An Integrated Framework for the Methodological Assurance of Security and Privacy in the Development and Operation of MultiCloud Applications

THESIS

Electronics and Telecommunications Engineering Doctorate

at the

UNIVERSITY OF THE BASQUE COUNTRY



Bilbao, 2020

Abstract

Cloud Computing market forecasts and technology trends confirm that Cloud is an Information Technology (IT) disrupting phenomena. However, security, privacy and data protection continue to be major barriers to Cloud adoption. The users' concerns on security and privacy of Cloud systems strive from the lack of trust, visibility and auditability of the security and privacy controls the Cloud providers offer in their services.

There are strong initiatives and recent standards at European and International level aiming to solve the issues of end-user trust in Cloud as well as transparency in Cloud offerings. They are paving the path towards trustworthy and certified Cloud services. Moreover, compliance with the new GDPR is an urgent necessity for Cloud consumers and providers acting as personal data processors or controllers because of the need to perform privacy risks assessments of their systems.

In recent years, the number of companies world-wide adopting multiCloud architectures in their business strategies has grown significantly. However, cost optimisation and increased competitiveness of companies exploiting multiCloud will only be possible when they are able to leverage multiple cloud offerings, while mastering both the complexity of multiple cloud provider management as well as security strategies for ensuring the protection against the higher exposure to attacks that multiCloud brings. To this end, it is necessary to consider not only functionality and business aspects of the multiCloud services, but security and privacy aspects as well.

In this context, the importance of tackling holistic security and privacy assurance of Cloud and Cloud-based IT systems is clear. Furthermore, there is a need to follow a systematic approach to cyber risk management in multiCloud that addresses both security and privacy threats. This is even more challenging in multiCloud systems because of the need of assessing not only system components' own risks but also those of the Cloud providers of outsourced components.

Fundamental research questions arise about how to design multiCloud applications taking into account security and privacy requirements to protect the system from potential risks and about how to decide which security and privacy protections to include in the system. In addition, solutions are needed to overcome the difficulties in assuring security and privacy properties defined at design time still hold all along the system life-cycle, from development to operation.

In this Thesis an innovative DevOps integrated methodology and framework are presented, which help to rationalise and systematise security and privacy analyses in multiCloud to enable an informed decision-process for risk-cost balanced selection of the protections of the system components and the protections to request from Cloud Service Providers used.

The focus of the work is on the Development phase of the analysis and creation of multiCloud applications. The main contributions of this Thesis for multiCloud applications are four: i) The integrated DevOps methodology for security and privacy assurance; and its integrating parts: ii) a security and privacy requirements modelling language, iii) a continuous risk assessment methodology and its complementary risk-based optimisation of defences, and iv) a Security and Privacy Service Level Agreement Composition method.

The integrated DevOps methodology and its integrating Development methods have been validated in the case study of a real multiCloud application in the eHealth domain. The validation confirmed the feasibility and benefits of the solution with regards to the rationalisation and systematisation of security and privacy assurance in multiCloud systems.

Laburpena

Hodei Konputazioaren merkatu iragarpenek eta teknologia joerek baieztatzen dute Hodeia Informazio Teknologien (IT) fenomeno disruptiboa dela. Hala ere, segurtasuna, pribatutasuna eta datuen babesa oztopo handiak izaten jarraitzen dute Hodeiaren harrerako. Erabiltzaileek Hodei sistemen segurtasunari eta pribatutasunari buruz dituzten kezkak Hodei hornitzaileek haien zerbitzuetan eskaintzen dituzten pribatutasun eta segurtasun kontrolen konfiantza, gardentasun eta ikuskagarritasun faltan oinarritzen dira.

Europako eta Nazioarteko mailan ekimen sendoak eta estandar berriak daude azken erabiltzailearen Hodeiarekiko konfiantza eta Hodei eskaintzetako gardentasuna konpontzeko. Hodei zerbitzu fidagarriak eta ziurtatuak lortzeko bidea zabaltzen ari dira. Gainera, Europako Datu Babeseko Arau Orokor berria betetzea premiazkoa da Hodei kontsumitzaileentzat eta datu pertsonalen prozesatzaile edo kontrolatzaile gisa jarduten duten hornitzaileentzat pribatutasun arriskuen ebaluazioak egin behar dituztelako.

Azken urteotan, beren negozio estrategietan multiHodei arkitekturak erabiltzen dituzten enpresa kopurua asko hazten ari da mundu osoan. Hala ere, multiHodei erabiltzen duten enpresen kostu optimizazioa eta lehiakortasun hobekuntza bakarrik lortu ahal izango da Hodei eskaintza anitzez baliatzeko gai direnean, Hodei hornitzaile ezberdinen kudeaketaren zailtasuna eta baita multiHodei dakarren erasoekiko esposizio handiagotik babesteko segurtasun estrategiak menderatzen dituztenean. Horretarako, multiCloud zerbitzuetako funtzionaltasun eta negozio alderdiak ez ezik, segurtasun eta pribatutasun alderdiak ere kontuan hartu behar dira.

Testuinguru honetan, Hodei eta Hodeian oinarritutako IT sistemen segurtasun eta pribatutasun segurtatze holistikoari aurre egiteko garrantzia garbia da. Bestalde, multiHodei sistemetan, ziber-arriskuen kudeaketa sistematikoari ekin behar zaio segurtasun eta pribatutasun-mehatxuei aurre egiteko. Hauxe are zailagoa da multiCloud sistemetan, sistema osagaien arriskuak ez ezik, baita kanpoko osagaien Hodei hornitzaileenak ere ebaluatu behar baitira.

Oinarrizko ikerketa galderak sortzen dira multiHodei aplikazioei buruz: nola diseinatu sistema arriskuengandik babesteko segurtasun- eta pribatutasun-betekizunak kontutan hartuz, eta baita nola erabaki zein segurtasun eta pribatutasun babesak hartu behar diren sisteman. Gainera, diseinuan zehaztutako segurtasun eta pribatutasun propietateak bermatzeko zailtasunak gainditzeko konponbideak behar dira sistema guztiaren bizitza-zikloan zehar, garapenetik operaziora.

Tesi honetan DevOps metodologia integratu bat eta esparru berritzaile bat aurkezten dira, multiHodeiean segurtasun- eta pribatutasun-analisiak arrazionalizatzen eta sistematizatzen laguntzen dutena, eta sistemaren osagaien babesak eta erabilitako Hodei hornitzaileei eskatu beharreko babesak erabakitzeko prozesua informatua izan dadila ahalbidetzen dutena, arrisku-kostu oreka mantenduz.

Lanaren ardatza multiHodei aplikazioen analisia eta sorreraren garapen fasean dago. Lau dira Tesi honen ekarpen nagusiak multiHodei aplikazioetarako: i) Segurtasuna eta pribatutasuna ziurtatzeko DevOps metodologia integratua; eta bere osagaiak: ii) segurtasun- eta pribatutasun-betekizunak modelatzeko hizkuntza, iii) arriskuak etengabe ebaluatzeko metodologia bat eta berarekin datorren arriskuan oinarritutako defentsak optimizatzeko teknika, eta iv) Segurtasun eta Pribatutasun Zerbitzu Mailako Hitzarmenen konposizio metodoa.

DevOps metodologia integratua eta bere garapen metodo osagaiak egiazko eOsasun domeinuko multiHodei aplikazio kasu baten azterketan balioztatu dira. Balidatze honek soluzioaren bideragarritasuna eta onurak baieztatu zituen segurtasun eta pribatutasuna ziurtatzeko arrazionalizazio eta sistematizazio multiHodei sistemetan.

Resumen

Los pronósticos del mercado de la Computación en la Nube y las tendencias tecnológicas confirman que la Nube es un fenómeno disruptivo en las Tecnologías de la Información (TI). Sin embargo, la seguridad, la privacidad y la protección de datos siguen siendo las principales barreras para la adopción de los servicios en la Nube. Las preocupaciones de los usuarios sobre la seguridad y la privacidad de los sistemas de la Nube estriban en la falta de confianza, visibilidad y auditabilidad de los controles de seguridad y privacidad que los proveedores de Nube ofrecen en sus servicios.

A nivel europeo e internacional existen estándares recientes e iniciativas sólidas dirigidas a resolver los problemas de confianza del usuario final en la Nube, así como la transparencia en las ofertas de la Nube. Éstas están allanando el camino hacia servicios en la Nube confiables y certificados. Además, el cumplimiento de la nueva Regulación General de Protección de Datos europea es una necesidad urgente para los consumidores y los proveedores de Nube que actúan como procesadores o controladores de datos personales, debido a la necesidad de realizar evaluaciones de riesgos de privacidad de sus sistemas.

En los últimos años, el número de empresas en todo el mundo que adoptaron arquitecturas multiNube en sus estrategias comerciales ha crecido significativamente. Sin embargo, la optimización de costes y el aumento de la competitividad de las empresas que explotan la multiNube solo será posible cuando puedan aprovechar las múltiples ofertas en la nube a la vez que dominan tanto la complejidad de la administración de múltiples proveedores de Nube como las estrategias de seguridad para garantizar la protección frente a una mayor exposición a ataques que trae la multiNube. Con este fin, es necesario considerar no sólo la funcionalidad y los aspectos comerciales de los servicios multiNube, sino también los aspectos de seguridad y privacidad.

En este contexto, la importancia de abordar de forma holística el aseguramiento de la seguridad y la privacidad de los sistemas TI de la Nube y basados en la Nube es clara. Además, es necesario seguir un enfoque sistemático para la gestión de riesgos cibernéticos en multiNube que aborde las amenazas a la seguridad y a la privacidad. Esto es aún más desafiante en los sistemas multiNube debido a la necesidad de evaluar no sólo los riesgos de los propios componentes del sistema, sino también los de los proveedores de Nube de aquellos componentes subcontratados.

Surgen cuestiones fundamentales de investigación sobre cómo diseñar aplicaciones multiNube teniendo en cuenta sus requisitos de seguridad y privacidad para proteger el sistema de riesgos potenciales y sobre cómo decidir qué protecciones de seguridad y privacidad se incluirán en el sistema. Además, se necesitan soluciones para superar las dificultades de garantizar que las propiedades de seguridad y privacidad definidas en el momento del diseño se mantengan durante todo el ciclo de vida del sistema, desde el desarrollo hasta la operación.

En esta Tesis se presenta una innovadora metodología y marco de DevOps integrado que ayudan a racionalizar y sistematizar los análisis de seguridad y privacidad en multiNube para permitir un proceso de decisión informado para una selección equilibrada en costo-riesgo de las protecciones de los componentes del sistema y las protecciones a solicitar a los proveedores de servicios en la Nube utilizados.

El foco del trabajo se sitúa en la fase de Desarrollo del análisis y creación de aplicaciones multiNube. Las principales contribuciones de esta Tesis para aplicaciones multiNube son cuatro: i) La metodología integrada DevOps para el aseguramiento de la seguridad y la privacidad; y sus partes integrantes: ii) un lenguaje de modelado de requisitos de seguridad y privacidad, iii) una metodología de evaluación continua de riesgos y su complementaria optimización de defensas

basada en el riesgo, y iv) un método de Composición de Acuerdos de Nivel de Servicio de Seguridad y Privacidad.

La metodología DevOps integrada y sus métodos de Desarrollo constituyentes se han validado en el caso de estudio de una aplicación real de MultiNube en el dominio de la eSalud o salud electrónica. La validación confirmó la viabilidad y los beneficios de la solución con respecto a la racionalización y sistematización del aseguramiento de la seguridad y la privacidad en sistemas multiNube.

Contents

Abstract i
Laburpena iii
Resumen v
Contents vii
List of figures ix
List of tables x
Acknowledgements 1
1 Introduction 3
1.1 Context 3
1.2 Problem statement and motivation 4
1.3 Research objectives and hypothesis 6
1.4 Research methodology 8
1.5 Organisation of the document 9
2 State of the art in multiCloud Security and Privacy 11
2.1 Introduction 11
2.2 Software frameworks for multiCloud application development and operation 13
2.3 Cloud Security and Privacy requirements modelling 15
2.3.1 Modelling languages for security and privacy aspects 15
2.3.2 Cloud modelling languages 16
2.4 Risk assessment in multiCloud 19
2.4.1 Graph-based risk analysis 19
2.4.2 ADT-based risk assessment 20
2.4.3 Risk assessment in Cloud 21
2.5 Cloud Security Service Level Agreements and Privacy Level Agreements 22
2.5.1 Security Level Agreements in Cloud 22
2.5.2 Privacy Level Agreements in Cloud 23
2.5.3 Service Level Agreements for multiCloud 24
2.6 Standards and regulations on Security, Privacy and SLAs for Cloud applications 24
2.6.1 EU Cloud Computing Strategy 24
2.6.2 GDPR 25
2.6.3 ETSI 25
2.6.4 ENISA 26
2.6.5 NIST 26
2.6.6 ISO 27
2.6.7 Cloud Security Alliance (CSA) 28
2.6.8 CSCC 28
2.6.9 Standard control frameworks 28
2.7 Conclusion 31
3 Integrated DevOps framework for Security and Privacy assurance in multiCloud applications 33
3.1 Introduction 33
3.2 DevOps methodology for security and privacy assurance in multiCloud applications 34
3.2.1 Overall approach 34
3.2.2 Workflow 35
3.2.3 Actors 37
3.2.4 Models 40

3.2.5 Relation with MUSA DevOps methodology....42
3.3 Security and privacy requirements modelling language for multiCloud applications...43
3.3.1 The security and privacy requirements modelling language: extended CAMEL 43
3.3.2 Contributions to security and privacy behaviour specification....46
3.3.3 Contributions to multiCloud deployment specification....48
3.3.4 Contributions to self-protection capability of multiCloud applications....49
3.4 Continuous Risk Management and risk-based optimisation of defences for multiCloud applications....50
3.4.1 Continuous Quantitative Risk Management Methodology for multiCloud DevOps 50
3.4.2 Risk-based optimisation of defences for multiCloud applications....65
3.5 Security and Privacy SLA composition for multiCloud applications....74
3.5.1 Security- and Privacy- aware SLA terms....74
3.5.2 The controls in NIST SP 800-53 Rev. 5....76
3.5.3 SecSLA and PLA Composition methodology....77
3.6 Conclusion....90
4 Solution validation....93
4.1 Introduction....93
4.2 Validation objectives....93
4.3 Validation methodology: eHealth multiCloud application case study....94
4.4 Integrated DevOps methodology supporting security and privacy....95
4.4.1 Solution software prototypes....96
4.4.2 Conclusion....97
4.5 Security and privacy requirements modelling....97
4.5.1 Solution software prototype: MUSA Modeller....98
4.5.2 Conclusion....99
4.6 Continuous Risk Management and Risk-based Optimisation of Defences....100
4.6.1 Modelling of ADTs....100
4.6.2 Risk assessment over ADTs....104
4.6.3 Risk-based optimisation of defences....110
4.6.4 Risk-driven selection of providers and refinement of risks....113
4.6.5 Continuous monitoring of attacks and defences....119
4.6.6 Solution software prototypes: ADToolRisk and ADMind....119
4.6.7 Conclusion....122
4.7 Security SLA and Privacy Level Agreement Composition....123
4.7.1 Application Composition Modelling....123
4.7.2 Per-component self-assessment of SLAs....124
4.7.3 Evaluation of the Per-Component SLA Composition rules....125
4.7.4 Evaluation of the Application SLA....128
4.7.5 Computation of SLO levels in the Application SLA....129
4.7.6 Solution software prototype: SLA Generator....130
4.7.7 Conclusion....130
4.8 Conclusions....131
5 Conclusions....133
5.1 Contributions....133
5.2 Dissemination of the results....135
5.3 Future work....137
References....139
Appendix A: CSA's PLA relationship with Security SLA and GDPR....151
Appendix B: Security and privacy DSL model of the case study....153

Appendix C: Subset of controls from NIST SP 800-53 Rev 5 Draft used in the case study 166

List of figures

Figure 1: Research Methodology followed 9
Figure 2: Thesis contributions integrated in the DevOps workflow of multiCloud applications ... 33
Figure 3: Overall approach of DevOps methodology for Security and Privacy assurance in multiCloud application 34
Figure 4: DevOps workflow for security and privacy assurance in multiCloud applications 35
Figure 5: Parties involved in the multiCloud application provision model 38
Figure 6: Models used in the DevOps workflow for security and privacy assurance in multiCloud applications 41
Figure 7: The proposed Security and Privacy Domain Specific Language for multiCloud applications 45
Figure 8: Continuous Quantitative Risk Management in multiCloud DevOps. 51
Figure 9: General structure of an ADT 52
Figure 10: Example of individual ADTs unification for a) disjunctive ADTs (above) and b) conjunctive ADTs (below). 54
Figure 11: a) System ADT with 4 attack events and 3 defences on k assets, b) 3-D relationship matrix T for system ADT in a). 56
Figure 12: Risk severity quadrants 62
Figure 13: Algorithm to simulate attack-defence scenarios in an ADT 65
Figure 14: Algorithm to find the optimal defence set for an ADT with objective function F2 and F3 69
Figure 15: SLA model integrating PLA and Security SLA. 75
Figure 16: The multiCloud Application SLA Composition Process. 78
Figure 17: Example of an ACM showing its structure. 79
Figure 18: Example of a CMDM between the nodes in the ACM of Figure 17. 81
Figure 19: Example control levels based on metrics' levels. 87
Figure 20: Example of multiple control levels based on metrics' levels. 88
Figure 21: Simplified architecture of the eHealth multiCloud application under study. 94
Figure 22: MUSA Modeller support for Security Agent selection 99
Figure 23: Extract of Steal health data ADT of the use case. 102
Figure 24: Case study ADT with risk vector evaluated in all the nodes. 106
Figure 25: Severity of attack events before and after countermeasures. 107
Figure 26: Risk Sensitivity to At6 attack event probability (left) and impact (right). 109
Figure 27: Case study ADT probability and risk sensitivity to SI-20 defence probability (left), and Case study ADT impact and cost sensitivity to SI-20 defence probability (right). 109
Figure 28: T matrix for the case study ADT. 111
Figure 29: AD matrix for the case study ADT. 112
Figure 30: Case study ADT with risk vector evaluated in all the nodes when only optimal defence set is applied. 115
Figure 31: Case study ADT with risk vector evaluated in all the nodes after optimal defence set refinement 118
Figure 32: ACM of the eHealth multiCloud application of the case study. 124
Figure 33: CMDM of the RA-5 control for the case study. 126
Figure 34: CMDM of the AC-3 control for the case study. 126
Figure 35: CMDM of the SI-20(4) control for the case study. 126

List of tables

Table 1: Overview of proposed multiCloud DevOps workflow stakeholders 39
Table 2: Overview of DevOps Team sub-roles within the DevOps workflow 39
Table 3: Risk vector rules for countered nodes in ADT. 58
Table 4: Risk vector propagation rules in ADT for Equation (2). 61
Table 5: Risk severity metrics 63
Table 6: Privacy related base controls “S” and “O/S” in NIST SP 800-53 Rev. 5 76
Table 7: Estimated risk vector values for attack events in Steal health data ADT 103
Table 8: Estimated risk vector values for defences in Steal health data ADT 104
Table 9: Risk vector values for attack events in Steal health data ADT after defence decoration 105
Table 10: Risk metrics for attack events in Steal health data ADT 107
Table 11: Case study system valuation 113
Table 12: NIST control levels on the basis of metric levels for the case study. 123
Table 13: Excerpt of SLO offers by the components and selected providers 124
Table 14: Example control metric values for controls RA-5, AC-3, SI-20(4) 125
Table 15: Composed control metrics for RA-5, AC-3 and SI-20(4) in the use case. 127
Table 16: Components’ Composed SLAs for controls RA-5, AC-3, SI-20(4). 127
Table 17: Application SLA for controls RA-5, AC-3, SI-20(4). 129
Table 18: SLOs for controls RA-5, AC-3, SI-20(4) in Components’ Composed SLAs 129
Table 19: SLOs for controls RA-5, AC-3, SI-20(4) in Application SLA 130

Acknowledgements

First and foremost, I would like to heartily thank my family for their emotional support and patient with me all along the Thesis work. My friends too showed always their unconditional trust and encouragement that I sincerely thank as well.

I would also like to thank my two PhD directors, Dr. Mariví Higuero from Euskal Herriko Univertsitatea /Universidad del País Vasco and Dr. Xabier Larrucea from Tecnalia Research & Innovation, for providing me the freedom of pursuing my research objectives and for guiding me through my PhD endeavour.

I would like to thank the funding received from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644429 MUSA project which gave me the opportunity to start the present research line as well as allow me to grow as project coordinator and technical manager.

The research leading to these results has also received funding from the same programme under grant agreement No 780351 ENACT project, where I am the coordinator of the Security, Privacy and Resilience in IoT systems Work package and No 787034 PDP4E project were I collaborate in the Privacy Risk Assessment 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.

I would also like to acknowledge all the members of the MUSA, ENACT and PDP4E project consortia for their valuable help. Special thanks for their support in my research to Eider Iturbe, Angel Rego, Jason Xabier Mansell, Massimiliano Rak, Valentina Casola, Maria Carmen Palacios, Borja Urquizu, Gorka Echevarria, Wissam Mallouli, Victor Muntés-Mulero, Smrati Gupta, Peter Matthews, Jacek Dominiak, Stefan Sparh, Balázs Somosköi, Luis E. Gonzalez Moctezuma, Samuel Afolaranmi, Antony Shimmin, Giancarlo Capone, Pasquale De Rosa and Alessandra De Benedictis.

CHAPTER I

1 Introduction

1.1 Context

In the last decade, Cloud Computing technologies have proved to bring enormous advantages compared to previous on-premise scenarios, such as the rapid elasticity of computing resources and cost-efficient business models based on pay-per-use of Cloud Computing services. The figures of annual forecasts of Cloud ecosystem market – including Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) and Software as a Service (SaaS) – are overwhelming with grow figures around $300 billion in 2020, as per predictions of the leading business analyst firm Forrester [1].

The current flourishing of Cloud service solutions of diverse nature and service models invites to think of a near future where multiple Cloud services from different providers are orchestrated at a time by Cloud consumers to take the most out of this technology.

Indeed, multiCloud scenarios where multiple smart services offered as-a-service are combined to accomplish sophisticated IT services, have attracted great interest lately. MultiCloud is perceived as a powerful means to reduce vendor-lock in or dependency with the Cloud Service Provider (CSP). Gartner forecasts that multiCloud will overcome this issue for two-thirds of organizations through 2024 [2].

Therefore, there is an ever-growing number of companies adopting multiCloud strategies in their business. According to [3] multiCloud is the preferred strategy among companies and the 84 percent of companies world-wide have already a multiCloud strategy set up. The amount of companies benefiting from hybrid strategies that combine public and private clouds has also grown to 58 percent in 2019 from 51 percent in 2018.

The European Commission's efforts to boost Cloud adoption have also been significant in the last years. Seeing Cloud as a key market and technology enabler for the European Commission Digital Strategy and the EU Digital Single Market [4], the European Commission Cloud Computing Strategy [5] promotes a cloud-first approach with a secure hybrid multiCloud service offering.

However, in this distributed heterogeneous Cloud context, from the Cloud Service Consumer (CSC) perspective, controlling the overall behaviour of the Cloud-based application is a major challenge. Cloud Computing poses the challenge of lack of visibility and control on how exactly the Cloud services consumed work and which are security and privacy guarantees they offer. Furthermore, the lack of certifications and obligations of following recent standards (such as ISO/IEC 27017 [6] and ISO/IEC 27018 [7]) makes it almost impossible to compare features between Cloud service offerings. This is even more arduous when it comes to the security and privacy aspects of the Cloud services, due to the confidentiality that Cloud Service Providers (CSPs) maintain about the insights of the offered services.

Furthermore, the entry into force of the General Data Protection Regulation (Regulation (EU) 2016/679) [8] in May 2018 demands that systems, including Cloud-based systems, adhere to a number of legal clauses requiring that the personally identifiable information (PII) is protected, which need to be technically implemented somehow.

The future EU Cybersecurity Certification Framework [9] under development by ENISA, and which is expected to be deployed in the year 2021, will establish an EU-wide cybersecurity certification framework for information and communication technology (ICT) products, services, and processes. The EU-wide Cybersecurity Certification is intended to advance trust through a set of cybersecurity certification schemes that include common cybersecurity requirements and

evaluation criteria across national markets and sectors. These certification schemes are oriented to ensuring trust and auditability of the certified ICT systems.

The compliance of Cloud services with this new framework would also require that technical measures are adopted to ensure Cloud services comply with specified cybersecurity requirements. Hesitant adopters of Cloud would therefore benefit from these certification schemes as Cloud services would have proved minimum security and privacy-respectful behaviour prior to their delivery. Still, it is not decided yet whether these certification schemes will be mandatory.

In this line, the present work contributes to the formalisation and implementation of technical measures that help in the realisation of security and privacy in multiCloud applications.

For the purpose of this work, a *multiCloud application* is a distributed application over heterogeneous Cloud resources whose components user or are deployed in multiple Cloud services offered by a-priori independent and non-federated Cloud service providers, and still they all work in an integrated way and transparently for the end-user. Developers of multiCloud applications need to devise sound strategies to architect their systems and manage all the functional aspects of the Cloud service mesh while ensuring the secure and privacy respectful behaviour of both the individual services as well as of the overall application.

The present work proposes an integrated framework that aids in this sense and enables to continuously ensure secure and privacy-respectful behaviour of multiCloud applications along their whole life cycle, from design to operations.

1.2 Problem statement and motivation

Cloud Computing is an emerging promising paradigm for enabling new business models and economies of scale based on on-demand provisioning of IT resources (both hardware and software) over a network as metered services, where consumers are billed only for what they consume. A recent Gartner forecast [10] shows that the revenue of investments on public Cloud services is expected to be $266.4 billion in 2020, $308.5 billion in 2021 and $354.6 billion in 2022. The growth is attributed to the increasing demands of modern smart applications and heavy workloads, which infrastructure requirements that traditional data centres cannot meet.

Nevertheless, since Cloud inception, enterprises consider security as the #1 inhibitor to Cloud adoption [11] [12]. Companies are reluctant to adopt Cloud Computing because of the difficulty in evaluating the trade-off between Cloud benefits and the additional security risks and privacy issues it may bring. Most concerns are related to data protection, regulations compliance [13] and other issues due to lack of insight (of controls and governance processes) in the outsourcing of data and applications: data confidentiality, trust on aggregators, control over data and/or code location, and resource assignment in multi-tenancy [14]. Businesses that want to exploit Cloud computing need to be vigilant in understanding the potential privacy and security breaches in this new environment [15].

Trustworthy Cloud environments are even more challenging today, since they are becoming more and more complex in reference to the number of Cloud resource types that are available “as a service”. Besides the traditional three service models defined by the NIST SP 500-292 Cloud Reference Architecture [16] (IaaS, PaaS and SaaS), new models are expanding such as Network as a Service specified by the ITU-T or Data as a Service defined in ISO/IEC 17826:2012 [17].

As the number of Cloud service models, Cloud resources and Cloud providers grow in the market, it becomes theoretically easy (but not necessarily technically) for the Cloud consumer to deploy and use multiple Cloud solutions at the same time in an integrated manner [18]. This means that despite the diverse characteristics of the Cloud services, such as own management interfaces and

own service level offerings (related to both functional and security aspects), all need to be smoothly integrated, monitored and managed as a single working entity or system: the *multiCloud application*.

MultiCloud applications benefit therefore from the adoption of Cloud services of different capability types (i.e. infrastructure, platform or software as a service) supplied by different Cloud Service Providers (CSPs). MultiCloud follows the concept of distributed computing in which the components are dispersed over heterogeneous Cloud resources but communicate in an integrated manner to achieve the desired goal.

MultiCloud applications combining heterogeneous Cloud services are the most challenging applications in Cloud ecosystems since they have to deal with the security and privacy of the individual components as well as with the overall application security and privacy including the communications and the data flow between the components.

Despite the Cloud service providers used may offer their own security and privacy controls, the multiCloud application must ensure integrated security and privacy across the whole composition. Therefore, the overall security and privacy depend on the security and privacy properties of the application components, which in turn depend on the security and privacy properties offered by the Cloud resources they exploit. For instance, the database component in charge of storing sensitive data cannot ensure a high confidentiality if the Cloud storage resource in which it is deployed does not use strong encryption algorithms. Consequently, the whole multiCloud application may not be sufficiently secure.

MultiCloud model offers the opportunity to maximise the benefits of the combination of the Cloud resources in use when the best CSPs that satisfy both application and component level requirements are selected. However, the distributed model makes security and privacy management even more complex as the need arises to tackle them at different levels: individual components, component-to-component communication and overall application. Specifically, for Cloud security solutions, the Forrester report [1] foresees that "*cross-cloud management providers must buy, build, and/or acquire security capabilities that go beyond past identity and access management".* Therefore, this calls for approaching security and privacy in multiCloud from a holistic point of view and providing sophisticated security analysis and assurance solutions that go beyond basic common security functionalities.

The ever growing flexibility and hybridation of provisioned services in the Cloud [19] and the increasing diversity of smart things and services in the Internet of Things (IoT) [20] are pushing multiCloud system architectures towards higher complexity and orchestration of more and more components, many of which come from third-party providers which often present limited transparency about the security and privacy features they offer. This introduces difficulties in evaluating threats against all system components and poses new challenges to risk management solutions as they need to consider the relationships among attacks against different parts of the application, because ignoring them may lead to erroneous interpretation of risks. When analysing the impact of cyber risks, it is imperative to take also into account the protections in place in all of the components of the multiCloud system, that is, the security and privacy controls already provided by internal components as well as by third-party components' CSPs.

Devising all potential risk situations for a system and full compliance assurance are challenging. However, there is a growing need for quantifiable and demonstrable risk assessment solutions [21]. This is particularly challenging in complex systems such as multiCloud applications where system components use or are deployed over multiple distributed Cloud and IoT services from a-priori independent providers. These systems require holistic security approaches that support experts in managing risks in all system components including outsourced services which often lack details of

security and privacy measures they adopt. Third-party components are usually beyond the control of the developers and their security falls on the hands of Cloud Service Providers (CSP) or IoT providers. Still, as part of the system, threats and protections of these components need to be analysed and considered when assessing overall system risks.

Even more, reliable risk evaluation methods and tools are required to respond to security and privacy conditions brought by new standards such as the General Data Protection Regulation (GDPR) [6], which article 35 requires organisations to carry out data protection impact assessments where privacy risks are evaluated and solutions to minimise them identified. Recent international standards such as ISO/IEC 27005 [22] and ISO/IEC 27701 [23], NIST Cybersecurity framework [24], etc. promote the adoption of risk management practices which drive the selection and maintenance of system protections.

In addition, as explained before, forthcoming security certifications for ICT digital products, services and processes announced by the EU Cybersecurity Certification Framework [9] will also require organisations to adopt systematic approaches to security assurance and cyber risks management.

The present work aims to address all these needs by proposing a solution which supports the analysis and reasoning of security and privacy aspects of the multiCloud application throughout the whole life cycle (including design, deployment and runtime phases). The smooth integration of Development phase methods with continuous security and privacy assessment would allow for continuous assurance at application operation phase.

The work seeks solutions relying on the use of security-by-design and privacy-by-design mechanisms that enable Cloud Service Consumers creating multiCloud applications using Cloud services from potentially independent and heterogeneous providers to be able to take informed decisions on security strategies to follow.

To this aim, as part of the security- and privacy-by-design mechanisms, research solutions to evaluate and analyse cyber risks dealing with architectural complexity of multiCloud will be studied. This will enable to understand the implications of risks to different parts of the system in the overall system risks and define security strategies according to overall risk reduction rather than individual components risks, which may not be the optimum solution for the system as a whole.

Similarly, mechanisms to identify which security and privacy levels can be promised to multiCloud application consumers will be studied. The protections (or controls) and their service levels agreed with the customers for the whole application running as a single service will necessarily be built upon the levels offered by each of the constituent components including components outsourced to Cloud. As for security the principle that the weakest link in a chain determines the security of the whole chain holds, it is necessary to determine the overall system security level that can be declared in different multiCloud architectures, including when the protections of the system are implemented by only some, all or different sets of collaborating components.

1.3 Research objectives and hypothesis

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

To research, design, and develop a solution to support the security- and privacy-aware development and operation of distributed applications over heterogeneous Cloud and IoT resources (multiCloud applications).

The solution will be formulated as a holistic framework that enables developers and operators of multiCloud applications to tackle security and privacy features of the system in their activities as

intrinsic to the system life-cycle, not as afterthoughts of the system design or as side-activities in system operation.

This goal can be broken down into the following objectives:

- Analyse, research and provide security-by-design and privacy-by-design mechanisms supporting the specification of security and privacy requirements in multiCloud applications.
- Analyse, research and provide quantitative risk assessment methods for multiCloud applications to drive the deployment in Cloud services that best match application security and privacy requirements while minimising cyber risks.
- Analyse, research and provide mechanisms to compute the offered composite security Service Level Agreements (SLAs) that can be used in operation to continuously ensure the fulfilment of the designed security and privacy properties in multiCloud environments.

Security and privacy assurance in the life-cycle of multiCloud applications involves many different open research aspects. Among all of them, the focus of the present work is on the following questions:

1. *How can we express the security and privacy requirements of a multiCloud application in the design so as we are able to ensure such requirements are satisfied by the application at runtime?*
2. *How can we deploy a multiCloud application minimising the security and privacy risks even when the control over some of its components is fully or partially on the hands of the CSPs?*
3. *How can we obtain the security and privacy Service Level Agreement (SLA) of a multiCloud application on the basis of its constituent components so as we know what controls and which levels for those controls can be guaranteed to application customers? And how can this be done when some of the components are outsourced to external CSPs?*

The major research hypothesis considered is:

H- It is possible to demonstrate that the proposed framework can contribute to the security- and privacy-aware creation and operation of multiCloud applications which specific security and privacy requirements can be analysed and specified at design time, as well as controlled in operation.

This hypothesis can be broken down into the following ones:

- *H1- It is possible to address security and privacy aspects assurance in a continuous way in the multiCloud application life-cycle through the DevOps approach.*

The DevOps paradigm [25] (a definition that mixes the terms "Development" and "Operations") promotes the close collaboration between Development and Operation teams in the application life-cycle and automation of software deployment and delivery. the adoption of DevOps approach increases reliability of software releases while reducing time to delivery.

Therefore, this hypothesis reflects the idea that it is possible to apply the DevOps approach in multiCloud applications to achieve the continuous alignment and feedback of security and privacy properties assessed at Operation with the security requirements defined for the application at design time of Development phase.

- *H2- It is possible to express the security and privacy requirements of multiCloud applications in a way that they can be assessed in operation.*

Hence, the security and privacy properties of in the multiCloud application can be captured as part of the system requirements at development phase so as they can be assessed later at runtime. These requirements shall include security as well as privacy characteristics, such as availability, access control and data protection (be it personally identifiable information (PII) which would lay on privacy field, or not, which will lay on security field).

- *H3- It is possible to continuously evaluate the security and privacy risks of multiCloud applications based on identified threats against application components and standard controls adopted by them so as to drive the selection of the best combination of Cloud Services that minimises the risks.*

This hypothesis outlines that it is feasible the computation of the security and privacy risk level of the overall multiCloud application by taking into account the threats envisaged by the end-user against the components (system assets) and the controls (defences) adopted to protect them. The initial evaluation of the system (overall application) risk shall enable the identification of the best defences to minimise the risks and among them those required from external Cloud services. As a consequence, it is possible to search for the best Cloud Services to be used by the multiCloud application components that offer such required defences. The continuous evaluation of the risk level at operation shall be done by continuously monitoring both threats and defences status.

- *H4 - It is possible to create Composed Service Level Agreements (SLAs) of multiCloud applications on the basis of security and privacy SLAs of their components taking into account the deployment relationships and the controls' metrics implementation delegations among the components.*

This hypothesis states that it is possible to derive the security and privacy Service Level Agreement (SLA) offered by the multiCloud application to its end-users from the SLAs of the individual components, Cloud services and IoT services exploited by the components. The composed Application Service Level Agreement (Application SLA) specifies the security and privacy related controls together with their Service Level Objectives (SLO) to be offered by the application to its customers. Hence, the composed Application SLA is instrumental to be able to ensure the security and privacy behaviour of the multiCloud application in operation by continuously monitoring the fulfilment of the SLOs therein.

In conclusion, the core contributions of this Thesis, i.e. the Integrated DevOps methodology framework for seamlessly supporting Security and Privacy in multiCloud applications together with its integrating parts address the identified need of supporting continuous security and privacy assurance in multiCloud applications, even when the Cloud services used are heterogeneous and components operate under changing threat conditions.

The main scientific contributions of this Thesis to Cloud-based systems security and privacy are described in Section 3 and discussed in Section 5.

1.4 Research methodology

The research methodology followed in this work is based on a double iteration in the research cycle shown in Figure 1.

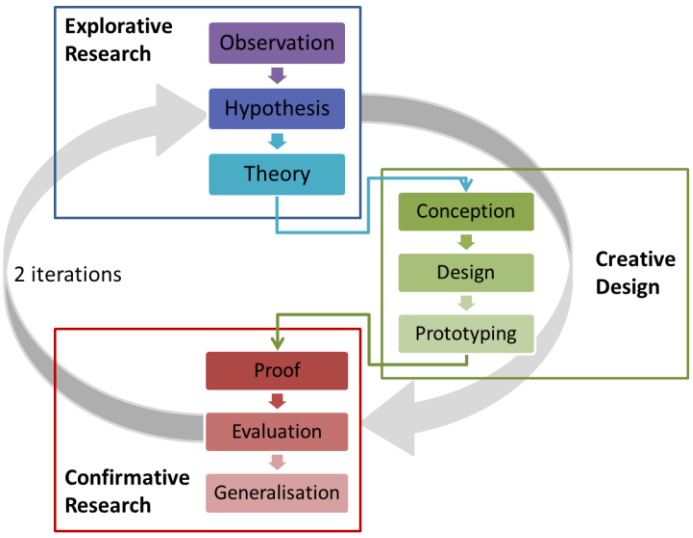


Figure 1: Research Methodology followed

As seen in the figure, it is basically a *design inclusive research* methodology as described by Imre [26] with two iterations on the prototyping phase. These will allow improving the work outcomes by updating the theory deduction after an initial validation phase.

The work started with the phase of *explorative research* actions, followed by a phase of *creative design* actions, and finally the phase of *confirmative research* actions.

First, the *explorative research phase* allowed for a thorough survey and analysis of state of the art models, theories, mechanisms, technologies and solutions to identify the baseline and define the hypothesis and expected scientific contributions and technical innovations of this work.

Second, the *creative design phase* consisted in conceptualising the solution and giving it form in the design specifications followed by the prototype implementation.

Finally, in the *confirmative research* phase, the solution prototype was validated in two use cases that involved two different types of multiCloud applications. The validation use case and methodology are fully described in Section 4. The methodology followed a continuous approach, with two major evaluation milestones designed within the use case, corresponding to the two major releases of the framework prototype. The continuous validation allowed to quickly react to failures in the conception, design and implementation of the solution. This way, the work rapidly progressed on tested solutions that could feed back the deduction phase in order the results could be improved addressing the validation findings.

1.5 Organisation of the document

The outline of the dissertation is structured as follows:

- Section 2 analyses the state of the art in multiCloud Security and Privacy, as follows:
  - Section 2.1 introduces the section explaining how the term multiCloud is understood in the present work and summarises major Security and Privacy threats and challenges in multiCloud.
  - Section 2.2 analyses existing software frameworks supporting multiCloud application development and operation.
  - Section 2.3 offers an overview of Security and Privacy requirements modelling in Cloud.
  - Section 2.4 describes Risk assessment techniques for Cloud applications.

  - Section 2.5 focuses on Cloud Security and Privacy Service Level Agreements (SLAs)
  - Section 2.6 summarises the standards and regulations that affect Cloud Security and Privacy assurance
  - Section 2.7 ends the section with the analysis of major challenges and unsolved issues.
- Section 3 presents the proposed Security- and Privacy-aware DevOps methodology for multiCloud applications and the supporting software framework. In particular:
  - Section 3.1 introduces the contributions and describes the structure of the section.
  - Section 3.2 provides the description of the proposed overall solution (methodology and workflow) for security- and privacy-aware DevOps of multiCloud applications together with the description of the proposed supporting software framework.
  - Section 3.3 presents the proposed Security and privacy requirements modelling language for multiCloud applications.
  - Section 3.4 describes the Continuous Risk Management and defence optimisation techniques proposed for multiCloud applications.
  - Section 3.5 details the proposed SLA-based Continuous Security and Privacy Assurance of multiCloud applications within the Operations activities of the workflow.
  - Section 3.6 concludes the contributions part.
- Section 4 describes the solution evaluation carried out in a real case study multiCloud application.
  - Section 4.1 introduces the validation approach.
  - Section 4.2 describes the validation objectives.
  - Section 4.3 details the case study and the methodology followed.
  - Section 4.4 corresponds to the validation of the overall methodology of Section 3.2.
  - Section 4.5 explains the validation of the modelling language of Section 3.3.
  - Section 4.6 gathers the validation of the risk methodology and defence optimisation of Section 3.4.
  - Section 4.7 describes the validation of the SLA composition methodology of Section 3.5.
  - Section 4.8 offers the major conclusions of the validation.
- Section 5 outlines the main contributions of the Thesis and gives the conclusions of the work. The list of publications associated to the Thesis is also provided herein together with the summary of future research lines.

2 State of the art in multiCloud Security and Privacy

2.1 Introduction

At state of art, the term *MultiCloud* is used in many different contexts and refers to the idea of accessing resources from different Cloud Service Providers (CSPs). MultiCloud solutions represent a new challenging field in order to add value to overall Cloud client experience [27]. In order to exploit multiCloud potentialities, different architectural approaches can be adopted [28]:

- replication of applications, i.e. the same system is deployed in more than one provider and malicious attacks can be easily discovered comparing operation results;
- partition of application system into tiers, that allows to separate logic from data;
- partition of application logic into fragments, that obfuscates the overall application logic to providers;
- partition of application data into fragments, that makes impossible to a single provider to reconstruct data, safeguarding confidentiality.

The framework proposed in this Thesis aims at addressing security and privacy in all types of multiCloud environments which may combine multiple scenarios from the list above. Therefore, *multiCloud* can be considered as a special case of *inter-cloud* computing, which has been defined in [29] as: *A cloud model that, for the purpose of guaranteeing service quality, such as the performance and availability of each service, allows on-demand reassignment of resources and transfer of workload through a interworking of cloud systems of different cloud providers based on coordination of each consumers requirements for service quality with each providers SLA and use of standard interfaces.*

Even if in the literature the terminology is not yet stable, Grozev and Buyya [30] proposed to adopt the term *inter-cloud* as the generic term indicating the adoption of multiple CSPs. The term *cloud federation* describes a set of cloud providers that voluntarily interconnect their infrastructures to allow sharing of resources while the term *multi-cloud* or *multiCloud* refers to the usage of cloud services from different CSPs without the need of having an explicit agreement between the service providers.

Security in multiCloud applications is an open and debatable topic. Part of the existing literature considers that multi-cloud paradigm can improve security, others, on the contrary, believe that multiCloud paradigm brings new security risks and vulnerabilities, due to when distributing applications among multiple Cloud Service Providers (CSPs) the attack surface is enlarged, and the number of potential security issues increases, reducing as a result the overall level of security.

Alzain et al. [31] and Bernstein and Vij [32] offer simple surveys of solutions that try to improve the security using multiCloud techniques. Particularly, the main results are available for storage services such as solutions by Yan et al. [33] and Oliveira et al. [34] who proposed techniques to distribute a file over multiple providers or untrusted networks, achieving higher data confidentiality and integrity. It is worth noticing that all the papers that sustain the higher security of the multiCloud approach focus on increasing one or more specific security properties offered to the customers.

Bohli et al. [35] and Singhal et al. [36] analysed multiCloud applications security from a different point of view: they analyse different multiCloud solutions and try to make a security assessment of the overall application behaviour, outlining the new security issues introduced by the multiCloud approach. While the security assessment approach is very interesting, both works deal with a very high-level description of the solution and do not offer a clear solution to make an assessment for a real multiCloud application.

In addition, it is interesting to note that most of the multiCloud related literature is focused on security and not on privacy features. This is understandable due to the strong interrelationships between both security and privacy mechanisms, because many of the means to achieve privacy of personal data rely on security mechanisms that ensure confidentiality, integrity and availability of data. Mechanisms such as access control, secure storage, data encryption, vulnerability analysis, DoS protection, etc. are also required for privacy protection. With the entry into force of the General Data Protection Regulation in 2018 [6], IT system privacy is gaining relevance and a growing number of works are dealing with privacy enhancing technologies.

However, at the best of the author's knowledge, no previous work has addressed yet the analysis of the development and assessment of multiCloud applications in a *propositive* and *systematic* way with respect to security and privacy. That is, no concrete approaches were found that apply security-by-design and privacy-by-design principles to the multiCloud application engineering process, supporting the identification of the relevant security and privacy threats that a multiCloud application faces and using ad-hoc mechanisms at operation to address the issues that security and privacy risk assessment indicates.

The security and privacy requirements definition and their compliance assurance studied in this Thesis are focused in data protection addressing the following security objectives: data confidentiality, data integrity, data localisation and data usage (regarding access). These are in line with the major security objectives related to Cloud Computing identified by the Cloud Standards Coordination Working Group within the European Cloud Computing Strategy [37]:

- Protect data from unauthorized access, disclosure and modification
- Prevent unauthorized access to cloud computing resources
- Ensure effective governance, control and compliance processes are in place
- Ensure appropriate security provisions for cloud applications
- Ensure security of cloud connections and networks
- Enforce privacy policies

In an integrated multiCloud application life-cycle management it is critical to keep in mind the security and privacy requirements of the application since the very beginning. Starting from application conception and design, all the phases in the lifecycle need to be carried out with the objective of obtaining as a result a security and privacy-aware application.

Still, knowing the security and privacy features of Cloud services in use by the application is a challenging task nowadays. Cloud consumers face the lack of transparency with respect to which security and privacy controls are applied by the Cloud Service Providers (CSPs), particularly if they are public Cloud services. No single or unified controls' taxonomy exists nor catalogues of Cloud services that provide such information to help consumers in benchmarking the Cloud services with respect to the security and privacy features they offer. Usually, available information in the Internet needs to be extracted from CSP's informative (but not legally binding) websites that most of the times focus on functional information. As an exception, the Cloud Security Alliance's STAR repository (see Section 2.6.7) provides publicly accessible self-assessments by CSPs about offered security controls in their services, but limited information on privacy controls is available therein. Therefore, in Cloud services arena, just as for many other IT services and systems, there is still a long way ahead for the formal declaration of the controls or defences offered that eases the comparison and selection of services. In any case, it is recommended that the control declaration follows a standard taxonomy. As it will be shown in Section 2.6.9 there are already some efforts in this line of standardisation.

The next subsections provide a summary of the state of the art on multiCloud application security and privacy with a focus on security and privacy assurance aspects addressed in the present work.

First, Section 2.2 provides an overview of software frameworks for multiCloud applications. Second, Section 2.3 analyses existing security and privacy modelling languages in Cloud. Third, Section 2.4 describes the state of the art on Risk assessment for multiCloud applications. Fourth, Section 2.5 recalls the Cloud security and privacy Service Level Agreement concepts and existing works on the subject. Finally, Section 2.6 describes the standards that impact the development of multiCloud applications, and which establish the baseline of the present research work.

2.2 Software frameworks for multiCloud application development and operation

This section analyses existing integrated software frameworks that support the design, development or operation of multiCloud-based applications.

Despite at state of the art few concrete integrated software frameworks for multiCloud exist, the topic is considered extremely relevant. The need for multiCloud solutions is well demonstrated by the number of research projects that are proposing techniques and tools to address the multiCloud approach. Multiple EU-funded research projects have already initiated the path of frameworks and solutions supporting multi-cloud application development, management and assurance.

For example, the main goal of the MODAClouds project [38] was to provide methods, a decision support system, an open source Integrated Development Environment (IDE) and run-time environment for the high-level design, early prototyping, semi-automatic code generation, and automatic deployment of applications on multiple Clouds with guaranteed Quality of Service (QoS). Model-driven development combined with novel model-driven risk analysis and quality prediction will enable developers to specify Cloud-provider independent models enriched with quality parameters, implement these, perform quality prediction, monitor applications at run-time and optimize them based on the feedback. Additionally, MODAClouds offered techniques for data mapping and synchronization among multiple Clouds. The approach followed in the framework proposed in this Thesis adopts the DevOps approach [25] initiated by MODAClouds in multiCloud and provides additional support to security and privacy features of the application which were not studied at all by MODAClouds.

By following a model-based management of Cloud applications, the PaaSage project [39] provided a (multi-)cloud application development and deployment platform. The focus is on how to define and execute the application components deployment in multi-cloud (“cross cloud”), based on both QoS and security parameters. The application design is made using CloudML [40] which serves to specify the deployment requirements. PaaSage is also providing a formalism (not yet fully developed) for the specification of security requirements in CAMEL language [41]. The monitoring of components is made at the level of virtual machine (VM) and can be combined with “other monitors”. Component invocations and execution engine actions are also monitored information. As described later, in the methodology presented in this Thesis an extension of the CAMEL language is proposed for the security and privacy requirements modelling.

The SeaClouds project [42] aims at adaptive management of complex applications deployed across multiple clouds by supporting the distribution, monitoring and migration of application modules over multiple heterogeneous PaaS. The focus is on assuring the QoS of the complex application but does not address specifically the security and privacy issues that are studied in this Thesis.

With a stronger focus on security and privacy, the A4CLOUD project [43] dealt with accountability issues in Cloud and Future Internet services as the most critical prerequisite for effective governance and control of corporate and private data processed by cloud-based services. The project delivered methods and tools, through which cloud stakeholders can be made accountable for the privacy and

confidentiality of information held in the cloud. These methods and tools combine risk analysis, policy enforcement, monitoring and compliance auditing. As part of the A4CLOUD work, the Cloud Security Alliance defined the Privacy Level Agreement concept adopted in this Thesis.

More recently, the CUMULUS project [44] delivered an integrated framework of models, processes and tools supporting the certification of security properties of cloud services (IaaS, PaaS or SaaS), but the approach did not follow the DevOps paradigm [25] and the focus was more on certification after the application is built.

The SPECS project [45] aimed at delivering an open source framework to offer Security-as-a-Service, by monitoring security parameters specified in SLAs, and also providing the techniques to systematically manage SLAs life-cycle. The project provided solutions for automatic Negotiation and Monitoring of SLAs between CSPs and SPECS platform based on security properties of cloud services. The work presented in this Thesis directly links with the outcomes of SPECS as it extends these to multiCloud environments and addressing not only security but also privacy features of multiCloud applications.

For further details on these frameworks, we suggest the interested reader the following papers which offer complete surveys of them: [29] [30] [46].

Most remarkably for this Thesis, the MUSA project [47], which was the germ of this work, developed a framework to address the issue of creating multiCloud applications taking into consideration the end-user security requirements from the early development stages and providing a continuum of security level assurance along the whole application life-cycle. Through the exploitation of DevOps paradigm [25], the MUSA approach integrates run-time assurance controls and mechanisms with design mechanisms so the security assurance at application operation is smoothly aligned with requirements introduced at design time (security-by-design).

The core methodology of MUSA was selected to be improved in this Thesis because it was the one most advanced in multiCloud security DevOps. As it will be explained in Section 3, the refinements developed are mostly to overcome major MUSA limitations in Development activities of MUSA methodology: i) privacy is not studied in the MUSA methods, ii) the risk assessment does not consider the relationships between different attributes of threats and defences, iii) no technique to optimise the security and privacy controls to apply in the system is offered, and iv) the SLA composition in MUSA is limited since it only addresses Security SLAs and not Privacy SLAs, control delegation relationships are not evaluated when building the SLAs, and no SLOs to declare in the SLA are deduced for the composed application.

In this Thesis the proposed solution to security and privacy assurance in multiCloud applications relies on an integrated approach that monitors the security and privacy Service Level Objectives (SLOs) stated in the Security and Privacy SLA and is able to react to violations of the objective values, in case they occur due to a flaw in the application or due to a deliberate attack. In this way, it brings an advance over the state of the art towards resilient self-healing cloud-based applications that can adapt to changing conditions in the environment or the security and privacy threat landscape.

The contributions of this Thesis are not focused on how the SLAs are monitored, but on how the needed risk analysis is carried out to identify the controls to use in the system components and how to obtain the composed application Service Level Agreement to be able to use it to monitor whether the guarantees promised to the customers hold. That is, the focus is on how to know what should be monitored at component level so as the Application SLA holds and how the computation of the risk level is determined on top of the monitoring results. The method to get to this knowledge differ from those provided by MUSA, as described in Section 3.

Whenever the risk level is not kept within the desired limits, the enforcement of security and privacy mechanisms or controls will be activated. The application layer agents that enable the Continuous Assurance promoted in this Thesis can be considered as a type of runtime application self-protection (RASP) [48] technique applied in multiCloud application operation, due to they are aimed at enforcing at application layer particular mechanisms to protect the system. The RASP technology term was introduced by Gartner [48] and advocates for security mechanisms built in or linked to an application runtime environment to control execution and protect from real time attacks. As application layer technique, RASP does not compete with but complements other protections on the network and system layers. Furthermore, it perfectly matches with continuous delivery and agile DevOps where runtime controls are baked in in every release of the application [49].

2.3 Cloud Security and Privacy requirements modelling

This section provides an analysis of the existing modelling languages and tools to formally conceptualise requirements in multiCloud applications and in distributed applications in general. The analysis focuses on how they support security and privacy requirements capturing.

2.3.1 Modelling languages for security and privacy aspects

During the last two decades multiple languages have appeared to capture information system architecture and requirements. Following the well-known Model Driven Engineering (MDE) discipline [50], which promotes the use (and reuse) of system models as the first entity in the software engineering process, information systems can be specified using general-purpose languages like the Unified Modelling Language (UML) [51] or the process-oriented language Business Process Modelling Notation (BPMN) [52]. These languages are used to represent implementation-agnostic abstractions which are then refined into system views nearer to actual implementation and deployment by applying the appropriate transformations.

To fully unfold the potential of MDE, models are frequently specified using domain-specific languages (DSLs), which are tailored to a specific domain of concern. In this line, modelling languages capturing security requirements like Secure Tropos [53], Misuse Cases [54], Mal-Activity Diagrams [55] assist analysts in describing and analysing security concerns. Secure Tropos adopts the agent-oriented paradigm for the integration of security into software engineering and its major limitation is that the security concepts are not captured in models that can be later refined into implementation models. As opposed to traditional use cases in UML, Misuse Cases model scenarios where hostile stakeholders misuse the system. Mal-Activity Diagrams capture the activities leading to negative impacts on the system. Most notably, Rodriquez et al. [56] define a BPMN extension called Business Process Security (BPSec), which enables the business analyst to specify security requirements as part of the business process models. These could be transformed into e.g. use case diagrams following the MDE approach, though this line has not been yet explored for multiCloud systems.

As explained in the following subsection, part of the state-of-the-art solutions also rely on the use of UML-based modelling languages for Cloud systems. Rather than a business process oriented approach, the solution adopted in this Thesis is also based on Model-Driven Engineering to capture the architectural and security requirements of multiCloud applications while the threats against the system and corresponding protections are captured in form of Attack Defence Trees (ADT) as explained in Section 3.4.

2.3.2 Cloud modelling languages

MultiCloud applications have complex composition, provisioning and deployment requirements, and the application design becomes even more complex at the time additional aspects such as security and privacy enter in the equation. Therefore, several initiatives are running in order to support this type of activities. In the following paragraphs, the main formalisms or languages that are currently used for modelling Cloud applications are collected. The collection starts with CAMEL language which has been selected as the basis for the present work.

CAMEL

Cloud Application Modelling and Execution Language (CAMEL) [41] is a family of domain-specific languages (DSLs) defined in the PaaSage EU project [39] in order to cover the necessary aspects of the modelling and execution of cross-Cloud applications, which is the name used in PaaSage [57] for denoting multiCloud applications.

As described in [58], CAMEL integrates and extends existing DSLs, namely Cloud Modelling Language (CloudML) [40] [59], Saloon [60], and the organisation part of CERIF [61]. In addition, CAMEL integrates new DSLs developed within the project, such as the Scalability Rule Language (SRL) [62] or new features (e.g. WS-Agreement parts etc.). Generally speaking, CloudML is used to describe the application structure and specify the topology of virtual machines and application components [63]. In brief, the key modelling elements that CAMEL shares with CloudML are: Cloud, VM type and VM instance, Internal component, Hosting and Hosting Instance, Communication and Communication Instance.

In addition, CAMEL enables engineers to specify multiple aspects of cross-Cloud applications, such as provisioning and deployment topology and requirements, service-level objectives, metrics, scalability rules, providers, organisations, users, roles, execution contexts, execution histories, etc.

On security and privacy aspects, CAMEL allows to specify *security controls* of the components, i.e. safeguards required from the Cloud resources that the components will use. Note that the language does not specifically address privacy controls, though the concept *security control* could be adopted as a generalisation of both security and privacy control, without the need of doing a specific differentiation between them.

CAMEL supports models@run-time, which provides an abstract representation of the underlying running system, whereby a modification to the model is enacted on-demand in the system, and a change in the system is automatically reflected in the model. By exploiting models at both design- and run-time, and by allowing both direct and programmatic manipulation of models, CAMEL enables self-adaptive cross-Cloud applications (i.e., cross-Cloud applications that automatically adapt to changes in the environment, requirements, and usage).

In order to facilitate the integration across the components managing the life cycle of multiCloud applications, PaaSage leverages upon CAMEL models that are progressively refined throughout the modelling, deployment, and execution phases of the PaaSage workflow [64]. This way, the CAMEL based Cloud provider-independent model is transformed into a Cloud provider-specific model and then to the Deployment model which respects all the provision and scalability constraints of the actual resources to use.

A similar approach of model refinement is promoted in this Thesis by using an enhanced version of CAMEL able to capture the security requirements of the specific application. The model capturing deployment and security constraints is progressively refined to the actual Cloud platform view, starting from provider-independent model to provider-specific model and finally to deployment or implementation model.

CloudML

CloudML (Cloud Modelling Language) [40] [59] is an initiative by SINTEF research center in Norway which aims at providing a domain-specific language to support the specification of provisioning, deployment and adaptation concerns related to multiCloud systems at design-time and their enactment at runtime. CloudML's background is PIM4Cloud [65] language, defined in REMICS project. CloudML has been further enhanced through projects like MODAClouds, PaaSage and REMICS.

CloudML is inspired by the Model-Driven Architecture (MDA) [66] which is an instance of MDE developed by Object Management Group (OMG). The language supports application deployments to be specified in terms of Cloud provider independent models (CPIM), where the refinement into Cloud provider-specific models (CPSM) is foreseen in a separate step.

CloudML meta-model abstract syntax is realized in terms of a meta-model based on Ecore. A CloudML model assembles components exposing ports (or interfaces), and bindings between these ports. Therefore, the main concepts of CloudML can be summarized as follows:

- *Internal component*: Represents a reusable type of application component to be deployed onto an external component.
- *External component*: Represents a reusable type of a virtual machine or platform service.
- *Port*: Represents a required or provided port to a feature of a component.
- *Communication*: Represents a communication binding between ports of two components, which implies a dependency between the components.
- *Hosting*: Represents a binding between a component deployed onto another one.
- *Cloud*: Represents a collection of virtual machines offered by a particular Cloud provider.

CloudML, like TOSCA (see below), is built on component-based approaches, which facilities reusability and separation of concerns [40]. Moreover, CloudML exploits the type-instance pattern [67] to foster reuse of defined types, e.g. a virtual machine type with specific characteristics.

CloudML in MODAClouds

In MODAClouds project, a large set of tool-supported domain-specific languages collectively called MODACloudML has been developed. MODACloudML relies on the following three layers of abstraction [68]:

- *Cloud-enabled Computation Independent Model (CCIM)* to describe an application and its data.
- *Cloud-Provider Independent Model (CPIM)* to describe Cloud concerns related to the application in a Cloud-agnostic way.
- *Cloud-Provider Specific Model (CPSM)* to describe the Cloud concerns needed to deploy and provision the application on a specific Cloud.

Within MODACloudML, CloudML is exploited both at design-time to describe the deployment of application components on Cloud resources as well as the provisioning of these resources at the CPIM and CPSM levels, and at run-time to manage the deployed applications. As a result, CloudML model encompasses runtime information such as IP addresses, Cloud resources ids and statuses. As a part of MODACloudML, CloudML interacts with CCIM models describing the application to be

deployed as well as models exploited for data migration and QoS optimisation and performance analysis:

- *Data Model*: describes the main data structures associated with the application to be built. It can be expressed in terms of typical ER diagrams and enriched by a meta-model that specifies functional and non-functional data properties. At the CPIM level, this model refines the CCIM data model to describe it in terms of logical models. At the CPSM level, it describes the data model based on the specific data structures implemented by the Cloud providers.
- *QoS Model*: includes QoS properties (e.g., response time) at the application level as well as QoS properties of Cloud resources in both a provider-independent (CPIM level) and a provider-specific (CPSM level) way. It includes cost information, thus offering the possibility to estimate an upper-bound for application costs.
- *Monitoring rules*: control the execution of specific software artefacts, including components and data assigned to specific resources. They are used to indicate to the run-time platform the components to be monitored.

CAML

The Cloud Application Modelling Language (CAML) [69] addresses scenarios of services migration to Cloud by introducing concepts that enable not only technical-related information to be captured (e.g., Cloud services and performance characteristics) but also business related ones (e.g., the costs of such services). Particularly the technical-related information is exploited in the refinement of deployment models towards the selected Cloud environment. For this reason, CAML extends UML *to reverse engineer models from software artefacts* [69].

As UML's standard deployment language does not provide support to modelling concepts specific to Cloud environments, CAML includes UML profiles to facilitate capturing environment-specific information in the models, which benefits both reverse-engineering and forward-engineering.

Therefore, CAML realises CloudML as a UML internal DSL based on lightweight extensions to the deployment viewpoint of CloudML in form of a library and profiles. UML profiles in CAML capture environment-specific information for a number of well-known Cloud environments, e.g., Amazon AWS, Google Cloud Platform, and Microsoft Azure, have been introduced. These Cloud environment profiles can be refined by using the so-called *meta-profiles* that allow detailing cross-cutting technical-related information (e.g., the performance of a virtual machine) and business-related information (e.g., the upfront and hourly costs of a virtual machine). Capturing domain knowledge in UML profiles allows for a clear separation between CPIM and CPSM abstraction levels and applying the UML profiles to CPIM in its refinement towards CPSM.

TOSCA

Topology and Orchestration Specification for Cloud Applications (TOSCA) specification [70] is an open standard that provides a language to describe service components and their relationships.

TOSCA defines a Cloud application as a service and it allows defining the topology of this service as well as its orchestration. TOSCA defines a meta-model for defining IT services that can represent a Cloud application.

The *topology template* defines the structure of a service and *plans* define the process models used to manage a service during its lifecycle. A topology template consists of a set of *Node Templates* and *Relationship Templates* that together define the topology of a service where the Node Template is a component of the whole service. A Node Template is an instance of a *Node Type*, which defines

properties of this component node and operations (via *Interfaces*) used to manage the component node. The Node Type also outlines the capabilities and requirements of the component of a service. These features can be used to express that a component (node) requires certain capabilities provided from other component or to express that a component has requirements over the deployment environment. The requirements and capabilities can optionally be connected via Relationship Templates to indicate that a specific requirement of one node is fulfilled by a specific capability provided by another node [70].

The Service Template is defined using XML Schema 1.0 specification. The Plans are defined as process models and the TOSCA specification relies on existing languages such as BPMN [52] or its execution variant BPEL [71] for that purpose.

TOSCA specification also defines an archive format for modelled Cloud applications: CSAR (Cloud Service ARchive). This archive will include beside the modelled service template of the Cloud application, the deployment and implementation artefacts that are needed in a certain environment (such as a deployment environment).

2.4 Risk assessment in multiCloud

2.4.1 Graph-based risk analysis

With the raising of cybersecurity and privacy awareness, multiple approaches are emerging to assess system cyber risks as a means to both tackle the concerns from the early design and try to keep the system controlled at runtime.

In the last decades multiple graphical methods for the analysis of attack and defence scenarios have emerged. A comprehensive survey by Kordy et al. [72] is available which compares all these formalisms. Threat logic trees introduced by Weiss in 1991 pioneered the graphical attack modelling techniques. Since then, most of the literature focuses on directed acyclic graphs (DAG)-based approaches mainly because they do not suffer from the state space explosion problem, which is a drawback of methods using graphs with cycles.

Two main trends can be distinguished in the field of threat analysis using directed acyclic graphs (DAGs): models that derive from or extend attack trees (AT), as the one followed in this Thesis, and models based on Bayesian networks.

Recently, pushed by the need of continuous assessment of threats which requires dynamic adaptation of defences in networked systems, Bayesian networks are gaining adepts as they allow to reason about network states and the causal dependencies of state transitions. One of the most prominent approaches for dynamic risk management using Bayesian networks is the work of Poolsappasit et al. [73]. Their threat modelling approach combines asset identification, system vulnerability and connectivity analysis, as well as mitigation strategies. The work focuses on likelihood of attacks rather than other risk factors such as impact or costs. Xie et al. [74] also use Bayesian networks for security risk analysis of networked systems relying on runtime observations from intrusion detection systems to evaluate security risks. Dantu et al. [75] approach for security risk management also relies on Bayesian networks that capture the influence of attacker profile on risk estimation. However, in all these works the focus is on network attacks rather than system attacks, as the ones studied in this Thesis.

In addition, the full potential of Bayesian networks is realised when conditional probabilities of attack events are known together with the pre-conditions and the order of the network state transitions, which is not always applicable to system domain, particularly when the system is composed of multiple services and infrastructures as in multiCloud. Furthermore, there is limited

dedicated tool support to the analysis of Bayesian networks for security [72], which is not the case of attack trees-based methods.

In any case, both trends are not opposite to one another but in fact they can converge as demonstrated to Qin and Lee [76], who proposed a conversion of an attack tree to a Bayesian network by adding dependency relations between attack tree nodes and conditional probability values that assume an order exists between actions in the nodes connected by AND logic gates.

Therefore, this Thesis proposes the use of Attack Defence Trees (ADT) to reason on the initial estimation of system risks in multiCloud scenarios where limited or none information is usually available about the order of the attack events or about the possible effects that some attack events may have on others, since they may target different parts of the composed system deployed on different providers. In this work, the refinement of risk estimation at system operation is also supported by ADTs, which could be enhanced in the future with Bayesian networks fed with inputs from system continuous monitoring and threat intelligence.

2.4.2 ADT-based risk assessment

Since originally proposed by Schneier [77], Attack Trees (AT) have been extensively studied as an easy-to-understand, reusable and effective formalism to analyse security threats by focusing on how potential attackers may try to attack systems. Attacks against a system are modelled in an acyclic tree structure where the root node represents the attack main goal, the branches in the tree represent the different paths an attacker can follow to achieve the main goal and leaf nodes represent elementary attack events. Branches in the tree are formed by logic OR gates that represent alternative ways to fulfil a goal, and AND gates that model conjunctive sub-goals which all need to be fulfilled in order for the attack to be successful.

Attack Defence Trees (ADT) also known as Attack Countermeasure Trees (ACT) introduced by Kordy et al. [78] extended the AT concept by adding to the attack model information on possible defences or countermeasures that the defenders of the system may adopt to try to prevent the success of the attack.

In order to aid in the quantitative evaluation of how attack and defence parameters may impact on the main attack goal, ADTs can be enriched by attribute decoration to both attack and defence nodes with different techniques such as Amoroso [79], Mauw and Oostdijk [80], Buldas et al. [81], Edge et al. [82], and Wang et al. [83]. These analyses are usually aimed at quantitative reasoning on different attack-defence scenarios towards informed decision-making of countermeasures [84] [85]. ADTs have shown to be efficient in this purpose in deep studies such as those offered by Henniger et al. [86] on vehicle communications systems, Abdulla et al. [87] on the GSM radio network, Byres et al. [88] on SCADA systems to name a few.

All these previous works considered individual attribute domains and some of them studied the derived attributes like risk as well. Salter et al. [89] (probability, impact, cost, severity, skill level, consequence), Edge et al. [82] first introduced defence cost, Byres et al. [88] introduced detectability of the attack and difficulty or skill level, attack time, Buldas et al. [81] and later Jürgenson and Willemson [90] proposed different methods to compute the expected outcome and expected penalty of the attacker, Fung et al. [91] proposed a metric of difficulty level to compute the scenario survivability and Roy et al. [85] studied the defence cost, attack impact, risk and Return on Investment in Attack Defence Trees. An extensive survey of quantitative attributes in ATs and ADTs can be found in [92]. Still, none of them has studied the quantitative analysis of ADTs with risk as the fundamental attribute to consider. As it will be shown, in our methodology we propose an algorithm to compute the risk vector in each of the ADT tree nodes and conclude that it is

necessary to evaluate first both individual attributes (probability, impact, cost) and the derived risk attribute in the leaf nodes to propagate the risk attributes to ascendant nodes.

Moreover, to the best of author's knowledge, no ADT-based risk assessment methodology has previously studied the need of multiCloud applications to identify the relationships between potential attacks (and their defences) over different system components, and none allows to evaluate the overall system risk sensitivity vs. risk sensitivity on specific component. As it will be demonstrated later, the approach proposed in this Thesis offers a solution for this.

2.4.3 Risk assessment in Cloud

The existing literature around Risk assessment in Cloud and multiCloud applications from the approach followed in this Thesis is limited.

There are many methods for IT system threats classification which aid in the identification of potential issues and attacks to the system. One of the most well-known methodologies is STRIDE [93] which classifies security threats in six categories: Spoofing identity, Tampering, Repudiation, Information disclosure, Denial of service and Elevation of privilege. DREAD [94] is a successor of STRIDE specialised in multi-stack applications. However, identification of threats is only a part of the risk assessment where probabilities of threat materialisation and their costs need to be considered as well.

There are many qualitative risk methodologies such as OCTAVE [95] and CORAS [96] which do not serve the purposes of this Thesis on trying to quantitative evaluate system risk sensitivity in different threats and defence scenarios. Multiple quantitative risk assessment methodologies exist too not specifically oriented to Cloud systems, such as ISRAM [97] and the OWASP Risk Rating Methodology [98]. The later includes a general-purpose risk severity rating scheme which is widely used in IT systems. Another widely adopted approach for threat likelihood estimation is the NIST's Common Vulnerability Scoring System (CVSS) [99]. As it will be described in Section 3.4, both CVSS and OWASP are complementary to this Thesis as they would allow to produce an initial estimation of the probability and impact of the threats modelled in the ADT.

M. Pasha et al. [100] recently conducted a thorough analysis of multiple risk management approaches with a focus on large-scale software systems. Particularly for Cloud, we find the QUIRC [101] risk assessment framework which defines six key Security Objectives (SO) for cloud platforms (Confidentiality, Integrity, Availability, Multi-trust, Auditability and Usability), and proposes to map the typical attack vectors and events to these categories and then estimate their probability and impact. Though the method could aid in the initial estimation of these factors, it does not consider cost in the equation, which is a significant limitation compared to this Thesis work.

Djemame et al. [102] proposed a risk assessment framework for Cloud but it was focused on performance risks of infrastructure providers and not on system security. There are also some initial attempts to continuous risk assessment in multiCloud, which also promote the evaluation of system risks in a continuum spanning Development and Operation phases of the engineering process. Most significantly, the approach followed by Gupta et al. [103] and later by Victor et al. [104] stands out, which is based on STRIDE classification of threats and characterises the risk severity by using the OWASP Risk Rating Methodology [98]. Even though, similarly to this Thesis, the approach is also oriented to risk-based Cloud services selection, it has some limitations compared to the method proposed in this Thesis. The method does not include the costs of attack and defences in the risks computation and does not study the conjunctive and disjunctive relationships between the threats against the system assets nor consider the risk minimisation weights of the defences applied to the assets. The overall system risk is computed as an average of the risks of all the threats identified,

which is a simplistic approach to risk propagation in multi component systems such as multiCloud. Hence, the Attack Defence Trees (ADT)-based method proposed herein significantly advances the state of the art.

2.5 Cloud Security Service Level Agreements and Privacy Level Agreements

This section analyses the state of the art in formalisms to express security and privacy features of Cloud services and Cloud-based services as part of the Service Level Agreements between the service provider and the service consumer.

The standard ISO/IEC 20000-1 [105] defines a Service Level Agreement (SLA) as a *documented agreement between the service provider and customer that identifies services and service level objectives* (SLOs). The agreed performance is described in terms of Service Level Objectives (SLOs) or target levels for the service capabilities. The SLOs are usually expressed in terms of metrics that unambiguously express the capability levels guaranteed in the agreement. With the terms *Security SLA* and *Privacy SLA* or *Privacy Level Agreement* (PLA) we therefore respectively refer to the agreements that specify *security level objectives* and *privacy level objectives* offered by a service, which can be considered as part of an overall SLA or as complementary to agreements on other service level objectives, such as quality or performance SLOs.

In the Cloud Computing context, a Cloud SLA is a contractual agreement between the Cloud Service Provider (CSP) and the Cloud Service Customer (CSC) that identifies services and cloud service level objectives [106], i.e. it specifies the grants (in form of SLOs) offered by the consumed Cloud service. The Cloud Security SLA and Cloud PLA would express respectively the *security policy* and *privacy policy* of Cloud services offered to CSCs. And a (multi)Cloud-based system would therefore offer a Security SLA and PLA that depend on the Cloud Security SLA and Cloud PLA of the Cloud services it uses.

2.5.1 Security Level Agreements in Cloud

The approach almost universally followed to define guarantees for users of a service is the introduction of Service Level Agreements (SLAs). An SLA is a formal agreement between a service provider and its end user that describes functional and non-functional aspects of the provided target service, together with clearly defined responsibilities of the involved parties.

The most well-known machine-readable SLA models are the Open Grid Forum's Web Services Agreement (WS-Agreement) [107] and IBM's Web Service Level Agreement (WSLA) [108]. The WS-Agreement specification proposes a domain-independent and standard way to create SLAs while its predecessor WSLA seems to be deprecated.

SLAs appear as a successful method to guarantee common Quality of Service parameters, like availability and performance indicators. As stated in many works on the subject, such as Kandukuri et al.'s [109], in order to deal with security requirements in the Cloud ecosystem, SLAs should be actually used to define target service security parameters.

Security Service Level Agreements (often named in short SecSLA), are recognized as a promising way to model security issues between Cloud Service Providers and their users. ENISA in [110] has also identified the importance of SecSLAs in the Cloud Computing field, pointing out that, in many circumstances, customers are not aware of many acquired services security aspects.

As introduced by Almorsy et al. [111] and by Luna et al. in [112], the current dearth of reasoning techniques on Security SLAs is preventing the diffusion of these approaches in production environments. Nevertheless, currently, many efforts are being made to fill this gap. For example,

Luna et al. in [113] aim to outline techniques to quantitatively reason about Cloud Security SLAs, defining security metrics and a proof of concept semi-automated framework in order to assess Cloud security of different providers.

Several European projects have worked or are working in this subject focusing mainly on SecSLA negotiation [44], the creation of a security-aware SLA based language and related Cloud security dependency model and on the accountability for Cloud-based services [43].

The use of Cloud SLAs has been significantly explored in the last years with the aim of increasing trust in Cloud systems and facilitating their adoption. Recent EU-funded projects such as SPECS [45], SLA-READY [115] and SLALOM [116] and guidelines such as those by Cloud Security Alliance (CSA) [117] and Cloud Standards Customer Council (CSCC) [118] have significantly advanced in Cloud SLA reference models and Cloud SLA life-cycle management systems.

Security Control Frameworks are widely adopted tools used to identify the security controls required to ensure the protection of an ICT system. A security control is a safeguard or a countermeasure prescribed to protect a system and meet a set of defined security requirements. Security Control Frameworks offer a structured list of security controls that help a security expert to select the checks to perform in order to guarantee the respect of security requirements of a given system. Example of such Control Frameworks are: the NIST SP 800-53 Rev. 5 [119], the ISO/IEC 27001 [120], the ISO/IEC 27002 [121], ISO/IEC 27701 [122] for privacy; and the frameworks addressing particularly Cloud related security controls such as ISO/IEC 27017 [6] and ISO/IEC 27018 [7], and the Cloud Control Matrix (CCM) by Cloud Security Alliance [123]. These frameworks are explained in detail in Section 2.6.9.

2.5.2 Privacy Level Agreements in Cloud

Privacy Level Agreements (PLAs) are intended to describe a service *privacy policy* in form of a collection of *privacy controls* offered by the service. In this sense, they are similar to Security SLAs but focused on the controls that guarantee the privacy-respectful behaviour of the service or application.

With the advent of General Data Protection Regulation in May 2018, the PLAs have gained importance as they may serve as the formal statement of the GDPR-compliant behaviour of the service. Just like Security SLAs with respect to security, PLAs facilitate the rationalisation of the capabilities offered by the service with respect to privacy, as well as the automation of the assessment of such capabilities.

With regards to the focus of this work, GDPR-oriented PLA metamodels can already be found in the literature [124] [119] [125]. For Cloud services, standard privacy control definitions are offered by privacy control frameworks such as ISO/IEC 27018 [7] for public Cloud PII processors and the PLA for Cloud services by the Cloud Security Alliance (CSA), named the *Privacy Level Agreement Code of Practise* (PLA CoP) [126].

The CSA's PLA CoP is published as part of their *Code of Conduct (CoC) for GDPR Compliance* and it includes a PLA Template intended to facilitate the declaration of the level of personal data protection a Cloud provider offers to its customers. Following the template, the PLA collects the privacy and security provisions implemented by the CSP acting as data controller or data processor (depending on the case) in a structured way in form of privacy control list. The CSA’s PLA Template defines a total of 94 privacy controls that CSPs acting as data controllers and/or data processors would specify in their privacy policy. Therefore, this can be used as reference for CSPs to grant transparency on the controls applied in their services to protect personally identifiable data.

2.5.3 Service Level Agreements for multiCloud

In general, multiCloud-based applications have their components deployed in or their components use a priori independent Cloud services. Following this definition, federated Cloud-based and hybrid Cloud-based applications fall in the category of multiCloud applications too. Therefore, multiCloud applications are Cloud Service Consumers (CSC) that can be considered as the composition of individual components that exploit Cloud resources in diverse models (IaaS, PaaS, SaaS). The challenge is therefore the computation of the SLA offered by the multiCloud application to its customers as a function of how the components are deployed, the type and number of Cloud services they use, the relationships among the Cloud services and among the components themselves and the SLAs offered by each party, i.e. components and Cloud services.

State of the art techniques of SLA composition are limited and mainly focused on reliability and performance controls using different techniques that range from ontology-based techniques [127] to functional service composition techniques [128]. In all these previous works the focus was on SLAs that address functional and performance requirements (e.g. response time, MTTR, etc.) and no security or privacy levels nor controls were studied, hence they can hardly be reused in security and privacy context. In our work we propose a common methodology to tackle with compositions that can deal with both security and privacy types of policies or SLAs.

The approach promoted in this Thesis to Security SLA composition builds upon Rak's [129] method for security SLA composition for multiCloud applications and extends it to take into account privacy and joint controls as well. And most importantly, the methodology proposed herein takes into account the control delegation relationships between the components for the different types of controls (common, system-specific or hybrid controls).

Furthermore, no previous method exist, not even Rak's [129], that suggest a way to compute the Service Level Objectives (SLO) that can be declared in the composed Application SLA. In this Thesis we propose a technique to calculate the SLOs of the controls in the Application SLA based on the SLOs granted by individual components.

2.6 Standards and regulations on Security, Privacy and SLAs for Cloud applications

This section summarises the main standards and regulations that impact the Cloud and multiCloud application security and privacy landscape as well as those around the formalisation of controls for system assurance. For a complete survey on Cloud Computing standards see [130].

2.6.1 EU Cloud Computing Strategy

Cloud Computing represents one of the key areas identified by the European Community to stimulate growth and create jobs. Understanding the economic impact of the cloud and the opportunity it creates, the EC devised a European Cloud Strategy in [5], to promote the rapid adoption of cloud computing in all sectors of the economy with the aim of boosting productivity.

As a result of an analysis of the overall policy, regulatory and technology landscapes for the Cloud, and considering a wide consultation with stakeholders to identify ways to maximise the potential offered by this technology, the strategy adopted by the EC for "Unleashing the potential of cloud computing in Europe" [131] defined cloud computing as: "*the storing, processing and use of data on remotely located computers accessed over the Internet".* The Commission therefore has intensively worked at enabling and facilitating faster adoption of Cloud Computing, which can cut ICT costs, and when combined with new digital business practices, can boost productivity, growth, and jobs [132].

To achieve this objective, the EC proposed three key actions [131]:

- Unify the variety of cloud standards, to promote interoperability, data portability and reversibility, and also to enhance trust in cloud computing services by recognising at EU-level technical specifications in the field of information and communication technology for the protection of personal information.
- Identify safe and fair contract terms and conditions between stakeholders and clients (consumers and SMEs, and also SLAs between larger corporations and public authorities).
- Establish a European Cloud Partnership, with the participation of public authorities and industry, to stimulate the take-up and effective use of cloud computing, particularly by European public sector.

With this initiative, the EC expects to deliver a net gain of 2.5 million new European jobs, and an annual boost of €160 billion to EU GDP (representing around 1%), by 2020.

2.6.2 GDPR

The entry into force of the European General Data Protection Regulation (Regulation (EU) 2016/679 [8], from now on GDPR) in May 2018 has definitively increased the concerns on better assuring privacy measures adopted by software systems. Privacy capabilities are intrinsically related to security capabilities in personal data processing information systems. Even the GDPR itself requires that personal identifiable information (PII) shall be *processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organisational measures ('integrity and confidentiality').* (Article 5.1(f)).

Therefore, there is a need to follow a holistic approach to risk assessment that addresses both privacy and security threats. This is even more challenging in multiCloud-based systems because of the need of controlling not only system components' own risks but also those of the Cloud providers. Security, privacy and data protection continue to be major barriers to Cloud adoption [133]. The users' concerns on security and privacy of Cloud systems strive from the lack of trust, visibility and auditability of the privacy and security controls the Cloud providers offer in their services.

Since the arrival of GDPR, solving these issues is an urgent necessity for Cloud consumers and providers acting as data processors or controllers, because the personal data processing principles in Article 5.1(a) 'lawfulness, fairness and transparency' and Article 5.2 'accountability' require systematic privacy assessment and evidence collection for assurance and transparency towards data subjects, collaborators in processing and supervisory authorities.

The work of this Thesis contributes directly to facilitating the identification, formalisation, transparency and assurance of privacy properties of multiCloud applications and hence, aligns with the principles of GDPR [8].

2.6.3 ETSI

In the year 2012, the European Commission asked the European Telecommunications Standards Institute (ETSI) *to coordinate with stakeholders in the Cloud standards ecosystem and devise standards roadmaps in support of EU policy in critical areas such as security, interoperability and data portability, and to analyse end users' needs and the relationship with open source*. The Cloud Standards Coordination (CSC) group was launched within ETSI in collaboration with relevant players and finalised in January 2016 with the publication of the following reports (all available

from the website of the CSC [134]): *Cloud Computing users' needs*, *Standards and Open Source*, *Interoperability and Security*, *Standards Maturity Assessment*.

Among its works it is also remarkable for this Thesis the ETSI TR 103 125 V1.1.1 (2012-11) technical report [135] - Definition of main roles in cloud SLA and recommendations for SLA specification, though it focuses on Quality of Service (QoS) metrics in SLAs for Cloud rather than considering security and privacy measures.

2.6.4 ENISA

There are a number of surveys and technical reports issued by the European Network and Information Security Agency (ENISA) that establish the foundations of cloud security best practices. The most relevant for this Thesis can be summarized as follows:

- *Cloud Computing Risk Assessment* [136] report provides an in-depth and independent analysis of information security benefits and key security risks of cloud computing. The report provides also a set of practical recommendations to tackle those risks.
- *Survey and analysis of security parameters in cloud SLAs across the European public sector* [137]. *This survey gives a snapshot of how the IT officers in the European public sector are currently managing the security aspects of these service contracts. The survey produced full responses from 117 IT officers, from 15 different EU countries and all layers of government, who are either involved in procuring cloud or IT services or responsible for managing the SLAs*.
- *Procure Secure: A guide to monitoring of security service levels in cloud contracts* [138]. This is a *practical guide aimed at the procurement and governance of cloud services. This guide provides advice on questions to ask about the monitoring of security.*
- *Exploring Cloud Incidents* [139] report provides an overview of the current status of the forensic analysis techniques and processes of cloud incidents.
- *Security and Resilience in Governmental Clouds* [140] is a guideline to aid public bodies in the definition of their requirements for information security and resilience when evaluating private and public cloud computing delivery models;

2.6.5 NIST

Since beginning of Cloud Computing era, the National Institute of Standards and Technology in the U.S. has also issued a number of reference documents and guidelines that set up the baseline context in which this Thesis work is founded. As it will be shown in the different parts of this Thesis, the work fully adheres with all these NIST standards below which are compatible with ISO/IEC standards with respect to the areas considered in this work:

- *NIST Cloud Computing Reference Architecture (NIST 500-292)*: In 2011 NIST published this Reference Architecture specification [16] which set a common understanding on Cloud service models and roles.
- *NIST Cloud Computing Service Metrics Description (NIST SP 500-307)*: In 2015 NIST published this guide [141] on Cloud metrics which introduced a reference model for Cloud Service Metrics which captured the higher-level concepts of the metric definition for a specific cloud service property, such as service uptime.

- *NIST Cybersecurity Framework v1.1*: In 2018 NIST published the second version of the *NIST Framework for Improving Critical Infrastructure Cybersecurity* [24] which aids organisations to adopt business drives to guide cybersecurity activities and to apply risk management best practices to improving security and resilience. The framework core proposes five *functions* that organise cybersecurity activities at their highest level: Identify (cybersecurity risks), Protect (through implementation of safeguards), Detect (cybersecurity events), Respond (to detected cybersecurity incident), and Recover (from the incident).

  The multiCloud framework proposed in this Thesis address the first four functions of NIST Cybersecurity Framework.

- *NIST Risk Management Framework for Information Systems and Organizations (NIST SP 800-37 Rev. 2)*: This report [142] describes the Risk Management Framework (RMF) proposed by NIST which aim is to help ensure that, throughout the system development life-cycle, information systems, organizations, and individuals are appropriately protected, and that decision makers have the information needed to make risk-based decisions regarding the operation or use of systems or the provision of controls.

  Hence, the NIST RMF promotes risk-based decision making by understanding the security and privacy posture of information systems. In the NIST RMF the security and privacy posture of information systems and organizations is determined by continuously assessing and monitoring system-specific, hybrid, and common controls. This is exactly the approach followed in this Thesis, and thus, this standard is fully supported in our methodology.

2.6.6 ISO

The ISO/IEC JTC 1/SC 38 Cloud Computing and Distributed Platforms is a standardization subcommittee which is part of the Joint Technical Committee ISO/IEC JTC 1 of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). This subcommittee has published multiple standards around Cloud Computing, Distributed Platforms, and the application of these technologies. In the following we provide the list of the most relevant, all available from the ISO website [143].

- ISO/IEC 17788:2014 Information technology -- Cloud computing -- Overview and vocabulary.
- ISO/IEC 17789:2014 Information technology -- Cloud computing -- Reference architecture. This document specifies the Cloud Computing Reference Architecture (CCRA) which is used in this work to provide a reference of the different roles that the stakeholders of the presented workflow may play.
- ISO/IEC 19086-1:2016 Information technology -- Cloud computing -- Service level agreement (SLA) framework -- Part 1: Overview and concepts.
- ISO/IEC 19086-2:2018 Cloud computing -- Service level agreement (SLA) framework -- Part 2: Metric model.
- ISO/IEC 19086-3:2017 Information technology -- Cloud computing -- Service level agreement (SLA) framework -- Part 3: Core conformance requirements.
- ISO/IEC 19086-4:2019 Cloud computing -- Service level agreement (SLA) framework -- Part 4: Components of security and of protection of PII. Specification of the security and protection of personally identifiable information components, SLOs and SQOs for cloud service level agreements (Cloud SLA) including requirements and guidance.

- ISO/IEC 29101:2018(en) Information technology -- Security techniques -- Privacy architecture framework.

Other ISO standards relevant for the Thesis work around the risk management aspects of multiCloud applications are:

- ISO Guide 73:2009 Risk management – Vocabulary
- ISO 31000:2009, Risk management – Principles and guidelines

The risk management techniques proposed herein adhere to both standards.

The ISO standards relevant on the security and privacy controls to use as defences against system risks and to express protection levels within Service Level Agreements are described later in Section 2.6.9.4.

2.6.7 Cloud Security Alliance (CSA)

The Cloud Security Alliance (CSA) is a leading private organisation that has the mission to promote the use of best practices for cloud computing security. There are 25 working groups in CSA looking into cloud standards, certification, education and training. The main initiative for this Thesis that is worthy of further examination is the CSA Governance, Risk and Compliance stack (GRC) that delivers a toolkit for assessing both private and public clouds against industry established security best practices. The GRC includes the following initiatives:

- The *Cloud Controls Matrix (CCM)* [123] security control catalogue for Cloud Services is one of the standards for cloud security assurance and compliance. Due to its relation to the SLA creation and assessment in this Thesis its contents are detailed in Section 2.6.9.
- *Privacy Level Agreement Code of Practise* (PLA CoP) [126] already mentioned in Section 2.5.2 and detailed in Section 2.6.9.
- The *Consensus Assessments Initiative Questionnaire (CAIQ)* [144] is a self-assessment questionnaire for Cloud Service Providers (CSP) to express the Cloud security controls they offer. This is described in Section 2.6.9 as well.
- The *STAR Registry* [145] gathers the self-assessments by popular CSPs following the CAIQ questionnaire. STAR is described in Section 2.6.9 too.

2.6.8 CSCC

Cloud Standards Customer Council [146] is an end user association that works on promoting and easing the adoption of Cloud technologies. They have been active since the inception of Cloud Computing with the aim to offer contributions in the areas of *standards, security and interoperability*.

Their works on cloud security and SLAs are those that are of main importance for this Thesis, such as: *Practical Guide to Cloud Service Agreements V3.0* [117], *Security for Cloud Computing: 10 Steps to Ensure Success V3.0* [147] and *Cloud Security Standards: What to Expect & Negotiate* [148], which paved the path towards formalised assessment and assurance of security measures in Cloud.

2.6.9 Standard control frameworks

Several control families can be used to analyse and formally express the security and privacy assurance controls of ICT systems in a standard way, for example in Service Level Agreement (SLA) specifications. These include but are not limited to NIST SP 800-53 Rev. 5 [119], Cloud

Security Alliance's CCM [123], ISO/IEC 27017 [6], ISO/IEC 27018 [7], ISO/IEC 27001 [120], ISO/IEC 27002 [121], ISO/IEC 27701 [122], etc. These catalogues aid, with higher or lower level of details in technical guidance, in the formal specification of required and/or offered (provided) capabilities of the ICT system as a whole or of its individual components and of the organisation providing them.

Some of these catalogues where already mentioned in previous sections as part of the reference standardisation works offered by international standardisation bodies. In the following we provider insights on their contents.

2.6.9.1 NIST SP 800-53 Revision 5

The most complete and detailed standard security control family, the NIST Security and Privacy Control Framework NIST SP 800-53 Rev. 5 [119], provides a comprehensive collection of security and privacy controls that an organisation and/or service can offer. The latest public revision of the standard, Revision 5 Draft, collects 912 fine-grained standard controls and is freely available to the public, and therefore it was selected in this Thesis as the standard define the controls of the multiCloud application. This revision extends the previous version of the framework and defines, in addition to *security controls*, *privacy controls* that are specifically devoted to meet privacy requirements and to manage the privacy risks in an organisation, and *joint controls* that can meet privacy and security requirements.

While in its predecessor NIST SP 800-53 Revision 4 only identified security controls, a total of 160 privacy related controls are identified by NIST SP 800-53 Rev. 5 from 12 different areas or groups (named *control families* by NIST), namely: AC - Access Control, AT - Awareness and Training, AU - Audit, CA - Continuous Assessment, CM - Configuration Management, CP - Contingency Planning, IA - Identification and Authentication, IP - Individual Participation, IR - Incident Response, MP - Media Sanitization, PA - Privacy Authorization, PL - Planning, PM - Project Management, RA - Risk Assessment, SA - System and Services Acquisition, SC - System and Communications, SI - System and Information Integrity. From these, 59 controls are *privacy* controls and 101 are *joint* controls.

A deeper analysis of this control framework will be provided in Section 3.5 when describing the Security SLA and Privacy SLA composition method.

2.6.9.2 Cloud Security Alliance's CCM v3.0.1, CAIQ v3.1 and STAR

The Cloud Controls Matrix (CCM) by the Cloud Security Alliance (CSA) [123] is currently considered as one of the de-facto standards for cloud security assurance and compliance. The CCM is a catalogue or meta-framework of cloud-specific security controls. The 131 controls in the catalogue are mapped to leading standards, best practices and regulations, including those of NIST SP 800-53 Rev. 5 [119].

The CCM groups the controls in 16 control categories, ranging from Application and Interface Security to Threat and Vulnerability Management. It is important to note that the focus of CCM is on security controls rather than privacy controls, though some controls in CCM can support privacy capabilities. However, the number and granularity of controls in NIST SP 800-53 Rev. 5 [119] is significantly higher.

The *Consensus Assessments Initiative Questionnaire (CAIQ)* by CSA [144] offers an industry-accepted way to document what security controls exist in IaaS, PaaS, and SaaS services, offering security control transparency. The latest version CAIQ v3.1 provides a set of Yes/No questions a cloud consumer and cloud auditor may wish to ask of a cloud provider to ascertain their compliance to the Cloud Controls Matrix (CCM) v3.0.1.

The STAR Registry by CSA [145] documents the CCM controls provided by popular cloud computing offerings. This publicly accessible registry allows cloud customers to assess their security providers in order to make the best procurement decisions. The registry data are the results of self-assessment of security controls by CSPs following the CAIQ methodology.

2.6.9.3 Cloud Security Alliance's PLA

The Cloud Security Alliance (CSA) has also published a Privacy Level Agreement (PLA) for Cloud services named the *Privacy Level Agreement Code of Practise* (PLA CoP) [126]. The CSA's PLA CoP is published as part of their Code of Conduct (CoC) for GDPR [8] compliance and it includes a PLA Template intended to facilitate the assertion of the personal data protection level a Cloud Service Provider offers to its customers.

Following the template, the PLA collects in a structured way in form of privacy control list the privacy and security provisions implemented by the Cloud Service Provider taking the role of PII controller or processor.

A total of 94 privacy controls are defined in the CSA's PLA that CSPs acting as data controllers and/or data processors would specify in their privacy policy.

2.6.9.4 ISO/IEC standards for security and privacy controls

The major standards published by the International Standards Organisation [143] about security and privacy controls relevant for the present work are summarised below.

- ISO/IEC 27002:2013 Information technology — Security techniques — Code of practice for information security controls: The ISO/IEC 27002:2013 gives guidelines for organisational information security standards and information security management practices including the selection, implementation and management of controls taking into consideration the organization's information security risk environment(s). The standard is focused on security controls and was published prior to the GDPR [8] entry into force.
- ISO/IEC 27017:2015 Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services: The ISO/IEC 27017:2015 gives guidelines for information security controls applicable to the provision and use of cloud services by providing additional implementation guidance for relevant controls specified in ISO/IEC 27002; and additional controls with implementation guidance that specifically relate to cloud services. The recommendations address both cloud service providers and cloud service customers.
- ISO/IEC 27018:2019 Information technology — Security techniques — Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors: The ISO/IEC 27018:2019 standard establishes commonly accepted control objectives, controls and guidelines for implementing measures to protect Personally Identifiable Information (PII) in line with the privacy principles in ISO/IEC 29100 for the public cloud computing environment. Based on ISO/IEC 27002, the guidelines specified take into consideration the regulatory requirements for the protection of PII which can be applicable within the context of the information security risk environment(s) of a provider of public cloud services.
- ISO/IEC 27701:2019 Information technology — Security techniques — Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management — Requirements and guidelines: The ISO/IEC 27701:2019 standard specifies requirements and provides guidance for establishing, implementing, maintaining and continually improving a Privacy

Information Management System (PIMS) in the form of an extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy management within the context of the organization. The document address PII controllers and/or PII processors processing PII within an ISMS, and while Annex A lists all applicable controls for PII Controllers, Annex B lists all applicable controls for PII Processors.

2.7 Conclusion

The work of this thesis arises at a time when Cloud Computing standards are being defined, and more specifically privacy and security standards. The European Commission itself is promoting the standardization of concepts, taxonomies and models that help in the transparency of cloud services and facilitate their adoption in an environment of trust between providers and customers. The ultimate goal is to boost Cloud Computing as it has proved to be an enormously beneficial technology for the digitalisation of European Industry.

This Thesis contributes to the trustworthiness of Cloud based systems through enabling the development of multiCloud applications which consider security and privacy requirements from the early design and transfer them to operation phase, taking always into account the risks of the system and the needed formalisation and assurance of the security and privacy guarantees in the Security and Privacy Service Level Agreements.

Unsolved issues remain in the state-of-the-art solutions for multiCloud systems in different aspects.

First, none of the existing software frameworks for multiCloud applications development fully supports the assurance of security and privacy properties in this system. Even if some of them are oriented to multiCloud system engineering, the expressiveness of the languages used, the lack of consideration of security and privacy risks, the null observation of security and privacy guarantees or the limited alignment between Development and Operation activities related to security and privacy make them not suitable for the objectives of pursued in this Thesis. Therefore, as shown in next section, the work herein advances in integrated frameworks to create multiCloud applications taking into consideration the security at privacy requirements from the design stages and providing a continuum of security and privacy assurance along the whole application life-cycle to runtime phase.

Second, current modelling languages for multiCloud are focused on deployment and scalability issues but lack expressiveness with regards to the security and privacy features that the application components require from and offer to the other components. This is crucial for understanding security and privacy implications and for arriving to implementation models that enable the deployment of security mechanisms as part of the overall deployment of the application. In this work we develop an advanced security and privacy requirements modelling language which extends the richest language for modelling multiCloud systems at present, the CAMEL language.

Third, major challenges remain about how to address the continuous quantitative risk assessment of multiCloud systems. No solution exist that addresses particularly multiCloud applications, where their multi component nature makes it fundamental to take into account the relationships of defences applied in different system components as well as the role of outsourced services or components in the system risk assessment. Moreover, recent works even promote the continuous security risk assessment in DevOps, but still the evaluation of risks is limited as the costs of attacks and defences are not included. In addition, no methodology exists for multiCloud that enables the selection of protections in the system components based on the analysis of overall system risk sensitivity and costs of defences. This work advances in the risk evaluation techniques for multiCloud DevOps scenarios by making use of attack and defence attribute quantification in ADTs and proposing a holistic risk methodology that includes refinements of the risks assessed at multiCloud system

design once the actual external services are selected for outsourced components and further refinements at operation through continuously monitoring the countermeasures deployed to control risks. The methodology includes also an innovative method for risk-based optimisation of defences to enable the informed decision on security investments in multiCloud which does not have a competitor in the literature yet.

Fourth, a few methods exist for generating security SLAs for multiCloud applications and no method has studied yet the privacy SLAs (PLAs). The maximum exponent of this work is the solution by Rak [129] which initiated the path towards security SLA composition in multiCloud and which we extend in this work to take into account the control (defence) parts' implementation delegations between the components of the multiCloud application when declaring the controls. It is required that the SLA composition builds on top of these relationships as the controls may be implemented not in a single component but in multiple components each of which is responsible for the delivery of different control parts. Moreover, no method exist that includes in the composition the evaluation of the security and privacy Service Level Objectives that can be guaranteed in the overall application SLA. Therefore, the SLA composition considering control implementation delegations and enabling the computation of control levels is also a major innovation brought by our proposal.

Last but not least, no method exist that relates the existing security control (defence) standards with the security mechanisms considered in the formalisation of security requirements in the system deployment model, in the risks assessment and in the security and privacy SLA of in multiCloud applications. The methodology of this Thesis proposes the use of a standard taxonomy for security defences modelled in the system model, in the ADT and in the security and privacy SLA. This definitively contributes to the comparability with external providers' control offerings as well as facilitates the certification of the defences used by the system. In most of the ADT analyses works, the taxonomy of attack and defences used in the tree nodes does not adhere to any security or privacy standard. As a result, comparability, reusability and auditability of defences in ADTs is limited and no alignment with defences stated in the SLA is possible.

In our work we propose to model in ADTs attacks and defences aligned with the well-known and internationally recognised control standard NIST SP 800-53 Rev. 5 [119] which will aid in the auditability and certifiability of defences in the ADTs and in smooth integration with controls in the system deployment model and in the system SLA. The NIST SP 800-53 Rev. 5 control framework was selected over other similar standards such as ISO/IEC 27001 [120] and Cloud Control Matrix (CCM) by CSA [123], because of the granularity and detail of the 912 controls offered, which are classified in security, privacy and joint controls. Furthermore, the NIST standard provides control mappings with other standards such as ISO/IEC 27001 [120] and CCM [123] and ISO/IEC 15408 [149] which easies the translation between them.

3 Integrated DevOps framework for Security and Privacy assurance in multiCloud applications

3.1 Introduction

This section describes the four major research contributions for multiCloud applications brought by this Thesis:

1. The DevOps methodology for Security and privacy assurance.
2. The Security and privacy requirements modelling language.
3. Continuous Risk Management and risk-based optimisation of defences.
4. The Security SLA and Privacy SLA (PLA) composition.

Figure 2 shows how each of the four contributions fits in the multiCloud application life-cycle.

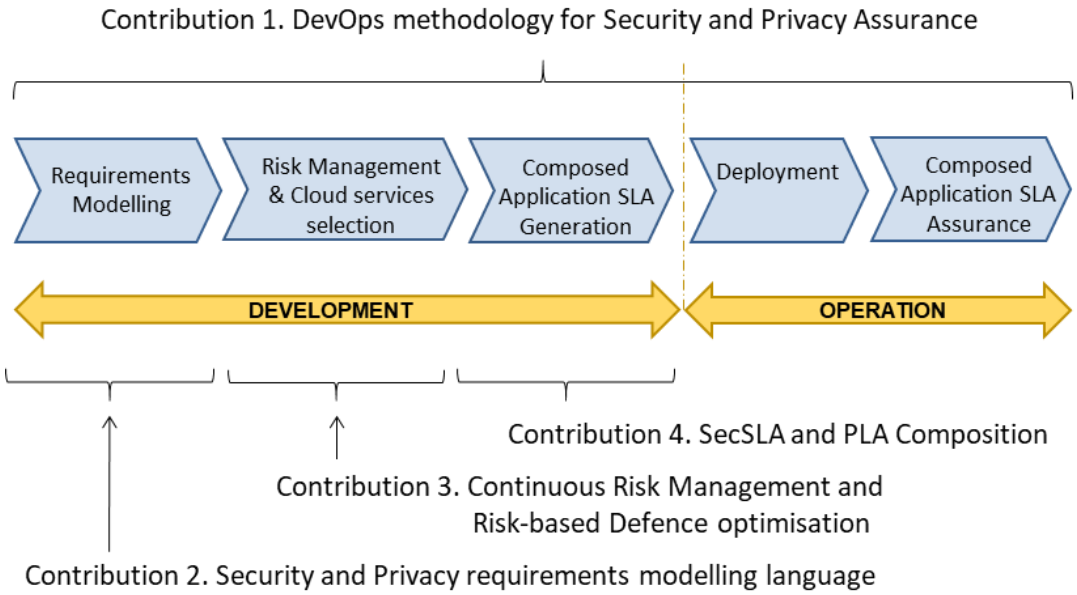


Figure 2: Thesis contributions integrated in the DevOps workflow of multiCloud applications

The first contribution is the DevOps methodology for Security and Privacy assurance of multiCloud applications and it relates to the whole application life-cycle, spanning from Development (Dev) phase to Operation (Ops) phase.

The rest of the three contributions match with one of the steps of the DevOps methodology in the Development phase, following the sequential order of the process in Figure 2.

The Deployment and Composed application SLA Assurance at Operation fall outside the scope of this Thesis. However, as it will be explained in the following subsection, the methods and techniques developed in the Thesis for the Development phase prepare the software engineering artefacts of the multiCloud application so as the Operation activities are possible.

The next subsections provide a detailed description of each of the contributions collected in the Thesis work in the order above, and in Section 4 the validation carried out for each of the parts is fully described.

3.2 DevOps methodology for security and privacy assurance in multiCloud applications

3.2.1 Overall approach

In this section we describe the first contribution of the Thesis as per Figure 2, i.e. the integrated DevOps Methodology for Security and Privacy Assurance in multiCloud.

The proposed solution to holistic security and privacy technical assurance in multiCloud applications involves the integration of preventive measures and reactive measures. While the preventive activities aim at preparing the application and defining its SLA including the offered security and privacy controls, the purpose of the reactive activities is to control the actual fulfilment of the defined SLA.

The approach follows the DevOps paradigm [25] to support all the phases of the security- and privacy-aware life-cycle of multiCloud applications, from application privacy-by-design and security-by-design (including the SLA creation) to deployment on Cloud services selected, and finally continuous assurance of SLA fulfilment at operation. This approach enables multi-disciplinary DevOps teams, which, as explained later, gather together different stakeholders in application life-cycle, to manage security and privacy risks in all the phases of the multiCloud application life-cycle.

Figure 3 illustrates the major four activities of the proposed DevOps methodology which are detailed in the next subsection.

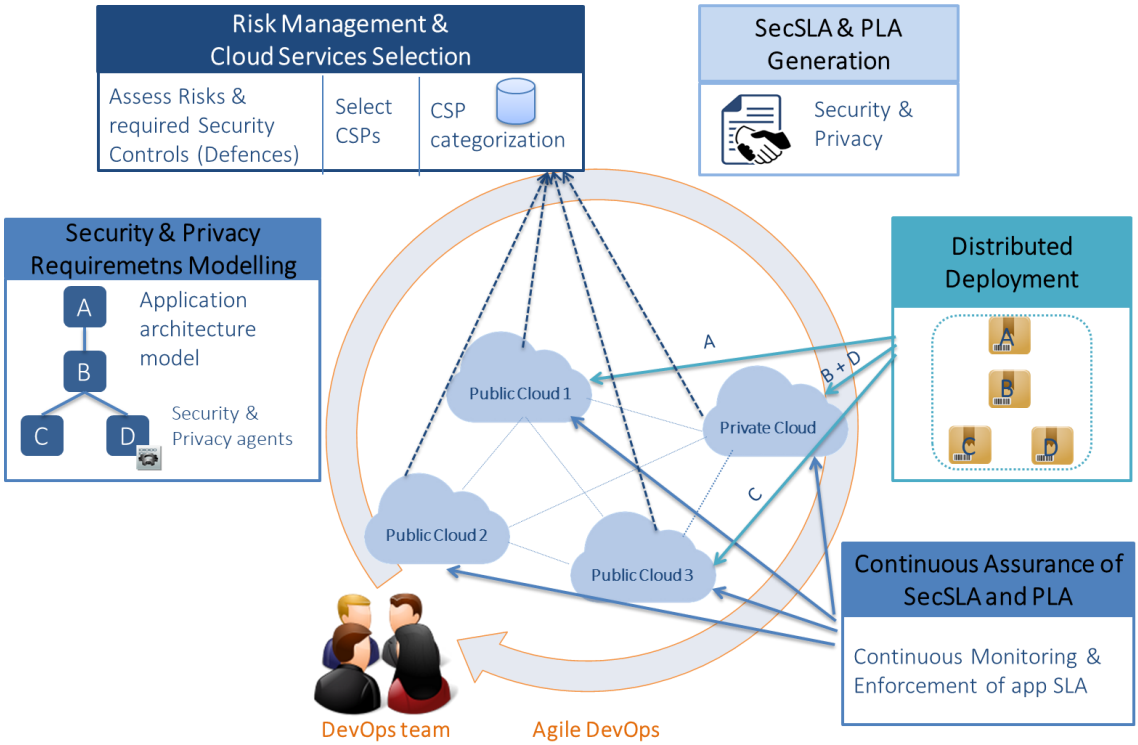


Figure 3: Overall approach of DevOps methodology for Security and Privacy assurance in multiCloud application

The figure represents a multiCloud application with four components, A, B, C and D that need to be developed taking into account their security and privacy properties and those of the overall application when they interact with each other. Then, they need to be deployed in the available

Cloud resources represented in the figure by a Private Cloud and Public Clouds 1, 2, and 3. Once they are deployed, they would be subject to a continuous monitoring so as their secure behaviour is ensured. The proposed DevOps methodology encompasses preventive security and privacy methods at system Development time (Requirements Modelling, Risk Management, SecSLA and PLA generation) with preventive (Monitoring) and reactive methods (Enforcement) at system Operation so as application security and privacy levels hold.

As we see in the figure, the methodology follows an agile DevOps cycle where the early feedback from Operation activities serve to improve Development activities. The major and single actor of the process is the "DevOps Team" which represents a multi-role group of stakeholders in charge of the different responsibilities in the process with the goal of developing, delivering and operating the multiCloud application. The participating roles and responsibilities within the DevOps Team are explained later in Section 3.2.3.

The details of the activities in Figure 3 are formally described in next subsection.

3.2.2 Workflow

The complete DevOps activities workflow proposed involves both Development activities and Operation activities of the multiCloud application life-cycle as shown in Figure 4.

While the last three activities are Operation activities, the first three activities are Development activities that can be considered security- and privacy-by-design practices which prepare the application to be compliant with security and privacy requirements and regulations.

Please note that the step Composed Application SLA Assurance of the methodology in Figure 2 has been refined into two steps Monitoring and Enforcement in Figure 4, to differentiate between security incident detection and reaction details.

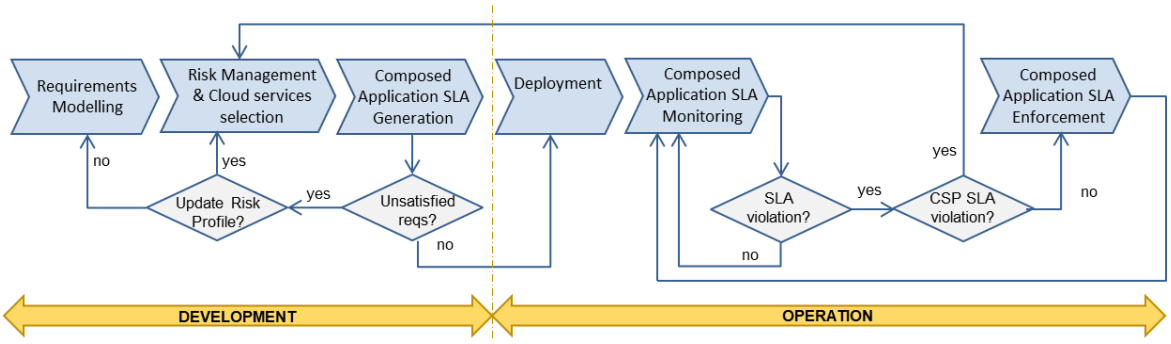


Figure 4: DevOps workflow for security and privacy assurance in multiCloud applications

The whole DevOps workflow proposed is made of six main steps, described in the following:

1. **Requirements Modelling**: The start of the engineering process is the creation of the application model which specifies both the component level architecture of the application, i.e. its Cloud deployment together with the security and privacy requirements. The model is created in a language developed in the Thesis as an extension to the CAMEL language, as explained in Section 3.3. The model is a Cloud Provider Independent Model (CPIM) where Cloud requirements of the components are defined without references to the actual Cloud services that will be selected for the deployment in the next step. The purpose of the model is therefore to define which are the application components, which are their communication relationships, which type and location of Cloud services they need, which are their Cloud resources needs (e.g. size of VM memory, CPU, etc.) and most importantly for our work, which are the security and privacy requirements of the multiCloud

application. These requirements come in form of security and privacy capabilities required and offered by the components. The offered capabilities are protection or enforcement mechanism (agents) that will operate with the components to implement a required security or privacy control.

2. **Risk Management and Risk-driven Cloud Services selection**: In this step a thorough analysis of the system risks is performed where the risks are identified and evaluated both at component (asset) level and at overall composite application (system) level. The result of the risk assessment will drive the selection of the actual Cloud services to use by the application components so as they match the requirements stated in the application model expressed as required Cloud security and privacy controls or protections. The risk profile is the result of the risk assessment process carried out by analysing the threats against the application components and selecting the desired treatments or controls. As described in Section 3.4, the Risk management method proposed in this Thesis enables the optimisation of the controls in the application components by enabling the evaluation of system risks and identification of the risk minimised and the residual risk in different scenarios of attacks and defences.
3. **Composed Application SLA generation**: This step consists in the generation of the multiCloud application SLA that can be offered to its clients. The SLA granted will be computed as the composition of the SLAs of the application components and the SLAs of the Cloud services used after an SLA validation process to learn the actual controls that can be effectively supported. The details of the composition methodology can be found in Section 3.5.
4. **Deployment**: Once the Cloud services to use are selected and the Composed SLA is obtained, the components of the application will be automatically deployed and the Cloud resources initialised and configured as needed. As part of the security and privacy capabilities of the system, the monitoring and enforcement agents to be used together with the components are also deployed and configured in this step. They will be the responsible for controlling at operation that the application behaves as promised in the SLA. The automatic deployment has limited security and privacy specificities and thus it has been left out of the Thesis. However, note that the created in a language developed in the Thesis enables to express at Cloud Provider Independent Model (CPIM) level which monitoring and enforcement agents to deploy with the components and this model would be transformed in the Cloud Service Provider Specific Deployment model.
5. **Monitoring of Composed Application SLA**: The main objective of compliance and security assurance is to make sure that the Composed Application SLA that states the security and privacy guarantees to the customers holds during application provisioning. This is ensured in our approach by continuously monitoring the security and privacy Service Level Objectives (SLOs) through metrics defined in the Composed Application SLA. The monitoring details of the workflow are not part of this Thesis.
6. **Enforcement of Composed Application SLA**: In case actual or potential violations of the promised SLOs are detected, it is necessary to try to enforce the SLOs and take prompt remediation actions to avoid the violation or to recover the security and privacy behaviour as soon as possible. The cause of the violation of the Application SLA may reside in a failing application component (including enforcement agents used) or a failing Cloud service (i.e. the CSP is not fulfilling its Cloud SLA). Depending on the failing SLO, reaction actions may be procedural activities (e.g. the redesign of the application to update the architecture and include in the CPIM model enforcement agents like access control

agents) or automatic enforcement mechanisms supported by the multiCloud application itself (e.g. the activation of a data encrypting component) a or by external systems (e.g. the activation of a vulnerability scanner). The enforcement management in our framework is not part of the present Thesis work.

The agile and DevOps paradigms are achieved in the methodology by two main iteration loops in the workflow. First, at design time the initial CPIM model of the application (in Requirements Modelling) and/or its risk profile (in Risk Management and Risk-driven Cloud Services selection) are revisited by the DevOps team until the Application SLA satisfies all the requirements expressed in both, i.e. until the application architecture and Cloud deployment plan enable to grant a feasible Application SLA that includes only those controls and levels that can be effectively granted after the selection of the Cloud services to use. Second, at operation time, in case a CSP is identified as the cause of the Application SLA violation, in order to solve the situation and replace the Cloud service, a redeployment action is tried which would include a new risk assessment iteration.

As indicated before, the three Operations steps in the workflow are not part of the work of this. Yet, the contributions of the Thesis in the Development phase are security- and privacy-by-design techniques that enable the preparation of the multiCloud application so as Operation activities can support the system security and privacy assurance:

- The Security and privacy requirements modelling enables the creation of models that can be later transformed into Deployment plans that make it possible the distribution of the application components together with security and privacy mechanisms (agents) that enforce the security and privacy requirements at Operation.
- The Continuous Risk Management and the risk-based optimisation of defences enable the identification of the defences required in the application components which, in the case of outsourced components, need to be requested to Cloud Service Providers. Therefore, the selection of CSP for the Deployment is driven by the decisions in the Risk assessment step.
- The composition of the multiCloud application security and privacy SLA allows to know which controls need to be monitored at Operation phase, as they are the promises made to multiCloud application customers.

3.2.3 Actors

The main stakeholders of the proposed workflow for multiCloud application engineering are those that interact either directly or indirectly with it; that is, intervene in one of the following phases of the multiCloud security- and privacy-aware application management lifecycle: development (including design) and operation (including deployment).

According to ETSI Cloud SLA [135], the Cloud Computing Overview and vocabulary ISO/IEC 17788 [150] and the Cloud Computing Reference Architecture (CCRA) ISO/IEC 17789 [151] the parties (i.e. human or legal entities involved in the system) may assume three main roles in Cloud architecture: Cloud Service Customer (CSC), Cloud Service Provider (CSP) and Cloud Service Partner (CSN) (that collaborates with the CSP in service provision). Such roles have sub-roles that specialise roles' activities in the system. The standard ISO/IEC 17789 offers a full definition and map of all the roles and sub-roles typical of the Cloud environment.

In this line, considering the multiCloud application uses multiple services offered by external CSPs, the main parties involved in a multiCloud application provision model are the following, as outlined in Figure 5:

- MultiCloud application Customer or End User (EU): the user or customer of the multiCloud application.

- MultiCloud application Service Provider (multiCloud application SP): the party that offers the multiCloud application services.
- MultiCloud application Service Partner (multiCloud application SN): the party that supports the multiCloud application SP in providing the multiCloud application services (e.g. developers, brokers, etc.).
- Cloud Service Provider (CSP): a party that acts as Cloud Service Provider for the Cloud services used by the multiCloud application. Note that this party does not directly provide any of the functionalities of the multiCloud application but provides Cloud services that support them.

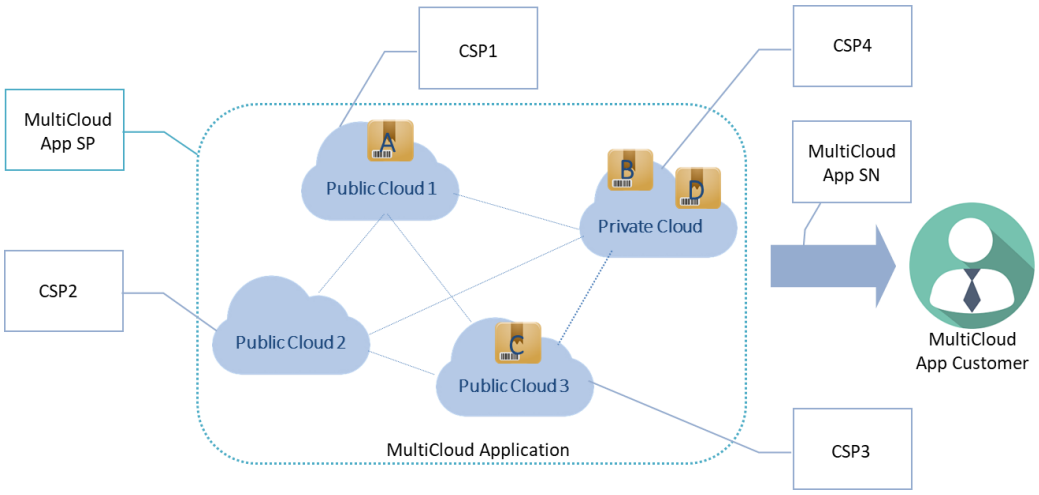


Figure 5: Parties involved in the multiCloud application provision model

As it can be seen in Figure 5, the multiCloud application SP acts as Cloud Service Customer (CSC) of the Cloud Services that the multiCloud application components use and as a Cloud Service Provider (CSP) for the services that the multiCloud application offers to its own customers (the End-users). According to the standards above, other developers and brokers that may take part in the multiCloud application creation and delivery act as Cloud Service Partners (CSN), which are represented, in the above proposed schema, as multiCloud application Service Partner (*MultiCloud application SN*).

Our DevOps framework supporting security and privacy relies on a DevOps paradigm [25] (a name that combines the terms "Development" and "Operations") that emphasises the close collaboration and communication between software developers and other information-technology (IT) professionals in the engineering and provision processes, while automating the process of software delivery and infrastructure changes. According to the DevOps approach, which we consider is the best for a seamless management of security aspects in multiCloud applications, building, testing, and releasing software systems can happen rapidly, frequently, and more reliably.

To respond to the need of a smooth integration of Development and Operation responsibilities in the multiCloud application engineering and provision process, our methodology proposes the “DevOps Team” as the main stakeholder responsible of the multiCloud application development, deployment and execution management (overall lifecycle). In practice, the DevOps Team merges the typical roles of Provider (multiCloud application SP) and Service Partner (multiCloud application SN) in a multiCloud application. As a result, only three main stakeholders can be distinguished in the workflow:

- End User (EU): the customer of the multiCloud application.

- DevOps Team: the party which develops the multiCloud application, supports its deployment and offers the multiCloud application services.
- (external) Cloud Service Provider (CSP): a party that acts as Cloud Service Provider for the Cloud services used by the multiCloud application. Note that this party does not directly provide any of the functionalities of the multiCloud application but provides Cloud services that support them. In this sense, they are *external* Cloud Service Providers whose services are consumed by our multiCloud application.

According to the Cloud Computing roles and sub-roles defined by the ISO/IEC 17789 [151], these stakeholders play different Cloud Computing roles depending on the responsibilities that they assume in the multiCloud application lifecycle.

Table 1 summarizes the possible roles that each stakeholder can play along the workflow.

Table 1: Overview of proposed multiCloud DevOps workflow stakeholders

Stakeholder	Role according to ISO/IEC 17789:2014	Description
End-user	CSC	The end-user who uses the services (functionalities) of the multiCloud application. His only role is Cloud Service Customer. In the workflow proposed, the End-user is the last actor of the supply chain we consider.
(external) Cloud Service Provider	CSP	The CSPs that are in business relation with the multiCloud application Service Provider and which provide the Cloud Services that the different multiCloud application components use.
DevOps Team	CSP	The group composed of the roles played by Application Developers, System Operators, System Administrators and Business Managers that collaborate in the development and management of the multiCloud application, following a DevOps approach [25]. As outlined above, the DevOps Team is actually the main executor of the workflow and the Cloud Service Provider of the services offered by the multiCloud application. Nevertheless, from now on in the text, unless explicitly outlined, the DevOps Team will not be referred as CSP, in order to avoid confusion with (external) CSPs.

Table 2 outlines the possible sub-roles that the DevOps Team can play along the workflow.

Table 2: Overview of DevOps Team sub-roles within the DevOps workflow

Process phase	Sub-role according to ISO/IEC 17789:2014	Description
Development	Application Developer	The DevOps Team will act as Application Developer when executing the responsibilities related to development of multiCloud applications. The development shall be understood herein as the set of all activities that span from application requirements specification to implementation, including architecting, detail design, coding, testing, etc.

Process phase	Sub-role according to ISO/IEC 17789:2014	Description
		Therefore, Application Architect (and Security Architect) role is also an Application Developer.
Development	Application Architect	The DevOps Team will act as Application Architect when executing the responsibilities related to the design of multiCloud applications or services. They pursue accomplishing the maximum benefits in a multiCloud application in terms of functional, security and business features by combining the different Cloud offerings.
Development	Security Architect	The DevOps Team will act as Security Architect when executing the responsibilities related to the design of the solutions which aim at ensuring the security in the design of multiCloud applications. As the ISO 17789:2014 was issued before GDPR and Privacy Architect is not included as a specific role therein, we can consider that the DevOps Team acting as Security Architect is also responsible for ensuring the required privacy-by-design and privacy-by-default principles hold in the multiCloud application design in cases when private information is processed by any component (outsourced or not) of the application.
Operation	System Operator	The DevOps Team will act as System Operator when executing the responsibilities related to the deployment of multiCloud applications.
Operation	Service Administrator	The DevOps Team will act as Service Administrator when executing the responsibilities related to runtime management of the multiCloud applications, which includes the monitoring of these applications. (Note that, even if this role is assumed by the multiCloud application SP, this is a typical customer role).
Development & Operation	Service Business Manager	The DevOps Team will act as Service Business Manager when executing the responsibilities related to the business aspects of offering Cloud services to Cloud service customers. They create and track the business plan, define the service offering strategy and manage the business relationship with Cloud service customers.
Operation	System Operator	The DevOps Team will act as System Operator when executing the responsibilities related to the deployment of multiCloud applications.
Operation	Service Administrator	The DevOps Team will act as Service Administrator when executing the responsibilities related to runtime management of the multiCloud applications, which includes the monitoring of these applications. (Note that, even if this role is assumed by the multiCloud application SP, this is a typical customer role).

3.2.4 Models

Different formalisms are used in the workflow proposed to support the security-by-design and privacy-by-design process of multiCloud applications.

Figure 6 represents the chain of system abstraction models that are created within the different methods of the overall DevOps methodology of Figure 4. The details of each of the models used

will be described in the corresponding section later, as part of the method description. However, in this figure we represent the overall flow of the modelling formalisms indicating in which process step they are created and used as well as how they are related to one another.

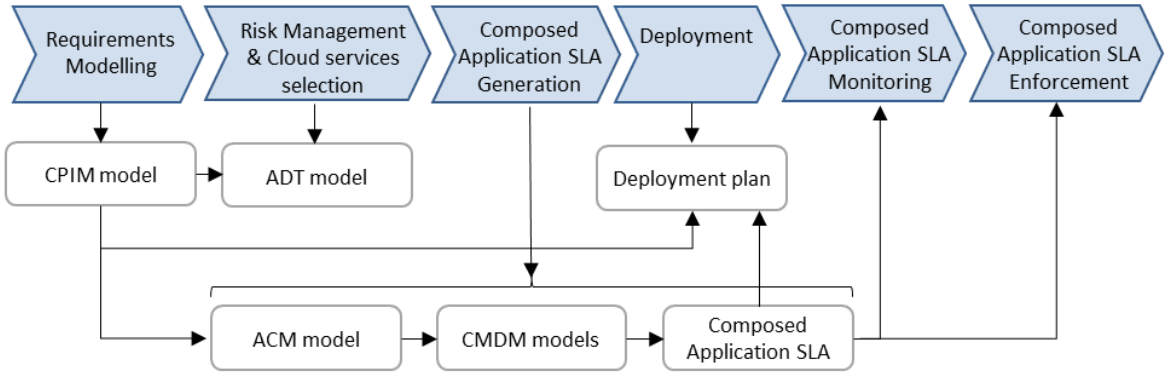


Figure 6: Models used in the DevOps workflow for security and privacy assurance in multiCloud applications

Each of the models supports the study of different aspects of the security and privacy of multiCloud systems and they relate to one another as follows:

- **Cloud Provider Independent Model (CPIM)**: The model is created in the first step of the process when analysing the system components and their architecture. The methodology herein advocates for starting the analysis of security and privacy requirements of the application components together with the deployment and communication requirements. Therefore, this model captures these requirements in form of needed and provided security capabilities in the components and states which enforcement agents are also used by the components. The language used to create the model is the new language proposed in this Thesis explained in Section 3.3.
- **Attack-Defence Tree (ADT) model**: This model is a tree structure graph which captures all the envisaged attacks and defences in the system in form of attack goals against the system and corresponding protections or defences to safeguard the system from them (see Section 3.4). The assets (or components) as well as the attacks and defences to include in the model are the result of the previous analysis of the CPIM.
- **Application Composition Model (ACM)**: This model is a directed acyclic model graph representing the components of the application as nodes representing the services offered by the components and the relationships between the nodes in the provision or consumption of the services and the associated SLAs. The model allows to reason on SLA provision and consumption needs which drive the SLA Composition. See Section 3.5 for further details.
- **Control Metric Delegation Models (CMDMs)**: On top of the ACM model multiple CMDMs are defined describing for each of the controls. The model aims at supporting the analysis of control implementation relationships between application components, because the implementation of (parts of) controls may be inherited from other components, which influences how the controls can be declared in the SLA of individual components and thus in the overall Application SLA. This model is also explained in Section 3.5.
- **Composed Application SLA (including SecSLA and PLA)**: This model is created as a result of the reasoning over the CMDMs created for the application following the composition rules of the SLA Composition methodology explained in Section 3.5. The

model will serve in Operations phase to perform the Continuous Monitoring and Enforcement of the security and privacy levels promised in the Application SLA.

- **Deployment Plan**: This is the Cloud Provider Specific Model (CPSM) which abstracts the system in a view closer to the actual implementation and execution of deployment itself. The inputs of the model are: i) the CPIM created before expressing the distributed deployment and communication requirements of the components together with the required enforcement agents to deploy, and ii) the Application SLA which indicates the actual Cloud service providers that will be used by the application components and in which the provision of the Cloud resources need to be done prior followed by the deployment execution itself.

3.2.5 Relation with MUSA DevOps methodology

The initial version of the DevOps methodology presented in this Thesis was born in the MUSA Horizon 2020 European Union funded research project (grant agreement No 644429) [47].

The version of the methodology designed and developed in this Thesis is an improved evolution of the previous version. While keeping the agile DevOps approach and six workflow steps of MUSA methodology for multiCloud application creation and operation, this version enhances the three Development phase activities as follows:

1. The security and privacy requirements modelling language of this Thesis was developed as part of the results of the MUSA project, and the version of the meta-model presented and validated in this Thesis enables not only security requirements specification but also privacy requirements definition. To this aim, the control family standard adopted in this version is the new version of the NIST SP 800-53 [119], Revision 5, which includes both types of controls. See Section 3.3 for further details of the new language.

2. The methodology for Risk Management and CSP Selection is completely new. The MUSA version relied on the use of OWASP and risk severity quadrants for evaluating system risks, while the version in this Thesis utilizes Attack Defence Trees (ADTs) to perform the necessary quantitative and probabilistic risk assessments and risk sensitivity analyses. ADTs enable to evaluate risks in different scenarios of combinations of attacks against system components and defences applied to protect them. Moreover, ADTs make it possible to consider the relationships between the risk attributes of the different attacks and defences when computing the system risks. Therefore, the new method developed enables to make fine-grained reasoning on not only system risks but also risks at particular component level and take informed decisions about which security strategies to apply in different cases.

   Similarly, the risk-based optimisation of system defences presented in this Thesis cannot be found in MUSA as this is a new method. The major advantage of including this method is that it enables to identify the optimum set of defences that minimises the system risk or asset risk (as desired) under different constraints, for example, at the minimum cost. Therefore, our optimisation of defences allows to know which security and privacy defences (controls) are required in the system components and which need to be requested to Cloud Service Providers so as the selection of CSPs is made according to the risk analysis results.

3. The method for multiCloud Application SLA composition is new. While keeping the roots in the MUSA method by Rak [129], the new composition method brings three major innovations. First, the new method serves to compute not only Security SLAs but also Privacy SLAs (PLAs) of multiCloud applications. Second, in the new method the existing

control delegations between system components are considered when evaluating the composition. These delegations represent cases when some components inherit the implementation and declaration of (parts of) the control from other components which is a critical aspect to include in the SLA composition. Finally, the new method not only allows to know which controls can be declared in the multiCloud Application SLA, but makes it also possible to compute the SLOs of the controls declared, which is fundamental to know at the monitoring step which are the target levels of application security and privacy behaviour that need to be maintained and which would rise alerts in case they are not achieved.

Please note that the three steps corresponding to Operation phase, i.e. Deployment, Monitoring of composed Application SLA and Enforcement of composed Application SLA, are inherited from MUSA methodology and were not included as part of this Thesis work.

3.3 Security and privacy requirements modelling language for multiCloud applications

In this section we describe the second contribution of the Thesis as per Figure 2, i.e. the Security and privacy requirements modelling language for multiCloud applications.

3.3.1 The security and privacy requirements modelling language: extended CAMEL

In the last years the number of market offerings of Cloud based infrastructure services and platform services has increased notably along with the number of providers. One of the main problems is that Cloud providers support different interfaces for different set of services. As a result, DevOps teams need to learn and automate the process behind the provisioning of application services when deploying a Cloud application. Among other tasks, they need to learn how to create VMs and how to choose the right VM sizes for each service. To overcome these limits, current research stakeholders propose to take advantage of the well-known Model Driven Engineering (MDE) techniques to configure the deployment of Cloud applications.

The MDE techniques become really interesting for multiCloud application specification when the model captures multi-concern information, expressed at high level first and detailed at low level application platform afterwards, and when the model is enacted at runtime. This allows for a seamless alignment of design decisions with actual deployment and application execution.

In this work stream, Cloud Application Modelling and Execution Language (CAMEL) language, which includes CloudML languages as Deployment model for expressing deployment needs, which were described in state of the art Section 2.3 excel from other languages and is the basis of the work of this Thesis.

In the DevOps methodology proposed, CAMEL is adopted to cope with modelling of security and privacy concerns of multiCloud applications. The rationale for the selection of CAMEL on top of other versions of CloudML and TOSCA is the following.

First, compared to CloudML variants, CAMEL includes CloudML as Deployment model, so adopting it implies adopting CloudML in this sense. In addition, CAMEL includes also other models such as Security and Requirements models that are valuable when placing the focus on security. CAMEL follows the same approach of CloudML in relation to the provision of a single set of abstractions so that developers can define declaratively: (i) the application architecture made of components, (ii) their use and host relationships so that they can be properly configured and

deployment orders automatically derived, (iii) constraints on the characteristics of the required types of VMs and (iv) the execution commands to provision application components.

Second, as the objective of this work is to facilitate the analysis of security and privacy concerns of Cloud and multiCloud systems, it is required to adopt a language that allows the users to friendly and easily create and deploy security-aware and privacy-aware components balancing security and privacy with performance properties. To this aim, CAMEL includes two main security-oriented meta-models [41]: the *Security* meta-model to support the specification of security requirements posed by users and capabilities of Cloud providers (in form of security controls and Service Level Objectives) and the *Organisation* meta-model that captures security-oriented information about organisations (parties in the multiCloud application life-cycle) including organisation security policies, users and roles.

The application requirements in CAMEL are mainly captured by the *Requirements* meta-model. Thus, both the *Security* and the *Requirements* meta-models can complementarily capture security requirements. CAMEL offers support to the following tasks [41]: (i) matching in deployment phase security capabilities and requirements of the application to the security controls offered by the Cloud providers; (ii) monitoring and assessing security SLOs which can be mapped to adaptation rules in order to adapt the structure or behaviour of an application to exhibit the security level required.

The CAMEL language already supports some degree of access control in the form of allowing the specification of the organisation policies that rule which organisation roles can have access to which services and which private information in the application. This access control could be combined with the access control support in our approach which is done through the enforcement agents in the components to achieve a fine-grained specification of access policies.

Third, compared to TOSCA, the CAMEL language expressiveness is higher as it is the only language that already provided baseline Security model and Requirements model (for deployment). TOSCA provides a language for specifying the application components comprising the topology of Cloud-based applications along with the processes for their orchestration. TOSCA supports the specification of types and templates, but not instances in deployment models.

Fourth, the CAMEL language enables the creation of high-abstraction models of the application that express architecture, security, privacy and deployment requirements and that can be easily further refined into automatically executable deployment plans (which are Cloud platform dependent models).

Last but not least, CAMEL language comes along rich editors such as Editor Eclipse plugin from PaaSage (textual) and other graphical editors (such as Modelio). The non-graphical CAMEL editor includes friendly functionalities such as identification of attributes required versus optional, auto completion capabilities and model validation among others. Thus, as these are exactly the features aimed to be maintained in our modelling language, the textual format was decided. As both graphical and textual models allow for the same degree of detail and completeness, the text-based modelling option was selected.

The meta-model capturing the security and privacy requirement concepts of the proposed modelling language can be found in Figure 7 and a report of its major contributions is provided in the following subsections.

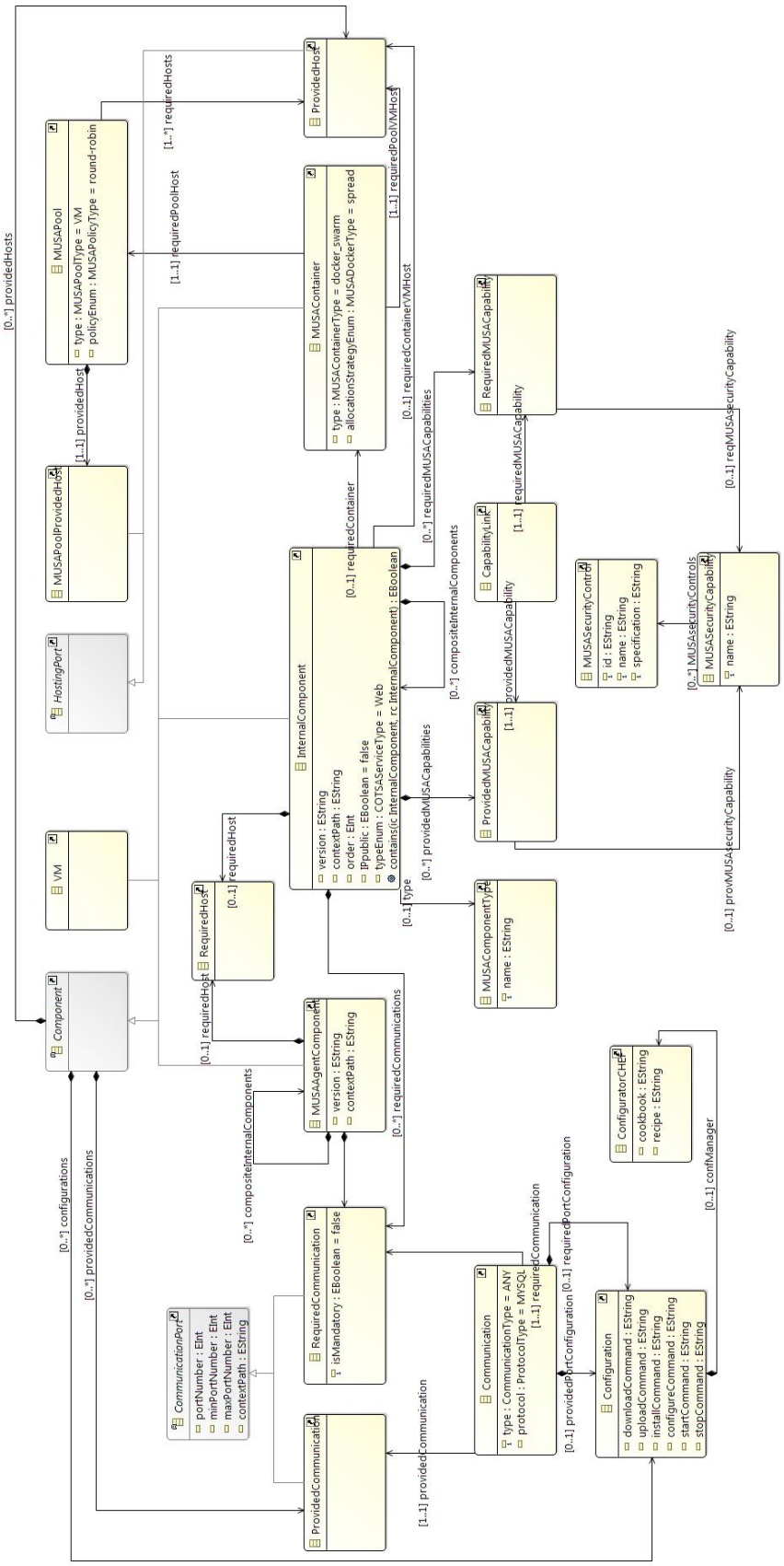


Figure 7: The proposed Security and Privacy Domain Specific Language for multiCloud applications

3.3.2 Contributions to security and privacy behaviour specification

The extensions to CAMEL language proposed include the enhanced component security and privacy behaviour characterization, which addresses concepts required to support both composition of components' security Service Level Agreements (SLAs) as well as security and privacy risk analysis. They are described in the following:

- ***Classification of components by their nature which allows describing what the component does***. Our CAMEL extension allows classifying components by specifying two features of them: WHAT and HOW. While WHAT indicates the type of the functionality delivered by the component, HOW indicates the way the component is delivering such functionality.

  Currently, three types of HOW have been defined:

  a. COTS, which refers to Commercial of-the-shelf software that the application uses.

  b. SERVICE, meaning that the component is not a commercial package but developed by the DevOps Team responsible for the multiCloud application engineering.

  c. AGENT, i.e. a security or privacy mechanisms in form of software component that needs to be deployed together with multiCloud application components and which is responsible for executing some security or privacy protection or reaction in one or multiple system components. The potential values for AGENT are those corresponding to security and privacy mechanisms (developed internally or outsourced from third parties) which are available to use by the multiCloud application.

  The types of WHAT include:

  a. in case the component is COTS or SERVICE, the possible WHAT values are: Web, Storage, IDM or Firewall. Web refers to any functionality provided through a Web interface, Storage refers to data storage solutions (e.g. MySQL), IDM stays for Identity Management and Firewall for any software solution that protects resources from unauthorised access.

  b. in case the component is AGENT, the possible values of WHAT represent the security or privacy functionality offered by the AGENT to the application components.

  The WHAT and HOW information is required at the SLA Generation step to create the Composite Security SLA of the multiCloud application from individual components' SLAs. The Risk Analysis activity in order to identify the security threats and risks at component level.

- ***Controls information that properly supports Security and Privacy Control Framework families***. To this aim, within Sec *name* attribute has been updated to <Family>-<Number>(Number) format. In addition, the *subdomain* attribute in the Security Control entity now is optional instead of compulsory.

  Through the Security Control entity, the CAMEL extension developed allows specifying which security capabilities are required and which ones are provided by each multiCloud application component. The security capabilities are defined in the model by selecting and grouping the security controls part of the capability. The security controls of a Security

Control Family are pre-defined, and the list is offered to the user to ease the selection of the ones to be included in the security capability.

Currently, the security and privacy controls from the NIST SP 800-53 Rev. 5 [119] are supported to express in form of Security Control entity sets the security and privacy capabilities. In the following example two security capabilities CAP1 and CAP2 are defined, the first with three security controls and the second with only two. Privacy capabilities would be defined in a similar way by selecting privacy controls in the control framework.

```
security model SEC {
      security capability CAP1 {
      controls [MUSASEC.AC-11(1), MUSASEC.AU-13(2),
      MUSASEC.AC-17(6)]
      }
      security capability CAP2 {
      controls [MUSASEC.AC-11(1), MUSASEC.AC-17(6)]
      }
}
```

Once the security capabilities are defined in the CAMEL (in the security model part of the model), the user can specify what security and/or privacy capabilities the components require and/or provide.

In the following example, Comp1Cap is a provided security capability and Comp1CapReq a requested one.

```
provided security capability Comp1Cap {
security capability SEC.CAP2
}
required security capability Comp1CapReq
```

When a component requires a specific security or privacy capability from another component (in the example, Comp1CapReq) then the matching of the capability needs to be modelled as follows.

```
capability match Comp1ToComp2 {
from Comp1.Comp1CapReq to Comp2.Comp2Cap
}
```

3.3.3 Contributions to multiCloud deployment specification

Other extensions to CAMEL language developed address improvements for enhancing the expressiveness of the deployment requirements, as follows:

- ***Explicit characterization of the nature of the IP address associated with virtual machines in which the components will be deployed***. At deployment phase, when acquiring new Cloud resources such as VMs, the system operator needs to indicate whether a public IP address is required. The CAMEL extension allows specifying whether the IP address required for a VM should be public or not by the IP public attribute on each component. The possible values for this attribute are: true or false.
- ***Specification of the deployment order of the application components***. Dealing with multiCloud environments, it is critical to identify the order in which each component should be deployed and configured, since there are inter-dependencies among the components that are part of the same application. For example, the start up of a component may require that another component is up and running in advanced. The CAMEL extension allows specifying the order in which the components are required to be deployed. This can be done by using the order attribute for each component. The expected value for the order attribute is an integer number.
- ***Explicit definition of data exchange protocols***. In our extended CAMEL, users can model the communications between the components (e.g. by setting the IP addresses and ports in the configuration of the components) and specify the need to use a specific data exchange protocol (e.g. MySQL, OAuth, Other).
- ***Modelling of dynamic configurations of communications between components***. In CAMEL, users model the communications between the components in a static way (i.e. through specific port numbers and operating system configuration variables). However, in the new language dynamic characteristics have been introduced such as context paths (instead of IP addresses) and dynamic port ranges.

  Such new capabilities are useful, for instance, to configure explicitly inbound traffic when users deploy components in Docker [152] containers.
- ***Modelling of deployment handlers***. In CAMEL, the user can model components and associate deployment instructions for installing, configuring, starting, and stopping the components on virtual machines. However, such deployment instructions are restricted to scripted commands and CAMEL lacks support to the specification of configuration management tools such as Cloudify [153], Puppet [154], or Chef [155]. Therefore, in the new meta-model this gap between multiCloud application models and these advanced frameworks has been faced via the new Configuration entity and its associated concepts (e.g. cookbooks and recipes in case of Chef).
- ***Modelling of PaaS layer elements***. CAMEL lacks support to the description of architectures where the application components are not directly deployed in Virtual Machines (VMs) but in containers. Our extension allows specifying the container type that will be used in deployment and defining the component allocation strategy it should follow, even in cases when the container uses VM pools. The new elements in our extension include:
  - *pool*: is a cluster of VMs, which will be used by a container.
  - *manager*: the VM in a pool that will act as the manager in contrast to the rest that will act as workers.

- *container*:
  - *type*: the container solution to use, for example, Docker swarm [156].
  - *allocationStrategy*: defines the allocation strategy of the containers on top of the acquired VMs for resource optimization (e.g. automatically scheduling container workloads). It supports the following four values:
    - *spread*: balance containers across the VMs in a pool based on the available CPU and RAM of the VMs.
    - *binpak*: schedule containers to fully use each VM capacity. Once the full VM capacity has been used, the container moves on to the next one in the pool.
    - *random*: choose a VM randomly.
    - *custom*: the user defines the specific VMs in which the containers should run.

- ***Refinement of security aspects in Organisation, User, Credentials and Role entities.*** A number of enhancements to CAMEL have been made in order to manage the authorisation of different roles in the DevOps Team to multiCloud application deployment execution. For instance, the types of credentials available in the meta-model to authenticate a user have been extended. Moreover, *expiration date* has been added to Credentials and the re-use of Role Assignments has been improved by allowing an easy assignment of roles to multiple groups and users.

3.3.4 Contributions to self-protection capability of multiCloud applications

Considering *self-healing* as the capability of a multiCloud application of being able to self-control or modify its security and privacy behaviour at runtime so as security incidents, privacy incidents or attacks are corrected or mitigated, self-healing is enabled by the Enforcement Agents that are deployed together with application components.

As their name suggests, the Enforcement Agents enforce multiCloud application security and privacy properties at runtime such as access control, security vulnerability scanning or Denial of Service (DoS) mitigation mechanisms. For these mechanisms to work, they need to be deployed at the same time as the application components are deployed. Some of these mechanisms require to be deployed together with the component that they will enforce the property on (e.g. in the same Virtual Machine).

Therefore, the proposed extended CAMEL language allows the definition of Enforcement Agents as Internal Components of the application, similarly to application components themselves, so as they can be included in the deployment plan. Such agents are already pre-defined in a file or repository representing the available security and privacy mechanisms that can be used for protecting the multiCloud application or for adding security or privacy functionality to it, e.g. access control, high availability, vulnerability scanning, anonymisation, etc. This way, the users are able to re-use and configure them in a friendly way. Some of these agents may be always running and some may be managed through the enforcement services in Operation phase.

3.4 Continuous Risk Management and risk-based optimisation of defences for multiCloud applications

In this section we describe the third contribution of the Thesis as per Figure 2, i.e. the Continuous Risk Management and risk-based optimisation of defences for multiCloud applications.

The proposed methodology to continuous quantitative risk assessment is based on capturing in form of Attack-Defence Trees (ADT) the envisaged attack-defence scenarios of the multiCloud application. The ADTs enable to reason on the relationships between potential attack events against different parts or components of the system. Therefore, the methodology leverages ADTs to evaluate the system risks when different attack situations are faced and different defence strategies are adopted. The evaluated risks guide the selection of the needed defences in system components including the controls to require to external Cloud and IoT service providers.

The approach follows the DevOps paradigm [25] and promotes early updates in risks assessed at development time by taking into account at all times the security status of the system in operation. The methodology integrates in the calculation of the risks the influences of the deployment of the different system components in the chosen external providers as well as the current status of the defences and the attacks in operation, so as continuous risk assessment is offered by refining the risk evaluation made at design time.

In this Thesis we put the focus on the risk assessment techniques and we have left out of scope of the work the details of the continuous monitoring at operation which can be consulted in [157].

3.4.1 Continuous Quantitative Risk Management Methodology for multiCloud DevOps

Continuous risk management involves the identification and initial evaluation of the risks over system assets followed by the continuous monitoring of the evolution of the risk severity level. Therefore, continuous risk management in DevOps process should rely on continuous assessment at Operations of the status of the risk attributes identified during Development (design) so as the risk level can be tuned according to the actual occurrence of attacks or their symptoms, as well as the status of deployed defences.

In order to assess system risks, we propose to adopt a systematic approach which exploits the use of Attack Defence Trees (ADT) to explore the quantitative relationships between threats, controls and risks. After the initial estimation of the system risks thanks to the ADT model, the track of the risk status when system is in operation would also be performed using the ADT.

Figure 8 illustrates the iterative process proposed to systematically perform the continuous risk management in multiCloud DevOps detailed in the rest of the section. Taking into account risk management is a multi-stakeholder business, the main actor in all the steps of the process is the multi-disciplinary DevOps team which includes software developers (system architects, security and privacy experts, programmers, etc.) and operators of the system (deployment experts, security operators, etc.) as well as business decision-makers.

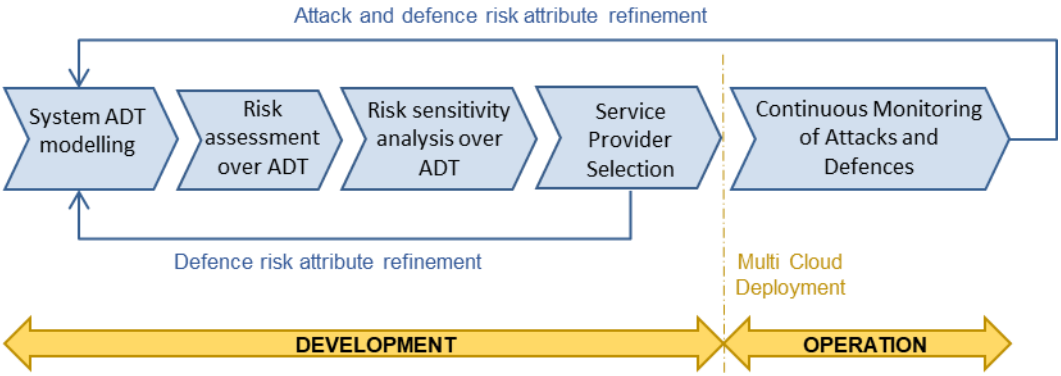


Figure 8: Continuous Quantitative Risk Management in multiCloud DevOps.

The overall process consists of five main steps, namely:

1. **System ADT modelling** where system developers create the ADTs representing the potential attack-defence scenarios and merge them into a unified system ADT.
2. **Risk assessment over system ADT** that consists in the evaluation of system risks by setting the risk attribute values to the tree leaf-nodes representing the possible attacks and required defences and propagating the values up to the tree top node.
3. **Defence optimisation over system ADT** where different combinations of defences are studied in the search of the optimal set which protects from desired attacks while minimising some parameters (such as risks, security investment, or attack impact) or maximising others (such as Return on Investment).
4. **Risk-driven service provider selection for third-party components** on the basis of the needed defences identified for the system and for the components in the previous step.
5. **Continuous monitoring** of the status of attacks and defences in operation which enables early feedback to risk assessment.

As it can be seen in Figure 8, two major risk refinement loops are considered in the methodology. The first risk assessment rectification occurs after service providers are selected for third-party components and the offered defences are known. This refinement is made once when system components' deployment options are decided and deployment results are known. Subsequent iterative refinements occur when system components are up and running in operation. The continuous monitoring of the performance of security defences in the components will offer information on the status of potential attacks on the components and their countermeasures, which allows for refining to more realistic values the risks parameters initially estimated in system risks such as the probability of a defence to fail, the protection effectiveness to reduce attack impact, etc. In the following sections the details of each of the methodology steps are provided.

3.4.1.1 System ADT modelling: Analysis of threats and defences

The modelling of system Attack Defence Trees consists in creating the ADT models capturing the potential attack scenarios against the system as well as the respective defensive controls that may be adopted to counter the attacks. In this section we explain the proposed approach to build the ADTs that represent diverse attack scenarios to the system and how they can be integrated into a single system ADT that enables the evaluation of overall system risks. In the methodology the idiosyncrasies of multiCloud composite applications are addressed by deriving from the system ADT the set of attack events and controls that correspond to each of the system assets, so as later the risk analysis on particular assets or components is possible.

3.4.1.1.1 Modelling of ADTs

Following a hierarchical attack modelling approach, for each attack-defence scenario envisaged an ADT shall be created where the high-level potential threat is represented by the root node which is decomposed in lower-level threats represented by intermediate nodes. The tree leaves are the attacker actions which exploit particular vulnerabilities of the system assets and therefore are not further decomposable. In general, attack actions against the assets (application components) depend on component nature, type, interfaces, etc. Defences or protections that system developers may adopt to counteract the external attack actions are represented associated to the attack events in the lower level. Figure 9 represents the ADT structure where attacks are represented by ellipses in red and countermeasures (defences) by rectangles in green. Attack goal refinement relations are depicted by solid edges between nodes, while defences are connected to the countered attacks by dotted edges. Two types of refinements from parent to children nodes (all of the same type) can be made: i) conjunctive refinement depicted by an arc which connects the parent's edges to its children and ii) disjunctive refinement with no mark in the graph.

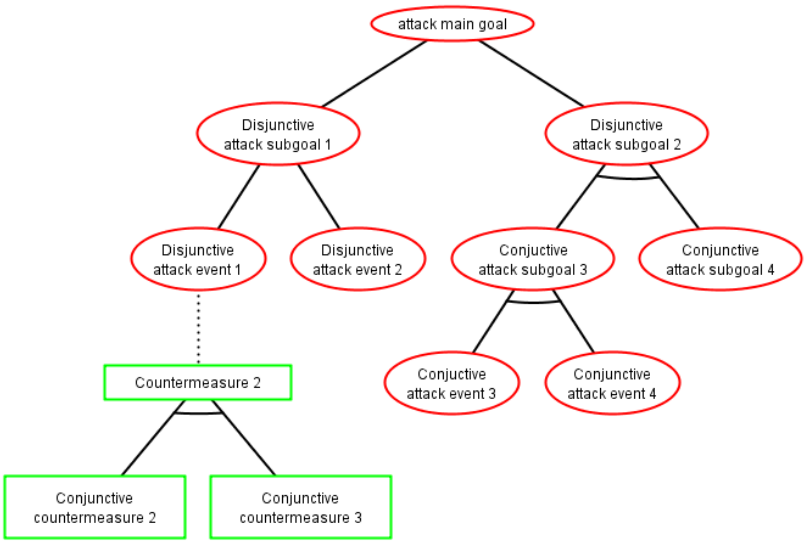


Figure 9: General structure of an ADT.

As it can be seen in Figure 9, the tree structure of ADTs enables to learn on the AND/OR relationships between the attacks and at the same time between the controls tackling different weaknesses of the system. They facilitate to reason on whether the intermediate attack sub-goals collectively (conjunctive relationship joined by AND operator in the parent) contribute to their parent goal achievement or alternatively (disjunctive sub-goals joined by OR operator in the parent). Similarly, ADTs illustrate whether the defences contribute jointly to parent countermeasure mechanism (joined by AND gate in the parent) or are alternative solutions (joined by OR gate in the parent). This will allow for quantitative expression of both: i) attack events' contribution to system risk severity level, and ii) defensive controls contribution to threat mitigation and therefore, to risk severity level reduction.

In ADTs it is necessary to take into account the defences that the system components offer. To this aim, components need to be self-assessed first in order to know which protections they already implement, so as to discard a number of potential damages. In these cases, the potential attack should be represented as a tree node together with the defence implemented in the asset.

Adding defences to individual attack events is a non-trivial task that requires expertise in security and privacy mechanisms for the system architecture components under study. Security self-

assessment techniques that can help in identifying appropriate defences to model in ADTs are offered by OWASP [158], Berkley [159] and CSA [144].

In multiCloud scenarios where system components may be outsourced to Cloud Service Providers (CSP), knowledge on Cloud security issues and measures adoptable by CSPs is needed. The MUSA Security Metric Catalogue [160] is a comprehensive collection of threats and security controls that can aid in this task. The threats in the catalogue are mapped to both controls that could be used to counteract the threats and metrics over the controls which would help in evaluating the control performance. The controls in the catalogue follow the standard taxonomy of NIST SP 800-53 Rev. 4, the predecessor of NIST SP 800-53 Rev. 5 [119], which is focused on security controls. The catalogue has been updated in this Thesis to use the taxonomy of NIST SP 800-53 Rev. 5 [119] which facilitates the auditability of defences in the ADTs as these controls are mapped to other standards such as CCM [123], ISO/IEC 27001 [120] and ISO/IEC 15408 [149].

Envisaged attacks to outsourced components need also to be modelled in the ADT together with the defences that would be requested to the external providers. This issue is extremely important in multiCloud systems, where multiple outsourced components may exist. The selection of providers of external services which the system will use (e.g. a Cloud IaaS service, a SaaS service, an IoT edge service, etc.) will be driven by the identified needed defences and the affordable security expenses in third-party components. The risk-driven selection of service providers step of the process explains this point later in Section 3.4.2.5.

In order to ease the comparison, reuse and audit of countermeasures modelled in the ADTs, the taxonomy used should be related to standard control frameworks presented in Section 2.6.9. The control classification and guidelines in these standard frameworks help in devising the required security and privacy defences protecting from the elementary threats of the tree. Moreover, the standard taxonomy of the controls can be adopted to formally name the defences which easies the selection of providers for external components according to standard controls specified in the providers' security Service Level Agreements (SLAs). In our approach we recommend the use of the NIST SP 800-53 Rev. 5 [119] as it is the one offering fine-grained controls, including not only security but also privacy controls.

3.4.1.1.2 Generation of the system ADT: Resolution of conflicts and overlaps between individual ADTs

In composed applications where the overall system is constituted by multiple collaborating parts or components, once all the different potential harm goals of external attackers are captured in ADTs, the relationships between created ADTs need to be analysed in order to understand how the main attack goals of each tree (root nodes) contribute to the highest-level goal of aggressors to "attack the system". The main objective is to understand how all the attack events to different parts of the system identified in the individual ADTs are interleaved so as a unified system ADT can be created. Similarly, relationships between defences applied to system parts or components shall also be analysed. This way conflicts and overlaps between individual ADTs can be solved when building the merged system ADT.

As shown in Figure 10, attack scenarios modelled by individual ADTs could have two types of relationship between them with respect to the attacker's ultimate goal of "attacking the overall system": i) *disjunctive* attack scenarios, thus representing alternative ways to harm the system, or ii) *conjunctive* attack scenarios where one scenario is totally or partially already captured by the other. While disjunctive attack scenarios present no conflicts or overlaps, conjunctive attack scenarios need to be studied carefully and decide on the best strategy to integrate them. In cases where one of the ADTs is totally represented in another larger ADT the integration is

straightforward. And when only a part of the ADT is represented in another ADT, the tree structure part that is not common should be transformed into a separated ADT which its own main goal as root. This new ADT will be treated as one of the disjunctive ADTs integrating the set of ADTs of the system.

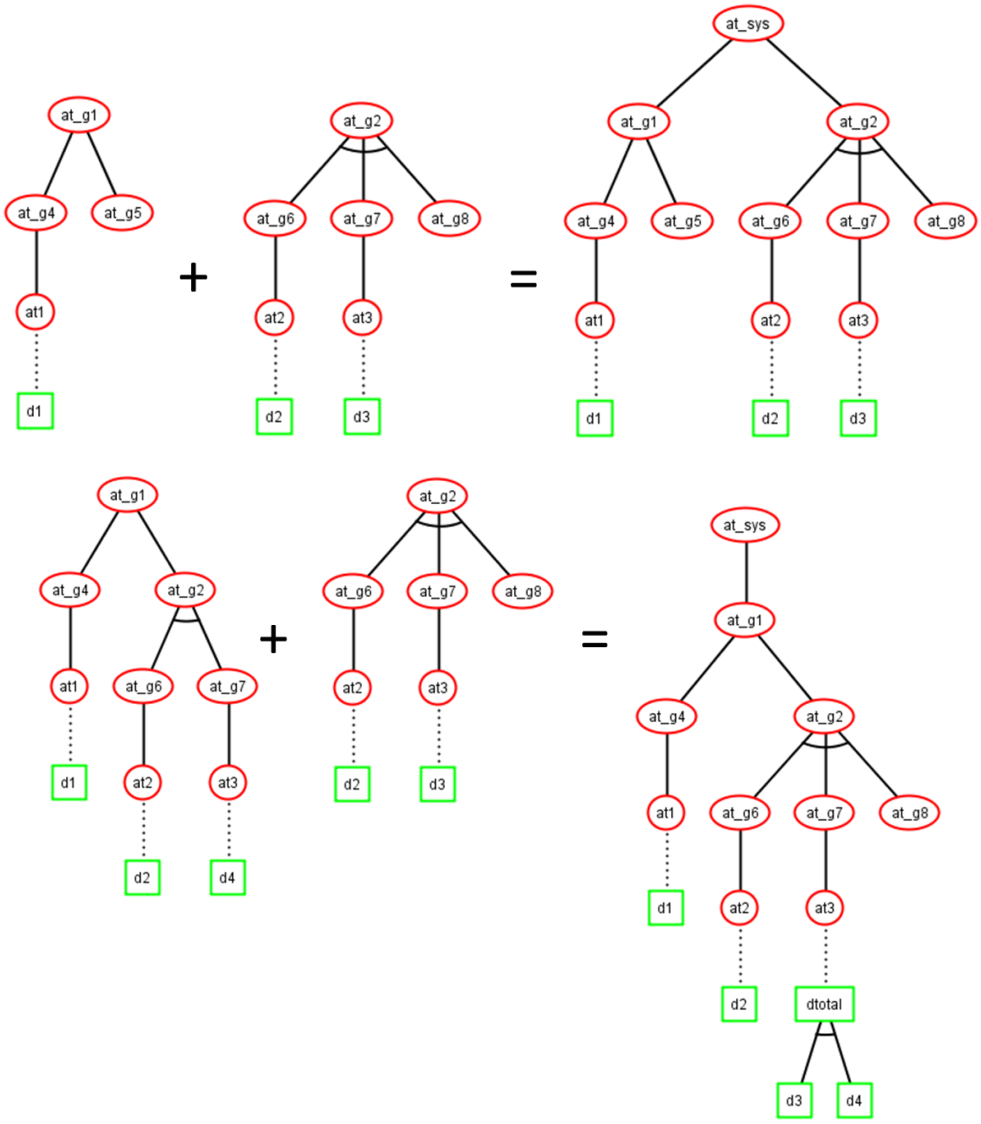


Figure 10: Example of individual ADTs unification for a) disjunctive ADTs (above) and b) conjunctive ADTs (below).

Potential discrepancies between modelled defences may also occur when for the same attack event two different ADTs propose a different countermeasure (or sets of countermeasures). Considering the attack event modelled in both ADTs exploits the same vulnerability of the same asset, as different defence mechanisms may have been identified, both of them should be included in the merged ADT as contributing together to counteract the attack event. This case has been exemplified in Figure 3 b) with defences *d3* and *d4* in both versions of ADT *at_g2* which in the merged ADT *at_sys* are jointly protecting from attack event *at3*.

From this point on, in our methodology we propose that, as the outcome of the analysis of the relationships between the ADTs, a common ADT for the complete system, namely the *system ADT*, is created which represents in a single tree node structure all individual attacks and defences to be

deployed in system components. As a result, the overall system risk level will be computed on top of the system ADT created and all paths and relationships between nodes will be taken into account when calculating risk metrics according to different node configurations, as explained later.

For large composite systems or systems where many individual ADTs have been modelled, building a unified system ADT may lead to a too large tree structure which visualisation is no longer easy, and therefore, it is advisable that the set of individual ADTs is maintained together with a simplified system ADT where its root node would have the root nodes of all individual disjunctive ADTs as children nodes aggregated by an OR relationship between them. Please note that, as explained before, the cases of conjunctive attack scenarios need to have been solved previously. This way, in order to calculate the risk of "attack the system" main goal first the risk values over individual ADTs could be computed in isolation and then moved to the system ADT children nodes for the final calculation.

3.4.1.2 Risk Assessment over ADTs

3.4.1.2.1 Qualitative Risk Analysis over ADTs

Once the system ADT is modelled on the basis of a set of ADTs that describe potential attack scenarios against the system, it is possible to perform both qualitative and quantitative analysis on the system ADT as explained below.

Qualitative analysis of ADT explores the satisfiability of the ADT and the relative importance of individual attack events and defences on the basis of the logical structure of the ADT.

Provided that $X = (X_1, X_2, X_3, \dots, X_n)$ is a state vector for the ADT where X_i is the boolean variable associated with event E_i (attack or defence event) represented by node n_i, the main goal (root node) of the ADT can be expressed [85] [90] as a Boolean structure function of the leaf nodes $\overline{F(X)}$ where X is a state vector of the ADT which elements X_i are boolean variables such that $X_i = 1$ when the event E_i represented by node n_i occurs, else $X_i = 0$. Note that $\overline{F(X)}$ is independent of the attributes used to decorate the ADT nodes.

The ***importance measures*** such as the structural importance [161] or Birnbaum importance [162] are well-known measures that enable to identify the most critical attacks and defences in the ADT structure. For multi-component systems the significance of the relative importance measure is stressed as it allows to determine which attacks have greater structural weight and thus which target components are the weakest links and deserve more attention for protection.

The structural importance measure of an event E_i represented by node n_i in an ADT with n leaf nodes can be expressed as the normalized count of state vectors where the component is relevant for the boolean structure function:

$$I_{E_i}^{ST} = \frac{\sum_{i=1}^{n}(\overline{\varphi(X)}_{E_i} - \overline{\varphi(X')}_{E_i})}{2^n}$$

Where $\overline{\varphi(X)}_{E_i}$ represents the root node boolean structural function when the E_i event occurs and $\overline{\varphi(X')}_{E_i}$ in the absence of the E_i event.

In our methodology we will use structural importance measures to identify the more relevant events, attack and defences, according to the tree logic.

Mincuts* or *minimum cuts of an ADT are the different attack-defence suites (or scenarios) that realise the main goal. An ADT with n leaf nodes can at the most have $\binom{n}{2} = \frac{n\,(n-1)}{2}$ mincuts. It is

important to note that a defence present in an ADT mincut covers every attack event in the mincut [85].

In our methodology we will use mincuts to reason on the best defence strategies to apply over both the individual system components and the system as a whole. By additionally studying in the system ADT which system assets are the targets of the elementary attacks, it is possible to know the set of attack events, and respective countermeasure set (if any), that correspond to particular system assets of interest.

Consequently, a three-dimensional ***relationship matrix*** (T) can be derived from the ADT mincuts where the rows represent attack events (AT_i), the columns represent the defences (D_j), and the layers or pages are mapped to system assets (A_k). Therefore, from the system ADT we can obtain a matrix $T = f\left(AT_i, D_j, A_k\right)$ where, if for asset A_k, AT_i is an attack against A_k and D_j covers AT_i, then the element t_{ijk} in T = 1, else t_{ijk} = 0.

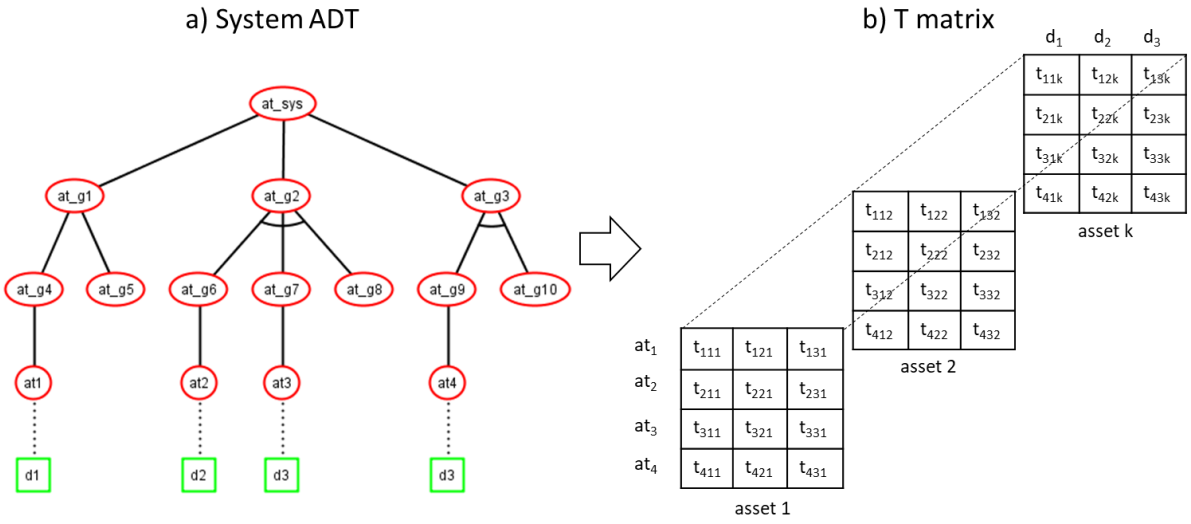


Figure 11: a) System ADT with 4 attack events and 3 defences on k assets, b) 3-D relationship matrix T for system ADT in a).

An example of how to obtain the system T matrix from a system ADT as depicted in Figure 11. In this example, four attacks (at1, at2, at3 and at4) are captured in the ADT which are treated by three defences: d1 protects from at1, d2 protects from at2, and d3 safeguards from at3 and at4 simultaneously. By identifying which system assets' vulnerabilities are exploited by each of the four attacks, it is possible to relate the attacks and defences to system assets and hence, build the T matrix by setting to value 1 the elements in the cells that correspond defences that safeguard the asset from an attack as per the main goal mincuts (i.e. attack-defence suites that realise the main goal) and the rest to value 0.

The relationship T matrix will be used to evaluate system and asset level risks as well as to reason on potential defence strategies for the system and for individual assets as explained in the following sections.

3.4.1.2.2 Quantitative Risk Assessment over ADTs

The quantitative analysis starts by decorating the attack events and the defences in the system ADT with estimated values for the proposed risk attributes as explained in the rest of the section. Once the attribute values of the leaf nodes are defined, different measures on the ADT can be obtained.

For example, the quantitative analysis enables to assess the risks of different attack-defence scenarios by propagating up the system ADT root node the risk attribute values of the nodes in the tree starting from the leaf nodes.

The following subsections provide the details of the quantitative risk assessment methodology proposed, starting from the proposed attack and defence risk attributes.

3.4.1.2.3 Attack risk attributes

Multiple information security risk evaluation methods exist in the literature [164] most of which are based on calculating the risk level by multiplying the *probability* of the attack to be successfully realised by the *impact* the attack would have over the asset (i.e. the damage or penalty to the asset), as show in Equation (1). This assessment has a very extended variant shown in Equation (2) which considers the *cost* of the perpetration of the attack to explicitly capture the idea of the higher the effort level for the attacker (i.e. the amount of resources required by the attacker), the lower the risk of the attack. In Equation (1) the *cost* is usually interpreted within the *probability* element, as more costly attacks are less probable to happen and therefore to be successful.

Other risk formulae [22] evaluate the risk according to the importance or *value* of the asset following a formulation like in Equation (3). The asset value is often associated to the *cost* or *business value* of the asset, so the greater the value the more or best protections should the asset deserve. The asset value could be captured within the *impact* concept considered in the expression of Equation (2), meaning that attacks on the most valuable assets are the ones with highest negative impact and most harmful to the business.

$$R_i = P_i \times I_i \tag{1}$$

$$R_i = \frac{P_i \times I_i}{C_i} \tag{2}$$

$$R_i = P_i \times I_i \times V_i \tag{3}$$

Where i represents each threat or potential attack in a set of T threats against the asset, i.e. $i \in [0, T]$.

In our methodology we propose to use the three attributes-based risk evaluation in Equation (2) as it facilitates the analysis of the cost-effectiveness of the defence strategies to adopt in order to minimise risks in both individual assets and in the system as a whole. With the values of these three attributes together with the resulting risk value a risk attribute vector $\{P_i, I_i, C_i, R_i\}$ is built where P_i is the *probability* of success of attack, I_i *impact* of the attack on the asset, C_i the *cost* of the attack, and R_i the *risk severity* evaluated by using Equation (2).

It is important to note the units and potential value ranges of the operands in the formulas so as the resulting risk level is meaningful. In the three Equations (1), (2) and (3) the successful occurrence probability values fell in the [0,1] interval, while the impact values are usually between 0 and 10, with 0 expressing no impact and 10 the maximum impact over the system. Threats with 0 likelihood or 0 impact are not worthy to consider for risks, so the lowest limits are usually not used in the standard guidelines (e.g. ISO/IEC 27005 [22], OWASP Risk Rating Methodology [98]).

It is worth to mention that the units and values for *costs* in Equation (2) can be given in dollars, euros, man-hours or generic cost units that later need to be converted into money units using a previously selected conversion factor. When considering attack costs can range from 0 money units to infinity, the risks values calculated by making use of Equation (2) may lead easily to apparently negligible risk values in the order of thousandths of severity or lower. For this reason, we propose

to adopt a normalised scale for attack costs and defence costs where the costs are normalised to the node with the minimum cost in the tree as follows.

$$Cost_{normalised} = Cost/Cost_{unit} \quad (4)$$

Where $Cost_{unit}$ represents the cost unit used for the tree which is equivalent to a tenth of the highest cost node in the tree, i.e. $Cost_{unit} = Cost_{max}/10$, so the normalised value of this node is 10.

In order to avoid issues in computation of Equation (2), it is advisable that nodes with zero costs are decorated with an approximation of 10^{-n} cost units, where n is a natural value. This way, the cost scale will be within $[10^{-n}$,10] interval which renders to risks of $[10^{-n-1}$, 10] interval where low n values makes it easier for the analysts to compare risks between tree nodes.

Please note that logarithmic scales can also be used for costs to avoid this issue, where the cost values of the nodes should be normalised to the node with the minimum cost ($Cost_{min}$), which would have a normalised value of 1, as follows:

$$Cost_{normalised} = Cost/Cost_{unit} \quad (5)$$

3.4.1.2.4 Defence risk mitigation attributes

Defence or countermeasures can also present risk attributes related to their cost-efficiency in mitigating threat risks. In this line, we propose to decorate the defences in the system ADT with three risk attributes similar to those proposed for threats: i) the *probability* of the defence to successfully counteract the attack event, which ranges in the interval [0,1], ii) the *impact* the defence can protect as a percentage of the impact of the attack that can be avoided when adopting the defence, which interval is [0,10], and iii) the normalised *cost* of the application of the countermeasure mechanism for the defender which ranges within [0,10] interval as well. Equation (2) will be used for defence risk mitigation cost-efficiency level evaluation and the risk vector for the defences would include the four attributes $\{P_i, C_i, I_i, R_i\}$.

When adding a defence to a leaf attack in the ADT the risk posed to the system by the countered attack gets modified by the risk mitigation effectiveness of the defences and vice versa, when an attack event is modelled targeting a defence in the ADT its protection effectiveness is weakened. Therefore, defences in ADT act as countermeasures to attacks and conversely attacks act as countermeasures of defences. Therefore, if we generalise this situation for both proponent (attacker) and opponent (defender) perspectives of the ADT, a method for calculating the risk attributes in countered nodes is necessary. Table 3 below presents the rules proposed to evaluate the nodes countered by nodes of the opposite type.

Table 3: Risk vector rules for countered nodes in ADT.

Risk Attribute	Proponent	Opponent	Countered proponent	Countered opponent
Probability	P_p	P_o	$P = P_p \times (1 - P_o)$	$P = P_o \times (1 - P_p)$
Impact	I_p	I_o	$I = I_p \times I_o/10$	$I = I_p \times I_o/10$
Cost	C_p	C_o	$C = C_p$	$C = C_o$
Risk	$R_p = P_p \times I_p/C_p$	$R_o = P_o \times I_o/C_o$	$R = P \times I/C$	$R = P \times I/C$

As expressed in Table 3, the success of a countered node gets reduced when its safeguarding opposite node succeeds. Therefore, the success probability of a countered node can be computed as the success probability of the node multiplied by the probability of the countermeasure failing (i.e., 1 – success probability of the countermeasure).

The impact on the system of the countered node gets lowered by the impact of the countering node because they have opposite directions in the protection of the system. That is, the damage caused on the system by an attack is reduced if a corresponding defence is implemented in the system. Similarly, when an attack is designed against a defence, it reduces the defence system protection effectiveness.

Finally, the cost of a countered node is not affected by the cost of the countermeasure because in the worst-case assumption the proponent has no information about the costs of the actions by the opponent. In fact, while the attacker costs relate to the means they use for attacking the system (e.g. exploits software, botnet node infrastructure, etc.), the costs for the defender include all resources employed in the system protections (antiviruses, purchased security services in Cloud, security hardware, detection system infrastructure, developer costs, etc.). A priori these costs are independent as the attacker will launch the attack campaign with all the available resources ignorant of whether the defender has invested in implementing defences against them.

3.4.1.2.5 Estimation of risk attributes on leaf-nodes

One of the most important steps in the methodology is the initial estimation of the risk attributes values for both attack events (leaf nodes in the attack tree) and for the adoptable defences to mitigate their impact. The attribute value assignment is usually performed by security experts based on their previous experience and according to the assets' and system characteristics. Knowledge on ethical hacking as well as attack detection and analysis by system operators and experts will definitively help in risk values assignment. Countermeasure attribute value estimation will require the collaboration of both designers and operators as defensive or reactive protections could be adopted in the system with diverse costs and effectiveness. Different techniques for ADT decoration were studied by Bagnato et al. [165] and M.H. de Bijl [166] which may serve as reference.

The *probability* of an attack i $\in [0, T]$ to be successful can be evaluated as the aggregation of several factors like in the well-known OWASP Risk Rating Methodology [98], and thus the function for the threat probability computation would follow Equation (6).

$$P_i = \frac{\sum_1^{F_P} wPfi}{F_P} \tag{6}$$

Where F_P is the total number of quantitative factors to be considered in the threat probability calculation and $wPfi$ is the estimated weight for each of the probability factors with $wPfi \in [0, 1]$.

The *impact* value is also generally estimated by aggregating several impact factors. In this case, the total impact of a threat i $\in [0, T]$ over an asset is calculated by the Equation (7).

$$I_i = \frac{\sum_1^{F_I} wIfi}{F_I} \tag{7}$$

Where F_I is the total number of quantitative factors to be considered in the impact calculation and $wIfi$ is the estimated weight for each of the impact factors with $wIfi \in [0, 10]$.

For example, the OWASP Risk Rating Methodology [98] includes $F_P = 8$ likelihood or probability factors grouped in two main categories: four factors related to the vulnerability exploitability and four related to the threat agent capacity for exploiting the vulnerability. Note that for OWASP

$wPfi \in [0, 10]$ and the risk calculation follows Equation (2), and therefore, risk values $R_i \in [0, 100]$.

The *impact* evaluation in OWASP Risk Rating Methodology considers both technical and business impacts, each with four subfactors which are evaluated by means of discrete values. The total *impact* I_i calculated would therefore need to include $F_I = 8$ factors.

The NIST's CVSS scoring system [99] also supports vulnerability severity estimation as a combination of a set of exploitability and impact metrics. Other standards such as ISO/IEC 27005 [22] and ISO/IEC 29134 [167] provide also guidelines on how to estimate likelihood and impact of security and privacy threats, respectively.

In multiCloud systems where one or multiple components or services are consumed from third-party service providers, modelled attack scenarios may include attack events to those services as well as potential countermeasures that could be adopted in the outsourced services to avoid or mitigate such attacks. In these cases, it is necessary to perform an initial estimation of the risk attributes of those attack events and defences too.

The continuous monitoring at operation of the status of the system services, their defences and attack symptoms will allow to refine the risk attributes' vectors $\{P_i, C_i, I_i, R_i\}$ in the attack event nodes and in their respective defences.

3.4.1.2.6 Risk propagation algorithm

The definition of the values for the risk attributes in the leaf nodes of the tree is followed by processing the bottom-up propagation algorithm proposed on the risk attributes. This leads to know the risk level of each node in the tree and thus the risk level of the root node "attack the system", which depends upon the risks levels of all the nodes in the ADT.

In order to quantitative evaluate the risk of the system to be attacked, it is necessary to calculate the risk attributes' vector $\{P_i, C_i, I_i, R_i\}$ of the root node in the system ADT. To this aim a bottom-up algorithm which propagates the risk vector up the logic tree hierarchy is computed. There are several approaches in the literature for attack trees and ADT attributes propagation rules to compute the utility value of tree root node, such as those proposed by Weiss [168], Buldas et al. [81] and Edge et al. [84].

In our approach we adopt the principles of Weiss [168] by assuming the worst-case scenario where a *smart adversary* would intelligently apply all available resources to attack the system. This assumption influences the risk bottom-up propagation rules in the ADT as long as the behaviours of the AND operand and the OR operand differ in the evaluation of risk attributes from their children, and OR operand requires a local optimisation with respect to risk as follows.

- The risk associated with an AND node is calculated in terms of sum of efforts of the children. That is, while the satisfiability (probability of success) of the parent requires that all the children are satisfied, the cost for the parent is the sum of the costs of the children nodes and the parent impact also aggregates the children impacts. In the impact case, the formula proposed is that of Edge et al. [84] which accommodates the fact that in most cases the effect over a system of a set of successful actions is greater than the sum of the individual events. The risk of a parent OR node is the maximum of the risks associated with its descendants as the smart adversary will choose to carry out the attack that has higher probability of success and produces the highest damage with respect to the expenditures of performing the attack.
- In summary, the proposed *smart adversary case* risk assessment over ADT derives the risk attribute vector of the root node in the ADT as the result of the bottom-up propagation of the risk vectors of child nodes to parent nodes by the rules defined in Table 4. As in the Weiss'

proposal [168], the specificity of the risk propagation algorithm proposed resides in the need of computing first the individual attributes (probability, impact, cost) and the derived attribute (risk) for all the children in order to obtain the values for the parent node. An advantage of our method with respect to the Weiss method is that instead of relying on the empirical assessment of impacts we adopt the Edge et al. [84] formulation for this attribute, which enables risk estimations prior to system deployment.

Table 4: Risk vector propagation rules in ADT for Equation (2).

Risk Attribute	AND	OR
Probability	$P = \prod_{i=1}^{N} P_i$	$P = P_{maxRi}$
Impact	$I = \frac{10^N - \prod_1^N (10 - I_i)}{10^{N-1}}$	$I = I_{maxRi}$
Cost	$C = \sum_{i=1}^{N} C_i$	$C = C_{maxRi}$
Risk	$R = P \times I \,/ C$	$R = P \times I \,/ C$

[a] *N: number of children in the gate, max_{Ri}: the child with maximum risk computed with Equation (2).*

In OR cases where two or more children have the same risk Ri, the node with max_{Ri} among them will be selected as the one with highest probability value, or the one with the highest impact value if both have the same probability. In case all the children have the same risk vectors, the OR parent will adopt the risk vector of the first child.

As it can be seen in Table 4, our approach leads to commutativity and associativity of ADTs, while the ADT distributivity is not guaranteed. Therefore, in our method the calculated risk vector for root node in attack defence tree T_1 = A v (B&C) is not necessarily equal to the risk vector evaluated for the equivalent binary formula T_2 = (A v B)&(A v C), where A, B, C, T_1, T_2 are all attack defence trees and T_1 and T_2 are semantically equivalent. This is because in our method the worst-case assumption implies that in disjunctive options (OR operands) which are not equally equivalent in risk weight, the decision will be made on the one with highest risk weight. Hence, in cases where A has intermediate risk value between B and C, the result of T_1 = B&C will not be equal to T_2 = B&A for ordered risk values B>A>C or T_2 = A&C for C>A>B.

As a result, as well described by Jürgenson and Willemson in [90], similarly to other risk propagation methods that propose local maximums such as e.g. Weiss [168], Buldas et al. [81] and Edge et al. [84], our risk vector propagation method keeps only partial consistency with the semantics framework by Mauw and Oostdijk [80], who established the foundations for attack tree semantics and advocated for the commutativity, associativity and distributivity of the conjunctive combinator (AND gate) and disjunctive combinator (OR gate) of attack trees so as attack trees could be transformed to logically equivalent attack trees. However, our method reflects a more realistic paradigm where not all the candidate options do not represent the same risk to the system and therefore a smart adversary would not have the same appetence for all of them.

It is interesting to note that our methodology addresses both perspectives of ADT, proponent (attacker) perspective and opponent (defender) perspective, and computation rules for risk attributes are the same for both, which makes our method to adhere to the attack defence tree foundations by Kordy et al. [72].

3.4.1.2.7 Risk severity metrics proposed

Once the risk vectors are estimated for the attack events in the ADT, the risk landscape for the system could be obtained by depicting all the attack event risks in a two (Impact, Probability) or three (Impact, Probability, Cost) dimensional space. For simplicity and similarity with guidelines by standard risk frameworks like ISO/IEC 27005 [22] and OWASP Risk Rating Methodology [98], the two-dimensional space depicted in Figure 12 is usually preferred.

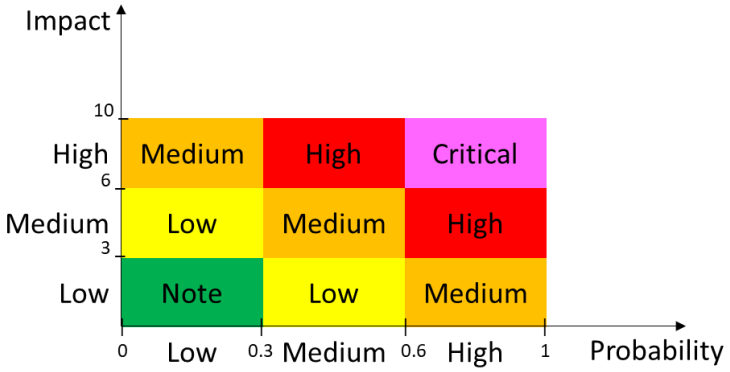


Figure 12: Risk severity quadrants

However, note that *cost* values could be considered within the *probability* attribute. Then, the attacks could be categorised into buckets of *high*, *medium* and *low* risk and ranked by their severity and position in the quadrants as in OWASP Risk Rating Methodology. The defence strategy would start fixing vulnerabilities associated to attacks with highest scores.

When risk quadrants are used, relevant metrics related to the density and risk severity of the threats within the *critical* quadrant such as the ones proposed in Table 5 should be studied, because these are a priori the most problematic risks for the system. The minimum and maximum values for both probability and impact of threats within *critical* quadrant reveal the points which limit the interval that should be studied. While threat density indicates the amount of critical threats to be considered, the risk centre of mass indicates which are the mean probability and mean impact in the critical quadrant. The mean risk value in the *critical* quadrant is calculated straightforward using these two metrics in Equation (1). The threats showing the maximum risk and minimum risk in the *critical* quadrant would be the ones in the region with the highest and lowest values in Equation (1), respectively.

Table 5: Risk severity metrics

Risk metric name	Risk metric value
Threat density in Critical quadrant	$T =$ total number of threats in Critical quadrant
Point of maximum risk in Critical quadrant	$P_{max} = \max_T P_t$, $I_{max} = \max_T I_t$
Point of minimum risk Critical quadrant	$P_{min} = \min_T P_t$, $I_{min} = \min_T I_t$
Risk center of mass in Critical quadrant	$\overline{P_T} = \frac{1}{T}\sum_{t=1}^{T} P_t$, $\overline{I_T} = \frac{1}{T}\sum_{t=1}^{T} I_t$
Maximum Risk in Critical quadrant	$R_{max} = \max_T(P_t \times I_t)$
Minimum Risk in Critical quadrant	$R_{min} = \min_T(P_t \times I_t)$

Note that the risks of any of the nodes in the system ADT obtained from the risk propagation algorithm could also be classified in the risk quadrants. The metrics proposed in Table 5 shall then be used only with sets of independent threats which comparison makes sense, that is, the set of elementary threats in the leaf nodes as explained before, or the set of attack scenarios represented by the root nodes of disjunctive individual ADTs that compose the system ADT.

As complementary to these traditional approaches that consider attacks (and their respective controls) as independent events, we propose to evaluate the overall risk of the system by considering attack events' relationships defined by the ADT which enable risk-driven design of protection strategies based on the outcomes of the risk sensitivity analyses explained below.

3.4.1.2.8 Risk sensitivity analysis

Risk sensitivity analysis over the system ADT enables informed decision on system protection strategies. Provided all the risk attributes of the leaf nodes in the ADT are estimated, the risk sensitivity analysis investigates which attack events and which defences have more impact on the overall risk of the system. Moreover, the sensitivity analysis would allow to study risk variability in any of the tree nodes based on variations in the risk attributes of other nodes.

From the ADT proponent or attacker point of view, studying the swings in attack event attributes permits to identify which attacks produce a higher risk at system level, which are the minimum attack sets that realize the ADT at the lowest prize for the attacker, which do harm the system more, etc.

From the defender perspective, fluctuations in the defence attributes allow to deduce which protections do minimize most the overall risk or the overall attack impact, which would be the cost to fully minimize the risk of the system attack success, etc.

In the following we describe the three main types of risk sensitivity analysis that can be performed on ADTs:

- **Risk sensitivity to attack attributes**: In this analysis the value of the desired attribute (attack probability, impact, or cost) of one or multiple attack leaf nodes in the ADT is progressively increased in order to study the effects on the risk values in the rest of the tree

nodes. This analysis is relevant when studying how the risk of the root or of any branch in the tree varies with modifications in a specific attribute of selected attack events. Furthermore, the analysis is also relevant to study risk variations in ADT nodes caused by different increasingly successful (or increasingly impacting or increasingly costing) combinations of attack events (different sets of leaf-nodes). These studies are usually performed when details of one or some attack events are subject to uncertainty or when there is a wish to simulate small variations in the parameters of attack actions so as to better understand their influence on the risk of the complete attack-defence scenario.

- **Risk sensitivity to defence attributes**: Similarly, the impact of the defence attributes in the overall system risk or any subtree risk can be studied. In this analysis the value of the desired attribute (defence probability, minimized attack impact, or defence cost) of one or multiple defence nodes in the ADT is progressively increased in order to deduce the effects on the risk values in the ADT nodes. This analysis is conducted to study the impact of variations in one defence attribute on the overall risk or on the risk of a particular attack action, for example to understand to what extent the cost-effectiveness of the defence would impact in attack risk minimization. By selecting more than one defences in the analysis, it is possible to study risk variations in ADT nodes caused by increasingly successful (or increasingly effective or increasingly costing) combinations of defences. These studies are usually part of the defence strategy decision process when the attributes of one or a set of defences are being analysed to test their impact on the attack-defence risk scenario.
- **Risk sensitivity to combined attack and defence attributes**: This analysis combines the two previous ones where the value of one desired attribute (probability, impact, or cost) of a combination of defences and attack event nodes in the ADT is progressively increased in order to study the effects on the risk values in the remaining tree nodes. This type of risk analysis can be made for example when adjusting the design of the defences while there are some uncertainties on initially estimated attack attribute details.

The following Figure 13 describes the algorithm developed to evaluate the risk vector in the ADT tree root node when different attack-defence scenarios are simulated on the basis of variations of attributes in risk vectors of attack events, defences or both. When no attribute to simulate is entered, the default algorithm calculates the root node risk vector in all the possible defence combinations. In order to consider the defence is applied the algorithm sets the success probability of the defence to its nominal value (estimated value) and when it is not applied the probability is set to zero. When an attack set (AT) is marked to be simulated, the ADT root node risk vector will be calculated when the selected risk attribute (either probability, impact, or cost) of all marked attack event nodes are incremented from 0 to 1 in twenty steps. Similarly, when a defence set (DS) is the target of the simulation, the selected risk attribute of all the marked defences will be incremented in steps of 0.05 from 0 to 1 and the ADT root node risk vector will be evaluated. In both cases, the risk vectors of the unmarked nodes are the nominal ones initially estimated.

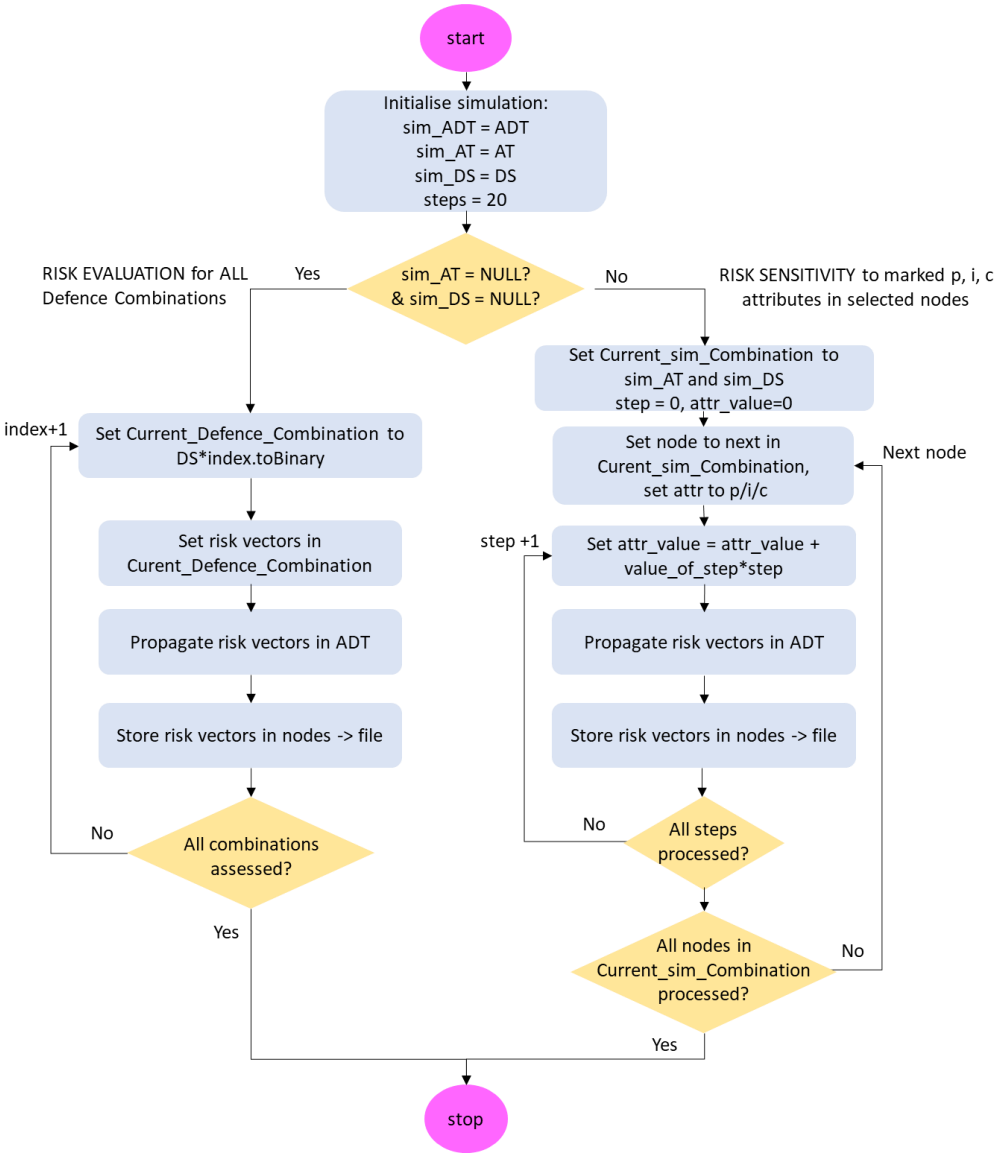


Figure 13: Algorithm to simulate attack-defence scenarios in an ADT

3.4.2 Risk-based optimisation of defences for multiCloud applications

Protections to system could be optimised with respect to overall security cost, efficiency or both. In this section we describe the proposed defence optimisation methods introduced by Roy et al. [85] extended and tailored to composite systems such as multiCloud applications. The major advantages with the original methods strive in three facts: i) our methods allow better informed decisions on the optimal defence set with respect to risk minimisation, ii) they enable to balance between component and system defences as they take into account the assets in which the defences are applied, and iii) the use of matrix computation yields to higher efficiency for large ADTs that are often found in multi-component and complex systems.

The methods developed not only enable the identification of optimal defence set to cover selected attacks against the system but permit to find out the optimal set in minimising risks. Moreover, the methods allow to assess system risks with respect to attributes of defences on particular assets. In multi-component systems when little information on system threats is available, the focus is often placed on protecting particular components which impacts the selection of the countermeasures to

adopt in the system. With our methodology, different risk minimisation effectiveness may be evaluated for instance when the set of threats to be covered includes all affecting the system or only those against a particular asset.

In the selection of the optimal defence or control mechanisms different optimisation problems may arise for whose solution the relationships between attack events and defences captured by the T matrix (described in section 3.4.1.2) will be used, as explained in the following sections.

3.4.2.1 Single Objective Optimization - Full cover of attack events

The selection of the minimal set of defences that fully covers all attack events in the ADT is a single objective optimisation problem, a special case of the attack set cover problem [169], which can be expressed as a binary objective function [85]:

$$F_1 = \min_{\forall OPT \in 2^J} \sum_{j=1}^{J} \mathbb{I}_{OPT}(D_j) : covered\ set = AT \tag{8}$$

Where $AT = \{AT_1, AT_2, AT_3, \dots AT_T\}$ is the set of all attack events in the ADT and $D = \{D_1, D_2, D_3, \dots D_N\}$ the set of all defences or countermeasures in the ADT. The $\mathbb{I}_{OPT}(D_j)$ is the *indicator function* such that $\mathbb{I}_{OPT}(D_j)$ = 1 if defence D_j is within the *OPT* optimal set of defences ($D_j \in OPT$), else $\mathbb{I}_{OPT}(D_j) = 0$.

As each of the countermeasures in an ADT mincut protects from all the attacks in the mincut, the minimal defence set would be the smallest possible suite of defences that contains at least one countermeasure from each mincut.

Therefore, the constraint $covered\ set = AT$ can be expressed in terms of the T matrix which captured all the mincuts in the ADT. The objective is to minimise the number of countermeasures selected from T matrix subject to the condition that each row (attack event) is covered by at least one column (defence). For a T matrix with $AT = \{AT_1, AT_2, AT_3, \dots AT_T\}$ attack events, $D = \{D_1, D_2, D_3, \dots D_J\}$ countermeasures and $A = \{A_1, A_2, A_3, \dots A_K\}$ assets, the constraint would be as follows:

$$\forall AT_i \in AT, \sum_{j=1}^{J} \sum_{k=1}^{K} t_{ijk} \times \mathbb{I}_{OPT}(D_j) \geq 1 \tag{9}$$

Where t_{ijk} represents the (i,j,k)th entry in the T matrix, and if defence D_j covers AT_i, for any asset $\forall k, t_{ijk}$= 1, else $\forall k, t_{ijk}$= 0.

To ease the computation of the full cover optimisation problem above, reduction techniques such as Branch & Bound [170] could be applied to the relationship matrix T. For a complete description of reduction technique usage please refer to Roy et al. [85]. Nevertheless, with current state of the art multi-dimensional array programming languages such as java script [171], Python [172] and R [173], the problem resolves easily in practical time scales even for large numbers of attack events and defences as described in validation section (Section 4).

The programme designed to solve Equation (9) therefore reduces the three-dimensional T matrix (of size $M \times J \times K$) to the mincuts that correspond to the *critical threat set* CTS (of size $1 \times M$ as the size of the AT vector), which M entries would all be ones, because all attacks should be covered, and identifies the minimum set of columns (defences) that covers all the rows (attacks).

First, we compute a matrix Z with all possible defence combinations for the ADT, which would have 2^J rows and J columns where J is the number of defences in the ADT, and z_{ij} entry in the Z matrix would be z_{ij}= 1 in case the defence is present in the combination, else z_{ij}= 0.

Second, when there is no constraint for the optimal defence set to be applied to any particular asset, a bidimensional matrix AD is computed from the three dimensional T matrix (of size $M \times J \times K$) as the result of the product of T with a vector *asset set AS* (of size $1 \times K$ as the size of A vector of assets in ADT), which K entries would all be equal to one indicating that all system assets are considered. The product results in a matrix AD (of size $M \times J$) which rows would be the attack events of the system and the columns the defences protecting from them; the coefficients 1 or 0 would indicate respectively whether the defence covers the attack or not:

$$AD = T \times AS \tag{10}$$

Note that in cases where the optimal set of defences is desired to protect vulnerabilities in a particular asset or component of interest, the procedure explained in section 3.4.2.4 shall be followed.

Third, we calculate the valid mincuts in the AD matrix by multiplying AD by each of the rows of matrix Z and discarding all resulting vectors (of size J) which do not have all their entries equal to one, that is, do not cover all the attacks. Each row of Z which produced a valid vector would represent a combination of defences that actually covers all attacks. Hence, the combination with the smallest number of defences will be the seek optimal minimum set of defences. Please note that more than one minimum defence combinations could be valid solutions to the full coverage problem.

As it can be seen, the algorithm designed obtains not only all the possible minimum optimal defence suites, but all the sets of defences that cover the attacks indicated. Moreover, as the mincuts in the ADT are treated independently, this algorithm can be parallelised which would further reduce computing time.

3.4.2.2 Single Objective Optimization - Partial cover of attack events

Partial cover of attack events occurs when due to different hampering factors such as limited security budget, limited security mechanisms available, etc. it is not possible for the system administrators to implement all the protections necessary to cover all the potential attacks but only a subset of them. The selection of the subset of attacks to cover could be the result of a previous analysis where a critical vulnerability set in the system is identified and only protections against attacks exploiting those vulnerabilities will be implemented, i.e. defences against the *critical threat set (CTS)*.

The partial cover problem therefore reduces to a special case of the full cover problem and the objective function of Equation (9) reduces to:

$$F_2 = \min_{\forall OPT \in 2^J} \sum_{j=1}^{J} \mathbb{I}_{OPT}(D_j) : covered\ set = CTS \tag{11}$$

Where $CTS = \{CTS_1, CTS_2, CTS_3, \dots CTS_M\}$ is the set of selected attack events that need to be covered and $D = \{D_1, D_2, D_3, \dots D_J\}$ the set of all defences or countermeasures in the ADT. The covered set CTS is generated from $AT = \{AT_1, AT_2, AT_3, \dots AT_M\}$ where the attack events that want to be covered are set to 1 and else set to 0. Hence, the constraint vector CTS would be of the form {0,1,0,1,1, ... 1} and size M as AT vector. As in Equation (9), the $\mathbb{I}_{OPT}(D_i)$ is the indicator function

such that $\mathbb{I}_{OPT}(D_i)$ = 1 if defence D_i is within the OPT optimal set of defences ($D_i \in OPT$), else $\mathbb{I}_{OPT}(D_i) = 0$.

Hence, the constraint $covered\ set = CTS$ can be formulated as a specialisation of Equation (9):

$$\forall AT_i \in CTS, \sum_{j=1}^{J}\sum_{k=1}^{K} t_{ijk} \times \mathbb{I}_{OPT}(D_j) \tag{12}$$

The program to resolve Equation (11) with the constraint of Equation (12) would first compact the AD matrix with just the attacks to be covered, i.e. it would create a CAD matrix (of size $m \times J$) from AD matrix (of size $M \times J$) , with m ≤ M where only the rows in AD corresponding to the m attacks of interest, i.e. those attacks set to 1 in vector CTS (of size $M \times 1$), will be extracted, that is:

$$CAD = AD_{CTS} \tag{13}$$

Where AD_{CTS} represents the AD matrix resulting from keeping only the rows of AD corresponding to the attacks set to 1 in the CTS vector.

Then, the search of the minimum optimal defence set in partial cover by using matrix computation follows the same procedure explained before for full cover of attacks, starting from the product of CAD matrix by Z matrix.

As in the case of full coverage problem, the solution found for the minimum set of defences in the partial coverage problem may be multiple optimal combinations, among which the security experts would need to decide on the basis of secondary functions to optimise, like minimising the cost of the defences, maximising the risk reduction or maximising the Return on Investment of the countermeasures, as described below.

3.4.2.3 Multiple Objective Optimization – Full/Partial cover of attack events with a second constraint

Often the selection of the optimal defences is made not only with respect to the covered attacks but tries to optimise other variables at the same time. This is the case for example when due to a limited security budget, the aim is to **minimise the cost investment** of the optimal minimum set which covers all attack events in the ADT (covered attack set is AT). The Equation (14) expresses the binary optimisation problem.

$$F_3 = \min_{\forall OPT \in 2^J} \sum_{j=1}^{J} \mathbb{I}_{OPT}(D_j) \times C_{D_j} \tag{14}$$

Where C_{D_j} represents the cost of the defence D_j.

The problem above could also be particularised to a *multi-objective partial coverage* case, when the optimal defence set includes only the countermeasures against the *critical threat set* (CTS) and the formulation would be similar to Equation (12) where the constraint would be *covered set = CTS*.

Figure 14 below describes the final algorithm created to solve the problem of dual objective optimisation searching for the optimal combination of defences which besides covering a specified set of attacks (CTS) optimises a second cumulative variable such as defence cost (i.e., fulfils F3 in Equation (14)). To this aim, besides the AD matrix and the CTS vector, the algorithm takes as input a vector with the weights or values that countermeasures take for the second variable to be minimised.

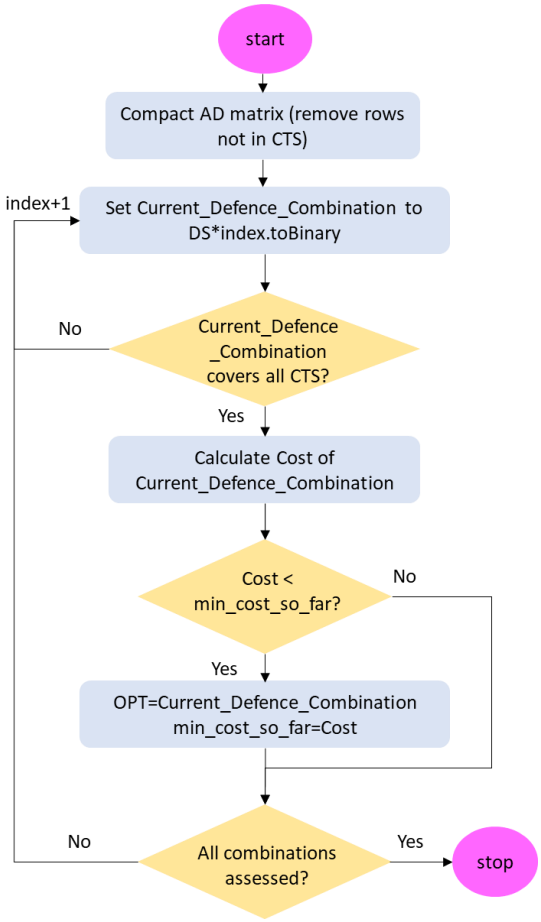


Figure 14: Algorithm to find the optimal defence set for an ADT with objective function F2 and F3

Please note that this algorithm can also be applied to full coverage dual optimisation cases where the defence set needs to fulfil together F1 and F3 objective functions.

A dual objective optimisation problem arises also when trying to maximise the Return on Defence or the Return on Investment of the optimal set of defences employed in covering the desired set of attacks. However, the values for these variables are not directly obtained from the defence decoration in the ADTs but need to be calculated as the result of the bottom-up propagation of risk vector in the tree. Hence, for this optimisation, the use of the algorithm in Figure 14 for objective function F1 or F2 (with no objective F3) is combined with the analysis of system risk sensitivity to defence attributes (see Section 3.4.1.2.8). As detailed in Section 4.6.6, dedicated simulation framework was developed in this Thesis for these analyses.

The Return on Defence (ROD) for the defender or system risk reduction is defined as the risk minimised in the system by the incorporation of a suite of defences S_{D_j}:

$$ROD_{S_{D_j}} = Risk_minimised_{sys_S_{D_j}} = \Delta Risk_{goal_S_{D_j}} \quad (15)$$

And:

$$\Delta Risk_{goal_S_{D_j}} = Risk_{goal_without_S_{D_j}} - Risk_{goal_with_S_{D_j}} \quad (16)$$

Where $Risk_{goal_with_S_{D_j}}$ is the risk of the root node in the ADT when the probabilities of all the countermeasure nodes in the defence suite S_{D_j} are set to their estimated values while the probabilities of the rest of the defences in the tree are set to zero, and $Risk_{goal_without_S_{D_j}}$ is the risk of the root node when no defences are applied, i.e. when the probabilities of all the defences in the set are equal to zero.

Hence, dual objective optimisation is required in order to identify the minimum set of countermeasures OPT that covers all the selected attacks in CTS while **maximises the system risk reduction** obtained, with objective function:

$$F_4 = \max_{\forall OPT \in 2^J} \Delta Risk_{goal_OPT} \tag{17}$$

In order to find out the best combination of defences that, while covering a selected attack set, is able to maximise the reduction of risk in the root node of the ADT, i.e. the optimal countermeasures set OPT that fulfils the objective function F1 of Equation (10) subject to the constraint of function F4, it is necessary to combine the optimal defence set search performed by algorithm in Figure 14 with the attribute-based risk sensitivity simulation in Figure 13. First, all the possible OPT sets that cover all the attacks are found by using Figure 14 routine. Second, the default simulation procedure in Figure 13 is used to compute the ADT root node risk vector in all the possible defence scenarios. Finally, the OPT defence scenarios are extracted from all the possible scenarios, and ranked by lowest risk attribute in the root node.

It should be noted that in all partial cover cases, the subset of attack events not being covered is a measure of the system's risk exposure which could be computed over the ADT by Equation (18):

$$Risk_{sys_exposure} = Risk_{goal_with_S_{D_j}} \tag{18}$$

In terms of economic gains, the **Return on Investment (ROI)** for the defender when implementing defence suite S_{D_j} can be defined as the gains achieved from risk minimisation, which can be expressed by adapting Sonnenreich's definition of the Return on Security Investment [174] to the context of risk in ADT, as follows.

$$ROI_{S_{D_j}} = \frac{\Delta Risk_{goal_S_{D_j}} - C_{S_{D_j}}}{C_{S_{D_j}}} \tag{19}$$

Where $\Delta Risk_{goal_S_{D_j}} = Risk_{goal_without_S_{D_j}} - Risk_{goal_with_S_{D_j}}$ represents the gains obtained from the incorporation of a suite of countermeasures S_{D_j}, i.e. the profit from system risk reduction by the implementation of only the suite S_{D_j} at a cost of $C_{S_{D_j}} = \sum_{j=1}^{J} C_{D_j}$. Please note that the reduction of risk to the system $\Delta Risk_{goal_S_{D_j}}$ shall be expressed in monetary terms just the same as security costs.

Therefore, in case the measure to be maximised is the ROI of the minimum set of defences to apply, then the objective function would be captured as:

$$F_5 = \max_{\forall OPT \in 2^J} ROI_{OPT} \tag{20}$$

As it can be seen, the dual optimisation of objective functions F1 and F5 presents a similar problem to the search of the optimal OPT as in the case of objectives F1 and F4 together, and the algorithm in Figure 14 can be applied as well.

Complementary to this work, please notice that the proponent or attacker perspective could be adopted in ADTs and calculate similar optimisations for attacks. For instance, the ADTs enable to estimate indicators of returns for the attacker such as the Return on Attack (ROA) which can be defined as the risk caused to the system by a suite of attacks S_{ATi} [175]:

$$ROA_{S_{AT_i}} = Risk_{sys_S_{AT_i}} = Risk_{goal_S_{AT_i}} = \frac{P_{goal_S_{AT_i}} \times I_{goal_S_{AT_i}}}{C_{goal_S_{AT_i}}} \quad (21)$$

Where $Risk_{goal_S_{AT_i}}$ is the risk of the root node in the ADT when the probabilities of all the attack event nodes in the attack suite S_{ATi} are set to their estimated values and the probabilities of the remaining attack events to zero.

The risk assessment software framework introduced in Section 5 is able to support proponent perspective analyses as well.

3.4.2.4 Defence optimisation in system vs. in individual components

The attack events in the system ADT target different parts of the system, i.e. each attack event exploits a specific vulnerability of a particular asset (component) of the composite systems such as multiCloud applications. Therefore, multiple branches in the ADT may include attacks against an asset under study and associated defences.

Defence optimisation following single or multiple objective optimisation techniques proposed could be performed on either individual components or the overall system. The results for system level optimal countermeasure set may significantly differ with the optimal set found out to protect a particular asset, and the system security and privacy experts shall decide on whether to invest in the reduction of the risks of a particular component or in minimising the risks at system level, which seems a priori more reasonable though other factors such as defence costs (in internal and outsourced components), implementation time, etc. may impact the final decision.

Optimising the defences at asset level is important for example when some system components are outsourced as in the case of multiCloud applications. As explained in Section 3.4.2.5, the selection of component providers may be done according to the resulting optimal countermeasure set to reduce the risks in the asset or the security costs to pay to external providers. Nevertheless, usually the identification of which countermeasures are required in an asset is not the outcome of a partial research on that asset but results from the study of risks to the whole system. Therefore, it is recommended to combine the analyses of defence optimisation at system and at asset level, with respect to threat coverage and risk minimisation, to conclude the final set to be implemented.

The defence optimisation at asset level is a particularisation of the system defence optimisation. The T matrix for the system ADT explained in Section 3.4.1.2.1 indicates which attacks and defences are associated to a particular asset or component. By selecting the page of the T matrix corresponding to the asset under study, defence optimisation could be performed considering the covered attack set is composed only by attacks against the selected asset, i.e. attack rows which have any entry value equal to 1 in the asset page. The selection of the specific asset or combination of assets to be studied is made by entering in Equation (10) a vector *asset set AS* (of size $1 \times K$ as the size of A vector of assets in ADT), which k[th] entry a_k would be $a_k = 1$ if the asset is required to be part of the study and else $a_k = 0$.

Consequently, for full cover optimisation all attacks against the asset will be selected within the CTS vector, while for partial cover optimisation only a subset of these will be included in the CTS. It should be noted that an asset may not appear in all the ADT mincuts, and therefore, the minimum

optimal countermeasure set found out for an asset may not necessarily protect from all the possible system attacks.

In security cost optimisation cases, the experts shall decide whether the cost of countermeasures for the asset amount for all the available defence budget for the system or just a portion of it. Once the maximum defence cost for the asset is known, the dual optimisation is similar to the process explained for the system but limiting the analysis to the asset page in T matrix.

3.4.2.5 Risk-driven selection of service providers

In cases when third-party software components are part of the composite system architecture, such as in multiCloud applications where cloud services are outsourced, it is necessary to assess the risks over these components with all available information about vulnerabilities they may have and security controls the providers offer to protect them.

To this aim, first the system ADT including attacks and defences for these components shall be created which may result in a challenging task when little information on the external component is available.

3.4.2.5.1 Identification of required defences in outsourced components

The modelling of third-party component attacks and their attribute decoration shall be done in the ADT with the information on potential attacks obtained from the provider if possible or from a prior research on potential threats against the component from previous survey or experiences with components of similar characteristics.

The defence modelling in the system ADT shall be done with the defences specified in the security service level agreement (SLA) offered by the provider, which indicates the controls implemented to protect the component together with their price. Usually even if the third-party SLA includes the available controls, the corresponding threats or attacks are not explicitly indicated, and system security experts should add in the ADT attacks associated to each of the defences in the SLA in case they were not identified yet.

Then, the study of the countermeasure optimisation at asset level could be used to identify which countermeasures are the most appropriate to cover all the attacks and minimise the risks in third-party components and this, together with the system level defence optimisation, will enable the informed selection providers offering the resulting optimal countermeasure set. The decision on which are the required defences or controls in the asset would be made from the balance between optimal countermeasure set in the asset and in the system as explained in Section 3.4.2.4.

3.4.2.5.2 Decision on service providers

Once the required defences are identified for each outsourced component, it is possible to search for service providers that offer them in their security Service Level Agreement (SLA) or terms of use.

Taha et al. [176] and Farokhi et al. [177] proposed effective SLA-based methods to search for the best match of cloud service providers offering the desired controls for multiCloud applications. These techniques allow to rank providers with respect to the security requirements fulfilment rate expressed as the percentage of offered controls from the desired control set and the degree of control level achievement.

However, it is not always possible to find providers that offer all the needed defences for the asset. Furthermore, even if providers offering all the defences in the required set are found, their price may exceed the available security budget and thus a dual optimisation problem with a cost objective

in outsourced countermeasures for the asset would be faced which should be solved as explained before.

The result of provider selection for the asset impacts the defences that will finally be available in the asset. This information shall be added in the system ADT to refine the raw risk assessment made for the asset and for the system following the risk attributes rules in Table 3 and Table 4. The subsequent assessment of how risks at system and asset levels are impacted is explained in the next section.

3.4.2.5.3 Risk assessment refinement after deployment

Once the selection of the final external service providers of the third-party components such as Cloud services is made, the system ADT should be refined with the specific defences that providers offer in their components, i.e. the tree should be updated and the defences offered added in the form of standard security controls associated to the threat events in the leaf-nodes together with the best estimates as possible for their risk attributes. The estimation of the attributes for actual defences in outsourced components shall be obtained with the help of the providers who have full insight of the service or from analysis of service documentation which may include a self-assessment such as Consensus Assessments Initiative Questionnaire (CAIQ) [144] in STAR Registry [145].

When placing the focus on a specific asset, the system risk sensitivity analysis can be done by varying the attributes of attacks and defences for the asset and studying how they impact risk attributes of the ADT main goal. In addition, it is possible to analyse in parallel the severity of individual attacks against the asset and the efficiency of the defences by representing the attack events in the probability-impact graph, where low, medium and high risk quadrants are defined as shown in Figure 12. This representation together with metrics in Table 5 enable to learn the impact over asset risks that modifications of the attributes in tree leaves corresponding to the asset may have.

3.4.2.6 Continuous monitoring of attacks and defences

The initially assessed risk needs to be revisited continuously during system operation due to attacks may occur against the system assets or defences deployed may not work properly leaving the assets unprotected. Therefore, it is necessary to continuously re-evaluate system risks to include the effects of detected attacks or to update the controls deployed according to their actual status.

When using the ADT based method proposed, the continuous risk assessment consists in iterative refinements of the risk evaluation based on updates performed on risk attributes of attack and defence nodes made according to the sensed system status. Continuous security monitoring would allow for revisiting the probability and impact values of attacks and defences in the ADT, while the estimated costs for the attacker and the defender will less likely require frequent updates. In fact, defence costs shall be set to the actual expenditures in the final security resources employed.

Whenever the monitoring system detects a threat AT_i is materialised into an attack event, the attack probability of success would be set to 1, i.e. P_{AT_i}= 1, which would rise the risk of the root node in the ADT. Similarly, when the damaged caused on the system by detected attack are studied, the impact attribute of the event node could be updated, and it may be possible that this information helps in the refinement of impact values for similar attacks in the ADT as well. Once the reaction mechanism to counteract the detected attack is in place, the corresponding defence shall be added in the ADT model and a new risk assessment iteration shall be performed.

Defence monitoring will aid in updates to defence node attributes as well. Risk exposure would increase whenever a deployed protection D_j fails and a new calculation of root node risk shall be

made by considering its probability of success as zero, i.e. P_{D_j}= 0. Thanks to continuous defence performance check, empirical tests of defence success and effectiveness at operation can be used to tune either the estimated success probability or the attack impact reduction ratio (i.e. I_{D_j}) for the defence nodes in the tree.

Please refer to [157] for a comprehensive description of an example monitoring solution addressing continuous check of the status of both attacks and defences in multiCloud systems.

3.5 Security and Privacy SLA composition for multiCloud applications

In this section we describe the fourth contribution of the Thesis as per Figure 2, i.e. the Security SLA and Privacy SLA (PLA) composition methodology for multiCloud applications.

Security and privacy assurance in complex architectural scenarios like multiCloud applications where multiple distributed components are orchestrated to provide system features depends on the protections offered by each of the single components and the infrastructures used. Therefore, assessing overall security and privacy level of composed applications requires the analysis of each of the components in the system and how it impacts in the overall architecture. This way it is possible to learn the security and privacy features that can be promised to application consumers which are served the whole application running as a single service.

The expression of security and privacy policies as Service Level Agreements (SLA) in form of structured collections of standard controls avoids ambiguity in the description of the policies and facilitates the comparison between policies and their evaluation.

In the following we describe the methodology proposed first we recall the main relationships between the terms of Security SLAs, Privacy SLAs and controls to express the Service Level Objectives (SLOs) in the SLAs. Then, we describe in detail the controls gathered in the standard Security Control Framework adopted in this Thesis: the NIST SP 800-53 Rev. 5 [119].

3.5.1 Security- and Privacy- aware SLA terms

Considering the close relationship between privacy and security assurance and the fact that many of the security controls applicable in Cloud-based systems do support also privacy protection in cases when the data are personal data (or in terms of GDPR [8], personally identifiable information, PII), we propose to address simultaneously the technical specification, deployment and assurance of both types of controls.

The first step to this integrated approach would consist in the definition of the Service Level Agreement of the multiCloud application that is security and privacy-aware in the sense that captures the definition of both aspects of the application.

Therefore, an integrated reference metamodel for multiCloud SLAs has been developed which is able to support not only security controls specification but also privacy controls and joint controls specification. The integrated SLA metamodel developed for our approach is shown in Figure 15. The model is a derivation of the SPECS Security SLA metamodel [178] where we have consolidated the concepts to embrace the privacy perspective, thus integrating Security SLA with PLA.

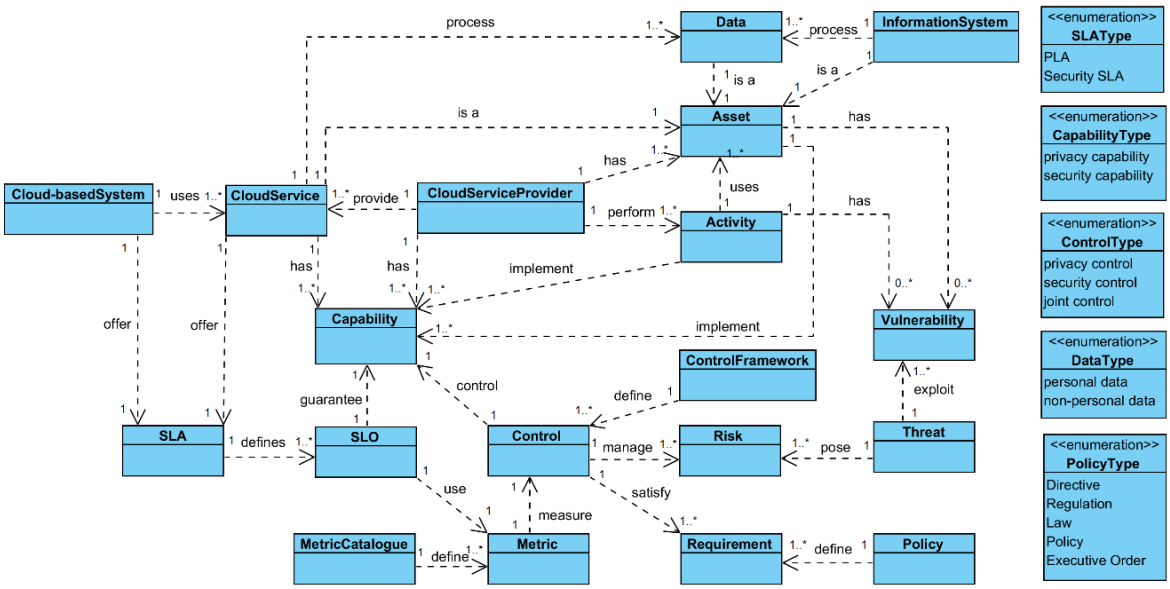


Figure 15: SLA model integrating PLA and Security SLA.

As shown in the model, an *SLA* defines the *Service Level Objectives (SLOs)* and associated *controls*. *Controls* ensure that the Cloud service's and/or the organisational CSP's *capabilities* satisfy the necessary *requirements* derived from the *policies*, which can range from regulations (like GDPR) to organisational policies or orders. The *SLOs* are expressed in terms of *metrics* to quantitative and unambiguously specify the capability levels guaranteed in the SLA. Therefore, Security SLAs associate to each service both the *security controls* that are implemented on top of it and the *Service Level Objectives (SLOs)* of the *security capabilities* of the service and its provider.

As explained in Section 2.6.9, different control families can be used to express the security and privacy controls of the systems in a standard way. These include but are not limited to NIST SP 800-53 Rev. 5 [119], ISO/IEC 27002 [121], SO/IEC 27017 [6], ISO/IEC 27018 [7], CSA CCM [123], etc. These catalogues aid in the formal specification of required and/or offered (provided) capabilities of the IoT System, its constituent components and the organisations providing them. In those cases where personally identifiable data is processed by any of the components of the system, privacy controls need to be adopted on top of security controls.

In our approach we adopt the NIST SP 800-53 Rev. 5 [119] to define the controls in the SLA, which extends the previous version of the framework and defines, besides *security controls*, *privacy controls* that are specifically devoted to meet privacy requirements and to manage the privacy risks in an organisation, and *joint controls* that can meet privacy and security requirements at a time. Security controls are defined by NIST as *the safeguards or countermeasures prescribed for an information system or an organization to protect the confidentiality, integrity, and availability of the system and its information*, while privacy controls are *the administrative, technical, and physical safeguards employed within an agency to ensure compliance with applicable privacy requirements and manage privacy risks*. The details of this standard are given in the next subsection.

Particularly to Privacy SLAs, the CSA's PLA template collects high-level privacy controls for CSPs being able to specify their privacy policy. In the table of Appendix A we propose the main relationship between the 94 privacy controls in the CSA's PLA, the provisions of the GDPR and the Security SLA controls. As seen in the table, we could say that the PLA contains or can refer to the Security SLA of the personal data, as security mechanisms applied by the processor on the PII

are required to be expressed as part of the PLA. As explained before, the SLA controls could be expressed by using those of the CSA's CCM [123] or any other security control framework.

3.5.2 The controls in NIST SP 800-53 Rev. 5

As the major exponent of internationally recognised control catalogues, we base our work in the NIST SP 800-53 Rev. 5 [119] which offers fine-grained controls and is publicly available for free. The main advantages of NIST SP 800-53 Rev. 5 [119] over other security control frameworks for Cloud such as Cloud Security Alliance's Cloud Control Matrix (CCM) [123] and ISO/IEC 27017 [6], are its greater maturity, granularity of the controls and the integration of privacy and security controls.

The NIST SP 800-53 Rev. 5 [119] organises the controls in families, such as Access Control (AC), Identification and Authentication (IA), Risk Assessment (RA), System and Communications Protection (SC), System and Information Integrity (SI), etc. And a new Privacy Authorization (PA) family has been added. The standard collects 912 controls grouped in security controls (752 controls), privacy controls (59 controls) and joint controls (101 controls, which serve either security or privacy or both). A total of 160 privacy-related controls are identified by NIST SP 800-53 Rev. 5 [119] from 12 different *control families* or categories, namely: AC – Access Control, AT – Awareness and Training, AU – Audit, CA – Continuous Assessment, CM – Configuration Management, CP – Contingency Planning, IA – Identification and Authentication, IP – Individual Participation, IR – Incident Response, MP – Media Sanitization, PA – Privacy Authorization, PL – Planning, PM – Project Management, RA – Risk Assessment, SA – System and Services Acquisition, SC – System and Communications, SI – System and Information Integrity.

NIST SP 800-53 Rev. 5 [119] also distinguishes between organisational controls ("O" – a control implemented by a human in the organization through nontechnical means) or system controls ("S" – a control typically implemented by an organisational system through technical means) or controls implemented by either or the combination of both nontechnical and technical means ("O/S"). The privacy related "S" and "O/S" controls identified by the standard are only 12 and 15 respectively, summing up a total of 27 technically implementable privacy controls. Limiting the analysis to base controls and not considering enhancement controls, there are only 67 "S" or "O/S" base controls, while 62 are security base controls and only 5 privacy base control. Therefore, pursuant to NIST, most of the means for tackling with privacy assurance in the systems reside at the organisational or procedural level rather than at system level.

For informative purposes, Table 6 collects the system level base controls related to privacy identified by NIST together with their description.

Table 6: Privacy related base controls "S" and "O/S" in NIST SP 800-53 Rev. 5

Control ID	Control name	Description
SC-16	TRANSMISSION OF SECURITY AND PRIVACY ATTRIBUTES	Associate [Assignment: organization-defined security and privacy attributes] with information exchanged between systems and between system components.
SI-6	SECURITY FUNCTION VERIFICATION	a. Verify the correct operation of [Assignment: organization-defined security and privacy functions]; b. Perform this verification [Selection (one or more): [Assignment: organization-defined system transitional states]; upon command by user with appropriate privilege; [Assignment: organization-defined frequency]]; c. Notify [Assignment:

Control ID	Control name	Description
		organization-defined personnel or roles] of failed security and privacy verification tests; and d. [Selection (one or more): Shut the system down; Restart the system; [Assignment: organization-defined alternative action(s)]] when anomalies are discovered.
SI-18	INFORMATION DISPOSAL	Use [Assignment: organization-defined techniques or methods] to dispose of, destroy, or erase information.
SI-19	DATA QUALITY OPERATIONS	a. Upon collection or creation of personally identifiable information, check for the accuracy, relevance, timeliness, impact, completeness, and de-identification of that information across the information life cycle; and b. Check for and correct as necessary [Assignment: organization-defined frequency] and across the information life cycle: 1. Inaccurate or outdated personally identifiable information; 2. Personally identifiable information of incorrectly determined impact; or 3. Incorrectly de-identified personally identifiable information.
SI-20	DE-IDENTIFICATION	Remove personally identifiable information from datasets.

According to NIST SP 800-53 Rev. 5 [119], controls may address security and privacy from two angles: i) from a functional perspective to ensure the strength of the mechanisms and functions implementing the security and privacy capabilities of the system or service, and ii) from an assurance perspective to measure the confidence of the security and privacy capabilities. In either case, controls can be considered as the security and privacy capabilities of the service and its service provider. These controls can be expressed within the SLAs in terms of SLOs and associated metrics because the implementation of each control could be associated to a value assessing the security or privacy capability level (capability strength or confidence). The metrics can be either qualitative (e.g. “YES/NO”, “low/medium/high”, etc.) or quantitative (e.g. Boolean for control implemented or not, numerical or an enumeration element).

In the last revision of NIST catalogue [119], all security and privacy control definitions follow the imperative wording with actions to be fulfilled by the organisation or system implementing the controls, such as: “Enforce”, “Employ”, “Provide”, “Implement”, “Identify”, “Isolate”, “Require”, “Allocate”, “Prevent”, “Separate”, etc. The assessment of all these can be performed through a qualitative scale in most of the cases (e.g. “YES/NO implemented”) or by means of a quantitative metric (e.g. the encryption strength in AC-17(2) Remote Access | Protection Of Confidentiality And Integrity Using Encryption could be based on encryption key size with possible values {64bits, 128bits, 1024bits, …, 2048bits}). The quantification of the metrics allows to quantify the security and privacy Service Level Objectives to declare within the multiCloud Application SLA, which is a key step in the proposed composition methodology as described below.

3.5.3 SecSLA and PLA Composition methodology

In this section we explain the method proposed to compose the SLA of the multiCloud application on the basis of the combination of the SLAs of its integrating components. The resulting Application SLA will be the SLA offered by the application provider to its own clients. Note that the application provider is a Cloud Service Consumer (CSC) when any of the application components is deployed

in or consumes a Cloud service (of either IaaS, PaaS or SaaS type) or an IoT service consumer when any of the components is deployed in an IoT infrastructure or platform.

The study of the security and privacy controls that can be granted by multiCloud distributed applications as a result of the combination of multiple components' controls the application uses is a challenging task.

The process to identify and formalise the security and privacy controls that can be guaranteed in the overall composed application policy, the so-called composed Application Service Level Agreement (SLA), depends on individual components controls and how the components are distributed in the architecture and deployed in multiple providers.

The SLA composition process proposed includes the steps illustrated in Figure 16 and briefly summarized in the following:

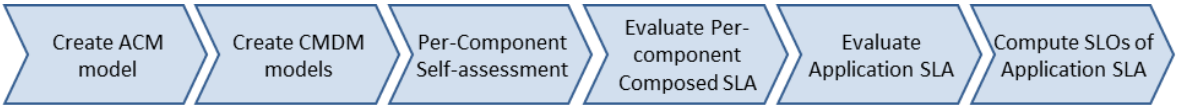


Figure 16: The multiCloud Application SLA Composition Process.

- **Step 1: Create the Application Composition Model (ACM)** following the method proposed by Rak in [129]. The ACM identifies the software artefacts or components in the composed application and how they are interconnected and distributed in Cloud and IoT resources.
- **Step 2: Create the Control Metric Delegation Model** (CMDM) for each of the controls that need to be assessed in the application SLA. The CMDM captures the control metric delegation relationships between the nodes in the ACM model. These relationships will drive the application of the SLA composition rules evaluation in Step 4. The detailed modelling method to create the CMDMs of the system is explained in next subsections.
- **Step 3: Perform the per-component self-assessment of application components.** In this step the individual SLA templates (SLATs) of the internal components and the SLAs of third-party components are obtained. The SLATs of internal components provide the set of security and privacy controls the component is able to grant by taking into account only the internal capabilities of the component, regardless the target deployment infrastructure capabilities. The SLAs of the outsourced services impact directly on how the application behaves in terms of security and privacy too. Hence, both the SLATs of internal components and SLAs of third-party components will be necessary for the computation of the composition rules in Step 4.
- **Step 4: Evaluate the per-component SLA composition rules**. The per-component assessment (step 2 above) identifies the controls that each of the components is able to grant considering its own implementation, but such evaluation cannot take into account how the security controls and the SLOs granted by other components affect each other. During the per-component SLA composition we define the SLA that each component effectively grants to the others taking into account all the relationships among components (outlined by the ACM model) in order to learn which specific controls can be finally granted in the application SLA by following the process described in section 3.5.3.4.
- **Step 5: Evaluate the Application SLA**. Once the controls that can be granted by each component are identified, it is possible to evaluate the SLA at application level, that is, identify the controls the composed application is able to grant to its own customers.
- **Step 6: Compute the SLOs of the controls** that can be declared in the Application SLA by applying the technique explained in section 3.5.3.6. This step enables the calculation of the

security and privacy capability levels of the composed application in the form of the levels that the controls could actually reach on the basis of the individual components' ones.

In the following subsections we provide the details of the methodology steps above.

3.5.3.1 Application Composition Modelling

The first step of the SLA Composition methodology is to identify the architectural internal components of the system together with the deployment and service provision relationships between them and with other external components. The components and their relationships will be captured in the Application Composition Model as proposed by Rak in [129] which will enable the reasoning over the security and privacy of a composed application made of a collection of cooperating software components, which in turn will be offered as a single service or system. A component can be directly offered as-a-service by a Cloud Service Provider (CSP) or IoT provider, or by deploying a suitable software artefact over a cloud or IoT infrastructure capability type (i.e., over a virtual machine or over an IoT edge or device).

The Application Composition Model (ACM) of a composed application therefore captures the deployment and service provision relationships among the components, i.e. which components are internal components (i.e. components not provided by third parties but developed by the team constructing the application), which are external components (provided by third-party providers), which are deployed in the Cloud, in IoT platform or edge, which on-premises, which consume other off-the-self services, etc.

As shown in Figure 17, the ACM graphically models the application architectural internal and external components in form of nodes that represent either services (which could be Cloud services, such as Infrastructure-as-a-service, Software-as-a-service, and Platform-as-a-service) or service providers. The last ones are denoted as "CSP" for Cloud Service Provider or "SP" for Service Provider in general, i.e. a non-Cloud service such as an on-premise service or IoT service. The interactions between the services are modelled as "uses" and "hosts" edges and providers have "provides" relationships with the services offered by them. In the background, services are associated to their corresponding Service Level Agreements (SLAs) and Service Level Agreement Templates (SLATs) where a service "grants" a SLA, and "supports" a SLAT, and a provider "grants" the SLAs of the services it "provides".

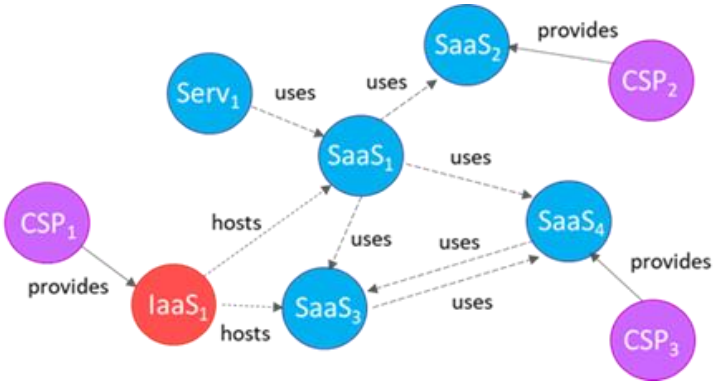


Figure 17: Example of an ACM showing its structure.

The particular ACM example in Figure 17 represents an application with one non-Cloud service, $Serv_1$, that uses four components deployed in multiCloud consuming from the three CSPs (CSP1, CSP2 and CSP3) different Cloud services: the Infrastructure-as-a-service IaaS1 that hosts the Software-as-a-service couple SaaS1 and SaaS3 is provided by CSP1, while CSP2 and CSP3 provide SaaS2 and SaaS4, respectively.

3.5.3.2 Control Metric Delegation Modelling

In this section the method to create the Control Metric Delegation Models for the application is explained.

When constructing the SLA of a multicomponent system such as a multiCloud application, besides the deployment and usage relationships between the components captured by the ACM, it is necessary to identify which are the dependencies between the components or services (nodes) with respect to the implementation of the security and privacy controls in the SLA. The main reason for this is that the ACM abstracts the service capability usage relationships between the nodes of the system which do not necessarily reflect the security (and privacy) capabilities implementation and usage relationships between the nodes.

Therefore, we propose a general model of SLA composition based on control metric declaration conforming to control implementation delegation relationships between nodes in the ACM as follows.

The starting point is the classification of the controls by the NIST SP 800-53 Rev. 5 [119] control catalogue, which distinguishes three types of controls:

- common controls: A control is deemed common or inheritable "*when the information system or program receives protection from the implemented control but the control is developed, implemented, assessed, authorized, and monitored by entities other than those responsible for the system or program*".
- system-specific controls: "*are the primary responsibility of information system owners and the authorizing officials for those systems*".
- hybrid controls: correspond to cases when "*one part of the control is common and another part of the control is system-specific*".

As described by NIST the distinction cannot be made based on the control wording, but it shall be made by humans in a case-by-case analysis, and for example, one control can be designated as common control in one information system and as system-specific control in another.

Provided the controls of an information system follow the NIST classification above, for each control in the component SLA we could create a control dependency model that specifies the different delegation relationships of control implementation between nodes in the ACM. The model is built by considering three types of controls:

- *Delegated control*: A control which implementation is delegated from the source node to one or multiple target nodes, i.e. it needs to be fully implemented by other target nodes and therefore, all the metrics on the capacity assessed by the control are obtained in the target nodes and none in the source node. Common controls are fully delegated controls.
- *Owned control*: A control which implementation cannot be delegated and needs to be implemented by the node. Therefore, all the metrics on the control existence and performance will be assessed in the node and none can be delegated. System-specific controls correspond to this case.
- *Hybrid control*: A control which implementation is partially owned by the source node and partially delegated to one or multiple target nodes. Hence, the control metrics shall be assessed in all the nodes sharing the implementation. Hybrid controls fell under this category.

Pursuant to the specification of control types and the implied metric delegation relationships above, it is possible to assess the implementation of a control in a component based on the assessment of

the control parts implementation in the component or in the components to which it delegates the parts.

Hence, for a given multiCloud application, it is possible to define a set of Control Metric Delegation Models (CMDMs) each of which captures for each control under study a different view of the ACM expressing the control metrics implementation delegation relationships between the nodes in the ACM, as per whether the control is owned, delegated or hybrid control.

The CMDM is a directed graph (digraph) where the vertices represent the ACM nodes and the edges the delegations of the implementation of the control parts measured by specific metrics. For short, we will use the term "delegation/ownership of a metric" to refer to the "delegation/ownership of the implementation of the control part measured by a specific metric". In the CMDM, the metric ownership relationships are modelled by loops. Please note that the formal specification of the metric delegation relationships modelled by the CMDM are provided later when explaining how to evaluate the component Composed SLA on top of the CMDMs in section 3.5.

Figure 18 represents a Control Metric Delegation Model example for the multiCloud application with nine nodes (vertices No 0 to 8) in the example ACM of Figure 17.

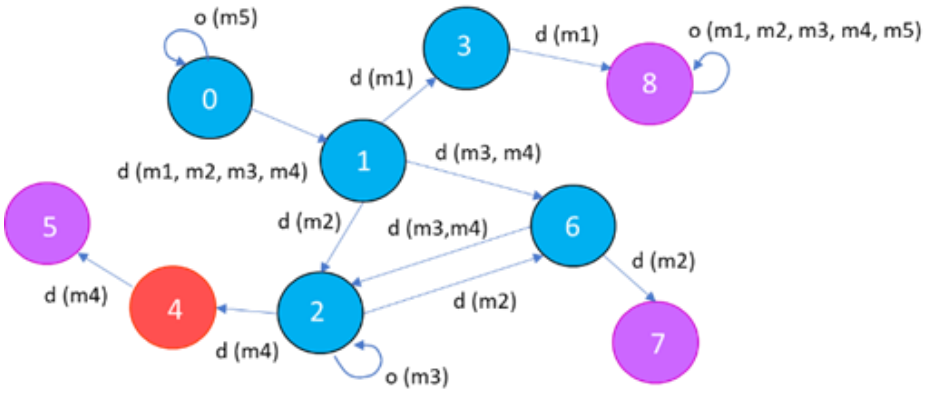


Figure 18: Example of a CMDM between the nodes in the ACM of Figure 17.

The CMDM model represented corresponds to a control which will be assessed by a set of five metrics {m1, m2, m3, m4, m5}. On one hand, the control is an owned control for node 8 which does not delegate its implementation to any other node. The ownership is depicted in node 8 as a self-delegation or loop. On the other hand, the control is a hybrid control for node 0, which implements only the mechanism measured by metric m5 and delegates to node 1 the implementation of the mechanisms measured by the metric set {m1, m2, m3, m4}. Node 1 in turn delegates the implementation of the control to nodes 2, 3 and 6, to which it delegates m2, m1, and {m3, m4} respectively. Node 3 delegates partially the implementation of the control to node 8, to which it delegates m1. Similarly, node 2 delegates the implementation of the part of the control corresponding to m4 to node 4. In turn, node 4 further delegates m4 to node 5. While node 2 keeps the implementation of the mechanism measured by m3, it further delegates the implementation of m2 to node 6, which relies on node 7 to implement it.

As it can be seen, the assessment of delegated and hybrid controls poses the major challenges to the SLA composition as their assessment needs to be carried out in all the nodes that participate in the delegation chains for the metrics of that control. On the contrary, owned controls would be assessed exclusively in the source node. Multiple nodes could intervene in a metric delegation chain while a source node cannot delegate the same metric to different target nodes. This way, the metric that measures the implementation of a part of a control would always be measured in a single node, i.e. no further partitioning of control implementation parts is allowed.

3.5.3.3 Per-component self-assessment of SLAs

The self-assessment is a necessary step towards understanding which security and privacy controls are being implemented in the application components. The activity consists in checking which controls from the selected security control framework the component is able to guarantee by itself, as follows.

3.5.3.3.1 *Self-assessment of internal components' SecSLA and PLA templates*

For components developed by the own development team, the so-called *internal components*, the identification of which are the actual controls implemented by the components considers only the internal implementation of the component independently of the Cloud or IoT platform and services it will be deployed in.

Self-assessment is a best practice promoted in the community of security and privacy experts in different fields and different approaches of performing this analysis can be adopted. At Cloud layer, the Cloud Security Alliance promotes the use of the Consensus Assessments Initiative Questionnaire (CAIQ) [144] that provides a set of Yes/No questionnaire for assessing the status of the controls in the Cloud Controls Matrix CSA CCM [123]. Casola et al. [114] proposed a self-assessment method in multiCloud environments where the application developers or product owners would go through a guided checklist indicating which security controls from the NIST SP 800-53 Rev. 4 were offered by the component or service under analysis. In our approach we promote the use of latest NIST SP 800-53 Rev. 5 [119] instead, in order the analysis can include privacy controls if necessary. According to the nature of the system, the needs and interests of the organisation, etc. other control frameworks could be adopted such as ISO/IEC 27017 [6] and ISO/IEC 27018 [7] for Cloud, and more generally ISO/IEC 27001 [120], etc.

In web application context, the Application Security Verification Standard (ASVS 4.0) [158] proposed by OWASP could be used as the basis of the security features check. Similarly, Berkley DB best Practices [159] may help in assessing database security features and correct implementation of countermeasures in very specific contexts.

The component self-assessment will result in the list of controls already available in the component and thus not needed to be requested to any third-party component. As these controls are internal capabilities of the component responding to internal configuration and not taking into account the impact future deployments, they constitute the template for the policy of the component or, in other words, the SLA template or SLAT as defined by Rak [129], i.e. the SecSLAT with self-assessed security controls, or PLAT with self-assessed privacy controls in cases where the component processes any personal data.

Remarkably, in the self-assessment it is necessary to include the selected enforcement agents for the multiCloud application, if any, which would implement security and privacy mechanisms and thus would have to be part of the MACM and follow the assessment process to learn the SLAs they can grant so as it can be taken into account in the composition. In case the enforcement agents are outsourced components their SLA would need to be get from the corresponding provider, as explained in the next subsection.

Once all internal components in the system are assessed, the developers will have a clearer picture of the security and privacy posture of the components integrating the system in form of a well-structured collection of offered security controls and privacy controls. In some cases, internal components or services in the application may consume other services by third-party service providers, and therefore the self-assessment will need to focus on the owned part only for which the developers have control on its functionality and behaviour. For external components the following step should be followed.

3.5.3.3.2 *Identification of the SecSLAs and PLAs of outsourced components*

In cases when the composed application exploits external third-party components or services, which may be Cloud services (Infrastructure as a Service, Platform as a Service or Software as a Service) or IoT services (e.g. Raspberry Pi [179] or Arduino [180] computing platforms), it is necessary to learn in advance which are the offered security and privacy policies of these components. Usually, the policies depend on the type of service and/or supporting device (e.g. the available encryption modes in a gateway, the authentication mechanism used by a sensor, etc.). This information is generally accessible from the service/component (or device) provider, though often they do not come in form of well-structured SecSLAs and PLAs.

In some cases, the offered SLA will not be known beforehand, and it would need to be constructed from the information in the usage policy or privacy policy of the service, most likely from the service usage terms, the provider website, or product specification (e.g. platform manual in case of IoT edge). The construction of the outsourced components' SLAs would require mapping all the public information on service provision terms to standard controls in the selected control family. The process would be similar to the self-assessment done on owned internal components, though for some controls there may be a lack of information and it would need to be requested to the provider whenever possible.

In this process, it is the task of the application developers to try to homogenise the way the offered security and privacy mechanisms are specified for external components with that of the internal components, so the overall security and privacy posture of the composed system is understood and can be controlled easily. It advisable that the same security control framework used for building internal components SLAT and PLAT is used for the external components as well.

In the case of CSPs, the CSA's STAR Registry [145] can be used to support the creation of external cloud components' SLAs. The STAR is a publicly accessible repository of cloud controls self-assessed by a significant number of CSPs. The self-assessed controls in STAR were obtained through responses to a dedicated questionnaire CAIQ [144] which assesses the adoption of Cloud Control Matrix controls, which gathers security controls rather than privacy controls.

3.5.3.4 Evaluation of the Per-Component SLA Composition rules

In this step we propose a method to obtain in a per-component basis the Component Composed SLA, i.e. the SLA that each component is able to grant considering the relationships with all the other pieces of the composed application. The method adopts Rak's [129] SLA composition technique for security controls of multiCloud applications and extends it to take into account privacy and joint controls as well. Furthermore, we generalise the methodology to a wider scenario where application deployment layout may require not only multiple Cloud services but also multiple IoT services and resources.

The evaluation of the SLA composition rules for the application components assumes that three previous steps in Figure 16 are already performed, that is, modelling application architecture in the ACM, obtaining the internal components' SLA templates by self-assessing their capabilities, and finally, getting the external components' SLAs.

We define Control Set (CS) as the set of all the security and privacy controls in our security control framework (NIST SP 800-53 Rev. 5 [119]):

$$CS = \{c_j : a\ control\ defined\ in\ NIST\ SP\ 800-53r5\} \quad (22)$$

For each control in CS, we denote the set of the security metrics able to assess the control c_j implementation as follows:

$$M_{c_j} = \{m_i : m_i \; assess \; c_j\}_{i=1}^{K} \tag{23}$$

where *K* is the number of metrics associated to that control.

A metric delegation situation exists between two nodes in ACM when one of the nodes delegates the measurement of the metric assessing the implementation of (part of) a control to the other. Note that metric delegation cannot be done over multiple nodes, and it is only done to one node, which may further delegate the metric to another node.

We could formally define this as follows:

Definition 1: We define the **control metric delegation** $D(cj, m_k, n_i, n_z)$ as the Boolean function which assumes value 1 when the node n_i delegates to another node n_z, with $z \neq i$, the implementation of the part of the control c_j measured by a given metric m_k, and 0 otherwise:

$$\begin{cases} D(c_j, m_k, n_i, n_i) = 0 \\ D\,(c_j, m_k, n_i, n_z) = 1, \quad if\ n_i\ delegates\ m_k\ to\ n_z \\ D(c_j, m_k, n_i, n_z) = D(c_j, m_k, n_i, n_h) \wedge D(c_j, m_k, n_h, n_z) \end{cases} \tag{24}$$

Note that following the definition above, self-delegation of a metric is not possible for a node, and indirect delegations are also considered, when the node n_i delegates to another node n_h, and n_h to n_z, with $h \neq z \neq i$.

In a CMDM digraph directed cycles between nodes can occur which make complex the computation of the metric delegation paths between nodes due to infinite loops could be followed. Therefore, for each metric of the control under consideration, in order to be able to evaluate the function $D(c_j, m_k, n_i, n_z)$ we apply Depth-First Path (DFP) search algorithm on the CMDM graph to extract the metric delegation paths between the nodes of the ACM.

By selecting as root node the component n_i which composed SLA we are evaluating, the DPF search visits exactly once each vertex in the CMDM digraph reachable from the root node, i.e. it starts at the selected root n_i and lists the visited nodes along each path until the path ends or the root node is reached back.

In the example of Figure 18, for node 0 the delegation path of metric m2 is nodes {0, 1, 2, 6, 7}, while for node 8 the delegation path of m2 is just node 8.

Definition 2: We define the **Component Composed SLA**, $SLA(c_j, n_i)$, as the Boolean variable that assumes value 1 if the service associated to node n_i declares the control c_j in its policy or SLA, 0 otherwise.

Let $I(c_j, m_k, n_i)$ be the Boolean function that takes value 1 if the service associated to node n_i implements the part of control c_j measured by metric m_k, 0 otherwise.

And let $DI(c_j, m_k, n_i)$ be the Boolean function which assumes value 1 when the node n_i delegates to another node n_z the implementation of the part of the security control c_j measured by a given metric m_k and the delegate node n_z implements it:

$$DI(c_j, m_k, n_i) = \exists n_z : D(c_j, m_k, n_i, n_z) \wedge I(c_j, m_k, n_z) \tag{25}$$

Then, for each node n_i of the CMDM (which is node n_i in the ACM too) we can build a SLA composition rule in the form of Equation (26), taking into account all the metric delegations in which the node is involved for each control.

$$SLA(c_j, n_i) = \wedge_1^K [I(c_j, m_k, n_i) \vee DI(c_j, m_k, n_i)] \quad (26)$$

Hence, Equation (26) represents the Component Composed SLA of node n_i for control c_j and interprets that the control can be declared in the Component Composed SLA of the node only when all metrics that evidence the correct implementation of the control are declared in the policies of the nodes (components) from which the metric implementations will be ultimately inherited, which could be either nodes owning the metric implementation or last nodes in the metric implementation delegation chain.

The Equation (26) can be specialised for security controls as:

$$SecSLA(sc_j, n_i) = \wedge_1^K [I(sc_j, m_k, n_i) \vee DI(sc_j, m_k, n_i)] \quad (27)$$

And, similarly, for privacy controls we would have:

$$PLA(pc_j, n_i) = \wedge_1^K [I(pc_j, m_k, n_i) \vee DI(sc_j, m_k, n_i)] \quad (28)$$

Thanks to the composition rules obtained in the Per-Component Composition evaluation we are able to identify the controls that each component is effectively able to grant to the others and the overall application. It is worth noticing that the effect of the above composition is that even controls that are not directly implemented and/or not applicable to a component, can now be considered.

The ACM distinguishes three types of nodes or components that can be found in multiCloud applications: software services (non-cloud services, SaaS or PaaS), IaaS (virtualization) and CSPs themselves. When deploying a software component on a Virtual Machine (VM) provided by a CSP, all the three component types may contribute to the implementation of the security and privacy controls. Still, some controls may not be applicable to a particular type of component due to the component nature. For instance, the security control PE-3 (Physical Access Control), which states that there is a specific access control for each access to the physical resources, is not applicable for a software component, while typically an IaaS service provider (CSP) implements it and grants it through its own Security SLA. Therefore, the metrics on the implementation of a control can only be measured on those component types in which the control has sense. The lack of meaning of a whole control or of some control metrics for a particular type of node in the ACM can be represented as metric delegation relationships, where all or part of the metrics are delegated to other nodes and only the metrics that are possible to be measured on the node could remain owned by the node.

3.5.3.5 Evaluation of the Application SLA

Thanks to the per-component SLA composition we are able to evaluate the SLA of the overall multiCloud application, which assess whether a specific control can be declared or not at application level, i.e. whether composite application providers could grant it in the SLA offered to their customers.

Definition 3: We define the **Application SLA**, $SLA(c_j, app)$, as the Boolean variable that assumes value 1 if the application declares the security or privacy control c_j in its policy or SLA, 0 otherwise.

According to the definition of the security controls, a composed system or application has a control declared in its Application SLA if and only if all the N services composing the application declare such control in their SLA as expressed by the logical conjunction in Equation (29) below.

$$SLA(c_j, app) = \wedge_1^N SLA(c_j, n_i) \quad (29)$$

Where $SLA(c_j, n_i)$ refers to the Component Composed SLAs defined in Definition 2 and obtained by using Equation (26), which equals 1 if the service associated to node n_i declares the control c_j in its policy or SLA, 0 otherwise.

The Equation (29) can be specialised for security controls in the SLA to create the Application Security SLA as:

$$SecSLA(sc_j, app) = \wedge_1^N SecSLA(sc_j, n_i) \quad (30)$$

And for privacy controls to create the Application Privacy Level Agreement as:

$$PLA(pc_j, app) = \wedge_1^N PLA(pc_j, n_i) \quad (31)$$

Considering the nature and scope of privacy controls described by NIST SP 800-53 Rev. 5 [119], the Equation (29) could be adapted to Equation (31). Please note that *joint* controls can be considered in either Equation (30) or Equation (31).

In the present work we argue that the composition of privacy controls (and joint controls) follows the same schema of Equation (29), where the controls under consideration would be privacy related controls declared in the Application PLA rather than security controls in the SecSLA.

The composition method reflects that for a control to be declarable in the Application SLA, not all the components need to implement all the metrics of the control but rather to conform to the CMDM specified by the system security experts. An example helps in getting the overall idea of the SLA composition method. Consider again the control PE-3 (Physical Access Control) discussed above. A multiCloud application that orchestrates many different components will declare such a control if and only if all the components declare it in their Component Composed SLA, which means that all the software pieces are deployed over physical resources that have the control correctly implemented. In case the application is made of two software pieces deployed on different VMs offered by different CSPs and only one of the CSPs grants the control, the application cannot declare the security control in its overall SLA. It is worth noticing that the single components, in their own SLA obtained from self-assessment, simply do not declare the control as it is not applicable to them.

It is possible to apply such criterion to all the controls, i.e. leaving to SLA composition the role of extending the controls to all components that can be affected, and simply verifying in the overall application that their metrics are correctly declared in the corresponding components.

3.5.3.6 Computation of SLO levels in the Application SLA

Security and privacy policies stipulate not only which are the set of controls guaranteed for a specific system or service but also the particular provision level that can be granted for each of the controls. Therefore, when composing SecSLAs or PLAs for composed applications, it is necessary to not only assess whether the control can be declared or not in the SLA of the composed application, but also to calculate the provision level or Service Level Objective (SLO) value that can be guaranteed. This will support the reasoning on quantitative security and privacy levels of the application and would aid in the negotiation of application prices with the customers.

Following the line of security ranges in [182], Casola et al. [183] proposed a security level quantification methodology based on the notion of Local Security Level (LSL) defined as a function that maps any type of security value (either qualitative or quantitative) to a user-defined quantitative security level. In our work we rely on LSL method to quantify the security level or privacy level of a specific implementation of a security control or privacy control, respectively. This method was selected because it allows for the calculation of the LSL of each control and therefore quantitative

declare the guarantees within the Component's Composed SLAs, and thus, within the Application SLA.

3.5.3.6.1 *SLO level measurement in a node of ACM on top of control metric levels*

As explained before, the level of a security capability implementation may be measured by monitoring different metrics or indicators on the control provision evidences. Therefore, whichever the delegation path for a metric and whichever the node of the ACM that finally implements the part of the control measured by the metric, the provision level of the metric contributes to the control level evaluation.

For instance, the control RA-5 "Vulnerability scanning" could be assessed by three metrics: "vulnerability scanning frequency", "vulnerability remediation ratio" and "vulnerability feed update frequency". The possible values of each of these metrics can be associated to a metric security scale in the form of LSL. E.g. the metrics "vulnerability scanning frequency" and "vulnerability feed update frequency" could both have provisions {none, weekly, daily, continuous} which could be associated to respective security levels in an ordered list LSL_{vsf} {0, 1, 2, 3} and LSL_{vfuf} {0, 1, 2, 3}. The metric "vulnerability remediation ratio" could adopt four provision scales {x<=20%, 20%<x<=30%, 30%<x<=50%, 50%<x<=70%, x>70%} which could be ordered by increasing security level in LSL_{vrr} {0, 1, 2, 3, 4}. It is assumed that an implementation metric which takes a particular level in the scale within LSL metric levels, is able to take also all the security levels below, which reflect less security.

The Global Security Level (GSL) in Casola et al. [183] aggregates in a unified scale multiple LSL scales to evaluate the overall security level of composed policies. This method could be adopted to compute the control security level SLO on the basis of the LSL of individual metrics for the control. The resulting GSL for the control would still be a local security level (LSL), as it only represents the level for a single control, not the whole security policy (or SLA). By setting the GSL scale levels of the controls to desired metric provision level combination, each level in the global scale would be represented by the vector of the values measured for the control metrics.

The specification of LSL scales for the controls in the SLA shall be made by security and privacy experts in a case by case basis. In all cases, all the metric levels that build the control levels need to be in incremental order. That is, it is not possible to set a control level vector that includes a metric level lower than the level the metric has in any precedent control level. This way we ensure that highest control levels correspond to highest values of the metrics considered for that control.

Following the example with three metrics above, the global scale for the control RA-5, $LSL(RA-5) = \{0, 1, 2, 3, 4\}$ (denoted as LSL and not GSL because it is a local security level scale for that control), could be decided to be {(LSL_{vsf} = 0, LSL_{vrr} = 0, LSL_{vfuf} = 0), (LSL_{vsf} = 1, LSL_{vrr} = 1, LSL_{vfuf} = 1), (LSL_{vsf} = 2, LSL_{vrr} = 2, LSL_{vfuf} = 2) , (LSL_{vsf} = 3, LSL_{vrr} = 3, LSL_{vfuf} = 2), (LSL_{vsf} = 3, LSL_{vrr} = 4, LSL_{vfuf} = 3)}. Figure 19 shows the LSL scale of the RA-5 control with respect to the example metrics described.

Control	Metric	LSL1	LSL2	LSL3	LSL4	LSL5
RA-5 "Vulnerability scanning"	vsf= vulnerability scanning frequency	0 (none)	1 (weekly)	2 (daily)	3 (continuous)	3 (continuous)
	vrr= vulnerability remediation ratio	0 (x<=20%)	1 (20%<x<=30%)	2 (30%<x<=50%)	3(50%<x<=70%)	4 (x>70%)
	vfuf= vulnerability feed update frequency	0 (none)	1 (weekly)	2 (daily)	3 (continuous)	3 (continuous)

Figure 19: Example control levels based on metrics' levels.

Next Figure 20 depicts an extended example with three controls which levels (LSLs) are computed on the basis of different metrics.

Control	Metric	LSL1	LSL2	LSL3	LSL4	LSL5
$control_1$	m_{11}	0	1	2	3	3
	m_{12}	0	1	2,3	4	5
	m_{13}	0	1, 2, 3,4	5,6	7	8
$control_2$	m_{21}	0	1	2	2	
	m_{22}	0	1	2,3	3	
	m_{23}	0	1,2	2	4	
	m_{24}	0	1,2,3,4	5	6	
$control_3$	m_{31}	0	1	2		
	m_{32}	0	1	2		
	m_{33}	0	1,2	3,4,5		

Figure 20: Example of multiple control levels based on metrics' levels.

The global levels are represented coloured and as we can see, while $control_1$ scale reaches level 5, $control_2$ scale stops in level 4 and $control_3$ scale in level 3. Furthermore, for instance, $control_1$ only reaches LSL level 3, i.e. LSL_3 (yellow) when both m_{11} and m_{12} measure at least a value of 2 and m_{13} reaches value 5. Any other lower values of these metrics would make $control_1$ stay in LSL_2 (blue). In case the metric has more possible values than the amount of levels in the control LSL scale, i.e. more than 5 values, the security expert needs to decide how to "split" the values among the control levels. For example, for the metric m_{13}, the values 1, 2, 3 and 4 are associated to LSL_2, and values 5 and 6 are associated to LSL_3 and LSL_4, respectively.

3.5.3.6.2 *Normalisation of SLO levels of all nodes in ACM*

For the evaluation of the level of each control (LSL) within the Application SLA to be possible, it is necessary to first homogenise the way each of the control implementations is valued in each of the constituent nodes in the ACM. Even if controls adhere to official definitions in the standard control family (e.g. the adopted NIST SP 800-53 Rev. 5 [119]), differences may exist in the way the controls are implemented by different service providers, and therefore, non-equivalent quantifications may apply for the same control part (measured by a metric) being implemented by different components of the composed application.

That is, control metrics could have different quantification scales in the ACM nodes which would lead to non-equivalent quantification scales for the controls as well. For example, provider CSP1 may quantify the RA-5 control metric "vulnerability scanning frequency", i.e. the frequency the tests are conducted on the component, as {never, daily, weekly, continuous}, while provider CSP2 may quantify it as {never, periodically, continuous}. Similarly, other metrics for RA-5 could have different scales in both providers, and therefore, the provision scales of control RA-5 may differ between two nodes offering the control in their Composed SLA.

Normalisation of control provision level quantification would require that prior to the composition, a relation $f(c_j, m_k, SP_i)$ is computed for each control metric and service provider as a mapping between a reference scale selected for the control metric provision quantification and the scale adopted by the provider so all providers' metric scales are normalised to the same scale and comparisons are possible.

Let the space of Local Security Levels defined by the user for a control c_j be $LSL(c_j) = \{l_1, l_2, l_3, \dots, l_z\}$ where z is the number of LSLs and 1/z the numerical value associated to each l_i. It is desired that the $LSL(c_j)$ is the reference scale for the normalisation of the control levels of all the providers in the ACM.

Let $P(c_{j,}\, m_k, SP_i) = \{v_1, v_2, v_3, \dots, v_m\}$ be the provision scale associated for a particular metric m_k of the control c_j by service provider SP_i which includes m provisions. According to the definition of LSL, each instance of the provision v_i will be assigned a numerical value in $LSL(c_j)$, as per the following transformation function proposed:

$$\bar{f}(v_i) = l_n \;\; \forall\, n = 1 \dots m\,, \forall\, i = 1 \dots z \quad if \;\; m = z$$

And if $(m > z)$ & $((m-2)/(z-2) \in \mathbb{N})$, then:

$$\bar{f}(v_i) = \begin{cases} l_1 \Leftrightarrow i = 1 \\ l_z \Leftrightarrow i = m \\ l_n \Leftrightarrow i = \left[2 + \left(\frac{m-2}{z-2}\right)(n-2)\right] .. \left[1 + \left(\frac{m-2}{z-2}\right)(n-1)\right] \\ \forall\, n = 2 \dots (z-1) \end{cases}$$

And if $(m > z)$& $((m-2)/(z-2) \notin \mathbb{N})$, then we need to use the whole part of $(m-2)/(z-2)$ denoted as $\lfloor (m-2)/(z-2) \rfloor$:

$$\bar{f}(v_i) = \begin{cases} l_1 \Leftrightarrow i = 1 \\ l_z \Leftrightarrow i = m \\ l_{z-1} \Leftrightarrow i = \left[2 + \left\lfloor\frac{m-2}{z-2}\right\rfloor(n-2)\right] .. (m-1) \\ l_n \Leftrightarrow i = \left[2 + \left\lfloor\frac{m-2}{z-2}\right\rfloor(n-2)\right] .. \left[1 + \left\lfloor\frac{m-2}{z-2}\right\rfloor(n-1)\right] \\ \forall\, n = 2 \dots (z-2) \end{cases}$$

In short, the transformation proposed assigns the minimum and maximum provision levels to the minimum and maximum LSLs respectively, while intermediate provision levels are assigned intermediate LSLs, depending on the sizes of the scales. Note that $\bar{f}(v_i)$ is not a function as the instance v_i may assume several discrete LSLs.

When $m < z$ the mapping would be made similarly but swapping v_i and l_n, as in this case there would be fewer possible provisions than defined LSLs.

In the example of CSP1 and CSP2 above, when the RA-5 control normalisation scale is selected to be $LSL(RA-5) = \{0, 1, 2, 3, 4\}$, for the RA-5 control metric "vulnerability scanning frequency", the five provisions of the metric {never, daily, weekly, monthly, continuous} by CSP1, which are associated to $\{0, 1, 2, 3, 4\}$ metric levels, would be mapped one by one to the five $LSL(\mathrm{RA}-5)$ control levels (m=z in $\bar{f}(v_i)$ above). For provider CSP2 instead, the provisions {never, periodically, continuous} of this metric may be quantified in $\{0, 1, 2\}$ scale, and hence, the mapping to the control levels $LSL(\mathrm{RA}-5)$ would fall in the case of m < z (3 < 5), and the provision "periodically", quantified as level 1 in the metric scale, would be mapped to levels 1, 2 and 3 in $LSL(\mathrm{RA}-5)$, while provision "never" would be mapped to 0 and "continuous" to 4, respectively. This way, it is possible to establish a common RA-5 control level scale for both CSP1 and CSP2 providers.

3.5.3.6.3 *SLO level evaluation in the Application SLA*

Once the quantification methods for each control are normalised, it is possible to compute for all the controls the control level (SLO) that can be guaranteed as follows. For every control c_j in the Component Composed $SLA(c_j, app)$, the level that it is possible to grant in the whole system is the minimum level granted in the set of nodes in the ACM that offer such control in their Component Composed SLA.

Since for every control c_j declared in the Application SLA exists a set of K nodes $\{n_1, n_2, n_3, \dots, n_K\}$, with K >= 1, that grant the control in their Component Composed SLA, each with provision level $LSL\left(c_j, n_k\right)$, then, $\forall\, c_j \in SLA(c_j, app)$ we have:

$$SLO(cj, app) = LSL(cj, app) = min\left(LSL\left(c_j, n_1\right), LSL\left(c_j, n_2\right), \dots, LSL\left(c_j, n_K\right)\right) \quad (32)$$

Once the composition rules are applied and the SLO levels calculated, the final Application SLA can be built as each of the controls that can be granted are known together with the maximum policy level that can be promised for each.

It is important to note that a direct conclusion from the SLA composition is the fact that it allows to identify critical nodes in each CMDM (and thus, in the ACM) which are the ones that actually offer a control with the minimum level LSL, and therefore these are the ones which SLOs should be maximised to raise the overall declaratory security posture of the system. Similarly, the nodes which get the highest number of control metric delegations devote greater attention, as their metrics will impact in more Component Composed SLAs. Vertices in the ends of metric dependency chain of the CMDM are also critical as delegating nodes do not inherit the control declaration unless the delegates offer corresponding metrics in their SLA. Therefore, according to SLA composition rules, a clever strategy to maximise the grantable security and privacy levels of the overall application would be to maximise the amount of control metrics that are offered by the nodes with greater structural importance in the CMDM, i.e. those receiving more metric delegations in overall, i.e. considering all the CMDMs of all the controls, and to raise the control LSL in those nodes which offer the minimum LSL in their Component Composed SLAs, by raising the values of the metrics declared by them.

3.6 Conclusion

In this section we summarise the major advances over state of the art achieved by the contributions of this Thesis.

The proposed **integrated DevOps methodology for multiCloud applications supporting security and privacy** is the first holistic methodology that seamlessly integrates key security and privacy activities in multiCloud system life-cycle, from security and privacy requirements capturing to continuous assurance at operation. The methodology proposes a circular process workflow where early feedback from continuous monitoring at Operation phase enables the reassessment of system risks and evaluation of the need of updates in system design models or activation of enforcement agents to improve the system protections. The rich enhanced expressiveness of the security and privacy modelling language facilitates the automatic deployment of system components together with security and privacy mechanisms. The powerful risk assessment and sensitivity analysis achieves informed decisions about CSP selection to deploy system components and about most convenient security strategies to adopt. The generation of composite application SLA makes it possible a structured and metric-oriented monitoring of the multiCloud application at runtime.

It is interesting to note that the security and privacy concerns in our integrated DevOps methodology are captured in multiple models at different life-cycle steps which all align among them: (i) the security and privacy DSL-based model that captures initial requirements, (ii) the system Attack Defence Tree (ADT) for risk analysis and decision on Cloud Service Providers, (iii) the Application Composition Model and Control Metric Delegation Models used in the SLA composition method, and (iv) the Application Security and Privacy SLA resulting from the composition which express the guarantees of the overall application and allow to control the application behaviour at runtime.

MultiCloud applications redesigns (remodelling of the extended-CAMEL models) will be driven by actual measurements taken about the Service Level Objectives and metrics stated in the composed Application Security SLA.

In addition, a key benefit of the integrated DevOps methodology is the fact that in all of the models the taxonomy adopted for the security and privacy controls specification is the same: the standard NIST SP 800-53 Rev. 5 [119]. This contributes significantly to the transparency, auditability and reuse of these models, and therefore of the resulting multiCloud application itself.

The proposed **security and privacy requirements modelling language** (or security and privacy Domain Specific Language (DSL) for multiCloud systems) is based on the most advanced language for multiCloud, the CAMEL language and extends it to bring contributions in three decisive aspects of security considerations in multiCloud: i) security and privacy behaviour specification (primarily through enabling the expression of both required and offered security and privacy mechanisms by the components), ii) enhanced secure distributed deployment, and iii) the declaration in the model of security and privacy agents that need to be part of application deployment to enable self-healing capabilities of the system at runtime.

The new method for **continuous quantitative risk management and defence optimisation in multiCloud** of this Thesis significantly advances in the state-of-the-art solutions for systematic and continuous risk management in multiCloud applications which pose a significant challenge to system engineering and security threat control due to the fact that they orchestrate multiple components of diverse nature which may even be provisioned by different external service providers.

The work proposes an effective solution for the continuous quantitative evaluation of composite system risks on top of Attack Defence Trees (ADT) capturing the attack-defence scenarios including all the system components, both internal and outsourced ones. The risk management is performed thanks to the early feedback from operations to development as impelled by the DevOps paradigm, which allows to iteratively revisit initially assessed risks to consider the effects of system components' deployment and the actual situation of attacks and defences in the system operation.

The innovations brought in this area can be summarised as follows:

i. A comprehensive methodology for continuous risk management in multiCloud systems based on the use of ADTs and continuous feedback from Operations to Development following the DevOps approach. The methodology addresses the needs of multiCloud applications where it is crucial to take into account the impact of outsourced services or components in the system risk assessment.

ii. A risk assessment algorithm for ADTs to propagate the risk attributes from the attack event nodes and defence nodes up to the root of the tree based on a smart adversary and smart defender strategies respectively, which would have opposite goals with respect to system risks: while the smart adversary would seek to cause the maximum risks to the system, the smart defender would try to minimise the risks.

iii. As part of the risk management process a defence optimisation method is proposed which considers not only the attack coverage and costs of deployed countermeasures, but also other variables such as their efficiency in risk and impact mitigation. This way, the method allows to find out the optimum countermeasure set that offers required attack coverage while simultaneously optimises other parameters such as security cost, minimised risk, or reduced attack impact.

iv. The methodology enables the re-evaluation of system risks after providers of outsourced components are selected and their defences known, as well as continuous updates of assessed risks when all system components are deployed and running.

Finally, the new methodology for **Security and Privacy Service Level Agreement composition** builds upon and extends previous research on secSLA composition for multi-Cloud systems by Rak [129] and brings the following contributions:

i. The extended SLA composition methodology presented enables the evaluation of the controls that can be guaranteed in the multiCloud Application SLA by considering the control metric delegation relationships between the components.

ii. The methodology includes the technique for the computation of the SLO levels that can be guaranteed for the controls in the composed Application SLA based on the SLO levels that individual components grant.

iii. The methodology applies to both secSLAs and PLAs as the controls considered are standard controls addressing technical means of both aspects. The composition can be performed over secSLAs containing security controls only, PLAs containing privacy controls only or SLAs containing both.

iv. The methodology addresses not only multiCloud systems but also IoT systems and combinations of both, i.e. Cloud-based IoT systems. Therefore, the methodology can be used in different architectural scenarios where the system or application components can use or can be deployed in multiple types of services and providers, be they federated or independent.

In view of all the advances above, we can conclude that the methods and techniques proposed in this Thesis significantly advance the state of the art in multiCloud security and privacy assurance, and contributes to raising trust in multiCloud environments, which will push the Cloud Computing technology adoption as a fundamental enabler of the digitalisation of industries and business in Europe.

4 Solution validation

4.1 Introduction

The technical feasibility of the integrated DevOps solution for the assurance of security and privacy in multiCloud applications was validated through piloting the solution in a realistic industrial application case study which is a good representative of highly relevant services for the European economy. The multiCloud application selected is a telehealth system in the eHealth domain with high security and privacy constraints.

It is important to note that all the contributions of this Thesis were validated upon the same eHealth application so as to achieve a better understanding of how the overall DevOps methodology and its integrating parts aided in tackling with security and privacy aspects in each of the steps of the application creation process.

In the following subsections we describe the validation carried out for each of the parts or contributions of the solution, following the order in Section 3.1. First, we describe the validation objectives and then the case study methodology followed, where the eHealth multiCloud system used to evaluate the research hypothesis is detailed.

4.2 Validation objectives

The main objective of the solution evaluation is to assess if the solution proposed is being correctly developed towards fulfilling the objectives described in Section 1, that is:

- Prove the feasibility of proposed security-by-design and privacy-by-design mechanisms, i.e. the three methodologies in the Development phase of the DevOps methodology proposed (the three contributions 2, 3 and 4 of this Thesis as described in Section 3.1), to enable the assurance of security and privacy features of multiCloud applications.
- Demonstrate that it is possible to analyse and quantitatively assess risks in multiCloud applications and perform a risk-based decision on best Cloud services matching application security and privacy requirements and reducing risk.
- Demonstrate that the Service Level Agreements (SLAs) for the overall application can be obtained, which formalise security and privacy levels that can be continuously monitored to ensure that they hold during Operation phase.

This evaluation was conducted by testing the usage of the proposed solution to develop the eHealth application of the case study (i.e. use case) described in Section 4.3.

In particular, the validation scenario aims to demonstrate how the DevOps Team of an application that combines distributed Cloud and IoT resources can apply our integrated DevOps methodology to carry out the following required objectives:

- Capture in form of a Cloud Service Independent Model (CPIM) the security and privacy requirements of the application such as access control (AC family controls) and Vulnerability scanning (RA family controls) capabilities required by the components or offered by external enforcement agents to use by the components.
- Transformability of the created CPIM into a CSPM in form of deployment model which enables the automation of distributed deployment in multiCloud.

- Evaluate and continuously re-evaluate the risks of the overall system and the individual components while supporting the DevOps Team in deciding which protections or controls to use in the system and which to require from third-party Cloud and IoT providers of outsourced components.
- Select the external Cloud and IoT providers to be used by the application according to the desired risk minimisation level as well as other constraints such as the costs of the defences offered by these providers.
- Create the Composed Application SLA to be offered to application consumers on top of the SLAs of individual components and providers used, and which can be continuously assessed to ensure the security and privacy level promised to them holds, and which can be used to rise alarms in case violations occur or are about to occur.

4.3 Validation methodology: eHealth multiCloud application case study

In this section the methodology followed for the validation of the solution is described.

The proposed methodology was tested in a set of real-world systems orchestrating Cloud and IoT services and infrastructures, from which we have selected the most illustrative one to demonstrate the approach, as it required not only security but also privacy aspects to be considered in the system development process.

The case study analysed is a real-world eHealth scenario that combines multiple distributed components as shown in the simplified architecture of Figure 21.

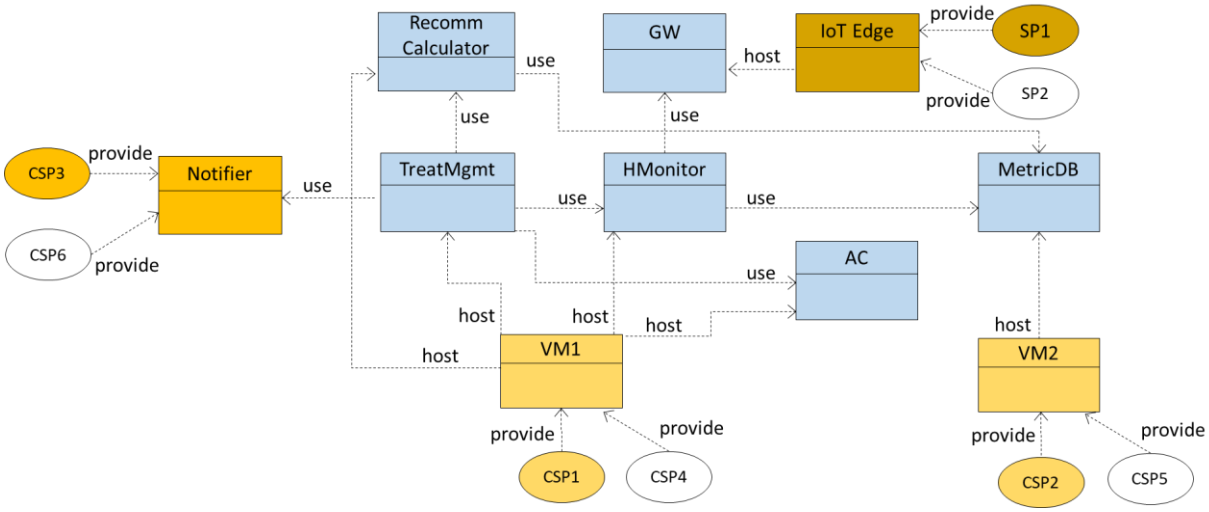


Figure 21: Simplified architecture of the eHealth multiCloud application under study.

In Figure 21, the components coloured in blue represent internal components while the ones in orange colour represent services outsourced from external Cloud Service Providers (CSPs) and services deployed in IoT resources acquired from external Service Providers (SPs).

The internal components eHealth system under test include:

- Treatment Management (in short TreatMgmt or TM) as the main Web service responsible for main business process of the system and the orchestration of other components.
- Health Monitoring (in short HMonitor or HM) system to periodically sense the health indicators of the patient coming from the IoT Gateway (GW) which will be stored in the measurement database named MetricDB (DB).

- MetricDB (DB) is a database which stores patient health data which are sensitive data to protect.
- IoT Gateway (GW) software component which is deployed in an IoT Edge device.
- IoT Edge device which is connected to the health sensors that the patient uses at home and which measure the health data.
- Recommendation Calculator (in short RecommCalculator or RC) which calculates the optimum treatment recommendation for the patient.
- Access Control (AC) used by the system for the authentication and authorisation of the users.

And the external Cloud component of the application is the:

- Notification component (in short Notifier or Ntf) which is an external Software-as-a-service used to inform the patients on the recommended action for the treatment.

As outlined in the architecture, two Virtual Machines (VMs), i.e. VM1 and VM2, were finally used to deploy the internal system components, provided by CSP1 and CSP2 Cloud providers respectively. The IoT Edge device was acquired from service provider SP1. The external service providers that finally were selected for the composition are shown coloured whereas white coloured ones correspond to other alternative providers considered.

This example eHealth application serve to conduct the validation of the security and privacy assurance solution proposed, which is composed of four main contributions described in Section 3. Therefore, the description of the validation process and results has also been structured following the same order of the contributions in Section 3, where each of the next subsections matches with one of the contributions.

For the validation of each of the activities and associated models proposed in the solution a dedicated software prototype tool was used. In some cases, the software prototype was created from scratch and in other cases background tools were improved or extended to be able to demonstrate and fully validate the proposed solution capabilities. In the following subsections, details of the tools and extensions produced are offered.

4.4 Integrated DevOps methodology supporting security and privacy

The software engineering team that created the multiCloud application of the case study was already experienced in agile and DevOps methodologies. The DevOps Team was built by gathering together members of the company that had the responsibility of the eHealth application as software architects, developers, analysist and business managers or decision makers on the product. Each of the roles took the responsibilities described in Table 2.

To build the eHealth application under study, the DevOps Team followed all the steps of the DevOps process workflow proposed in Figure 4.

The validation of the DevOps methodology involved the validation of the three first steps of the overall process which are explained in subsequent sections. These steps correspond to Development phase of the DevOps methodology:

1. **Requirements modelling**: The Cloud Provider Independent Model (CPIM) of the eHealth system in Figure 21 was created and validated as described in Section 4.5.
2. **Risk Management and Risk-driven Cloud Services selection**: this validation involved the creation of the system ADT for the eHealth case study application and the verification

of the feasibility and effectiveness for the eHealth application of the quantitative risk analysis and continuous refinement of risks. The validation also demonstrated the feasibility and utility of the defence optimisation techniques proposed. The full validation details are given in Section 4.6

3. **Composed Application SLA generation** included the creation and validation of Application Security SLA and PLA as described in Section 4.7.

Even if the mechanisms supporting the Operation phase of the application are not part of this Thesis, the resulting models from the Development activities were used in the Operation phase as described in Section 3.2.4.

The DevOps Team used the CPIM created in the first step of the Development and the Composed Application SLA generated in the last step to be able execute the **Deployment** step (see Figure 4) respecting the required provision of Cloud virtualisation resources, VM1 and VM2 in CSP1 and CSP2 respectively, followed by the distribution of components over these VMs as per the plan of Figure 21.

The composed Application SLA obtained as a final outcome of the SLA Composition method served as input to the **Composed Application SLA Monitoring** and **Enforcement** steps in the workflow of Figure 4 and was used to perform a continuous tracking on the fulfilment of the guarantees specified therein.

4.4.1 Solution software prototypes

For the validation of the methods proposed in the Development phase of multiCloud applications different tools were used that are described later in the corresponding sections explaining the validation of each of the methods.

In order to validate the utility of the models developed at Development phase, two software tools were utilised to demonstrate the seamless continuum of the DevOps methodology with Operation phase.

The collection of the software prototype tools used for the case study in the steps of the DevOps workflow in Figure 4 is this:

1. **Requirements modelling**: MUSA Modeller from MUSA project [47] enabled to build the CPIM of the multiCloud application including security and privacy reequipments. See Section 4.5.1.
2. **Risk Management and Risk-driven Cloud Services selection**: The ADTool software tool in [184] and [185] was largely extended in an ADToolRisk version [186] of the tool to be able to create, reason and evaluate the risks on the system ADT model. The tool is able to treat risk as derived complex attribute associated to the tree nodes. Completely new risk scenario simulation features were also added in the ADToolRisk. For the optimisation of defences technique, a new ADTMind software tool [187] was also created from scratch. See Section 4.6.6 for more details on both tools.
3. **Composed Application SLA generation** the SLA Generator tool from MUSA project [47] supported the creation of the ACM model for the application using the CPIM as input. The tool is described in Section 4.7.6.
4. **Deployment:** The MUSA Deployer from MUSA project [47] was used to enact the CPIM created in Requirements Modelling step and transform it into an executable deployment plan which the tool engine executes. The description of how this process is achieved is included in a publication resulting from this Thesis work that explains the creation of the

CPIM and the details of how it is exploited in Deployment. Please find MUSA Deployer publication within *Security and privacy requirements modelling language for multiCloud* category in Section 5.2. The tool is an open source multiCloud deployer able to acquire the Cloud infrastructure, configure it, deploy the application software artefacts and configure them. The current version is compatible with Eucalyptus, Openstack and Amazon cloud environments.

5. **Monitoring of Composed Application SLA:** The MUSA Security Assurance Platform in [157] from MUSA project [47] served to perform the continuous monitoring of the Application Security SLA and PLA created. This is a powerful Security Information and Event Management (SIEM) tool that offers comprehensive situational awareness of the status of the multiCloud system by measuring metrics at network, operating system and application layers. The metrics measured are those extracted from the application and components SecSLA and PLAs that the tool is able to interpret. The solution integrates monitoring, notification, enforcement and reaction capabilities. This way, the tool enables to control the security behaviour of the application components deployed over a single or multiple Clouds.
6. **Enforcement of Composed Application SLA:** The MUSA Security Assurance Platform [157] was used to enforce the access control protections in the application.

4.4.2 Conclusion

The validation of the DevOps methodology for security and privacy assurance in the eHealth application of the case study was mainly focused on the Development part of the process, as the contributions of these Thesis are oriented to address this phase. Nevertheless, the models resulting from the development methods proposed (CPIM, ADT, SeSLA and PLA) showed to be usable in Operation phase to drive the deployment itself and to guide the assurance of security and privacy when application components are deployed and running.

As it will be shown in the following validation sections, all the methods suggested by this Thesis have proved to be efficient in the consecution of their objectives.

Therefore, as overall result, we can conclude that the integrated DevOps methodology has proved to serve in contributing to security and privacy assurance in multiCloud since it helped to:

- the reasoning and formalisation of security and privacy aspects,
- the identification of the best security strategies to adopt to protect internal and external system components so as system risks are minimised,
- the identification of the protections (controls) to request to Cloud Service Providers to guide their selection pursuant to risk minimisation strategies,
- the identification of the protections (controls) that can be declared in the SLA of the application, i.e. identification of the declarable security posture for transparency with application customers.
- the seamless integration between security-by-design and privacy-by-design methods with Operation phase.

4.5 Security and privacy requirements modelling

As the initial step of the Development phase, the DevOps Team of the eHealth created the Cloud Provider Independent Model (CPIM) of the multi-cloud application using the security and privacy

DSL proposed in this Thesis in Section 3.3. The created CPIM is to be used as input to the SLA composition step and the multiCloud application deployment processes.

The CPIM model created is a formal specification of the architecture shown in Figure 21, which includes not only deployment requirements but also security and privacy requirements enabled by the new meta-model.

Due to readability purposes, we do not include in the present section the complete CPIM model of the case study, which can be consulted in Appendix B.

Specifically, in this case study the new language enabled to capture in the CPIM model the multiCloud application component specification together with an Enforcement Agent for a access control that could optionally be used by the multiCloud application as a protection mechanism to execute security policies in granting access to Treatment Management component.

4.5.1 Solution software prototype: MUSA Modeller

In order to support the evaluation of the extended-CAMEL modelling language proposed in this Thesis, the MUSA modelling tool was developed within the context of the MUSA project [47], the so-called MUSA Modeller described below.

The tool allowed developers creating and storing multiCloud application models based on the modelling language proposed in this Thesis.

The MUSA tool prototype leverages upon the PaaSage CAMEL modelling tool and combines Eclipse Modelling Framework (EMF) [188], Object Constraint Language (OCL) [189] and Xtext [190] technologies. The current version of the tool is realized in terms of an Ecore based meta-model organised into packages that include cross references of native CAMEL DSLs.

The tool enables to specify models through the extended-CAMEL Textual Editor as well as to programmatically manipulate and persist models in a Connected Data Objects (CDO) repository. multi-cloud application models can be edited and updated remotely by end-users while they are stored in shared repositories. Furthermore, it allows users to not only specify extended-CAMEL models but also to syntactically and semantically validate them.

At high level, the MUSA Modeller is structured as follows:

- A *Web component*, the GUI from which the end-users access the MUSA Modeller tool and exploit all the internal web-services offered. It also includes the web Xtext libraries that offer syntactic and sematic services for remote management of syntax validation and auto completion.
- A *Server component*, which is the core of the MUSA Modeller since it implements all the business functionality and offers a series of web services that are consumed by the Web component.
- A *Database component*, which manages all the models and data stored in the central database. It uses the Hibernate framework [191].

The Server component of the MUSA Modeller offers a series of REST API interfaces that support the following functionality:

- Creation of multiCloud application models in the new extended-CAMEL language. End-users can instantiate previously defined templates for particular component types or applications.
- Definition and storage of multiCloud application component templates for reusing CAMEL models of the components.

- Edition and storage of multiCloud application models, where application models can be defined by multiple end-users.
- Model checking for syntax and semantic correctness and integrity of the created models. The tool provides messages of warnings and errors whenever a non-conformity is identified in the model.
- Selection of Security Controls previously identified and stored in external referenced catalogues.
- Selection of Enforcement Agents previously identified and stored in external referenced catalogues.

The following figure shows the look and feel of the MUSA Modeller GUI.

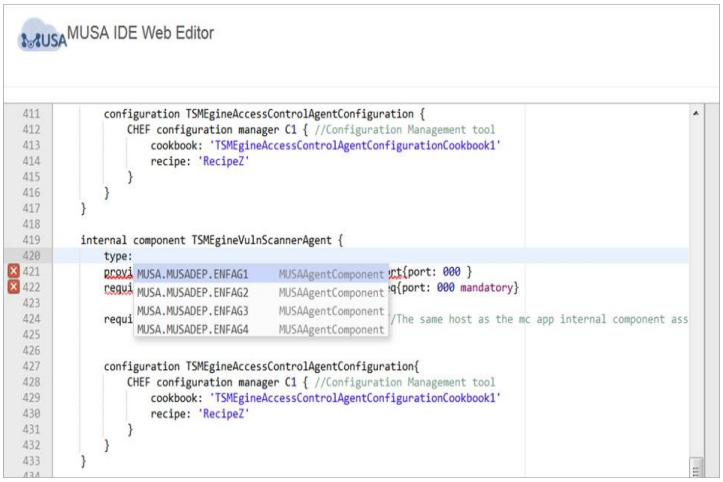


Figure 22: MUSA Modeller support for Security Agent selection.

4.5.2 Conclusion

The main innovations achieved in our CAMEL language for security- and privacy-aware multiCloud application modelling reside in the improved expressiveness of security aspects of multiCloud applications that will be relevant for the SLA composition afterwards. Moreover, we also provided extensions for enhancing the expressiveness of the deployment planning itself. The deployment extensions include concepts required, for example, in those situations when a Configuration Management tool will be used for deployment execution and when application components will not be deployed on top of Virtual Machines but in containers.

More concretely, the following features of the language resulted valuable for the eHealth application of the use case:

- Modelling of component nature - Required by SLA Composition step aligning the security-by-design and privacy-by-design approach.
- Modelling of Security controls provided by components properly supporting Security control families - Required by the SLA Composition step.
- Modelling of nature of the IP address - Required in virtual machines provisioning phase.
- Modelling of components deployment order - Required by Deployment phase.
- Modelling of data exchange protocols - Required by Deployment phase.

- Modelling of dynamic configurations of communications between components - Required in deployments where components may have dynamic configurations.
- Modelling of deployment handler - Required in deployments managed by deployment handlers.
- Modelling of PaaS layer elements - Required in deployments that use clusters of containers.

The added value of the new modelling language has been proved in the evaluation performed in the case study. The usefulness and feasibility of the approach was confirmed not only in the usage of the Security DSL language to model the deployment and security requirements identified in the case study, but more significantly in its integration with the rest of the approaches in this Thesis, i.e. in the DevOps methodological framework, particularly in collaboration with SLA Composition step to obtain the ACM model of the system and further with Deployment step to obtain the deployment model.

4.6 Continuous Risk Management and Risk-based Optimisation of Defences

4.6.1 Modelling of ADTs

The modelling of the ADTs for the case study started with drawing the tree structure graphs representing all the possible attack scenarios for the eHealth application. The attack goals and sub-goals were refined down to elementary attack events exploiting particular vulnerabilities of the assets.

The attacks and defences studied considered all system components depicted in Figure 6, including both internal components, i.e. RecommCalculator, IoT Gateway, TreatMgmt, HMonitor, AC and MetricDB, and external components, which involved three Cloud services Notifier, VM1, VM2, and the infrastructure IoT Edge. A security self-assessment was performed on internal components which enabled to ascertain the defence mechanisms they offered against potential attacks. The external components' defences were studied with the help of candidate Cloud Service Providers' SLAs (for Notifier, VM1 and VM2) and the user manual of the IoT Edge device gave the information on protections implemented therein.

The taxonomy used for the attack events was adopted from the MUSA Security Metric Catalogue [160] while the countermeasures in the tree were named after controls from the NIST SP 800-53 Rev. 5 [119] standard which collects 912 fine grained security and privacy controls. It is important to note that for some defences even if the same control name was used, the actual mechanism implementing the control was not exactly the same as the implementation depends on the nature of the asset. For this reason, the defence "RA-5 Vulnerability scanning" (from NIST) was refined to "RA-5 Vulnerability scanning-1" for IoT Gateway component, to "RA-5 Vulnerability scanning-2" for Database and to "RA-5 Vulnerability scanning-3" for virtual machines.

Finally, all the created ADTs were merged into a unified system ADT by considering the relationships among them and eliminating existing overlaps. Due to space limitations and in order to ease the risk management procedure demonstration, the explanation will be based on one of the ADTs created shown in Figure 23 and named "Steal health data", which is an appropriate representative of the ADTs created and could be considered for the example as the system ADT. In fact, this ADT is composed of three disjunctive ADTs, "Steal in origin", "Steal in transit GW->HMonitor" and "Steal in storage" joined by an OR relationship which indicates three potential independent means to steal the patients' health data by exploiting system vulnerabilities either when data is captured, transmitted between components or stored in the database. Note that for

simplification of the example the branch corresponding to the sub-goal of stealing the data in transmission between HMonitor and DB components named “Steal in transit HMonitor->DB” has been removed as well as the “Steal in sensor” branch from “Steal in origin”.

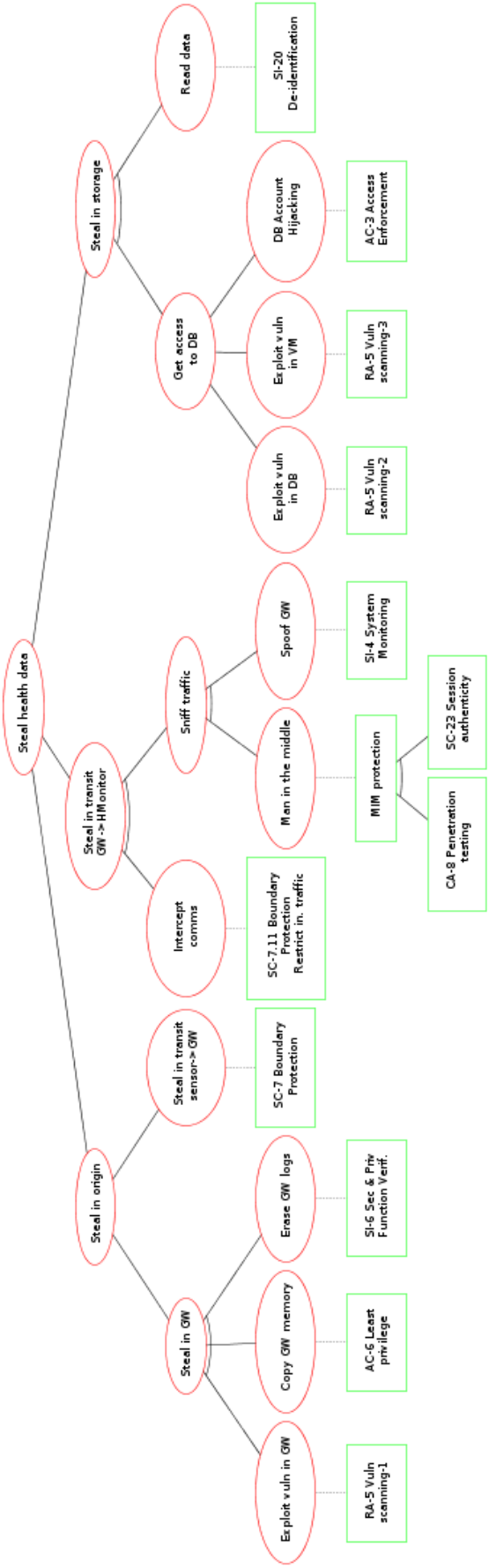


Figure 23: Extract of Steal health data ADT of the use case.

The modelling of the ADTs and evaluation of the risk attributes in the ADT nodes was facilitated by the risk assessment tool framework developed to this aim and explained in Section 4.6.6. The decoration of the ADT models assigned probability, impact and normalised cost values to all the attack leaf-nodes in the tree and the identified countermeasures in Figure 23.

Table 7 below collects the initially estimated values and derived risks in the elementary attack nodes according to Equation (2). The table includes in the last column the target asset for each of the attack events.

Table 7: Estimated risk vector values for attack events in Steal health data ADT

Attack event id	ADT node name	Probability	Impact	Cost	Risk	Asset
At1	Exploit Vuln in GW	0.40	4.00	6.00	0.27	GW
At2	Copy GW Memory	0.30	9.00	6.00	0.45	GW
At3	Erase GW logs	0.25	5.00	2.00	0.63	GW
At4	Steal in transit Sensor -> GW	0.10	6.00	3.00	0.20	GW
At5	Intercept comms	0.25	7.00	5.00	0.35	HMonitor
At6	Man in the middle	0.63	8.00	4.92	1.02	HMonitor
At7	Spoof GW	0.15	4.50	4.00	0.17	GW
At8	Exploit Vuln in DB	0.45	7.00	3.00	1.05	DB
At9	Exploit Vuln in VM	0.90	7.50	4.00	1.69	VM2
At10	DB Account Hijacking	0.80	6.50	7.50	0.69	DB
At11	Read Data	0.60	8.00	4.00	1.20	DB

Similarly, Table 8 shows the estimated risk vector attribute values for defences counteracting the attack events in the ADT of Figure 23. Please note that the derived risk attribute for the defences represents the cost-effectiveness in risk reduction of the control.

Table 8: Estimated risk vector values for defences in Steal health data ADT

Defence id	ADT node name	Probability	Impact	Cost	Risk (cost-effectiveness)
D1	RA-5_Vuln_scanning-1	0.60	6.00	7.00	0.51
D2	AC-6_Least_privilege	0.40	4.00	3.00	0.53
D3	SI-6_Security_and_Privacy _Function_Verification	0.50	5.00	5.00	0.50
D4	SC-7_Boundary_Protection	0.30	7.00	6.00	0.35
D5	SC-7.11_Boundary_Protection _Restrict_incoming_traffic	0.10	4.50	5.00	0.09
D6	CA-8_Penetration_testing	0.25	4.00	5.00	0.20
D7	SC-23_Session_Authenticity	0.15	6.00	4.00	0.23
D8	SI-4_System_Monitoring	0.50	5.00	3.00	0.83
D9	RA-5_Vuln_scanning-2	0.50	4.00	5.00	0.40
D10	RA-5_Vuln_scanning-3	0.60	6.00	2.00	1.80
D11	AC-3_Access_Enforcement	0.80	7.00	8.00	0.70
D12	SI-20_De-identification	0.80	7.00	7.00	0.80

4.6.2 Risk assessment over ADTs

4.6.2.1 Attack event risk assessment

Just after the decoration of all risk attributes for the leaf-nodes in the tree and their countermeasure nodes, the value for the derived risk attribute was automatically obtained and visualised in the extended ADTool by computing risk as in Equation (2). This value is represented in Figure 24 as the fourth value in all the yellow nodes. As soon as the values were set for the countermeasures, the risk vector values in the associated attack events were updated to reflect the effect of the defence risk vector values. The updated and initial risk vectors for countered attack events are shown in Figure 24 in the row just below the name of the event and in the row beneath, respectively.

Table 9 shows the resulting updated attribute values for attack events. Comparing with initial values in Table 7, note that defences do not have any impact on the attack costs, while probability, impact and risk of the countered attacks have all been reduced.

Table 9: Risk vector values for attack events in Steal health data ADT after defence decoration

Attack event id	ADT node name	Probability	Impact	Cost	Risk
At1	Exploit Vuln in GW	0.16	2.40	6.00	0.06
At2	Copy GW Memory	0.18	3.60	6.00	0.11
At3	Erase GW logs	0.12	2.50	2.00	0.16
At4	Steal in transit Sensor -> GW	0.07	4.20	3.00	0.10
At5	Intercept comms	0.23	3.15	5.00	0.14
At6	Man in the middle	0.6	6.08	4.92	0.75
At7	Spoof GW	0.07	2.25	4.00	0.04
At8	Exploit Vuln in DB	0.23	2.80	3.00	0.21
At9	Exploit Vuln in VM	0.36	4.50	4.00	0.41
At10	DB Account Hijacking	0.16	4.55	7.50	0.10
At11	Read Data	0.12	7.20	4.00	0.22

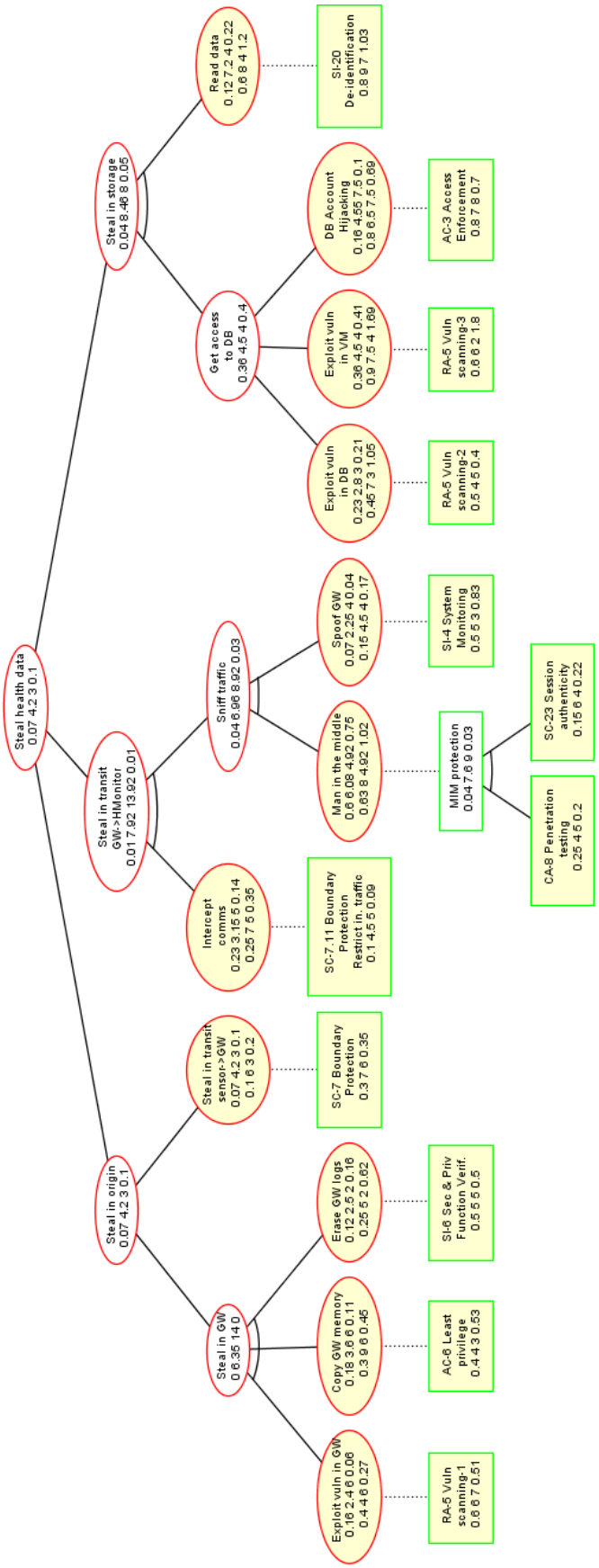


Figure 24: Case study ADT with risk vector evaluated in all the nodes.

Figure 25 depicts attack events' risks within severity quadrants, where blue points represent the initial risk vector points from Table 7 and orange points show the updated risk vectors from Table 9 after countermeasures' attributes were decorated with values in Table 8 .

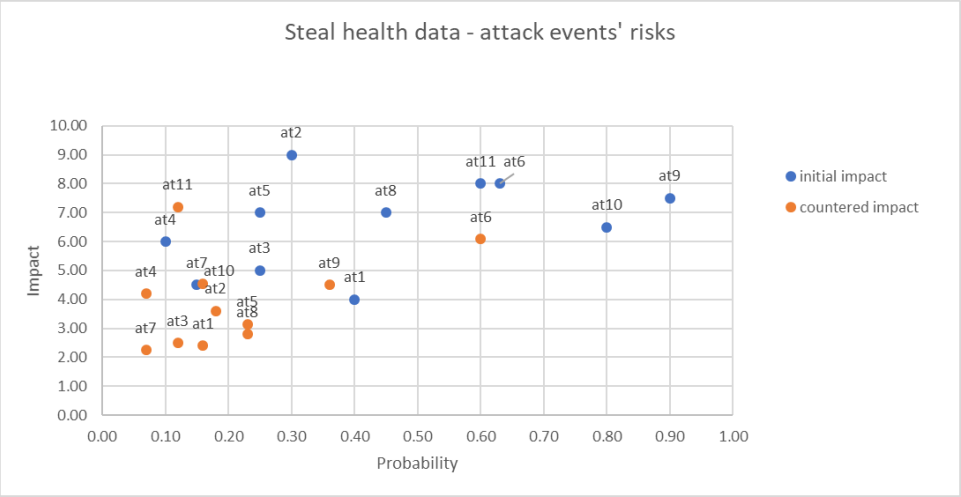


Figure 25: Severity of attack events before and after countermeasures.

By using the severity quadrants limits from OWASP Risk Rating Methodology [98], all the attack events with high probabilities from 0.6 to 1 and high impacts from 6 to 10 fall within the critical quadrant. As shown in Figure 25, initially these attacks were {At6_Intercept_comms, At9_Exploit_Vuln_in_DB, At10_Exploit_Vuln_in_VM, At11_DB_Account_Hijacking}, and the addition of countermeasures achieved that only at6 remained in the critical quadrant. Therefore, according to the metric definitions of Table 5, the metrics obtained for the risk vectors in critical quadrant are those shown in Table 10.

Table 10: Risk metrics for attack events in Steal health data ADT

Risk metric name	Value before countermeasure estimation	Value after countermeasure estimation
Threat density in Critical quadrant	4	1
Point of maximum risk in Critical quadrant	At9 (0.90, 7.50)	At6 (0.6, 6.08)
Point of minimum risk in Critical quadrant	At10 (0.80, 6.50)	At6 (0.6, 6.08)
Risk centre of mass in Critical quadrant	(0.73, 7.50)	(0.6, 6.08)
Maximum Risk in Critical quadrant	1.69	0.75
Minimum Risk in Critical quadrant	0.69	0.75

4.6.2.2 System risk assessment

The extensions to ADTool enabled the propagation of the risk vectors in Table 7 and Table 8 from bottom nodes up to the root node in the ADT following the rules in Table 4. This way the probability of success, impact to the system and overall attack cost for the "Steal health data" main goal were deduced. As shown in Figure 25, the resulting risk vector in the ADT root node was {0.07, 4.2, 3, 0.1}, where the value risk 0.1 gives a measure of the system risk exposure after the effectiveness of all defences was estimated.

According to Equation (15), the comparison of this vector with the risk vector {0.54, 9.5, 8, 0.64} evaluated in the root node without countermeasures (when all countermeasures probability was set to 0) provides the risk reduction achieved:

$$ROD_{All_{D_j}} = Risk_minimised_{sys_All_{D_j}} = 0.64 - 0.1 = 0.54$$

A reduction of 0.54 points in risk means that, by applying all the defences modelled in the system ADT, the 84.37% of risk minimisation can be achieved with respect to not implementing any.

4.6.2.2.1 System risk sensitivity analysis

Thanks to the simulation capabilities deployed in the ADToolRisk modeller, risk sensitivity analyses were performed on diverse weighting factors including attack attributes such as probability of success, impact to system or attack perpetration cost; and defence attributes such as probability, minimized attack impact, and cost. The algorithm used in these simulations was the one in Figure 13.

Sensitivity to attack attributes:

From the results of the initial risk assessment performed, the attack At6_Intercept_comms seems to be the most relevant for the system. The risk sensitivity analysis with respect to this attack attributes allows to check this statement.

By leaving the estimated values for the defences unchanged, Figure 26 left shows the results of ADT root node risk vector values when increasing the probabilities of At6 in 20 steps. As it can be seen, with all defences applied, At6 node probability variation has no impact on ADT root node which risk vector always adopts the values {0.07, 4.2, 3, 0.1}. Similarly, Figure 26 right shows null risk sensitivity of ADT root with variations of At6 impact from 0 to 10. A similar situation is obtained when changing At6 cost values from 0 to 10. This is the consequence of the application of the algorithm rules in Table 4, which makes the root node adopt the risk vector of the branch of the node named "Steal in origin" as it is always more risky than the one of "Steal in transit" node, regardless of the values of At6 risk vector.

Therefore, the analysis throws light on the fact that even if the At6 node risk is critical, irrespective of whether the At6 is countered or not, the overall risk of the ADT does not depend on the risk attributes of At6 attack. Therefore, it is not worthy to spend security budget on defending from that attack solely as other defence strategies will be more efficient as demonstrated below.

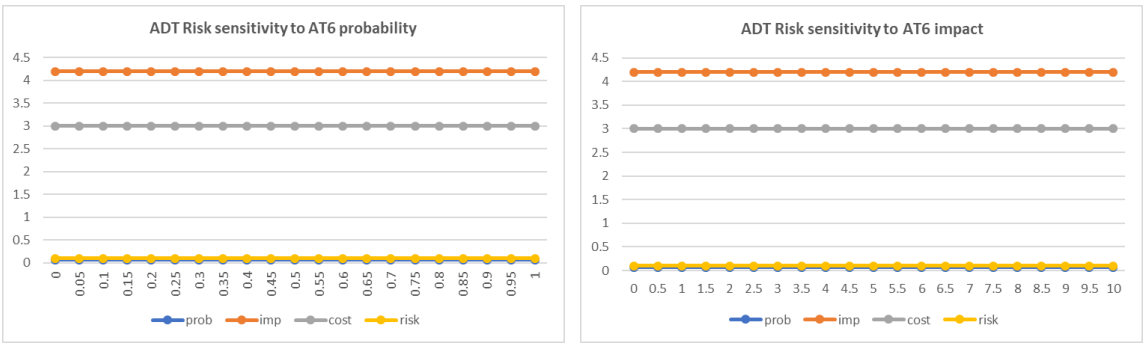


Figure 26: Risk Sensitivity to At6 attack event probability (left) and impact (right).

Sensitivity to defence attributes:

The methodology proposed for risk evaluation enabled the simulation of all the possible defence combination scenarios. The support to simulation is a powerful quality which showed very interesting results with respect to prospections on security strategies to adopt. In this simulation the probabilities of all the countermeasures were alternatively set to 0 or to the estimated value in Table 8 which enables to evaluate the ADT risks vectors of the $2^{12} = 4096$ combinations.

As a result of these calculations, the countermeasure set {RA-5_Vuln_scanning-3, SI-20_De-identification, SC-7_Boundary_Protection} renders the risk vector with the minimum risk {0.07, 4.2, 3, 0.1} even if they do not cover all the attacks in the ADT as will be shown in next section. The cost of this optimal defence set is 15 cost units.

The set {SI-20_De-identification, SC-7_Boundary_Protection} gives a risk vector of {0.11, 9.3, 8, 0.13} with a slightly higher risk (0.13) at a security cost of 13 cost units. However, the probability of success of the attack scenario (0.11) as well as its impact (9.3) would be much higher than with the minimum risk defence set. Furthermore, the defence SI-20_De-identification by itself produces a very low risk in the ADT goal {0.1, 6, 3, 0.2} with only 7 cost units, which reveals it is the one which applied in isolation gets the highest utility in system risk reduction.

Figure 27 below shows the ADT root node risk sensitivity to the SI-20_De-identification probability, impact and cost. As it can be seen, there is an inflexion point when SI-20 probability surpasses 0.6 where the risk vector values propagation rules of Table 4 makes the root node risk vector adopt the values {0.07, 4.2, 3, 0.1} of the "Steal in origin" node. Therefore, in the inflexion point, the risk vector of SI-20_De-indentification losses its impact on the system overall risk.

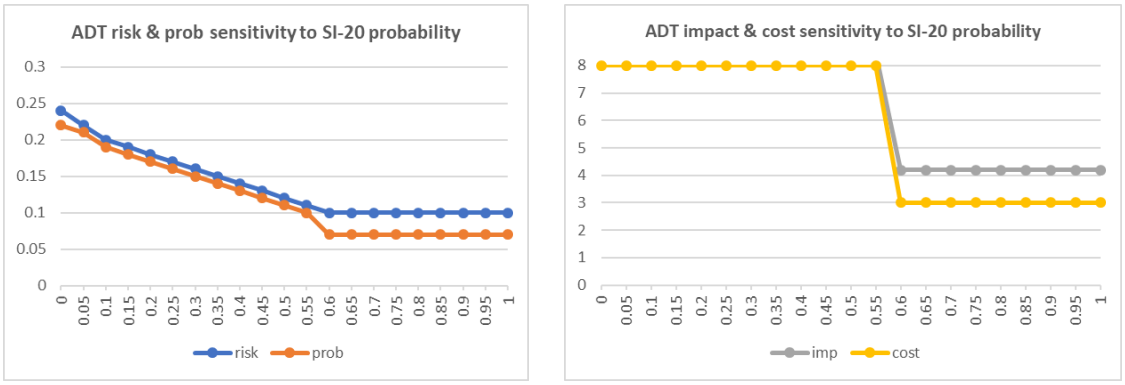


Figure 27: Case study ADT probability and risk sensitivity to SI-20 defence probability (left), and Case study ADT impact and cost sensitivity to SI-20 defence probability (right).

4.6.3 Risk-based optimisation of defences

This step involved the validation of the countermeasure optimisation methods proposed in the methodology which aided in the definition of the security strategy for the system.

The algorithm in Figure 14 was developed as part of the ADMind tool in java script language which automated the computation of all the defence combinations covering the specified set of attacks. Furthermore, java script enabled to easily enter inputs and display the optimisation results in a web page while saving in programme installation.

Full cover and minimum cost:

When studying the optimal defence set that covers all the attacks in the ADT of Figure 23, first, all the attack events in the ADT were listed in the AT vector:

AT = {

At1_Exploit_Vuln_in_GW,

At2_Copy_GW_Memory,

At3_Erase_GW_logs,

At4_Steal_in_transit_Sensor->GW,

At5_Intercept_comms,

At6_Man_in_the_middle,

At7_Spoof_GW,

At8_Exploit_Vuln_in_DB,

At9_Exploit_Vuln_in_VM,

At10_DB_Account_Hijacking,

At11_Read_Data}

And the defence set vector DS was created with all the defences in the ADT, i.e. DS = {RA-5-1, AC-6, SI-6, SC-7, SC-7.11, CA-8, SC-23, SI-4, RA-5-2, RA-5-3, AC-3, SI-20}.

Then, all the mincuts in the ADT of Figure 23 were obtained:

RA-5-1, At1, AC-6, At2, SI-6, At3

SC-7, At4

SC-7.11, At5, CA-8, SC-23, At6, SI-4, At7

RA-5-2, At8, SI-20, At11

RA-5-3, At9, SI-20, At11

AC-3, At10, SI-20, At11

By noting the target asset of each attack event, the mincuts were then expressed as a three-dimensional matrix T (11 rows x 12 columns x 4 pages) corresponding to (number of attacks x number of defences x number of assets) in the ADT as shown in Figure 28.

GW	RA-5-1	AC-6	SI-6	SC-7	SC-7.11	CA-8	SC-23	SI-4	RA-5-2	RA-5-3	AC-3	SI-20
At1	1	1	1	0	0	0	0	0	0	0	0	0
At2	1	1	1	0	0	0	0	0	0	0	0	0
At3	1	1	1	0	0	0	0	0	0	0	0	0
At4	0	0	0	1	0	0	0	0	0	0	0	0
At5	0	0	0	0	0	0	0	0	0	0	0	0
At6	0	0	0	0	0	0	0	0	0	0	0	0
At7	0	0	0	0	1	1	1	1	0	0	0	0
At8	0	0	0	0	0	0	0	0	0	0	0	0
At9	0	0	0	0	0	0	0	0	0	0	0	0
At10	0	0	0	0	0	0	0	0	0	0	0	0
At11	0	0	0	0	0	0	0	0	0	0	0	0

HMonitor	RA-5-1	AC-6	SI-6	SC-7	SC-7.11	CA-8	SC-23	SI-4	RA-5-2	RA-5-3	AC-3	SI-20
At1	0	0	0	0	0	0	0	0	0	0	0	0
At2	0	0	0	0	0	0	0	0	0	0	0	0
At3	0	0	0	0	0	0	0	0	0	0	0	0
At4	0	0	0	0	0	0	0	0	0	0	0	0
At5	0	0	0	0	1	1	1	1	0	0	0	0
At6	0	0	0	0	1	1	1	1	0	0	0	0
At7	0	0	0	0	0	0	0	0	0	0	0	0
At8	0	0	0	0	0	0	0	0	0	0	0	0
At9	0	0	0	0	0	0	0	0	0	0	0	0
At10	0	0	0	0	0	0	0	0	0	0	0	0
At11	0	0	0	0	0	0	0	0	0	0	0	0

VM2	RA-5-1	AC-6	SI-6	SC-7	SC-7.11	CA-8	SC-23	SI-4	RA-5-2	RA-5-3	AC-3	SI-20
At1	0	0	0	0	0	0	0	0	0	0	0	0
At2	0	0	0	0	0	0	0	0	0	0	0	0
At3	0	0	0	0	0	0	0	0	0	0	0	0
At4	0	0	0	0	0	0	0	0	0	0	0	0
At5	0	0	0	0	0	0	0	0	0	0	0	0
At6	0	0	0	0	0	0	0	0	0	0	0	0
At7	0	0	0	0	0	0	0	0	0	0	0	0
At8	0	0	0	0	0	0	0	0	0	0	0	0
At9	0	0	0	0	0	0	0	0	0	1	0	0
At10	0	0	0	0	0	0	0	0	0	0	0	0
At11	0	0	0	0	0	0	0	0	0	0	0	0

DB	RA-5-1	AC-6	SI-6	SC-7	SC-7.11	CA-8	SC-23	SI-4	RA-5-2	RA-5-3	AC-3	SI-20
At1	0	0	0	0	0	0	0	0	0	0	0	0
At2	0	0	0	0	0	0	0	0	0	0	0	0
At3	0	0	0	0	0	0	0	0	0	0	0	0
At4	0	0	0	0	0	0	0	0	0	0	0	0
At5	0	0	0	0	0	0	0	0	0	0	0	0
At6	0	0	0	0	0	0	0	0	0	0	0	0
At7	0	0	0	0	0	0	0	0	0	0	0	0
At8	1	0	0	0	0	0	0	0	1	0	0	1
At9	0	0	0	0	0	0	0	0	0	0	0	0
At10	0	0	0	0	0	0	0	0	0	0	1	1
At11	1	0	0	0	0	0	0	0	1	1	1	1

Figure 28: T matrix for the case study ADT.

In order to consider all the attacks of the system regardless of the asset they target, after including all the assets in the ADT within the asset vector A ={GW, HMonitor, VM2, DB}, the AD matrix (11 x 12) of Figure 29 was created by applying Equation (10):

All assets	RA-5-1	AC-6	SI-6	SC-7	SC-7.11	CA-8	SC-23	SI-4	RA-5-2	RA-5-3	AC-3	SI-20
At1	1	1	1	0	0	0	0	0	0	0	0	0
At2	1	1	1	0	0	0	0	0	0	0	0	0
At3	1	1	1	0	0	0	0	0	0	0	0	0
At4	0	0	0	1	0	0	0	0	0	0	0	0
At5	0	0	0	0	1	1	1	1	0	0	0	0
At6	0	0	0	0	1	1	1	1	0	0	0	0
At7	0	0	0	0	1	1	1	1	0	0	0	0
At8	1	0	0	0	0	0	0	0	1	0	0	1
At9	1	0	0	0	0	0	0	0	0	1	0	1
At10	0	0	0	0	0	0	0	0	0	0	1	1
At11	1	0	0	0	0	0	0	0	1	1	1	1

Figure 29: AD matrix for the case study ADT.

Hence, the input parameters for the algorithm in Figure 14 in ADMind tool were the AD matrix shown above, together with an attack vector CTS = {1,1,1,1,1,1,1,1,1,1,1}. The weights vector W was built with the defence costs in Table 8, i.e. W = {7,3,5,6,5,5,4,3,5,2,8,7}, which was used as input parameter as well to be able to calculate the second constraint of minimum security costs (maximise = false).

The total amount of defence suites that are able to guarantee full coverage of attacks was identified to be 1125 sets, i.e. 27% of the total possible combinations. From these all, the combination covering all attacks (fulfilling constraint F1 in Equation (10)) that at the same time produced the minimum cumulative cost (F3 in Equation (14)) was {AC-6_Least_privilege, SC-7_Boundary_Protection, SI-4_System_Monitoring_for_GW_whitelist_check, SI-20_De-identification} with a cumulative cost of 19 cost units.

However, by using the defence simulation in the extended ADTool, setting only the probabilities of these four countermeasures to their estimated values while the probabilities of all the rest of the defences were set to zero, produced a risk vector of the ADT root node of {0.11, 9.3, 8, 0.13}. This means that even if the attacks are all covered, the adoption of this set will reduce less the overall system risk than the minimum risk defence set {RA-5_Vuln_scanning-3, SC-7_Boundary_Protection, SI-20_De-identification} which was able to lower it till 0.1.

Partial cover and minimum cost:

The partial cover problem corresponding to the attack set in the critical quadrant, i.e. {At6_Intercept_comms, At9_Exploit_Vuln_in_DB, At10_Exploit_Vuln_in_VM, At11_DB_Account_Hijacking}, was studied in order to learn which countermeasures were most appropriate to counteract the critical attack events. Initially, one could thought on the defence set which corresponds to the defences with a direct link to the nodes in the attack set, i.e. {SC-23_Session_Authenticity, CA-8_Penetration_testing, RA-5_Vuln_scanning-3, AC-3_Access_Enforcement, SI-20_De-identification} with a total cost of 14 money units. However, as a defence in a tree mincut counteracts all attack events in the mincut, other defence sets could also counteract the selected attacks.

By setting the critical attack set vector corresponding to those four attack events only, CTS = {0,0,0,0,0,1,0,0,1,1,1}, the algorithm in Figure 14 with the same AD matrix and defence cost vector W as above, produced a minimum countermeasure set of {SI-4_System_Monitoring, SI-20_De-identification} with only a cost of 10 cost units. Now the total amount of defence combinations that cover all the attacks in CTS are 2640, the 64% of the total possible combinations (2^{12} = 4096).

Thanks to the risk sensitivity simulation, we learned that when only the suite {SI-4_System_Monitoring, SI-20_De-identification} is applied the ADT root vector is {0.1, 6, 3, 0.2}.

Therefore, the system risk decrement when applying these two defences (both defence probabilities changed from 0 to estimated values while the rest remain with nominal probability) is $\Delta Risk_{goal_\{SI-4,SI-20\}} = Risk_{goal_without_\{SI-4,SI-20\}} - Risk_{goal_\{SI-4,SI-20\}} = 0.24 - 0.2 = 0.04$, i.e. the 16.66 %.

By translating the gain in risk reduction to its money value we could deduce the Return on Investment of this optimal defence set following Equation (19).

$$ROI_{\{SI-4,SI-20\}} = \frac{\Delta Risk_{goal_\{SI-4,SI-20\}} - C_{\{SI-4,SI-20\}}}{C_{\{SI-4,SI-20\}}}$$

Thus,

$$ROI_{\{SI-4,SI-20\}} = \frac{(0.04 * \text{Risk_value}) - 10 * \text{Cost_unit}}{10 * \text{Cost_unit}}$$

By considering the estimated valuation of the elements in

Table 11 for the use case, we obtain a very high ROI when applying these two defences because the system value at risk is very high with respect to the cost of the defences:

$$ROI_{\{SI-20,AC-3\}} = \frac{(0.04 * 30{,}000{,}000) - 10 * 10{,}000}{10 * 10{,}000} = 1100\ \%$$

Table 11: Case study system valuation

Concept	Description	Monetary value	Percentage wrt value at risk
System value at risk	Proportional part of the company business value at risk when ADT attack goal is achieved.	€ 30,000,000.0	100 %
System security cost	Maximum budget for defences: internal development of security mechanisms, infrastructure required, consultancy on security analysis, defences in Cloud, etc.	€ 200,000	0.2 %
Cost_unit	Security cost unit used in ADT	€ 10,000	0.01 %

[a]*wrt= with respect to.*

4.6.4 Risk-driven selection of providers and refinement of risks

4.6.4.1 Identification of required defences in outsourced components

As explained in Section 4.6.3, thanks to the sensitivity analysis to defence attributes enabled by our methodology, the optimal defence set minimising the risk of the system was identified to be {RA-5_Vuln_scanning-3, SI-20_De-identification, SC-7_Boundary_Protection}, with a risk vector in the root node of {0.07, 4.2, 3, 0.1}. Figure 30 shows the risk evaluation resulting from the application of the optimal defence set to the ADT of Figure 23.

Therefore, the optimal defence set involves the main defences to apply in the components of Figure 6. As shown in the ADT of Figure 23, while SI-20 protects a vulnerability in the DB component developed by the internal team, the rest two defences in the set would need to be required from external providers in charge of implementing them. First, RA-5_Vuln_scanning-3 protects from the At9_Exploit_Vuln_in_VM in the outsourced VM2 service in Cloud, and second, the SC-7_Boundary_Protection is for protecting communications in the IoT GW consumed from an IoT service provider. All the rest of modelled defences did not qualify as required in the defence set with highest utility in the overall system risk minimisation at lowest cost, and therefore, from the system risk perspective, are less relevant for investment.

In summary, the decision on whether to invest in this set would need to be made on the basis of the available security budget as well as the final availability of the defences in the candidate providers of outsourced component as follows.

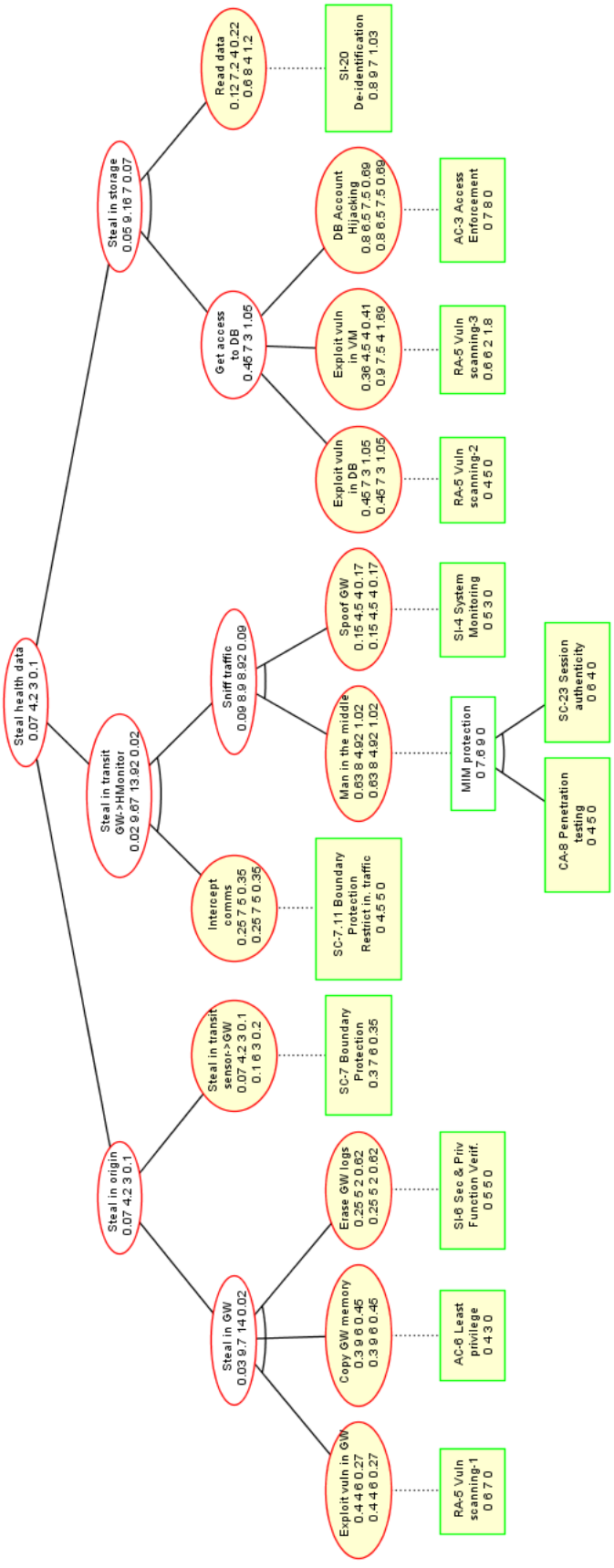


Figure 30: Case study ADT with risk vector evaluated in all the nodes when only optimal defence set is applied.

4.6.4.2 Decision on providers

For the selection of the IoT Gateway service providers of the use case in Figure 6 first a functionality match-making analysis was made on available Edge providers. The result of this analysis identified private providers SP1 and SP2 as best candidates and imposed the use of SP1 over SP2. The terms of use of the Edge device offered by SP1 provider matched the required SC-7_Boundary Protection defence in its communications.

The Fast QHP technique by Modic et al. [181] was adopted to decide on the best Cloud Service Provider (CSP) combination matching the countermeasures required for outsourced Cloud components within the modelled complete system ADT, which included the RA-5_Vuln_scanning-3 defence. In order to use this method, it was necessary to first obtain the security service level agreements (SLA) of each of the candidate external services so as to be able to compare the offered controls in the CSPs' SLAs to the identified required defences.

In the same way as in the process followed for outsourced IoT component, first from all the possible candidate CSPs, a selection of those providers matching the operational and functional requirements (such as location of data centres, high availability, etc.) was made. The SLAs of the VM Cloud Service Providers CSP1, CSP2, CSP3, CSP4, CSP5 and CSP6 in Figure 6 were taken from STAR self-assessment repository. The controls in the evaluated security SLAs were expressed in standard NIST SP 800-53 Rev. 5 [119] notation so as the match with modelled defences in the ADT was possible.

As a result, providers CSP3 for Notifier component, CSP1 for VM1 and CSP2 for VM2 were identified as the best match for the eHealth system under study because they were the ones with the best ratio of required control fulfilment. However, this optimal provider set did not offer all defences required in the complete study conducted with the whole ADT of the case study and not only the extract of Figure 7, and some defences were updated to probability 0 in the complete system ADT. Furthermore, the prices of the offered defences were refined with actual prices of services and capabilities from these Cloud Service Providers.

Following our example, the CSP2 offered two quality levels for the vulnerability scanning service on the virtual machine at different prices: the standard service at no cost and a premium service with cost of € 2,000 annually. This meant that the initially planned cost for the RA-5_Vuln_scanning-3 was not accurate as we had planned for this defence 2 cost units, which means a total of € 20,000. Therefore, this new information was used to update the ADT of the use case as explained in the following.

4.6.4.3 Risk assessment refinement

The next step after the Cloud service provider selection consisted in revisiting the system ADT and refining the modelled countermeasures for the outsourced components with the updated information about the available defences in the chosen providers together with their costs and efficacy in reducing impact. In the example, the cost of RA-5_Vuln_scanning-3 defence was updated to 0.2 cost units in the tree and its probability of success was reduced from 0.6 to 0.3, emulating a limited accuracy of the free vulnerability scanner in finding vulnerabilities. A similar process was followed with all the cloud protections required in the complete ADT.

Risk attributes of other protections for the IoT components were updated as well after security investments were done. Among them, the probability of success of SC-7_Boundary_Protection for the IoT GW in Figure 21 was increased from 0.3 to 0.8 to reflect that a new security monitoring tool was installed in the communications of the IoT GW software developed. The new risk vector for this defence resulted in {0.8, 7, 6, 0.93}.

After the optimal set {RA-5_Vuln_scanning-3, SI-20_De-identification, SC-7_Boundary_Protection} exclusively was applied in the ADT of Figure 23 and corrected estimations in RA-5_Vuln_scanning-3 and SC-7_Boundary_Protection were performed, the new risk evaluation was conducted in this new situation shown in Figure 31. The evaluation of the risk attributes propagation rules in Table 4 resulted in a new risk status for the ADT, with an updated overall system risk vector {0.05, 9.16, 7, 0.07} reflecting a further reduction of the system risk.

This result is the consequence of the updates made in SC-7_Boundary_Protection probability attribute rather than corrections to RA-5_Vuln_scanning-3 which did not affect the root node risk vector at all. The increase from 0.3 to 0.8 in the success probability of SC-7_Boundary_Protection, reduces the risk of "Steal in origin" goal from 0.1 to 0.03, which moves from a risk vector of {0.07, 4.2, 3, 0.1} (see Figure 30) to {0.02, 4.2, 3, 0.03} (see Figure 31). Therefore, in this case of refined defences, according to Table 4, the root node adopts the risk vector of “Steal in storage” goal, i.e.{0.05, 9.16, 7, 0.07}, as it is the child node with the highest risk value in the OR relationship between the children of the root node.

Similar refinements and analyses were done with the complete updated ADT which was used in further improvement of the initial risk sensitivity analyses performed and in the following continuous evaluation and refinement of attacks and defences status in all the branches of the eHealth system.

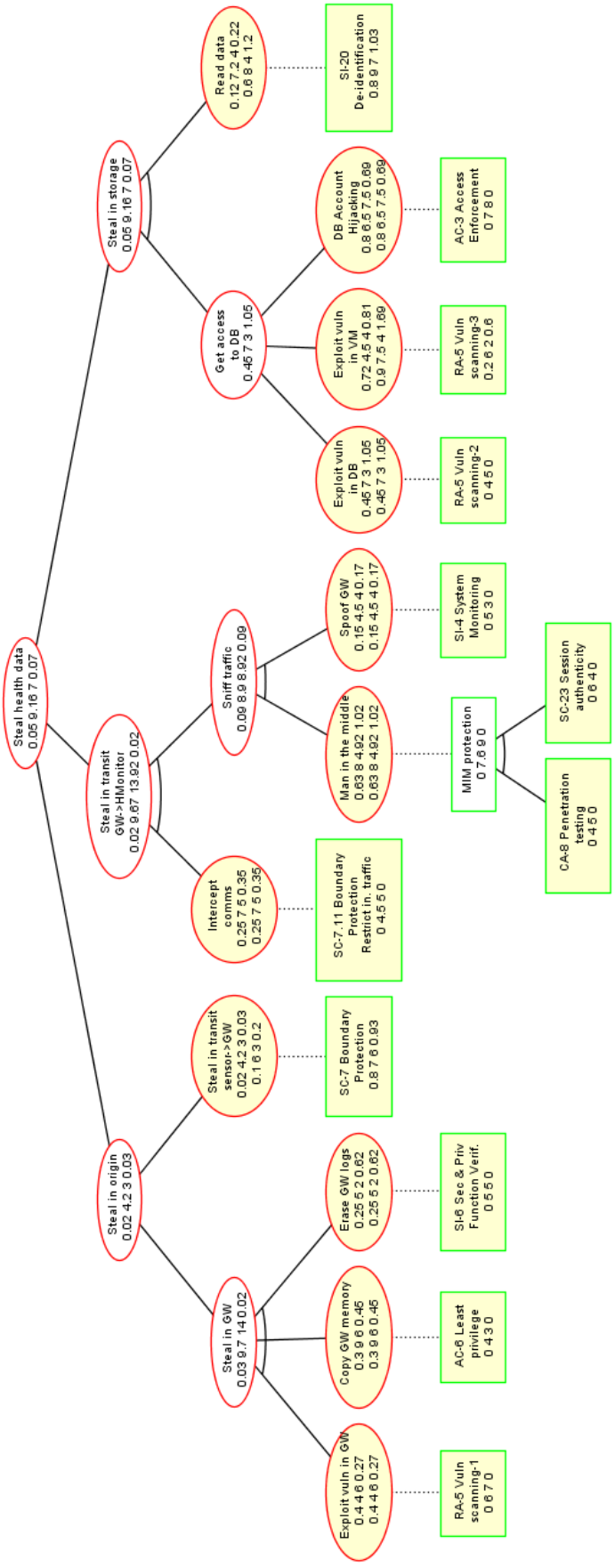


Figure 31: Case study ADT with risk vector evaluated in all the nodes after optimal defence set refinement.

4.6.5 Continuous monitoring of attacks and defences

The continuous monitoring of the eHealth system was performed by using the MUSA Security Assurance Platform described in [157] which enabled having insights of different parts of the eHealth multiCloud application. The tool required that multiple monitoring probes at network, system (IoT Edge in Figure 21) and application (IoT Gateway) were distributed together with the system components. Measures taken by the distributed probes about the status of the components were retrieved to the back-end to continuously inform on whether the deployed protections were working properly. In addition, defence agents such as access control agents were deployed as part of the system which communicated their status and performance events to the MUSA Security Assurance Platform. This allowed to continuously adjust the actions of the agents and monitor the status of these defences.

In parallel, for improving the raw estimations of the attack events in the ADT a continuous surveillance and education on the latest news about attacks occurred in similar systems as well as on all events appeared in information sharing systems which may be useful for the system was necessary.

In the evaluated time frame the system protections worked properly and all the agents' status was correct during the eHealth application runtime.

However, a Denial of Service attack event was detected by the tool when monitoring the network which affected the VM2 asset provisioned by Cloud provider CSP2 selected. In order to prevent further incidents, the DevOps team decided to upgrade the RA-5_Vuln_scanning-3 on the VM2 to the premium service model as well as activate or improve the efficacy of other defences in the ADT according to the insight enabled by our methodology on defences relevance for risk mitigation.

In summary, the methodology proposes to combine continuous monitoring of the system together with continuous surveillance on potential attacks against the system to be able to keep the ADT up to date as much as possible through accurately refining leaf-nodes' risk attributes in the tree and consequently root node risk vector.

4.6.6 Solution software prototypes: ADToolRisk and ADMind

In order to be able to support the validation of the continuous risk management methodology proposed herein the development of two dedicated tools, the so-called ADToolRisk and ADMind tools explained below, was required.

4.6.6.1 ADToolRisk

As part of our research on multiCloud risk management, we significantly enhanced the open-source software tool ADTool [184] that already supports the ADT analysis and reasoning on attack and defence strategies. The main requirements for the new ADT-based risk assessment tool were the following:

- Support to the specification of risk attribute vectors for both attacks and defences in the ADT model.
- Support to the computation of the risk vectors in all the tree nodes by application of the propagation rules.
- Graphical visualisation in the ADT model of initially specified risk vectors.
- Graphical visualisation in the ADT model of resulting risk assessment values in all the node of the tree.

- Risk sensitivity simulation cases with respect to different risk attributes in both attacks and defences.
- Risk assessment simulation of risk scenarios corresponding to different combinations of attack and defences in the system.

While keeping the original architecture and open source license of the ADTool [184], the created ADToolRisk software tool [186] fulfils all these requirements and automates the computation of risk attributes on the ADT nodes following the methodology of this Thesis and implements an enhanced graphical interface that aids in the visualisation of resulting risk attribute vectors in countered nodes as well as the bottom-up propagation of the risk attribute vectors in all ADT nodes.

Most importantly, the extended ADTool allows also prospections on defence strategies by enabling the simulation of diverse scenarios where the risk attributes of the ADT nodes are configured to desired values so as different combinations of attacks and defences can be evaluated. Therefore, the tool can be used by security and privacy analysts to reason on both privacy and security risks in diverse scenarios over ADTs of the system and aids in the decisions on the best defences to mitigate them. Furthermore, the ADToolRisk is able to rank the attack-defence scenarios on the basis of the obtained risk attribute in the tree root node.

The following risk analysis features are at the core of the developed ADToolRisk:

1. **Risk derived attribute bottom-up propagation algorithm**

   The original ADTool enables to quantitative evaluate the ADT on diverse security attribute domains, such as probability of success of an attack, costs of an attack, minimal skill level required for the attacker, time to implement all necessary defences, etc. These analyses are based on a bottom-up algorithm which propagates the attribute value from leaf-nodes to the root node of the ADT by making use of domain-specific operators while calculating attribute values for different node configurations.

   In addition to these single attribute domains, the new ADToolRisk allows to define derived attribute domains, i.e. complex attribute domains which are not simple variables but functions of other attributes. This is the case of the risk domain where it is necessary to calculate the derived risk attribute of each node in the ADT by applying Equation (2). Therefore, according to the proposed smart adversary case risk propagation rules, the evaluation of the risk attribute in an ADT node is made on top of three attribute values, which in turn are calculated on top of all the values of its children nodes. The specificity of risk propagation algorithm resides in the need of computing first the individual attributes (probability, impact, cost) and the complex attribute (risk) of all the children in order to obtain the values of the parent node.

   The tool is able to propagate bottom-up in the tree structure the calculated risk vectors in the nodes according to the rules in Table 3 and Table 4 so as the root node risk vector is obtained which represents system risk.

2. **Risk sensitivity simulation**

   A major enhancement to ADTool consisted in adding quantitative risk simulation features to the tool which aid in the informed decision on risk sensitivity and defence scenarios over the ADTs. The ADToolRisk simulation performs risk attribute propagation automation in varying scenarios aimed to identify which attack actions have more impact on the overall risk of the attack goal, which protections do better minimize the overall risk, how the combinations of different defences impact in risk minimization, how much would it cost to fully minimize the risk of the attack success, to what extent the attack impact would be minimized by a set of defences, etc.

The new simulator allows to configure which nodes and which attribute in their risk attribute vectors will be subject of simulation. The simulation algorithm is illustrated in the right-hand branch of Figure 13. The simulation iterates with progressive values of the selected attribute in the attribute vector of the node or nodes selected and computes the risk attribute vectors in all the nodes of the ADT as explained before. As a result, the three types of risk sensitivity analysis explained in section 3.4.1.2.8 can be performed with the new ADToolRisk:

- Risk sensitivity to attack attributes: In this simulation the desired attributes of selected attack nodes in the ADT are progressively increased in order to study the effects on the risk values in all the ADT nodes.
- Risk sensitivity to defence attributes: This simulation is similar to the previous one where the attributes under study are those desired on selected defences.
- Risk sensitivity to combined attack and defence attributes: This simulation combines variations of the attributes in selected both defence and attack nodes in the ADT to study their impact on the system risk.

3. **Defence strategy simulation**

   The ADToolRisk offers a powerful simulation feature capable to evaluate risks in all ADT nodes in all potential defence scenarios. The simulation algorithm is illustrated in the left-hand branch of Figure 13. Based on the hypothesis that attacks in the ADT are initially specified and the attack risk attributes are not subject to changes in the attack nodes, this simulation is able to evaluate all the possible combinations of available defences in the ADT by calculating the risk attributes in all the ADT nodes in two types of scenarios: with the defence applied (probability defence success set by the user) or not applied (which is equivalent to probability of defence success 0). Consequently, it is possible to learn which are all the possible risk scenarios resulting from the different combinations of the countermeasures captured in the ADT.

   The simulation of different defence strategies is key to identify which is the combination of countermeasures that minimises the risk in a specific node of the ADT. By knowing the resulting risk value of the root node in every single combination of available countermeasures, it is possible to select the set of defences that best minimizes the risk of the attack scenario represented by the ADT.

   Moreover, the defence strategy simulation allows to learn the necessary investment in the identified best set of defences that minimizes the risk of the attack success to the maximum, which is the final probability of the attack success or which would be the impact of the attack even if the risk is minimised.

4.6.6.2 ADMind

The defence optimisation based on attack event coverage was performed by using the ADMind software tool [187] built for that purpose.

This tool allows to identify the countermeasure set that, while covering a set of attack events of interest, enables the optimisation of a second constraint by applying the optimisation algorithm in Figure 14. Therefore, the ADMind tool enables the computation of full and partial cover problems expressed in functions F1 (Equation (10)) and F2 (Equation (11)) respectively, together with objective function F3 (Equation (14)) to minimise security costs.

This tool is based on matrix computations and it requires as input the (AD) matrix that defines the mincuts with the relations between attack events and defences in the system ADT, together with

the attack vector (AT) of attack events to be covered and the weights vector (W) of the values the defences take for the second variable that shall be optimised. The second variable is usually the cumulative cost of the defences to employ, and in this case the ADMind routine minimises it.

Remarkably, the optimal defence set for full or partial cover problems at the minimum system risk (as explained in Section 3.4.2), can also be solved by using ADMind in combination with ADT attack-defence scenario simulation in ADToolRisk.

4.6.7 Conclusion

In this Thesis we propose a holistic continuous risk assessment methodology and supporting framework for multiCloud applications based on the use of ADTs to capture the interrelation between threats and defences in the system parts. The methodology can in general be adopted in applications and complex systems with multiple components where the risk assessment requires understanding the relationships between system risks and risks at component level.

The ADT semantics proposed are consistent with the general principles of the framework of Mauw and Oostdijk [80], even though the risk derived attribute proposed is not distributive, as it adopts a worst-case scenario approach where a smart adversary would always select the option with highest risk weight for the system.

Most importantly, the methodology allows also prospections on defence strategies by enabling the simulation of diverse scenarios where the risk attributes of the ADT nodes are configured to desired values so as different combinations of attacks and defences can be evaluated. Therefore, our method supports security and privacy analysts to reason on both privacy and security risks in diverse scenarios over ADTs of the system and aids in the decisions on the best defences to mitigate them. Furthermore, the analysts are able to rank the attack-defence scenarios on the basis of the obtained risk attribute in the tree root node.

Hence, the methodology enables risk sensitivity analyses to be presented to decision making so as to guide in prioritisation of security investments. These analyses allow the fine-tuning of the initial estimations made on attack and defences risk attributes as well as assessing the variability of assessed system risks with respect to errors in initially modelled attribute values. This is particularly relevant when searching for the optimal countermeasure set that minimises system risk at the lowest cost.

The simulation of all the possible combinations of attacks and all the possible combinations of defences that the developed algorithms and risk assessment framework enable is a powerful tool to aid in the risk evaluation and countermeasure selection in complex systems such as multiCloud where multitude of attack-defence scenarios can be devised. By using matrix computations, the routines proposed in this work are computationally efficient for continuous risk assessment in ADTs with number of leaf nodes larger than 20 which is a limit identified in previous methods [90].

Furthermore, the methodology proposed takes into account the assets affected by the attacks and the protections adopted and enables informed decisions on countermeasures that can balance the risks at component and system levels. This information is relevant when protecting internal components and when requesting security controls to external providers of outsourced components such as Cloud Service Providers and IoT service providers, as is the case of multiCloud applications.

Despite the increasing interest that automatic risk assessment is attracting in the last years, few open source tools are available to efficiently compute the risks over systems which are capable to consider all risks attributes of both attacks and mitigating defences. As part of this Thesis, two tools were developed ADToolRisk and ADMind with the aim to support two key analyses that the

methodology makes possible: the automation of the continuous risk assessment and the simulation of cyber risk scenarios to analyse security implications.

4.7 Security SLA and Privacy Level Agreement Composition

The proposed SecSLA and PLA Composition methodology was tested in a set of real-world systems orchestrating multiple Cloud and IoT services and infrastructures. As explained before, the demonstration explained herein corresponds to the validation performed over the case study application depicted in Figure 21. The validation scenario demonstrates how the developers of an application that combines distributed Cloud and IoT resources can apply our SLA composition methodology to compute the Application SLA to be offered to application consumers.

The validation of the methodology was performed on a selected subset of controls (reported in Appendix C) from NIST SP 800-53 Rev. 5 [119], which are adequate representatives of the control families in real SecSLAs and PLAs. The sample selected includes security and privacy controls, as well as organisational and system controls.

From all these NIST controls considered in the use case, for SLA composition method description clarity purposes, in order to exemplify the metric delegations as well as the SLA and SLO level computation, we will use the controls shown in Table 12 together with their metrics and SLO levels.

Table 12: NIST control levels on the basis of metric levels for the case study.

Control ID	Metric ID	Metric name	Control SLO level					
			0	1	2	3	4	5
RA-5	m1	vuln scanning frequency	none	weekly	daily	cont	cont	-
	m2	vuln remediation ratio	x<=20%	20%<x<=30%	30%<x<=50%	50%<x<=70%	x>70%	-
	m3	vuln feed update frequency	none	weekly	daily	daily	cont	-
AC-3	m4	identity assurance	No	Yes	Yes	Yes	-	-
	m5	enforcement false positive rate	x>40%	30%<x<=40%	10%<x<=30%	x<=10%	-	-
	m6	shared account percentage	x>50%	20%<x<=50%	20%<x<=50%	x<=20%	-	-
	m7	access policies verification	No	Yes	Yes	Yes	-	-
SI-20(4)	m8	de-identification technique	none	mask	replace	hash	encrypt	remove
	m9	data utility loss	x>=15%	10%=<x<15%	2%<x<10%	x<=2%	x<=2%	x<=2%
	m10	de-identification only wrt marked identifiers	No	Yes	Yes	Yes	Yes	Yes

[a]*cont= continuous.*

The SLA evaluation was performed by following the methodology steps in Figure 16 as described next.

4.7.1 Application Composition Modelling

First, the ACM was obtained from the architectural model of Figure 21 by considering the nature of the components, their capability usage relationships as well as the deployment needs. The resulting ACM is shown in Figure 32 which will be later used for demonstrating the evaluation of SLA composition rules.

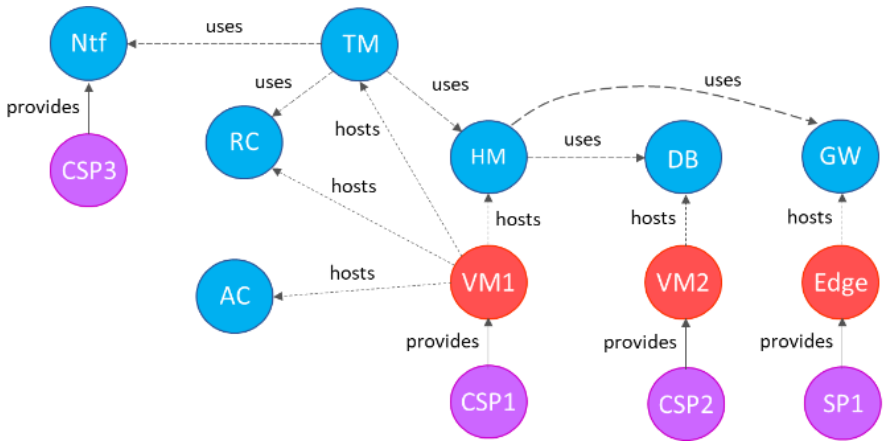


Figure 32: ACM of the eHealth multiCloud application of the case study.

4.7.2 Per-component self-assessment of SLAs

The internal components in blue in Figure 21 were individually assessed by the development team by means of a checklist based on the MUSA checklist [192] updated to include dedicated questions on NIST SP 800-53 Rev. 5 [119] security and privacy controls. As a result of the assessment the controls granted by each internal component together with the levels of the metrics they were able to offer were identified.

For these external components in the example, namely the Notifier or for short Ntf (SaaS), the virtual machines VM1 and VM2 (IaaS), and the IoT Edge (IoT infrastructure service), it was necessary that prior to starting the computation of the composition rules the development team obtained, from the outsourced service specification, SLAs and terms of use policies, the list of security and privacy controls offered by these services together with their SLOs and available metrics.

Then, as explained before, the process of control levels normalisation for all the components in the ACM followed so as a common policy level model (LSL scale) for each control was defined to be able to compare the different SLOs offered for the controls by the nodes.

In Table 13 the normalised SLO offerings of the system components with respect to the three controls of the example, RA-5, AC-3 and SI-20(4), are shown. The table shows the values for the internal components considered (all components in blue in Figure 21) as well as for the selected CSPs (CSP1 for VM1, CSP2 for VM2 and CSP3 for Notifier) and IoT Service Provider (SP1 for Edge component). Note that SI-20 family controls in the tables of Appendix C are only offered by DB component as they apply only to datasets storing personally identifiable information.

Table 13: Excerpt of SLO offers by the components and selected providers

Control ID	HM	AC	RC	TM	VM1	CSP1	DB	VM2	CSP2	GW	Edge	SP1	Ntf	CSP3
RA-5	2	2	1	2	2	2	3	2	2	1	1	False	2	4
AC-3	False	2	False	False	2	2	3	3	3	3	False	False	3	3
SI-20(4)	False	N/A	False	False	N/A	N/A	3	N/A	N/A	False	N/A	N/A	False	N/A

[a] *N/A = Not Applicable, False = No value declared.*

Table 14 collects the values of the metrics for AC-3, RA-5 and SI-20(4) controls offered by the application components for the case study deployment shown in the ACM of Figure 32. The metrics for the controls correspond to the ones defined in Table 12.

Table 14: Example control metric values for controls RA-5, AC-3, SI-20(4)

Control ID	Metric ID	HM	AC	RC	TM	VM1	CSP 1	DB	VM2	CSP 2	GW	Edge	SP1	Ntf	CSP 3
RA-5	m1	daily	daily	daily	daily	daily	daily	cont	daily	daily	weekly	daily	D	daily	D
	m2	45%	45%	20%	45%	87%	68%	52%	92%	79%	65%	95%	D	D	81%
	m3	daily	daily	daily	daily	daily	daily	cont	cont	daily	daily	weekly	D	cont	D
AC-3	m4	D	Yes	D	D	Yes	D	Yes	Yes	D	Yes	D	D	Yes	D
	m5	D	25%	D	D	15%	D	10%	5%	D	8%	D	D	10%	D
	m6	D	10%	D	D	D	22%	10%	D	10%	0%	D	D	D	18%
	m7	D	Yes	D	D	D	Yes	Yes	D	Yes	Yes	D	D	D	Yes
SI-20(4)	m8	D	N/A	D	D	N/A	N/A	hash	N/A	N/A	D	N/A	N/A	D	N/A
	m9	D	N/A	D	D	N/A	N/A	1%	N/A	N/A	D	N/A	N/A	D	N/A
	m10	D	N/A	D	D	N/A	N/A	Yes	N/A	N/A	D	N/A	N/A	D	N/A

[a]*D= delegated control metric, N/A= Not Applicable, cont= continuous.*

4.7.3 Evaluation of the Per-Component SLA Composition rules

The Control Metric Delegation Models (CMDMs) capturing control metric delegations were built by security and privacy experts of the eHealth application for each control in in the tables of Appendix C. Figure 33, Figure 34 and Figure 35 show the CMDMs defined for the RA-5, AC-3 and SI-20(4) controls and metrics of Table 12. In the figures the metric implementation ownership relationship was denoted as “o” while metric delegation is marked with “d”. “N/A” denotes that the metric implementation is not applicable in the component.

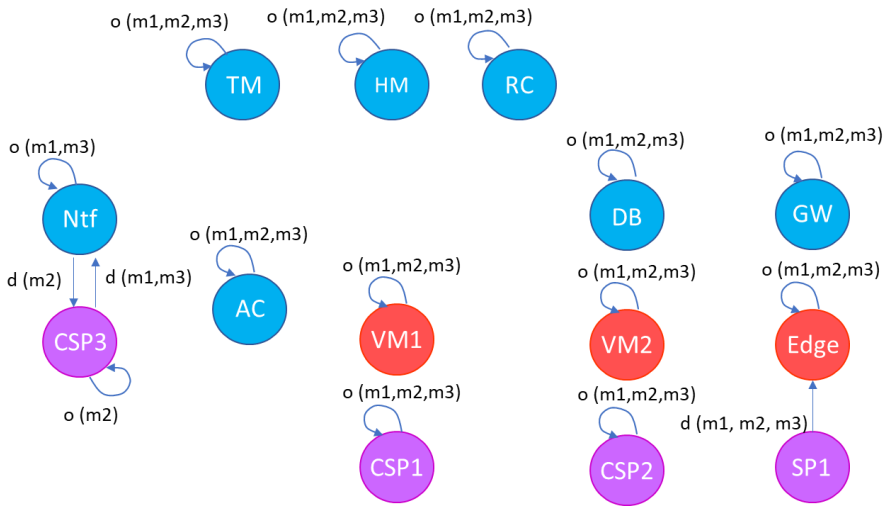


Figure 33: CMDM of the RA-5 control for the case study.

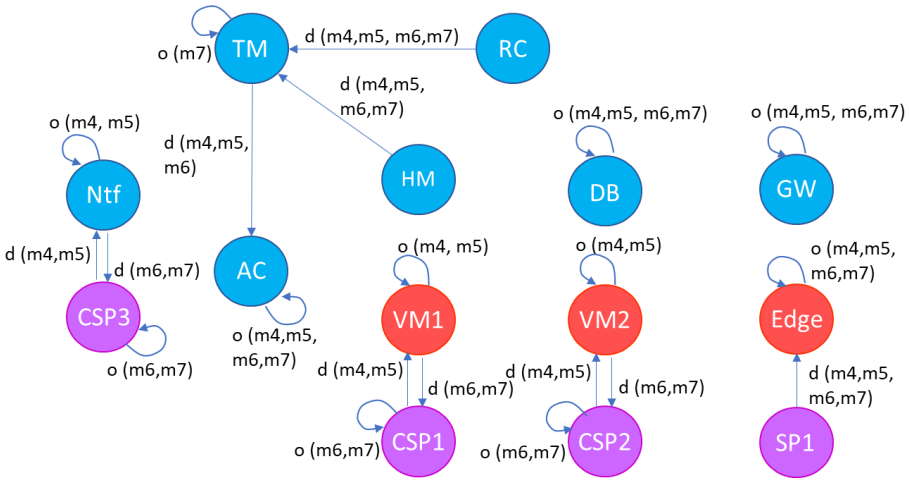


Figure 34: CMDM of the AC-3 control for the case study.

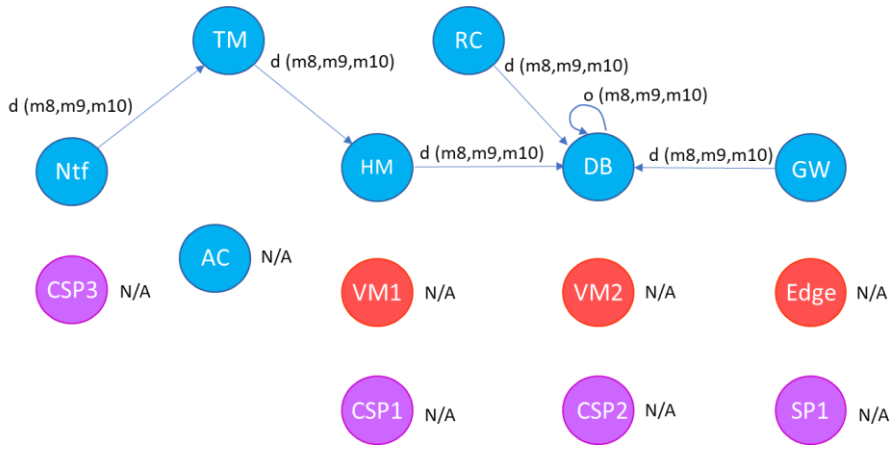


Figure 35: CMDM of the SI-20(4) control for the case study.

In the delegation of the metrics of CMDM of Figure 35 it is reflected the fact that the privacy control family SI-20 that addresses de-identification of datasets is only applicable to the DB component of the use case. Hence, the metrics for the control SI-20(4) are all owned by the DB only, which was reflected in the self-assessment of the DB component (in its SLAT) (see DB column in Table 14 for the SI-20(4) rows).

Considering the CMDMs above and the values for the metrics declared in Table 14, the result of the composed control metrics for RA-5, AC-3 and SI-20(4) controls is provided in Table 15 below.

Table 15: Composed control metrics for RA-5, AC-3 and SI-20(4) in the use case.

Control ID	Metric ID	HM	AC	RC	TM	VM1	CSP1	DB	VM2	CSP2	GW	Edge	SP1	Ntf	CSP3
RA-5	m1	daily	daily	daily	daily	daily	daily	cont	daily	daily	weekly	daily	daily	daily	daily
	m2	45%	45%	20%	45%	87%	68%	52%	92%	79%	65%	95%	95%	81%	81%
	m3	daily	daily	daily	daily	daily	daily	cont	cont	daily	cont	weekly	weekly	cont	cont
AC-3	m4	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	m5	25%	25%	25%	25%	15%	15%	10%	5%	5%	10%	10%	8%	10%	10%
	m6	10%	10%	10%	10%	22%	22%	10%	10%	10%	10%	10%	0%	18%	18%
	m7	False	Yes	False	False	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SI-20(4)	m8	hash	N/A	hash	hash	N/A	N/A	hash	N/A	N/A	hash	N/A	N/A	hash	N/A
	m9	1%	N/A	1%	1%	N/A	N/A	1%	N/A	N/A	1%	N/A	N/A	1%	N/A
	m10	Yes	N/A	Yes	Yes	N/A	N/A	Yes	N/A	N/A	Yes	N/A	N/A	Yes	N/A

[a]*N/A = Not Applicable, False = No value declared, cont= continuous.*

And the result of the application of the composition rule in Equation (26) for the Composed SLAs of the components is shown in Table 16.

Table 16: Components' Composed SLAs for controls RA-5, AC-3, SI-20(4).

Control ID	HM	AC	RC	TM	VM1	CSP1	DB	VM2	CSP2	GW	Edge	SP1	Ntf	CSP3
RA-5	T	T	T	T	T	T	T	T	T	T	T	T	T	T
AC-3	F	T	F	F	T	T	T	T	T	T	T	T	T	T
SI-20(4)	T	N/A	T	T	N/A	N/A	T	N/A	N/A	T	N/A	N/A	T	N/A

[a]*T=True, F=False, N/A = Not Applicable.*

As we can see, RA-5 and AC-3 controls would not be granted in the Composed SLAs of all the components, due to different reasons. First, the control RA-5 in the Composed SLA of RC component cannot be granted because, even if in the CMDM of Figure 33 all the delegations for the implementation of the control parts are satisfied (i.e. for all the parts exists a node that owns the

implementation of the part), the level of the metric m2 "vulnerability remediation ratio" that assesses part of the control has a value of 20 % which does only reach to level 0.

Second, the control AC-3 cannot be granted in Composed SLAs of components HM, RC and TM, because in the CMDM specification of AC-3 in Figure 34 the component TM is requested to own the metric m7 "access policies verification", and in the use case implementation Table 14, TM does not declare the implementation of m7 but a delegation (shows "D" value), so the part of the AC-3 control measured by m7 cannot be ensured in any of these components, and therefore AC-3 appears as False, i.e. not declarable in their Composed SLAs.

4.7.4 Evaluation of the Application SLA

This step involved the computation of the overall application SecSLA and PLA according to Equation (30) and Equation (31) respectively.

For each security control studied in the use case (in Appendix C), the Equation (30) considers all Composed SLAs obtained for the components in the previous step and evaluates the SecSLA for the multiCloud application as follows:

$$SecSLA\ (sc_j, app) = SecSLA(sc_j, GW) \wedge SecSLA(sc_j, Edge) \wedge SecSLA(sc_j, SP1)$$
$$\wedge SecSLA(sc_j, TM) \wedge SecSLA(sc_j, RC) \wedge SecSLA(sc_j, AC)$$
$$\wedge SecSLA(sc_j, HM) \wedge SecSLA(sc_j, VM1) \wedge SecSLA(sc_j, CSP1)$$
$$\wedge SecSLA(sc_j, DB) \wedge SecSLA(sc_j, VM2) \wedge SecSLA(sc_j, CSP2)$$
$$\wedge SecSLA(sc_j, Ntf) \wedge SecSLA(sc_j, CSP3)$$

Similarly, for each privacy and joint control in in the tables of Appendix C, the Equation (31) was computed to obtain the Application PLA as follows:

$$PLA\ (pc_j, app) = PLA(pc_j, GW) \wedge PLA(pc_j, Edge) \wedge PLA(pc_j, SP1)$$
$$\wedge PLA(pc_j, TM) \wedge PLA(pc_j, RC) \wedge PLA(pc_j, AC)$$
$$\wedge PLA(pc_j, HM) \wedge PLA(pc_j, VM1) \wedge PLA(pc_j, CSP1)$$
$$\wedge PLA(pc_j, DB) \wedge PLA(pc_j, VM2) \wedge PLA(pc_j, CSP2)$$
$$\wedge PLA(pc_j, Ntf) \wedge PLA(pc_j, CSP3)$$

This way, considering the values in each of the three rows of Table 16, the results of the above formulae for the three controls of the example (RA-5, AC-3 and SI-20(4)) are:

$$SecSLA\ (RA-5, app) = True$$
$$SecSLA\ (AC-3, app) = False$$
$$PLA\ (SI-20(4), app) = True$$

Table 17 below presents the results of this composition as they would appear in the Application SLA.

Table 17: Application SLA for controls RA-5, AC-3, SI-20(4).

Control ID	Application SLA
RA-5	T
AC-3	F
SI-20(4)	T

[a]*T=True, F=False.*

4.7.5 Computation of SLO levels in the Application SLA

For all the controls studied in the case study (in Appendix C) the levels that could be offered by the overall eHealth application were obtained by applying Equation (32) to learn the minimum of the levels shown by all components declaring the corresponding control, as follows:

$$SLO(cj, app) = min(SLO\ (c_j, HM\ ), SLO\ (c_j, DB), SLO\ (c_j, GW), \dots)$$

The result of the SLO level calculation method for the example controls is shown in Table 18, which provides the SLO levels of the controls in the Components' Composed SLAs. The components' SLOs are calculated from the metric values declared in the individual Composed SLAs of the components in Table 15 and using the control SLO level scales in Table 12.

In Table 18, each control can show three possible values: i) the control SLO level when the control can be declared, ii) False when the control cannot be declared due to either the delegations of its metrics do not fulfil the specification in the CMDM or there is no information on the control parts implementation (all metrics are delegated to other components), and iii) Not Applicable (N/A) when the control does not apply to the component, i.e. when it is not possible to implement such control in the component and no measurement of the control implementation can be gathered in the component.

Table 18: SLOs for controls RA-5, AC-3, SI-20(4) in Components' Composed SLAs

Control ID	HM	AC	RC	TM	VM1	CSP1	DB	VM2	CSP2	GW	Edge	SP1	Ntf	CSP3
RA-5	2	2	0	2	2	2	3	2	2	1	1	1	2	2
AC-3	False	2	False	False	2	2	3	3	3	3	3	3	3	3
SI-20(4)	3	N/A	3	3	N/A	N/A	3	N/A	N/A	3	N/A	N/A	3	N/A

[a]*N/A = Not Applicable, False = Not Declared.*

Finally, the result of the Application SLO level calculation for the controls in the example is shown in Table 19. For example, the SLO level for RA-5 was computed as the minimum value of the SLOs in the first row of Table 18, i.e. $SLO(RA-5, app) = 0$.

Table 19: SLOs for controls RA-5, AC-3, SI-20(4) in Application SLA

Control ID	Application SLO
RA-5	0
AC-3	False
SI-20(4)	3

[a]*False = Not Declared.*

Consequently, in the SLA composition analysis we appreciate that individual components' security and privacy levels impact those of the overall application. On one hand, some controls get eventually discarded from the overall Application SLA due to the impact of delegation relationships between the components for control implementation. This is the case of AC-3 control in our example which cannot be granted at application level, as a result of the impact of the TM component in the composition which delegates the four metrics of the control and fails to own metric m7 (see Figure 33), and therefore AC-3 could not be granted in the Composed SLAs of HM, RC and TM.

On the other hand, the control implementation delegation may render the control declaration by a component irrelevant for the application when the control metrics are all implemented and declared in other components. This is the case of SI-20(4) control metrics for component HM, RC, TM, GW and Ntf. Even though initially these components do not declare this control (see Table 14), they can actually declare it in their Composed SLAs with level 3 (see Table 18) due to the inheritance of the control implementation from the DB component which offers it with value 3, and therefore, even the application can grant SI-20(4) with level 3. The DB component is therefore key to guarantee SI-20(4) control at application level.

4.7.6 Solution software prototype: SLA Generator

The validation of the multiCloud SLA Composition methodology was partially supported by the SLA Generator tool developed in the context of the MUSA project [47].

The SLA Generator is an open source tool available in [193] and described in [192].The tool uses the Neo4j framework [194] to represent and analyse the ACM models.

The tool was used in the case study to support the creation of the ACM model from the CPIM model of the application created in the Requirements Modelling step of the DevOps methodology (see Section 4.5). The ACM model obtained for the multiCloud application under study is the one shown in Figure 32.

The SLA Generator tool is currently being updated to support the automatic generation of CMDMs from ACMs and to compute the Depth First Search algorithm in the metric delegation chains between the components in the CMDMs.

4.7.7 Conclusion

The security and privacy of complex distributed applications depend on a plethora of organisational and technical aspects that need to be properly addressed and controlled during application life-cycle, both on internal components (developed by the application development team) and on third-party components which control resides beyond the consumer. In our work we have presented a holistic methodology to compute security and privacy SLAs of multiCloud applications over standard security controls which aid in the formalisation, negotiation and assessment of the security and privacy levels of the application.

The standard controls from NIST SP 800-53 Rev. 5 [119] were selected for the methodology because they include security, privacy and joint controls while they offer a high level of completeness with respect to aspects coverage, detail and granularity. Still, the methodology is valid for other standard controls as well such as those in ISO/IEC 27000 family standards, including the latest ISO/IEC 27701 [23] on privacy controls. In fact, NIST SP 800-53 Rev. 5 [119] controls are mapped in the standard to ISO/IEC 27001 [120] and ISO/IEC 15408 [149] controls so the methodology could easily be applied with these standards as well.

The computation of the security and privacy-aware Composed SLA enables the identification of which controls can actually be granted in the overall multiCloud application and with which implementation level based on how the constituent components implement them. The proposed methodology relies on the analysis of the controls' capabilities implementation relationships between system components. To this aim, the Control Metric Delegation Model is defined to capture the control metric ownership and delegation relationships between the components in the Application Composition Model. The composition assumes the assessment of SLOs that can be granted by internal components as well as the SLOs offered by the third-party services used.

The resulting multiCloud Application SLA is built on top of the Composed SLAs of the individual components evaluated with our method. The Composed SLA may include controls that the component does not implement by itself but by other components to which it delegates the implementation of the whole or part of the control. This way, the SLA composition enables the reasoning of which security and privacy control metrics are more worthy to ensure in internal and external components so as the declarative security and privacy posture of the multiCloud application is maximised.

The method proposed has proved to advance the method by Rak [129] from which it takes its roots, since the SLA composition contemplates the control dependency models between application components which may lead to partial guarantees of controls which would prevent the control to be declared in the overall system Application SLA. Our method therefore allows identifying these situations and creating an Application SLA that only includes controls that can actually be implemented by the multiCloud components and their providers.

4.8 Conclusions

As overall result of the solution validation performed, we can conclude that the proposed integrated DevOps framework and constituent methods achieve the initial objectives of this Thesis work:

- We have proved the feasibility of proposed security-by-design and privacy-by-design mechanisms to support the specification of security and privacy requirements in multiCloud applications. The models and methods integrated in the Development phase of the DevOps methodology proposed (the modelling language, the risk assessment methodology and the SLA composition methodology) do significantly advance in the reasoned and systematic assurance of security and privacy features of multiCloud applications.
- We have also shown that our ADT-based risk assessment methodology enables to analyse and quantitatively assess multiCloud system risks, as well as the optimisation of defences to include in the system based on different constraints. The risk sensitivity analysis allowed by our methodology enables to reason on system risks when facing different attack-defence scenarios and informed decision making on how to best protect the system or individual components, as desired. The validation of the defence optimisation has shown that it permits the identification of the best defences (controls) to use in internal components and to require from CSPs providing outsourced components. The defence selection drives the system deployment over Cloud services, as the decision of which CSPs to use is determined

by the search of the best match with required security and privacy defences to minimise cyber risks.

- Finally, we have also demonstrated that our SLA composition methodology for multiCloud makes it possible to compute the offered composite Security Service Level Agreements (SecSLAs) and Privacy SLA (PLA) for the overall system, which can be used in operation to continuously ensure the fulfilment of the designed security and privacy properties in the systems with the level stated in the Service Level Objectives specified in the Application SecSLA and PLA.

The major research hypothesis of the Thesis work has therefore been proved to be true, i.e. it has been demonstrated *H- the proposed framework can contribute to the security- and privacy-aware creation and operation of multiCloud applications which specific security and privacy requirements can be analysed and specified at design time, as well as controlled in operation*.

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

- *H1- It is possible to address security and privacy aspects assurance in a continuous way in the multiCloud application life-cycle through the DevOps approach.*

The proposed integrated DevOps methodology for multiCloud applications has proved to achieve the seamless integration of the workflow activities and the support to early feedback from Operation to Development activities, as all the models defined for the application at development time (security and privacy requirements capturing models, the system ADT and the composed Application SLA) are aligned with the models used at Operation to control the security and privacy behaviour of the system (the deployment model, the system ADT and composed Application SLA).

- *H2- It is possible to express the security and privacy requirements of multiCloud applications in a way that they can be assessed in operation.*

The validated modelling language enables to capture security and privacy requirements of multiCloud applications at Cloud Provider Independent Model abstraction level which can be later transformed into deployment plans at Cloud Provider Specific Model level that include deployment details of security and privacy mechanisms or controls to be used with the application components. The security and privacy controls in the deployed components could be monitored so as violations to the composed Application SLA are detected and prompt reactions triggered.

- *H3- It is possible to continuously evaluate the security and privacy risks of multiCloud applications based on identified threats against application components and standard controls adopted by them so as to drive the selection of the best combination of Cloud Services that minimises the risks.*

The validation of the ADT-based risk assessment for multiCloud applications and the defence optimisation methods proposed have clearly shown this hypothesis is true.

- *H4 - It is possible to create Composed Service Level Agreements (SLAs) of multiCloud applications on the basis of security and privacy SLAs of their components taking into account the deployment relationships and the controls' metrics implementation delegations among the components.*

The validation of the Security SLA and PLA composition methodology has proved that it is possible to obtain the composed Application SLA including security and/or privacy controls as needed. The methodology considers in the composition the deployment architecture and the control metrics delegation relationships between the components in the architecture.

5 Conclusions

This chapter recaps the main contributions made by the research carried out in this Thesis work. More specifically, the chapter shows that the objectives proposed in this Thesis have been successfully addressed since it has been proved that the proposal presented is valid and that the results obtained in its analysis are satisfactory. In addition, publications in journals and in international conferences that have been derived from the work carried out are listed. Finally, the future lines of work that will give continuity to this Thesis research work are described.

5.1 Contributions

The technical research and development work of this Thesis has successfully led to the consecution of the first open source integrated DevOps solution in multiCloud applications which addresses security and privacy assurance in all the steps of the application creation and operation workflow.

Security assurance and compliance with GDPR [8] in multiCloud systems are two major challenges obstructing trust in Cloud services and, therefore, hindering their adoption.

Assurance in multiCloud applications requires the holistic control of multiple security and privacy capabilities at different components and layers of Cloud and IoT. To this aim it is proposed the adoption of joint security- and privacy-by-design strategies as part of a complete DevOps approach combined with continuous monitoring of security and privacy controls for the prompt reaction to incidents and attacks at runtime.

In this work a new holistic integrated DevOps methodology for security and privacy assurance in multiCloud systems is proposed which seamlessly integrates security-by-design, privacy-by-design and quantitative assurance at operation. The methodology aids in the rationalisation and systematisation of security and privacy analyses in multiCloud and enables informed decisions about security strategies to adopt in the system with regards protections to use and how they can be guaranteed to the system end-user.

The methodology relies in the creation of a number of abstraction models of the multiCloud system that, seamless integrating with one another, capture the different aspects of security and privacy and enable the study and analysis of these capabilities in the system and the selection of the best security strategy to follow.

The main contributions are highlighted below:

1. **An integrated DevOps workflow and supporting framework for security and privacy aspects consideration in the life-cycle of multiCloud applications**. The methodology enables the seamless and agile integration of security- and privacy-by-design activities, such as security and privacy requirements capturing and risk assessment, with operation activities that support the assurance of security and privacy in multiCloud. Therefore, the methodology enables multi-disciplinary DevOps teams to manage all required security and privacy aspects in the life-cycle of the overall multiCloud application.

   The continuous security and privacy assurance methodology for multiCloud applications is based on the continuous risk assessment on top of computation of system security and privacy risks and continuous tracking of controls performance and their dynamic enforcement at runtime to keep risks under desired conditions. Continuous assurance at operation involves the measuring and control of the security and privacy service level objectives specified in the application SLA. Our proposal is based on dynamically enforcing security and privacy mechanisms that are deployed together with the components

and are able to add security and privacy capabilities to the component so as the risks are kept under control.

2. **A new Security Domain Specific Language (DSL) for multiCloud applications** that builds on top of the CAMEL language. Its main contributions are related to enhanced expressiveness to define and configure desired security and privacy features of multiCloud applications at design-time. Such a powerful definition can be used later in the multiCloud application life-cycle to perform the risk analysis and to generate the application individual components' security SLAs and PLAs and the composite Application Security SLA to be offered to the customers. The defined models can also be transformed later at deployment time into infrastructure dependent deployment models to guide the deployment execution of both multiCloud application components as well as controls (enforcement agents) over them so as the security and privacy properties in the SLA can be enforced at operation.

   As part of the validation of the language, a modelling tool was created that supports the creation and checking of models for describing multiCloud applications based on the proposed Security Domain Specific Language (DSL) for multiCloud.

3. **A new Continuous Risk Management methodology for multiClud applications** which relies on capturing in the system Attack Defence Tree all the envisaged attack-defence situations of the system and assessing the risks of the system by propagating in the risk vectors in the leaf tree nodes up to the root node representing the system risk. The propagation rules in the methodology are outcomes of this research together with the algorithms to perform the risk assessment and defence optimisation.

   This way, the methodology makes it possible to continuously obtain the risk minimisation and residual risk ratios on top of the coverage of system threats calculated for the security and privacy defences applied in system components.

   In addition, the methodology enables a number of fundamental probabilistic and quantitative risk sensitivity analyses are enabled by the methodology and allows to perform simulations about different security protections to adopt in the system. Furthermore, the methodology enables to identify not only risks in the overall system but also in the individual components. This way it is possible to reason on how to best select the defences to employ at component or system level according to constraints such as available security investment.

   As part of the work, the ADToolRisk tool supporting the quantitative risk assessment simulation was developed. The tool supports the simulation of probabilistic evaluation of risks in all possible attack-defence scenarios of the system and the analysis of system risk sensitivity with respect to variability of multiple attributes of both attacks and defences.

4. **An innovative method for risk-based optimisation of defences** which enables to identify which security strategy is optimum to guarantee a cost-effective investment to reduce risks in the overall system or in desired components. This enables to perform the optimum selection of the protections of the system components and the protections to require in outsourced components supplied by external Cloud Service Providers.

   Together with the optimisation method a tool for risk-based optimisation of defences is offered, the ADMind tool, which enables searching for the optimum set of defences to apply in the system that minimises cyber risks while constrained to a second condition such as the maximum security expenses to invest in defences.

5. **A new Security SLA and Privacy Level Agreement (PLA) Composition methodology for multiCloud applications** which enables to identify the security and privacy controls

that can be guaranteed in the SLA of the composite multiCloud or Cloud-based IoT application. The methodology relies on the computation of SLA composition rules over components' SLAs according to components' relationships and deployment needs captured in the Application Composition Model. The methodology has been complemented with a proposal for benchmarking of providers using the Composed SLA. Obtaining the multiCloud Application SLA enables the operation assurance by identifying the guarantees that need to be maintained for multiCloud Application customers during application runtime. These formal guarantees shall be later be used as input to the continuous monitoring where the composed Application SLA fulfilment is verified, and early detection of security and privacy incidents in the application components and the Cloud services used is possible.

Finally, to support smooth integration of the methods above, the integrated DevOps methodology adopts in all the models created the NIST SP 800-53 Rev. 5 [119] standard taxonomy for security and privacy controls. This way, the alignment of the analyses in each of the methodology steps is guaranteed together with the transparency, auditability and reuse of the created models, which increases formalisation and transparency of the resulting multiCloud application itself.

5.2 Dissemination of the results

The major outcomes from the research work presented in this Thesis have led to the publication of several papers in international journals, in the proceedings of international conferences and in book chapters. In this section we enumerate these contributions by their category.

1. **Integrated DevOps Methodology supporting Security and Privacy in multiCloud**

   - Ortiz, A. M., Rios, E., Mallouli, W., Iturbe, E., & de Oca, E. M. (2015). Self-protecting multi-cloud applications. In *2015 IEEE Conference on Communications and Network Security (CNS)* (pp. 643-647). IEEE.
   - Rios, E., Iturbe, E., Orue-Echevarria, L., Rak, M., & Casola, V. (2015). Towards Self-Protective Multi-Cloud Applications: MUSA–a Holistic Framework to Support the Security-Intelligent Lifecycle Management of Multi-Cloud Applications. In *Proceedings of CLOSER 2015 - The 5th International Conference on Cloud Computing and Service Science*, Lisbon, May 2015.
   - Casola, V., De Benedictis, A., Rak, M., & Rios, E. (2016). Security-by-design in clouds: a security-SLA driven methodology to build secure cloud applications. In *2nd International Conference on Cloud Forward: From Distributed to Complete Computing.*, *97*, 53-62.

2. **Security and privacy requirements modelling language for multiCloud**

   - Rios, E., Iturbe, E., & Palacios, M. C. (2017). Self-healing multi-cloud application modelling. In *Proceedings of the 12th international conference on availability, reliability and security* (p. 93). ACM.
   - Casola, V., Benedictis, A. D., Rak, M., Villano, U., Rios, E., Rego, A., & Capone, G. (2019). Model-based deployment of secure multi-cloud applications. In *International Journal of Grid and Utility Computing*, 10(6), 639-653.

- Casola, V., De Benedictis, A., Rak, M., Villano, U., Rios, E., Rego, A., & Capone, G. (2017). MUSA deployer: deployment of multi-cloud applications. In *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)* (pp. 107-112). IEEE.

3. Continuous Risk assessment and risk-based defence optimisation in multiCloud

- Rios, E., Rego, A., Iturbe, E., Higuero, M., Larrucea, X., (2020). Continuous Quantitative Risk Management in Smart Grids using Attack Defence Trees. *In Sensors Journal, Special Issue "Cybersecurity and Privacy-Preserving in Modern Smart Grid", Multidisciplinary Digital Publishing Institute (MDPI).* Awaiting evaluation.
- Gupta, S., Ferrarons-Llagostera, J., Dominiak, J., Muntés-Mulero, V., Matthews, P., & Rios, E. (2017). Security-Centric Evaluation Framework for IT Services. In *International Conference on Green, Pervasive, and Cloud Computing* (pp. 736-747). Springer, Cham.

4. Security SLA Composition for multiCloud

- Rios, E., Higuero, M., Larrucea, X., Rak, M., Casola, V. & Iturbe, E. (2020). Security and Privacy SLA composition for multiCloud systems on top of standard controls. In *IEEE Transactions of Cloud Computing*. Awaiting evaluation.
- Rios, E., Iturbe, E., Larrucea, X., Rak, M., Mallouli, W., Dominiak, J., Muntés, V., Matthews, P., & Gonzalez, L. (2019). Service Level Agreement-based GDPR Compliance and Security assurance in (multi) Cloud-based systems. In *IET Software*.
- Rios, E., Mallouli, W., Rak, M., Casola, V., & Ortiz, A. M. (2016). SLA-driven monitoring of multi-cloud application components using the MUSA framework. In *2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)* (pp. 55-60). IEEE.
- Rios, E., Iturbe, E., Mallouli, W., & Rak, M. (2017, October). Dynamic security assurance in multi-cloud DevOps. In *2017 IEEE Conference on Communications and Network Security (CNS)* (pp. 467-475). IEEE. Las Vegas, Nevada, USA.
- Rios, E., Rak, M., Iturbe, E., & Mallouli, W. SLA-based continuous security assurance in multi-cloud DevOps. In *Proceedings of the International Workshop on Secure Software Engineering in DevOps and Agile Development SECSE 2017*, Oslo, Norway, September 14, 2017, Vol. 1977.

5. Additional related publications:

- Afolaranmi, S. O., Gonzalez Moctezuma, L. E., Rak, M., Casola, V., Rios, E., & Martinez Lastra, J. L. (2016). Methodology to obtain the security controls in multi-cloud applications, In *Proceedings of CLOSER 2016 –The 6th International Conference on Cloud Computing and Service Science*, Rome, April 2016.
- Carvallo, P., Cavalli, A. R., Mallouli, W., & Rios, E. (2017). Multi-cloud applications security monitoring. In *International Conference on Green, Pervasive, and Cloud Computing* (pp. 748-758). Springer, Cham.

- Rios, E., & Rak, M. (2016). Cloud challenges towards Free Flow of Data. In *Procedia Computer Science*, 97, 135-139.
- Somoskői, B., Spahr, S., Rios, E., Ripolles, O., Dominiak, J., Cserveny, T., Bálint, P., Matthews, P., Iturbe, E. & Muntés-Mulero, V. (2019). Airline Application Security in the Digital Economy: Tackling Security Challenges for Distributed Applications in Lufthansa Systems. In *Digitalization Cases* (pp. 35-58). Springer, Cham.
- Ferry, N., Solberg, A., Song, H., Lavirotte, S., Tigli, J. Y., Winter, T., Muntés-Mulero V., Metzger A., Rios Velasco E., & Castelruiz Aguirre A. (2018). ENACT: Development, Operation, and Quality Assurance of Trustworthy Smart IoT Systems. In *Proceedings of International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment* (pp. 112-127). Springer LNCS, Cham.

5.3 Future work

The outcomes of this Thesis open the door to several future research lines related to security and privacy assurance in multiCloud applications.

The philosophy and many of the concepts of the proposed security and privacy DSL for multiCloud have been studied and adopted by GeneSIS framework [195], which continues the research to support the deployment models enactment in Cloud-based IoT systems.

As part of the future research lines for continuous risk assessment in multiCloud DevOps, we plan for parametrizations of attack tree nodes in ADTs with other derived attributes such as ROI to allow deducing system level attributes relevant for security decision making. Other planned lines of research will explore possible enhancements of the risk assessment methodology to consider serialisation in attacks and in defence application over system components. This would lead to advanced decision support in the reaction step of Operations phase, when in view of the incidents actually happening in the system, security protections can be adjusted in series, for example, by enabling sequentially the available Enforcement agents.

Having robust methods to control the security during the operation of the applications is essential to obtain good results in our continuous risk assessment method. It is necessary to know with certainty the status of the system and thus be able to accurately refine the probabilities and degrees of impact estimated for the attacks. Likewise, knowing the state of the defences and their real effectiveness in counteracting attacks, it is possible to appropriately adjust the estimated values of their effectiveness to minimise risk. Therefore, we are also working on monitoring solutions for multiCloud in this line.

With regards to security and privacy SLA composition future research, we plan the integration of the methodology with the extended approach of ACM model described in [196] which includes new concepts in the multiCloud architecture such as “network” to consider also in the composition the SLAs of the communication channels between the system components. As part of this work, the SLA Generator software tool will be enhanced to support the new SLA Composition method and include CMDM derivation from ACMs and automated reasoning over the CMDMs so as the SLAs can automatically be evaluated.

Other future research lines include the cybersecurity audit and certification of multiCloud systems by using the Application SecSLAs and PLAs created with our method. The goal would be to use these models as enablers to automate the validation of required system security and privacy features.

References

[1] Forrester Inc. Predictions 2020: Cloud Computing Sees New Alliances And New Security Concerns, https://go.forrester.com/blogs/predictions-2020-cloud/ [Accessed: 13 April 2020]

[2] Gartner, 4 Trends Impacting Cloud Adoption in 2020. https://www.gartner.com/smarterwithgartner/4-trends-impacting-cloud-adoption-in-2020/ [Accessed: 13 April 2020]

[3] Flexera, Cloud Computing Trends 2019: State of the Cloud Survey. https://www.flexera.com/blog/cloud/2019/02/cloud-computing-trends-2019-state-of-the-cloud-survey/ [Accessed: 13 April 2020]

[4] European Commission. European Commission Digital Strategy, https://ec.europa.eu/info/publications/EC-Digital-Strategy_en [Accessed: 13 April 2020]

[5] European Commission. Digital Agenda for Europe - European Cloud Computing Strategy. https://ec.europa.eu/digital-single-market/en/european-cloud-computing-strategy [Accessed: 13 April 2020]

[6] ISO, 2015, ISO/IEC 27017:2015 Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services, https://www.iso.org/standard/43757.html [Accessed: 13 April 2020]

[7] ISO, 2019, ISO/IEC 27018:2019 Information technology — Security techniques — Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors, https://www.iso.org/standard/76559.html [Accessed: 13 April 2020]

[8] Regulation (EU) 2016/679, https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1532348683434&uri=CELEX:02016R0679-20160504 [Accessed: 13 April 2020]

[9] European Commission. The EU Cybersecurity Certification Framework. https://ec.europa.eu/digital-single-market/en/eu-cybersecurity-certification-framework [Accessed: 13 April 2020]

[10] Stamford, Conn. November 2019. Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17% in 2020. https://www.gartner.com/en/newsroom/press-releases/2019-11-13-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-17-percent-in-2020 [Accessed: 13 April 2020]

[11] Waidner, M. Cloud computing and security. Lecture Univ. Stuttgart. 2009.

[12] Buyya, R., Srirama, S. N., Casale, G., Calheiros, R., Simmhan, Y., Varghese, B., ... & Toosi, A. N. (2018). A manifesto for future generation cloud computing: Research directions for the next decade. ACM computing surveys (CSUR), 51(5), 1-38.

[13] Choosing a Cloud Hosting Provider with Confidence, Symantec. 2013. https://symantecssl-integrate.cloud-papers.com/content/92/choosing-cloud-hosting-provider-confidence-symantec-ssl-certificates-provide-secure-bridg [Accessed: 13 April 2020]

[14] The Future of Cloud Computing: Opportunities for European Cloud Computing Beyond 2010:--expert Group Report. European Commission, Information Society and Media, 2010.

[15] Cloud Security Alliance, 2019. Top Threats to Cloud Computing: Egregious Eleven, https://cloudsecurityalliance.org/press-releases/2019/08/09/csa-releases-new-research-top-threats-to-cloud-computing-egregious-eleven/ [Accessed: 13 April 2020]

[16] NIST SP 500-292 Cloud Computing Reference Architecture, https://www.nist.gov/publications/nist-cloud-computing-reference-architecture [Accessed: 13 April 2020]

[17] ISO, 2012, ISO/IEC 17826:2012 Information technology -- Cloud Data Management Interface (CDMI), https://www.iso.org/ [Accessed: 13 April 2020]

[18] Miller, P., Sector RoadMap: Multicloud management in 2013, September 2013.

[19] Flexera, Rightscale: 2019 State of the Cloud Report from Flexera, February 2019, https://www.flexera.com/blog/cloud/2019/02/cloud-computing-trends-2019-state-of-the-cloud-survey/ [Accessed: 13 April 2020]

[20] Forbes, 2018 Roundup Of Internet Of Things Forecasts And Market Estimates, December 2018, https://www.forbes.com/sites/louiscolumbus/2018/12/13/2018-roundup-of-internet-of-things-forecasts-and-market-estimates/#523b0d927d83 [Accessed: 13 April 2020]

[21] J. Jones, FAIR Institute. Gartner 2019 Debate: Quantitative vs. Qualitative Cyber Risk Analysis. June 2019. https://www.risklens.com/blog/gartner-2019-debate-quantitative-vs-qualitative-cyber-risk-analysis/ [Accessed: 13 April 2020]

[22] ISO, 2018, ISO/IEC 27005:2018 Information technology — Security techniques — Information security risk management, https://www.iso.org/ [Accessed: 13 April 2020]

[23] ISO, 2019, ISO/IEC 27701:2019 Security techniques — Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management — Requirements and guidelines., https://www.iso.org/ [Accessed: 13 April 2020]

[24] National Institute of Standards and Technology (NIST). NIST Cybersecurity Framework v1.1, https://www.nist.gov/cyberframework/framework [Accessed: 13 April 2020]

[25] Gartner IT Glossary – DevOps. https://www.gartner.com/en/information-technology/glossary/devops [Accessed: 13 April 2020]

[26] Imre, H. (2007). Comparison of three methodological approaches of design research. Guidelines for a Decision Support Method Adapted to NPD Processes, 361-362.

[27] Vukolić, M. (2010). The Byzantine empire in the intercloud. ACM Sigact News, 41(3), 105-111.

[28] Bohli, J. M., Gruschka, N., Jensen, M., Iacono, L. L., & Marnau, N. (2013). Security and privacy-enhancing multicloud architectures. IEEE Transactions on Dependable and Secure Computing, 10(4), 212-224.

[29] Global Inter-Cloud Technology Forum. Use Cases and Functional Requirements for Inter-Cloud Computing. Technical Report, Global Inter-Cloud Technology Forum, 2010.

[30] N. Grozev and R. Buyya, Inter-cloud architectures and application brokering: taxonomy and survey. Software: Practice and Experience, vol. 44, no. 3, pp. 369–390, 2014.

[31] Alzain, M., Soh, B., and Pardede, E. (2014). TMR-MCDB: Enhancing Security in a Multi-cloud Model through Improvement of Service Dependability. International Journal of Cloud Computing and Services Science (IJ-CLOSER), 3(3):133-144.

[32] Bernstein, D. and Vij, D. (2010). Intercloud security considerations. Proceedings - 2nd IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2010, pages 537-544.

[33] Z. Yan, H. Hongxin, A. Gail-Joon, and Y. Mengyang, Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage. IEEE Transactions on Parallel and Distributed Systems,, vol. 23, no. 12, pp. 2231–2244, 2012.

[34] P. F. Oliveira, L. Lima, T. T. V. Vinhoza, J. Barros, and M. Medard, "Trusted Storage over Untrusted Networks," Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, pp. 1–5, 2010.

[35] Bohli, J.-M., Gruschka, N., Jensen, M., Iacono, L. L., and Marnau, N. (2013). Security and Privacy-Enhancing Multicloud Architectures. IEEE Transactions on Dependable and Secure Computing, 10(4):212-224.

[36] Singhal, M., Chandrasekhar, S., Ge, T., Sandhu, R., Krishnan, R., Ahn, G.-j., and Bertino, E. (2013). Collaboration in Multicloud Computing Environments: Framework and Security Issues. IEEE Computer, pages 76-84.

[37] Cloud Standards Coordination Final Report, November 2013. https://www.etsi.org/images/files/Events/2013/2013_CSC_Delivery_WS/CSC-Final_report-013-CSC_Final_report_v1_0_PDF_format-.PDF [Accessed: 13 April 2020]

[38] The MODAClouds project. MOdel-Driven Approach for design and execution of applications on multiple Clouds (2012-2015). https://cordis.europa.eu/project/id/318484/en [Accessed: 13 April 2020]

[39] The PAASAGE project. PaaSage: Model Based Cloud Platform Upperware (2012-2016) https://cordis.europa.eu/project/id/317715 [Accessed: 13 April 2020]

[40] Ferry, N., et al. Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems. In CLOUD 2013: IEEE 6th International Conference on Cloud Computing. 2013. p. 887-894.

[41] CAMEL language documentation. https://paasage.ercim.eu/images/documents/docs/D2.1.3_CAMEL_Documentation.pdf [Accessed: 13 April 2020]

[42] The SeaClouds project. Seamless adaptive multi-cloud management of service-based applications (2013-2016). https://cordis.europa.eu/project/id/610531 [Accessed: 13 April 2020]

[43] The A4Cloud project. Accountability For Cloud and Other Future Internet Services (2012-2016). https://cordis.europa.eu/project/id/317550 [Accessed: 13 April 2020]

[44] The CUMULUS project. Certification infrastrUcture for MUlti-Layer cloUd Services (2012-2015). https://cordis.europa.eu/project/id/318580 [Accessed: 13 April 2020]

[45] The SPECS project. Secure Provisioning of Cloud Services based on SLA management (2013-2016). https://cordis.europa.eu/project/id/610795 [Accessed: 13 April 2020]

[46] D. Zeginis, F. D'andria, S. Bocconi, J. Gorronogoitia Cruz, O. Collell Martin, P. Gouvas, G. Ledakis, and K. A. Tarabanis, A usercentric multi-paas application management solution for hybrid multicloud scenarios. Scalable Computing: Practice and Experience, vol. 14, no. 1, pp. 17–32, 2013.

[47] The MUSA Project, MultiCloud Secure Applications (2015-2017). https://cordis.europa.eu/project/id/644429 [Accessed: 13 April 2020]

[48] Gartner IT Glossary: Runtime Application Self-Protection (RASP). https://www.gartner.com/en/information-technology/glossary/runtime-application-self-protection-rasp [Accessed: 13 April 2020]

[49] Prevoty, Whitepaper, A Guide to Runtime Application Self-Protection (RASP), 2018.

[50] Mens, T., & Van Gorp, P. (2006). A taxonomy of model transformation. Electronic notes in theoretical computer science, 152, 125-142.

[51] Object Management Group. Unified Modeling Language Specification; 2015. https://www.omg.org/spec/UML/2.5/ [Accessed: 13 April 2020]

[52] Object Management Group, 2006. Business Process Modeling Notation Specification version 1.0. OMG Available Specification, 17.

[53] Mouratidis, H., & Giorgini, P. (2007). Secure tropos: a security-oriented extension of the tropos methodology. International Journal of Software Engineering and Knowledge Engineering, 17(02), 285-309.

[54] Alexander, I. (2003). Misuse cases: Use cases with hostile intent. IEEE software, 20(1), 58-66.

[55] Sindre, G. (2007, June). Mal-activity diagrams for capturing attacks on business processes. In International working conference on requirements engineering: foundation for software quality (pp. 355-366). Springer, Berlin, Heidelberg.

[56] Rodríguez, A., Fernández-Medina, E., & Piattini, M. (2007, September). Towards CIM to PIM transformation: from secure business processes defined in BPMN to use-cases. In International Conference on Business Process Management (pp. 408-415). Springer, Berlin, Heidelberg.

[57] PaaSage project Deliverable D2.1.2 CloudML Implementation Documentation. April 2014. https://www.researchgate.net/publication/281600327_PaaSage_project_deliverable_D212-CloudML_Implementation_Documentation_First_version [Accessed: 13 April 2020]

[58] CAMEL Documentation v2015.9 by PaaSage EU Project. http://camel-dsl.org/documentation/ [Accessed: 13 April 2020]

[59] CloudML project. Model-based provisioning and deployment of cloud-based systems. https://github.com/SINTEF-9012/cloudml [Accessed: 13 April 2020]

[60] Quinton, C., Romero, D., and Duchien, L., Cardinality-based feature models with constraints: a pragmatic approach. In: SPLC 2013: 17th International Software Product Line Conference. Ed. by Tomoji Kishi, Stan Jarzabek and Stefania Gnesi. ACM, 2013, pp. 162–166. ISBN: 978-1-4503-1968-3. DOI: 10.1145/2491627.2491638.

[61] Keith Jeffery, Nikos Houssos, Brigitte Jörg and Anne Asserson. Research information management: the CERIF approach. In: IJMSO 9.1 (2014), pp. 5–14. DOI: 10.1504/IJMSO.2014.059142.

[62] Kritikos, K., Domaschka, J. and Rossini, A. SRL: A Scalability Rule Language for Multi-Cloud Environments. In: Cloud-Com 2014: 6th IEEE International Conference on Cloud Computing Technology and Science. Ed. by Juan E. Guerrero. IEEE Computer Society, 2014, pp. 1–9. ISBN: 978-1-4799-4093-6. DOI: 10.1109/CloudCom.2014.170.

[63] Karniavoura, F., Papaioannou, A., & Magoutis, K. (2015). C2C: An Automated Deployment Framework for Distributed Applications on Multi-Clouds. In Proceedings of 9th Symposium and Summer School On Service-Oriented Computing, Hersonissos, Crete, Greece.

[64] Alessandro Rossini. Cloud Application Modelling and Execution Language (CAMEL) and the PaaSage Workflow. In: Advances in Service-Oriented and Cloud Computing—Workshops of ESOCC 2015. Ed. by Antonio Celesti and Philipp Leitner. Vol. 567. Communications in Computer and Information Science. Springer, 2016, pp. 437–439. isbn: 978-3-319-33313-7. doi: 10.1007/978-3-319-33313-7.

[65] Deliverable 4.1 PIM4Cloud, the REMICS Project. FP7- ICT-2009.1.2. April 2012. http://www.remics.eu/system/files/REMICS_D4.1_V2.0_LowResolution.pdf [Accessed: 13 April 2020]

[66] OMG Model-Driven Architecture. http://www.omg.org/mda/ [Accessed: 13 April 2020]

[67] C. Atkinson and T. Kühne, Rearchitecting the UML infrastructure. TOMACS, vol. 12, no. 4, pp. 290–321, 2002.

[68] MODAClouds consortium. D4.2.2 MODACloudML development – Final version. September 2014. http://www.modaclouds.eu/wp-content/uploads/2012/09/MODAClouds_D4.2.2_-MODACloudMLDevelopmentFinalVersion.pdf [Accessed: 13 April 2020]

[69] Bergmayr, A., Rossini, A., Ferry, N., Horn, G., Orue-Echevarria, L., Solberg, A., & Wimmer, M. (2015). The Evolution of CloudML and its Manifestations. In Proceedings of the 3rd International Workshop on Model-Driven Engineering on and for the Cloud (CloudMDE) (pp. 1-6).

[70] Topology and Orchestration Specification for Cloud Applications Standard. TOSCA standard by OASIS. http://docs.oasis-open.org/tosca/TOSCA/v1.0/TOSCA-v1.0.html [Accessed: 13 April 2020]

[71] White, S. (2005). Using BPMN to model a BPEL process. BPTrends, 3(3), 1-18.

[72] Kordy, B., Piètre-Cambacédès, L., & Schweitzer. P., (2014). DAG-based attack and defense modeling: Don't miss the forest for the attack trees. Computer science review, 13, 1-38.

[73] Poolsappasit, N., Dewri, R. & Ray, I. (2011). Dynamic security risk management using bayesian attack graphs. IEEE Transactions on Dependable and Secure Computing, 9(1), 61-74.

[74] P. Xie, J.H. Li, X. Ou, P. Liu, and Levy, R. Using Bayesian Networks for Cyber Security Analysis. Proc. 40th IEEE/IFIP Int'l Conf. Dependable Systems and Networks, 2010.

[75] Dantu, R., Kolan, P., and Cangussu, J. Network Risk Management Using Attacker Profiling. Security and Comm. Networks vol. 2, pp. 83-96, 2009.

[76] Xinzhou Qin, X. and Lee, W. Attack plan recognition and prediction using causal networks. In Proceedings of the 20th Annual Computer Security Applications Conference, pages 370–379. IEEE Computer Society, December 2004.

[77] Schneier, B. Attack trees. Dr. Dobb's journal, 1999, 24(12), pp. 21-29.

[78] Kordy, B., Mauw, S., & Schweitzer, P. Foundations of Attack–Defense Trees. In Sandro Etalle Joshua Guttman Pierpaolo Degano, editor, FAST, vol. 6561 of LNCS, Springer, 2011. pp 80–95.

[79] Amoroso, E. G., Fundamentals of computer security technology. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994. ISBN 0-13-108929-3.

[80] Mauw, S. and Oostdijk, M., Foundations of Attack Trees. In Dongho Won and Seungjoo Kim, editors, ICISC, volume 3935 of LNCS, Springer, 2005. pp 186–198. ISBN 3-540-33354-1.

[81] Buldas, A., Laud, P., Priisalu, J., Saarepera, M., & Willemson, J.. Rational choice of security measures via multi-parameter attack trees. In International Workshop on Critical Information Infrastructures Security, vol 4347 of LNCS, Springer, Berlin, Heidelberg, August 2006, pp. 235-248.

[82] Edge, K. S., Dalton, G. C., Raines, R. A., & Mills, R. F. Using attack and protection trees to analyze threats and defenses to homeland security. In Proceedings of the 2006 Military Communications Conference, MILCOM 2006. IEEE, October, 2006, pp 1-7.

[83] Wang, J., Whitley, J. N., Phan, R. C. W., & Parish, D. J. Unified Parametrizable Attack Tree. International Journal for Information Security Research, 1(1):20–26, 2011.

[84] Edge, K., Raines, R., Baldwin, R., Grimaila, M., Reuter, C., & Bennington, R. Analyzing security measures for mobile ad hoc networks using attack and protection trees. In 2nd International Conference on i-Warfare and Security, pages 47-56, 2007.

[85] Roy, A., Kim, D. S., & Trivedi, K. S. Attack countermeasure trees (ACT): towards unifying the constructs of attack and defense trees. Security and Communication Networks, 2012, 5(8), pp 929-943.

[86] Henniger, O., Apvrille, L., Fuchs, A., Roudier, Y., Ruddle, A., & Weyl, B.. Security requirements for automotive on-board networks. In 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST), Lille, 2009, pp 641–646.

[87] Abdulla, P. A., Cederberg, J., & Kaati, L. Analyzing the Security in the GSM Radio Network Using Attack Jungles. In Tiziana Margaria and Bernhard Steffen, editors, ISoLA (1), vol 6415 of LNCS, Springer, 2010, pp 60–74. ISBN 978-3-642-16557-3.

[88] Byres, E. J., Franz, M., & Miller, D. The use of attack trees in assessing vulnerabilities in SCADA systems. In: International Infrastructure Survivability Workshop (IISW'04), IEEE, Lisbon, Portugal, December 2004.

[89] Salter, C., Saydjari, O. S., Schneier, B., & Wallner, J. Toward a Secure System Engineering Methodolgy. In Proceedings of the 1998 New Security Paradigms Workshop, ACM. Charlottesville, VA, USA, September 1998, pp 2–10.

[90] Jürgenson, A. & Willemson, J., Computing exact outcomes of multi-parameter attack trees. In OTM Confederated International Conferences "On the Move to Meaningful Internet Systems". Springer, Berlin, Heidelberg. November 2008, pp. 1036-1051

[91] Fung, C., Chen, Y. L., Wang, X., Lee, J., Tarquini, R., Anderson, M., & Linger, R., Survivability analysis of distributed systems using attack tree methodology. In Proceedings of the 2005 IEEE Military Communications Conference, vol. 1, pp 583–589. IEEE, October 2005.

[92] Kordy, B., Mauw, S., & Schweitzer, P., Quantitative questions on attack–defense trees. In International Conference on Information Security and Cryptology, Springer, Berlin, Heidelberg.November, 2012, pp. 49-64.

[93] The STRIDE Threat Model, https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx [Accessed: 13 April 2020]

[94] LeBlanc, D., & Howard, M. (2002). Writing secure code. 2nd edition, Microsoft Press.

[95] Alberts, C. J., & Dorofee, A. (2002). Managing information security risks: the OCTAVE approach. Addison-Wesley Longman Publishing Co., Inc.

[96] Lund, M. S., Solhaug, B., & Stølen, K. (2010). Model-driven risk analysis: the CORAS approach. Springer Science & Business Media.

[97] B. Karabacak and I. Sogukpinar. Isram: Information security risk analysis method. Comput. Secur., 24(2):147–159, March 2005.

[98] OWASP Risk Rating Methodology. https://owasp.org/www-community/OWASP_Risk_Rating_Methodology [Accessed: 13 April 2020]

[99] Schiffman, M., FIRST, Common Vulnerability Scoring System (CVSS), https://www.first.org/cvss/specification-document [Accessed: 13 April 2020]

[100] Pasha, M., Qaiser, G., & Pasha, U. (2018). A critical analysis of software risk management techniques in large scale systems. IEEE Access, 6, 12412-12424.

[101] Saripalli, P., & Walters, B. (2010, July). Quirc: A quantitative impact and risk assessment framework for cloud security. In 2010 IEEE 3rd international conference on cloud computing (pp. 280-288). IEEE.

[102] Djemame, K., Armstrong, D., Guitart, J., & Macias, M. (2014). A risk assessment framework for cloud computing. IEEE Transactions on Cloud Computing, 4(3), 265-278

[103] Gupta, S., Muntes-Mulero, V., Matthews, P., Dominiak, J., Omerovic, A., Aranda, J., & Seycek, S. (2015, May). Risk-driven framework for decision support in cloud service selection. In 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (pp. 545-554). IEEE.

[104] Muntés-Mulero, V., Ripolles, O., Gupta, S., Dominiak, J., Willeke, E., Matthews, P., & Somosköi, B. (2018). Agile risk management for multi-cloud software development. IET Software, 13(3), 172-181.

[105] ISO, 2011. ISO/IEC 20000-1:2011 Information technology — Service management — Part 1: Service management system requirements, https://www.iso.org/ [Accessed: 13 April 2020]

[106] European Commission, Cloud Service Level Agreement Standardisation Guidelines. (2014). http://ec.europa.eu/newsroom/dae/document.cfm?action=display&doc_id=6138 [Accessed: 13 April 2020]

[107] Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., ... & Xu, M. (2007, March). Web services agreement specification (WS-Agreement). In Open grid forum (Vol. 128, No. 1, p. 216).

[108] Ludwig, H., Keller, A., Dan, A., King, R. P., Franck, R. (2003). Web Service Level Agreement (WSLA) Language Specification V1.0., https://www.researchgate.net/publication/200827750_Web_Service_Level_Agreement_WSLA_Language_Specification [Accessed: 13 April 2020]

[109] Kandukuri, Balachandra Reddy; Paturi, V. Ramakrishna; Rakshit, Atanu. Cloud security issues. En Services Computing, 2009. SCC'09. IEEE International Conference on. IEEE, 2009. p. 517-520.

[110] Dekker, M.; Hogben, G. ENISA. Survey and analysis of security parameters in cloud SLAs across the European public sector. 2011. https://www.enisa.europa.eu/publications/survey-and-analysis-of-security-parameters-in-cloud-slas-across-the-european-public-sector/at_download/fullReport [Accessed: 13 April 2020]

[111] Almorsy, M., Grundy, J., & Ibrahim, A. S. Collaboration-based cloud computing security management framework. In Cloud Computing (CLOUD), 2011 IEEE International Conference on. IEEE, 2011. p. 364-371.

[112] Luna, J., et al. Quantitative Assessment of Cloud Security Level Agreements: A Case Study. Proc. of Security and Cryptography, 2012.

[113] Luna, J., Taha, A., Trapero, R., & Suri, N. (2015). Quantitative reasoning about cloud security using service level agreements. IEEE Transactions on Cloud Computing, 5(3), 457-471.

[114] Casola, V., De Benedictis, A., Modic, J., Rak, M., Villano, U. Per-service security sla: a new model for security management in clouds. Proc. IEEE 25th Int. Conf. on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2016, pp. 83-88.

[115] The SLA-READY project. Making Cloud SLAs readily usable in the EU private sector (2015-2016), https://cordis.europa.eu/project/id/644077 [Accessed: 13 April 2020]

[116] The SLALOM project: Service Level Agreement - Legal and Open Model (2015-2016), https://cordis.europa.eu/project/id/644270 [Accessed: 13 April 2020]

[117] Cloud Security Alliance, CSA. CSA Security Guidance for Critical Areas of Focus in Cloud Computing v4.0. https://cloudsecurityalliance.org/research/guidance/ [Accessed: 13 April 2020]

[118] Cloud Standards Customer Council. OMG: Practical Guide to Cloud Service Agreements V3.0. https://www.omg.org/cloud/deliverables/practical-guide-to-cloud-service-agreements.htm [Accessed: 13 April 2020]

[119] National Institute of Standards and Technology (NIST), Security and Privacy Controls for Information Systems and Organizations. NIST SP-800-53, Revision 5 Draft, August 2019.

[120] ISO, 2013, ISO/IEC 27001:2013 Information technology — Security techniques — Information security management systems — Requirements, https://www.iso.org/ [Accessed: 13 April 2020]

[121] ISO, 2013, ISO/IEC 27002:2013 Information technology — Security techniques — Code of practice for information security controls, https://www.iso.org/ [Accessed: 13 April 2020]

[122] ISO, 2019, ISO/IEC 27701:2019 Security techniques — Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management — Requirements and guidelines, https://www.iso.org/ [Accessed: 13 April 2020]

[123] Cloud Control Matrix (CCM), CSA, Cloud security alliance, cloud controls matrix v3.0.1. https://cloudsecurityalliance.org/artifacts/cloud-controls-matrix-v3-0-1/ [Accessed: 13 April 2020]

[124] Ahmadian, A.S., and Jürjens J. Supporting model-based privacy analysis by exploiting privacy level agreements. Proc. Int Conf. Cloud Computing Technology and Science (CloudCom), 2016 IEEE , pp. 360-365.

[125] Diamantopoulou, V., Pavlidis, M., & Mouratidis, H. (2017). Privacy level agreements for public administration information systems.

[126] Cloud Security Alliance (CSA), Code of Conduct for GDPR Compliance, https://gdpr.cloudsecurityalliance.org/wp-content/uploads/sites/2/2018/06/CSA-Code-of-Conduct-for-GDPR-Compliance.pdf [Accessed: 13 April 2020]

[127] Liu, H., Bu, F., Cai, H.: Sla-based service composition model with semantic support. Proc. Services Computing Conference (APSCC), 2012 IEEE Asia-Pacic, IEEE (2012) 374-37920.

[128] Zappatore, M., Longo, A., Bochicchio, M.A.: SLA composition in service networks. Proc. of the 30th Annual ACM Symposium on Applied Computing - SAC '15, ACM Press (2015) pp. 1219-1224.

[129] Rak, M.: 'Security assurance of (multi-) cloud application with security sla composition'. Proc. Int. Conf. on Green, Pervasive, and Cloud Computing, Springer (2017) pp. 786-799.

[130] Becker, B., Darmois, E., Kingstedt, A., Le Grand, O., Schmitting, P., & Ziegler, W. (2016, April). Survey of the Cloud Computing Standards Landscape 2015. In CLOSER vol 1, pp. 230-238.

[131] European Commission, Unleashing the Potential of Cloud Computing in Europe. https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2012:0529:FIN:EN:PDF [Accessed: 13 April 2020]

[132] Kretschmer, T. (2012), Information and Communication Technologies and Productivity Growth: A Survey of the Literature. OECD Digital Economy Papers, No. 195, OECD Publishing. https://doi.org/10.1787/5k9bh3jllgs7-en. [Accessed: 13 April 2020]

[133] Deloitte: Measuring the economic impact of cloud computing in Europe, smart number: 2014/0031, April 2016, http://ec.europa.eu/newsroom/document.cfm?doc_id=41184 [Accessed: 13 April 2020]

[134] ETSI Cloud Standards Coordination, http://csc.etsi.org/ [Accessed: 13 April 2020]

[135] ETSI TR 103 125 v1.1.1 (2012-11). https://www.etsi.org/deliver/etsi_tr/103100_103199/103125/01.01.01_60/tr_103125v010101p.pdf [Accessed: 13 April 2020]

[136] Catteddu, D., & Hogben, G. (2009). Cloud computing risk assessment, ENISA, 583-592.

[137] Dekker, M., & Hogben, G. (2011). Survey and analysis of security parameters in cloud SLAs across the European public sector. ENISA, Tech. Rep.

[138] Hogben, E. G., & Dekker, M. (2012). A guide to monitoring of security service levels in cloud contracts. ENISA, Tech. Rep. TR-2012-04-02.

[139] Liveri, D., & Skouloudi, C. (2016). Exploring Cloud Incidents. The European Network and Information Security Agency (ENISA), 1-14.

[140] Catteddu, D., & Gogben, G. (2011). Security and resilience in governmental clouds. European Network and Information Security Agency (ENISA).

[141] NIST SP 500-307 Cloud Computing Service Metrics Description, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-307.pdf [Accessed: 13 April 2020]

[142] NIST SP 800-37 Rev. 2 Risk Management Framework for Information Systems and Organizations, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-37r2.pdf [Accessed: 13 April 2020]

[143] International Organization for Standardization, ISO. https://www.iso.org/home.html [Accessed: 13 April 2020]

[144] CSA's Consensus Assessments Initiative Questionnaire (CAIQ) v3.1, https://cloudsecurityalliance.org/artifacts/consensus-assessments-initiative-questionnaire-v3-1/ [Accessed: 13 April 2020]

[145] CSA's STAR Registry. https://cloudsecurityalliance.org/star/registry/ [Accessed: 13 April 2020]

[146] Cloud Standards Customer Council (CSCC). https://www.omg.org/cloud/index.htm [Accessed: 13 April 2020]

[147] CSCC, Security for Cloud Computing. https://www.omg.org/cloud/deliverables/security-for-cloud-computing-10-steps-to-ensure-success.htm [Accessed: 13 April 2020]

[148] CSCC, Cloud Security Standards. https://www.omg.org/cloud/deliverables/cloud-security-standards-what-to-expect-and-what-to-negotiate.htm [Accessed: 13 April 2020]

[149] ISO, 2009. ISO/IEC 15408-1:2009 Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model. https://www.iso.org/ [Accessed: 13 April 2020]

[150] ISO, 2014. ISO/IEC 17788: 2014 Information technology -- Cloud computing -- Overview and vocabulary. https://www.iso.org/standard/60544.html [Accessed: 13 April 2020]

[151] ISO, 2014. ISO/IEC 17789: 2014 Information technology -- Cloud computing -- Reference architecture. https://www.iso.org/standard/60545.html [Accessed: 13 April 2020]

[152] Docker Inc. Docker. https://www.docker.com/ [Accessed: 13 April 2020]

[153] Cloudify Platform Ltd. Cloudify. http://cloudify.co/ [Accessed: 13 April 2020]

[154] Puppet Labs: IT Automation Software for System Administrators. Puppet documentation. https://docs.puppet.com/puppet/ [Accessed: 13 April 2020]

[155] Chef Software Inc. Chef technology. https://www.chef.io/chef/ [Accessed: 13 April 2020]

[156] Docker Inc. Docker Swarm. https://docs.docker.com/swarm/ [Accessed: 13 April 2020] .

[157] E. Rios, E. Iturbe, W. Mallouli & M. Rak, Dynamic security assurance in multi-cloud DevOps". In 2017 IEEE Conference on Communications and Network Security (CNS), IEEE. October, 2017, pp. 467-475.

[158] OWASP: Application Security Verification Standard, https://owasp.org/www-pdf-archive/OWASP_Application_Security_Verification_Standard_4.0-en.pdf [Accessed: 13 April 2020]

[159] Berkley Database Hardening Best Practices, https://security.berkeley.edu/education-awareness/best-practices-how-tos/system-application-security/database-hardening-best [Accessed: 13 April 2020]

[160] Casola, V., De Benedictis, A., Rak, M., Villano, U.. A security metric catalogue for cloud applications. Proc. Int. Conf. on Complex, Intelligent, and Software Intensive Systems (CISIS), July, 2017, pp. 854-863.

[161] Meng, F. C. Comparing the importance of system components by some structural characteristics. IEEE Trans. on Reliability, 45(1):59-65, 1996.

[162] Birnbaum, Z. W.. On The Importance of Different Components in a Multicomponent System. Technical report, Washington University Seattle Lab of Statistical Research, 1968.

[163] Jones, J., Gartner 2019 Debate: Quantitative vs. Qualitative Cyber Risk Analysis, FAIR Institute, June 2019, https://www.risklens.com/blog/gartner-2019-debate-quantitative-vs-qualitative-cyber-risk-analysis/ [Accessed: 13 April 2020]

[164] D. G. Feng et al. "Survey of information security risk assessment". Journal-China Institute of Communications, vol 25(7), 2004, pp 10-18.

[165] Bagnato, A., Kordy, B., Meland, P. H., & Schweitzer, P. (2012). Attribute decoration of attack–defense trees. International Journal of Secure Software Engineering (IJSSE), 3(2), 1-35.

[166] de Bijl, M. H. Using Data Analysis to Enhance Attack Trees. In Proc. Twente Student Conference, 2017.

[167] ISO, 2017, ISO/IEC 29134:2017 Information technology — Security techniques — Guidelines for privacy impact assessment, https://www.iso.org/ [Accessed: 13 April 2020]

[168] J. D. Weiss. A system security engineering process. In Proceedings of the 14th National Computer Security Conference, 1991, pp. 572-581.

[169] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms. MIT press, 2001.

[170] F.S. Hillier, G.J. Lieberman, and G.J. Liberman. Introduction to operations research. McGraw-Hill New York, 1990.

[171] D. Flanagan, G. M. Novak, Java-script: The definitive guide, 1998.

[172] G. Van Rossum, F. L. Drake, The python language reference manual, Network Theory Ltd., 2011.

[173] R. C. Team, R language definition, Vienna, Austria: R foundation for statistical computing (2000).

[174] Sonnenreich, W., Albanese, J., & Stout, B., Return on security investment (rosi)-a practical quantitative model, Journal of Research and practice in Information Technology 38 (2006) 45.

[175] Cremonini, M. and Martini, P., 2005. Evaluating information security investments from attackers perspective: the return-on-attack (ROA). In WEIS.

[176] Taha, A., Manzoor, S., & Suri, N., SLA-based service selection for multi-cloud environments. In 2017 IEEE International Conference on Edge Computing (EDGE). IEEE. June 2017, pp. 66-72.

[177] Farokhi, S., Jrad, F., Brandic, I., & Streit, A., Hierarchical SLA-based service selection for multi-cloud environments. In 4th International Conference on Cloud Computing and Services Science, 2014, pp. 722-734.

[178] Casola, V., De Benedictis, A., Rak, M., Modic, J., Erascu, M.: Automatically enforcing security slas in the cloud. IEEE Transactions on Services Computing (2016)

[179] Raspberry Pi, https://www.raspberrypi.org/ [Accessed: 13 April 2020]

[180] Arduino, https://www.arduino.cc/ [Accessed: 13 April 2020]

[181] Modic, J., Trapero, R., Taha, A., Luna, J., Stopar, M., & Suri, N. (2016). Novel efficient techniques for real-time cloud security assessment. Computers & Security, 62, 1-18.

[182] Irvine C. and Levin T. Quality of security service. In Proc. Of ACM Workshop on New security paradigms, pp. 91–99, 2001

[183] Casola, V., Mazzeo, A., Mazzocca, N., & Vittorini, V. (2007). A policy-based methodology for security evaluation: A security metric for public key infrastructures. Journal of Computer Security, 15(2), 197-229.

[184] Kordy, B., Kordy, P., Mauw, S., & Schweitzer, P. (2013). "ADTool: security analysis with attack-defense trees (extended version)", 2013.

[185] Gadyatskaya, O., Jhawar, R., Kordy, P., Lounis, K., Mauw, S., & Trujillo-Rasua, R. (2016, August). Attack trees for practical security assessment: ranking of attack scenarios with ADTool 2.0. In International Conference on Quantitative Evaluation of Systems (pp. 159-162). Springer, Cham.

[186] The ADToolRisk tool, https://github.com/ax1/ADTool2/ [Accessed: 13 April 2020]

[187] The ADMind tool, https://github.com/ax1/admind/ [Accessed: 13 April 2020]

[188] Budinsky, F., Steinberg, D., Merks, E., Ellersick, R., & Grose, T. J. (2003). Eclipse Modelling Framework: Developer's Guide.

[189] Warmer, J. B., & Kleppe, A. G. (2003). The object constraint language: getting your models ready for MDA. Addison-Wesley Professional.

[190] Eclipse Foundation Inc. 2017. Xtext Web Editor. https://www.eclipse.org/Xtext/documentation/330_web_support.html [Accessed: 13 April 2020]

[191] Red Hat Inc. 2017. Hibernate framework. http://hibernate.org [Accessed: 13 April 2020]

[192] MUSA Project Deliverable D2.3 Final SbD methods for multi-cloud applications, https://www.musa-project.eu/documents2/d23-final-sbd-methods-multi-cloud-applications [Accessed: 13 April 2020]

[193] The MUSA SLA Generator. https://bitbucket.org/cerict/sla-generator-v2/src/master/ [Accessed: 13 April 2020]

[194] Vukotic, A., Watt, N., Abedrabbo, T., Fox, D., & Partner, J. (2014). Neo4j in action. Manning Publications Co..

[195] Nicolas Ferry, Phu H. Nguyen, Towards Model-Based Continuous Deployment of Secure IoT Systems, 1st International Workshop on DevOps at MODELS (DevOps@MODELS) colocated with MODELS, Munich, Germany, 2019

[196] Casola, V., De Benedictis, A., Rak, M., & Villano, U. (2019). Toward the automation of threat modeling and risk assessment in iot systems. Internet of Things, 7, 100056.

Appendix A: CSA's PLA relationship with Security SLA and GDPR

Table Appendix A: Proposed CSA's PLA relationship with security controls in Security SLA and GDPR requirements

PLA requirement	PLA control	GDPR requirement
1. CSP Declaration of Compliance and Accountability.	DCA-1.1 to DCA-1.4	Art. 24 - Responsibility of the controller, Art. 28 - Processor
2. CSP Relevant Contacts and its Role.	CAR-1.1 to CAR-1.5	Art. 24 - Responsibility of the controller, Art. 26 - Joint controllers, Art. 27 - Representatives of controllers or processors not established in the Union, Art. 28 - Processor, Art. 29 - Processing under the authority of the controller or processor
3. Ways in which the Data will be Processed.	WWP-1.1 to WWP-1.15, WWP-2.1, WWP-3.1 to WWP-3.5, WWP-4.1 to WWP 4.2, WWP-5.1 to WWP-5.9	Art. 25 - Data protection by design and by default
4. Recordkeeping.	REC-1.1 to REC-1.8, REC-2.1 to REC-2.5	Art. 30 - Records of processing activities
5. Data Transfer.	DTR-1.1 to DTR-1.2	Chapter 5 (Art. 44 – 50) - Transfers of personal data to third countries or international organisations
6. Data Security Measures. (Security Controls -> in Security SLA.)	**SEC-1.1, SEC-1.2, SEC-1.2i - availability, SEC-1.2ii - integrity, SEC-1.2.iii - confidentiality, SEC-1.2.iv - transparency, SEC-1.2.v - isolation (purpose limitation), SEC-1.2.vi - intervenability, SEC-1.2.vii - portability, SEC-1.2.viii - accountability.**	Art. 32 - Security of processing, Art. 5 - Principles relating to processing of personal data 1(f) – integrity and confidentiality.
7. Monitoring.	MON-1.1	Art. 4 (1). The information provided to the public and to data subjects, Art. 5 - Principles relating to processing of personal data 1(a) -transparency
8. Personal Data Breach.	PDB-1.1 to PDB-1.7	Art. 33 -Notification of a personal data breach to the supervisory authority, Art. 34 -Communication of a personal data breach to the data subject, Art. 5 -

		Principles relating to processing of personal data 1(a) - transparency
9. Data Portability, Migration and Transfer Back.	PMT-1.1 to PMT-1.2	Art. 20 - Right to data portability
10. Restriction of Processing.	ROP-1.1	Art. 18 - Right to restriction of processing, Art. 5 -Principles relating to processing of personal data 1(b) - purpose limitation and 1(c) - data minimisation
11. Data Retention, Restitution and Deletion.	RRD-1.1 to RRD-1.2, RRD-2.1, RRD-3.1, RRD-4.1 to RRD-4.5	Art. 16 - Right to rectification, Art. 17 - Right to erasure ('right to be forgotten'), Art. 5 - Principles relating to processing of personal data 1(d) - accuracy and 1(e) - storage limitation.
12. Cooperation with The Cloud Customers.	CPC-1.1 to CPC-1.2	Cooperation with data subject to fulfil Chapter 3 (Art. 12 – 23) -Rights of the data subject
13. Legally Required Disclosure.	LRD-1.1	Art. 31 - Cooperation with the supervisory authority
14. Remedies for Cloud Customers.	RMD-1.1	Art. 77 - Right to lodge a complaint with a supervisory authority, Art. 79 - Right to an effective judicial remedy against a controller or processor.
15. CSP Insurance Policy	INS-1.1	Art. 82 - Right to compensation and liability

Appendix B: Security and privacy DSL model of the case study

```
camel model TreatMgmntModel {

security model SEC {
    security capability CAP1 {
        controls [MUSASEC.AC-3, MUSASEC.AC-12]
    }

    security capability CAP2 {
        controls [MUSASEC.SI-20]
    }

}


location model TreatMgmntLocation {
      region EU {
            name: Europe
      }

      country UK {
            name: UnitedKingdom
            parent regions [TreatMgmntLocation.EU]
      }

      country NO {
            name: Norway
            parent regions [TreatMgmntLocation.EU]
      }

}

requirement model TreatMgmntRequirement {
      // example of QuantitativeHardwareRequirement only applicable to a
specific Component
      quantitative hardware HMonitor {
            ram: 1024..   // always in MEGABYTES
            storage: 10.. // always in GIGABYTES
      }

      /**EUORG: maybe 32 cores is a little bit too much if we are talking
about microservices **/
      quantitative hardware CoreIntensive {
            core: 4..16
            ram: 1024..8192
      }

      quantitative hardware CPUIntensive {
            core: 1..       // min and max number of CPU cores
            ram: 1024..8192 // size of RAM
            cpu: 1.0..      // min and max CPU frequency
      }

      quantitative hardware StorageIntensive {
            storage: 1000..
      }
```

```
      os Ubuntu {os: Ubuntu 64os}

      location requirement UKReq {
            locations [TreatMgmntLocation.UK]
      }

      location requirement NorwayReq {
            locations [TreatMgmntLocation.NO]
      }

} // requirement model TreatMgmntRequirement

type model EUORGType {
      enumeration VMTypeEnum {
            values [ 'M1.MICRO' : 0,
            'M1.TINY' : 1,
            'M1.SMALL' : 2,
            'M1.MEDIUM' : 3,
            'M1.LARGE' : 4,
            'M1.XLARGE' : 5,
            'M1.XXLARGE' : 6,
            'M2.SMALL' : 7,
            'M2.MEDIUM' : 8,
            'M2.LARGE' : 9,
            'M2.XLARGE' : 10,
            'C1.SMALL' : 11,
            'C1.MEDIUM' : 12,
            'C1.LARGE' : 13,
            'C1.XLARGE' : 14,
            'C1.XXLARGE' : 15 ]
      }
      range MemoryRange {
            primitive type: IntType
            lower limit {
                  int value 256 included
            }
            upper limit {
                  int value 32768 included
            }
      }
      range StorageRange {
            primitive type: IntType
            lower limit {
                  int value 0 included
            }
            upper limit {
                  int value 160 included
            }
      }
      range CoresRange {
            primitive type: IntType
            lower limit {
                  int value 1 included
            }
            upper limit {
                  int value 16 included
            }
      }
      string value type StringValueType {
```

```
            primitive type: StringType
      }

      list StorageList {
            values [ int value 0,
            int value 20,
            int value 40,
            int value 80,
            int value 160 ]
      }

      list MemoryList {
            values [ int value 256,
            int value 512,
            int value 2048,
            int value 4096,
            int value 8192,
            int value 16384,
            int value 32768 ]
      }

      list CoresList {
            values [ int value 1,
            int value 2,
            int value 4,
            int value 8,
            int value 16 ]
      }

      range Range_0_100 {
            primitive type: IntType
            lower limit {int value 0 included}
            upper limit {int value 100}
      }

      range Range_0_10000 {
            primitive type: IntType
            lower limit {int value 0}
            upper limit {int value 10000 included}
      }

      range DoubleRange_0_100 {
            primitive type: DoubleType
            lower limit {double value 0.0 included}
            upper limit {double value 100.0 included}
      }
} // type model EUORGType

unit model TreatMgmntUnit {
      storage unit { StorageUnit: GIGABYTES }

      time interval unit {minutes: MINUTES}

      time interval unit {seconds: SECONDS}

      /*
      memory unit { MemoryUnit: MEGABYTES }
      */

      /* some examples... */
```

```
    monetary unit {Euro: EUROS}
      throughput             unit             {SimulationsPerSecondUnit:
TRANSACTIONS_PER_SECOND}
      time interval unit {ResponseTimeUnit: MILLISECONDS}
      time interval unit {ExperimentMakespanInSecondsUnit: SECONDS}
      transaction     unit     {NumberOfSimulationsLeftInExperimentUnit:
TRANSACTIONS}
      dimensionless {AvailabilityUnit: PERCENTAGE}
      dimensionless {CPUUnit: PERCENTAGE}
} // unit model EUORGUnit


organisation model EUORGOrganisation {
      organisation EUORG {
            www: 'link to Organisation web page'
            postal address: 'Organisation address'
            email: 'test.test@EUORG.no'
      }

      user test_user1 {
            first name: test_name
            last name: test_surname
            email: 'test_name.test_surname@EUORG.no'
            musa credentials {
                end time: 2025-12-31
                  username: 'erv'
                  password: 'test_name_surname'
                  }
      }

      user test_user2 {
      first name: user2_name
      last name: user2_surname
      email: 'test_name2.test_surname2@EUORG.no'
      musa credentials {
          username: 'user2'
          password: 'user2_passw'
          }
      }

      user group test_group {
      users [EUORGOrganisation.test_user1, EUORGOrganisation.test_user2]
      }


      role devop

    role assignment test_nameDevop {
        start: 2019-02-26
        end: 2020-02-26
        assigned on: 2019-02-25
        users:                              [EUORGOrganisation.test_user1,
EUORGOrganisation.test_user2]
        role: devop
    }

    role assignment test_groupDevop {
        start: 2019-02-01
        end: 2020-02-26
        assigned on:  2019-02-25
```

```
        role: devop
        user groups: [EUORGOrganisation.test_group]
    }

      security level: HIGH
} // organisation model EUORGOrganisation



application TreatMgmntApplication {
      version: 'v1.0'
      owner: EUORGOrganisation.test_user1
      deployment models [TreatMgmntModel.TreatMgmntDeployment]
} // application TreatMgmntApplication


deployment model TreatMgmntDeployment {
      requirement set HMonitorHostRS {
            os: TreatMgmntRequirement.Ubuntu
            quantitative hardware: TreatMgmntRequirement.HMonitor
            location: TreatMgmntRequirement.UKReq
      }

      requirement set CoreIntensiveUbuntuUKRS {
            os: TreatMgmntRequirement.Ubuntu
            quantitative hardware: TreatMgmntRequirement.CoreIntensive
            location: TreatMgmntRequirement.UKReq
      }

      requirement set CPUIntensiveUbuntuUKRS {
            os: TreatMgmntRequirement.Ubuntu
            quantitative hardware: TreatMgmntRequirement.CPUIntensive
            location: TreatMgmntRequirement.UKReq
      }

      requirement set CPUIntensiveUbuntuNorwayRS {
            os: TreatMgmntRequirement.Ubuntu
            quantitative hardware: TreatMgmntRequirement.CPUIntensive
            location: TreatMgmntRequirement.NOReq
      }

      requirement set StorageIntensiveUbuntuNorwayRS {
            os: TreatMgmntRequirement.Ubuntu
            quantitative                                          hardware:
TreatMgmntRequirement.StorageIntensive
            location: TreatMgmntRequirement.NOReq
      }


      vm HMonitor {
            requirement set HMonitorHostRS
            provided host HMonitorHost
      }


      vm CoreIntensiveUbuntuUK {
            requirement set CoreIntensiveUbuntuUKRS
            provided host CoreIntensiveUbuntuUKHost
      }
```

```
      vm CPUIntensiveUbuntuUK{
            requirement set CPUIntensiveUbuntuUKRS
            provided host CPUIntensiveUbuntuUKHost
      }

      vm StorageIntensiveUbuntuNorway {
            requirement set StorageIntensiveUbuntuNorwayRS
            provided host StorageIntensiveUbuntuNorwayHost
      }

      vm CPUIntensiveUbuntuNorway {
            requirement set CPUIntensiveUbuntuNorwayRS
            provided host CPUIntensiveUbuntuNorwayHost
      }

      internal component MUSAAgentAC {
            type: MUSACONF.AGENT.IDC_AC

            IP public: true

            provided security capability MUSAAgentACCap{
                security capability SEC.CAP1
            }

            required   communication   MsgBrokerPortReq1   {port:   9092
mandatory} // Required for MUSA SAP
            required   communication   MsgBrokerPortReq2   {port:   2181
mandatory} // Required for MUSA SAP

            required host CoreIntensiveUbuntuNorwayHostReq

            configuration MUSAAgentACConfigurationCHEF {
                  CHEF   configuration   manager   C1   {   //Configuration
Management tool
                        cookbook: 'musa'
                        recipe: 'musa_agent_ac'
                  }
            }
      }


      internal component TreatMgmntEngine {
            type: MUSACONF.COTS.IDM

            order:  3

            IP public:  false

          required security capability MUSAAgentACCapReq {
                security capability SEC.CAP1
            }

            provided communication TreatMgmntEnginePort { port: 8185 }
            provided communication TreatMgmntEngineRESTPort { port: 443
}
            required   communication   IDManagerPortReq   {port:   3000
mandatory}
            required communication RecommCalculatorPortReq {port: 9090
mandatory}
```

```
            required    communication    HMonitorPortReq    {port:    8085
mandatory}


            required host CoreIntensiveUbuntuNorwayHostReq

            configuration TreatMgmntEngineConfigurationCHEF {
                  CHEF   configuration   manager   C1   {   //Configuration
Management tool
                        cookbook: 'EUORG'
                        recipe: 'musa_TreatMgmnt'
                  }
            }
      }

      internal component HMonitor {
          type: MUSACONF.SERVICE.Firewall

            order: 4

            provided security capability HMonitorCap {
                security capability SEC.CAP1
            }

            provided communication HMonitorPort {port: 8085}
            required communication DatabasePortReq {port: 3306 mandatory}

            required host HMonitorHostReq

            configuration HMonitorManualConfiguration{
                  CHEF configuration manager C1 {
                        cookbook: 'EUORG'
                        recipe: 'musa_hmon'
                  }
            }
      }

      internal component RecommCalculator {
          IP public: false
          provided security capability ccc {
              security capability SEC.CAP1
          }


            provided communication RecommCalculatorPort {port: 9090}
            required host CPUIntensiveUbuntuUKHostReq
            configuration RecommCalculatorManualConfiguration{
                  CHEF configuration manager C1 {
                        cookbook: 'EUORG'
                        recipe: 'musa_rec'
                  }
            }
      }

      internal component IDMAM {
            provided communication IDManagerPort {port: 3000}
            provided communication MongoDBPort {port: 27017}
            required host StorageIntensiveUbuntuNorwayHostReq
            configuration IDManagerManualConfiguration{
```

```
                  CHEF  configuration  manager  C1  {  //Configuration
Management tool
                        cookbook: 'EUORG'
                        recipe: 'musa_idm'   // IDM database installed
within the same recipe
                  }
            }
      }

      internal component eHealthDatabase {
            provided communication DatabasePort {port: 3306}
            required host StorageIntensiveUbuntuNorwayHostReq
            required security capability eHealthDatabaseCapReq{
                security capability SEC.CAP2
            }
            configuration DatabaseManualConfiguration{
                  CHEF  configuration  manager  C1  {  //Configuration
Management tool
                        cookbook: 'EUORG'
                        recipe: 'musa_db'
                  }
            }
      }

      hosting TreatMgmntEngineToCoreIntensiveUbuntuUKHost {
            from    TreatMgmntEngine.CoreIntensiveUbuntuUKHostReq    to
CoreIntensiveUbuntuUK.CoreIntensiveUbuntuUKHost
      }


      hosting HMonitorToSpecificHMonitorHost {
            from HMonitor.HMonitorHostReq to HMonitor.HMonitorHost
      }

      hosting RecommCalculatorToCPUIntensiveUbuntuUK {
            from    RecommCalculator.CPUIntensiveUbuntuUKHostReq    to
CPUIntensiveUbuntuUK.CPUIntensiveUbuntuUKHost
      }

      hosting IDManagerToCoreIntensiveUbuntuUK {
            from          IDMAM.CoreIntensiveUbuntuUKHostReq          to
CoreIntensiveUbuntuUKRS.CoreIntensiveUbuntuUKHost
      }

      hosting DatabaseToStorageIntensiveUbuntuNorway {
            from      IDMAM.StorageIntensiveUbuntuNorwayHostReq      to
StorageIntensiveUbuntuNorway.StorageIntensiveUbuntuNorwayHost
      }

      communication HMonitorToDatabase {
            type: REMOTE
            from               HMonitor.DatabasePortReq               to
eHealthDatabase.DatabasePort
            protocol MYSQL
      }

      communication TreatMgmntEngineToIDManager {
            type: REMOTE
            from          TreatMgmntEngine.IDManagerPortReq          to
IDMAM.IDManagerPort
```

```
	}

	communication TreatMgmntEngineToHMonitor {
		type: REMOTE
		from		TreatMgmntEngine.HMonitorPortReq		to
HMonitor.HMonitorPort
	}

	communication TreatMgmntEngineToRecommCalculator {
		type: REMOTE
		from	TreatMgmntEngine.RecommCalculatorPortReq	to
RecommCalculator.RecommCalculatorPort
	}

	capability match TreatMgmntEngineTanner {
	  from		TreatMgmntEngine.MUSAAgentACCapReq		to
MUSAAgentAC.MUSAAgentACCap
	}

} // end deployment model TreatMgmnt App Deployment


//Metric model for TreatMgmnt App
metric model TreatMgmntMetric {
	window Win5Min {
		window type: SLIDING
		size type: TIME_ONLY
		time size: 5
		unit: TreatMgmntModel.TreatMgmntUnit.minutes
	}

	window Win1Min {
		window type: SLIDING
		size type: TIME_ONLY
		time size: 1
		unit: TreatMgmntModel.TreatMgmntUnit.minutes
	}

	schedule Schedule1Min {
		type: FIXED_RATE
		interval: 1
		unit: TreatMgmntModel.TreatMgmntUnit.minutes
	}

	schedule Schedule1Sec {
		type: FIXED_RATE
		interval: 1
		unit: TreatMgmntModel.TreatMgmntUnit.seconds
	}

	property AvailabilityProperty {
		type: MEASURABLE
		sensors [TreatMgmntMetric.AvailabilitySensor]
	}

	property CPUProperty {
		type: MEASURABLE
		sensors [TreatMgmntMetric.CPUSensor]
	}
```

```
	property ResponseTimeProperty {
		type: MEASURABLE
		sensors [TreatMgmntMetric.ResponseTimeSensor]
	}

	property FrequencyOfVulnerabilityScanningProperty {
		type: MEASURABLE
		sensors [TreatMgmntMetric.FreqOfVulnScanSensor]
	}

	sensor AvailabilitySensor {
		configuration: 'MMTAgent.Availability'
		push
	}

	sensor CPUSensor {
		configuration: 'MMTAgent.CPU'
		push
	}

	sensor ResponseTimeSensor {
		push
	}

	sensor FreqOfVulnScanSensor {
		configuration: 'MMTAgent.FreqOfVulnScan'
		push
	}
	raw metric AvailabilityMetric {
		value direction: 1
		layer: SaaS
		property:
TreatMgmntModel.TreatMgmntMetric.AvailabilityProperty
		unit: TreatMgmntModel.TreatMgmntUnit.AvailabilityUnit
		value type: TreatMgmntModel.EUORGType.DoubleRange_0_100
	}

	raw metric CPUMetric {
		value direction: 0
		layer: IaaS
		property: TreatMgmntModel.TreatMgmntMetric.CPUProperty
		unit: TreatMgmntModel.TreatMgmntUnit.CPUUnit
		value type: TreatMgmntModel.EUORGType.Range_0_100
	}

	raw metric ResponseTimeMetric {
		value direction: 0
		layer: SaaS
		property:
TreatMgmntModel.TreatMgmntMetric.ResponseTimeProperty
		unit: TreatMgmntModel.TreatMgmntUnit.ResponseTimeUnit
		value type: TreatMgmntModel.EUORGType.Range_0_10000
	}

	composite                                                    metric
MeanValueOfResponseTimeOfAllTreatMgmntEngineMetric {
		value direction: 0
		layer: SaaS
		property:
TreatMgmntModel.TreatMgmntMetric.ResponseTimeProperty
```

```
            unit: TreatMgmntModel.TreatMgmntUnit.ResponseTimeUnit

            metric                                                    formula
MeanValueOfResponseTimeOfAllTreatMgmntEngineFormula {
                  function arity: UNARY
                  function pattern: MAP

      MEAN(TreatMgmntModel.TreatMgmntMetric.ResponseTimeMetric)
            }
      }

      composite metric CPUAverage {
            description: "Average usage of the CPU"
            value direction: 1
            layer: PaaS
            property: TreatMgmntModel.TreatMgmntMetric.CPUProperty
            unit: TreatMgmntModel.TreatMgmntUnit.CPUUnit

            metric formula Formula_Average {
                  function arity: UNARY
                  function pattern: REDUCE
                  MEAN( TreatMgmntModel.TreatMgmntMetric.CPUMetric )
            }
      }

      raw metric context TreatMgmntEngineAvailabilityContext {
            metric: TreatMgmntModel.TreatMgmntMetric.AvailabilityMetric
            sensor: TreatMgmntMetric.AvailabilitySensor
            component:
TreatMgmntModel.TreatMgmntDeployment.TreatMgmntEngine
            quantifier: ANY
      }

      raw metric context CPUMetricConditionContext {
            metric: TreatMgmntModel.TreatMgmntMetric.CPUMetric
            sensor: TreatMgmntMetric.CPUSensor
            component:
TreatMgmntModel.TreatMgmntDeployment.TreatMgmntEngine
            quantifier: ANY
      }

      raw metric context TreatMgmntEngineResponseTimeContext {
            metric: TreatMgmntModel.TreatMgmntMetric.ResponseTimeMetric
            sensor: TreatMgmntMetric.ResponseTimeSensor
            component:
TreatMgmntModel.TreatMgmntDeployment.TreatMgmntEngine
            quantifier: ANY
      }

      raw metric context HMonitorResponseTimeContext {
            metric: TreatMgmntModel.TreatMgmntMetric.ResponseTimeMetric
            sensor: TreatMgmntMetric.ResponseTimeSensor
            component: TreatMgmntModel.TreatMgmntDeployment.HMonitor
            quantifier: ANY
      }

      raw metric context CPURawMetricContext {
            metric: TreatMgmntModel.TreatMgmntMetric.CPUMetric
            sensor: TreatMgmntMetric.CPUSensor
```

```
            component:
TreatMgmntModel.TreatMgmntDeployment.TreatMgmntEngine
            schedule: TreatMgmntModel.TreatMgmntMetric.Schedule1Sec
            quantifier: ALL
      }

      composite metric context CPUAvgMetricContextAll {
            metric: TreatMgmntModel.TreatMgmntMetric.CPUAverage
            component:
TreatMgmntModel.TreatMgmntDeployment.TreatMgmntEngine
            window: TreatMgmntModel.TreatMgmntMetric.Win5Min
            schedule: TreatMgmntModel.TreatMgmntMetric.Schedule1Min
            composing                  metric                  contexts
[TreatMgmntModel.TreatMgmntMetric.CPURawMetricContext]
            quantifier: ALL
      }

      composite metric context CPUAvgMetricContextAny {
            metric: TreatMgmntModel.TreatMgmntMetric.CPUAverage
            component:
TreatMgmntModel.TreatMgmntDeployment.TreatMgmntEngine
            window: TreatMgmntModel.TreatMgmntMetric.Win1Min
            schedule: TreatMgmntModel.TreatMgmntMetric.Schedule1Min
            composing                  metric                  contexts
[TreatMgmntModel.TreatMgmntMetric.CPURawMetricContext]
            quantifier: ANY
      }

      metric condition TreatMgmntEngineAvailabilityCondition {
            context:
TreatMgmntModel.TreatMgmntMetric.TreatMgmntEngineAvailabilityContext
            threshold: 99.0
            comparison operator: >
      }

      metric condition CPUMetricCondition {
            context:
TreatMgmntModel.TreatMgmntMetric.CPUMetricConditionContext
            threshold: 80.0
            comparison operator: >
      }

      metric condition TreatMgmntEngineResponseTimeCondition {
            context:
TreatMgmntModel.TreatMgmntMetric.TreatMgmntEngineResponseTimeContext
            threshold: 0.3
            comparison operator: <
      }

      metric condition HMonitorResponseTimeCondition {
            context:
TreatMgmntModel.TreatMgmntMetric.HMonitorResponseTimeContext
            threshold: 700.0
            comparison operator: >
      }

      metric condition CPUAvgMetricConditionAll {
            context:
TreatMgmntModel.TreatMgmntMetric.CPUAvgMetricContextAll
            threshold: 50.0
```

```
            comparison operator: >
      }

      metric condition CPUAvgMetricConditionAny {
            context:
TreatMgmntModel.TreatMgmntMetric.CPUAvgMetricContextAny
            threshold: 80.0
            comparison operator: >
      }
} // end metric model TreatMgmntMetric {


}
```

Appendix C: Subset of controls from NIST SP 800-53 Rev 5 Draft used in the case study

Table Appendix C (a): Subset of controls from NIST SP 800-53 Rev 5 Draft used in the case study.

Control ID	S/P/J[a]	O/S[b]	Control Name & Definition
RA-5	S	O	**Vulnerability scanning**: a. Scan for vulnerabilities in the system and hosted applications [Assignment: organization-defined frequency and/or randomly in accordance with organization-defined process] and when new vulnerabilities potentially affecting the system are identified and reported; b. Employ vulnerability scanning tools and techniques that facilitate interoperability among tools and automate parts of the vulnerability management process by using standards for: 1. Enumerating platforms, software flaws, and improper configurations; 2. Formatting checklists and test procedures; and 3. Measuring vulnerability impact; c. Analyze vulnerability scan reports and results from control assessments; d. Remediate legitimate vulnerabilities [Assignment: organization-defined response times] in accordance with an organizational assessment of risk; e. Share information obtained from the vulnerability scanning process and control assessments with [Assignment: organization-defined personnel or roles] to help eliminate similar vulnerabilities in other systems; and f. Employ vulnerability scanning tools that include the capability to readily update the vulnerabilities to be scanned.
AC-3	S	S	**Access Enforcement**: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.
SI-20(4)	P	O/S	**De-Identification | Removal, Masking, Encryption, Hashing, Or Replacement of Direct Identifiers**: Remove, mask, encrypt, hash, or replace direct identifiers in a dataset.
AC-6	S	O	**Least privilege**: Employ the principle of least privilege, allowing only authorized accesses for users (or processes acting on behalf of users) which are necessary to accomplish assigned tasks in accordance with organizational missions and business functions.
CA-8	S	O	**Penetration testing**: Conduct penetration testing [Assignment: organization-defined frequency] on [Assignment: organization-defined systems or system components].
SC-5	S	S	**Denial of service protection**: Protect against or limit the effects of the following types of denial of service events: [Assignment: organization-defined types of denial of service events or references to sources for such information] by employing [Assignment: organization-defined security safeguards].
SC-16	J	S	**Transmission of Security and Privacy Attributes**: Associate [Assignment: organization-defined security and privacy attributes] with information exchanged between systems and between system components.

Control ID	S/P/J[a]	O/S[b]	Control Name & Definition
SI-4	S	O/S	**System Monitoring**: a. Monitor the system to detect: 1. Attacks and indicators of potential attacks in accordance with [Assignment: organization-defined monitoring objectives]; and 2. Unauthorized local, network, and remote connections; b. Identify unauthorized use of the system through [Assignment: organization-defined techniques and methods]; c. Invoke internal monitoring capabilities or deploy monitoring devices: 1. Strategically within the system to collect organization-determined essential information; and 2. At ad hoc locations within the system to track specific types of transactions of interest to the organization; d. Protect information obtained from intrusion-monitoring tools from unauthorized access, modification, and deletion; e. Adjust the level of system monitoring activity when there is a change in risk to organizational operations and assets, individuals, other organizations, or the Nation; f. Obtain legal opinion regarding system monitoring activities; and g. Provide [Assignment: organization-defined system monitoring information] to [Assignment: organization-defined personnel or roles] [Selection (one or more): as needed; [Assignment: organization-defined frequency]].
SI-4(25)	P	O/S	**System Monitoring | Personally Identifiable Information Monitoring:** Employ automated mechanisms to monitor: (a) For unauthorized access or usage of personally identifiable information; and (b) The collection, creation, accuracy, relevance, timeliness, impact, and completeness of personally identifiable information.
SI-6	J	S	**Security and Privacy Function Verification**: a. Verify the correct operation of [Assignment: organization-defined security and privacy functions]; b. Perform this verification [Selection (one or more): [Assignment: organization-defined system transitional states]; upon command by user with appropriate privilege; [Assignment: organization-defined frequency]; c. Notify [Assignment: organization-defined personnel or roles] of failed security and privacy verification tests; and d. [Selection (one or more): Shut the system down; Restart the system; [Assignment: organization-defined alternative action(s)]] when anomalies are discovered.
SI-10	S	S	**Information input validation**: Check the validity of [Assignment: organization-defined information inputs].
SI-20	P	O/S	**De-Identification**: Remove personally identifiable information from datasets.
SI-20(1)	P	O/S	**De-Identification | Collection:** De-identify the dataset upon collection by not collecting personally identifiable information.

Table Appendix C (b): Example policy levels for the controls in Table A used in the case study.

Control ID	Metric ID	Example Metric name	Example Control SLO level					
			0	1	2	3	4	5
RA-5	m1	vuln scanning freq	none	weekly	daily	continuous	continuous	-
	m2	vuln remediation ratio	x<=20%	20%<x<=30%	30%<x<=50%	50%<x<=70%	x>70%	-
	m3	vuln feed update freq	none	weekly	daily	daily	continuous	-
AC-3	m4	identity assurance	No	Yes	Yes	Yes	-	-
	m5	enforcement false positive rate	x>40%	30%<x<=40%	10%<x<=30%	x<=10%	-	-
	m6	shared account percentage	x>50%	20%<x<=50%	20%<x<=50%	x<=20%	-	-
	m7	access policies verification	No	Yes	Yes	Yes	-	-
SI-20(4)	m8	de-identification technique used	none	mask	replace	hash	encrypt	remove
	m9	data utility loss	x>=15%	10%=<x<15%	2%<x<10%	x<=2%	x<=2%	x<=2%
	m10	de-identification only wrt marked identifiers	No	Yes	Yes	Yes	Yes	Yes
AC-6	m11	isolation of domains applied	No	Yes	-	-	-	-
CA-8	m12	pentesting tool used	v1.0	v2.0	v3.0	v3.4	v3.4	-
	m13	pentesting frequency	none	continuos	continuos	continuos	continuos	-
	m14	pentesting coverage	x<=10%	10%<x<=30%	30%<x<=60%	60%<x<=70%	x>70%	-
SC-5	m15	usage of boundary protection	No	Yes	-	-	-	-
	m16	usage of redundancy techniques	No	Yes	-	-	-	-
	m17	network capacity optimisation	No	Yes	-	-	-	-
SC-16	m18	Intrusion Detection used	No	Yes	Yes	Yes	Yes	-
	m19	app level AC monitoring	No	Yes	Yes	Yes	Yes	-
	m20	internal monitors used	No	Yes	Yes	Yes	Yes	-
	m21	risk assesm freq	none	daily	weekly	monthly	continuos	-
	m22	high severity risks treating ratio	x<=10%	10%<x<=30%	30%<x<=50%	x>50%	x>50%	-
	m23	freq of monitoring info analysis	none	daily	weekly	weekly	weekly	-

Control ID	Metric ID	Example Metric name	Example Control SLO level					
			0	1	2	3	4	5
SI-4	m24	input validation applied	No	Yes	-	-	-	-
SI-4(25)	m25	security attributes of data in transit identified	No	Yes	-	-	-	-
	m26	privacy attributes of data in transit identified	No	Yes	-	-	-	-
SI-6	m27	frequency of security verification of defined critical functions.	none	at release	continuous	-	-	-
	m28	test results notification	No	Yes	Yes	-	-	-
	m29	anomaly response applied	No	Yes	Yes	-	-	-
SI-10	m30	intrusion detection used	No	Yes	Yes	Yes	-	-
	m31	PII monitoring frequency	none	monthly	daily	daily	-	-
	m32	correctness of PII verified	No	Yes	Yes	Yes	-	-
	m33	timeliness of PII (days till PII consent due date)	Obsolete	x < 7	30 > x >= 7	x > = 30	-	-
	m34	consent of PII verified	No	Yes	Yes	Yes	-	-
SI-20	m35	result of removal of PII	Fail	Pass	-	-	-	-
SI-20(1)	m36	PII collection procedure verification used	No	Yes	-	-	-	-