get metrics from logs ()
Return the metrics ()
Compute the cost ()
Send a notification to the accounting department ()

Figure 69. *CP costs estimation Sequence diagram*

Cloud Service termination

Service withdrawal

Table 47. *Service withdrawal from the catalogue.*

Section	Description
Use Case id	Service withdrawal from the catalogue
Description	The service Broker deletes a service form the catalogue.
Actors	Cloud Broker (CB), Cloud Consumer, CSP
Objective	The objective of this process is to remove a service from the service registry so it cannot be used in the discovery process.
Pre-Conditions	The service must exist in the registry. The decision of removing the service from the service registry should come from the broker administrator (repeatedly CSLA violation), or from the CSP itself.
Process Dialog	1. Check for the active contract associated to a service 2. If there is any contract associated to a service → • Check if the contract includes “withdrawal protection” Launch the withdrawal protection service • It is notified to the user that the service is going to be de-activated → The Intelligent Service Discovery process is launched to offer an alternative • If a new service is selected the Service Contracting process and Data portability processes are launched • If no service is selected the Service Termination process is launched If a backup service exists, the CB informs the user about the time that this back up service is going to be active

Section	Description
	3. The CSP is informed about the service termination and terminates it following the corresponding procedure The service registry is updated (service inactive) and evidences about the actions performed for the service termination are kept

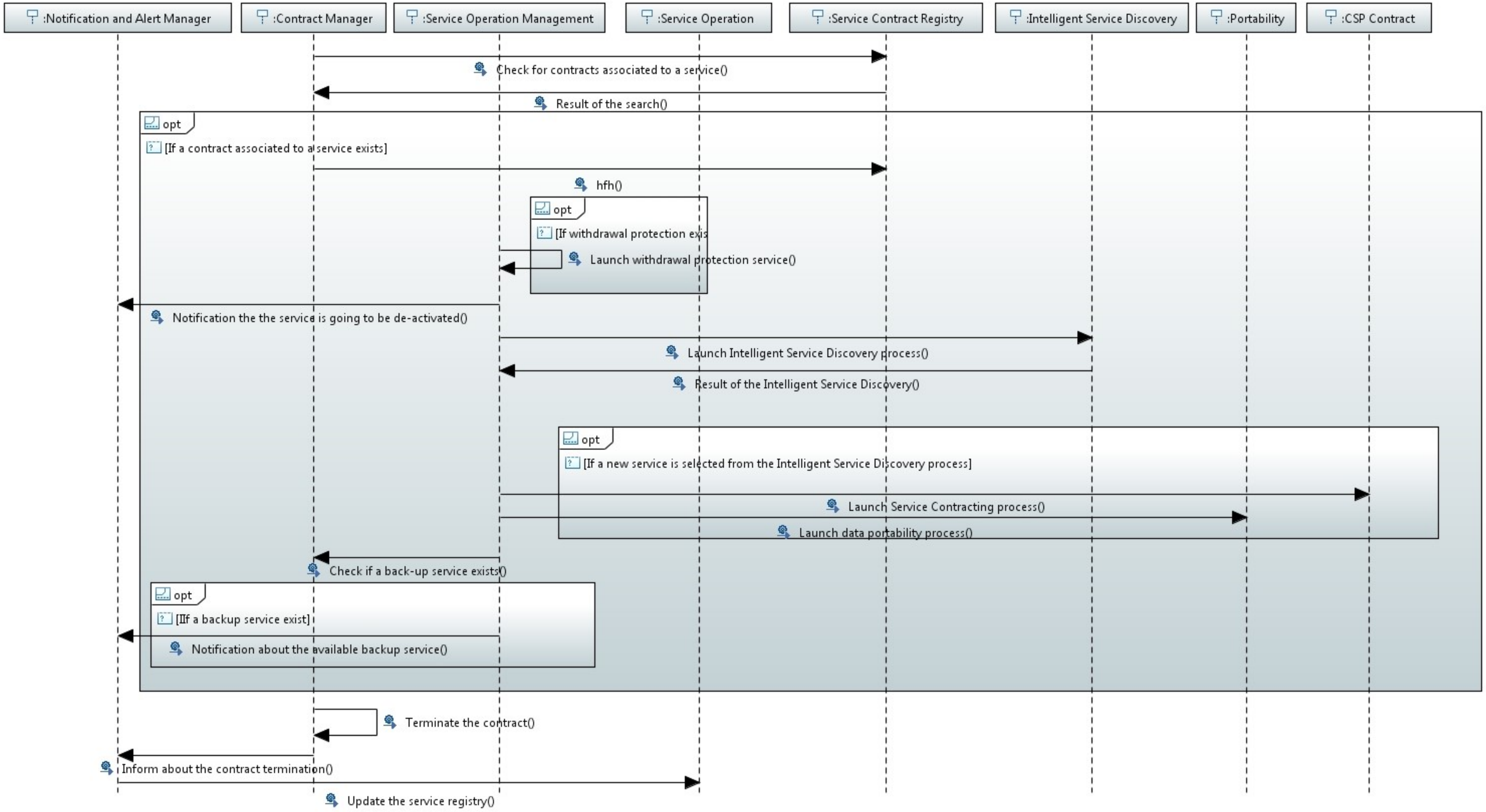

:Notification and Alert Manager
:Contract Manager
:Service Operation Management
:Service Operation
:Service Contract Registry
:Intelligent Service Discovery
:Portability
:CSP Contract
Check for contracts associated to a service()
Result of the search()
opt
[If a contract associated to a service exists]
hfh()
opt
[If withdrawal protection exis
Launch withdrawal protection service()
Notification the the service is going to be de-activated()
Launch Intelligent Service Discovery process()
Result of the Intelligent Service Discovery()
opt
[If a new service is selected from the Intelligent Service Discovery process]
Launch Service Contracting process()
Launch data portability process()
Check if a back-up service exists()
opt
[IIf a backup service exist]
Notification about the available backup service()
Terminate the contract()
Inform about the contract termination()
Update the service registry()

Figure 70. *Service Withdrawal Sequence diagram.*

Service contract termination

Table 48. *Service contract termination.*

Section	Description
Use Case id	Service contract termination
Description	The Cloud Consumer terminates a contract associated to a service offered by the Broker.
Actors	Cloud Broker (CB), Cloud Consumer
Objective	The objective of this process is to terminate a contract of a service between the Cloud Broker and the PA.
Pre-Conditions	The service must be contracted by the user The service termination request must exist.
Process Dialog	1. The request for service termination is gathered with the required information (contract id and service id). It is notified to the user the implications of the service termination (information loss, migration possibility, costs, etc.) and re-confirmation for the service termination is asked 2. The service offered by the CB is stopped Contract termination report is sent to the user 3. CB terminates the contract with the CSP 4. CSP is informed about the service contract termination and terminates it following the corresponding procedure 5. The service registry is updated (service inactive) and evidences about the actions performed for the service termination are kept

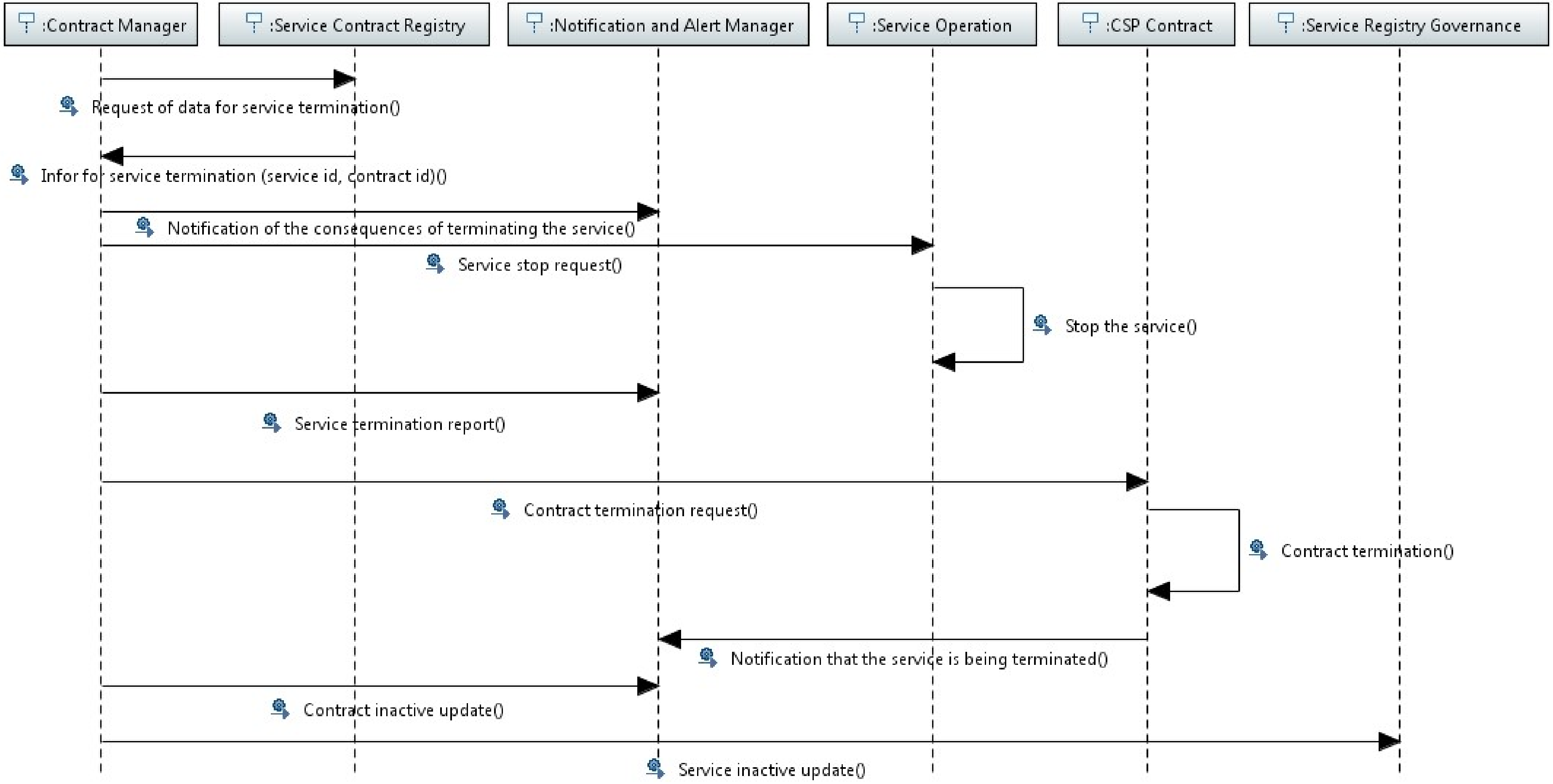
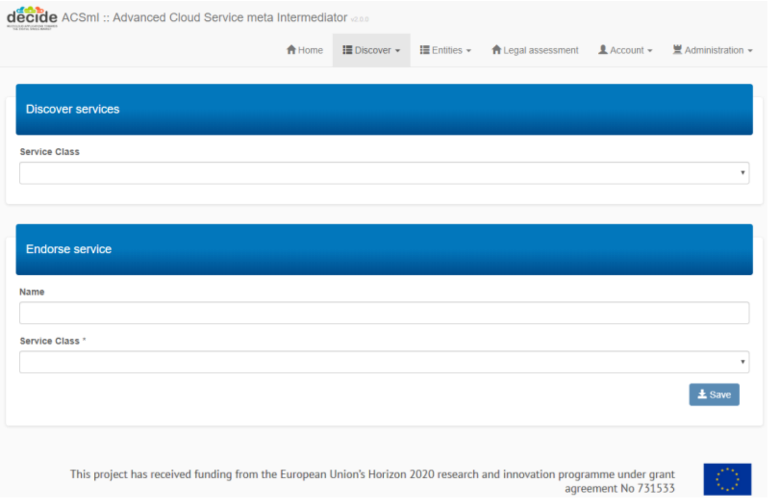

:Contract Manager
:Service Contract Registry
:Notification and Alert Manager
:Service Operation
:CSP Contract
:Service Registry Governance
Request of data for service termination()
Infor for service termination (service id, contract id)()
Notification of the consequences of terminating the service()
Service stop request()
Stop the service()
Service termination report()
Contract termination request()
Contract termination()
Notification that the service is being terminated()
Contract inactive update()
Service inactive update()

Figure 71. *Service contract termination Sequence diagram*

APPENDIX B: User Manual examples

In this section, the screenshots of some of the key functionalities of ACSmI are presented.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731533

Figure 72. *Discover and Endorse services functionalities – Admin view. Source: Author's own contribution*

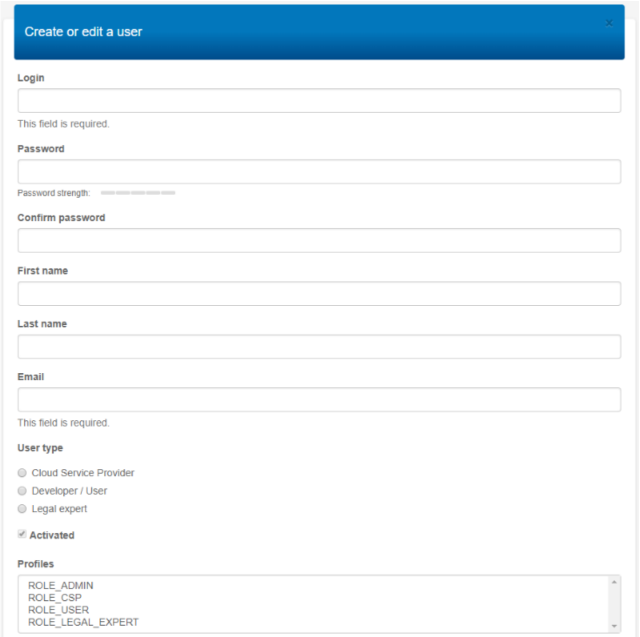
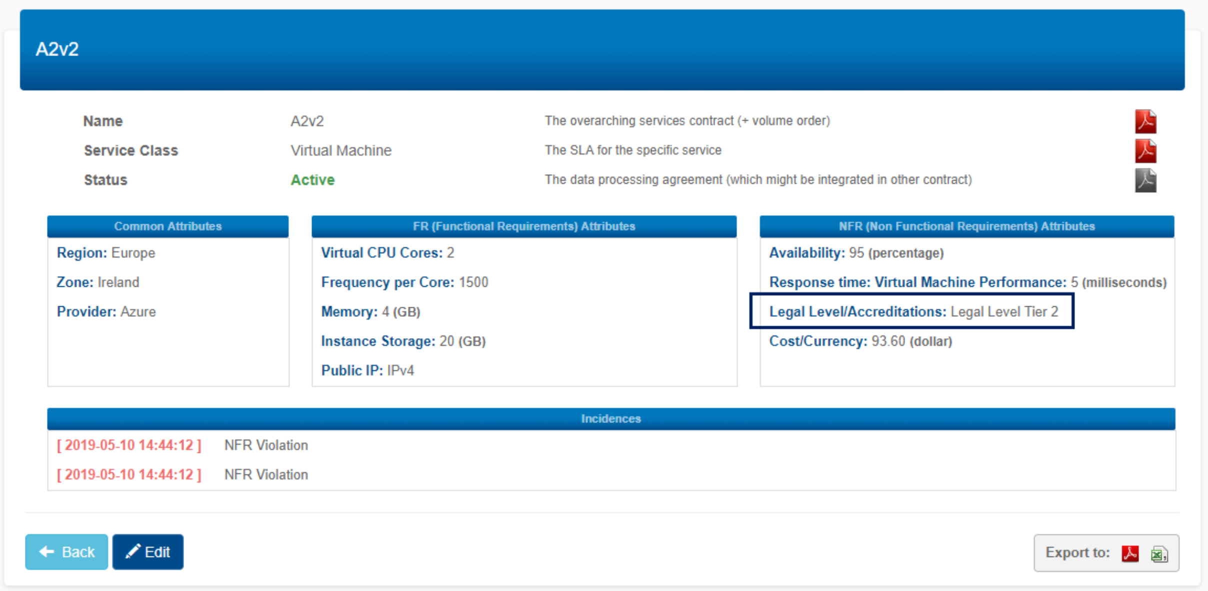


Figure 73. *Create or edit a user. Source: Author's own contribution*



A2v2

Name	A2v2	The overarching services contract (+ volume order)
Service Class	Virtual Machine	The SLA for the specific service
Status	Active	The data processing agreement (which might be integrated in other contract)

Common Attributes

Region: Europe
Zone: Ireland
Provider: Azure

FR (Functional Requirements) Attributes

Virtual CPU Cores: 2
Frequency per Core: 1500
Memory: 4 (GB)
Instance Storage: 20 (GB)
Public IP: IPv4

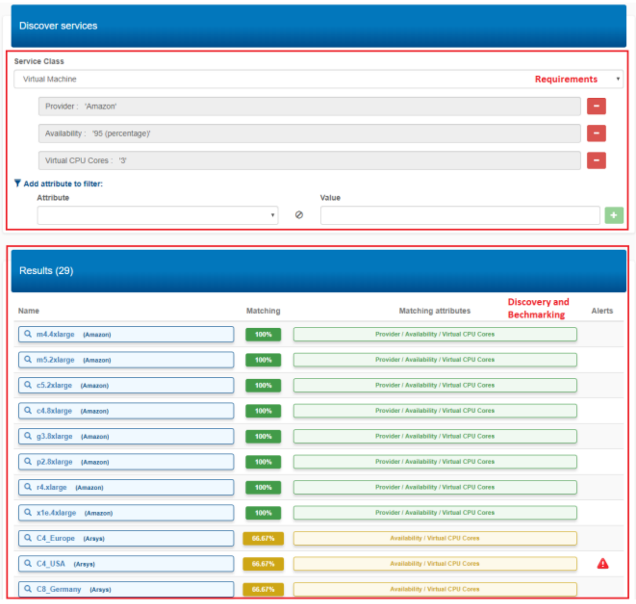
NFR (Non Functional Requirements) Attributes

Availability: 95 (percentage)
Response time: Virtual Machine Performance: 5 (milliseconds)
Legal Level/Accreditations: Legal Level Tier 2
Cost/Currency: 93.60 (dollar)

Incidences

[2019-05-10 14:44:12]	NFR Violation
[2019-05-10 14:44:12]	NFR Violation

Figure 74. *Cloud service with a legal level assigned. Source: Author's own contribution*



Discover services

Service Class
Virtual Machine

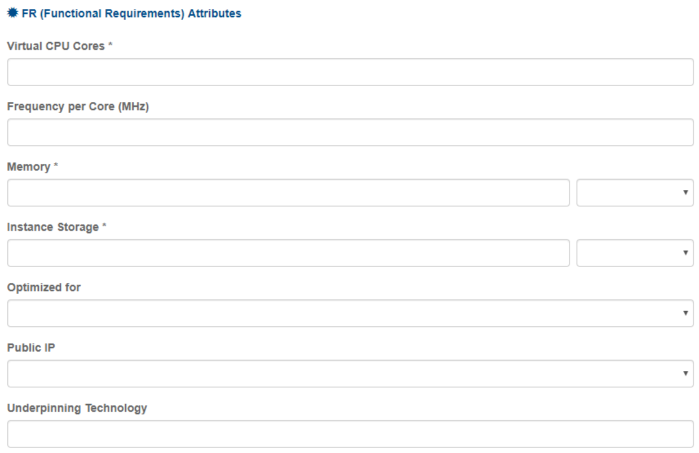
Provider : 'Amazon'
Availability : '95 (percentage)'
Virtual CPU Cores : '3'

Add attribute to filter:
Attribute
Value

Results (29)

Name	Matching	Matching attributes	Alerts
m4.4xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
m5.2xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
c5.2xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
c4.8xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
g3.8xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
p2.8xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
r4.xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
x1e.4xlarge (Amazon)	100%	Provider / Availability / Virtual CPU Cores	
C4_Europe (Arsys)	66.67%	Availability / Virtual CPU Cores	
C4_USA (Arsys)	66.67%	Availability / Virtual CPU Cores	⚠
C8_Germany (Arsys)	66.67%	Availability / Virtual CPU Cores	

Figure 75. *Service discovery and benchmarking. Source: Author's own contribution*



FR (Functional Requirements) Attributes

Virtual CPU Cores *

Frequency per Core (MHz)

Memory *

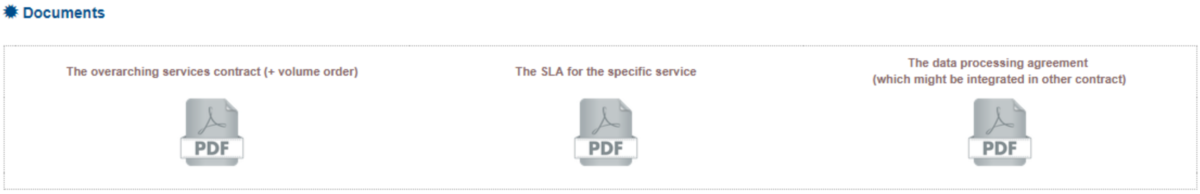
Instance Storage *

Optimized for

Public IP

Underpinning Technology

Figure 76. *Service endorsement (Functional requirements). Source: Author's own contribution*



Documents

The overarching services contract (+ volume order) — PDF

The SLA for the specific service — PDF

The data processing agreement
(which might be integrated in other contract) — PDF

Figure 77. *Legal information endorsement. Source: Author's own contribution*

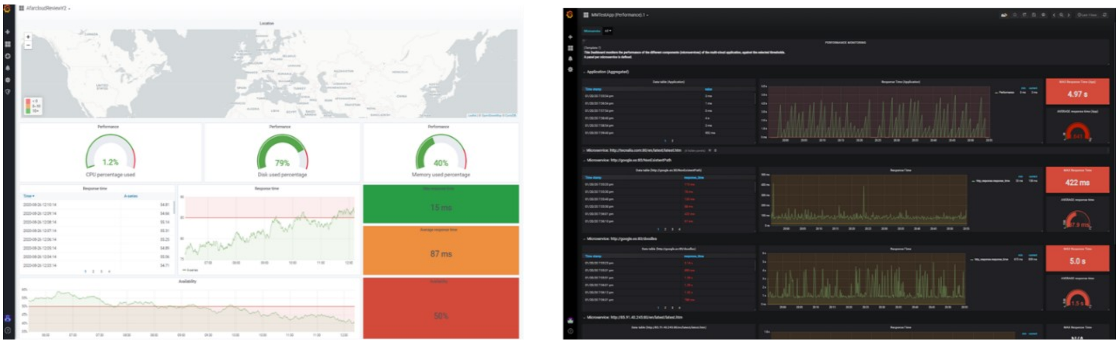
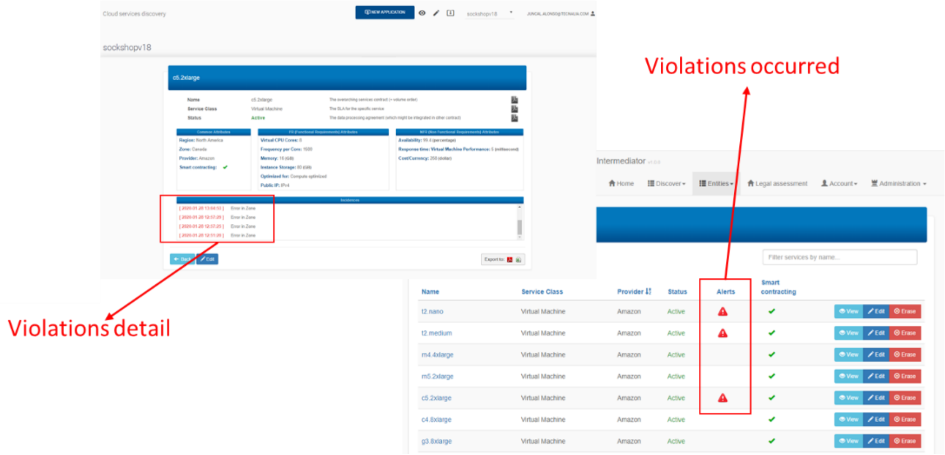


Figure 78. *Example of the monitoring information gathered by ACSml monitoring. Source: Author's own contribution*



Cloud services discovery

sockshopv18

c5.2xlarge

Name	c5.2xlarge	The overarching services contract (+ volume order)	
Service Class	Virtual Machine	The SLA for the specific service	
Status	Active	The data processing agreement (which might be integrated in other contract)	

Common Attributes

Region: North America
Zone: Canada
Provider: Amazon
Smart contracting: ✓

FR (Functional Requirements) Attributes

Virtual CPU Cores: 8
Frequency per Core: 1500
Memory: 16 (GB)
Instance Storage: 80 (GB)
Optimized for: Compute optimized
Public IP: IPv4

NFR (Non Functional Requirements) Attributes

Availability: 99.4 (percentage)
Response time: Virtual Machine Performance: 5 (millisecond)
Cost/Currency: 268 (dollar)

Incidences

[2020-01-28 13:04:53] Error in Zone
[2020-01-28 12:57:29] Error in Zone
[2020-01-28 12:57:25] Error in Zone
[2020-01-28 12:51:29] Error in Zone

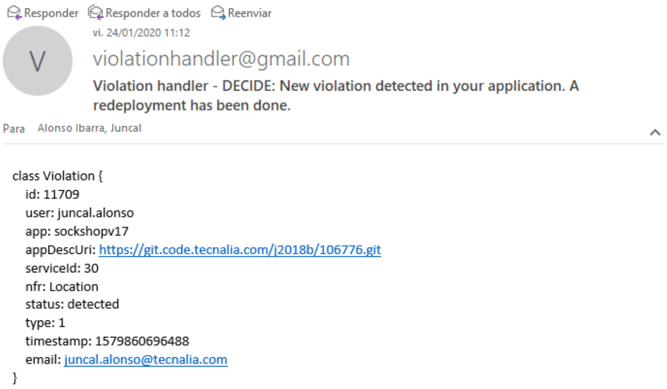
Violations occurred

Violations detail

Intermediator v1.0.0

Name	Service Class	Provider	Status	Alerts	Smart contracting	
t2.nano	Virtual Machine	Amazon	Active	⚠	✓	View Edit Erase
t2.medium	Virtual Machine	Amazon	Active	⚠	✓	View Edit Erase
m4.4xlarge	Virtual Machine	Amazon	Active		✓	View Edit Erase
m5.2xlarge	Virtual Machine	Amazon	Active		✓	View Edit Erase
c5.2xlarge	Virtual Machine	Amazon	Active	⚠	✓	View Edit Erase
c4.8xlarge	Virtual Machine	Amazon	Active		✓	View Edit Erase
g3.8xlarge	Virtual Machine	Amazon	Active		✓	View Edit Erase

Figure 79. *Information about the violations detected registered in ACSml. Source: Author's own contribution*



vi. 24/01/2020 11:12

violationhandler@gmail.com

Violation handler - DECIDE: New violation detected in your application. A redeployment has been done.

Para Alonso Ibarra, Juncal

```
class Violation {
   id: 11709
   user: juncal.alonso
   app: sockshopv17
   appDescUri: https://git.code.tecnalia.com/j2018b/106776.git
   serviceId: 30
   nfr: Location
   status: detected
   type: 1
   timestamp: 1579860696488
   email: juncal.alonso@tecnalia.com
}
```

Figure 80. *Example of an email sent to the user by the ACSmI monitoring when a violation of the NFR location is detected. Source: Author's own contribution*