# Applicability of Systems and Software Quality Engineering Methods and Models to Information Security in Cloud Computing Services

by

Jonathan ROY

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLEMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, MAY 12, 2021

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

## BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Witold Suryn, Thesis Supervisor
Department of Software Engineering and Information Technology, École de technologie supérieure

Mr. Michel Kadoch, President of the Board of Examiners
Department of Electrical Engineering, École de technologie supérieure

Mr. Claude Y. Laporte, Member of the jury
Department of Software Engineering and Information Technology, École de technologie supérieure

Mr. François Coallier, Member of the jury
Department of Software Engineering and Information Technology, École de technologie supérieure

Mrs. Ebba Þóra Hvannberg, External Evaluator
Department of Computer Science, University of Iceland

THIS THESIS WAS PRENSENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

MAY 6, 2021

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

# ACKNOWLEDGMENTS

I would first like to thank my supervisor, Professor Witold Suryn, for inspiring and encouraging me to pursue my Ph.D. and whose advice and guidance were invaluable during these years of research. He taught me to focus on the essentials, and his presence and insights greatly contributed to the improvement of the quality of my research.

I would like to thank my wife Karolina: You have provided all the support I could have asked for throughout this entire process and made countless sacrifices to help me complete my research. For this I am grateful.

Finally, I would also like to thank my mother Lise, my father Jean-Guy, and my brother Benjamin. I am forever grateful for your patience and encouragement.

**Applicabilité des méthodes et modèles d'ingénierie de la qualité des systèmes et logiciels à la sécurité de l'information dans les services informatiques en nuage**

Jonathan ROY

## RÉSUMÉ

Puisque l'industrie, motivée par les avantages potentiels des caractéristiques propres aux solutions infonuagiques, continue d'adopter de manière agressive des services informatiques en nuage, et que les experts de l'externalisation des technologies de l'information (TI) considèrent que la sécurité est l'exigence non fonctionnelle la plus importante à laquelle doivent se conformer les services informatiques en nuage, un nombre croissant d'ingénieurs en systèmes et logiciels doivent se tourner vers l'ingénierie de la qualité des systèmes d'information (SI) en nuage pour atténuer ou éviter les risques potentiels pour les utilisateurs ou les intervenants. Il s'avère que l'application de l'ingénierie de la qualité à l'élaboration du SI pendant les activités de définition des exigences est une étape importante et contribue à l'établissement d'un SI de qualité. En outre, l'application de l'ingénierie de la qualité nécessite l'utilisation d'un modèle ayant la capacité de soutenir à la fois la définition des exigences de qualité et leur évaluation ultérieure. Cependant, la mesure dans laquelle les modèles de qualité ISO/IEC 25000 sont applicables à la sécurité de l'information dans les services informatiques en nuage n'a pas été établie.

En conséquence, cette thèse propose l'utilisation du cadre d'exigences de qualité ISO/IEC 25030 comme approche systématique pour répondre à la question de recherche « Dans quelle mesure les modèles de qualité ISO/IEC 25000 prennent-ils en charge la définition des exigences de qualité liées à la sécurité de l'information dans les services informatiques en nuage? ». Nous utilisons ici des pratiques scientifiques d'établissement de plan de recherche sur les SI et étendons l'applicabilité du cadre d'exigences de qualité ISO/IEC 25030 à la sécurité de l'information dans le cadre des services informatiques en nuage. Nous utilisons également ce cadre élargi pour répondre à la question de recherche et illustrer son applicabilité à la définition des exigences de qualité des SI en nuage.

Les résultats de cette thèse montrent que, d'une part, les modèles de qualité ISO/IEC 25000 ne peuvent pas atténuer de manière adéquate les principales menaces pesant sur les services informatiques en nuage. D'autre part, l'application de ce cadre élargi démontre que ces modèles de qualité peuvent être adaptés pour la définition des exigences de qualité des SI en nuage dans le but d'atténuer ou d'éviter les risques potentiels pour les utilisateurs ou les intervenants.

**Mots-clés :** qualité des systèmes et des logiciels, ingénierie de la qualité, ingénierie des exigences de qualité, sécurité de l'information, informatique en nuage, services informatiques en nuage

# Applicability of Systems and Software Quality Engineering Methods and Models to Information Security in Cloud Computing Services

Jonathan ROY

## ABSTRACT

Industry continues to aggressively adopt cloud computing services, motivated by the potential benefits of their native characteristics. Additionally, information technology (IT) outsourcing experts consider security to be the most important non-functional requirement of cloud computing services. Consequently, a growing number of systems and software engineers must confront the quality engineering of cloud-based information systems (IS) to mitigate or avoid potential user or stakeholder risks. It has been well-established that the early application of quality engineering to IS development during requirement activities is a major milestone and contributor to building high quality IS. Furthermore, the application of quality engineering requires the use of a quality model with the capacity to support both the definition of quality requirements and their subsequent evaluation. However, the extent to which ISO/IEC 25000 quality models are applicable to information security in cloud computing services has not been identified.

Accordingly, this thesis proposes the use of the ISO/IEC 25030 quality requirements framework as a systematic approach to answer the research question “To what extent do ISO/IEC 25000 quality models support the definition of quality requirements related to information security in cloud computing services?” Here, we use IS research design science practices to evaluate and extend the applicability of the ISO/IEC 25030 quality requirements framework to information security in cloud computing services. We also use the extended framework to answer the research question while illustrating its applicability to the quality requirement definition of cloud-based IS.

Results from this thesis show that on the one hand ISO/IEC 25000 quality models cannot adequately mitigate the top threats to cloud computing services. On the other hand, the application of the extended framework demonstrates that these quality models can be tailored to the quality requirements definition of cloud-based IS for the purposes of mitigating or avoiding potential user or stakeholder risks.

**Keywords:** systems and software quality, quality engineering, quality requirements engineering, information security, cloud computing, cloud computing services

## TABLE OF CONTENTS

# LIST OF TABLES

Page

# LIST OF FIGURES

Page

# LIST OF ABREVIATIONS

| | |
|---|---|
| CCM | Cloud Controls Matrix |
| CSA | Cloud Security Alliance |
| CSP | Constraint Satisfaction Problem |
| CQ | Competency Question |
| DQR | Data Quality Requirement |
| FOL | First Order Logic |
| IaaS | Infrastructure as a Service |
| ICT | Information and Communications Technology |
| IEC | International Electrotechnical Commission |
| IEEE | Institute of Electrical and Electronics Engineers |
| IS | Information System |
| ISO | International Organization for Standardization |
| IT | Information Technology |
| JTC | Joint Technical Committee |
| KPI | Key Performance Indicator |
| NIST | National Institute of Standards and Technology |
| SaaS | Software as a Service |
| SC | Subcommittee |
| PaaS | Platform as a Service |
| PII | Personally Identifiable Information |
| PQR | Product Quality Requirement |

| | |
|---|---|
| QME | Quality Measure Element |
| QoS | Quality of Service |
| RE | Requirement Engineering |
| UML | Unified Modeling Language |
| VLAN | Virtual Local Area Network |
| VM | Virtual Machine |
| VXLAN | Virtual Extensible Local Area Network |
| WG | Working Group |

# INTRODUCTION

Industry continues to aggressively adopt cloud computing services, motivated by the potential benefits of their native characteristics. Consequently, growing numbers of systems and software engineers must confront the quality engineering of cloud-based information systems (IS). However, researchers have argued that ISO/IEC 25000 quality models used in the application of quality engineering do not cover cloud computing service-specific quality characteristics (Choi & Jeong, 2014) or do not address cloud computing service-specific technical quality characteristics in sufficient detail due to their generic nature (Wollersheim & Krcmar, 2014). On the other hand, the ISO/IEC 25030 quality requirements framework that guides the application of ISO/IEC 25000 quality models is not used by researchers to evaluate their applicability in cloud computing services. Moreover, ISO/IEC experts highlight the need to present guidelines and examples on how to apply the ISO/IEC 25030 quality requirements framework to cloud computing services (Nakajima, 2020), and IT outsourcing experts consider security the most important non-functional requirement of cloud computing services (Schlauderer & Overhage, 2015). Additionally, industry security experts have also expressed the need to focus on threats, vulnerabilities, and risks, which often result from the "shared, on-demand nature of cloud computing" (Alliance, 2019).

The objective of this thesis is to evaluate and extend the applicability of the ISO/IEC 25030 quality requirements framework to information security in cloud computing services. Here, we use IS research design science practices: to evaluate the applicability of the ISO/IEC 25030 quality requirements framework to information security in cloud computing services; to construct an ontology that integrates the frameworks required to extend its applicability; to evaluate the applicability of the extended framework by its application to answer the research question "To what extent do ISO/IEC 25000 quality models support the definition of quality requirements related to information security in cloud computing services?"; and to illustrate its applicability to information security in a multi-tenant virtualized infrastructure. Finally, we discuss how its addition to the quality engineering knowledgebase can potentially support systems and software quality engineers with the quality engineering of cloud-based IS as well as the ISO/IEC JTC 1/SC 7/WG 6 on systems and software product quality along with the further development of ISO/IEC 25000 quality models and quality measures.

# CHAPTER 1

# PRESENTATION OF THE RESEARCH

## 1.1 Background and motivation

In cloud computing environments, information systems (IS) are no longer static and limited by dedicated hardware and finite resources. They are dynamic virtualized entities capable of rapid computing resource scalability to match demand as much as possible. The capacity required to develop, test, deploy, and deliver IS is provided as services through a pay-per-use business model, on-demand, over the web, and with minimal human interactions. Contrary to on-premises IS that operate in specific and controlled locations, processing, communication, and storage tasks are ceded to third parties that comprise of geographically distributed and shared resources. This introduces an additional level of abstraction and complexity that constitutes a trade-off that reduces on-premises IS constraints while benefiting from acquisition or delivery of services through the cloud computing paradigm. The native characteristics of cloud computing environments and the diversity of the cloud computing services offered (ranging from consumer control of the entire software stack to the application of domain-specific platforms (Armbrust et al., 2009)) constitute a conceptual shift from conventional computing environments.

From a systems and software quality engineering perspective, empirical studies have shown that locating and repairing problems after delivery is significantly more expensive than locating and repairing problems during the design phase (Boehm & Basili, 2005). In some cases, this could involve redesigning, reimplementing, retesting, and revalidating the entire IS. Hence, the early application of quality engineering to IS development during, for example, the requirement activities (i.e., the discovery, analysis, and architectural design activities) is a major milestone and contributor in building high quality IS (Suryn, 2014). This allows systems and software quality engineers and designers to study design trade-offs and assess their impact on the overall quality of an IS. Therefore, during early stages of systems and software quality engineering, systems and software quality engineers must be able to define cloud-based IS quality requirements.

The application of systems and software quality engineering requires the use of a quality model with the capacity to support both the definition of quality requirements and their evaluation. Moreover, measurement approaches should follow standardized documentation linked to the selected quality model. Similarly, the quality evaluation method should be selected and executed in connection with the quality model itself (Suryn, 2014). Therefore, quality requirement frameworks and quality models used under systems and software quality engineering practices, such as the ISO/IEC 25030 quality requirements framework, which includes ISO/IEC 25010, ISO/IEC 25011, and ISO/IEC 25012 quality models, require the capacity to support definitions of cloud-based IS quality requirements and their subsequent evaluation.

However, researchers argue that these quality models either do not cover specific quality characteristics and technical aspects of cloud computing services or do not address them in sufficient detail due to their generic nature (Choi & Jeong, 2014; Wollersheim & Krcmar, 2014). On the other hand, the associated ISO/IEC 25030 quality requirements framework is not used by researchers as a systematic approach to identify applicable and missing quality characteristics and measures in cloud computing services. Moreover, in 2018, systems and software experts from the ISO/IEC JTC 1/SC 7 subcommittee working group (WG) on software product and system quality (i.e., ISO/IEC JTC 1/SC 7/WG 6) mandated the implementation of a study group on the future direction of the ISO/IEC 25000 series which highlighted the need to provide guidelines and examples on how to apply ISO/IEC 25030 quality requirement framework methods and models to cloud computing services (Nakajima, 2020). Additionally, a select group of IT outsourcing experts surveyed on requirements and cloud computing services determined that security was the most important non-functional requirement (Schlauderer & Overhage, 2015). As for industry security experts, they expressed the need to focus on threats, vulnerabilities, and risks which often result from the “shared, on-demand nature of cloud computing” (Alliance, 2019).

## 1.2 Research objectives

For this research project, our research objectives must address the following research question:

- To what extent do ISO/IEC 25000 quality models support the definition of quality requirements related to information security in cloud computing services?

To answer the research question, our goal is to use the ISO/IEC 25030 quality requirement framework as a systematic approach by which to evaluate the applicability of ISO/IEC 25000 quality models to information security in cloud computing services. However, the applicability of the framework to information security in cloud computing services must be first evaluated and, if required, the framework must be extended before it can be applied to answer the research question.

Consequently, our research consisted of four main objectives:

1. Evaluate the applicability of the ISO/IEC 25030 quality requirement framework for the definition of quality requirements related to information security in cloud computing services;
2. If required, build extensions to the ISO/IEC 25030 quality requirement framework for the definition of quality requirements related to information security in cloud computing services;
3. Evaluate the applicability of the extended quality requirement framework through its application to answer the research question;
4. Illustrate the application of the extended quality requirement framework to information security in cloud computing services.

## 1.3 Research contributions and foreseen benefits

The additional relevant value of this research can be described as follows:

1. The research provides the systems and software industry as well as the research community an ontology that integrates the concepts of frameworks which extend the applicability of the ISO/IEC 25030 quality requirements framework to information security in cloud computing services;
2. The research answers the following research question "To what extent do ISO/IEC 25000 quality models support the definition of quality requirements related to information security in cloud computing services?";
3. The research results will be conferred to the ISO/IEC JTC 1/SC 7/WG 6 once published;
4. The resultant extended framework can potentially be used to:
    a. Support the quality requirements definition of cloud-based IS;
    b. Support the further development of ISO/IEC 25000 quality models.

## 1.4 Research limitations

Our research project is limited to the construction and evaluation of an ontology that integrates the frameworks required to extend the applicability of the ISO/IEC 25030 quality requirements framework to information security in cloud computing services.

It is also important to note that the ISO/IEC 25000 series include ISO/IEC 25011 service quality models. However, the associated ISO/IEC 25025 quality measures at the time of this research are under development. Consequently, although we include the concepts extracted from the published ISO/IEC 25011 (ISO/IEC, 2017) quality model in the construction of the proposed integrated ontology, the evaluation of the quality model and the associated quality measures are not covered in this research project.

## 1.5 Thesis organization

Chapter 2 presents a literature review on the applicability of ISO/IEC 25000 quality models to cloud computing services. Chapter 3 presents the research methodology and design for this research project. Chapter 4 presents the evaluation of the applicability of the ISO/IEC 25030 quality requirements framework to information security in cloud computing services. Chapter 5 presents the construction of an ontology that integrates the frameworks required to extend the applicability of the ISO/IEC 25030 quality requirements framework to information security in cloud computing services. Chapter 6 presents the evaluation of the applicability of the extended quality requirements framework through its application to answer the research question as presented above. Chapter 7 presents an illustrative example of the application of the extended quality requirements framework to the case of information security in a multi-tenant virtualized infrastructure.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

This chapter presents our literature review. Section 2.2 presents a review on the applicability of ISO/IEC 25000 quality models in cloud computing services. It is important to note that ISO/IEC 25010 has replaced ISO/IEC 9126. However, considering that the research on the applicability of ISO/IEC standards in cloud computing services started before the publication of the ISO/IEC 25000 series, articles that refer to ISO/IEC 9126 were included in this review. Section 2.3 presents a review on proposed cloud computing services quality models. Additionally, in 2018 the ISO/IEC JTC 1/SC 7/WG 6 on software product and system quality mandated the implementation of a study group on the future direction of the ISO/IEC 25000 series to investigate the applicability of the quality requirements framework under the current technological context, which includes cloud computing services. Therefore, section 2.4 presents the objectives of this study group. Finally, in section 2.5 we present our findings.

## 2.2 Applicability of ISO/IEC 25000 quality models in cloud computing services

Like the following studies, Choi & Jeong (2014) proposed a method to evaluate software-as-a-service (SaaS) quality of service (QoS). The method is based on priority weight, set by the user, for each quality attribute. To construct the quality attribute matrix required for the evaluation, they extracted six quality attributes with different characteristics from existing web service quality standards and the INSPIRE Network Services Performance Guidelines (Infrastructure for Spatial Information in Europe). In support of their approach, they argued that standards such as ISO/IEC 9126 do not effectively evaluate quality aspects in cloud computing environments due to the differences in conventional computing environments. However, they did not offer an analysis as to what led to this conclusion. They also pointed out that neither ISO/IEC 9126-1 nor ISO/IEC 14598-1 have processes from which to specify the quality model and the evaluation.

Wollersheim & Krcmar (2014) investigated quality analysis approaches for cloud computing services and structured their findings following the dimensions proposed by Grönroos (2007), namely, the technical quality of the outcome, the functional quality of the process, and the image of the service provider. According to Wollersheim & Krcmar (2014), ISO/IEC 25010 addresses technical and functional quality dimensions but ignores criteria covering image dimensions of service quality. However, their study did not evaluate the ISO/IEC 25011 service quality models. They also pointed out that some cloud computing service-specific technical quality aspects have not been addressed in detail because of the generic nature of the standard.

Jeong & Hong (2013) proposed a quality model and a method for the quality evaluation of SaaS. This study used ISO/IEC 9126 for the set of quality characteristics of the software product quality model. Additionally, it used ITIL and COBIT for characteristics related to the management and improvement of QoS. A complementation and prioritization of characteristics has also been conducted by experts within the SaaS domain. To validate their method, an evaluation process based on the ISO/IEC 14598 standard was developed and tailored for SaaS context. A case study was then designed to validate their method using a comparison between the proposed method and MEDE-PROS, a conventional method, with the objective of identifying similarities, differences, and limitations. The aim of MEDE-PROS is to support evaluators in their evaluation of software products from the perspective of the end user. It is based on ISO/IEC 9126 quality characteristics and ISO/IEC 14598 evaluation process. Their comparison showed data convergence relative to software product quality characteristics and data divergence relative to service level and support quality characteristics. According to Jeong & Hong (2013), this divergence was due to the fact that ISO/IEC 9126 covers quality characteristics that are similar between conventional software products and SaaS but does not cover SaaS specific quality characteristics. Therefore, they concluded that conventional methods cannot effectively evaluate SaaS quality.

Instead of directly applying ISO/IEC standards, certain studies have chosen to incorporate aspects or concepts of such standards in the development of their own proprietary evaluation models. The study by Lee, Lee, Cheun & Kim (2009) proposed a quality model to evaluate SaaS quality. Through content analysis of industry and academic literature relevant to cloud computing and, specifically, SaaS, they identified six of what they described as key SaaS characteristics, namely, reusability, availability, data management by providers, scalability, customizability, and pay-per-use. Based on a mapping between key SaaS characteristics and ISO/IEC 9126 quality characteristics, two quality characteristics were expanded upon, namely, efficiency and reliability, and three new characteristics were introduced, namely, reusability, availability, and scalability. Ten metrics were proposed to evaluate quality characteristics while the assessment of their usefulness and practicability was based on IEEE 1061, the IEEE Standard for a Software Quality Metrics Methodology.

Through industry and academic literature content analysis, Schlauderer & Overhage (2015) identified 39 different requirements for the suitability evaluation of a cloud service provider. They borrowed the quality model structure of the ISO/IEC 9126 standard to create a three-level hierarchy, namely, assessment criteria (evaluation topic), assessment properties (provider characteristics), and measurable items (requirements). ISO/IEC 9126 standard measures for availability and efficiency were also proposed to specify the level of required quality as part of a service contract between the service consumer and provider. Among the 39 proposed requirements, only three were covered by ISO/IEC 9126 standard measures. However, the authors of this study did not propose measures to cover the remaining requirements. To evaluate the relevance of the requirements, the authors surveyed a select group of IT outsourcing experts. Results of this survey revealed that all identified requirements were relevant, and that security was the most important non-functional requirement.

One study used the ISO/IEC 25010 standard to investigate the impact of cloud environments on software engineering. Barbosa & Charão (2012) investigated the impact of pay-per-use cloud environments on software requirements engineering. For their purposes, the quality characteristic performance efficiency was put into context. According to their study, in conventional computing, the quality characteristic performance efficiency is often used to guide software engineers in the design of software products as it relates to response time quality requirements and fixed resource operations under a specified load. As indicated by Barbosa & Charão (2012), service providers, with respect to software operations in a pay-per-use cloud environment, must optimize the total resources consumed by software offered “as-a-service” to facilitate greater profit.

Another study chose to directly apply the ISO/IEC 25010 quality model in the development of their evaluation model. Wen & Dong (2013) defined a quality model and metrics from the perspective of the SaaS platform, application providers, and consumers separately. The perspectives were identified based on the analysis of SaaS architecture proposed by the Cloud Computing Use Cases discussion group. Three quality aspects were also defined, namely, software quality, QoS, and security. The proposed quality model consists of mapping between different perspectives and quality aspects. However, the analysis and methodology used to identify perspectives and quality aspects as well as the proposed quality model were not provided. The authors referred to ISO/IEC 25010 quality models for the definition and evaluation of software quality aspects of SaaS from an application provider and consumer perspective. However, its use and capacity to support definitions and the evaluation of quality in the context of SaaS were not demonstrated. To cover both QoS and security aspects, the authors introduced ISO/IEC 27001.

Other studies have concentrated on specific quality characteristics and their evaluation. Abdeladim, Baina & Baina (2014) focused on elasticity and scalability and proposed a quality model for their evaluation as well as their impact on cloud service QoS. Additionally, another study (Villalpando, April & Abran, 2014) focused on a specific application and quality characteristic and proposed a method that integrates ISO/IEC 25010 quality models for performance analysis of big data applications from a software engineering perspective. Based on a previous study (Bautista, Abran & April, 2012) that proposed a performance measurement framework for cloud computing (PMFCC), a performance analysis scheme was defined to establish a set of performance characteristics and possible outcomes for service requests. This scheme allows PMFCC to map potential outcomes onto quality characteristics and sub-characteristics extracted from ISO/IEC 25010 quality models. Performance measures are then collected and mapped onto quality characteristics, and their associated formulae are defined using the ISO/IEC 25023 quality measures. The formulae were also adapted to the context of the application. Using statistical methods, they then identified relationships between various base measures and performance quality characteristics. Although the experiments were limited to the Hadoop framework and the MapReduce programming model, the authors of this study provided a concrete example on the use of the ISO/IEC 25010 quality models for quality evaluation of software operating in cloud environments.

## 2.3 Cloud computing services quality models

To address the need to define and evaluate cloud computing service quality, other studies have proposed cloud computing service-specific quality models that do not make use of ISO/IEC 25000 quality models and quality measures.

Garg, Versteeg & Buyya (2011) proposed a framework called SMICloud to measure QoS for cloud services. The approach is based on Service Measurement Index (SMI) attributes that are based on the Cloud Service Measurement Index Consortium (CSMIC). The SMI specifies Key Performance Indicators (KPI) applicable to businesses to standardize methods to measure and compare business services. According to Garg et al. (2011), the SMI provides the high-level attributes needed by the customer to select a cloud service provider. Although the SMI specified high-level attributes, namely, accountability, agility, cost, performance, assurance, security, privacy, and usability, it did not define any metrics. The authors also argued that cloud services can be evaluated based on qualitative and quantitative KPIs, that SMI KPI definitions can vary according to the service, and certain parameters are dependent on customer applications while others remain independent. According to Garg et al. (2011), accurately defining SMI values for a provider is therefore complex. Accordingly, the proposed quality model is based on the most important quantifiable KPIs in the context of IaaS together with their definitions and associated metrics. However, the authors did not consider security as being a quantifiable KPI in the context of IaaS. An assessment of the usefulness and practicability of the metrics based on IEEE 1061 was also provided but not demonstrated.

In advocating CLOUDQUAL, Zheng, Martin, Brohman & Da Xu (2014) proposed a quality model that includes metrics for the evaluation of cloud computing services. Six quality dimensions, namely, usability, availability, reliability, responsiveness, security, and elasticity, were proposed as well as six quality metrics. This quality model was inspired by SERVQUAL (Parasuraman, Zeithaml & Berry, 1988) as well as an e-service perspective remodel of this instrument by Swaid & Wigand (2009), wherein quality is defined as the extent to which the Internet assists in the effective delivery of products and/or services. SERVQUAL was developed to measure the service quality of traditional services. According to the study by Zheng et al. (2014), the proposed model by Swaid is an effective tool in which to evaluate e-service quality. However, Zheng's study argued that the quality dimensions of the proposed SERVQUAL model were entirely subjective and did not offer quality measurements and, consequently, could not be applied to cloud services. Accordingly, they argued that a cloud service quality model must be objective, computable, and verifiable. This would allow cloud providers to gauge the QoS delivered and allow cloud consumers to validate the QoS received. Zheng's study conducted an empirical case study to demonstrate the capacity of the quality model in evaluating cloud service quality. It showed that the proposed quality model can effectively evaluate cloud quality. Empirical data were used to validate the proposed quality metrics in their capacity to differentiate cloud service quality based on IEEE 1061.

## 2.4 ISO/IEC 25000 future direction

In 2018, the ISO/IEC JTC 1/SC 7/WG 6 on software product and system quality mandated the establishment of a study group on the future direction of ISO/IEC 25000. In their report (Nakajima, 2020), the study group highlighted the need to present guidelines and examples on how to apply the standard series to the current technological context. The objective is to support systems and software quality engineers with an interpretation of the ISO/IEC 25030 quality requirement framework and the quality evaluation processes when they are applied to technology such as cloud computing services (e.g., SaaS, PaaS, IaaS).

Additionally, a key objective of the ISO/IEC JTC 1/SC 7/WG 6 on software product and system quality was the development of a series of quality models for use with SaaS, PaaS, and IaaS. This objective involves the application of ISO/IEC 25000 series quality models (i.e., ISO/IEC 25010 software and system quality models, the ISO/IEC 25011 IT service quality model, and the ISO/IEC 25012 data quality model) for their development with a specialization that reflects major considerations of cloud computing services as defined by explicit ISO/IEC standards of cloud computing services, such as ISO/IEC 17788, ISO/IEC 17789, ISO/IEC 19086, etc. At the time of publication of this research project, the first quality models of this series, namely, ISO/IEC 25052-1 quality models for use with SaaS, is under development and is only available as working document. Figure 2.1 shows the relationships between ISO/IEC 25000 series quality models and standards related to cloud computing services for the development of ISO/IEC 25052-1 quality models for use with SaaS.

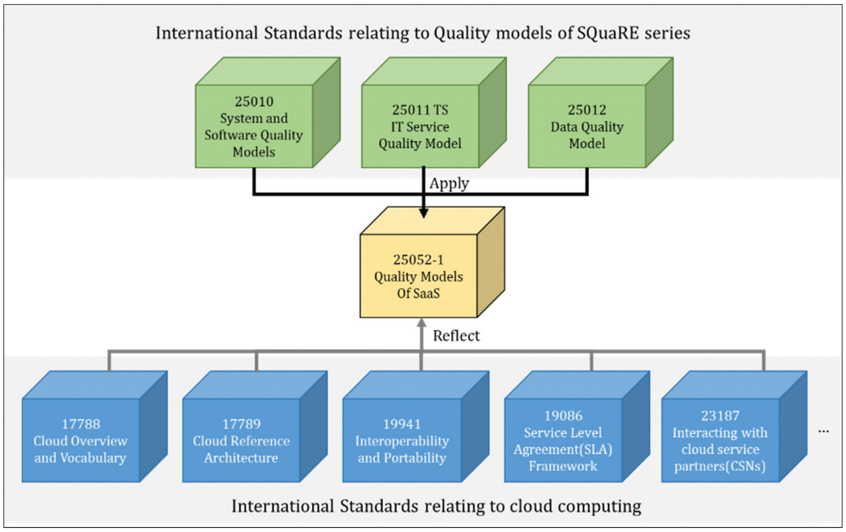

International Standards relating to Quality models of SQuaRE series
25010 System and Software Quality Models
25011 TS IT Service Quality Model
25012 Data Quality Model
Apply
25052-1 Quality Models Of SaaS
Reflect
17788 Cloud Overview and Vocabulary
17789 Cloud Reference Architecture
19941 Interoperability and Portability
19086 Service Level Agreement(SLA) Framework
23187 Interacting with cloud service partners(CSNs)
...
International Standards relating to cloud computing

Figure 2.1 Relationships between ISO/IEC 25000 series
quality models and standards
related to cloud computing services
taken from ISO/IEC 25052-1 (ISO/IEC, 2020)

## 2.5 Cloud computing services security

When surveyed on requirements and cloud computing services, a select group of IT outsourcing experts determined that security was the most important non-functional requirement (Schlauderer & Overhage, 2015). Consequently, we reviewed the security characteristics and measures previously discussed in the preceding section.

Wollersheim & Krcmar (2014) argued that some specific technical quality "aspects" related to cloud computing services, such as those outlined by Repschlaeger, Wind, Zarnekow & Turowski (2013) or Benlian, Koufaris & Hess (2011), have not been addressed in detail because of the generic nature of the ISO/IEC 25010 standard. Such technical quality "aspects" include cloud computing service security. More specifically, Repschlaeger et al. (2013) included "IT Security and Compliance" in their model to evaluate cloud computing service providers as a target dimension with "Data Center Protection", "Network Protection", "Operations Protection", and "IT Compliance" being abstract requirements and "Building Safety", "Connection Opportunities", "Communication Security", "Application Access (Identity Management)", "Application Protection", "Data Center Location", and "Data Protection" being evaluation criteria. However, their evaluation proposed no measures.

Wen & Dong (2013) proposed a quality model for the quality evaluation of SaaS from the perspective of the service provider and customer separately. The proposed quality model for use with SaaS includes ISO/IEC 25010 quality characteristics as well as what Wen & Dong (2013) refers to as the "security metrics" from ISO/IEC 27001. However, what is referred by Wen & Dong (2013) as security metrics, such as physical and environmental security and communication security (network security), are in fact ISO/IEC 27001 and ISO/IEC 27002 security clauses.

With respect to ISO/IEC 25052-1 quality models for use with SaaS, at the time of publication of this research project the working document indicates that the quality models include security characteristics and sub-characteristics obtained from ISO/IEC 25010 system and software quality models and ISO/IEC 25011 IT service quality models, namely, service security, confidentiality, integrity, traceability, non-repudiation, accountability, and authenticity. Additional quality sub-characteristics which were not obtained from ISO/IEC 25000 quality models also included, namely, isolation, PII protection conformance, and security responsibility clarity. However, as previously discussed, these quality models remain under development at the time publication of this research project. As a result, there is no study available in the literature on the applicability of the quality models and associated quality measures for the quality evaluation of SaaS.

We also observed that quality models for cloud computing services have been proposed that do not make use of ISO/IEC 25000 quality models and quality measures. Garg et al. (2011) proposed a quality model for use with IaaS based on SMI KPIs. The proposed model includes the attribute "security and privacy". Garg et al. (2011) indicated that this attribute is multi-dimensional in nature while it also includes many attributes, such as privacy, data loss, and integrity. Although their study defined quantifiable KPIs and proposed metrics for attributes, no KPIs or metrics were proposed for the attribute "security and privacy".

Additionally, Zheng et al. (2014) argued that the quality model for use with IaaS proposed by Garg et al. (2011) does not define important quality dimensions, such as "security". As a result, Zheng et al. (2014) proposed CLOUDQUAL, a quality model for cloud computing services, which specifies six quality dimensions, including "security" and its associated quality metric, where "security" is defined as "the assurance that cloud services are free from viruses, intrusions, spyware, attacks, and other security vulnerabilities that could put them at risk". Zheng et al. (2014) also emphasized the necessity of evaluating the level of cloud computing service security from an end-user perspective by applying historical information that is publicly available or obtained from a third-party. Furthermore, Zheng et al. (2014) indicated that the objective of the proposed security dimension and its associated quality metric used for the evaluation of cloud computing services is not to identify security issues nor to enforce security mechanisms for cloud computing services.

## 2.6 Industry information security in cloud computing

Additionally, we conducted searches to identify expert reports from widely recognized organizations on information security in cloud computing. We found two relevant initiatives: The CSA report on the top threats to cloud computing and the European Union Agency for Cybersecurity (ENISA) report on cloud computing risks.

The CSA report provides information on the top security risks and threats to cloud computing, which are unique or highly influenced by cloud computing characteristics and reflects the current consensus among security experts in the CSA community. More specifically, the report focuses on threats, vulnerabilities, and risks that often result from the “shared, on-demand nature of cloud computing” (Alliance, 2019). The report also includes applicable controls stipulated in the CSA Cloud Controls Matrix (CCM) which are mapped to standards, such as ISO/IEC 27001, ISO/IEC 27002, ISO/IEC 27017, and ISO/IEC 27018.

The ENISA report on cloud computing information security risks is the results of risk assessments by industry security experts based on ISO/IEC 27005. Although the report presents cloud computing information security risks, it does not include applicable controls to treat such risks.

## 2.7 Literature review results

The literature review on the applicability of ISO/IEC 25000 quality models in cloud computing services revealed that on the one hand researchers have argued that these models do not cover cloud computing service-specific quality characteristics or do not address cloud computing service-specific technical quality characteristics in sufficient detail due to their generic nature.

On the other hand, a key objective of the ISO/IEC JTC 1/SC 7/WG 6 on software product and system quality was the development of a series of quality models for use with SaaS, PaaS, and IaaS through the application of quality characteristics/sub-characteristics and quality measures from ISO/IEC 25000 quality models. This objective also entails the specialization of the models through the application of quality characteristics/sub-characteristics that reflect major considerations of cloud computing services as defined by specific ISO/IEC standards of cloud computing services. However, given that these quality models are still under development at the time of publication this research project, no study is currently available in the literature on their applicability for the quality evaluation of SaaS.

Our literature review also revealed that the ISO/IEC 25030 quality requirements framework that is guiding the application of the quality models has not been used by researchers to identify applicable and missing quality characteristics/sub-characteristics and quality measures in cloud computing services. Additionally, the ISO/IEC JTC 1/SC 7/WG 6 on the future direction of ISO/IEC 25000 highlighted the need for guidelines and examples that describe how the ISO/IEC 25030 quality requirements framework is applied to cloud computing services.

Our literature review also revealed that a select group of IT outsourcing experts surveyed on cloud computing service requirements determined that security was the most important non-functional requirement. However, although security characteristics are included in some of the proposed quality models for the quality evaluation of cloud computing services, measures have been proposed for their evaluation in only two cases. The first case is the proposed quality model for use with SaaS by Wen & Dong (2013), which includes ISO/IEC 25010 quality characteristics as well as what the authors refer to as "security metrics" from ISO/IEC 27001. However, what is referred by the authors as security metrics are in fact ISO/IEC 27001 and ISO/IEC 27002 security clauses. The second case is the proposed quality model for cloud computing services by Zheng et al. (2014), which includes security as a characteristic and its associated quality metric. However, in this case, the proposed quality metric is for the evaluation of cloud computing services from an end-user perspective only.

Finally, we found that industry security experts refer to the ISO/IEC 27005 information risk management framework to define cloud computing information security risks and to standards such as ISO/IEC 27001, ISO/IEC 27002, ISO/IEC 27017, and ISO/IEC 27018 for their treatment. We were unable to find industry expert reports on information security and cloud computing services that refer to the ISO/IEC 25000 series. We also found that industry security experts focus on threats, vulnerabilities, and risks related to the “shared, on-demand nature of cloud computing”.

## 2.8 Conclusions

Our literature review revealed that the applicability of ISO/IEC 25000 quality models in cloud computing services has clearly not been fully investigated prior to the proposed new quality models and measures. Given that the literature review also revealed that a select group of IT outsourcing experts surveyed on cloud computing service requirements determined that security was the most important non-functional requirement, this research project aims to address this issue by means of using the ISO/IEC 25030 quality requirements framework as a systematic approach to evaluate the applicability of ISO/IEC 25000 quality models to information security in cloud computing services.

# CHAPTER 3

# RESEARCH METHODOLOGY AND DESIGN

## 3.1 Introduction

The literature review on the applicability of ISO/IEC 25000 quality models in cloud computing services presented in the preceding chapter revealed that on the one hand researchers have argued that these models do not cover cloud computing service-specific quality characteristics or do not address cloud computing service-specific technical quality characteristics in sufficient detail due to their generic nature. On the other hand, it also revealed that there is a lack of a systematic approach to evaluate their applicability in cloud computing services. Moreover, the literature review revealed that the ISO/IEC 25030 quality requirements framework that guides the application of the quality models and quality measures has not been used by researchers to identify applicable and missing quality characteristics and measures in cloud computing services. Additionally, a select group of IT outsourcing experts surveyed on cloud computing service requirements determined that security was the most important non-functional requirement. It also revealed that ISO/IEC 27001 has been used to introduce security characteristics to a proposed new quality model. Finally, the ISO/IEC JTC 1/SC 7/WG 6 study group on the future direction of ISO/IEC 25000 highlighted the need for guidelines and examples on how to apply the ISO/IEC 25030 quality requirements framework to cloud computing services.

Consequently, this research project aims to address these issues and needs by means of using the ISO/IEC 25030 quality requirements framework as a systematic approach to evaluate the applicability of ISO/IEC 25000 quality models to information security in cloud computing services.

This chapter discusses the research objectives, the selected research methodology, the research project design, applicable research guidelines, and evaluation activities.

## 3.2 Research methodology

Because design science in IS research comprises the building and evaluation of the utility of an information technology (IT) artefact, such as the frameworks and models applied in IS development, we selected it as a research methodology.

This research project used the design science in IS research framework provided by Hevner, March, Park & Ram (2004). This conceptual framework provides guidelines for the application of design science in IS research. Figure 3.1 illustrates the design science in IS research framework.

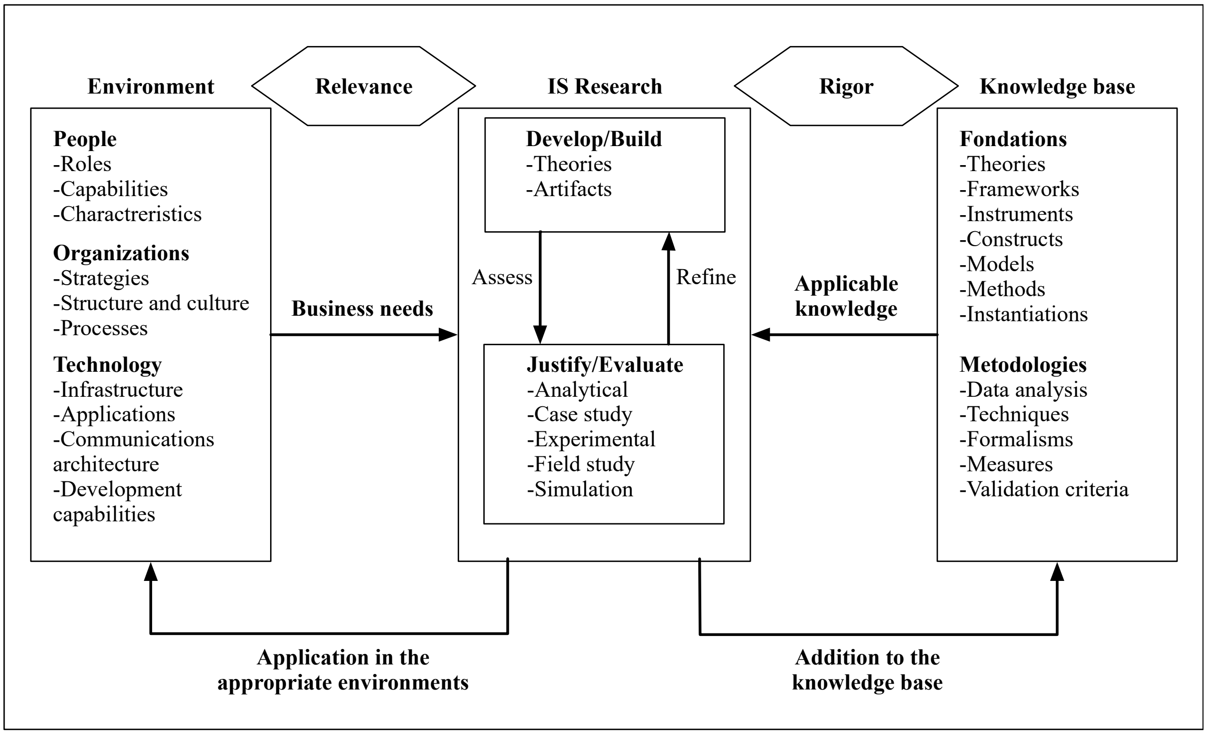


Figure 3.1 IS research framework adapted from Hevner et al. (2004)

This research project proceeds by mapping the key elements of the IS research framework provided by Hevner et al. (2004).

The *business needs* originate from the ISO/IEC JTC 1/SC 7/WG 6 on software product and system quality (i.e., *organization*), who mandated a study group on the future direction of ISO/IEC 25000. As discussed in the preceding chapter, this study group highlighted the need to support systems and software quality engineers (i.e., *people*) with their interpretation of the ISO/IEC 25030 quality requirement framework and quality models (i.e., *framework and model*) when they are applied to cloud computing services (i.e., *technology*) (details are provided in section 2.4). It also originates from findings that derived from our literature review on the applicability of ISO/IEC 25000 quality models (i.e., *model*) to cloud computing services (i.e., *technology*) (details are provided in section 2.5). The research objectives derived from the *business needs* (details are provided in section 3.2).

In Chapter 4, we first *evaluate* the applicability of the ISO/IEC 25030 quality requirement *framework* (i.e., *artifact*) to information security in cloud computing services (i.e., *environment* and *technology*) to determine whether it provides the *methods*, *models*, and guidance for the definition of quality requirements related to information security in cloud computing services, or if extensions to the *framework* need to be *built*.

In Chapter 5, we *build* an ontology (i.e., *artefact*) integrating the *frameworks* required to extend the applicability of the ISO/IEC 25030 quality requirements *framework* to information security in cloud computing services (i.e., *environment* and *technology*). The required *frameworks* are as follows:

1. ISO/IEC/IEEE 15288 system requirements definition framework;
2. NIST SP 800-160 system security requirements definition framework;
3. ISO/IEC 27005 information security risk management framework;
4. ISO/IEC 27017 information security controls for cloud computing services;
5. ISO/IEC 27018 information security controls for the protection of personally identifiable information (PII) in public cloud computing services acting as a PII processor.

We *build* the integrated ontology (i.e., *artefact*) using the ontological metamodeling *methodology* from Agrawal (2016)  and the semantic mapping *techniques* from Rizopoulos & Mçbrien (2005). Then, we *evaluate* its completeness.

In Chapter 6, we *evaluate* the integrated ontology (i.e., *artefact*) to determine if its application provides answers to the research question "To what extent do ISO/IEC 25000 quality models support the definition of quality requirements related to information security in cloud computing services?" (i.e., *model*, *environment*, and *technology*).

In Chapter 7, we *evaluate* the applicability of the integrated ontology (i.e., *artefact*) through means of using an illustrative example of its application to the case of information security in a multi-tenant virtualized infrastructure (i.e., *environment* and *technology*) to determine if it supports the definition of quality requirements related to controls that are applicable in mitigating threats and vulnerabilities to cloud computing services. Additionally, we determine if the integrated ontology fulfills our design objectives to avoid possible causes of IS failure, namely, lack of a systematic RE process, poor communication between people and lack of a shared understanding of the system being built, and poor management of the RE process (Lyytinen & Hirschheim, 1988; Macaulay, 1996). In the latter case, we determine if the integrated ontology fulfills our design objective, namely, to establish traceability of quality requirements and their sources.

## 3.3 Design science research guidelines

Hevner et al. (2004) established seven guidelines to support effective design science research. Table 3.1 provides the guidelines and their descriptions.

Table 3.1 Design science research guidelines extracted from Hevner et al. (2004)

| **Guideline** | **Description** |
|---|---|
| Guideline 1: Design as an Artifact | Design science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation. |
| Guideline 2: Problem Relevance | The objective of design science research is to develop technology-based solutions to important and relevant business problems. |
| Guideline 3: Design Evaluation | The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods. |
| Guideline 4: Research Contributions | Effective design science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies. |
| Guideline 5: Research Rigor | Design science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact. |
| Guideline 6: Design as a Search Process | The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment. |
| Guideline 7: Communication of Research | Design science research must be presented effectively both to technology-oriented as well as management-oriented audiences. |

We applied these seven guidelines to conduct and evaluate this research. Moreover, we proceeded by mapping the seven guidelines to this research project.

1. **Design as an artifact.** The integrated ontology produced during the construction phase meets the criteria of an artifact, as it embodies a construct (the definition of quality requirements related to information security in cloud computing services), a model (the ontology integrating the concepts of the ISO/IEC 15288, ISO/IEC 25030, ISO/IEC 27005, ISO/IEC 27017, ISO/IEC 27018, and NIST SP 800-160 frameworks), and instantiations (its application to answer our research question and the illustration of its application to the case of information security in a multi-tenant virtualized infrastructure).

2. **Problem relevance.** The ISO/IEC JTC 1/SC 7/WG 6 study group on the future direction of ISO/IEC 25000 highlighted the need to provide guidelines and examples on how to apply the ISO/IEC 25030 quality requirement framework to cloud computing services. Additionally, the literature review revealed that researchers have argued that ISO/IEC 25000 quality models neither cover specific quality characteristics and technical aspects of cloud computing services nor do they address them in sufficient detail due to their generic nature. Chapter 2 provides more detail.

3. **Design evaluation.** Artifact evaluation guidance in design science research (Sonnenberg & Vom Brocke, 2011) was used for the evaluation of the ISO/IEC 25030 quality requirements framework and quality models and for the applicability of the extended quality requirements framework and quality models to information security in cloud computing services. This chapter (Chapter 3) provides more detail.

4. **Research contributions.** This research project is intended to provide the systems and software industries as well as the research community an ontology that integrates ISO/IEC 15288, ISO/IEC 25030, ISO/IEC 27005, ISO/IEC 27017, ISO/IEC 27018, and NIST SP 800-160 frameworks, which will extend the applicability of the ISO/IEC 25030 quality requirements framework to information security in cloud computing services. Chapter 2 provides more detail.

5. **Research rigor.** Ontological metamodeling (Agrawal, 2016) and semantic mapping (Rizopoulos & Mçbrien, 2005) techniques are applied to the construction of the integrated ontology. Evaluation guidance in design science research is applied (Sonnenberg & Vom Brocke, 2011). Chapter 5 provides more detail.

6. **Design as a search process.** As per recommendations by Sonnenberg & Vom Brocke (2011), an ex ante evaluation was conducted prior to the construction of the artifact, and two ex post evaluations were conducted following the construction of the artifact, creating a feedback loop from the evaluation activities to the preceding design activities. This evaluation pattern aims to account for the emergent nature of design science artefacts. This chapter (Chapter 3) and chapters 4, 6, and 7 provide more detail.

7. **Communication of research.** Research outcomes are to be communicated and provided for the disposition of ISO/IEC JTC 1/SC 7/WG 6 to support the further development of quality models.

## 3.4 Evaluation activities in this research project

This research project follows evaluation guidance from Sonnenberg & Vom Brocke (2011) while applying an evaluation pattern wherein each design science research activity is followed by an evaluation activity. An ex ante evaluation was conducted prior to the construction of the artifact, while two Ex post evaluations were conducted following its design and construction. Table 3.2 provides the design science research evaluation patterns extracted from Sonnenberg & Vom Brocke (2011).

Table 3.2 Design science research evaluation patterns extracted from Sonnenberg & Vom Brocke (2011)

| **Evaluation pattern** | **Intent** |
|---|---|
| Assertion | "Make an *informed argument* Hevner et al. (2004) about why the artefact design is superior and will work in a given situation." |
| Demonstration | "Demonstrate that an artefact design embodies the solution to the identified business problem and works in the context of an artificial setting." |
| Prototyping | "Implement an artefact design as a generic solution to demonstrate the artefact's suitability (March & Storey, 2008)." |

The evaluation pattern *demonstration* is selected and applied for the ex ante evaluation to demonstrate whether the ISO/IEC 25030 (ISO/IEC, 2019) quality requirements framework embodies the solution to the identified problem, namely, the definition of quality requirements related to information security in cloud computing services.

This pattern was also selected and applied for two Ex post evaluations that were given two different data sets. The first Ex post evaluation was conducted to demonstrate whether the constructed artifact (the integrated ontology) extended the applicability of the ISO/IEC 25030

(ISO/IEC, 2019) quality requirements framework to the definition of quality requirements related to information security in cloud computing services. To conduct this demonstration, we instantiated the said artifact as it applies to the data set: 1) the cloud computing threats and the mapped cloud computing control extracted from the Cloud Security Alliance (CSA) report on the top threats to cloud computing (Alliance, 2019); 2) the controls extracted from ISO/IEC 27017 (ISO/IEC, 2015) and ISO/IEC 27018 (ISO/IEC, 2019); 3) the security functions, the security architecture, and the security testing methodology extracted from ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015); 4) the quality measures extracted from ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015); and 5) the quality characteristics and quality sub-characteristics extracted from ISO/IEC 25010 (ISO/IEC, 2019) and ISO/IEC 25012 (ISO/IEC, 2008). More specifically, we determined whether the research question could be answered by instantiating the artifact.

In the second Ex post evaluation, we instantiated the said artifact as it applies to the relevant information of the data set extracted from an example selected from the literature on the application of ISO/IEC 27017 controls to treat an information security threat, vulnerability, and risk resulting from the “shared, on-demand nature of cloud computing”. More specifically, we determined if the artifact could be instantiated to define quality requirements.

# CHAPTER 4

# EVALUATION OF THE ISO/IEC 25030 QUALITY REQUIREMENTS FRAMEWORK APPLICABILITY TO INFORMATION SECURITY IN CLOUD COMPUTING SERVICES

## 4.1 Introduction

As per the design science research guidelines of Sonnenberg & Vom Brocke (2011), this research project conducted an evaluation to demonstrate whether "the envisioned design problem is important for practice, is novel and thus represents a research gap, or results from the inability of existing artefacts to accommodate a new environment or context." As discussed in the preceding chapters, researchers have argued that ISO/IEC 25000 models neither cover specific quality characteristics and technical aspects of cloud computing services nor do they address them in sufficient detail due to their generic nature. Additionally, there is a lack of a systematic approach for the evaluation of their applicability and coverage, and the ISO/IEC 25030 quality requirements framework is not used by researchers to identify applicable and missing quality characteristics and measures.

Given that our goal is to use the ISO/IEC 25030 quality requirement framework as a systematic approach to evaluate the applicability and coverage of ISO/IEC 25000 quality models to information security in cloud computing services, we conducted this evaluation to determine whether the framework provides the methods, models, and guidance for the definition of quality requirements related to information security in cloud computing services or if extensions to the framework need to be built.

Being a component of the research design presented in the preceding chapter, this constitutes the ex ante evaluation of the artifact.

## 4.2 Evaluation design

The definition of quality requirements is embedded in the requirement engineering process. Figure 4.1 shows the ISO/IEC 25030 relationship to the ISO/IEC/IEEE 29148 Requirements Engineering (RE) process.

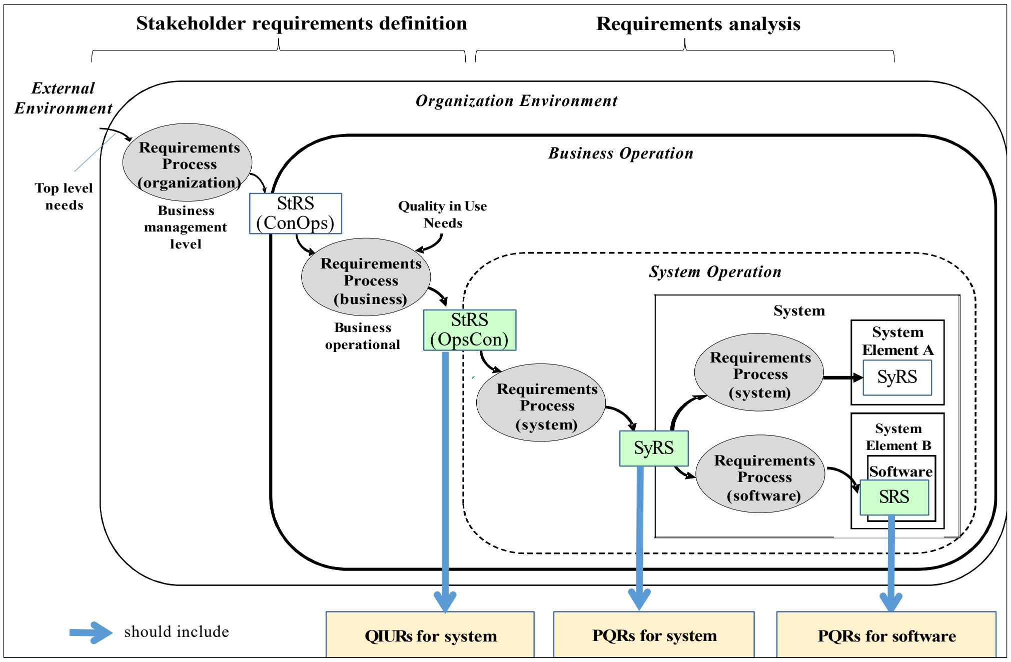


Figure 4.1 Relationship to the ISO/IEC/IEEE 29148 Requirements Engineering process taken from ISO/IEC 25030 (ISO/IEC, 2019)

According to Macaulay (1996), the following defines the objective of a requirements engineering process: “Thus if the objective of the RE process is to specify a successful system then the requirements engineer needs to be aware of the possible causes of failure and must use techniques which will help avoid failure.”

As indicated by Macaulay (1996), possible causes of IS failure are: a lack of a systematic process; poor communication between people and lack of a shared understanding of the system being built; lack of appropriate knowledge or shared understanding; inappropriate, incomplete or inaccurate documentation; poor management of people or resources. Other possible causes of IS failure include: a lack of quality assurance to define and assess the adequacy of a process, and a lack of an audit to ensure that a process, as defined, is both implemented and implemented correctly.

The evaluation consisted of extracting and analyzing the methods (i.e., processes, activities/tasks, and the key artifacts produced), models, and guidance from the ISO/IEC 25030 quality requirements framework that relate to information security and cloud computing services to determine whether they provide the support for the definition of quality requirements. This is a necessary step as our goal is to use the framework as a systematic approach to answer the research question: "To what extent do ISO/IEC 25000 quality models support the definition of quality requirements related to information security in cloud computing services?" Additionally, we determine whether they provide the support to avoid possible causes of IS failure.

The ISO/IEC 25030 quality requirements framework complies with the technical processes defined in ISO/IEC/IEEE 15288 Systems and Software Engineering — System Life Cycle Processes and ISO/IEC/IEEE 12207 Systems and Software Engineering — Software Life Cycle Processes. The latter follows a process model that is identical to ISO/IEC/IEEE 15288. Therefore, the mapping of ISO/IEC 25030 quality requirements definition methods in relation to ISO/IEC/IEEE 15288 system requirements definition methods was extracted from ISO/IEC 25030 (ISO/IEC, 2019) and used in this analysis to identify relationships to other standards. Tables 4.6 and 4.7 show the extracted methods and their mapping. When relationships to other standards that relate to information security and cloud computing services are identified, we also extracted and analyzed their methods, models, and guidance.

## 4.3 Applicability of methods and models to information security in cloud computing services

### 4.3.1 Systems and software quality requirements definition: methods and models

The ISO/IEC 25030 framework provides methods and models for the definition of systems and software quality requirements. Tables 4.6 and 4.7 show ISO/IEC 25030 quality requirements definition process methods.

The framework specifies that two types of requirements for information and communication technology (ICT) products should be considered based on their sources, namely, domain-based requirements and ICT requirements. The former derives directly from stakeholder domain needs determined through requirements analysis processes whereas the latter is introduced when an ICT, such as cloud computing technology, is selected as a technical solution during the design process.

The framework also specifies that the category of the target system is used to determine which quality characteristics have higher priority and which quality measures should be used. For this purpose, the framework refers to ISO/IEC TR 12182 to provide categorization guidance. This indication relates to information security when the categorization of the target IS indicates that quality characteristics, such as freedom from risk, have higher priority. This quality characteristic is defined in ISO/IEC 25022 (ISO/IEC, 2016) as “the degree to which the quality of a product or system mitigates or avoids potential risk to the user, organisation or project, including risks to economic status, human life, health, or the environment” where the risks can arise from “…inadequate operational safety or protection of security or privacy.”

Additionally, the framework specifies that quality requirements cannot be defined and analysed separately from functional/data requirements. In some cases, quality requirements are attached to functional/data requirements or achieved by specifying requirements for new functions. To illustrate the latter, the standard presents the following example in relation to information security: “Some confidentiality requirements are achieved by requirements for access control function”. ISO/IEC 25023 refers to such functions in the security measures section that stipulates that:

> Security protection requirements vary widely from the case of a stand-alone system to the case of a system connected to the Internet. The determination of the required security functions and the assurance of their effectiveness have been addressed extensively in related International Standards. The user of this International Standard must determine what kind of security functions need to be used in each case depending on the level of risk. (ISO/IEC, 2016, p. 22)

Moreover, the framework specifies that architectures and design policies can implement quality requirements.

Therefore, we extracted and analyzed ISO/IEC 25023 (ISO/IEC, 2016) security measures to determine whether security functions were comprised. In addition to security functions, we determined whether evaluation methods for the assurance of their effectiveness were comprised. Moreover, we determined whether architectures and design policies that relate to security were comprised. Table 4.1 shows that the security function extracted from the quality measure data encryption correctness is data encryption.

Table 4.1 Security function data encryption extracted from ISO/IEC 25023 (ISO/IEC, 2016)

| **Extracted from ISO/IEC 25023 quality measures** | **ISO/IEC 25023 quality measures** | **ISO/IEC 25010 quality model** | |
|---|---|---|---|
| **Function** | **Quality measure** | **Description** | **Measurement function** |
| Data encryption | Data encryption correctness | How correctly is the encryption/decryption of data items implemented as stated in the requirement specification? | X = A/B<br>A = Number of data items encrypted/decrypted correctly<br>B = Number of data items that require encryption/decryption |

Table 4.2 shows the results of the analysis. The function’s data encryption, user input bound check, digital signature, user access to system and data logging, and authentication were extracted as security functions from ISO/IEC 25023 (ISO/IEC, 2016) security measures.

Table 4.2 Security functions extracted from ISO/IEC 25023 (ISO/IEC, 2016)

| **Extracted from ISO/IEC 25023 quality measures** | **ISO/IEC 25023 quality measures** | **ISO/IEC 25010 quality model** | |
|---|---|---|---|
| **Function** | **Quality measure** | **Quality sub-characteristic** | **Quality characteristic** |
| Data encryption | Data encryption correctness | Confidentiality | Security |
| | Strength of cryptographic algorithms | | |
| User input bound check | Buffer overflow prevention | Integrity | |
| Digital signature | Digital signature usage | Non-repudiation | |
| User access to system and data logging | User audit trail completeness | Accountability | |
| | System log retention | | |
| Authentication | Authentication mechanism sufficiency | Authenticity | |
| | Authentication rules conformity | | |

We then proceeded with the same analysis for all remaining ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015) quality measures. Tables 4.3 and 4.4 show the results of the analysis. Although the function backup of data was extracted from reliability and recoverability measures (rather than security measures), we defined it as a security function given that its desired outcome is recognized by information security experts as follows: “Data backups are the first line of defense against crashing of systems, corruption of data, exploits leading to data integrity issues, and accidental loss of data” (Rao & Nayak, 2014). This is also the case for the function back-up/restore procedures, which was extracted from a recoverability measure rather than a security measure, with a desired outcome that is also recognized by information security experts as follows: “Data backups provide for continued operation by effective restoration of data and assure continued availability of the systems albeit the time taken for bringing up of the system by restoring the data for the corrupted or crashed part of the system” (Rao & Nayak, 2014). Additionally, we extracted redundancy of components from a reliability measure which we defined as security architecture given that it is recognized as a safety and security architecture tactic (Rehn, 2009). Finally, we extracted penetration testing from a confidentiality measure, which we defined as a security-focused verification method based on the following definition: “A test methodology intended to circumvent the security function of a system” (Ross, McEvilley & Oren, 2018). Table 4.5 shows the consolidated results of this extraction.

Table 4.3 Security architecture and security function extracted from ISO/IEC 25023 (ISO/IEC, 2016)

| **Extracted from ISO/IEC 25023 quality measures** | **ISO/IEC 25023 quality measures** | **ISO/IEC 25010 quality model** | |
|---|---|---|---|
| **Function** | **Quality measure** | **Quality sub-characteristic** | **Quality characteristic** |
| Redundancy of components | Redundancy of components | Fault tolerance | Reliability |
| Backup of data | Backup data completeness | Recoverability | |

Table 4.4 Security function and security-focused verification method extracted from ISO/IEC 25024 (ISO/IEC, 2015)

| **Extracted from ISO/IEC 25024 quality measures** | **ISO/IEC 25024 quality measures** | **ISO/IEC 25012 quality model** |
|---|---|---|
| **Function** | **Quality measure** | **Quality characteristic** |
| Data encryption | Encryption usage | Confidentiality |
| Penetration test | Non vulnerability | |
| Backup of data | Periodical backup | Recoverability |
| Back-up/restore procedures | Architecture recoverability | |

Table 4.5 Extracted security functions, security architecture, and the security-focused evaluation method

| **Extracted from ISO/IEC 25023 and ISO/IEC 25024 quality measures** | **Security function** | **Security architecture** | **Security-focused evaluation method** |
|---|---|---|---|
| Data encryption | X | | |
| User input bound check | X | | |
| Digital signature | X | | |
| Logging of user access to system and data | X | | |
| Authentication | X | | |
| Redundancy for components | | X | |
| Penetration testing | | | X |
| Backup of data | X | | |
| Back-up/restore procedures | X | | |

In summary, we observed that systems and software quality requirements definition methods and models relate to information security in the following cases:

1. The categorization of the target IS indicates that quality characteristics and quality sub-characteristics, such as freedom from risk, have higher priority;
2. The target IS includes quality requirements specified by quality characteristics and quality sub-characteristics, such as freedom from risk;
3. The target IS includes quality requirements derived from quality requirements specified by quality characteristics and quality sub-characteristics, such as freedom from risk;
4. The user of the framework determines security functions that need to be used for the target IS based on the level of risk;

5. Quality requirements of the target IS are achieved by functional requirements that specify security functions;
6. Quality requirements of the target IS are attached to functional requirements that specify security functions;
7. Quality requirements of the target IS are implemented by security architectures;
8. Quality requirements of the target IS are evaluated by security-focused evaluation methods.

Additionally, these systems and software quality requirements definition methods relate to cloud computing services in the following case:

- The target IS includes quality requirements derived from ICT requirements. The latter are introduced by the selection of cloud computing technology as technical solutions in the design process.

Finally, these systems and software quality requirements definition methods relate to information security and cloud computing services in the following case:

- Cloud computing information security risks and their treatments are a source of quality requirements

### 4.3.2 Systems and software requirements definition: methods and models

The ISO/IEC 15288 framework provides methods and models to define system and software requirements. Tables 4.6 and 4.7 show ISO/IEC 15288 (ISO/IEC/IEEE, 2015) systems and software requirements definition process methods.

In the stakeholder needs and requirements definition process, the task "define the stakeholders needs and requirements definition strategy" relates to information security as per its guidance. The guidance stipulates that: "…some stakeholders can have interests that oppose the system or oppose each other" and that "the intent or desires of those that oppose the system, or detractors of the system, are addressed through the Risk Management process, threat analyses of the System Analysis process, or the system requirements for security, adaptability, or resilience." The risk management process includes the planning of risk management, where the risk management strategy is defined and includes the risk management of the supply chain.

The tasks "identify the stakeholders requirements and functions that relate to critical quality characteristics", "define stakeholder requirements, consistent with life cycle concepts, scenarios, interactions, constraints, and critical quality characteristics", "analyze the complete set of stakeholder requirements", "define critical performance measures that enable the assessment of technical achievement", "obtain explicit agreement on the stakeholder requirements", and "maintain traceability of stakeholder needs and requirements" are mapped with ISO/IEC 25030 methods and consequently relate to information security as previously presented in section 4.3.1.

In the system and software requirements definition process, the task "define the functional boundary of the system in terms of the behavior and properties to be provided" is mapped with ISO/IEC 25030 methods and consequently relates to information security as previously presented in section 4.3.1.

The task "identify system requirements that relate to risks, criticality of the system, or critical quality characteristics" refers to ISO/IEC 27036 for information security requirements of outsourcing products and services and to ISO/IEC 25030 guidance for external system quality factors and characteristics. The task definition stipulates that "The security-related risks are defined, including administrative, personnel, physical, computer, communication, network, emission and environment factors using, as appropriate, applicable security standards." It also refers to ISO/IEC 27036 guidance for information security requirements for the outsourcing of products and services and ISO/IEC 25030 guidance for external system quality factors and characteristics.

The ISO/IEC 27036 Information technology — Security techniques — Information security for supplier relationships series provide specific guidance for cloud computing services in its fourth part (i.e., Part: 4 Guidelines for security of cloud services) and refers to the ISO/IEC 27005 information security risk management process. ISO/IEC 27036-4 provides guidance on the application of controls to treat risks and threats related to cloud computing services. It stipulates that applicable controls can differ depending on the combination of capability type, service category, deployment model, and target customer profile. As it relates to the controls, the standard refers to ISO/IEC 27017 and ISO/IEC 27018. More specifically, the standard provides guidance on the application of ISO/IEC 27002 and ISO/IEC 27017 to treat risks and threats related to a public cloud computing service deployment model in combination with different capabilities types, namely, infrastructure, platform, and application. Guidance is also provided for hybrid and private cloud computing service deployment models.

ISO/IEC 27036-4 also provides guidance to address specific information security risks in ICT supply chains and refers to the third part of ISO/IEC 27036-3 (i.e., Part 3: Guidelines for information and communication technology supply chain security) for guidance on the implementation of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 methods. A list of activities is also provided in addition to ISO/IEC/IEEE 15288 risk management methods, the second part of ISO/IEC 27036-2 (i.e., Part 2: Requirements), and ISO/IEC 27005 information security risk management methods.

As it pertains to the tasks, namely, “define system requirements and rationale”, “analyze the complete set of system requirements”, “define critical performance measures that enable the assessment of technical achievement”, “obtain explicit agreement on the system requirements”, and “maintain traceability of the system requirements”, they are mapped with ISO/IEC 25030 methods and consequently relate to information security as previously presented in section 4.3.1.

Consequently, we found that system requirements definition methods and models relate to information security in the following cases:

1. Stakeholders whose interests oppose the system or oppose each other and the intent or desires of those that oppose the system, or detractors of the system, are addressed through the risk management process, threat analyses of the system analysis process, or the system requirements for security, adaptability, or resilience.

Additionally, they relate to information security and cloud computing services in the following cases:

2. Cloud computing service information security risks and threats, and information security risks in ICT supply chains are identified, analyzed, prioritized, and their treatment identified, prioritized, and selected using ISO/IEC 27036-4 guidance and the ISO/IEC 27005 information security risk management framework;
3. Controls are selected from ISO/IEC 27017 and ISO/IEC 27018 to treat risks and threats related to public, private, and hybrid cloud computing service deployment models in combination with different capabilities types, namely, infrastructure, platform, and application.

### 4.3.3 System security requirements definition: methods and models

The NIST SP 800-160 framework provides methods and models for the definition of system security requirements. Therefore, all methods under NIST SP 800-160 (NIST, 2018) system security requirements definition processes shown in Tables 4.6 and 4.7 relate to information security.

Additionally, the activities “analyze stakeholders security requirements”, “prepare for system security requirements”, “define system security requirements”, and “analyze system security in system requirement” refer to ISO/IEC 27001 and 27002 guidance. Consequently, these activities relate to information security as previously presented in section 4.3.2. The activity “define system security requirements” also refers to ISO/IEC 27036 guidance and consequently relates to information security and cloud computing services as previously presented in section 4.3.2.

The task "define stakeholder protection needs" refers to ISO/IEC 25010 guidance, while the activities "transform stakeholder protection needs into security requirements" and "define system security requirements" refer to ISO/IEC 25030 guidance. Consequently, these tasks and activities relate to information security as previously presented in section 4.3.1.

Consequently, we conclude that systems security requirements definition methods and models relate to information security. Additionally, they relate to information security and cloud computing services in the following cases:

1. Cloud computing service information security risks and threats, and information security risks in ICT supply chains are identified, analyzed, prioritized, and their treatment identified, prioritized, and selected using ISO/IEC 27036-4 guidance and the ISO/IEC 27005 information security risk management framework;
2. Controls are selected from ISO/IEC 27017 and ISO/IEC 27018 to treat risks and threats related to public, private, and hybrid cloud computing service deployment models in combination with different capability types, namely, infrastructure, platform, and application.

### 4.3.4 Information security risk management: methods and models

The ISO/IEC 27005 framework provides methods and models to identify, analyze, and prioritize information security risks as well as to identify, select, and prioritize treatments. Controls can be selected from ISO/IEC 27001, ISO/IEC 27002, ISO/IEC 27017, and ISO/IEC 27018 to treat information security risks. Therefore, all methods under ISO/IEC 27005 (ISO/IEC, 2018) shown in Table 4.9 relate to information security.

Additionally, when ISO/IEC 27036-4 guidance on the application of controls to treat risks related to a cloud computing service is applied (as previously presented in section 3.3.2), the methods extracted from the framework shown in Table 4.9 relate to information security and cloud computing services.

Consequently, we conclude that information security risk management methods, models, and guidance relate to information security and cloud computing services in the following cases:

1. Cloud computing service information security risks and threats and information security risks in ICT supply chains are identified, analyzed, prioritized, and their treatment is identified, prioritized, and selected under ISO/IEC 27036-4 guidance and the ISO/IEC 27005 information security risk management framework;
2. Controls are selected from ISO/IEC 27017 and ISO/IEC 27018 to treat risks and threats related to public, private, and hybrid cloud computing service deployment models in combination with different capability types, namely, infrastructure, platform, and application.

Table 4.6 Stakeholder needs and requirements definition methods (extracted processes, activities, tasks, and guidance)

| **ISO/IEC/IEEE 15288:2015** | | | **ISO/IEC 25030:2019** | | **NIST SP 800-160** | |
|---|---|---|---|---|---|---|
| **6.4.2 Stakeholder needs and requirements definition process** | | **Referred guidance** | **Clause 7 Quality requirements processes** | **Referred guidance** | **3.4.2 Stakeholder needs and requirements definition process** | **Referred guidance** |
| Activity | a) Prepare for stakeholder needs and requirements definition. | | | | SN-1 Prepare for stakeholder protection needs and security requirements definition. | ISO/IEC/IEEE 15288, Section 6.4.2.3 a),<br>ISO/IEC 27001,<br>ISO/IEC 27002 |
| Task | 1) Identify the stakeholders who have an interest in the system throughout its life cycle. | | 7.3.1 Identification of stakeholders | | SN-1.1 Identify the stakeholders who have a security interest in the system throughout its life cycle. | |
| Task | 2) Define the stakeholder needs and requirements definition strategy. | | | | SN-1.2 Define the stakeholder protection needs and security requirements definition strategy. | |
| Task | 3) Identify and plan for the necessary enabling systems or services needed to support stakeholder needs and requirements definition. | | | | SN-1.3 Identify, plan for, and obtain access to enabling systems or services to support the security aspects of the stakeholder needs and requirements definition process. | |
| Task | 4) Obtain or acquire access to the enabling systems or services to be used. | | | | | |
| Activity | b) Define stakeholder needs. | | | | SN-2 Define stakeholder protection needs. | ISO/IEC/IEEE 15288, Section 6.4.2.3 b),<br>ISO/IEC 25010,<br>ISO/IEC 27001,<br>ISO/IEC 27002 |
| Task | 1) Define context of use within the concept of operations and the preliminary life cycle concepts. | | 7.3.2 Defining stakeholder needs | | SN-2.1 Define the security context of use across all preliminary life cycle concepts. | |
| Task | | | | | SN-2.2 Identify stakeholder assets and asset classes. | |
| Task | | | | | SN-2.3 Prioritize assets based on the adverse consequence of asset loss. | |
| Task | | | | | SN-2.4 Determine asset susceptibility to adversity and uncertainty. | |
| Task | 2) Identify stakeholder needs. | | | | SN-2.5 Identify stakeholder protection needs. | |
| Task | 3) Prioritize and down-select needs. | | | | SN-2.6 Prioritize and down-select the stakeholder protection needs. | |
| Task | 4) Define the stakeholder needs and rationale. | | | | SN-2.7 Define the stakeholder protection needs and rationale. | |
| Activity | c) Develop the operational concept and other life cycle concepts. | | | | SN-3 Develop the security aspects of operational and other life cycle concepts. | ISO/IEC/IEEE 15288, Section 6.4.2.3 c),<br>ISO/IEC 27001,<br>ISO/IEC 27002 |

| ISO/IEC/IEEE 15288:2015 | | | ISO/IEC 25030:2019 | | NIST SP 800-160 | |
|---|---|---|---|---|---|---|
| **6.4.2 Stakeholder needs and requirements definition process** | | **Referred guidance** | **Clause 7 Quality requirements processes** | **Referred guidance** | **3.4.2 Stakeholder needs and requirements definition process** | **Referred guidance** |
| Task | 1) Define a representative set of scenarios to identify all required capabilities that correspond to anticipated operational and other life cycle concepts. | | | | SN-3.1 Define a representative set of scenarios to identify all required protection capabilities and security measures that correspond to anticipated operational and other life cycle concepts. | |
| Task | 2) Identify the interaction between users and the system. | | | | SN-3.2 Identify the security-relevant interaction between users and the system. | |
| Activity | d) Transform stakeholder needs into stakeholder requirements. | | | | SN-4 Transform stakeholder protection needs into security requirements. | ISO/IEC/IEEE 15288, Section 6.4.2.3 d), ISO/IEC 25030, ISO/IEC 27001, ISO/IEC 27002, ISO/IEC 27034-1 (SDL) Section A.9.2 |
| Task | 1) Identify the constraints on a system solution. | | | | SN-4.1 Identify the security-oriented constraints on a system solution. | |
| Task | 2) Identify the stakeholder requirements and functions that relate to critical quality characteristics, such as assurance, safety, security, environment, or health. | | 7.4.2 (2) Select quality characteristics to be specified (Quality in use) | | SN-4.2 Identify the stakeholder security requirements and security functions. | |
| Task | 3) Define stakeholder requirements, consistent with life cycle concepts, scenarios, interactions, constraints, and critical quality characteristics. | | 7.4.2 (3) State quality characteristics (Quality in use) | | SN-4.3 Define stakeholder security requirements, consistent with life cycle concepts, scenarios, interactions, constraints, and critical quality characteristics. | |
| Task | | | | | SN-4.4 Apply security metadata tagging to identify stakeholder security requirements and security-driven constraints. | |
| Activity | e) Analyze stakeholder requirements. | | | | SN-5 Analyze stakeholders security requirements. | ISO/IEC/IEEE 15288, Section 6.4.2.3 e), ISO/IEC 27001, ISO/IEC 27002 |

| **ISO/IEC/IEEE 15288:2015** | | | **ISO/IEC 25030:2019** | | **NIST SP 800-160** | |
|---|---|---|---|---|---|---|
| **6.4.2 Stakeholder needs and requirements definition process** | | **Referred guidance** | **Clause 7 Quality requirements processes** | **Referred guidance** | **3.4.2 Stakeholder needs and requirements definition process** | **Referred guidance** |
| Task | | | 7.4.2 (4) Prioritize quality requirements (QIURs) | | | |
| Task | 1) Analyze the complete set of stakeholder requirements. | | 7.4.2 (5) Specify quality requirements using measures and their required criteria (QIURs)<br>7.4.2 (6) Analyse quality requirements (QIURs) | | SN-5.1 Analyze the complete set of stakeholder security requirements. | |
| Activity | 2) Define critical performance measures that enable the assessment of technical achievement. | | | | SN-5.2 Define critical security-relevant performance and assurance measures that enable the assessment of technical achievement. | |
| Task | 3) Feedback the analyzed requirements to applicable stakeholders to validate that their needs and expectations have been adequately captured and expressed. | | | | SN-5.3 Validate that stakeholder protection needs and expectations have been adequately captured and expressed by the analyzed security requirements. | |
| Task | 4) Resolve stakeholder requirements issues. | | | | SN-5.4 Resolve stakeholder security requirements issues. | |
| Activity | f) Manage the stakeholder needs and requirements definition. | | | | SN-6 Manage stakeholder protection needs and security requirements definition. | ISO/IEC/IEEE 15288, Section 6.4.2.3 f) |
| Task | 1) Obtain explicit agreement on the stakeholder requirements. | | 7.4.2 (7) Manage quality requirements (QIURs) | | 7.4.2 (7) Manage quality requirements (QIURs) | |
| Task | | | | | SN-6.2 Record asset protection data. | |
| Task | 2) Maintain traceability of stakeholder needs and requirements. | | | | SN-6.3 Maintain traceability between stakeholder protection needs and stakeholder security requirements. | |
| Task | 3) Provide key information items that have been selected for baselines. | | | | SN-6.4 Provide security-relevant information items required for stakeholder needs and requirements definition to baselines. | |

Table 4.7 System requirements definition methods (extracted processes, activities, tasks, and guidance)

| **ISO/IEC/IEEE 15288:2015** | | | **ISO/IEC 25030:2019** | | **NIST SP 800-160** | |
|---|---|---|---|---|---|---|
| **6.4.3 System requirements definition process** | | **Referred guidance** | **Clause 7 Quality requirements processes** | **Referred guidance** | **3.4.3 System Requirements Definition Process** | **Referred guidance** |
| Activity | a) Prepare for system requirements definition. | | | | SR-1 Prepare for system security requirements. | ISO/IEC/IEEE 15288, Section 6.4.3.3 a), ISO/IEC 27001, ISO/IEC 27002 |
| Task | 1) Define the functional boundary of the system in terms of the behavior and properties to be provided. | | 7.4.2 (1) Define target entities to managed to achieve quality | | SR-1.1 Define the security aspects of the functional boundary of the system in terms of the security behavior and security properties to be provided. | |
| Task | | | | | SR-1.2 Define the security domains of the system and their correlation to the functional boundaries of the system. | |
| Task | 2) Define the system requirements definition strategy. | | | | SR-1.3 Define the security aspects of the system requirements definition strategy. | |
| Task | 3) Identify and plan for the necessary enabling systems or services needed to support system requirements definition. | | | | SR-1.4 Identify, plan for, and obtain access to enabling systems or services to support the security aspects of the system requirements definition process. | |
| Task | 4) Obtain or acquire access to the enabling systems or services to be used. | | | | | |
| Activity | b) Define system requirements. | | | | SR-2 Define system security requirements. | **ISO/IEC 27036**, **ISO 25030**, ISO/IEC 27001, ISO/IEC 27002, ISO/IEC 27034-1 (SDL) Section A.9.2 |
| Task | 1) Define each function that the system is required to perform. | | | | SR-2.1 Define each security function that the system is required to perform. | |
| Task | 2) Define necessary implementation constraints. | | | | SR-2.2 Define system security requirements, security constraints on system requirements, and rationale. | |
| Task | 3) Identify system requirements that relate to risks, criticality of the system, or critical quality characteristics. | **ISO/IEC 27036** | 7.4.2 (2) Select quality characteristics to be specified (Product/Data quality) | | SR-2.3 Incorporate system security requirements and associated constraints into system requirements and define rationale. | |
| Task | 4) Define system requirements and rationale. | | 7.4.2 (3) State quality characteristics (Product/Data quality) | | | |

| **ISO/IEC/IEEE 15288:2015** | | | **ISO/IEC 25030:2019** | | **NIST SP 800-160** | |
|---|---|---|---|---|---|---|
| **6.4.3 System requirements definition process** | | **Referred guidance** | **Clause 7 Quality requirements processes** | **Referred guidance** | **3.4.3 System Requirements Definition Process** | **Referred guidance** |
| Activity | c) Analyze system requirements. | | | | SR-3 Analyze system security in system requirements. | ISO/IEC/IEEE 15288, Section 6.4.3.3 c), ISO/IEC 27001, ISO/IEC 27002 |
| Task | | | 7.4.2 (4) Prioritize quality requirements (PQRs/DQRs) | | | |
| Task | 1) Analyze the complete set of system requirements. | | 7.4.2 (5) Specify quality requirements using measures and their required criteria (PQRs/DQRs)<br>7.4.2 (6) Analyse quality requirements (PQRs/DQRs) | | SR-3.1 Analyze the complete set of system requirements in consideration of security concerns. | |
| Task | 2) Define critical performance measures that enable the assessment of technical achievement. | | | | SR-3.2 Define security-driven performance and assurance measures that enable the assessment of technical achievement. | |
| Task | 3) Feedback the analyzed requirements to applicable stakeholders for review. | | | | SR-3.3 Provide the analyzed system security requirements and security-driven constraints to applicable stakeholders for review. | |
| Task | 4) Resolve system requirements issues. | | | | SR-3.4 Resolve system security requirements and security-driven constraints issues. | |
| Activity | d) Manage system requirements. | | | | SR-4 Manage system security requirements. | ISO/IEC/IEEE 15288, Section 6.4.3.3 d) |
| Task | 1) Obtain explicit agreement on the system requirements. | | 7.4.2 (7) Manage quality requirements (PQRs/DQRs) | | SR-4.1 Obtain explicit agreement on the system security requirements and security-driven constraints | |
| Task | 2) Maintain traceability of the system requirements. | | | | SR-4.2 Maintain traceability of system security requirements and security-driven constraints. | |
| Task | 3) Provide key information items that have been selected for baselines. | | | | SR-4.3 Provide security-relevant information items required for systems requirements definition to baselines. | |

## 4.4 Evaluation results

The evaluation revealed that, on the one hand, the ISO/IEC 25030 quality requirements framework provides the methods, models, and guidance to define the quality requirements originating from quality characteristics/sub-characteristics, such as freedom from risk. Additionally, it specifies that the user of the framework must determine the type of security function to be applied in each case dependent on the level of risk.

We also found that ISO/IEC 25030 provides a mapping of its activities and tasks with ISO/IEC/IEEE 15288 and that the ISO/IEC/IEEE 15288 activity "identify system requirements that relate to risks, criticality of the system or critical quality characteristic" refers to the ISO/IEC 27036 standard series (which include ISO/IEC 27036-4 guidelines for cloud computing services) and ISO/IEC 27005 guidance to define and treat information security risks. However, while these frameworks include reference and mapping relationships, they differ from each other as per their approach and concept. Moreover, the reference and mapping relationships are defined at a high level, and specific relationships at the requirement level are not provided for the users of these frameworks (as shown in Figure 4.2). Additionally, neither ISO/IEC 27036 nor ISO/IEC 27005 provide the methods, models, and guidance to define functional requirements to specify security functions or to define quality requirements related to information security in cloud computing services.

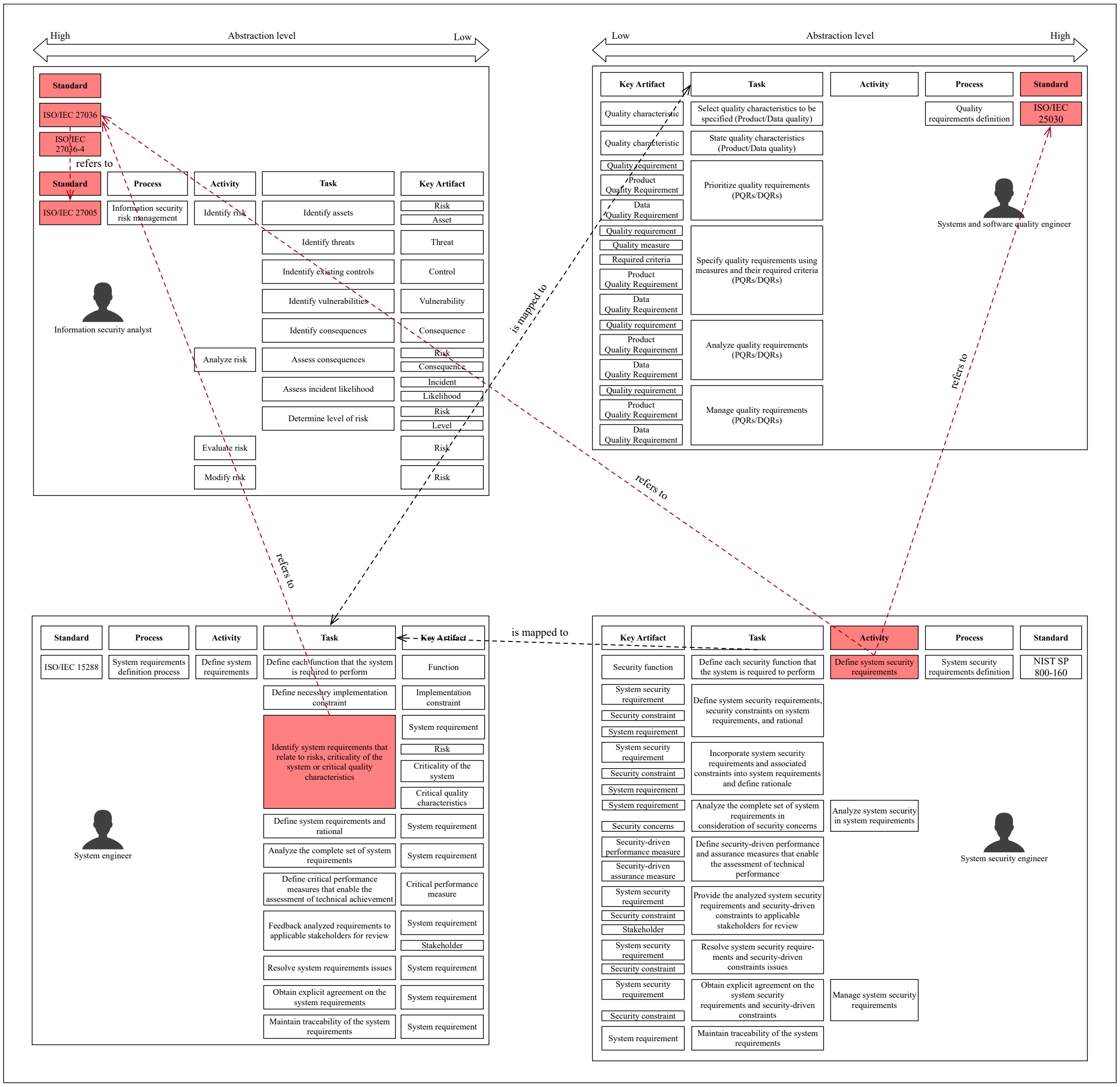

High
Abstraction level
Low
Standard
ISO/IEC 27036
ISO/IEC 27036-4
refers to
Standard
Process
Activity
Task
Key Artifact
ISO/IEC 27005
Information security risk management
Identify risk
Identify assets
Risk
Asset
Identify threats
Threat
Indentify existing controls
Control
Identify vulnerabilities
Vulnerability
Identify consequences
Consequence
Information security analyst
Analyze risk
Assess consequences
Risk
Consequence
Assess incident likelihood
Incident
Likelihood
Determine level of risk
Risk
Level
Evaluate risk
Risk
Modify risk
Risk
Low
Abstraction level
High
Key Artifact
Task
Activity
Process
Standard
Quality characteristic
Select quality characteristics to be specified (Product/Data quality)
Quality requirements definition
ISO/IEC 25030
Quality characteristic
State quality characteristics (Product/Data quality)
Quality requirement
Product Quality Requirement
Data Quality Requirement
Prioritize quality requirements (PQRs/DQRs)
Systems and software quality engineer
Quality requirement
Quality measure
Required criteria
Product Quality Requirement
Data Quality Requirement
Specify quality requirements using measures and their required criteria (PQRs/DQRs)
Quality requirement
Product Quality Requirement
Data Quality Requirement
Analyze quality requirements (PQRs/DQRs)
Quality requirement
Product Quality Requirement
Data Quality Requirement
Manage quality requirements (PQRs/DQRs)
is mapped to
refers to
refers to
refers to
is mapped to
Standard
Process
Activity
Task
Key Artifact
ISO/IEC 15288
System requirements definition process
Define system requirements
Define each function that the system is required to perform
Function
Define necessary implementation constraint
Implementation constraint
Identify system requirements that relate to risks, criticality of the system or critical quality characteristics
System requirement
Risk
Criticality of the system
Critical quality characteristics
System engineer
Define system requirements and rational
System requirement
Analyze the complete set of system requirements
System requirement
Define critical performance measures that enable the assessment of technical achievement
Critical performance measure
Feedback analyzed requirements to applicable stakeholders for review
System requirement
Stakeholder
Resolve system requirements issues
System requirement
Obtain explicit agreement on the system requirements
System requirement
Maintain traceability of the system requirements
System requirement
Key Artifact
Task
Activity
Process
Standard
Security function
Define each security function that the system is required to perform
Define system security requirements
System security requirements definition
NIST SP 800-160
System security requirement
Security constraint
System requirement
Define system security requirements, security constraints on system requirements, and rational
System security requirement
Security constraint
System requirement
Incorporate system security requirements and associated constraints into system requirements and define rationale
System requirement
Security concerns
Analyze the complete set of system requirements in consideration of security concerns
Analyze system security in system requirements
Security-driven performance measure
Security-driven assurance measure
Define security-driven performance and assurance measures that enable the assessment of technical performance
System security engineer
System security requirement
Security constraint
Stakeholder
Provide the analyzed system security requirements and security-driven constraints to applicable stakeholders for review
System security requirement
Security constraint
Resolve system security require-ments and security-driven constraints issues
System security requirement
Security constraint
Obtain explicit agreement on the system security requirements and security-driven constraints
Manage system security requirements
System requirement
Maintain traceability of the system requirements

Figure 4.2 Reference and mapping relationships between frameworks and methods

The NIST SP 800-160 systems security requirements definition framework provides the methods, models, and guidance to define system security requirements, including security functions. In this case, we found that the NIST SP 800-160 provides a mapping of its activities and tasks with ISO/IEC/IEEE 15288. We also found that the activity "define system security requirements" refers to ISO/IEC 25030 for guidance to define quality requirements as well as to the ISO/IEC 27036 standard series (which include ISO/IEC 27036-4 guidelines for cloud computing services), ISO/IEC 27001, and ISO/IEC 27002 guidance to define and treat information security risks. However, while these frameworks include reference and mapping relationships, they differ from each other as per their approach and concept. Moreover, the reference and mapping relationships are defined at a high level, and specific relationships at the requirement level are not provided for the users of these frameworks (as shown in Figure 4.2).

Based on our observations in conjunction with our research question, this research project focuses on methods for which the management of cloud computing services information security risks and the definition of requirements intersect as follows:

1. ISO/IEC/IEEE 15288 system requirements definition process activities “define system requirements”, “analyze system requirements”, and “manage system requirements” as well as their associated tasks;
2. NIST SP 800-160 system security requirements definition process activities “define system security requirements”, “analyze system security in system requirements”, and “manage system security requirements” as well as their associated tasks;
3. ISO/IEC 25030 quality requirements definition process tasks “select quality characteristics to be specified (Product/Data quality)”, “state quality characteristics (Product/Data quality)”, “prioritize quality requirements (PQRs/DQRs)”, “specify quality requirements using measures and their required criteria (PQRs/DQRs)”, “analyse quality requirements (PQRs/DQRs)”, and “manage quality requirements (PQRs/DQRs)”;
4. The ISO/IEC 27005 activities “identify risk”, “analyze risk”, “evaluate risk”, and “modify risk” and their associated tasks.

Tables 4.8 and 4.9 show the selected frameworks and methods for this research project.

Table 4.8 Selected methods (processes, activities, and tasks)

| **ISO/IEC/IEEE 15288:2015** | | | **ISO/IEC 25030:2018** | | **NIST SP 800-160** | |
|---|---|---|---|---|---|---|
| **6.4.3 System requirements definition process** | | **Extracted key artifacts** | **7 Quality requirements processes** | **Extracted key artifacts** | **3.4.3 System requirements definition process** | **Extracted key artifacts** |
| Activity | b) Define **system requirements**. | System requirement | | | SR-2 Define **system security requirements**. | System security requirement |
| Task | 1) Define each **function** that the **system** is required to perform. | Function<br>System | | | SR-2.1 Define each **security function** that the system is required to perform. | Security function |
| Task | 2) Define necessary **implementation constraints**. | Implementation constraint | | | SR-2.2 Define **system security requirements**, **security constraints** on **system requirements**, and rationale. | System security requirement<br>Security constraint<br>System requirement |
| Task | 3) Identify **system requirements** that relate to **risks**, **criticality** of the **system**, or **critical quality characteristics**. | System requirement<br>Risk<br>Criticality of the system<br>Critical quality characteristic | 7.4.2 (2) Select **quality characteristics** to be specified (**Product/Data quality**) | Quality characteristic<br>Data quality characteristic<br>Product quality characteristic | SR-2.3 Incorporate **system security requirements** and associated **constraints** into **system requirements** and define rationale. | System security requirement<br>Constraint<br>System requirement |
| Task | 4) Define **system requirements** and rationale. | System requirement | 7.4.2 (3) State **quality characteristics** (**Product/Data quality**) | Quality characteristic<br>Data quality characteristic<br>Product quality characteristic | | |
| Activity | c) Analyze **system requirements** | System requirement | | | SR-3 Analyze **system security** in **system requirements.** | System requirement |
| Task | | | 7.4.2 (4) Prioritize **quality requirements** (**PQRs/DQRs**) | Quality requirement<br>Data quality requirement<br>Product quality requirement | | |
| Task | 1) Analyze the complete set of **system requirements**. | System requirement | 7.4.2 (5) Specify **quality requirements** using **measures** and their required **criteria** (**PQRs/DQRs**)<br>7.4.2 (6) Analyze **quality requirements** (**PQRs/DQRs**) | Quality requirement<br>Quality measure<br>Data quality requirement<br>Product quality requirement<br>Criteria | SR-3.1 Analyze the complete set of **system requirements** in consideration of **security concerns**. | System requirement<br>Security concern |
| Task | 2) Define **critical performance measures** that enable the assessment of technical achievement. | Critical performance measure | | | SR-3.2 Define **security-driven performance** and **assurance measures** that enable the assessment of technical achievement. | Security-driven performance measure<br>Security-driven assurance measure |

| **ISO/IEC/IEEE 15288:2015** | | | **ISO/IEC 25030:2018** | | **NIST SP 800-160** | |
|---|---|---|---|---|---|---|
| **6.4.3 System requirements definition process** | | **Extracted key artifacts** | **7 Quality requirements processes** | **Extracted key artifacts** | **3.4.3 System requirements definition process** | **Extracted key artifacts** |
| Task | 3) Feedback the analyzed requirements to applicable **stakeholders** for review. | Stakeholder | | | SR-3.3 Provide the analyzed **system security requirements** and **security-driven constraints** to applicable **stakeholders** for review. | System security requirement<br>Security constraint<br>Stakeholder |
| Task | 4) Resolve **system requirements** issues. | System requirement | | | SR-3.4 Resolve **system security requirements** and **security-driven constraints** issues. | System security requirement<br>Security constraint |
| Activity | d) Manage **system requirements**. | System requirement | | | SR-4 Manage **system security requirements**. | System security requirement |
| Task | 1) Obtain explicit agreement on the **system requirements**. | | 7.4.2 (7) Manage **quality requirements (PQRs/DQRs)** | Quality requirement<br>Data quality requirement<br>Product quality requirement | SR-4.1 Obtain explicit agreement on the **system security requirements** and **security-driven constraints** | System security requirement<br>Security constraint |
| Task | 2) Maintain traceability of the **system requirements**. | | | | SR-4.2 Maintain traceability of **system security requirements** and **security-driven constraints**. | System security requirement<br>Security constraint |

Table 4.9 Selected methods (processes, activities, and tasks)

| | ISO/IEC 27005:2018 | | |
|---|---|---|---|
| | 8 Information security risk assessment | 9 Information security risk treatment | **Extracted key artifacts** |
| Activity | 8.2 Identify **risk** | | Risk |
| Task | 8.2.2 Identify **assets** | | Asset |
| Task | 8.2.3 Identify **threats** | | Threat |
| Task | 8.2.4 Identify existing **controls** | | Control |
| Task | 8.2.5 Identify **vulnerabilities** | | Vulnerability |
| Task | 8.2.6 Identify **consequences** | | Consequence |
| Activity | 8.3 Analyze **risk** | | Risk |
| Task | 8.3.2 Assess **consequences** | | Consequence |
| Task | 8.3.3 Assess **incident likelihood** | | Incident likelihood |
| Task | 8.3.4 Determine **level of risk** | | Level of risk |
| Activity | 8.4 Evaluate **risk** | | Risk |
| Activity | | 9.2 Modify **risk** | Risk |

## 4.5 Extraction of key artifacts

Given that the activities that manage systems requirements, manage system security requirements, and manage quality requirements were selected for this research project and entail the establishment and maintenance of traceability between the requirements and their sources, we extract the key artifacts (i.e., requirements and requirements attributes) from the activities/tasks selected for this research project following the generic traceability model by Wibowo & Davis (2020). Figure 4.3 provides the generic traceability model.

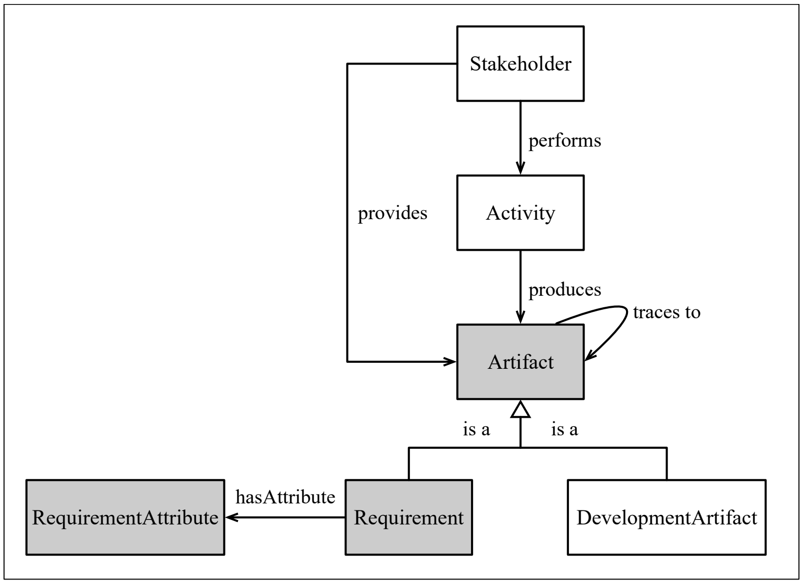


Figure 4.3 Generic traceability model
adapted from Wibowo & Davis (2020)

According to the generic traceability model by Wibowo & Davis (2020), a requirement *is an* artifact and *has* requirement attributes and a development artifact *is an* artifact. Given that development artifacts depend on the type of the development framework (Wibowo & Davis, 2020), they were not selected for this research project. As per requirements attributes, according to Wheatcraft, Ryan & Dick (2016), they can be classified in four broad categories: attributes to help define the requirement and its intent, attributes associated with the system-of-interest verification, attributes to help maintain the requirements, and attributes to show applicability and allow reuse.

For example, the ISO/IEC 25030 quality requirements definition process activity/task “specify quality requirements using measures and their required criteria (PQRs/DQRs)” *produces* the artifact quality requirement which *has* the requirement attributes quality measure and required criteria. In this case, both quality measure and required criteria fall under the requirement attributes categories: attributes to help define the requirement and its intent, and attributes associated with the system-of-interest verification. Table 4.10 shows the extracted key artifacts as well as their type and category.

Table 4.10 Type and category of the extracted key artifacts

| **Extracted key artifacts** | **Artifact type (Wibowo & Davis, 2020)** | | **Requirement attribute category (Wheatcraft et al., 2016)** | | | |
|---|---|---|---|---|---|---|
| | **Requirement** | **Requirement attribute** | **1** | **2** | **3** | **4** |
| Asset | | X | X | | | |
| Threat | | X | X | | | |
| Control | X | | | | | |
| Vulnerability | | X | X | | | |
| Consequence | | X | X | | | |
| Incident | | X | X | | | |
| Likelihood | | X | X | | | |
| Risk | | X | X | | | |
| Level of risk | | X | X | | | |
| Quality characteristic | | X | X | | | |
| Quality requirement | X | | | | | |
| Product quality requirement | X | | | | | |
| Data quality requirement | X | | | | | |
| Quality measure | | X | | X | | |
| Required criteria | | X | | X | | |
| Function | | X | X | | | |
| Implementation constraint | | X | X | | | |
| System requirement | X | | | | | |
| Criticality of the system | | X | X | | | |
| Critical performance measure | | X | X | X | | |
| Stakeholder | | X | X | | | |
| Security function | | X | X | | | |
| System | | X | X | | | |
| Security requirement | X | | | | | |
| System security requirement | X | | | | | |
| Security constraint | | X | X | | | |
| Security concern | | X | X | | | |
| Security-driven performance measure | | X | X | X | | |
| Security-driven assurance measure | | X | X | X | | |

## 4.6 Design problem and objectives

The standards and methods selected for this research project consist of four processes (i.e., system requirements definition process, system security requirements definition process, quality requirements definition process, and information security risk management), eight activities, 33 tasks, and 30 key artifacts (i.e., requirements and requirements attributes).

Because these standards and methods have reference or mapping relationships, stakeholders should consider all associated requirements during the definition of quality requirements related to information security in cloud computing services. However, these reference or mapping relationships have been defined at a high level, and specific relationships at key artifact (i.e., requirements and requirements attributes) levels have not been provided. If such relationships exist, they are neither defined nor documented for the users of these frameworks. Given that our goal is to use the ISO/IEC 25030 as systematic approach to evaluate the applicability of ISO/IEC 25000 quality models to information security in cloud computing services, the extension of its structure is therefore required. Moreover, as indicated by Lyytinen & Hirschheim (1988) and Macaulay (1996), a lack of a systematic RE process is a possible cause of IS failure.

Additionally, the users of these frameworks fall into at least three different groups of expertise, namely, information security analysts, system and software quality engineers, and system and software security engineers. These stakeholders have a direct responsibility for the design and development of the various IS components. As a result, they must be involved and interact in the RE process. Given that poor communication between people and a lack of a shared understanding of the system being built are possible causes of IS failure (Lyytinen & Hirschheim, 1988; Macaulay, 1996), stakeholders require techniques that "encourages multiparty interaction and helps build consensus and shared understanding" (Macaulay, 1996). However, as previously discussed, the reference or mapping relationships between selected standards and methods are defined at a high level, and specific relationships at key artifact (i.e., requirements and requirements attributes) levels have not been provided (as shown in Figure 4.1). If such relationships exist, they are neither defined nor documented for the users of these frameworks. Furthermore, to the best of our knowledge, there is no defined single cohesive view that can be shared among stakeholders.

Finally, the standards and methods that were selected for this research project include activities and tasks for the management of requirements. The activity "manage quality requirements" from the ISO/IEC 25030 quality requirements definition process consists of: obtaining explicit agreement on quality requirements and approval from stakeholders, establishing and maintaining traceability between defined quality requirements and their sources, performing the steps of the process iteratively to improve the quality requirements. Given that poor management of the RE process, which includes people, resources, and work products, is a possible cause of IS failure (Lyytinen & Hirschheim, 1988; Macaulay, 1996), among the techniques required by stakeholders to support the management of requirements, Macaulay (1996) included techniques to support the traceability of requirements. However, as previously discussed, the reference or mapping relationships between selected standards and methods are defined at a high level, and specific relationships at key artifact (i.e., requirements and requirements attributes) levels have not been provided (as shown in Figure 4.2). Such relationships are required to establish and maintain traceability between quality requirements and their sources.

Based on our observations, we defined the design objective and the design sub-objectives of this research project as follows:

1. Extend the applicability of the ISO/IEC 25030 quality requirement framework to information security in cloud computing services;
    a. Integrate complementary frameworks;
    b. Extend the structure of the quality requirements definition process;
    c. Support communication between stakeholders, namely, information security analysts, system and software quality engineers, and system and software security engineers;
    d. Support traceability between quality requirements and their sources.

## 4.7 Conclusions

Based on evaluation results, we concluded that the ISO/IEC/IEEE 15288, NIST SP 800-160, ISO/IEC 25030, and ISO/IEC 27005 frameworks are complementary. We also identified and selected methods from these frameworks for which the management of cloud computing services information security risks and the definition of requirements intersect as the focus of this research project.

Given that these frameworks and their methods include reference or mapping relationships, stakeholders should consider all associated key artifacts (i.e., requirements and requirements attributes) during the definition of quality requirements related to information security in cloud computing services. However, these reference or mapping relationships are defined at a high level, and specific relationships at key artifact (i.e., requirements and requirements attributes) levels have not been provided to stakeholders. Furthermore, to the best of our knowledge, there is no defined single cohesive view that can be shared among stakeholders.

We also found that these limitations can be linked to possible causes of IS failure, namely, a lack of a systematic RE process, poor communication between people and a lack of shared understanding of the system being built, and poor management of the RE process.

To address these limitations as well as to answer the research question, we identified as a design objective to extend the applicability of the ISO/IEC 25030 quality requirement framework to information security in cloud computing services, which we divided into four design sub-objectives: 1) integrate complementary frameworks; 2) extend the structure of the quality requirements definition process; 3) support communication between the stakeholders involved in the definition of quality requirements; and 4) support the traceability between defined quality requirements and their sources.

The following chapter (Chapter 5) proceeds with the construction of the artifact.

# CHAPTER 5

# CONSTRUCTION OF EXTENSIONS TO THE QUALITY REQUIREMENTS FRAMEWORK: AN INTEGRATED ONTOLOGY FOR INFORMATION SECURITY IN CLOUD COMPUTING SERVICES

## 5.1 Introduction

In the preceding chapter, we concluded that the ISO/IEC/IEEE 15288, NIST SP 800-160, ISO/IEC 25030, and ISO/IEC 27005 frameworks are complementary and that stakeholders should consider all key artifacts (i.e., requirements and requirements attributes) during the definition of quality requirements associated with information security in cloud computing services. However, the reference or mapping relationships between frameworks and their methods are defined at a high level, and specific relationships at the key artifact levels have not been provided to stakeholders. Furthermore, to the best of our knowledge, there is no defined single cohesive view that can be shared among stakeholders.

Given that these limitations can be linked to possible causes of IS failure, we identified as a design objective to extend the applicability of the ISO/IEC 25030 quality requirement framework to information security in cloud computing services. This is a necessary step in answering our research question, namely, “To what extent do ISO/IEC 25000 quality models support the definition of quality requirements related to information security in cloud computing services?”. However, to the best of our knowledge, neither the scientific literature nor the applicable standards define their complementarity or integration.

Additionally, we identified four design sub-objectives, namely, integrate complementary frameworks, extend the structure of the quality requirements definition process, support communication between the stakeholders involved in the definition of quality requirements, and support the traceability between defined quality requirements and their sources.

This chapter introduces metamodels as artifacts in IS research and discusses their selection in the fulfillment of the design objectives.

Being a component of the design science methodology, this constitutes artifact construction. In this case, the artifact is an integrated ontology that explicates the relationships between the concepts involved in the definition of quality requirements related to information security in cloud computing services.

We first evaluated the integrated ontology for completeness, then, in subsequent chapters, we use the integrated ontology to answer our research question, while illustrating its applicability from two perspectives, namely, the cloud service customer and the cloud service provider.

### 5.2 Methodology

In the preceding chapter, we identified as a design objective to extend the applicability of the ISO/IEC 25030 quality requirement framework to information security in cloud computing services, which we divided into four design sub-objectives. We selected ontological metamodeling as technique to achieve these objectives as follows:

1. The first design sub-objective is to integrate complementary frameworks. In integrating IS frameworks, ontological metamodeling has proven to be an effective technique used by IS researchers (Goeken & Alter, 2009);
2. The second design sub-objective is to extend the structure of the quality requirements definition process. As observed by Uschold & Jasper (1999), the use of ontologies in systems engineering benefit the RE process given that "the ontology can assist the process of identifying requirements and defining a specification for an IT system";
3. The third design sub-objective is to support communication between the stakeholders involved in the quality requirements definition process, namely, information security analysts, system and software quality engineers, and system and software security engineers. As indicated by Uschold & Jasper (1999), ontologies are used fundamentally to improve communication between either humans or computers. The authors also

stated that “A major benefit of ontology development is to promote common understanding among knowledge workers”;

4. The fourth design sub-objective is to support the traceability between defined quality requirements and their sources. In systems and software engineering, a trace is “a specified triplet of elements comprising: a source artifact, a target artifact and a link associating the two artifacts” (Gotel et al., 2012). An ontology can support the traceability of requirements because it describes the key concepts in certain knowledge domains and the relationships among them. As pointed out by Wibowo & Davis (2020), several studies have proposed using ontologies to support the traceability of systems and software requirements (e.g., Assawamekin, Sunetnanta & Pluempitiwiriyawej (2010); Ghaisas & Ajmeri (2013); Lima, Garcia, Amaral & Caran (2011); Ramesh & Jarke (2001); Siegemund (2014); Zhang, Witte, Rilling & Haarslev (2006)).

### 5.2.1 Ontological metamodeling

Goeken & Alter (2009) demonstrated that metamodels can support the analysis, comparison, and integration of IS frameworks on a level of abstraction. As Goeken & Alter (2009) reported, there are several ways to conduct metamodeling due to the multiple abstraction mechanisms that can be applied, which are dependent on the purpose and application. In this study, given our design objectives, we apply ontological metamodeling to describe what concepts exist in their respective domain as well as what properties they possess (Atkinson & Kuhne, 2003). Additionally, as reported by Goeken & Alter (2009), when metamodels are used to compare and integrate models on an abstract level, comparability is an essential characteristic. Therefore, in order to guide our metamodeling, comparison, and integration activities, we apply Schuette & Rotthowe (1998) principle of comparability which states that:

> A comparison on the level of the meta models is only possible, if the different languages and grammars, respectively, are compatible. Therefore, the formulation of this principle demands the possibility to transfer one meta model in a different meta model. For this transfer process, a relationship model for the integration of different meta models needs to be built. The transfer is possible, if each of the model elements can be expressed in the other language. As long as the powerfulness of both methods is similar, they can be transferred completely. (Schuette & Rotthowe, 1998, p. 10)

To achieve our design objectives, our approach consists of using ontological metamodeling to gain an understanding of the cloud computing service information security requirements definition, quality requirements definition, and system security requirements definition frameworks, while using the resultant ontologies for their comparison, to identify their complementarity, and for their integration. More specifically, the objective is to obtain an integrated ontology that explicates the relationships between the concepts involved in the definition of quality requirements related to information security in cloud computing services.

In section 5.4.1, we first conduct a literature review to determine whether ISO/IEC 27005 ontologies already exist. Additionally, we also include in the review ISO/IEC 27017 cloud computing service information security controls, ISO/IEC 27018 cloud computing service information security controls for the protection of personally identifiable information (PII), ISO/IEC 27002 from which ISO/IEC 27017 and ISO/IEC 27018 are based, and ISO/IEC 27001 as the Annex A of this standard contains the consolidated ISO/IEC 27002 information security controls. Our literature review revealed the ISO/IEC 27005 ontology by Agrawal (2016), the ISO/IEC 27001 ontology by Milicevic & Goeken (2010), the information security ontology based on ISO/IEC 27001 and ISO/IEC 27002 by Fenz & Ekelhart (2009), and the study by Parkin, van Moorsel & Coles (2009), the latter proposing an ontology that uses ISO/IEC 27002 and incorporates human behavioral factors along with information security. No ontologies exist in the cases of ISO/IEC 27017 and ISO/IEC 27018 in the literature at the time of this research project. The studies by Agrawal (2016) and Milicevic & Goeken (2010) were subsequently selected for this study based on the following selection criteria:

1. Standard concepts: The development of the ontology focuses on all relevant concepts rather than specific concepts of the standards;
2. Methodology: The development of the ontology follows a methodology;
3. Validation and assessment: The developed ontology is subject to validation and/or assessment.

Consequently, in sections 5.4.2 and 5.4.3, we present and analyze the existing ontologies constructed by Agrawal (2016) and Milicevic & Goeken (2010), respectively. In the case of ISO/IEC 27017 and ISO/IEC 27018, where there are no existing ontologies of the standards, we analyze their structure in section 5.4.4, which subsequently determined that they are based on ISO/IEC 27002 as in the case of ISO/IEC 27001. Although they retain the same structure, we found through our analysis one structural addition in the case of ISO/IEC 27017 and one structural addition in the case of ISO/IEC 27018. Therefore, we first extend the ISO/IEC 27001 ontology by Milicevic & Goeken (2010) to the cloud computing service information security control domain. To do so, we model the structural additions, which we then integrate to the ontology to construct the ISO/IEC 27017 and ISO/IEC 27018 information security control ontologies. Then, we analyze the ISO/IEC 27002, ISO/IEC 27017, and ISO/IEC 27018 standards to identify additional concepts and relationships. To conduct this analysis and complete the construction of the cloud computing information security control ontologies, we follow the ontological metamodeling methodology by Agrawal (2016) as it is based on the well-established ontology development methodology by Noy & McGuinness (2001).

In section 5.5.1, we first conduct a literature review to determine whether the ISO/IEC 25030 quality requirements definition ontologies already exist. Our literature review revealed studies by Castillo, Losavio, Matteo & Bøegh (2010) and Cherfi, Akoka & Comyn-Wattiau (2011). We then analyze the existing ontologies, observing that their representativeness is limited to the quality models used to define quality requirements. Consequently, we did not select the existing ontologies for this study as they do not meet the selection criteria. Accordingly, we conclude that the ontological metamodeling of the ISO/IEC 25030 quality requirements definition framework is necessary for this study. As it is the case in section 5.4.3, we follow the methodology proposed by Agrawal (2016) to develop the ISO/IEC 25030 quality requirements definition ontology.

In section 5.6, we first conduct a literature review to determine whether the NIST SP 800-160 system security requirements definition ontologies already exist. Our search revealed no existing ontologies in the literature. Therefore, the ontological metamodeling of the NIST SP

800-160 system security requirements definition framework is necessary for this study. As it is the case in sections 5.4.3 and 5.5.1, we follow the methodology proposed by Agrawal (2016) for the development of the ontology.

Finally, in section 5.7, we conduct a comparison between the information security risk management ontology, the cloud computing information security control ontology, the quality requirements definition ontology, and the system security requirements definition ontology to identify correspondences or mappings between ontology constructs using the semantic relationships proposed by Rizopoulos & Mçbrien (2005). We then proceed with their merging to complete the integration of the ontologies. The resulting integrated ontology is then evaluated for completeness in section 5.7, and our conclusions are presented in section 5.8.

## 5.3 The cloud computing service information security requirements definition ontology

The ISO/IEC 27005 information security risk management framework affords the processes, including activities and tasks, that an organization should execute to select information security controls from ISO/IEC 27001, ISO/IEC 27002, and cloud computing services information security controls from ISO/IEC 27017 and ISO/IEC 27018.

### 5.3.1 Existing ontologies

We initially conducted a search to identify existing ontologies of the said standards, and we found the following relevant studies: Agrawal (2016), Milicevic & Goeken (2010), Fenz & Ekelhart (2009), and Parkin et al. (2009). We evaluated each study using the selection criteria presented previously in section 4.2.

Agrawal (2016) proposed an ontology for ISO/IEC 27005 explicating the core concepts and their relationships. The methodology used to develop the ontology is drawn from Noy & McGuinness (2001). Additionally, Agrawal (2016) created instances of each class based on a case scenario to demonstrate an application of the proposed ontology. Milicevic & Goeken

(2010) used ontological metamodeling to obtain the ISO/IEC 27001 structure. In order to support the construction process of the metamodel, Milicevic & Goeken (2010) made use of ideas from grounded theory (GT) and qualitative data analysis (QDA). The resulting ontology is used to evaluate ISO/IEC 27001 comprehensiveness by correlating its concepts with selected information security ontologies. Fenz & Ekelhart (2009) proposed an information security ontology based on ISO/IEC 27001 and ISO/IEC 27002. The approach developed by Uschold & Gruninger (1996) is used to evaluate the ontology against its initial requirements. However, they did not provide the methodology used for the development of the ontology. Parkin et al. (2009) proposed an ontology that uses ISO/IEC 27002 while incorporating human behavioral factors along with information security. To develop the ontology, Parkin followed the methodology provided by Noy & McGuinness (2001). An assessment of the ontology is conducted based on the ontology evaluation criteria provided by Gómez-Pérez (1996).

Table 5.1 List of existing ISO/IEC 27000 ontologies

| Ontologies | ISO/IEC 27000 series standards | Selection criterion | | |
|---|---|---|---|---|
| | | 1 | 2 | 3 |
| Agrawal (2016) | ISO/IEC 27005 | X | X | X |
| Milicevic & Goeken (2010) | ISO/IEC 27001 | X | X | X |
| Fenz & Ekelhart (2009) | ISO/IEC 27001<br>ISO/IEC 27002 | X | | X |
| Parkin et al. (2009) | ISO/IEC 27002 | | X | X |

It can be seen that the studies by Agrawal (2016) and Milicevic & Goeken (2010) meet the selection criteria. Table 5.1 shows the evaluation results. Accordingly, sections 4.4.2 and 4.4.3 present their respective development methodologies and ontologies.

In the case of ISO/IEC 27017 and ISO/IEC 27018, the initial search revealed that there is no existing ontology in the literature. Given that ISO/IEC 27001, ISO/IEC 27017, and ISO/IEC 27018 are based on ISO/IEC 27002, we analyze their structures in section 4.4.4 and adapt the ISO/IEC 27001 ontology by Milicevic & Goeken (2010) to include ISO/IEC 27017 and ISO/IEC 27018 structural additions. Then, we extend the resulting ontology with additional classes and relationships found in ISO/IEC 27002, ISO/IEC 27017, and ISO/IEC 27018 to construct the cloud computing information security control ontology.

### 5.3.2 Information security risk management framework ontology

Agrawal (2016) proposed an ontology for ISO/IEC 27005 explicating the core concepts and their relationships. The methodology used to develop the ontology was drawn from Noy & McGuinness (2001) and consists of seven steps.

The first step is to determine the domain and the scope of the ontology. In this case, Agrawal (2016) proposed an ontology for ISO/IEC 27005 which represents the concepts and their relationships associated with the information security risk management domain.

The second step is to consider the reuse of existing ontologies. Agrawal (2016) identified several ontologies based on the concept of information security risk management as starting points for the construction of the ISO/IEC 27005 ontology, namely, Pereira & Santos (2012), dos Santos Moreira, Martimiano, dos Santos Brandao & Bernardes (2008), Den Braber, Hogganvik, Lund, Stølen & Vraalsen (2007), Arbanas & Čubrilo (2015), and Herzog, Shahmehri & Duma (2007).

The third step is to capture the key terms that describe the concepts of ISO/IEC 27005. For this step, Agrawal (2016) identified and extracted the concepts from the standard.

The fourth step is to define the classes and the class hierarchy. Each class is defined through a definition extracted from ISO/IEC 27005 and ISO/IEC 27000. Table 5.2 provides these extracted definitions.

The fifth step is to define the object properties of the classes. For this step, Agrawal (2016) identified the object properties that relate a class to another class. For example, in Figure 5.1, the classes control and vulnerability are related to each other through the relation *Mitigates*.

The sixth step is to define the datatype properties. For this step, Agrawal (2016) identified data properties of all classes identified in the fourth step. As indicated by this study, data properties relate a class to a literal. Agrawal (2016) exemplified that a data property value on the class asset would construe that every asset has some value measured in integers. It is important to note that this level of granularity is not represented in the ISO/IEC 27005 ontology extracted from Agrawal (2016).

The seventh step is to create individual instances of the classes. For this step, Agrawal (2016) created instances of each class based on a case scenario to demonstrate an application of the proposed ontology.

Figure 5.1 provides the ISO/IEC 27005 information security risk management ontology extracted from Agrawal (2016).

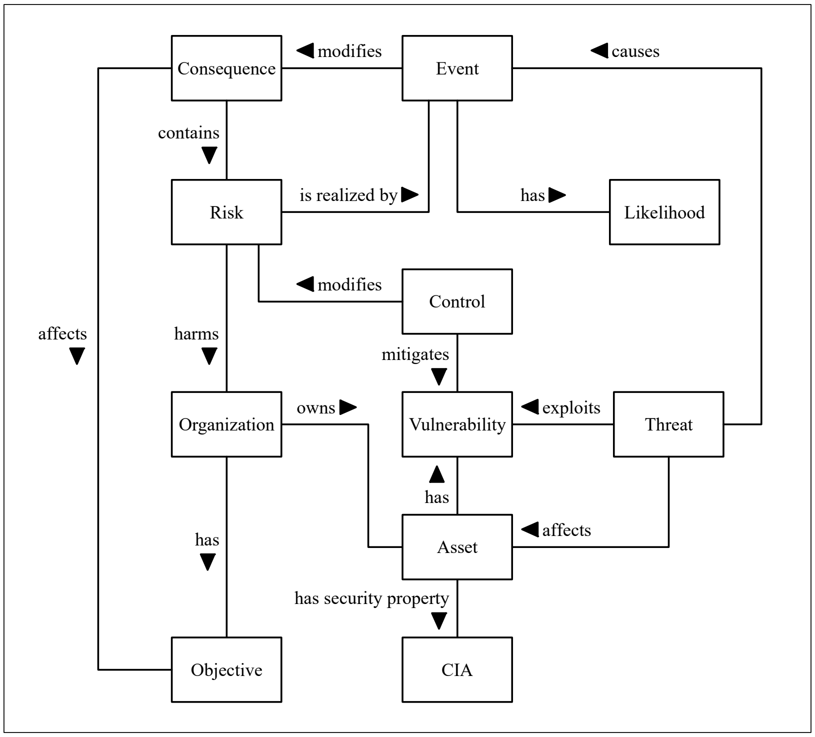


Figure 5.1 The ISO/IEC 27005
information security risk management ontology
taken from Agrawal (2016)

Table 5.2 Definitions taken from ISO/IEC 27005 and ISO/IEC 27000

| **Classes** | **Extracted definitions** |
|---|---|
| Organization | Person or group of people that has its own functions with responsibilities, authorities and relationships to achieve its objectives |
| Objective | Result to be achieved |
| Asset | Any resource that has value and importance to the owner |
| Threat | Potential cause of an unwanted incident, which may result in harm to a system or organization |
| CIA | The security properties, i.e., confidentiality (C), integrity (I) and availability (A) to be ensured |
| Confidentiality | Property that information is not made available or disclosed to unauthorized individuals, entities, or processes |
| Integrity | Property of accuracy and completeness |
| Availability | Property of being accessible and usable on demand by an authorized entity |
| Risk | Effect of uncertainty on objectives |
| Level of risk | Magnitude of a risk expressed in terms of the combination of consequences and their likelihood |
| Consequence | Outcome of event affecting objectives |
| Likelihood | Chance of something happening<br>Chance of an event to occur (Agrawal, 2016) |
| Event | Occurrence or change of a particular set of circumstances |
| Control | Measure that is modifying risk |
| Vulnerability | Weakness of an asset or control that can be exploited by one or more threats |

Finally, for the ontology to be selected for the construction of the integrated ontology, it must comprise of classes that can be mapped to the key artifacts (previously extracted from the selected processes, activities, and tasks discussed in Chapter 4). Therefore, we mapped the extracted key artifacts to the classes of the ontology. It is also important to note that according to Agrawal (2016) the concept event is also referred to as a security incident. As a result, we mapped the key artifact incident to the class event. Table 5.3 provides the mapping.

Table 5.3 Mapping of extracted key
artifacts and classes of the ontology

| **Extracted key artifacts** | **Classes** |
|---|---|
| Asset | Asset |
| Threat | Threat |
| Control | Control |
| Vulnerability | Vulnerability |
| Consequence | Consequence |

| | |
|---|---|
| Incident | Event |
| Likelihood | Likelihood |
| Risk | Risk |
| Level of risk | |

Based on the mapping, we found that eight out of the nine key artifacts related to information security risk management were covered by the ontology. As a result, we selected the ontology for the construction of the integrated ontology. Additionally, we were able to extract from ISO/IEC 27005 (ISO/IEC, 2018) and ISO/IEC 27000 (ISO/IEC, 2018) the missing concept, namely, level of risk, and its definition to extend the ISO/IEC 27005 information security risk management ontology from Agrawal (2016). Figure 5.2 provides the extended ISO/IEC 27005 information security risk management ontology. Table 5.4 shows the definition of the class level of risk.

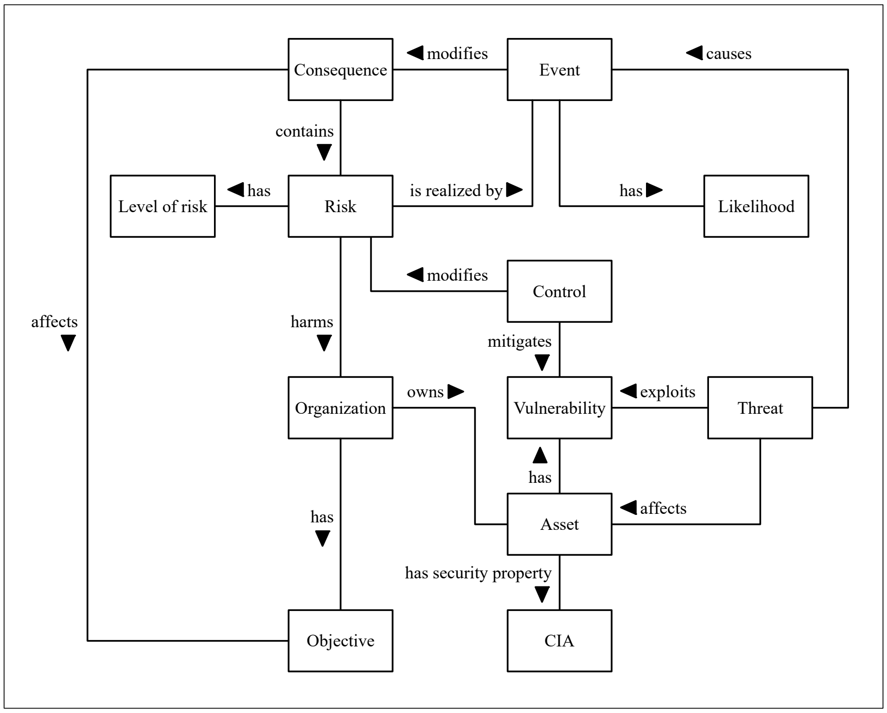


Figure 5.2 The extended ISO/IEC 27005
information security risk management ontology
adapted from Agrawal (2016)

Table 5.4 Definition taken from ISO/IEC 27005 (ISO/IEC, 2018) and ISO/IEC 27000 (ISO/IEC, 2018)

| Class | Extracted definition |
|---|---|
| Level of risk | Magnitude of a risk expressed in terms of the combination of consequences and their likelihood |

### 5.3.3 Information security controls ontology

Milicevic & Goeken (2010) used ontological metamodeling to obtain the ISO/IEC 27001 structure. To support the construction process of the metamodel, their study applied concepts from GT and QDA. The basic idea in GT (as with most QDA methods) is to use empirical data like transcripts from interviews, protocols, and documents that a researcher confronts in the field. The focus is on the inductive development of a theory, which is "grounded" in the respective empirical data. One central activity is the "coding", which means conceptualizing qualitative data and assigning categories as well as the relationships between them. Milicevic & Goeken (2010) used the inductive categorization method to derive relevant ontological metamodel components for M2 from the ISO/IEC 27001 standard, which is located on a model level (M1). They further used QDA software (ATLAS.ti) for coding and UML as a modeling language. Figure 5.3 provides the ISO/IEC 27001 ontology extracted from Milicevic & Goeken (2010).

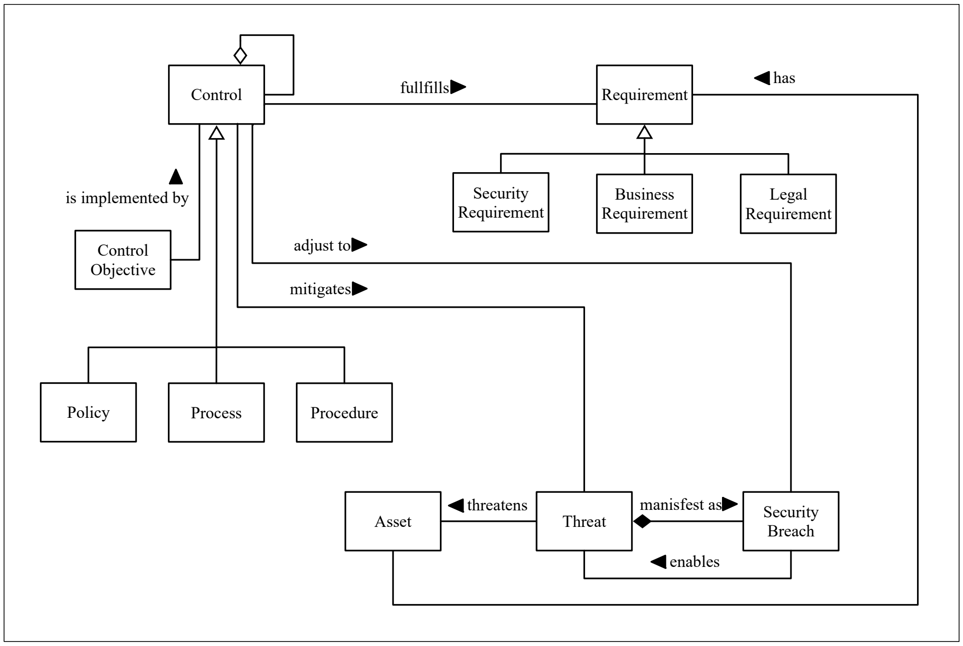


Figure 5.3 The ISO/IEC 27001
information security control ontology
taken from Milicevic & Goeken (2010)

As previously shown in section 5.3.2, Agrawal (2016) defined each ontology class through a definition. However, the Milicevic & Goeken (2010) ontology does not provide such definitions. Given that both ontologies were extracted from the same standard series, we refer to the definitions previously extracted for all common classes. As for the classes that are found exclusively in the Milicevic & Goeken (2010) ontology, namely, policy, requirement, process, and control objective, we used the definitions from the same sources, namely, ISO/IEC 27005 (ISO/IEC, 2018) and ISO/IEC 27000 (ISO/IEC, 2018). Table 5.5 provides the extracted definitions.

Table 5.5 Definitions taken from ISO/IEC 27005 (ISO/IEC, 2018) and ISO/IEC 27000 (ISO/IEC, 2018)

| **Classes** | **Extracted definitions** |
|---|---|
| Policy | Intentions and direction of an organization, as formally expressed by its top management |
| Requirement | Need or expectation that is stated, generally implied or obligatory |
| Process | Set of interrelated or interacting activities which transforms inputs into outputs |
| Control objective | Statement describing what is to be achieved as a result of implementing controls |

Finally, for the ontology to be selected for the construction of the integrated ontology, it must comprise of classes that can be mapped to the key artifacts (previously extracted from the selected processes, activities, and tasks discussed in Chapter 4). Therefore, we mapped the extracted key artifacts to the classes of the ontology. It is important to note that while Milicevic & Goeken (2010) identified the concepts security incident, security event, and security breach, these concepts were merged under security breach in their ontology. As a result, we mapped the key artifact incident to the class security breach. Table 5.6 provides the mapping.

Table 5.6 Mapping of extracted key
artifacts and classes of the ontology

| **Extracted key artifacts** | **Classes** |
|---|---|
| Asset | Asset |
| Threat | Threat |
| Control | Control |
| Vulnerability | |
| Consequence | |
| Incident | Security breach |
| Likelihood | Likelihood |
| Risk | Risk |
| Level of risk | |

Based on the mapping, we found that six out of the nine key artifacts related to information security control were covered by the ontology. As a result, we selected the ontology for the construction of the integrated ontology. However, we were unable to extract the missing concepts and their relationships from ISO/IEC 27001 to extend the ontology.

### 5.3.4 Cloud computing service information security control ontology

In the case of ISO/IEC 27017 and ISO/IEC 27018, our initial search revealed no existing ontology in the literature. However, the ISO/IEC 27001, ISO/IEC 27017, and ISO/IEC 27018 standards are based on ISO/IEC 27002; consequently, their structure is the same except for one structural addition in the case of ISO/IEC 27017 (ISO/IEC, 2015), namely, cloud service customer and provider implementation guidance, and one structural addition in the case of ISO/IEC 27018 (ISO/IEC, 2019), namely, public cloud PII protection implementation guidance.

As shown in the preceding chapter, ISO/IEC 27017 cloud computing service information security controls are applicable to either the cloud service provider, the cloud service customer, or both. ISO/IEC 27017 offers guidance on how a cloud service customer and cloud service provider can implement, manage, and operate information security for a cloud computing service. When objectives and controls specified in ISO/IEC 27002 are applicable without the need for any additional information, only a reference is provided to ISO/IEC 27002. When a control requires additional cloud computing service implementation guidance, it is given under the subtitle “implementation guidance for cloud services”. Additional controls and associated implementation guidance applicable to cloud computing services for cloud service providers and cloud service customers are described in Annex A of the standard. As it pertains to ISO/IEC 27018, the cloud computing service information security controls for the protection of PII in public clouds are applicable to the provider of public cloud services that act as a PII processor. When objectives and controls specified in ISO/IEC 27002 are applicable without a need for any additional information, only a reference is provided to ISO/IEC 27002. When a control requires additional PII protection for cloud service provider implementation guidance, it is given under the subtitle "public cloud PII protection implementation guidance". Additional controls and associated implementation guidance applicable to PII protection for cloud service providers are described in Annex A of the standard. The different permutations are provided in Tables 5.7 and 5.8.

Table 5.7 ISO/IEC 27002 and ISO/IEC 27017 cloud computing service controls applicability and implementation guidance

| **ISO/IEC 27002** information security control applicable to a cloud service | **ISO/IEC 27017** cloud service specific information security control | **ISO/IEC 27017** cloud service customer implementation guidance | **ISO/IEC 27017** cloud service provider implementation guidance |
|---|---|---|---|
| X | | | |
| X | | X | |
| X | | | X |
| X | | X | X |
| X | | Same implementation guidance | |
| | X | | |
| | X | X | |
| | X | | X |
| | X | X | X |
| | X | Same implementation guidance | |

Table 5.8 ISO/IEC 27002 and ISO/IEC 27018 cloud computing services controls applicability and implementation guidance

| **ISO/IEC 27002** information security control applicable to a cloud service | **ISO/IEC 27018** public cloud PII processor specific information security control for PII protection | **ISO/IEC 27018** public cloud PII protection implementation guidance |
|---|---|---|
| X | | |
| X | | X |
| | X | |
| | X | X |

Moreover, as shown in the preceding chapter, ISO/IEC 27036 (ISO/IEC, 2014) (i.e., all components) provides further detail regarding the specific requirements to be used to establish and monitor information security for supplier relationships, including the cloud service customer and provider relationship. It stipulates that "when information security controls provided by the cloud service provider are preset and cannot be changed by the cloud service customer, the cloud service customer may be required to implement his or her own additional controls to mitigate potential risks."

Consequently, this research project extended the Milicevic & Goeken (2010) ontology to include these structural additions in order to construct the ISO/IEC 27017 and ISO/IEC 27018 cloud computing services information security control ontology (as shown in Figure 5.4). The extension comprises of the classes cloud service customer and cloud service provider and the relationships *IsImplementedBy* and *IsPre-SetBy*.

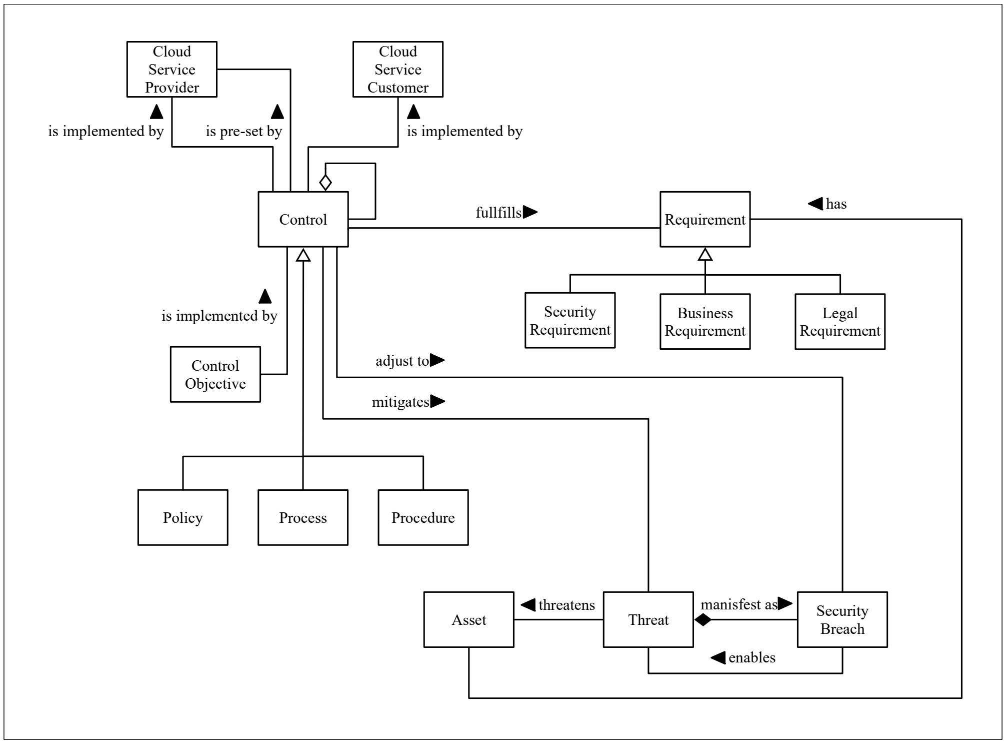


Figure 5.4 The ISO/IEC 27017 and ISO/IEC 27018
cloud computing services information security control ontology
adapted from Milicevic & Goeken (2010)

Following the Agrawal (2016) methodology, we defined each ontology class within the extension with definitions extracted from ISO/IEC 17788 (ISO/IEC, 2014). Table 5.9 provides the extracted definitions.

Table 5.9 Definitions taken from ISO/IEC 17788 (ISO/IEC, 2014)

| **Classes** | **Extracted definitions** |
|---|---|
| Cloud service provider | Party which makes cloud services available |
| Cloud service customer | Party which is in a business relationship for the purpose of using cloud services |
| Party | Natural person or legal person, whether or not incorporated, or a group of either |
| Cloud service | One or more capabilities offered via cloud computing invoked using a defined interface |
| Cloud computing | Paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on-demand |

Additionally, we analyzed the ISO/IEC 27017 (ISO/IEC, 2015) and ISO/IEC 27002 (ISO/IEC, 2013) standards to determine the presence of additional concepts and relationships. To do so, we selected the development ontology methodology used in Agrawal (2016) given that it is based on the well-established methodology by Noy & McGuinness (2001).

The first step is to determine the domain and the scope of the ontology. In this case, the domain is information security and the standards in the scope of the ontology are ISO/IEC 27002, ISO/IEC 27017, and ISO/IEC 27018.

The second step is to consider the reuse of existing ontologies. For this step, as previously discussed in section 5.3.1, our search revealed no existing ontology in the literature for ISO/IEC 27002, ISO/IEC 27017, and ISO/IEC 27018 that met our criteria.

The third step is to capture the key terms that describe the concepts of ISO/IEC 27002, ISO/IEC 27017, and ISO/IEC 27018. In this case, our objective was to determine the presence of additional concepts and relationships to complete the ISO/IEC 27017 and ISO/IEC 27018 ontology adapted from Milicevic & Goeken (2010). Therefore, our starting point was the concepts that comprise the said ontology. Table 5.10 provides a list of concepts extracted from the ISO/IEC 27017 and ISO/IEC 27018 ontologies adapted from Milicevic & Goeken (2010).

Table 5.10 List of concepts extracted from the ISO/IEC 27017 and ISO/IEC 27018 ontology

| **Concepts** |
|---|
| Control |
| Control objective |
| Policy |
| Process |
| Procedure |
| Asset |
| Threat |
| Security breach |
| Requirement |

| |
|---|
| Security requirement |
| Business requirement |
| Legal requirement |
| Cloud service provider |
| Cloud service customer |

The fourth step is to define the classes and the class hierarchy. For this step, each class was defined through a definition. Given that our starting point comprised of the ISO/IEC 27017 and ISO/IEC 27018 ontology adapted from Milicevic & Goeken (2010), these classes were previously defined in sections 5.3.2, 5.3.3, and 5.3.4. Tables 5.2, 5.4, 5.5, and 5.9 provide the classes and their definitions.

The fifth step is to define the object properties of the classes. For this step, we identified the object properties that relate a class to another class. Standards vary in scope and purpose, while they also vary in the level of detail and granularity. After examining the full text of the standards, we found that concepts were defined at a granular level, and that the text of the standards provided the relationships between classes. Therefore, we searched the full text to identify quotations that comprise the classes and their relationships. Table 5.11 provides the extracted quotation, classes, and relationships.

Table 5.11 Extracted quotation and results of the iteration

| **Classes** | **Extracted quotation** | **Additional classes extracted** | **Extracted relationships** |
|---|---|---|---|
| Control<br>Process<br>Procedure<br>Policy | Information security is achieved by implementing a suitable set of controls, including policies, processes, procedures, organizational structures and software and hardware functions. | Organizational structure<br>Software function<br>Hardware function<br>Function | Is a |

When we examined the extracted quotation, we found that it comprised of additional terms that represent relevant class entities. Therefore, for this quotation, we conducted an iteration of the third and fifth steps. Table 5.11 provides the additional classes extracted. Then, for each additional class, namely, organizational structure, software function, and hardware function, we conducted an iteration of the fourth and fifth steps to identify additional quotations that comprise the additional classes and their relationships. Table 5.12 provides the extracted quotation and results of the iteration.

Table 5.12 Extracted quotation and results of the iteration

| **Class** | **Extracted quotation** | **Additional classes extracted** | **Extracted relationship** |
|---|---|---|---|
| Control Software function Function | "…segregation of cloud customers in software virtualized environment can be designed and implemented using segregation functions of the software…" | | Implemented using |

When we examined the extracted quotation, we found that it only comprised of terms that represent existing class entities. As additional classes and their relationships were found, we extended the ISO/IEC 27017 and ISO/IEC 27018 ontology adapted from the Milicevic & Goeken (2010) ontology in order to construct the cloud computing services information security control ontology. The extension comprises of the classes organizational structure, function, software function, and hardware function (as shown in Figure 5.5).

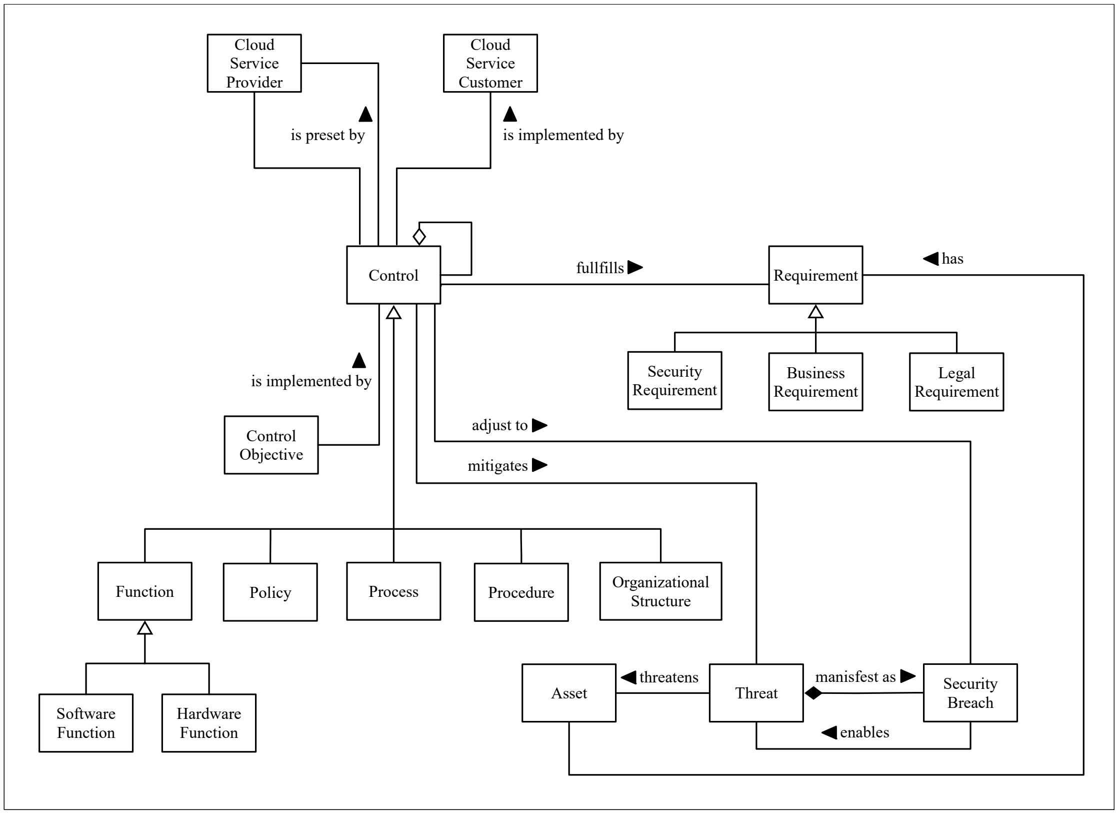


Figure 5.5 The cloud computing services information security control ontology

Finally, we defined the additional classes of the ontology. However, to the best of our knowledge the ISO/IEC 27000 series does not provide class definitions for organizational structure, function, software function, and hardware function.

## 5.4 The quality requirements definition framework ontology

The ISO/IEC 25030 systems and software quality requirements definition framework provides the processes, including activities and tasks, to define ICT products, IT services, and data quality requirements using ISO/IEC 25010, ISO/IEC 25011, and ISO/IEC 25012 quality models and quality measures.

### 5.4.1 Existing ontologies

We found the following relevant studies during our initial search to identify existing ISO/IEC 25030 ontologies: Castillo et al. (2010) and Cherfi et al. (2011). We evaluated each study using the selection criteria previously presented in section 5.2.1.

In their study, Castillo et al. (2010) integrated the aspect-oriented software development (AOSD) technology, classic requirements engineering approaches, and the standard ISO/IEC 25030 on software quality requirements. The conceptual REquirements, Aspects and Software Quality (REASQ) model was the primary achievement of this study, which was expressed in UML. Aspect-oriented programming (AOP) and software quality and requirements engineering concepts were formalized into three related ontologies. This study created instances of each class based on a case scenario to demonstrate that the ontologies can be used comprehensively to specify quality requirements during AOP processes. However, software quality ontological metamodel components are limited to the concepts of the quality models used in the definition of quality requirements. Finally, no methodology was provided on the development of the ontology.

In their study, Cherfi et al. (2011) proposed and discussed the main constituents of an ontology of quality that federates all aspects of IS component quality (i.e., software, data, models, etc.). Three ontology definition levels were developed, namely, the domain independent quality ontology, the interrelated domain specific quality ontology, and the operational quality ontology. This study used a process that consists of the exploration of sources of knowledge that involve "actors" at each definition level. The ontologies comprise of the concepts that result from this process. This study also provides scenarios to illustrate the usage of the proposed ontologies. However, software and data quality ontological metamodel components are limited to the concepts of the quality models used in the definition of quality requirements.

Findings showed that the existing ontologies do not meet the selection criteria for this study. Table 5.13 provides the evaluation results. Consequently, the ontological metamodeling of the ISO/IEC 25030 quality requirements definition framework is necessary for this research project as our objective is to integrate the said framework to information security risk management, cloud computing services information security control, and system security requirements definition frameworks on a level of abstraction.

Table 5.13 List of existing ISO/IEC 25030 ontologies

| Ontologies | ISO/IEC 25000 series standards | Selection criterion | | |
|---|---|---|---|---|
| | | 1 | 2 | 3 |
| Castillo et al. (2010) | ISO/IEC 25030 | | | X |
| Cherfi et al. (2011) | ISO/IEC 25030 | | X | X |

### 5.4.2 Quality requirements definition framework ontology

As discussed in section 5.2.1, the Schuette & Rotthowe (1998) principle of comparability must be applied in the ontological metamodeling of the ISO/IEC 25030 quality requirements definition framework. This is because its ontology must be compared to that of the cloud computing services information security risk management and system security requirements definition frameworks prior to their integration. In order to apply this principle in the construction of the ISO/IEC 25030 quality requirements definition framework ontology, we selected the same modeling language used in the studies by Agrawal (2016) and Milicevic & Goeken (2010). Additionally, we selected the development ontology methodology used in Agrawal (2016) given that it is based on the well-established methodology by Noy & McGuinness (2001).

The first step is to determine the domain and the scope of the ontology. In this case, as discussed in the preceding chapter, the domain is the system and software quality requirements definition, and Table 4.11 shows the methods under the scope of the ontology.

The second step is to consider the reuse of existing ontologies. For this step, as previously discussed in section 5.4.1, our search revealed that ontologies in the literature are limited to the concepts of the quality models used in the quality requirements definition.

The third step is to capture the terms that are key in describing the concepts of the system and software quality requirements definition. Given our design objectives (previously presented in section 5.2), for the ontology to be selected for the construction of the integrated ontology, it must comprise of classes that can be mapped to the key artifacts (previously extracted from the selected processes, activities, and tasks discussed in Chapter 4). Therefore, for this step, we captured the terms that describe the extracted key artifacts. Table 5.15 shows the list of extracted terms where each term represents a relevant class entity.

Table 5.14 List of extracted terms
from ISO/IEC 25030 (ISO/IEC, 2019)
methods in the scope of the ontology

| **Extracted terms** |
|---|
| Quality characteristic |
| Quality requirement |
| Product quality requirement |
| Data quality requirement |
| Quality measure |
| Required criteria |

The fourth step is to define the classes and the class hierarchy. Each class is defined through a definition extracted from ISO/IEC 25030 (ISO/IEC, 2019) and associated quality models and quality measures. When definitions were unavailable, we constructed them by adapting and combining definitions of related terms. Table 5.16 provides the classes and their definitions.

Table 5.15 Extracted definitions, classes, and relationships

| **Classes** | **Extracted definitions** |
|---|---|
| Quality characteristic | A category of quality attributes that bears on ICT product quality, data quality, and IT service quality where quality characteristic may be refined into multiple levels of sub-characteristics and finally into quality attributes |
| Quality requirement | A requirement for quality properties or attributes of an ICT product, data or IT service that satisfy needs which ensue from the purpose for which that ICT product, data or service is to be used |
| Product quality requirement | A quality requirement that specify levels of quality required from the viewpoint of the ICT product |
| Data quality requirement | A quality requirement that specify levels of quality required for the data associated with the product |
| Quality measure | A measure that is defined as a measurement function of two or more values of quality measure elements |
| Required criteria | i.e., target entity, selected characteristic, user and task (only for QIURs), quality goal with conditions, quality measure, target value, acceptable range of values |

The fifth step is to define the object properties of the classes. For this step, we identified the object properties that relate a class to another class. Standards vary in scope and purpose, while they also vary in the level of detail and granularity. After fully exploring the text of the ISO/IEC 25030 (ISO/IEC, 2019) standard, we determined that the level of the definition of the quality requirements definition concepts was granular and that the text of the framework included the relationships between classes. Therefore, we conducted a search on the full text to identify quotations that comprise the classes and their relationships. Table 5.17 provides the extracted quotations and relationships.

Table 5.16 Extracted quotations and iteration results

| **Classes** | **Extracted quotations** | **Additional classes extracted** | **Extracted relationships** |
|---|---|---|---|
| Quality characteristic | Quality required for one ICT product is different from that for another, and therefore the category of the target system is crucially important to determine which quality characteristics have higher priority and which quality measures should be used. | ICT product<br>Category<br>Quality measure | Determine the priority of |
| Requirement | Two types of requirements for ICT products should be considered based on their sources: domain-based requirements, which are derived directly from stakeholder needs for their domain through requirements analysis processes, and ICT requirements, which are newly introduced by the adoption of some ICT technical solutions through design processes | ICT product<br>Domain-based requirement<br>ICT requirement<br>ICT technical solution | Is a source of<br>Derive<br>Introduced by |
| Quality requirement | Functional requirements to implement quality requirements, e.g., security requirement → access control functional requirements | Functional requirement<br>Quality requirement<br>Security requirement<br>Access control | Implement |
| | Architecture to implement quality requirements, e.g., fault tolerant requirement → fault tolerant architecture | Architecture | Implement |
| | Some quality requirements for the service provision systems are derived from IT service quality requirements for the target service and QIURs for their context of use | Service provision system<br>IT service quality requirement<br>Service<br>Quality in use requirement<br>Context of use | Derive<br>Is a target of |
| | Some quality requirements are attached to functional/data requirements, and also some quality requirements are achieved by specifying requirements for new functions | Data requirement<br>Requirement<br>Function | Attached to<br>Achieved by |
| | Quality requirements that are achieved by specifying requirements for new functions | Requirement<br>Function | Achieved by |

| **Classes** | **Extracted quotations** | **Additional classes extracted** | **Extracted relationships** |
|---|---|---|---|
| Quality requirement | Quality requirements that are attached to functional requirements | Functional requirement | Attached to |
| | Some quality requirements for the service provision systems are derived from IT service quality requirements for the target service and QIURs for their context of use | Service provision system<br>IT service quality requirement<br>Quality in use requirement | Derive<br>Is a target of |
| | Other stakeholders, such as developers and regulatory bodies, also give some quality requirements on the target entities | Other stakeholders<br>Developers<br>Regulatory bodies | Give |
| | The primary source of quality requirements is users, from whom first QIURs for the information system including the target entities are elicited and documented | User<br>Quality in use requirement<br>Information system<br>Entity | Is a source of<br>Is a target of |
| | Specify quality requirement using quality measures and their required criteria… target entity, selected characteristic, user and task (only for QIURs), quality goal with conditions, quality measure, target value, acceptable range of values | Entity<br>Quality characteristic<br>User<br>Task<br>Quality in use requirement<br>Quality goal with conditions<br>Quality measure<br>Target value<br>Acceptable range of values | Is a target of<br>Is specified by<br>Perform |
| Product quality requirement | PQRs are defined on the ICT product or its constituents (including sub-ICT products, hardware, communication facilities, software, and in some case software components) | ICT product<br>Hardware<br>Communication Facilities<br>Software<br>Software components | Is a target of<br>Is part of |
| Product quality requirement | A QIUR can derive a set of functions to each of which some PQRs are attached, e.g., an efficiency requirement of a user task can derive a function to automate some portion of the task with a time efficiency requirement | Function<br>Quality in use requirement | Derive<br>Attached to |

| Classes | Extracted quotations | Additional classes extracted | Extracted relationships |
|---|---|---|---|
| Product quality requirement | PQRs can also derive a set of functions to each of which some PQRs are attached | Function | Derive<br>Attached to |
| | Product quality requirements (PQRs) specify levels of quality required from the viewpoint of the ICT product. Most of them are derived from stakeholder quality requirements including QIURs, which can be used as targets for verification and validation of the target ICT product | ICT product<br>Stakeholder<br>Quality requirement<br>Quality in use requirement | Derive<br>Is a target of<br>Has |
| Product quality requirement | A QIUR can imply several PQRs | Quality in use requirement | Imply |
| Data quality requirement | DQRs are defined on the data inside the ICT product. | Data<br>ICT product | Is the target of<br>Is part of |
| Data quality requirement | DQRs such as data integrity can be derived directly from QIURs | Quality in use requirement | Derive |
| Data quality requirement<br>Product quality requirement | DQRs can be derived from PQRs for the target product | Product | Derive<br>Is a target of |

When examining the extracted quotations, we found that they comprised of additional terms that represent relevant class entities. Therefore, for each quotation, we conducted iterations of the third and fifth steps. Table 5.17 provides the additional extracted classes. Then, for each additional class, we conducted iterations of the fourth and fifth steps to identify additional quotations. Table 5.18 provides the extracted quotations and iteration results.

Table 5.17 Extracted quotations and iteration results

| **Classes** | **Extracted quotations** | **Additional classes extracted** | **Extracted relationships** |
|---|---|---|---|
| Quality in use requirement<br>Stakeholders | Quality in use requirements (QIURs) specify the required levels of quality from the stakeholders' point of view. These requirements are derived from the needs of various stakeholders. | Needs | Derive<br>Has |
| Functional requirement<br>Function<br>Task | Functional requirements describe the system or system element functions or tasks to be performed | System<br>System element | Describe |
| Access control<br>Function | Some confidentiality requirements are achieved by requirements for access control function | Confidentiality requirement | Achieved by |
| ICT product<br>User<br>Information system | QIURs are defined on the information system, which includes not only an ICT product but also its users and relevant environments (e.g., mechanicals monitored / controlled by the ICT product and business processes in which the ICT product is used) | Relevant environment | Is part of |
| IT service<br>Service provision system | IT service has its own provision system which consists of people, process, technology, facilities and information, and the IT service quality model can be partly used to measure the quality of service provision system | People<br>Process<br>Technology<br>Facilities<br>Information | Has<br>Is part of |

When examining the extracted quotations, we found that they comprised of additional terms that represent relevant class entities. Therefore, for each quotation, we conducted iterations of the third and fifth steps. Table 5.18 provides the additional extracted classes. Then, for each additional class entity, we conducted iterations of the fourth and fifth steps to identify additional quotations. This iteration showed no additional quotations. The construction of the ontology proceeded after no additional quotations were identified to extract class entities. Figure 5.6 provides the resultant ISO/IEC 25030 quality requirement framework ontology.

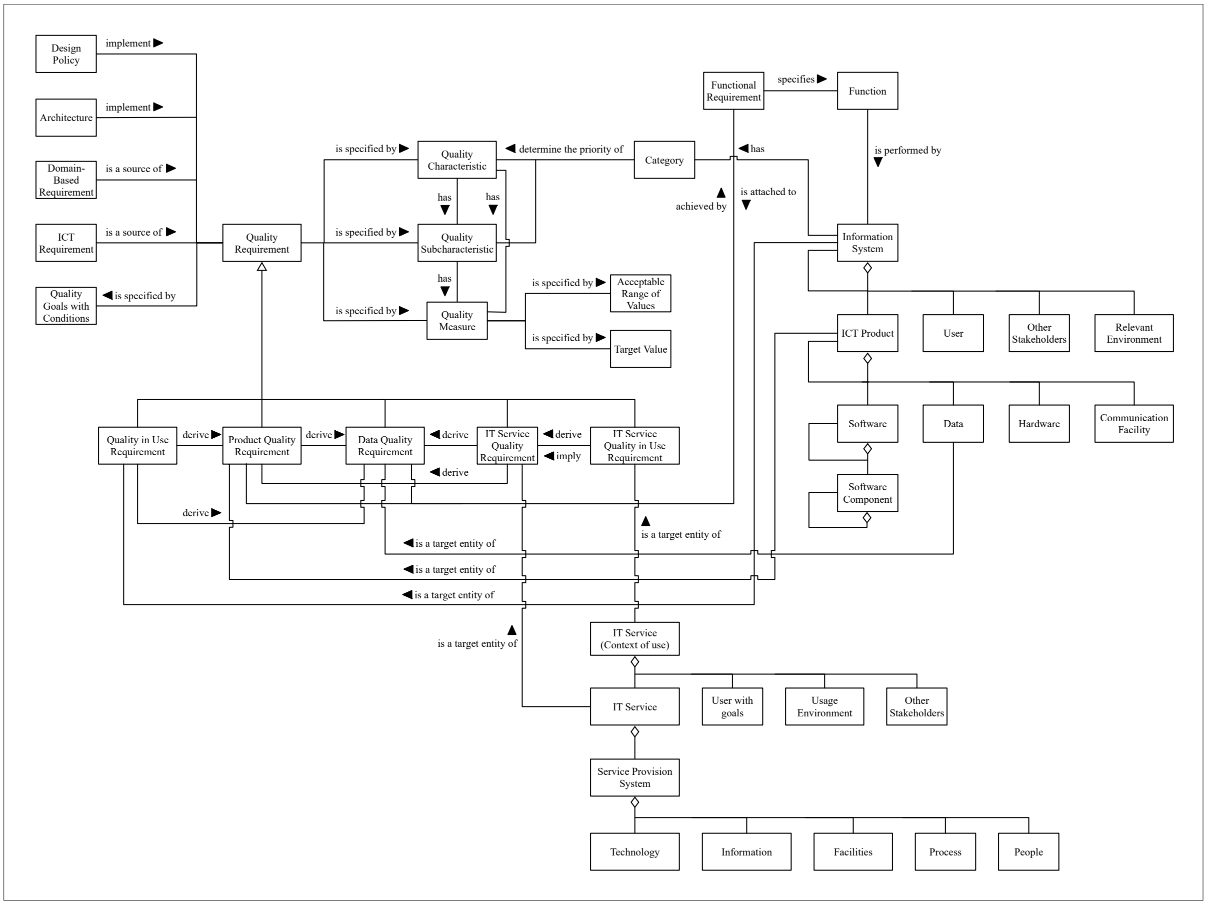


Figure 5.6 The ISO/IEC 25030 quality requirement definition framework ontology

During quote examination, we found that the class access control was solely grounded in quotations that provide an example of the application of quality requirements to information security, namely, "Some confidentiality requirements are achieved by requirements for access control function" and "Functional requirements to implement quality requirements, e.g., security requirement → access control functional requirements". Given that our objective is to construct an integrated ontology that explicates relationships between the concepts involved in the definition of quality requirements related to information security in cloud computing services, we extended the ISO/IEC 25030 quality requirement framework ontology to the information security domain. The extension comprised of the class control. Figure 5.7 provides the resultant ISO/IEC 25030 quality requirements framework ontology extended to the information security domain.

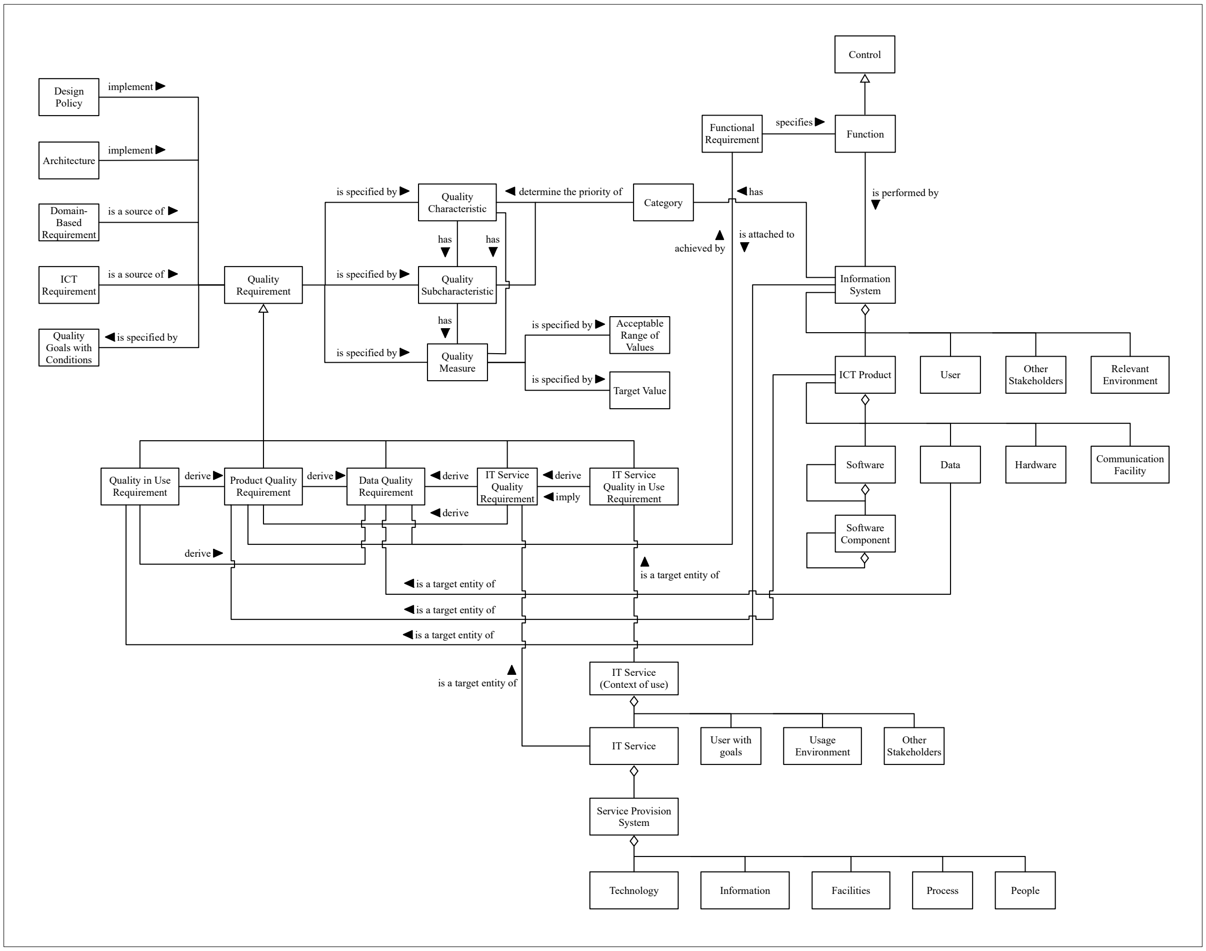


Figure 5.7 The ISO/IEC 25030 quality requirements framework ontology extended to the information security domain

Finally, as per the Noy & McGuinness (2001) methodology, we defined each ontology class extracted in the preceding iterations. Each class was defined through a definition extracted from ISO/IEC 25030 and its associated quality models and quality measures. When no definitions were available, we constructed them using quotations that comprised of the classes or by adapting and combining definitions of related terms. Table 5.19 provides the classes and their definitions.

However, in the case of the classes' function and design policy, to the best of our knowledge neither the ISO/IEC 25000 series nor ISO/IEC/IEEE 15288 provide their definitions. As the preceding chapter showed, while the ISO/IEC 25023 (ISO/IEC, 2016) standard includes the term function within the information security domain, namely, security function, the standard does not provide its definition. Additionally, no quotations could be found that comprise of the classes or definitions of related terms that can be used for their construction.

Table 5.18 Definitions extracted from ISO/IEC 25030 (ISO/IEC, 2019)

| **Classes** | **Extracted definitions** |
|---|---|
| Category | Categorization of the target system which determine the quality characteristics with higher priority and which quality measures should be used |
| Service provision system | System to provide IT service to users, including people, process, technology, facilities and information |
| Domain-based requirement | Requirement originating from its application domain |
| ICT requirement | Requirement resulting from adoption of some Information and Communication Technologies (ICTs) technical solutions in the design process |
| Quality in use requirement | Quality requirement that specify the required levels of quality from the stakeholders' point of view |
| Technical product quality requirement | Product quality requirement for technically identified properties or attributes (targeting specifications, source code, etc.) which are used in its development and maintenance processes to meet the other product quality requirements - adapted |
| IT service quality requirement | Requirement for quality properties or attributes of an IT service that satisfy needs which ensue from the purpose for which that IT service is to be used |
| IT service quality in use requirement | Quality requirement of an IT service that specify the required levels of quality from the stakeholders’ point of view |
| Architecture | System fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution |
| Design policy | Not available |
| Information system | System that comprises of software, hardware, communication facility, data, and the people who use it in a given environment to satisfy their information processing needs |
| Data | Reinterpretable representation of information in a formalized manner suitable for communication |
| ICT product | Product which uses Information and Communication Technologies (ICTs), and can be a part of information system |
| IT service | Service that makes use of IT systems as tools to support the needs of an individual user or a business |

| Classes | Extracted definitions |
|---|---|
| User | Individual or group that interacts with a system or benefits from a system during its utilization |
| Stakeholder | Individual or organisation having a right, share, claim or interest in a system or in its possession of characteristics that meet their needs and expectations |
| Functional requirement | Requirement that specifies a function that a system or system component shall perform |
| Function | Not available |

## 5.5 The system security requirements definition ontology

In this section, our objective is to obtain the ISO/IEC/IEEE 15288 system requirements definition and NIST SP 800-160 system security requirements definition ontology. We initially conducted a search to identify existing ontologies. The search revealed that there was no existing ontology in the literature. Consequently, we proceeded with the ontological metamodeling of the ISO/IEC/IEEE 15288 system requirements definition and NIST SP 800-160 system security requirements definition ontology domain.

As discussed in section 5.2.1, the Schuette & Rotthowe (1998) principle of comparability must be applied for the ontological metamodeling of the NIST SP 800-160 system security requirements definition framework. This is because its ontology must be compared to that of the cloud computing services information security risk management and quality requirements definition frameworks prior to their integration. In order to apply this principle in the construction of the NIST SP 800-160 system security requirements definition framework ontology, we selected the same modeling language used in the studies by Agrawal (2016) and Milicevic & Goeken (2010). Additionally, our selection of the ontology development methodology used in Agrawal (2016) was conditional on it being based on the well-established methodology by Noy & McGuinness (2001). The methodology consists of the following seven steps:

The first step is to determine the domain and the scope of the ontology. In this case, as determined in the preceding chapter, the domain is the system requirements definition and system security requirements definition, and Table 4.11 shows the methods under the scope of the ontology.

The second step is to consider the reuse of existing ontologies. For this step, as previously discussed in section 5.2.1, our search revealed no existing ontology in the literature.

The third step is to capture the terms that are key to describing the concepts of the system requirements definition and system security requirements definition. Given our design objectives (previously presented in section 5.2), for the ontology to be selected for the construction of the integrated ontology, it must comprise of classes that can be mapped to the key artifacts (previously extracted from the selected processes, activities, and tasks discussed in Chapter 4). Therefore, for this step, we captured the terms that describe the extracted key artifacts, with the exception of the key artifact risk previously covered by the ISO/IEC 27005 information security risk management ontology class risk as well as the key artifact function, criticality of the system, and critical quality characteristics, previously covered by the ISO/IEC 25030 quality requirement definition framework ontology class function, category of the system, and quality characteristic. Table 5.21 provides the list of extracted terms, where each term represents a relevant class entity.

Table 5.19 List of extracted terms

| **Extracted terms** |
|---|
| Implementation constraint |
| System requirement |
| Critical performance measure |
| Stakeholder |
| Security function |
| System |
| Security requirement |
| System security requirement |
| Security constraint |
| System requirement |
| Security concern |
| Security-driven performance measure |
| Security-driven assurance measure |

The fourth step is to define the classes and the class hierarchy. Each class is defined through a definition extracted from the NIST SP 800-160 (NIST, 2018) report. Table 5.22 provides the extracted definitions.

The fifth step is to define the object properties of the classes. For this step, we identified the object properties that relate a class to another class. As mentioned in the preceding section, standards vary in scope and purpose, while they also vary in the level of detail and granularity. After examining the full text of the NIST SP 800-160 (NIST, 2018) report, we found that relationships between classes were included in the term definitions of the framework, which are available as annotations in the text and/or in the terms and definitions provided in Appendix B. Therefore, we then searched text annotations and Appendix B of the NIST SP 800-160 (NIST, 2018) report to identify the definitions that comprise of the classes and extracted their relationships. Table 5.20 provides the extracted definitions and relationships.

Table 5.20 Extracted definitions, classes, and relationships

| **Classes** | **Extracted definitions** | **Extracted classes** | | **Extracted relationships** |
|---|---|---|---|---|
| | | **Existing** | **Additional** | |
| Security function | The capability provided by the system or a system element. The capability may be expressed generally as a concept or specified precisely in requirements. | System | System element<br>Requirement | Is provided by<br>Is specified in |
| System | Combination of interacting elements organized to achieve one or more stated purposes. | | | |
| Security requirement | A requirement that specifies the functional, non-functional, assurance, and strength characteristics for a mechanism, system, or system element. | System | Mechanism<br>System element | Specifies |

| Classes | Extracted definitions | Extracted classes | | Extracted relationships |
|---|---|---|---|---|
| | | Existing | Additional | |
| System requirement | A condition or capability that must be met or possessed by a system or system element to satisfy a contract, standard, specification, or other formally imposed documents. | System | System element | Is possessed by |
| System security requirement | System requirements that have security relevance. System security requirements define the protection capabilities provided by the system, the performance and behavioral characteristics exhibited by the system, and the evidence used to determine that the system security requirements have been satisfied. | System requirement<br>System | | Provided by<br>Exhibited by |
| | Each system security requirement is expressed in a manner that makes verification possible via analysis, observation, test, inspection, measurement, or other defined and achievable means. | | Measurement | Is verified via |

When examining the extracted definitions, we found that they comprised of additional terms that represent relevant class entities. Therefore, for each definition, we conducted iterations of the third and fifth steps. Table 5.22 provides the additional classes.

Then, for each additional class, namely, system element, requirement, and mechanism, we conducted iterations of the fourth and fifth steps. Table 5.23 provides the extracted definitions and results of the iterations.

Table 5.21 Extracted definitions and iteration results

| **Classes** | **Extracted definitions** | **Extracted classes** | | **Extracted relationships** |
|---|---|---|---|---|
| | | **Existing** | **Additional** | |
| System element | Member of a set of elements that constitute a system. | System | | Constitute |
| Requirement | Statement that translates or expresses a need and its associated constraints and conditions. | | | |
| Mechanism | A process or system that is used to produce a particular result.<br>The fundamental processes involved in or responsible for an action, reaction, or other natural phenomenon.<br>A natural or established process by which something takes place or is brought about. Refer to *security mechanism*. | | Security mechanism | |
| | Security control: A mechanism designed to fulfill a set of security requirements. | Security requirement | Security control | Is a |
| Measurement | Set of operations having the object of determining a value of a measure. | | Measure | Determine the value of |

When examining the extracted definitions, we found that two of them comprised of additional terms that represent relevant class entities, namely, security mechanism, security control, and measure. Therefore, for these additional classes, we conducted iterations of the fourth and fifth steps. Table 5.24 provides the extracted definitions and relationships.

Table 5.22 Extracted definitions and iteration results

| Class | Extracted definitions | Extracted classes | | Extracted relationships |
|---|---|---|---|---|
| | | Existing | Additional | |
| Security mechanism | A method, tool, or procedure that is the realization of security requirements. | Security requirements | | Is the realization of |
| | A security mechanism is the implementation of a concept or security function and can be performed by machine/technology elements; human elements; physical elements; environmental elements; and all associated procedures and configurations. | Security function | | Is the implementation of |
| | A security mechanism may enforce the security policy and therefore must have the capabilities consistent with the intent of the security policy. | | Security policy | Enforce |
| Security control | A mechanism designed to fulfill a set of security requirements. | Mechanism<br>Security requirements | | Is a<br>Fulfill |
| Measure | Variable to which a value is assigned as the result of measurement. | Measurement | | Is the result of |

When examining the extracted definition, we found that one of them comprised of an additional term that represents a relevant class entity, namely, security policy. Therefore, for the additional class security policy, we conducted iterations of the fourth and fifth steps. Table 5.25 provides the extracted definitions and relationships.

Table 5.23 Extracted definition and iteration results

| Classes | Extracted definitions | Extracted classes | | Extracted relationships |
|---|---|---|---|---|
| | | Existing | Additional | |
| Security policy | A set of rules that governs all aspects of security-relevant system and system element behavior. | System<br>System element | | Governs |

When examining the extracted definition, we found that it only comprised of terms that represent existing class entities. However, the definitions of the classes' security-driven performance measure, security-driven assurance measure, security constraint, and security concern were missing. When examining these classes, we found that each of them comprised of multiple terms that represent relevant class entities. For example, the class security constraint comprised of the term "constraint", which represents a relevant class entity. We also found that the class constraint was a generalization of the class security constraint. Therefore, we searched the full text of the NIST SP 800-160 (NIST, 2018) report to extract the definitions of the following classes: performance measure, assurance measure, performance, assurance, constraint, and concern. Table 5.26 provides the extracted definitions.

Table 5.24 Extracted definitions and iteration results

| Classes | Extracted definitions | Extracted classes | | Extracted relationships |
|---|---|---|---|---|
| | | Existing | Additional | |
| Assurance | Grounds for justified confidence that a claim has been or will be achieved. | | Claim | |
| | Assurance is typically obtained relative to a set of specific claims. The scope and focus of such claims may vary (e.g., security claims, safety claims) and the claims themselves may be interrelated. | | Security claim | Is obtained relative to |

| Classes | Extracted definitions | Extracted classes | | Extracted relationships |
|---|---|---|---|---|
| | | Existing | Additional | |
| Constraint | Factors that impose restrictions and limitations on the system or actual limitations associated with the use of the system. | System | | Is imposed on |
| Concern | Interest in a system relevant to one or more of its stakeholders. | System | Stakeholder | Is relevant to |

Although we found class definitions for assurance, constraint, and concern, we were unable to find definitions for the class's implementation constraint, security constraint, security concern, performance measure or performance. In this case, the class constraint represents a generalization of the class's implementation constraint and security constraint, and the class concern represents a generalization of the class security concern. With respect to the class assurance measure, although no definition was found, we extracted a definition of the class assurance, which represents a generalization of the class assurance measure given that inheritance can be applied in conjunction with the class measure. When examining the extracted definitions, we found that they comprised of additional terms that represent relevant class entities, namely, claim, security claim, and stakeholder. Therefore, for the additional classes, we conducted iterations of the fourth and fifth steps. Table 5.27 provides the extracted definitions and relationships.

Table 5.25 Extracted definitions and iteration results

| Classes | Extracted definitions | Extracted classes | | Extracted relationships |
|---|---|---|---|---|
| | | Existing | Additional | |
| Claim | A true-false statement about the limitations on the values of an unambiguously defined property called the claim's property; and limitations on the uncertainty of the property's values falling within these limitations during the claim's duration of applicability under stated conditions. | | | |
| Stakeholder | Individual or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations. | System | | Has interest in |

Although we found definitions for the classes claim and stakeholder, we found no definition for the class security claim. In this case, the class claim represents a generalization of the class security claim. When examining the extracted definitions, we found that they only comprised of terms that represent existing classes entities. However, before proceeding with the construction of the ontology, we extended our search to define the class performance measure as its definition was missing.

The activities in the scope of the ontology found in section 3.4.3 of the NIST SP 800-160 (NIST, 2018) report, namely, SR-2 defines system security requirements, and SR-3 analyzes system security requirements, refers to the INCOSE-TP-2003-020-01 (Roedler & Jones, 2005) technical report on technical measurements, which provides information on the selection, definition, and implementation of critical performance measures. Therefore, we searched the full text of the technical report to identify the term performance measure, which represents a relevant class entity, and subsequently extracted its definition. Moreover, we found two terms related to the term performance measure that represent relevant class entities, namely, measure of performance and technical performance measure, for which we extracted their definitions. In this case, the class performance measure represents a generalization of the classes measure of performance and technical performance measure. Table 5.28 shows the results.

Table 5.26 Extracted definitions

| **Classes** | **Extracted definitions** |
|---|---|
| Measure of performance (MOP) | The measures that characterize physical or functional attributes relating to the system operation, measured or estimated under specified testing and/or operational environment conditions. |
| Technical performance measure (TPM) | TPMs measure attributes of a system element to determine how well a system or system element is satisfying or expected to satisfy a technical requirement or goal. |

Given that all classes have now been defined, we proceeded with the construction of the ontology. Figure 5.8 provides the resulting ISO/IEC/IEEE 15288 system requirements definition and NIST SP 800-160 system security requirements definition framework ontology.

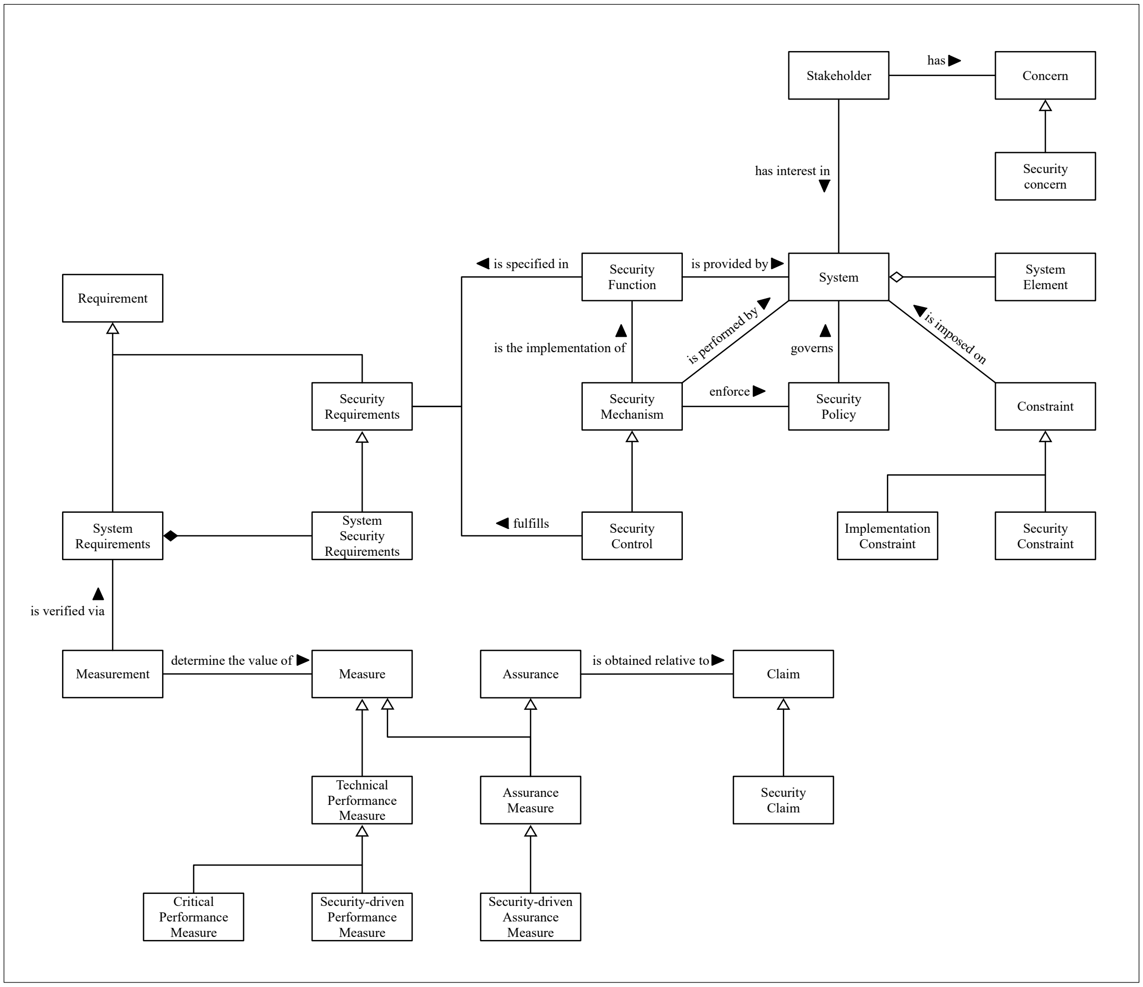


Figure 5.8 The ISO/IEC/IEEE 15288 system requirements definition and NIST SP 800-160 system security requirements definition framework ontology

## 5.6 Framework integration

In using metamodels to integrate IS frameworks, Goeken & Alter (2009) identified two specific problems. The first problem is the identification of homonyms, synonyms, and antonyms. The second problem is the treatment of identified correspondences. To address these problems, they refer to database research and more specifically schema integration. As their study pointed out, the core schema problems are as follows: schema matching, namely, the identification of correspondences (or mappings) between schema constructs; and schema merging, namely, the construction of a unified schema based on identified mappings (Magnani, Rizopoulos, Brien

& Montesi, 2005). Therefore, we used the semantic relationships proposed by Rizopoulos & Mçbrien (2005) to support ontology integration activities. Proposed are four semantic relationship types between the schema object based on a set of comparisons among *intentional domains*, namely, a set of real-world objects that they represent. $Dom_{int}(E)$ was used to define the *intentional domain*.

Rizopoulos & Mçbrien (2005) semantic relationships are as follows:

1. Equivalence: Two schema constructs A and B are equivalent, $A \overset{S}{=} B$, iff $D_i(A) = D_i(B)$;
2. Subsumption: Schema construct $A$ subsumes shema construct $B$, $B \overset{S}{\subset} A$, iff $D_i(B) \subset\ D_i(A)$;
3. Intersection: Two schema constructs A and B are intersecting, $A \overset{S}{\cap} B$, iff $D_i(A) \cap\ D_i(B)\ \neq \emptyset, \exists C\colon D_i(A) \cap\ D_i(B) = D_i(C)$;
4. Disjointness: Two schema constructs A and B are disjoint, $A\ B$, iff $D_i(A) \cap\ D_i(B)\ \neq \emptyset, \exists C\colon D_i(A) \cap\ D_i(B) = D_i(C)$.

When analyzing the framework ontologies to be integrated, we found that there was an even number of ontologies. We also found that they represented two domains. The ISO/IEC 27005 information security risk management and the ISO/IEC 27017 and 27018 cloud computing information security frameworks both represent real-world objects from the information security domain. In the case of the ISO/IEC 25030 quality requirements definition and the NIST SP 800-160 system security requirements definition frameworks, their respective activities and tasks are mapped to the ISO/IEC/IEEE 15288 system requirements definition framework. Consequently, they both represent real-world objects from the system requirements definition domain.

Given that the semantic relationships that specifies the mapping between entities are based on a set of comparison of their *intentional domains*, namely, a set of real-world objects that they represent, we propose pairing the framework ontologies per domain to simplify integration activities.

Therefore, in section 5.6.1, the ISO/IEC 27005 information security risk management framework ontology was integrated with the ISO/IEC 27017 and 27018 cloud computing information security framework ontology. Then, in section 5.6.2, the ISO/IEC 25030 quality requirements definition framework ontology was integrated with the ISO/IEC/IEEE 15288 system requirements definition and NIST SP 800-160 system security requirements definition framework ontology. Finally, in section 5.6.3, the resultant framework ontologies were integrated.

### 5.6.1 Integration of information security ontologies

As discussed in section 4.3, we applied the Schuette & Rotthowe (1998) principle of comparability to first determine whether the comparison of the ISO/IEC 27005 information security risk management framework ontology by Agrawal (2016) and the ISO/IEC 27017 and ISO/IEC 27018 cloud computing information security framework ontology (extended from the ISO/IEC 27001 ontology by Milicevic & Goeken (2010)) is possible. As the language and grammar used in both ontologies are compatible, we proceeded with the comparison.

Before proceeding with the comparison, however, it is important to note that while Milicevic & Goeken (2010) identifies the concepts security incident, security event, and security breach, these concepts are merged under security breach in their ontology. This is also the case for the ISO/IEC 27017 and ISO/IEC 27018 cloud computing information security framework ontology as it is an extension of the ISO/IEC 27001 ontology by Milicevic & Goeken (2010). It is also important to note that according to Agrawal (2016) the concept event is also referred to as security incident.

We applies the semantic relationships proposed by Rizopoulos & Mçbrien (2005) to support the comparison activity. The comparison of the ISO/IEC 27005 information security risk management framework ontology ($O_1$) and the ISO/IEC 27017 and 27018 cloud computing information security framework ontology ($O_2$) generated the following semantic mappings (5.1 to 5.4):

$$O_1{:} \ll asset \gg \cap\, O_2{:} \ll asset \gg \tag{5.1}$$

$$O_1{:} \ll exploits, threat, vulnerability \gg \cap\, O_2{:} \ll threatens, threat, asset \gg \tag{5.2}$$

$$O_1{:} \ll event \gg \subset O_2{:} \ll security\ breach \gg \tag{5.3}$$

$$O_1{:} \ll control \gg = O_2{:} \ll control \gg \tag{5.4}$$

We then used the semantic mappings to merge the ISO/IEC 27005 information security risk management framework ontology and the ISO/IEC 27017 and 27018 cloud computing information security framework ontology. Figure 5.9 provides the resultant cloud computing service information security risk management framework ontology.

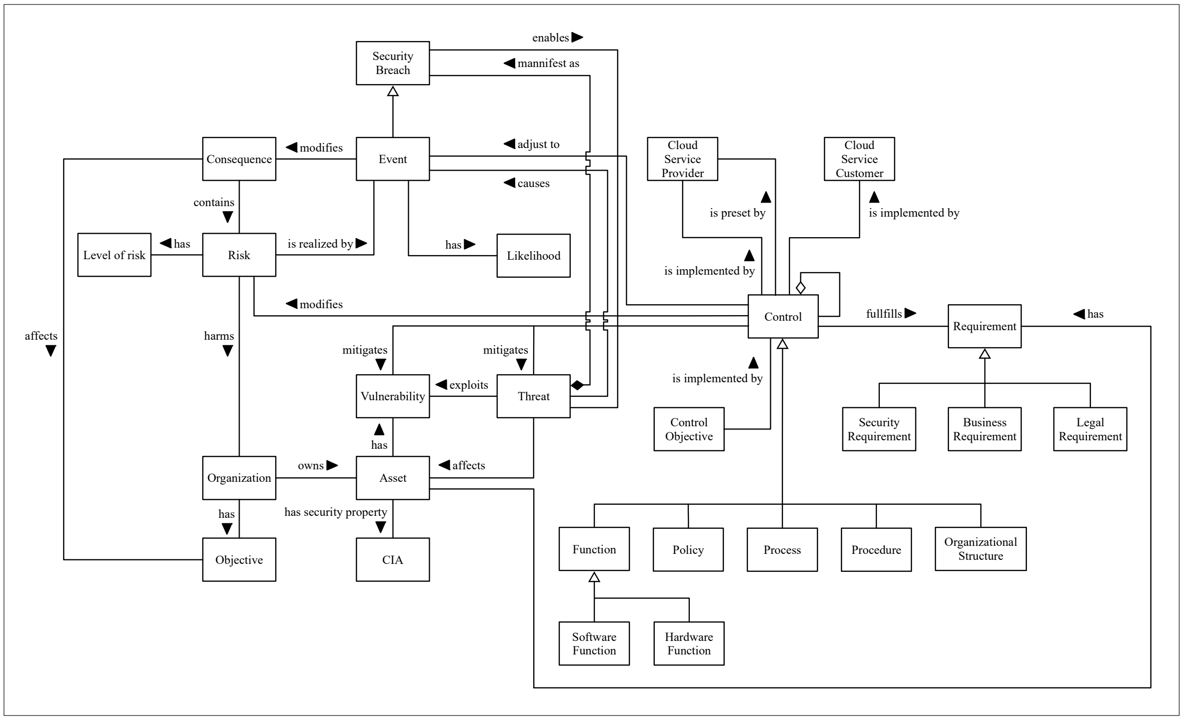


Figure 5.9 The cloud computing service information security risk management framework ontology

### 5.6.2 Integration of system requirements definition ontologies

To ensure that the comparison of the ISO/IEC 25030 quality requirements definition framework ontology and the NIST SP 800-160 system security requirements definition framework ontology is possible, we applied the Schuette & Rotthowe (1998) principle of comparability in their construction.

We also applied the semantic relationships proposed by Rizopoulos & Mçbrien (2005) to support the comparison. The comparison of the ISO/IEC 25030 quality requirements definition framework ontology ($O_3$) and the NIST SP 800-160 system security requirements definition framework ontology ($O_4$) generated the following semantic mappings (5.5 to 5.16):

$$O_3\!: \ll function \gg \supset O_4\!: \ll security\ function \gg \tag{5.5}$$

$$O_3\!: \ll control \gg \supset O_4\!: \ll security\ control \gg \tag{5.6}$$

$$O_3\!: \ll information\ system \gg \subset O_4\!: \ll system \gg \tag{5.7}$$

$$O_3\!: \ll ICT\ product \gg \subset O_4\!: \ll system\ element \gg \tag{5.8}$$

$$O_3\!: \ll user \gg \subset O_4\!: \ll system\ element \gg \tag{5.9}$$

$$O_3\!: \ll other\ stakeholders \gg \subset O_4\!: \ll system\ element \gg \tag{5.10}$$

$$O_3\!: \ll relevant\ environment \gg \subset O_4\!: \ll system\ element \gg \tag{5.11}$$

$$O_3\!: \ll software \gg \subset O_4\!: \ll system\ element \gg \tag{5.12}$$

$$O_3\!: \ll data \gg \subset O_4\!: \ll system\ element \gg \tag{5.13}$$

$$O_3\!: \ll hardware \gg \subset O_4\!: \ll system\ element \gg \tag{5.14}$$

$$O_3\!: \ll communication\ facility \gg \subset O_4\!: \ll system\ element \gg \tag{5.15}$$

$$O_3\!: \ll software\ component \gg \subset O_4\!: \ll system\ element \gg \tag{5.16}$$

We then used the semantic mappings to merge the ISO/IEC 25030 quality requirements definition framework ontology and the NIST SP 800-160 system security requirements definition framework ontology. Figure 5.10 provides the resultant system security quality requirement framework ontology.

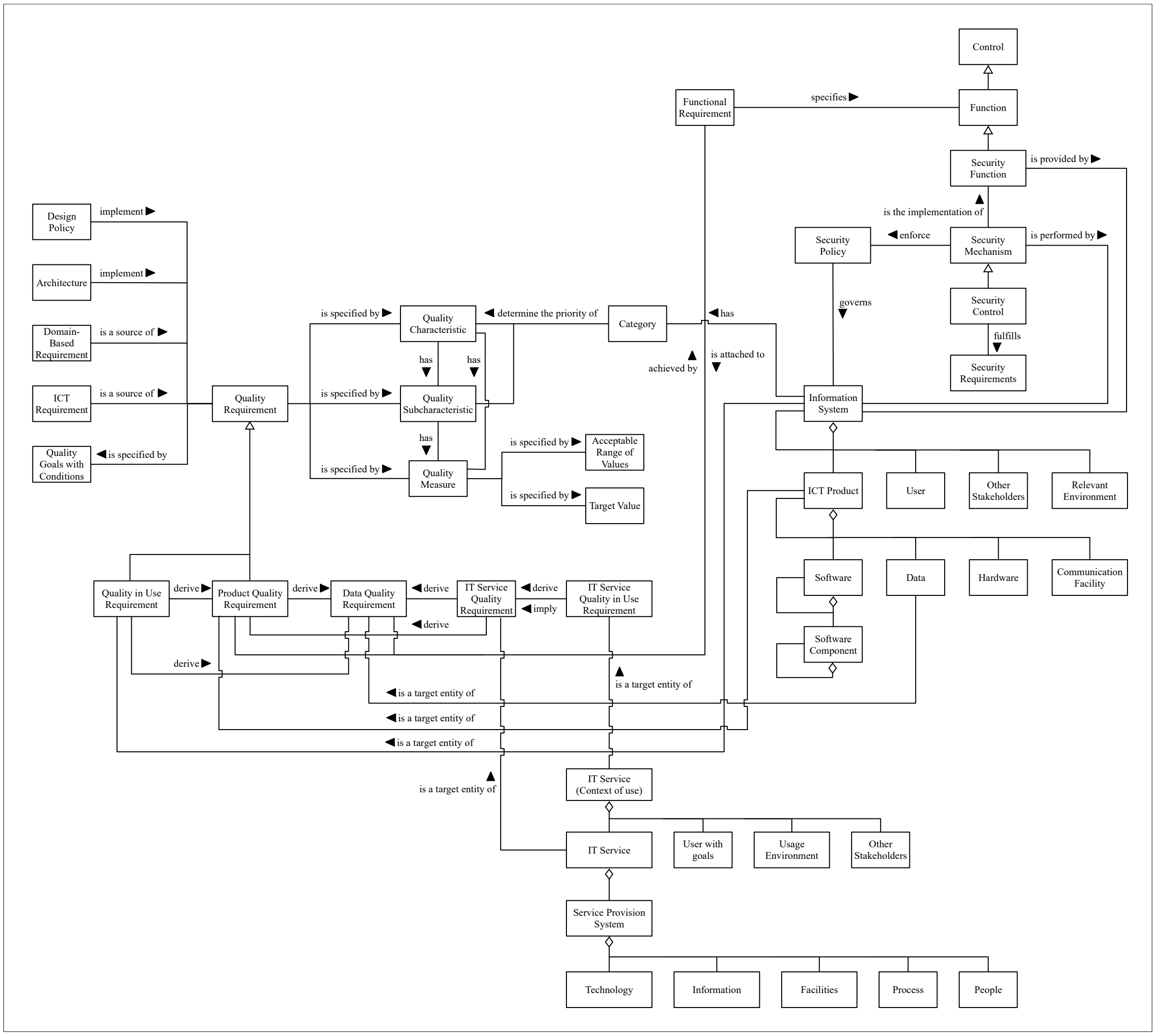


Figure 5.10 The system security quality requirement framework ontology

### 5.6.3 The integrated ontology

The cloud computing service information security risk management framework ontology and the system security quality requirement framework ontology were merged to construct the integrated ontology. To ensure that the comparison of the framework ontologies is possible, we applied the Schuette & Rotthowe (1998) principle of comparability in their construction.

We also applied the semantic relationships proposed by Rizopoulos & Mçbrien (2005) to support the comparison. The comparison of the cloud computing services information security risk management framework ontology ($O_5$) and the system security quality requirement framework ontology ($O_6$) generated the following semantic mappings (5.17 to 5.21):

$$O_5\colon \ll control \gg \cap\, O_6\colon \ll control \gg \tag{5.17}$$

$$O_5\colon \ll function \gg = O_6\colon \ll function \gg \tag{5.18}$$

$$O_5\colon \ll software\ function \gg \subset O_6\colon \ll security\ function \gg \tag{5.19}$$

$$O_5\colon \ll hardware\ function \gg \subset O_6\colon \ll security\ function \gg \tag{5.20}$$

$$O_5\colon \ll fullfills, control, security\ requirement \gg \supset O_6\colon \ll fullfills, security\ control, security\ requirement \gg \tag{5.21}$$

We then used the semantic mappings to merge the cloud computing services information security risk management framework ontology and the system security quality requirement framework ontology.

Additionally, we introduced a composition relationship between the classes quality requirements and system requirements. This is grounded in ISO/IEC/IEEE 15288 (ISO/IEC/IEEE, 2015), which stipulates that a key outcome of the successful implementation of the system requirement definition process is that “system requirements (functional, performance, process, non-functional, and interface) and design constraints are defined”, indicating that the system requirements are composed of non-functional requirements or quality requirements. We also introduced a composition relationship between the classes quality requirements and system security requirements given that the latter is defined in NIST SP 800-160 (NIST, 2018) as “system requirements that reflect the technical security view of the system”. Consequently, system security requirements are composed of non-functional requirements or quality requirements that reflect the technical security perspective of the system. Finally, we introduced a composition relationship between the classes technical performance measure and quality measure as Roedler & Jones (2005) stipulate that “TPMs can include, but are not limited to…product quality characteristics related to critical operational requirements (reliability figure of merit, failure rate, mean time to failure/repair/restore, availability, fault tolerance, etc.).” The resultant integrated ontology is presented in Annex I. Figure 5.11 provides a lighter integrated ontology reduced to extracted key artifacts classes.

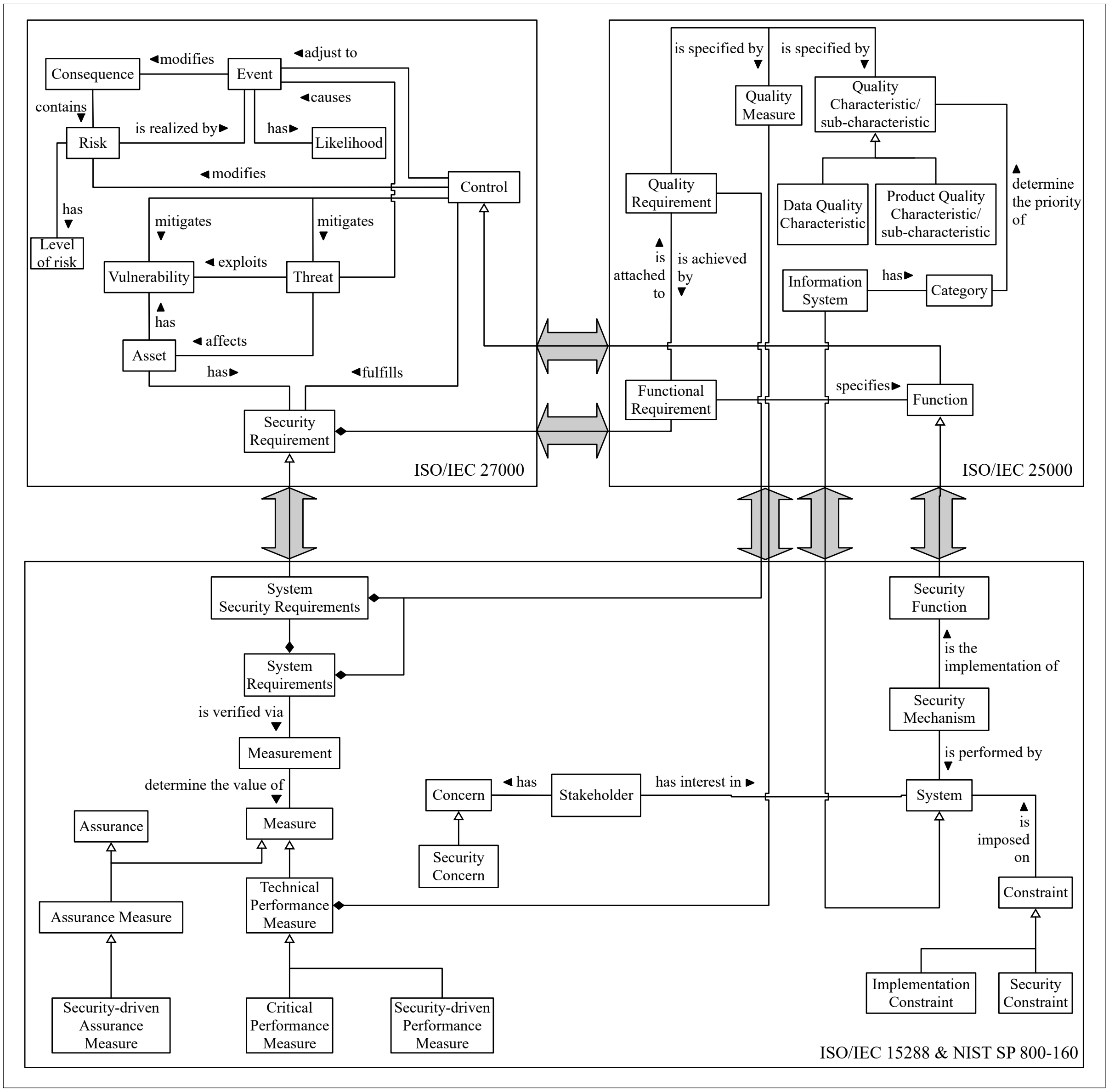


Figure 5.11 Lighter integrated ontology: Reduced to extracted key artifacts classes

## 5.7 Ontology evaluation

We could now evaluate the extent to which the integrated ontology fulfils our design objective. To achieve our design objectives (previously presented in section 5.2), the integrated ontology must include the classes that represent the key artifacts (previously extracted from the selected processes, activities, and tasks discussed in Chapter 4). Therefore, to evaluate its completeness, we mapped the extracted key artifacts to the classes of our integrated ontology. The mapping is presented in Table 5.30. Based on the mapping, we found that all extracted key artifacts are covered by the integrated ontology classes.

Table 5.27 Mapping of extracted key artifacts to the integrated ontology classes

| **Extracted key artifacts** | **Integrated ontology classes** |
|---|---|
| Asset | Asset |
| Threat | Threat |
| Control | Control |
| Vulnerability | Vulnerability |
| Consequence | Consequence |
| Incident | Incident |
| Likelihood | Likelihood |
| Risk | Risk |
| Level of risk | Level of risk |
| Quality characteristic | Quality characteristic |
| Quality requirement | Quality requirement |
| Product quality requirement | Product quality requirement |
| Data quality requirement | Data quality requirement |
| Quality measure | Quality measure |
| Function | Function |
| Implementation constraint | Implementation constraint |
| System requirement | System requirement |
| Criticality of the system | Category of the system |
| Critical performance measure | Critical performance measure |
| Stakeholder | Stakeholder |
| Security function | Security function |
| System | System |
| Security requirement | Security requirement |
| System security requirement | System security requirement |
| Security constraint | Security constraint |
| Security concern | Security concern |
| Security-driven performance measure | Security-driven performance measure |
| Security-driven assurance measure | Security-driven assurance measure |

Additionally, the integrated ontology must be able to answer our research question “To what extent do ISO/IEC 25000 quality models support the definition of quality requirements related to information security in cloud computing services?” As indicated by Fox, Barbuceanu & Gruninger (1995), ontology requirements can be defined in the form of questions that an ontology must be able to subsequently answer (i.e., competency questions (CQ)). This is known as the competency of the ontology.

For the integrated ontology to be able to answer our CQ (i.e., research question), it should explicate the relationships between the classes from which we can create an instance based on the case scenario of information security and cloud computing services and the class's quality characteristic, quality sub-characteristic, quality measures, as well as the class's security function, security architecture, and security-focused verification method.

Consequently, to conduct the evaluation, we first identified the classes of the integrated ontology for which we could create an instance based on the case scenario of information security and cloud computing services. The classes and their instances are presented in Table 5.31.

Table 5.28 Classes and their instances (the case scenario of information security and cloud computing services)

| **Class** | **Instances based on the case scenario** | **Relationship to the class** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Quality characteristic** | **Quality sub-characteristic** | **Quality measure** | **Security function** | **Security architecture** | **Security-focused verification method** |
| ICT requirement | Cloud computing technology | X | X | X | X | X | X |
| Information system | Cloud-based information system | X | X | X | X | X | X |
| Control | Cloud service information security control | X | X | X | X | X | X |
| Cloud service provider | Cloud service provider | X | X | X | X | X | X |
| Cloud service customer | Cloud service customer | X | X | X | X | X | X |
| Threat | Threat to cloud computing | X | X | X | X | X | X |
| Vulnerability | Cloud computing vulnerability | X | X | X | X | X | X |
| Risk | Cloud computing risk | X | X | X | X | X | X |

We then verified whether a relationship existed between each of these classes to the class's quality characteristic, quality sub-characteristic, quality measures, as well as the class's security function, security architecture, and security-focused verification method. The following shows an example of this analysis:

The class **ICT requirement** has the following relationships:

1. An **ICT requirement** *IsAsourceOf* **quality requirement** *SpecifiedBy* **quality characteristic**, **quality sub-characteristic**, and **quality measure**;
2. A **quality requirement** *SpecifiedBy* **quality characteristic**, **quality sub-characteristic**, and **quality measure** *IsAcheivedBy* **functional requirement** that *Specifies* a **security function**;
3. A **functional requirement** *Specifies* a **security function** and *IsAttachedTo* a **quality requirement** *SpecifiedBy* **quality characteristic**, **quality sub-characteristic**, and **quality measure**.

The results of this analysis are presented in Table 5.29, which show that each of the classes, namely, ICT requirement, information system, control, cloud service provider, and cloud service customer, has a relationship to the classes quality characteristic, quality sub-characteristic, quality measure, and security function.

Consequently, we concluded that our integrated ontology comprised of the classes and relationships to answer the CQ (i.e., research question). Figure 5.12 shows a lighter integrated ontology (i.e., reduced number of classes and relationships into minimum information model) that is comprised of the classes and relationships applicable to the research question. Figure 5.13 shows the mapping between extracted key artifact classes and their associated activities/tasks.

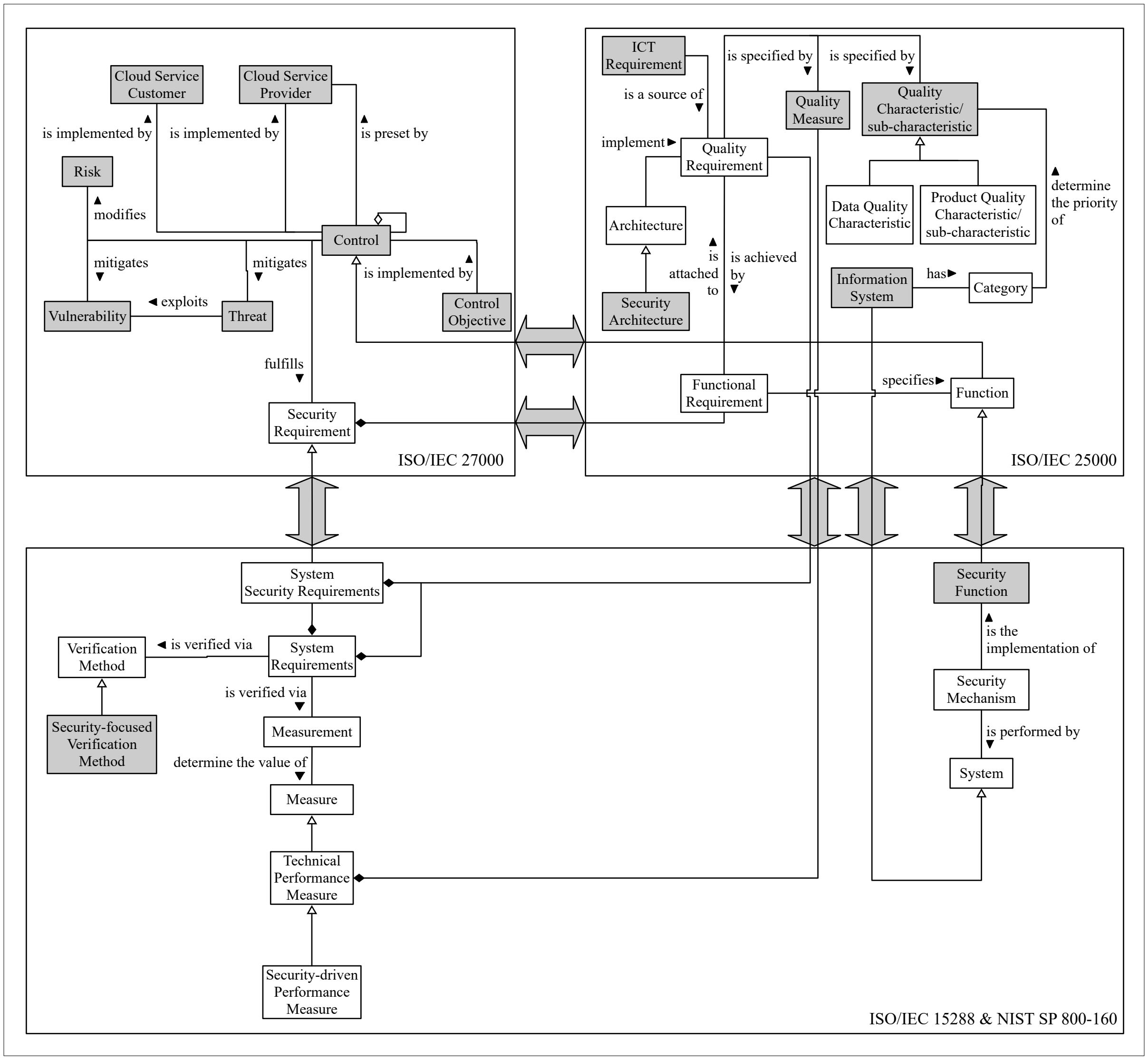


Figure 5.12 Lighter integrated ontology: Reduced to classes and relationships applicable to the research question

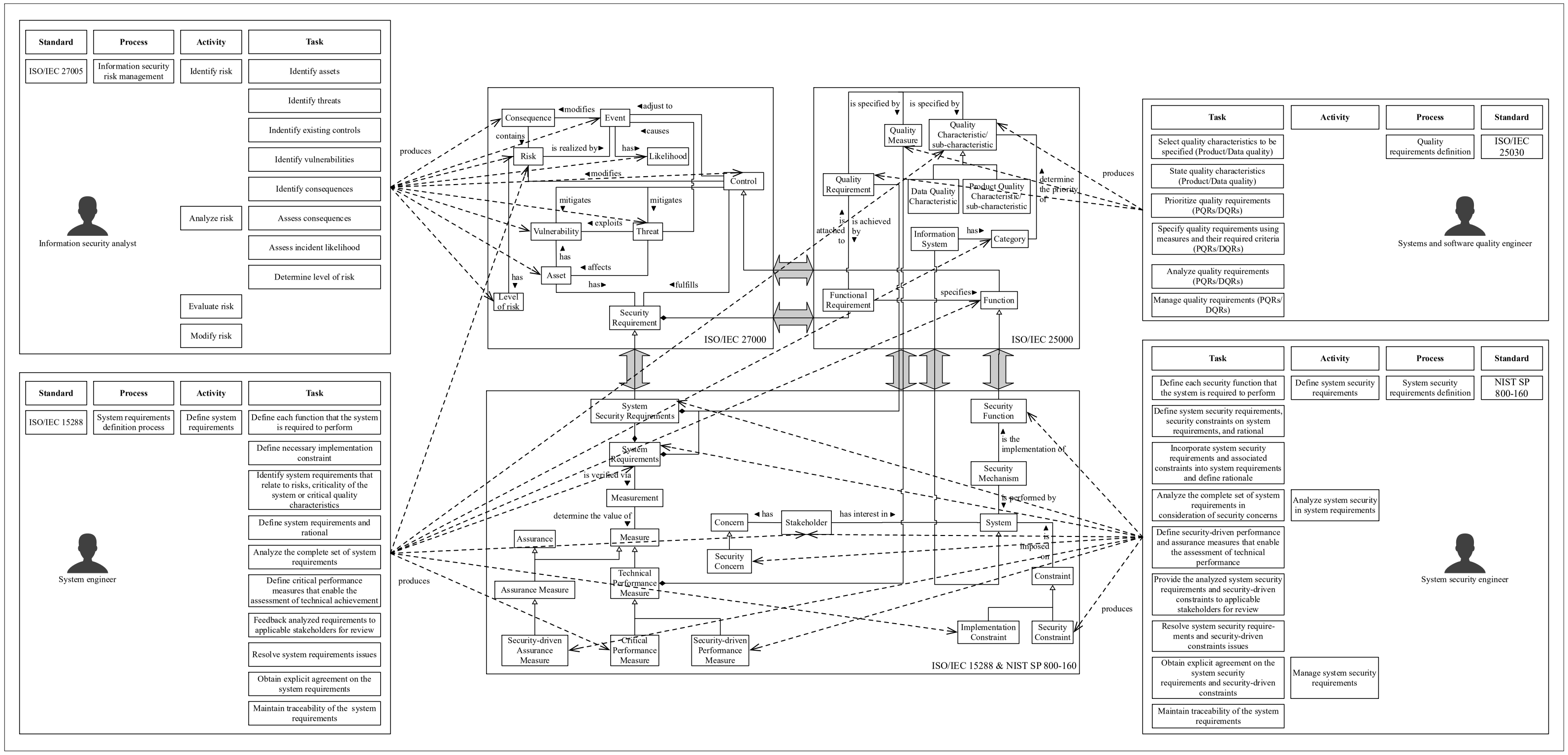

Standard
Process
Activity
Task
ISO/IEC 27005
Information security risk management
Identify risk
Identify assets
Identify threats
Indentify existing controls
Identify vulnerabilities
Identify consequences
Analyze risk
Assess consequences
Assess incident likelihood
Determine level of risk
Evaluate risk
Modify risk
Information security analyst
produces
Standard
Process
Activity
Task
ISO/IEC 15288
System requirements definition process
Define system requirements
Define each function that the system is required to perform
Define necessary implementation constraint
Identify system requirements that relate to risks, criticality of the system or critical quality characteristics
Define system requirements and rational
Analyze the complete set of system requirements
Define critical performance measures that enable the assessment of technical achievement
Feedback analyzed requirements to applicable stakeholders for review
Resolve system requirements issues
Obtain explicit agreement on the system requirements
Maintain traceability of the system requirements
System engineer
produces
Consequence
◀modifies
Event
◀adjust to
contains
Risk
is realized by▶
◀causes
has▶
Likelihood
◀modifies
Control
mitigates
mitigates
Vulnerability
◀ exploits
Threat
has
◀ affects
Asset
has
has▶
◀fulfills
Level of risk
Security Requirement
ISO/IEC 27000
is specified by
is specified by
Quality Measure
Quality Characteristic/ sub-characteristic
Quality Requirement
Data Quality Characteristic
Product Quality Characteristic/ sub-characteristic
determine the priority of
is attached to
is achieved by
Information System
has▶
Category
specifies▶
Functional Requirement
Function
ISO/IEC 25000
System Security Requirements
System Requirements
is verified via
Measurement
determine the value of
Assurance
Measure
Concern
◀ has
Stakeholder
has interest in ▶
Security Concern
Security Function
is the implementation of
Security Mechanism
is performed by
System
is imposed on
Constraint
Assurance Measure
Technical Performance Measure
Security-driven Assurance Measure
Critical Performance Measure
Security-driven Performance Measure
Implementation Constraint
Security Constraint
ISO/IEC 15288 & NIST SP 800-160
produces
Task
Activity
Process
Standard
Select quality characteristics to be specified (Product/Data quality)
Quality requirements definition
ISO/IEC 25030
State quality characteristics (Product/Data quality)
Prioritize quality requirements (PQRs/DQRs)
Specify quality requirements using measures and their required criteria (PQRs/DQRs)
Analyze quality requirements (PQRs/DQRs)
Manage quality requirements (PQRs/ DQRs)
Systems and software quality engineer
produces
Task
Activity
Process
Standard
Define each security function that the system is required to perform
Define system security requirements
System security requirements definition
NIST SP 800-160
Define system security requirements, security constraints on system requirements, and rational
Incorporate system security requirements and associated constraints into system requirements and define rationale
Analyze the complete set of system requirements in consideration of security concerns
Analyze system security in system requirements
Define security-driven performance and assurance measures that enable the assessment of technical performance
Provide the analyzed system security requirements and security-driven constraints to applicable stakeholders for review
Resolve system security requirements and security-driven constraints issues
Obtain explicit agreement on the system security requirements and security-driven constraints
Manage system security requirements
Maintain traceability of the system requirements
System security engineer

Figure 5.13 Lighter integrated ontology: Mapping among extracted key artifact classes and their associated activities/tasks

## 5.8 Conclusions

This chapter discussed the construction of an integrated ontology grounded in the ISO/IEC/IEEE 15288 (ISO/IEC/IEEE, 2015), ISO/IEC 25030 (ISO/IEC, 2019), NIST SP 800-160 (NIST, 2018), ISO/IEC 27005 (ISO/IEC, 2018), ISO/IEC 27017 (ISO/IEC, 2015), and ISO/IEC 27018 (ISO/IEC, 2019) standards, that explicate the relationships between the concepts involved in:

1. The activities and tasks that relate to information security and cloud computing services from the following processes:
    a. System and software requirements definition process;
    b. System and software quality requirements definition processes;
    c. System security requirements definition process;
    d. Information security risk management process;
2. The definition of quality requirements related to information security in cloud computing services.

In the following chapter, we apply the integrated ontology to answer our research question.

# CHAPTER 6

# EVALUATION OF THE APPLICABILITY OF THE EXTENDED FRAMEWORK TO INFORMATION SECURITY IN CLOUD COMPUTING SERVICES

## 6.1 Introduction

In the preceding chapter, we constructed an integrated ontology combining the information security risk management framework support for the selection of cloud computing services information security controls, the system security engineering framework support for the system security requirements definition, and the quality requirements framework support for the quality requirements definition of IS. The integrated ontology explicates the relationships between the concepts involved in the definition of quality requirements related to information security in cloud computing services.

In this chapter, we apply the integrated ontology to answer our research question: "To what extent do ISO/IEC 25000 quality models support the definition of quality requirements related to information security in cloud computing services?". As discussed in Chapter 3, the definition of quality requirements related to information security in cloud computing services entails the identification of applicable and/or missing quality characteristics, quality sub-characteristics, quality measures, security functions, security architecture, and the security-focused evaluation method extracted from ISO/IEC 25000 quality models. Being a component of the design science methodology, this constitutes an artifact evaluation.

## 6.2 Evaluation design

Based on Sonnenberg & Vom Brocke (2011) recommendations, the selection of the evaluation pattern demonstration for this evaluation is based on the pattern's context and applicability criteria as provided and discussed in Table 3.2.

The intent of this evaluation pattern is to demonstrate that the artifact design embodies the solution to the identified problem and is applicable in the context of an artificial setting. Therefore, we demonstrate in the context of an artificial setting that the integrated ontology can answer the CQ (i.e., research question). Based on the integrated ontology classes and relationships, the CQ can be divided into six CQs:

CQ1: What **control objective** is implemented by a **control** that fulfills a **quality requirement** specified by a **quality characteristic,** a **quality sub-characteristic**, and a **quality measure**?

CQ1.1: Is the **quality requirement** achieved by a **functional requirement** that specifies a **security function**?

CQ1.2: Is the **quality requirement** implemented by a **security architecture**?

CQ1.3: Is the **quality requirement** verified via a **security-focused verification method**?

CQ2: What **control objective** is implemented by a **control** found by CQ1?

CQ3: What **control** found by CQ1 or CQ2 mitigates a **threat** to cloud computing and/or a cloud computing **vulnerability** and/or modifies a cloud computing **risk**?

For each CQ, we create an instance of the integrated ontology using input data. We use the quality characteristics and quality sub-characteristics from the ISO/IEC 25010 (ISO/IEC, 2019) and ISO/IEC 25012 (ISO/IEC, 2008) quality models as instances of the class's quality characteristic and quality sub-characteristic, while we use the quality measures from ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015) as instances of the class quality measure. Moreover, we use the security functions, the security architecture, and the security-focused verification method previously extracted from the ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015) quality measures in Chapter 3 as instances of the class's security function, security architecture, and security-focused verification method. Finally, we use the cloud computing service information security controls from ISO/IEC 27017 (ISO/IEC, 2015) and ISO/IEC 27018 (ISO/IEC, 2019) as instances of the class control and the top threats to cloud computing previously extracted from the CSA report provided in Chapter 3 as instances of the class threat.

Each cloud computing service information security control has a category and description. The control category contains a control objective and one or more control that can be applied to achieve the control objective. The control description contains the control statement to satisfy the control objective, implementation guidance to support the implementation of the control to meet the control objective, and other relevant information that may need to be considered in certain cases.

The evaluation consists of five steps. In step 1, for each security function, the security architecture, and the security-focused verification method, we analyze the category and descriptive content of each control to instantiate our integrated ontology highlighted in Figure 6.1 to answer CQ1.

CQ1 instantiation:

- The **control objective** *IsImplementedBy* a **control** that *Fulfills* a **quality requirement** *SpecifiedBy* a **quality characteristic**, a **quality sub-characteristic**, and a **quality measure**.

  CQ1.1 instantiation:

  - The **quality requirement** *IsAcheivedBy* a **functional requirement** that *Specifies* the **security function**.

  CQ1.2 instantiation:

  - The **security architecture** *Implement* the **quality requirement**.

  CQ1.3 instantiation:

  - The **quality requirement** *IsVerifiedVia* the **security-focused verification method**.

In step 2, we analyze the category and description content of each control to determine whether the classes and relationships of our integrated ontology highlighted in Figure 6.1 can be instantiated to answer CQ2.

CQ2 instantiation:

- The **control objective** *IsImplementedBy* a **control** found by CQ1.

In step 3, we determine whether the classes and relationships of our integrated ontology highlighted in Figure 6.1 can be instantiated to answer CQ3.

CQ3 instantiation:

- The **control** found by CQ1 or CQ2 *Mitigates* the **threat**.

To do so, we determine if the **control**(s) found by CQ1 or CQ2 have been mapped in the CSA Cloud Controls Matrix (CCM) as a **control**(s) that is recommended to mitigate a **threat**(s).

In step 4, for each **quality measure** and associated **control** found by CQ1 or CQ2, we analyze, when available, the implementation guidance for cloud computing services and for the protection of PII in public cloud computing services (extracted from ISO/IEC 27017 (ISO/IEC, 2015) and ISO/IEC 27018 (ISO/IEC, 2019)) of the **control** to determine whether there is evidence to suggest that the information and/or the procedures and/or the support that should be provided by the cloud service provider to the cloud service customer is sufficient to verify the **quality requirement** specified by the **quality measure**. We also analyze, when available, the implementation guidance (extracted from ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015)) of the **quality measure** to determine whether there is evidence to suggest that the cloud service customer can verify the **quality requirement** specified by the **quality measure**.

Finally, in step 5, for each **quality measure** for which no security function, security architecture, or security-focused verification method was extracted, we analyze the category and description content of each control to determine if additional instantiations of our integrated ontology can be found.

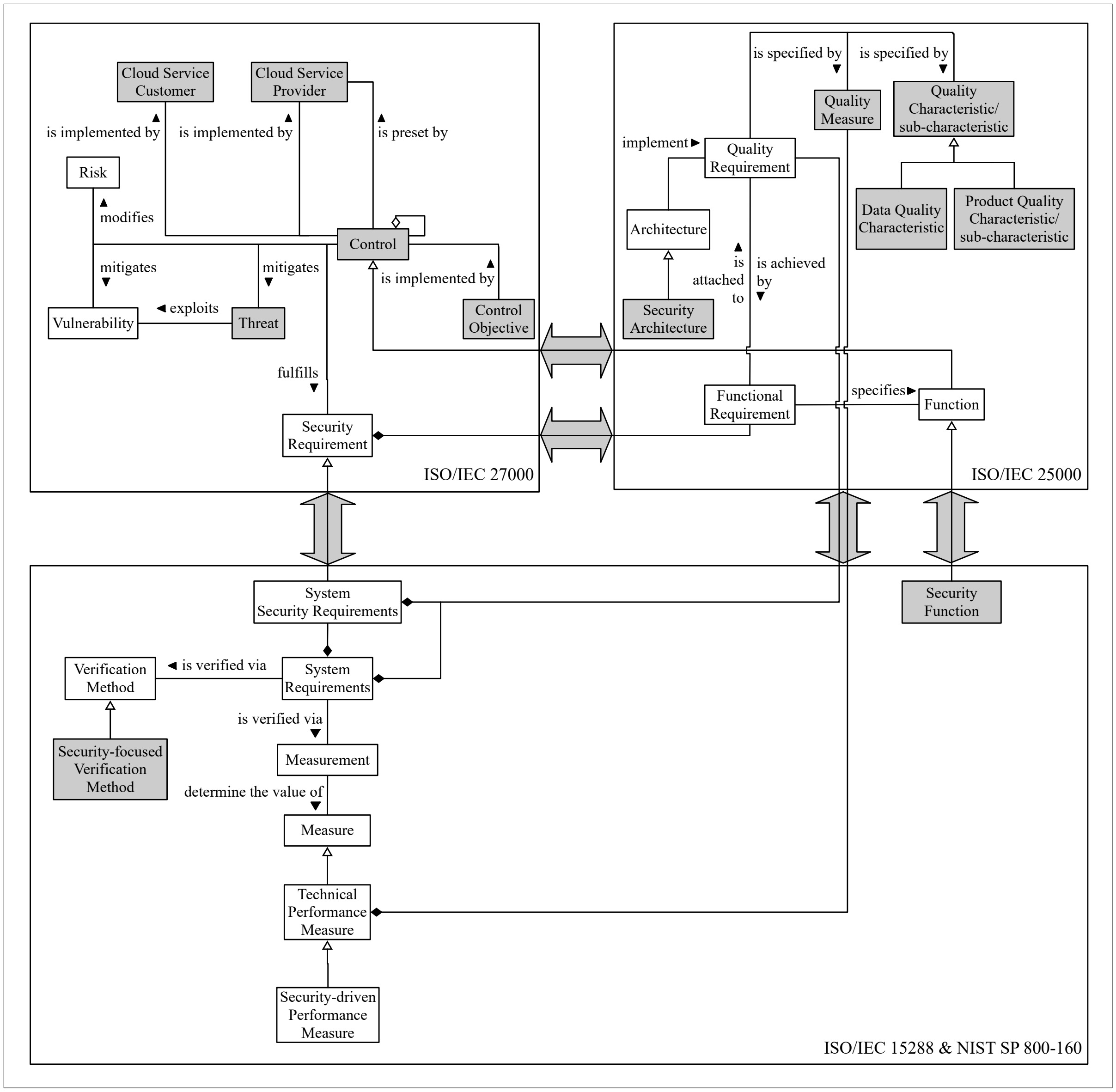


Figure 6.1 Lighter integrated ontology: Reduced to classes and relationships applicable to CQ1, CQ2, and CQ3 instantiations

## 6.3 Applicability of the integrated ontology and ISO/IEC 25000 quality models to information security in cloud computing services

The complete evaluation of the security functions, the security architecture, and the security-focused verification method extracted from ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015), using the controls extracted from ISO/IEC 27017 (ISO/IEC, 2015) and ISO/IEC 27018 (ISO/IEC, 2019), the controls extracted from the CSA CCM, and the threats extracted from the CSA report on the top threats to cloud computing is provided in Annex II. In the following section, we demonstrate the evaluation for the security function encryption.

### 6.3.1 Security function: Encryption

**Encryption** is the security function extracted from the quality measure **data encryption correctness** and **encryption usage**. Tables 6.1 and 6.2 show the security function extracted from the quality measures.

Table 6.1 Security function extracted from the quality measure data encryption correctness

| **Extracted from ISO/IEC 25023 quality measures** | **ISO/IEC 25023 quality measures** | **ISO/IEC 25010 quality model** | |
|---|---|---|---|
| **Function** | **Quality measure** | **Quality sub-characteristic** | **Quality characteristic** |
| Encryption | Data encryption correctness | Confidentiality | Security |
| | Strength of cryptographic algorithms | | |

Table 6.2 Security function extracted from the quality measure encryption usage

| **Extracted from ISO/IEC 25024 quality measures** | **ISO/IEC 25024 quality measures** | **ISO/IEC 25012 quality model** |
|---|---|---|
| **Function** | **Quality measure** | **Quality characteristic** |
| Encryption | Encryption usage | Confidentiality |

**Step 1: CQ1 instantiation**

We found the control description excerpt provided in Table 6.3 to instantiate our integrated ontology with the **security function** encryption to answer CQ1.

Table 6.3 Cryptography controls description excerpt from ISO/IEC 27002 (ISO/IEC, 2013)

10.1 Cryptography controls

Objective: To ensure proper and effective use of cryptography to protect the confidentiality, authenticity and/or integrity of information.

10.1.1 Policy on the use of cryptographic controls

Control
A policy on the use of cryptographic controls for protection of information should be developed and implemented.

Implementation guidance
When developing a cryptographic policy the following should be considered:

a) the management approach towards the use of cryptographic controls across the organization, including the general principles under which business information should be protected; […]

b) based on a risk assessment, the required level of protection should be identified taking into account the type, strength and quality of the encryption algorithm required; […]

According to its description, the **control objective** is “To ensure the proper and effective use of cryptography to protect the confidentiality, authenticity, and/or integrity of information” and *IsImplementedBy* the **control** policy on the use of cryptographic controls to *Fulfill*:

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** security, the **quality sub-characteristic** confidentiality, and **quality measure** strength of cryptographic algorithms defined as the measurement function of values of the quality measure elements:
    a. Number of cryptographic algorithms that are either broken or considered too risky to use;
    b. Number of cryptographic algorithms used;
2. A **quality requirement** *SpecifiedBy* the **quality characteristic** security, the **quality sub-characteristic** integrity, and **quality measure** data encryption correctness defined as the measurement function of values of the quality measure elements:
    a. Number of data items encrypted/decrypted correctly;
    b. Number of data items that require encryption/decryption;
3. A **quality requirement** *SpecifiedBy* the **quality characteristic** confidentiality and the **quality measure** encryption usage defined as the measurement function of values of the quality measure elements:
    a. Number of data values correctly and successfully encrypted and decrypted;
    b. Number of data values with encryption and decryption requirements.

Each **quality requirement** *IsAcheivedBy* a **functional requirement** that *Specifies* the **security function** encryption.

**Step 2: CQ2 instantiation**

Based on their descriptions, we found the **controls** provided in Tables 6.4 and 6.5 to instantiate our integrated ontology to answer CQ2 where the **control objective** *IsImplementedBy* the **control** policy on the use of cryptographic controls associated with the **security function** encryption.

Table 6.4 Control description excerpts from ISO/IEC 27002 (ISO/IEC, 2013)

| **Control objective instance** | **Control instance** | **Excerpts from the control instance description** |
|---|---|---|
| To prevent loss, damage, theft or compromise of assets and interruption to the organization's operations. | 11.2.7 Secure disposal or re-use of equipment | "…to secure disk erasure, whole-disk encryption reduces the risk of disclosure of confidential information when equipment is disposed of or redeployed…" |
| To protect against loss of data. | 12.3.1 Information backup | "…in situations where confidentiality is of importance, backups should be protected by means of encryption." |
| To ensure the protection of information in networks and its supporting information processing facilities. | 13.1.2 Security of network services | "Security mechanisms, service levels and management requirements of all network services should be identified and included in network services agreements, whether these services are provided in-house or outsourced."<br>"Security features of network services could be…technology applied for security of network services, such as authentication, encryption and network connection controls…" |
| | 13.1.3 Segregation in networks | "The authentication, encryption and user level network access control technologies of modern, standards based wireless networks may be sufficient for direct connection to the organization's internal network when properly implemented." |

Table 6.5 Control description excerpts from ISO/IEC 27018 (ISO/IEC, 2019)

| **Control objective instance** | **Control instance** | **Excerpts from the control instance description** |
|---|---|---|
| To maintain the security of information transferred within an organization and with any external entity. | 13.2.1 Information transfer policies and procedures | "Where possible, cloud service customers should be asked to put additional measures in place (such as encryption) to ensure that the data can only be accessed at the point of destination and not en route." |
| Not available | A.10.6 Encryption of PII transmitted over public data-transmission | "PII that is transmitted over public data-transmission networks should be encrypted prior to transmission." |
| Not available | A.11.4 Protecting data on storage media leaving the premises | "PII on media leaving the organization's premises should be subject to an authorization procedure and should not be accessible to anyone other than authorized personnel (e.g., by encrypting the data concerned)." |

**Step 3: CQ3 instantiation**

We first determine if the **control** associated to the **security function** encryption found by CQ1, namely, policy on the use of cryptographic controls, can be used to instantiate our integrated ontology to answer CQ3. We found the following instances, while the mapping is provided in Tables 6.6 and 6.7.

1. The **control** policy on the use of cryptographic controls *Mitigates* the **threats**;
   a. Data breaches;
   b. Misconfiguration and inadequate change control;
   c. Lack of cloud security architecture and strategy;
   d. Insufficient identity, credential, access, and key management;
   e. Insider threats;
   f. Insecure interfaces and APIs;
   g. Weak control plane;

h. Metastructure and applistructure failures;
i. Limited cloud usage visibility;
j. Abuse and nefarious use of cloud services.

Then, we determine if the **controls** associated to the **security function** encryption found by CQ2, namely, secure disposal or reuse of equipment, information backup, security of network services, segregation in networks, information transfer policies and procedures, encryption of PII transmitted over public data-transmission, and protecting data on storage media leaving the premises can be used to instantiate our integrated ontology to answer CQ3. We found the following instances, while the mapping is provided in Tables 6.8, 6.9, and 6.10.

1. The **control** secure disposal or reuse of equipment *Mitigates* the **threats**;
    a. Data breaches;
    b. Insider threats;
    c. Metastructure and applistructure failures;

2. The **control** security of network services *Mitigates* the **threats**;
    a. Data breaches;
    b. Misconfiguration and inadequate change control;
    c. Lack of cloud security architecture and strategy;
    d. Insider threats;
    e. Weak control plane;
    f. Metastructure and applistructure failures;
    g. Abuse and nefarious use of cloud services;

3. The **control** segregation in networks *Mitigates* the **threats**;
    a. Misconfiguration and inadequate change control;
    b. Lack of cloud security architecture and strategy;
    c. Insider threats;
    d. Weak control plane;

e. Metastructure and applistructure failures;

f. Abuse and nefarious use of cloud services;

4. The **control** information transfer policies and procedures *Mitigates* the **threats**;

   a. Data Breaches;

   b. Misconfiguration and Inadequate Change Control;

   c. Lack of Cloud Security Architecture and Strategy;

   d. Insider Threat;

   e. Insecure Interfaces and APIs;

   f. Weak Control Plane;

   g. Metastructure and Applistructure Failures;

   h. Limited Cloud Usage Visibility;

   i. Abuse and Nefarious Use of Cloud Services.

Although the **control** information backup is mapped to the CSA control retention policy in the CSA CCM, the latter is not mapped to the CSA top threats to cloud computing. As for the **controls** encryption of PII transmitted over public data-transmission and protecting data on storage media leaving the premises, they are not mapped to CSA controls in the CSA CCM and, as a result, no relationship exists with the CSA top threats to cloud computing.

**Step 4: Control and quality measure implementation guidance**

We analyze the available implementation guidance for cloud computing services of the **control** policy on the use of cryptographic controls to determine whether there is evidence to suggest that sufficient information and/or procedures and/or support is provided by the cloud service provider to the cloud service customer to verify the **quality requirement** instances *SpecifiedBy* the **quality measures** data encryption correctness, encryption usage, and strength of cryptographic algorithms.

The implementation guidance for cloud computing services stipulates that “when the cloud service provider offers cryptography, the cloud service customer should review any information supplied by the cloud service provider to confirm whether the cryptographic capabilities meet the cloud service customer’s policy requirements.” It also stipulates that “the cloud service provider should provide information to the cloud service customer regarding the circumstances in which it uses cryptography to protect the information it processes.”

In the case where the cryptographic capability, namely, the **security function** encryption, *IsImplementedBy* the **cloud service provider**, there is evidence to suggest that the information that should be provided by the cloud service provider to the cloud service customer is sufficient to verify the **quality requirements** *Specifiedby* the **quality measures** data encryption correctness, encryption usage, and strength of cryptographic algorithms.

The implementation guidance also stipulates that “the cloud service provider should also provide information to the cloud service customer about any capabilities it provides that can assist the cloud service customer in applying its own cryptographic protection.”

In the case where the cryptographic capability, namely, the **security function** encryption, *IsImplementedBy* the **cloud service customer**, there is evidence to suggest that the information that should be provided by the cloud service provider to the cloud service customer is sufficient to verify the **quality requirement** instances *Specifiedby* the **quality measures** data encryption correctness, encryption usage, and strength of cryptographic algorithms.

**Step 5: Additional instantiation**

No additional instantiations were found.

Table 6.6 Encryption applicability: CQ1 and CQ3 instantiations

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | **X** | | | | | | | | |
| | Misconfiguration and Inadequate Change Control | **X** | | | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | **X** | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | **X** | | | | | | | | |
| | Account Hijacking | | | | | | | | | |
| | Insider Threat | | | | **X** | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | | |
| | Weak Control Plane | | | | | | | | | |
| | Metastructure and Applistructure Failures | | | | | **Quality characteristic** | | Security | | Confidentiality |
| | Limited Cloud Usage Visibility | | | | | **Quality sub-characteristic** | | Confidentiality | | |
| | Abuse and Nefarious Use of Cloud Services | | | | | **Quality measure** | | Data encryption correctness | Strength of cryptographic algorithm | Encryption usage |
| | | Storage and Access | Wireless Security | Supply Chain Agreements | Third Party Audits | **Controls** / **Extracted function** | | Encryption | | Encryption |
| | | EKM-04 | IVS-12 | STA-05 | STA-09 | | | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 10.1.1 | Policy on the use of cryptographic controls | **CQ1** | | **CQ1** |

Table 6.7 Encryption applicability: CQ1 and CQ3 instantiations

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | **X** | **X** | **X** | **X** | | | | | |
| | Misconfiguration and Inadequate Change Control | | **X** | | **X** | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | **X** | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | **X** | **X** | | | | | |
| | Account Hijacking | | | | | | | | | |
| | Insider Threat | | | | **X** | | | | | |
| | Insecure Interfaces and APIs | **X** | **X** | | | | | | | |
| | Weak Control Plane | **X** | **X** | | | | | | | |
| | Metastructure and Applistructure Failures | **X** | **X** | | **X** | **Quality characteristic** | | Security | | Confidentiality |
| | Limited Cloud Usage Visibility | | | | **X** | **Quality sub-characteristic** | | Confidentiality | | |
| | Abuse and Nefarious Use of Cloud Services | | | | **X** | **Quality measure** | | Data encryption correctness | Strength of cryptographic algorithm | Encryption usage |
| | | Data Integrity | Data Integrity/ Security | Entitlement | Sensitive Data Protection | **Controls** / **Extracted function** | | Encryption | | Encryption |
| | | AIS-03 | AIS-04 | EKM-01 | EKM-03 | | | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 10.1.1 | Policy on the use of cryptographic controls | **CQ1** | | **CQ1** |

Table 6.8 Encryption applicability: CQ2 and CQ3 instantiations

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | **X** | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | | |
| | Account Hijacking | | | | | | | | | |
| | Insider Threat | | | | **X** | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | | |
| | Weak Control Plane | | | | | | | | | |
| | Metastructure and Applistructure Failures | | **X** | | | **Quality characteristic** | | Security | | Confidentiality |
| | Limited Cloud Usage Visibility | | | | | **Quality sub-characteristic** | | Confidentiality | | |
| | Abuse and Nefarious Use of Cloud Services | | | | | **Quality measure** | | Data encryption correctness | Strength of cryptographic algorithm | Encryption usage |
| | | Wireless Security | Network/ Infrastructure Services | Retention Policy | Off-site Authorization | **Controls** | **Extracted function** | Encryption | | Encryption |
| | | IVS-12 | STA-03 | BCR-11 | DCS-04 | | | | | |
| | | | | | **CQ3** | 11.2.7 | Secure disposal or reuse of equipment | **CQ2** | | **CQ2** |
| | | | | **CQ3** | | 12.3.1 | Information backup | **CQ2** | | **CQ2** |
| | | **CQ3** | **CQ3** | | | 13.1.2 | Security of network services | **CQ2** | | **CQ2** |
| | | **CQ3** | | | | 13.1.3 | Segregation in networks | **CQ2** | | **CQ2** |
| | | **CQ3** | | | | 13.2.1 | Information transfer policies and procedures | **CQ2** | | **CQ2** |

Table 6.9 Encryption applicability: CQ2 and CQ3 instantiations

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | | |
| | Misconfiguration and Inadequate Change Control | | **X** | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | **X** | **X** | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | |
| | Account Hijacking | | | | | | | | |
| | Insider Threat | **X** | | **X** | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | |
| | Weak Control Plane | | **X** | **X** | | | | | |
| | Metastructure and Applistructure Failures | | | **X** | **Quality characteristic** | | Security | | Confidentiality |
| | Limited Cloud Usage Visibility | | | | **Quality sub-characteristic** | | Confidentiality | | |
| | Abuse and Nefarious Use of Cloud Services | | **X** | | **Quality measure** | | Data encryption correctness | Strength of cryptographic algorithm | Encryption usage |
| | | Third-Party Audits | Network Security | Segmentation | **Controls** | **Extracted function** | Encryption | | Encryption |
| | | STA-09 | IVS-06 | IVS-09 | | | | | |
| | | | | | 11.2.7 | Secure disposal or reuse of equipment | **CQ2** | | **CQ2** |
| | | | | | 12.3.1 | Information backup | **CQ2** | | **CQ2** |
| | | **CQ3** | **CQ3** | | 13.1.2 | Security of network services | **CQ2** | | **CQ2** |
| | | | **CQ3** | **CQ3** | 13.1.3 | Segregation in networks | **CQ2** | | **CQ2** |
| | | | | | 13.2.1 | Information transfer policies and procedures | **CQ2** | | **CQ2** |

Table 6.10 Encryption applicability: CQ2 and CQ3 instantiations

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | **X** | **X** | **X** | **X** | **X** | | | | | |
| | Misconfiguration and Inadequate Change Control | | **X** | | | **X** | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | **X** | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | | | |
| | Account Hijacking | | | | | | | | | | |
| | Insider Threat | | **X** | | | | | | | | |
| | Insecure Interfaces and APIs | | | | **X** | **X** | | | | | |
| | Weak Control Plane | | **X** | | **X** | **X** | | | | | |
| | Metastructure and Applistructure Failures | **X** | **X** | **X** | **X** | **X** | **Quality characteristic** | | Security | | Confidentiality |
| | Limited Cloud Usage Visibility | | **X** | | | | **Quality sub-characteristic** | | Confidentiality | | |
| | Abuse and Nefarious Use of Cloud Services | | **X** | | | | **Quality measure** | | Data encryption correctness | Strength of cryptographic algorithm | Encryption usage |
| | | Ecommerce Transaction | Handling / Labeling / Security Policy | Secure Disposal | Data Integrity | Data Security / Integrity | **Controls** | **Extracted function** | Encryption | | Encryption |
| | | DSI-03 | DSI-04 | DSI-07 | AIS-03 | AIS-04 | | | | | |
| | | | | **CQ3** | | | 11.2.7 | Secure disposal or reuse of equipment | **CQ2** | | **CQ2** |
| | | | | | | | 12.3.1 | Information backup | **CQ2** | | **CQ2** |
| | | **CQ3** | | | | | 13.1.2 | Security of network services | **CQ2** | | **CQ2** |
| | | | | | | | 13.1.3 | Segregation in networks | **CQ2** | | **CQ2** |
| | | | **CQ3** | | **CQ3** | **CQ3** | 13.2.1 | Information transfer policies and procedures | **CQ2** | | **CQ2** |

## 6.4 Evaluation results

During our analysis, we observed that for six out of the seven security functions extracted from the ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015) quality measures, we were able to extract controls from ISO/IEC 27002 (ISO/IEC, 2013) and ISO/IEC 27018 (ISO/IEC, 2019) to instantiate our integrated ontology to answer CQ1 and CQ2 (described in section 5.2). Based on these instantiations, we found that these six security functions were applicable to information security in cloud computing services.

We also found that for the security architecture and the security-focused verification method extracted from the ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015) quality measures, we were able to extract controls from ISO/IEC 27002 (ISO/IEC, 2013) to instantiate our integrated ontology to answer CQ1 and CQ2 (described in section 5.2). Based on these instantiations, we found that the security architecture and the security-focused verification method were applicable to information security in cloud computing services.

These instantiations revealed that the security functions, the security architecture, and the security-focused verification method extracted from the ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015) quality measures are associated with only 25 out of the 114 controls from ISO/IEC 27002 (ISO/IEC, 2013) that are applicable to information security in cloud computing services. They also revealed that none of the security functions or the security architecture or the security-focused verification method are associated with controls specific to cloud computing services from ISO/IEC 27017 (ISO/IEC, 2015). Furthermore, the instantiation of our integrated ontology to answer CQ3 revealed that only 19 out of these 25 controls mitigates the CSA top threats to cloud computing.

It is also important to note that to mitigate threats and vulnerabilities and to modify the risks, information security experts implement a combination of controls. For example, to mitigate each CSA top threat to cloud computing, the CSA recommends a combination of controls from the CCM. Each control from the CCM has been mapped to one or to a combination of controls from ISO/IEC 27001, ISO/IEC 27002, ISO/IEC 27017, and ISO/IEC 27018. While a control that is recommended to mitigate a threat to cloud computing can be associated with a security function and/or the security architecture and/or the security-focused verification method extracted from ISO/IEC 25000 quality models, other controls for which no security function or the security architecture or the security-focused verification method extracted from ISO/IEC 25000 quality models can also be recommended in combination to mitigate the said threat.

Moreover, the application of information security risk management, system security engineering, and systems and software quality engineering processes throughout the life cycle of the target cloud-based IS can result in the implementation of security functions and security architectures as well as in the application of security-focused verification methods missing from ISO/IEC 25000 quality models.

These results therefore indicate that the security functions, the security architecture, and the security-focused verification method extracted from the ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015) quality measures are insufficient in mitigating threats to cloud computing.

Table 6.11 Quality measures applicability results

| **ISO/IEC 25023 and ISO/IEC 25024 quality measure** | **Applicable to information security in cloud computing services** | **Applicable from the cloud service provider perspective** | **With evidence to suggest the applicability from the cloud service customer perspective** | | | |
|---|---|---|---|---|---|---|
| | | | **ISO/IEC 27017** | **ISO/IEC 27018** | **ISO/IEC 25023** | **ISO/IEC 25024** |
| Data encryption correctness | X | X | X | | | |
| Encryption usage | X | X | X | | | |
| Strength of cryptographic algorithms | X | X | X | | | |
| Digital signature usage | X | X | X | | | |
| User audit trail completeness | X | X | X | X | | |
| System log retention | X | X | X | X | | |
| User access traceability | X | X | X | X | | |
| Authentication mechanism sufficiency | X | X | | | | |
| Authentication rules conformity | X | X | | | | |
| Redundancy of components | X | X | | | | |
| Failure avoidance | X | X | | | | |
| System availability | X | X | | . | X | |
| Mean down time | X | X | | | X | |
| Backup data completeness | X | X | X | X | | |
| Periodical backup | X | X | X | X | | |
| Architecture recoverability | X | X | X | X | | |
| Data recoverability ratio | X | X | X | X | | |
| Data availability ratio | X | X | X | X | | |
| Data integrity | X | X | X | X | | |
| Non vulnerability | X | X | | | | X |

We also found that quality measures for which no security function or security architecture or security-focused verification method can be extracted can also be found in additional instantiations where the controls fulfill quality requirements specified by quality measures extracted from ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015) quality measures. This is the case for user access traceability, failure avoidance, system availability, mean down time, data recoverability ratio, data availability ratio, and data integrity.

Finally, during our analysis of implementation guidance for cloud computing services and for the protection of PII in public cloud computing services (extracted from ISO/IEC 27017 (ISO/IEC, 2015) and ISO/IEC 27018 (ISO/IEC, 2019)) of the controls as well as the implementation guidance (extracted from ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015)) of the quality measures, we found evidence to suggest that for 16 out of the 20 quality measures found in the different instantiations the cloud service customer can verify the quality requirements specified by the said quality measures. Results are provided in Table 6.9.

## 6.5 Conclusions

Based on an analysis of the description of the controls from the ISO/IEC 27000 series, evaluation results show that our integration ontology supports the identification of quality characteristics, quality sub-characteristics, and quality measures from the ISO/IEC 25000 quality models applicable to information security in cloud computing services. It also supports the identification of the security functions, the security architecture, and the security-focused verification method extracted from ISO/IEC 25000 quality models that are applicable to information security in cloud computing services.

Our results revealed that although some of the security functions, the security architecture, and the security-focused verification method were applicable to information security in cloud computing services, their application was insufficient in mitigating the CSA top threats to cloud computing.

The results also revealed that no security function or the security architecture or the security-focused verification method was associated with controls specific to cloud computing services from the ISO/IEC 27000 series.

Additionally, we found evidence to suggest that for 16 out of the 20 quality measures found in the different instantiation scenarios the cloud service customer can verify the quality requirements specified by the said quality measures.

Consequently, in the following chapter, we evaluate whether our integrated ontology supports the customization of quality models and quality measures in the context of the application. To do so, we use information inputs associated with the implementation of controls from ISO/IEC 27017 to illustrate how the application of our integration ontology extends the applicability of ISO/IEC 25030 quality requirements framework to the definition of quality requirements related to information security in cloud computing services and to evaluate whether our integrated ontology supports the customization of the quality models and quality measures.

# CHAPTER 7

# AN ILLUSTRATIVE EXAMPLE OF THE APPLICATION OF THE EXTENDED FRAMEWORK TO THE CASE OF INFORMATION SECURITY IN A MULTI-TENANT VIRTUALIZED INFRASTRUCTURE

## 7.1 Introduction

In this chapter, our objective is to illustrate how the application of the integrated ontology extends the applicability of the ISO/IEC 25030 quality requirements framework to the definition of quality requirements related to information security in cloud computing services. To do this, we apply our integrated ontology to extract and classify information from an example found in the scientific literature, namely, information relevant to 1) the definition of quality requirements related to a control from ISO/IEC 27017 specific to cloud computing services and/or with implementation guidance for cloud computing services; and 2) threat and vulnerability information regarding the "shared, on-demand nature of cloud computing" as such threats and vulnerabilities are the focus of security experts from the CSA (Alliance, 2019). The illustration includes the identification of applicable and missing quality characteristics, quality sub-characteristics, and quality measures. It also includes the specification of applicable quality measures as well as the construction and specification of missing quality measures using the guidance from ISO/IEC 25020 and ISO/IEC 25021. We then discuss the application of these quality measures from the perspective of the cloud service customer and the cloud service provider.

## 7.2 Example selection

As previously discussed, we searched the literature to find studies that include:

1. The application of ISO/IEC 27017 controls specific to cloud computing services and/or with implementation guidance for cloud computing services;
2. Threats and vulnerabilities resulting from the shared, on-demand nature of cloud computing.

We found the following relevant studies: a study by Madi et al. (2018) and a study by Madi et al. (2016), which fulfill the selection criteria. These studies provide relevant information on threats and vulnerabilities resulting from the shared, on-demand nature of cloud computing as well as on the application of a control from ISO/IEC 27017 along with implementation guidance for cloud computing services, namely, segregation in networks. Additionally, these studies also provide information on the application of a control from ISO/IEC 27002 applicable to cloud computing services, namely, technical compliance review.

## 7.3 Example presentation

### 7.3.1 Information security in a multi-tenant virtualized infrastructure

The study by Madi et al. (2018) presents a case of multi-tenancy in a cloud computing environment supported by virtualization. This study used a two-layer OpenStack configuration for a multi-tenant virtualized infrastructure as an example. The two layers consist of a physical layer composed of networking, storage, and processing resources and a virtualization layer that runs on top of the physical layer, enabling infrastructure resource sharing. Figure 7.1 provides the virtualization (abstraction) layer which accounts for tenant-specific virtual resources such as virtual networks and virtual machines (VMs). Consequently, a tenant can provision multiple VM instances and virtual networks. Moreover, VMs can run on different hosts and, through means of virtual ports, can connect to many virtual networks. Virtualization techniques guarantee isolation amid multiple tenant boundaries, while host virtualization technologies allow VMs to run on top of the same host. Network virtualization mechanisms (e.g., VLAN and VXLAN) facilitate network traffic segregation of tenants, for which virtual networking devices (e.g., Open vSwitches) play a critical role in connecting both VM instances to their corresponding host machines and to virtual networks. Together with these virtual and physical resources, which are illustrated as nodes, Figure 7.1 provides the relationships between specific tenant resources and cloud provider resources.

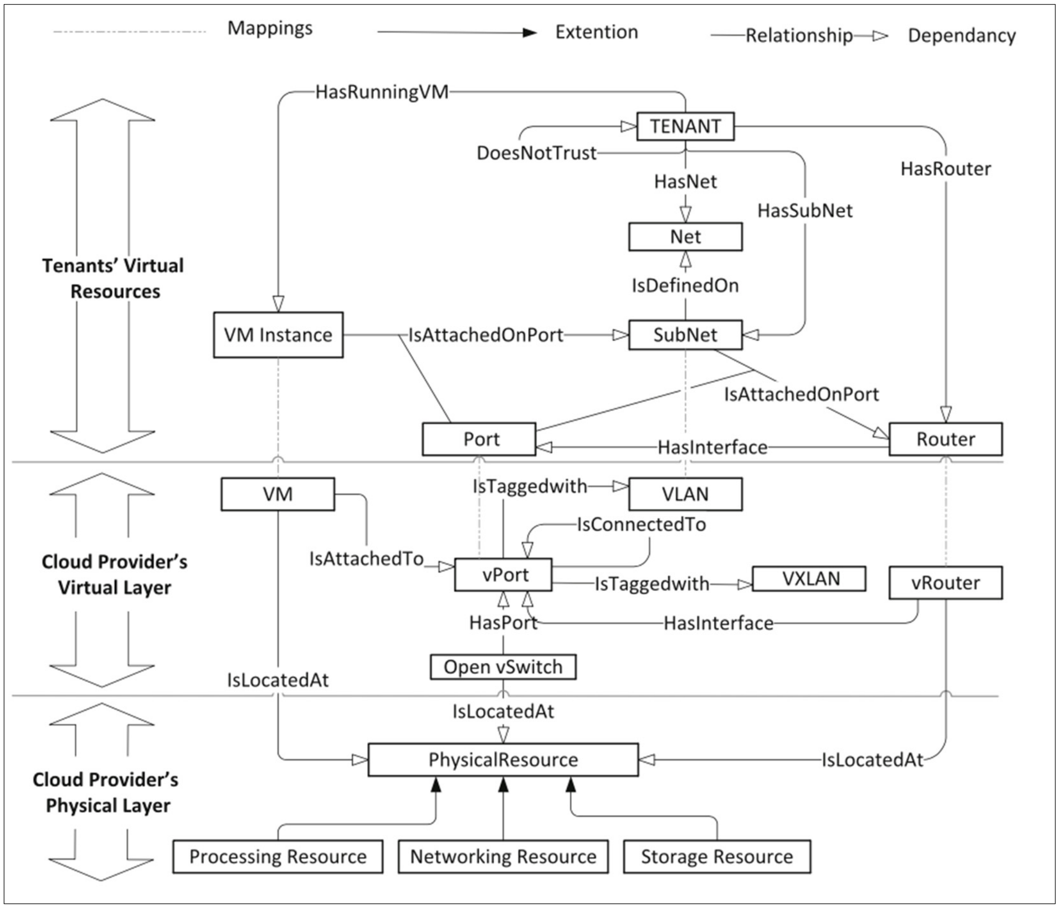


Figure 7.1 A generic model of virtualized infrastructures in cloud computing environments taken from Madi et al. (2016)

### 7.3.2 Information security in a multi-tenant virtualized infrastructure

Madi et al. (2018) built on the assumption that not all tenants trust each other within a multi-tenant virtualized infrastructure. As a result, a tenant can request compliance to information security standards specific to cloud computing services, such as ISO/IEC 27017, to protect its assets from untrusted tenants.

For example, based on its risk assessment, a tenant (i.e., the cloud computing service customer) may require the implementation of controls, such as segregation in networks from ISO/IEC 27017.

Implementation guidance for cloud computing services on control segregation in networks from ISO/IEC 27017 stipulates that "The cloud service customer should define its requirements for segregating networks to achieve tenant isolation in the shared environment and verify that the cloud service provider meets those requirements." Additionally, implementation guidance of the said control stipulates that the cloud service provider should enforce "segregation between tenants in a multi-tenant environment." Moreover, it also stipulates that "Where appropriate, the cloud service provider should help the cloud service customer verify the segregation implemented by the cloud service provider."

Madi et al. (2018) established a bridge between high-level description and implementation guidance for cloud computing services related to controls and their low-level implementation. To implement such recommendations, their study identified a set of security properties related to tenant isolation in a multi-tenant virtualized infrastructure. The focus of the study was on security properties that they acquired from the literature. For this example, our research project focuses on the security property **no VM co-residence**. Table 7.1 provides the security property **no VM co-residence** extracted from Madi et al. (2018).

Table 7.1 Security property extracted from Madi et al. (2018)

| Category | Property | |
|---|---|---|
| | Name | Description |
| Physical isolation | **No VM co-residence** | VMs of a tenant should not be placed on the same compute node as VMs of a non-trusted tenant. |

Although Madi et al. (2018) and Madi et al. (2016) provided information on multiple threats and vulnerabilities, from their study we extracted a scenario involving a threat and vulnerabilities that are documented in the scientific literature.

Madi et al. (2016) reported that according to Zhang's study (Zhang, Juels, Oprea & Reiter, 2011), it is possible to successfully identify the location of a target VM and to trigger the creation of malicious VMs to co-reside within the same host as the target VM. Madi et al. (2016) also pointed out that once co-located to its target, a malicious VM can exploit vulnerabilities within the hypervisor (Perez-Botero, Szefer & Lee, 2013) or use side channel techniques to violate the confidentiality and integrity of other guests or the availability of the hypervisor.

Following implementation guidance for cloud computing services of the control segregation in networks from ISO/IEC 27017, the cloud service customer (tenant) could specify a requirement (the tenant's security policy) to mitigate this threat and vulnerabilities, namely, compute nodes (physical servers) shall not be shared with other tenants.

To do this, in the case of a cloud computing services based on OpenStack, the cloud computing service provider administrator can use the cloud infrastructure management system's (OpenStack software) tenant isolation functions and mechanisms to implement the tenant's requirement (i.e., the tenant's security policy).

However, cloud computing environments are dynamic. Moreover, changes to application deployment configurations, such as VM migration, creation, and removal as well as workload variations and changes in network topology, etc., can result in misconfigurations and violations of security policies (Jarraya, Eghtesadi, Debbabi, Zhang & Pourzandi, 2012). Additionally, as indicated by Matthews, Garfinkel, Hoff & Wheeler (2009), the manual management of security policies in such a dynamic environment is error-prone and can potentially result in missing information and misconfigurations. For example, Figure 7.2 shows an instance of the generic model of virtualized infrastructures in cloud computing environments that Madi et al. (2016) used, illustrating a scenario where the VM migration of VM_02 from CN 85 to CN 96 for load balancing purposes results in the violation of the security property **no VM co-residence** and therefore is in violation of a tenant Alpha security requirement or a policy for which compute nodes (physical servers) shall not be shared with other tenants.

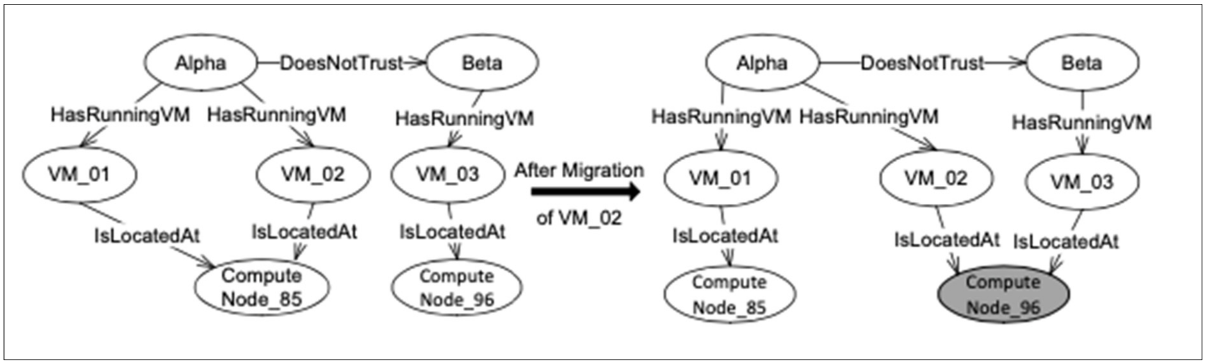


Figure 7.2 VM migration scenario
taken from Madi et al. (2016)

As previously discussed, implementation guidance for cloud computing services of the control segregation in networks from ISO/IEC 27017 stipulates that “The cloud service customer should define its requirements for segregating networks to achieve tenant isolation in the shared environment and verify that the cloud service provider meets those requirements.” In this case, the cloud service customer should verify that the cloud service provider meets its physical isolation requirement (tenant-specific security policy). Additionally, as previously discussed, it stipulates that “Where appropriate, the cloud service provider should help the cloud service customer verify the segregation implemented by the cloud service provider.” In this case, the cloud service provider should help the cloud service customer (tenant) to verify the physical isolation implemented by the cloud service provider.

To support this, Madi et al. (2018) proposed an auditing framework to automatically detect and report violations to security policies in a multi-tenant cloud computing environment. Figure 7.3 provides a high-level architecture of the auditing solution by Madi et al. (2018).

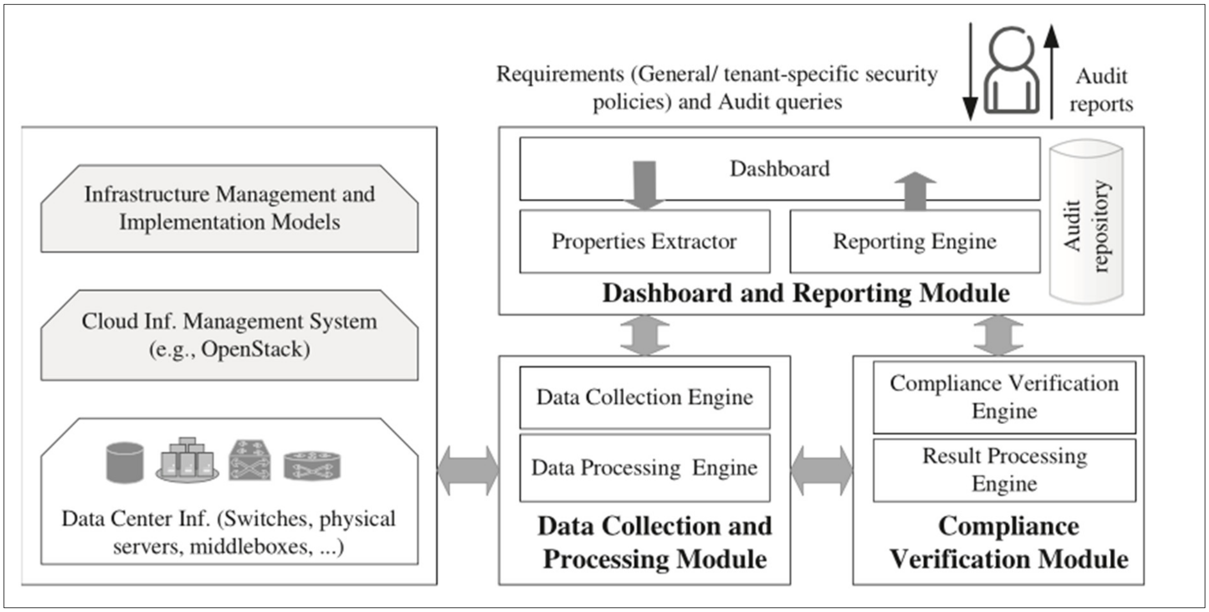

Requirements (General/ tenant-specific security policies) and Audit queries
Audit reports
Infrastructure Management and Implementation Models
Cloud Inf. Management System (e.g., OpenStack)
Data Center Inf. (Switches, physical servers, middleboxes, ...)
Dashboard
Properties Extractor
Reporting Engine
Audit repository
Dashboard and Reporting Module
Data Collection Engine
Data Processing Engine
Data Collection and Processing Module
Compliance Verification Engine
Result Processing Engine
Compliance Verification Module

Figure 7.3 A high-level auditing solution architecture
taken from Madi et al. (2018)

In this example, we focus on the data collection and processing module as well as the security compliance verification module. The data collection and processing module by Madi et al. (2018) is comprised of two sub-engines: a collection engine and a processing engine. The role of the collection engine is to collect the requisite audit data in batch mode, relying on the cloud management system to procure this data; the role of the processing engine is to filter, format, aggregate, and correlate this data. The requisite audit data may be disseminated throughout the cloud under different formats. Furthermore, the data must be pre-processed by the processing engine to provide the specific information necessary to verify the given properties. The final processing step is to generate the code used for compliance validation and subsequently store this code within the audit repository database for use by the compliance validation engine. The generated code is specific to the selected back-end verification engine.

The role of the compliance verification module is to verify de facto audited properties and to detect any violations. When triggered by an audit request or updated inputs, the compliance verification module will invoke the back-end verification engine. In the case of a violation, the verification engine will provide details on any breach, which will then be interpreted by the processing engine. More specifically, Madi et al. (2018) applied formal methods to capture system model and audit properties, facilitating automated reasoning, which is generally more practical and effective compared to the manual inspection approach. In instances for which a security audit property fails, evidence of this failure can be procured from the output of the back-end verification engine. According to Madi et al. (2018), the cloud computing service administrator can use this evidence to identify the root cause of such violations and thereafter fix them.

Table 7.2 First-order logic predicates taken from Madi et al. (2016)

| Relations in Properties | Evaluate to True if |
|---|---|
| *HasRunningVM(vm, t)* | The tenant *t* has a running virtual machine *vm* |
| *DoesNotTrust(t1, t2)* | Tenant *t2* is not trusted by tenant *t1* which means that *t1*’ resources should not share the same hardware with *t2*’ instances |
| *IsLocatedAt(vm, cn)* | The instance *vm* is located at the compute node *cn* |

Tenant security policies, such as physical isolation requirements, correspond to security properties and are expressed as predicates over relational constraints and other predicates. Table 7.2 provides the FOL predicates required for expressing the security property **no VM co-residence**. Madi et al. (2018) indicates that predicates that do not appear as relationships in Figure 7.1 are inferred through the correlation of other available relations.

Based on the audit input data collected, the following FOL predicate (7.1) was used by Madi et al. (2018) to verify that the tenant’s VMs are not co-located in the same compute node along with VMs from untrusted tenants:

$$\begin{aligned} &\forall t1, t2 \ \in \boldsymbol{TENANT}, \forall vm1, vm2 \ \in \boldsymbol{INSTANCE}, (2)\ \forall cn1, cn2 \\ &\quad \in \boldsymbol{COMPUTEN}: HasRunningVM(vm1, t1) \\ &\quad \wedge DoesNotTrust(t1, t2) \wedge IsLocatedAt(vm1, cn1) \\ &\quad \wedge IsLocatedAt(vm2, cn2) \ \rightarrow cn1 \neq cn2 \end{aligned} \tag{7.1}$$

Figure 7.4 provides the different data sources used to collect audit input data and their corresponding layers from the generic model of virtualized infrastructures in cloud computing environments. It is important to note that Madi et al. (2018) indicated that the proposed auditing framework is based on the assumption that the cloud infrastructure management system can be trusted with respect to the integrity of the audit input data collected.

Finally, as a back-end verification mechanism, Madi et al. (2018) proposed formalizing audit data and properties as constraint satisfaction problems (CSPs) and applying a constraint solver (i.e., Sugar (Tamura & Banbara, 2008)) to validate the compliance.

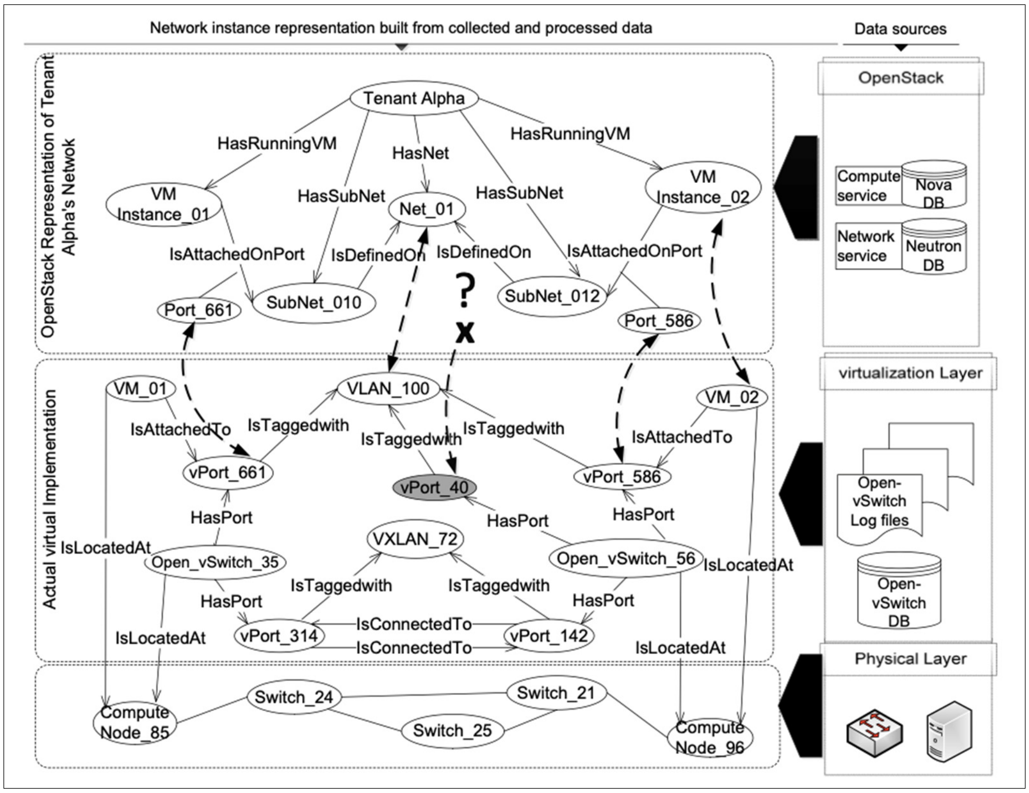


Figure 7.4 Data sources
taken from Madi et al. (2016)

## 7.4 Integrated ontology application

### 7.4.1 Information extraction and classification

Our research project uses the integrated ontology to extract and classify the information relevant to the definition of quality requirements related to information security in cloud computing services from the multi-tenant virtualized infrastructure example presented in section 7.3.

We were able to extract and classify information on the implementation of a control to fulfill tenant security requirements, such as physical isolation requirements to segregate its network from untrusted tenants. Table 7.3 provides the ontology classes and instances of each class, while Table 7.4 provides the list of relationships in the ontology, classes associated with the relations, and instances based on the tenant security requirement. We were also able to identify and extract information on the implementation of a control to mitigate a threat. Table 7.5 provides ontology classes and instances of each class, while Table 7.6 provides the list of relationships in the ontology, classes associated with the relations, and instances based on the mitigation of the threat.

Table 7.3 Classes and instances based on tenant security requirements

| **Class** | **Instance** |
|---|---|
| Asset | Tenant's compute resources and information |
| Threat | Untrusted tenants |
| Vulnerability | Hypervisor vulnerabilities (Perez-Botero et al., 2013) |
| CIA | Confidentiality of tenant's information |
| Risk | Unauthorized access to tenant's information |
| Event | A malicious tenant that has successfully identified the location of a tenant's VM and then creates a malicious VM to co-reside within the same compute node as the tenant's VM. Once co-located with the tenant's VM, the malicious tenant exploits vulnerabilities in the hypervisor (Perez-Botero et al., 2013) to access the tenant's confidential information. |
| Security requirement | The tenant compute nodes (physical servers) shall not be shared with other tenants |
| Control | Segregation in networks |
| Functional requirement | Isolate tenant's compute resources as per tenant's isolation policy |
| Security function | Tenant isolation functions (cloud infrastructure management layer) |
| Security mechanism | Tenant isolation mechanisms (cloud infrastructure implementation layer) |
| Security policy | Tenant's isolation policy |
| Quality goal with conditions | Tenant isolation shall be configured as per the tenant's isolation policy |

Table 7.4 Relationships, associated classes, and instances based on the tenant's security requirement

| **Relation** | **Class involved** | **Instance** |
|---|---|---|
| hasSecurityProperty | Asset, CIA | The tenant's information is confidential |
| affects | Threat, Asset | A malicious tenant can affect the tenant's compute nodes and information |
| exploit | Threat, Vulnerability | The malicious tenant can exploit hypervisor vulnerabilities (Perez-Botero et al., 2013) |
| isRealizedBy | Risk, Event | The risk of unauthorized access to the tenant's information is realized by a malicious tenant that has successfully identified the location of the tenant's VM and then creates a malicious VM to co-reside in the same compute node as the tenant's VM. Once co-located with the tenant's VM, the malicious tenant exploits vulnerabilities in the hypervisor to access the tenant's confidential information |
| fulfills | Control, Security requirement | The control segregation in networks fulfills the tenant's security requirement for the segregation of its compute resources from untrusted tenants |
| modifies | Control, Risk | The control segregation in networks modifies the risk of unauthorized access to the tenant's information |
| mitigates | Control, Threat, Vulnerability | The control segregation in networks mitigates the threat of malicious tenants and hypervisor vulnerabilities (Perez-Botero et al., 2013) |
| is a | Security function, Control | The security function tenant isolation is a control to segregate networks |
| mitigates | Security function, Threat, Vulnerability | The security function tenant isolation mitigates the threat of malicious tenants and hypervisor vulnerabilities (Perez-Botero et al., 2013) |
| specifies | Functional requirement, Security function | The functional requirement for the isolation of tenant's compute resources as per the tenant's isolation policy specifies the security function tenant isolation |
| enforce | Security mechanism, Security policy | The tenant isolation mechanisms enforce the tenant's isolation policy |
| isTheImplementationOf | Security mechanism, Security function | The tenant isolation mechanisms are the implementation of the security function tenant isolation |

Table 7.5 Classes and instances based on the mitigation of the threat

| **Class** | **Instance** |
|---|---|
| Asset | Tenant’s compute resources and information |
| Threat | Untrusted tenants |
| CIA | Confidentiality of tenant’s information |
| Risk | Unauthorized access to tenant’s information |
| Event | A malicious tenant that has successfully identified the location of the tenant’s VM and then creates a malicious VM to co-reside in the same compute node as the tenant’s VM. Once co-located with the tenant’s VM, the malicious tenant exploits vulnerabilities in the hypervisor (Perez-Botero et al., 2013) to access the tenant’s confidential information. |
| Security requirement | The cloud computing service customer (tenant) shall be able to verify that the cloud computing service provider meets the physical isolation requirement (tenant isolation policy) |
| Control | Technical compliance review |
| Functional requirement | Automate the detection and reporting of tenant isolation policy violations |
| Security function | Automated detection and reporting of tenant isolation policy violations |
| Security mechanism | Back-end verification mechanism (Sugar) |
| Security policy | Tenant’s isolation policy |
| Quality goal with conditions | The violation of the physical isolation rule **no VM co-residence** shall be detected correctly and reported in adequate time.<br>There should be **no VM co-residence**. If there is, the rule is violated and as a result the tenant isolation policy is violated. |
| Quality measure | Time to detect and report the violation |

Table 7.6 Relationships, associated classes, and instances based on the mitigation of the threat

| Relation | Class involved | Instance |
|---|---|---|
| hasSecurityProperty | Asset, CIA | The tenant’s information is confidential |
| affects | Threat, Asset | A malicious tenant can affect the tenant’s compute nodes and information |
| exploit | Threat, Vulnerability | The malicious tenant can exploit hypervisor vulnerabilities (Perez-Botero et al., 2013) |
| isRealizedBy | Risk, Event | The risk of unauthorized access to the tenant’s information is realized by a malicious tenant that has successfully identified the location of the tenant’s VM and then creates a malicious VM to co-reside in the same compute node as the tenant’s VM. Once co-located with the tenant’s VM, the malicious tenant exploits vulnerabilities in the hypervisor to access the tenant’s confidential information |
| fulfills | Control, Security requirement | The control technical compliance review fulfills the tenant’s security requirement for the verification of the cloud computing service provider compliance with the physical isolation requirement (tenant isolation policy) |
| modifies | Control, Risk | The control technical compliance review modifies the risk of unauthorized access to the tenant’s information |
| mitigates | Control, Threat, Vulnerability | The control technical compliance review mitigates the threat of malicious tenant and hypervisor vulnerabilities (Perez-Botero et al., 2013) |
| is a | Security function, Control | The security function automated detection and reporting of tenant isolation policy violation is a control to review technical compliance |
| mitigates | Security function, Threat, Vulnerability | The automated detection and reporting of tenant isolation policy violation mitigates the threat of malicious tenant and hypervisor vulnerabilities (Perez-Botero et al., 2013) |
| specifies | Functional requirement, Security function | The functional requirement for the automatic detection and reporting of tenant isolation policy violation specifies the security function automated detection and the reporting of tenant isolation policy violation |
| enforce | Security mechanism, Security policy | The back-end verification mechanism (Sugar) and engine provides evidence that can then help the cloud computing service administrator to identify the root cause of the tenant isolation policy violation and eventually fix it to enforce the tenant isolation policy |
| isTheImplementationOf | Security mechanism, Security function | The back-end verification mechanism (Sugar) is the implementation of the security function automated detection and reporting of tenant isolation policy violations |

We found that two security functions were extracted and classified, namely, tenant isolation and automated detection/reporting of the tenant isolation policy violation.

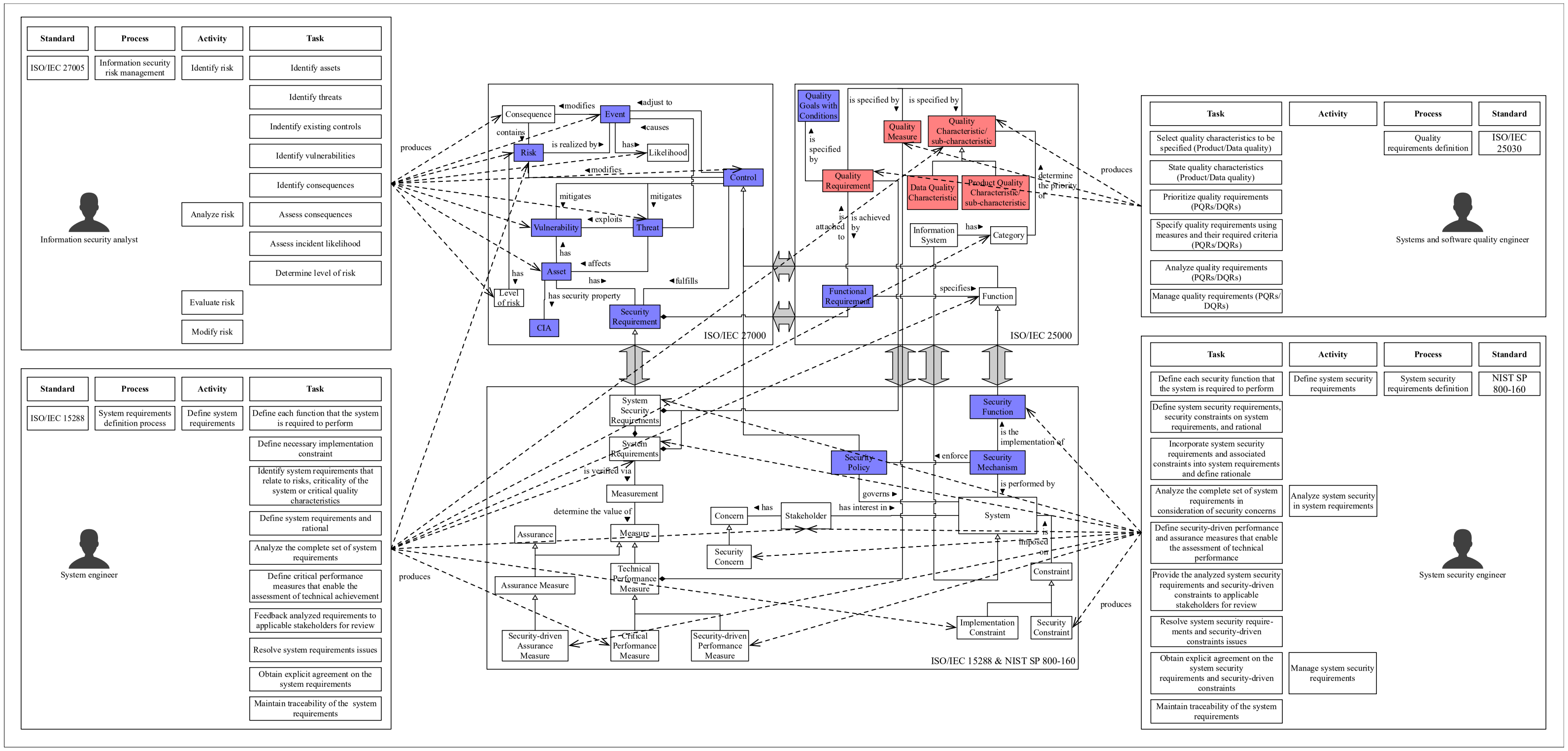
### 7.4.2 Applicability of methods

Based on the mapping between the extracted and classified key artifacts and their associated activities/tasks provided in Figure 7.5, we observed that:

1. The key artifacts threat, vulnerability, risk, event, asset, CIA, security requirement, and control are produced by the activities/tasks from the information security management process which are performed by information security analysts;
2. The key artifact functional requirement is produced by the activities/tasks from the system requirements definition process which are performed by system engineers;
3. The key artifacts security function, security mechanism, and security policy are produced by the activities/tasks from the system security requirements definition process which are performed by system security engineers;
4. The key artifact quality goals with conditions is produced by the activities/tasks from the systems and software quality requirements definition process which are performed by systems and software quality engineers.

While the quality goals with conditions related to the security functions tenant isolation and automated detection/reporting of the tenant isolation policy violation can be extracted and classified, based on the mapping provided in Figure 7.5, we observed that the studies by Madi et al. (2018) and Madi et al. (2016) neither completed the activities/tasks nor documented the produced key artifacts associated to the definition of the quality requirements.

Therefore, in the next section (i.e., sections 7.4.3, 7.4.4, 7.4.5, and 7.4.6), we define the quality requirements.



| Standard | Process | Activity | Task |
|---|---|---|---|
| ISO/IEC 27005 | Information security risk management | Identify risk | Identify assets |
| | | | Identify threats |
| | | | Indentify existing controls |
| | | | Identify vulnerabilities |
| | | | Identify consequences |
| | | Analyze risk | Assess consequences |
| | | | Assess incident likelihood |
| | | | Determine level of risk |
| | | Evaluate risk | |
| | | Modify risk | |

| Standard | Process | Activity | Task |
|---|---|---|---|
| ISO/IEC 15288 | System requirements definition process | Define system requirements | Define each function that the system is required to perform |
| | | | Define necessary implementation constraint |
| | | | Identify system requirements that relate to risks, criticality of the system or critical quality characteristics |
| | | | Define system requirements and rational |
| | | | Analyze the complete set of system requirements |
| | | | Define critical performance measures that enable the assessment of technical achievement |
| | | | Feedback analyzed requirements to applicable stakeholders for review |
| | | | Resolve system requirements issues |
| | | | Obtain explicit agreement on the system requirements |
| | | | Maintain traceability of the  system requirements |

| Task | Activity | Process | Standard |
|---|---|---|---|
| Select quality characteristics to be specified (Product/Data quality) | | Quality requirements definition | ISO/IEC 25030 |
| State quality characteristics (Product/Data quality) | | | |
| Prioritize quality requirements (PQRs/DQRs) | | | |
| Specify quality requirements using measures and their required criteria (PQRs/DQRs) | | | |
| Analyze quality requirements (PQRs/DQRs) | | | |
| Manage quality requirements (PQRs/ DQRs) | | | |

| Task | Activity | Process | Standard |
|---|---|---|---|
| Define each security function that the system is required to perform | Define system security requirements | System security requirements definition | NIST SP 800-160 |
| Define system security requirements, security constraints on system requirements, and rational | | | |
| Incorporate system security requirements and associated constraints into system requirements and define rationale | | | |
| Analyze the complete set of system requirements in consideration of security concerns | Analyze system security in system requirements | | |
| Define security-driven performance and assurance measures that enable the assessment of technical performance | | | |
| Provide the analyzed system security requirements and security-driven constraints to applicable stakeholders for review | | | |
| Resolve system security require-ments and security-driven constraints issues | | | |
| Obtain explicit agreement on the system security requirements and security-driven constraints | Manage system security requirements | | |
| Maintain traceability of the system requirements | | | |

Figure 7.5 Mapping among the extracted and classified key artifacts and their associated activities/tasks

### 7.4.3 Selection of quality characteristics, quality sub-characteristics, and quality measures

For each security function extracted in the preceding section (i.e., section 7.4.1), namely, tenant isolation and automated detection/reporting of the tenant isolation policy violation, we analyzed the category and description content of each control from ISO/IEC 27017 to determine whether the concepts and relationships of our integration ontology can be instantiated to select applicable quality characteristics, quality sub-characteristics, and quality measures extracted from ISO/IEC 25000 quality models.

We found that the control description excerpt provided in Table 7.7 instantiated the integrated ontology with the security function tenant isolation.

Table 7.7 Segregation in networks control description excerpt from ISO/IEC 27002 (ISO/IEC, 2013)

13.1 Network security management

Objective: To ensure the protection of information in networks and its supporting information processing facilities.

13.1.3 Segregation in networks

Control
Groups of information services, users and information systems should be segregated on networks

Implementation guidance
One method of managing the security of large networks is to divide them into separate network domains. The domains can be chosen based on trust-levels […] The segregation can be done using either physically different networks or by using different logical networks (e.g., virtual private networking).

The perimeter of each domain should be well defined. Access between network domains is allowed.

The implementation guidance of the **control** segregation in networks stipulates that “One method of managing the security of large networks is to divide them into separate network domains. The domains can be chosen based on trust-levels…" It also stipulates that "The segregation can be done using either physically different networks or by using different logical networks (e.g., virtual private networking)." Additionally, it stipulates that "The criteria for segregation of networks into domains, and the access allowed through the gateways, should be based on an assessment of security requirements of each domain." Finally, it stipulates that "The assessment should be in accordance with the access control policy […], access requirements, value and classification of information processed…"

In this case, the tenant's information is confidential and the **control** segregation in networks *Fulfills* the tenant's **security requirement** for the segregation of its network from untrusted tenants. Additionally, segregation is done physically using different compute nodes (physical servers) which are implemented using the **security function** tenant isolation.

Consequently, the **control objective** "To ensure the protection of information in networks and its supporting information processing facilities" *isImplementedBy* the **control** segregation in networks and the **security function** tenant isolation to *Fulfill:*

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** security, **quality sub-characteristic** confidentiality, and **quality measure** access controllability defined as the measurement function of values of the quality measure elements:
    a. Number of confidential data items that can be accessed without authorization;
    b. Number of data items that require access control.

Finally, we found the control description excerpt provided in Table 7.8 instantiated the integrated ontology with the security function **automated detection of tenant isolation policy violation**.

Table 7.8 Technical compliance review control description excerpt from ISO/IEC 27002 (ISO/IEC, 2013)

18.2 Information security reviews

Objective: To ensure that information security is implemented and operated in accordance with the organizational policies and procedures.

18.2.3 Technical compliance review

Control
Information systems should be regularly reviewed for compliance with the organization's information security policies and standards.

Implementation guidance
Technical compliance should ideally be reviewed preferably with the assistance of automated tools, which generate technical reports for subsequent interpretation by technical specialist.

Other information
Technical compliance review involves the examination of operational systems to ensure that hardware and software controls have been correctly implemented. This type of compliance review requires specialist technical expertise.

According to its description, the **control objective** is "To ensure that information security is implemented and operated in accordance with the organizational policies and procedures." Additionally, its implementation guidance for cloud computing services stipulates that "Technical compliance should ideally be reviewed preferably with the assistance of automated tools, which generate technical reports…" It also stipulates that "Technical compliance review involves the examination of operational systems to ensure that hardware and software controls have been correctly implemented."

In this case, the **security function** automated detection of tenant isolation policy violation automates the verification of compliance with the tenant's isolation policy.

Consequently, the **control objective** "To ensure that information security is implemented and operated in accordance with the organizational policies and procedures" *IsImplementedBy* the **control** technical compliance review and the **security function** automated detection of tenant isolation policy violation.

Our analysis reveals that the **controls** segregation in networks and technical compliance are applicable to information security in the example by Madi et al. (2018) on the multi-tenant virtualized infrastructure. It also reveals that the **quality characteristic** security, the **quality sub-characteristic** confidentiality, and the **quality measure** access controllability are applicable to the definition of a quality requirement related to information security in the said example.

However, as the **security functions** tenant isolation and automated detection of tenant isolation policy violation are missing from the security functions previously extracted from the ISO/IEC 25023 (ISO/IEC, 2016) and ISO/IEC 25024 (ISO/IEC, 2015) quality measures, and no quality measures are associated with them in the said standards to specify their quality requirements.

Therefore, in the next section, we use the integrated ontology and the guidance from ISO/IEC 25020 (ISO/IEC, 2019) and ISO/IEC 25021 (ISO/IEC, 2012) to select, construct, and specify quality measures for the specification of quality requirements achieved by or attached to these security functions.

### 7.4.4 Specification of quality requirements

ISO/IEC 25020 (ISO/IEC, 2019) stipulates that "…quality measures are selected to satisfy the needs of developers, acquirers, managers, and others for information. In the context of the SQuaRE series, information needs may be defined by quality requirements specifications and product quality evaluations."

In this case, the information need is defined by the **quality requirement** *SpecifiedBy* the **quality goals with conditions**, namely, tenant isolation shall be configured as per tenant's isolation policy and the violation of the physical isolation rule **no VM co-residence** shall be detected correctly and reported in adequate time.

Therefore, as suggested by the ISO/IEC 25020 (ISO/IEC, 2019) guidance, information needs are defined as specific questions that the quality measures must answer. Information needs (INF) are defined as follows:

1. INF1: Is tenant isolation configured as per the tenant's isolation policy?
2. INF2: Is the violation of the tenant isolation policy, namely, the physical isolation rule **no VM co-residence** detected correctly and reported in adequate time?

ISO/IEC 25020 (ISO/IEC, 2019) provides a list of criteria for selecting quality measures and quality measure elements to fulfill information needs. It also stipulates that these criteria shall be documented. From the criteria suggested in the said standard, we selected the following:

1. Relevance to the prioritized information needs;
2. Feasibility of collecting the data in the organizational unit.

We evaluated the quality measures of ISO/IEC 25022 (ISO/IEC, 2016), ISO/IEC 25023 (ISO/IEC, 2016), and ISO/IEC 25024 (ISO/IEC, 2015) to determine whether existing quality measures meet the selection criteria. This evaluation revealed that the **quality measures** functional correctness, mean turnaround time, and turnaround time adequacy meet the selection criteria for INF2. However, no existing quality measures met the selection criteria for INF1.

Therefore, we proceeded with the specification of the quality measures following ISO/IEC 25020 (ISO/IEC, 2019) guidance and using the mandatory items from the template for documenting quality measures presented in Annex C of the standard. Tables 7.9, 7.10, and 7.11 provide the documentation of the **quality measures** functional correctness, mean turnaround time, and turnaround time adequacy, respectively.

In the case of INF1, given that no existing quality measures are applicable, we proceeded with the construction of a new quality measure.

We followed the guidance from ISO/IEC 25020 (ISO/IEC, 2019) which stipulates that "when using a modified or a new measure not identified in ISO/IEC 25022 (ISO/IEC, 2016), ISO/IEC 25023 (ISO/IEC, 2016), or ISO/IEC 25024 (ISO/IEC, 2015), the user shall specify how the measure relates to its corresponding quality model and how it is to be constructed from quality measure elements."

Therefore, we proceeded with the construction and specification of the new quality measure following ISO/IEC 25020 (ISO/IEC, 2019) guidance and using the mandatory items from the template to document quality measures presented in Annex C of the standard. We also followed the guidance from ISO/IEC 25021 (ISO/IEC, 2012) for the construction of a new quality measure element and used the table format for documenting the quality measure elements (QMEs) presented in section 6.2 of this standard. Tables 7.12 and 7.13 show how the proposed **quality measure** tenant isolation rules conformity is constructed from the quality measure element tenant isolation rule.

Additionally, to define how the new quality measure relates to its corresponding quality model, namely, the ISO/IEC 25010 (ISO/IEC, 2019) system and software product quality model, we proposed a new **quality sub-characteristic** for **quality characteristic** security, namely, isolability, which we define as "the degree to which tenant isolation can be configured as specified in the requirements." Table 7.14 shows how the proposed quality measure relates to the ISO/IEC 25010 (ISO/IEC, 2019) system and the software product quality model.

Using this new quality measure and quality sub-characteristic, we can instantiate our integrated ontology.

The **control objective** "To ensure the protection of information in networks and its supporting information processing facilities" *IsImplementedBy* the **control** segregation in networks and the **security function** tenant isolation to *Fulfill:*

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** security, the **quality sub-characteristic** confidentiality, and the **quality measure** access controllability defined as the measurement function of values of the quality measure elements:
    a. Number of confidential data items that can be accessed without authorization;
    b. Number of data items that require access control;

The **functional requirement** that *Specifies* the **security function** tenant isolation *IsAttachedTo* a **quality requirement** *SpecifiedBy* the **quality characteristic** security and the **quality sub-characteristic** isolability and **quality measure** tenant isolation rules conformity defined as the measurement function of values of the quality measure elements:

1. Number of **tenant isolation rules** configured in the infrastructure management system as specified in requirements;
2. Number of **tenant isolation rules** specified in requirements.

Table 7.9 Functional correctness

| **Item** | **Content** |
|---|---|
| System and software quality measure name | Functional correctness |
| System and software product quality characteristic | Functional suitability |
| System and software product quality sub-characteristic | Functional correctness |
| Information needs | What proportion of functions provides the correct result?<br>Is the violation of the physical isolation rule **no VM co-residence** detected correctly? |
| Measurement function | $\mathrm{X} = \mathrm{A}/\mathrm{B}$ |
| Quality measure elements | $\mathrm{A} = \text{Number of functions that are incorrect}$<br>$\mathrm{B} = \text{Number of functions considered}$ |
| Measurement method | The degree of correctness is determined for an individual security function, namely, automated detection of the violation of the physical isolation rule **no VM co-residence**. This security function is tested to determine whether it provides the correct result. |
| Data source | Policy verification mechanism (Sugar) log files<br>Physical isolation rule **no VM co-residence** specification.<br>Model for the infrastructure management layer mapped into an implementation-specific model of the infrastructure layer.<br>Infrastructure management system configuration files, log files, and databases.<br>Implemented system configuration files, log files, and databases. |

Table 7.10 Mean turnaround time

| **Item** | **Content** |
|---|---|
| System and software quality measure name | Mean response time |
| System and software product quality characteristic | Performance efficiency |
| System and software product quality sub-characteristic | Time-behavior |
| Information needs | What is the mean time taken for completion of a job or an asynchronous process? |
| Measurement function | $\mathrm{X} = \sum_{\mathrm{i}=1 \text{ to } \mathrm{n}} (\mathrm{B_i} - \mathrm{A_i})/\mathrm{n}$ |
| Quality measure elements | $\mathrm{A_i} = \text{Time of starting a job i}$<br>$\mathrm{B_i} = \text{Time of completing the job i}$<br>$\mathrm{n} = \text{Number of measurements}$ |
| Measurement method | The job is to detect and report the first violation of the physical isolation rule **no VM co-residence**.<br>Each series of measurements is done for the same number of processing nodes, number of VMs, tenant isolation rule verified, and use case.<br>Example: 10 processing nodes, 5K VMs, physical isolation rule **no VM co-residence**, detect and report the first violation (offline) |
| Data source | Policy verification mechanism (Sugar) log files |

Table 7.11 Turnaround time adequacy

| **Item** | **Content** |
|---|---|
| System and software quality measure name | Turnaround time adequacy |
| System and software product quality characteristic | Performance efficiency |
| System and software product quality sub-characteristic | Time-behavior |
| Information needs | How well does the turnaround time meet the specified targets?<br>Is the violation of the physical isolation rule **no VM co-residence** reported in adequate time? |
| Measurement function | $\mathrm{X} = \mathrm{A}/\mathrm{B}$ |
| Quality measure elements | A = Mean turnaround time measured (refer to Table 7.10)<br>B = Target turnaround time specified in requirements |
| Measurement method | Refer to Table 7.10 |
| Data source | Mean turnaround time measured and requirements |

Table 7.12 Tenant isolation rules configuration conformity

| **Item** | **Content** |
|---|---|
| System and software quality measure name | Tenant isolation rules conformity |
| System and software product quality characteristic | Security |
| System and software product quality sub-characteristic | Isolability |
| Information needs | What proportion of the **tenant isolation rules** is configured in the infrastructure management system as specified in requirements? |
| Measurement function | X = A/B |
| Quality measure elements | A = Number of **tenant isolation rules** configured in the infrastructure management system as specified in requirements<br>B = Number of **tenant isolation rules** specified in requirements |
| Measurement method | Refer to Table 7.13 |
| Data sources | Refer to Table 7.9 |

Table 7.13 Tenant isolation rules

| | |
|---|---|
| QME name | Number of **tenant isolation rules** |
| Target entity | Tenant isolation rules |
| Objectives and properties to quantify | The objective is to count the number of tenant isolation rules configured in the infrastructure management system with reference to the tenant isolation rules specified in requirements.<br>Tenant isolation rule is the property to quantify. |
| Measurement method | A: Count the number of tenant isolation rules configured as specified in requirements in the infrastructure management system interface or configuration files.<br>B: Count the number of tenant isolation rules specified in requirements. |
| QME inputs | Tenant isolation requirements and infrastructure management system configuration files |
| Unit of measurement for the QME | Tenant isolation rules |
| Numerical rules | Adding |
| Scale type | Ratio |
| Context of the QME | This QME is used for the quality measure tenant isolation rules conformity |

Table 7.14 Tenant isolation rules conformity quality measure and isolability quality sub-characteristic

| Function | Quality measure | Quality sub-characteristic | Quality characteristic |
|---|---|---|---|
| No security function specified | Access controllability | Confidentiality | Security |
| Data encryption | Data encryption correctness | | |
| | Strength of cryptographic algorithms | | |
| **Tenant isolation** | **Tenant isolation rules conformity** | **Isolability** | |

### 7.4.5 Analysis of quality requirements

As previously discussed, the diversity offered by cloud computing services (ranging from consumer control of the entire software stack to the application of domain-specific platforms (Armbrust et al., 2009)) constitutes a conceptual shift from conventional computing environments. This diversity in the level of control offered to the cloud computing service customer can impact the applicability of the quality measures. For instance, from our analysis of control implementation guidance for cloud computing services we observed that while in some cases evidence was found to indicate that sufficient information and/or procedures and/or support is provided by the cloud service provider to the cloud service customer to verify the quality requirement specified by the quality measures, we did not find such evidence in other cases.

Consequently, we analysed the applicability of the **quality measures** specified in section 7.4.3, namely, functional correctness, mean turnaround time, turnaround time adequacy, and tenant isolation rules conformity, as well as the **quality measure** identified in section 7.4.2, namely, access controllability. Table 7.15 provides the results of this analysis.

As discussed in the preceding sections, the **security function** automated detection of tenant isolation policy violation automates the verification of compliance with the tenant's isolation policy **no VM co-residence**. If the violation of the tenant's isolation policy **no VM co-residence** is not detected and reported correctly and in adequate time, the tenant's security requirement is not met and the cloud service provider cannot correct the configuration in a timely manner, leaving the tenant's confidential information open to unmitigated threats and vulnerabilities. Therefore, the quality engineering of the said security function, which includes its quality evaluation, is essential to protect the tenant's confidential information.

Regarding the multi-tenant virtualized infrastructure presented in the preceding section, to test the **security function** automated detection of tenant isolation policy violation to determine whether it provides the correct result, access to the infrastructure management system is required to carry out changes to the virtualized infrastructure that will violate the physical isolation rule **no VM co-residence**. In this case, we consider that the infrastructure management system is only accessible to the cloud service provider. Consequently, the **quality measure** functional correctness is only applicable from the perspective of the cloud service provider. Regarding the **quality measure** mean turnaround time, each measurement series is conducted for the same number of processing nodes, number of VMs, the verified tenant isolation rule, and use case. In this case, we consider that the number of processing nodes allocated to the security function is controlled by the provider of the said function, namely, the cloud service provider. Additionally, the quality measure elements of the **quality measure** mean turnaround time are comprised of a "job", which is used to detect and report the first violation of the physical isolation rule **no VM co-residence**. In this case, as previously discussed, changes in the virtualized infrastructure required to violate the physical isolation rule **no VM co-residence** can only be performed by the cloud service provider as it necessitates access to the information management system. Consequently, the **quality measure** mean turnaround time is only applicable from the perspective of the cloud service provider. Regarding the **quality measure** turnaround time adequacy, one of its key quality measure elements is the mean turnaround time measured. Consequently, given that the **quality measure** turnaround time adequacy depends on the mean turnaround time measured, it is also only applicable from the perspective of the cloud service provider.

Regarding the **quality measure** tenant isolation rules conformity, we defined one of its quality measure elements as the number of tenant isolation rules configured in the infrastructure management system as specified in requirements. As previously discussed, in the multi-tenant virtualized infrastructure presented in the preceding section, we consider that the infrastructure management system is only accessible to the cloud service provider. However, implementation guidance of the **control** segregation in networks stipulates that “Where appropriate, the cloud service provider should help the cloud service customer verify the segregation implemented by the cloud service provider.” Consequently, we consider that the **quality measure** tenant isolation rules conformity is applicable from both the perspective of the cloud service customer and cloud service provider.

Finally, in the case of the **quality measure** access controllability, the quality requirement specified by the said quality measure is fulfilled by the **control** segregation of the network. This quality requirement can be verified via a **security-focused verification method**, such as network segmentation check, which consists of performing tests from each untrusted zone to verify that the tenant zone is unreachable and where the tenant zone includes the VMs and the physical servers hosting them. As previously discussed, the implementation guidance of the **control** segregation in the network stipulates that “Where appropriate, the cloud service provider should help the cloud service customer verify the segregation implemented by the cloud service provider.” Consequently, we consider that the **quality measure** access controllability is applicable from both the perspectives of the cloud service customer and the cloud service provider.

Table 7.15 Applicability results of quality measures

| **Quality measure** | **Applicable from the cloud service customer perspective** | **Applicable from the cloud service provider perspective** |
|---|---|---|
| Access controllability | X | X |
| Functional correctness | | X |
| Mean turnaround time | | X |
| Turnaround time adequacy | | X |
| Tenant isolation rules conformity | X | X |

### 7.4.6 Management of quality requirements

The management of quality requirements entails the establishment and maintenance of the traceability of quality requirements to their sources. As previously discussed, in systems and software engineering, a trace is “a specified triplet of elements comprising: a source artifact, a target artifact and a link associating the two artifacts” (Gotel et al., 2012). According to Wheatcraft et al. (2016), each requirement must be able to be traced to its source to identify where the requirement came from and/or how it was arrived at.

We select the quality requirement specified by the quality characteristic security, quality sub-characteristic isolability, and quality measure tenant isolation rules conformity to illustrate that the application of the integrated ontology establishes the traceability of the quality requirement to its sources. Table 7.16 shows the key artifacts, and Table 7.17 shows the links between the key artifacts.

Table 7.16 Key artifacts and instances

| **Key artifact (Class)** | **Instance** |
|---|---|
| Asset | Tenant’s compute resources and information |
| Threat | Untrusted tenants |
| Vulnerability | Hypervisor vulnerabilities (Perez-Botero et al., 2013) |
| CIA | Confidentiality of tenant’s information |
| Risk | Unauthorized access to tenant’s information |
| Event | A malicious tenant that has successfully identified the location of a tenant’s VM and then creates a malicious VM to co-reside within the same compute node as the tenant’s VM. Once co-located with the tenant’s VM, the malicious tenant exploits vulnerabilities in the hypervisor (Perez-Botero et al., 2013) to access the tenant’s confidential information. |
| Security requirement | The tenant compute nodes (physical servers) shall not be shared with other tenants |
| Control | Segregation in networks |
| Functional requirement | Isolate tenant’s compute resources as per tenant’s isolation policy |
| Security function | Tenant isolation function (cloud infrastructure management layer) |
| Security mechanism | Tenant isolation mechanism (cloud infrastructure implementation layer) |
| Security policy | Tenant’s isolation policy |
| Quality goal with conditions | Tenant isolation shall be configured as per the tenant’s isolation policy |
| Quality characteristic | Security |
| Quality sub-characteristic | Isolability |
| Quality measure | Tenant isolation rules conformity |

Table 7.17 Links, associated key artifacts, and instances

| **Link**<br>**(Relation)** | **Key artifact involved**<br>**(Class involved)** | **Instance** |
|---|---|---|
| hasSecurityProperty | Asset, CIA | The tenant's information is confidential |
| affects | Threat, Asset | A malicious tenant can affect the tenant's compute nodes and information |
| exploit | Threat, Vulnerability | The malicious tenant can exploit hypervisor vulnerabilities (Perez-Botero et al., 2013) |
| isRealizedBy | Risk, Event | The risk of unauthorized access to the tenant's information is realized by a malicious tenant that has successfully identified the location of the tenant's VM and then creates a malicious VM to co-reside in the same compute node as the tenant's VM. Once co-located with the tenant's VM, the malicious tenant exploits vulnerabilities in the hypervisor to access the tenant's confidential information |
| fulfills | Control, Security requirement | The control segregation in networks fulfills the tenant's security requirement for the segregation of its compute resources from untrusted tenants |
| modifies | Control, Risk | The control segregation in networks modifies the risk of unauthorized access to the tenant's information |
| mitigates | Control, Threat, Vulnerability | The control segregation in networks mitigates the threat of malicious tenants and hypervisor vulnerabilities (Perez-Botero et al., 2013) |
| is a | Security function, Control | The security function tenant isolation is a control to segregate networks |
| mitigates | Security function, Threat, Vulnerability | The security function tenant isolation mitigates the threat of malicious tenants and hypervisor vulnerabilities (Perez-Botero et al., 2013) |
| specifies | Functional requirement, Security function | The functional requirement for the isolation of tenant's compute resources as per the tenant's isolation policy specifies the security function tenant isolation |
| enforce | Security mechanism, Security policy | The tenant isolation mechanisms enforce the tenant's isolation policy |
| isTheImplementationOf | Security mechanism, Security function | The tenant isolation mechanisms are the implementation of the security function tenant isolation |
| isSpecifiedBy | Quality requirement, Quality characteristic, Quality sub-characteristic, Quality measure, Quality goal with conditions | The quality requirement is specified by the quality characteristic security, the quality sub-characteristic isolability, the quality measure tenant isolation rules conformity, and the quality goal with conditions: tenant isolation shall be configured as per the tenant's isolation policy |
| isAttachedTo | Functional requirement, Quality requirement | The functional requirement for the isolation of the tenant's compute resources as per the tenant's isolation policy is attached to the quality requirement |

## 7.5 Conclusions

The example we used to illustrate the applicability of our integrated ontology to the case of information security in multi-tenant virtualized infrastructures demonstrated that it fulfills our design objectives:

1. Extend the applicability of ISO/IEC 25030 quality requirement framework to information security in cloud computing services;
    a. Integrate the complementary frameworks;
    b. Extend the structure of the quality requirements definition process;
    c. Support communication among stakeholders, namely, information security analysts, system and software quality engineers, and system and software security engineers;
        i. As previously discussed, given that the ISO/IEC/IEEE 15288 (ISO/IEC/IEEE, 2015), NIST SP 800-160 (NIST, 2018), ISO/IEC 25030 (ISO/IEC, 2019), and ISO/IEC 27005 (ISO/IEC, 2018) frameworks and their methods include reference or mapping relationships, stakeholders should consider all associated key artifacts (i.e., requirements and requirements attributes) during the definition of quality requirements related to information security in cloud computing services. By applying the integrated ontology, we defined quality requirements that are associated with key artifacts produced by activities/tasks from the information security management process, the system requirements definition process, the system security requirements definition process, and the systems and software quality requirements definition process. These activities/tasks are performed by information security analysts, system engineers, system security engineers, and systems and software quality engineers;
2. Support the traceability between quality requirements and their sources;
    a. By applying the integrated ontology, we established the traceability of quality requirements to their sources.

Additionally, this example highlights the criticality of the quality engineering of security functions, including their quality evaluation, to achieve information security objectives. However, our analysis revealed that the quality measures specified for the quality evaluation of such security functions are in some cases only applicable from the perspective of the cloud service provider.

# CONCLUSION

## 8.1 Introduction

This research project set out to investigate the applicability of systems and software quality engineering methods and models to information security in cloud computing services.

Our literature review revealed that on the one hand researchers have argued that ISO/IEC 25000 quality models and quality measures used in the application of systems and software quality engineering do not cover cloud computing service-specific quality characteristics (Choi & Jeong, 2014) or do not address cloud computing service-specific technical quality characteristics in sufficient detail due to their generic nature (Wollersheim & Krcmar, 2014). On the other hand, our literature review also revealed the lack of a systematic approach to evaluate their applicability in cloud computing services, while further revealing that the ISO/IEC 25030 quality requirements framework which guides their application is not used by researchers to identify applicable and missing quality characteristics, quality sub-characteristics, and quality measures. Additionally, some researchers do not evaluate the applicability of ISO/IEC 25000 quality models and quality measures to cloud computing services prior to proposing cloud-specific quality models and quality measures. Finally, our literature review revealed that security is perceived by a select group of IT outsourcing experts as the most important non-functional requirement of cloud computing services.

Given that the applicability of both ISO/IEC 25000 quality models and quality measures to information security in cloud computing services has not been fully investigated, this research project aimed to answer the research question “To what extent do ISO/IEC 25000 quality models support the definition of quality requirements related to information security in cloud computing services?”

## 8.2 Research achievements

To answer the research question, our goal was to use the ISO/IEC 25030 quality requirement framework as a systematic approach. Consequently, our initial research objective was to first evaluate the applicability of the framework to information security in cloud computing services.

To do so, we extracted and analyzed the methods, models, and guidance from the ISO/IEC 25030 quality requirements framework that relate to information security and cloud computing services to determine whether they include the concepts within these domains (quality requirements, information security, and cloud computing services) and whether their interrelationships for the definition of quality requirements are defined and documented for the users. When we identified relationships to other standards that are associated with information security and cloud computing services, we also extracted and analyzed their methods, models, and guidance.

We found that ISO/IEC 25023 stipulates that "The determination of the required security functions and the assurance of their effectiveness have been addressed extensively in related International Standards." However, our findings showed that the integration of the frameworks required for this to be achieve in the case of cloud computing services has not been documented for the benefit of users. Moreover, as recommended in ISO/IEC 25023, "The user of this International Standard has to determine what kind of security functions need to be used in each case depending on the level of risk." Our research showed that the ISO/IEC/IEEE 15288, NIST SP 800-160, ISO/IEC 25030, and ISO/IEC 27005 frameworks provide the support necessary to apply this recommendation in the case of cloud computing services. However, these frameworks differ from one another in both their approach and concept. Furthermore, if relationships exist between their concepts, they have not been defined nor documented for the users of these frameworks.

Consequently, our second research objective was to integrate the ISO/IEC/IEEE 15288, NIST SP 800-160, ISO/IEC 25030, and ISO/IEC 27005 frameworks. To the best of our knowledge, prior to this research project, neither has the scientific literature nor the standards defined their complementarity or integration. In integrating IS frameworks, metamodeling has proven to be an effective approach for IS researchers (Goeken & Alter, 2009). Given that ontological metamodeling focuses on describing the concepts that exist within a given domain as well as their respective properties (Goeken & Alter, 2009), we applied ontological metamodeling (Agrawal, 2016) and semantic mapping (Rizopoulos & Mçbrien, 2005) techniques to the construction of an integrated ontology which extends the applicability of the ISO/IEC 25030 quality requirements framework to the definition of quality requirements related to information security in cloud computing services.

The integrated ontology shows that in some cases:

1. Control objectives can be implemented by controls and security functions to fulfill quality requirements specified by ISO/IEC 25000 quality models and quality measures;
2. Quality requirements specified by ISO/IEC 25000 quality models and quality measures can be achieved by functional requirements that specify security functions;
3. Functional requirements that specify security functions can be attached to quality requirements specified by ISO/IEC 25000 quality models and quality measures;
4. Security architectures can implement quality requirements specified by ISO/IEC 25000 quality models and quality measures;
5. Quality requirements specified by ISO/IEC 25000 quality models and quality measures can be verified via security-focused verification methods.

Our third research objective was to evaluate the applicability of the extended ISO/IEC 25030 quality requirements framework through its application to answer our research question "To what extent do ISO/IEC 25000 quality models support the definition of quality requirements related to information security in cloud computing services?"

To do so, we instantiated the integrated ontology using the information extracted from the description of ISO/IEC 27002, ISO/IEC 27017, and ISO/IEC 27018 controls and the CSA top threats to cloud computing as well as the quality characteristics and quality sub-characteristics from the ISO/IEC 25010 and ISO/IEC 25012 quality models and quality measures from ISO/IEC 25023 and ISO/IEC 25024. Such instantiation confirmed that our integrated ontology supports the identification of quality characteristics, quality sub-characteristics, and quality measures applicable to information security in cloud computing services. It also confirmed that it supports the identification of the security functions, the security architecture, and the security-focused verification method (previously extracted from ISO/IEC 25023 and ISO/IEC 25024 quality measures) that are applicable to information security in cloud computing services.

Although the results revealed that for six out of the seven security functions, the security architecture and the security-focused verification method were applicable to information security in cloud computing services (i.e., which were found in the instantiations), the results also suggest that their application is insufficient in mitigating the CSA top threats to cloud computing.

In fact, the results revealed that they were only associated to 25 out of the 114 controls from ISO/IEC 27002 that were applicable to information security in cloud computing services. We also observed that only 19 out of these 25 controls were mapped to controls from the CCM that are recommended to mitigate the CSA top threats to cloud computing.

Additionally, to mitigate threats and vulnerabilities and to modify the risks, information security practitioners implement a combination of controls. For example, to mitigate each CSA top threat to cloud computing, the CSA recommended a combination of controls from the CCM. Each control from the CCM has been mapped to one or to a combination of controls from ISO/IEC 27001, ISO/IEC 27002, ISO/IEC 27017, and ISO/IEC 27018.

While the CSA recommends controls to mitigate the CSA top threats to cloud computing, the application of the ISO/IEC 27005 information security risk management process can result in the selection of additional controls based on application context. Additionally, the application of NIST SP 800-160 system security engineering and ISO/IEC 25030 systems and software quality requirements definition processes can result in the implementation of security functions and security architectures that are missing from those that can be extracted from ISO/IEC 25023 and ISO/IEC 25024 quality measures. Furthermore, based on the integrated ontology, these controls and security functions can potentially fulfill quality requirements specified by quality characteristics, quality sub-characteristics, and quality measures that are missing from the quality models and quality measures. Regarding the functional requirements that specify these security functions, they can potentially be attached to quality requirements specified by quality characteristics, quality sub-characteristics, and quality measures that are missing from the quality models and quality measures. Additionally, these security architectures can potentially implement quality requirements specified by quality characteristics, quality sub-characteristics, and quality measures that are missing from quality models and quality measures. Finally, these quality requirements can potentially be verified by security-focused verification methods that are missing from the quality measures.

Additionally, analysis results from implementation guidance for cloud computing services and for the protection of PII in public cloud computing services of the controls revealed that for 16 out of the 20 quality measures applicable to information security in cloud computing services there was evidence to suggest that the cloud service customer can verify the quality requirements specified by the said quality measures.

These results showed that by applying our integrated ontology, we were able to answer our research question, namely, to identify applicable quality characteristics, quality sub-characteristics, and quality measures.

However, our integrated ontology must also support the customization of quality models and quality measures in the context of the application. Consequently, our fourth research objective was to evaluate whether our integrated ontology also supports this customization. To do so, we applied our integrated ontology to the case of information security in IaaS, and more specifically in multi-tenant virtualized infrastructures. This case was selected given that industry security experts have expressed the need to focus on information security threats, vulnerabilities, and risks, which often result from the “shared, on-demand nature of cloud computing” (Alliance, 2019).

Using our integrated ontology, we identified applicable and missing quality characteristics, quality sub-characteristics, and quality measures for the definition of quality requirements related to a control from ISO/IEC 27017, namely, segregation in networks, and the security function tenant network isolation. We then proceeded with the specification of applicable quality measures as well as the construction and specification of missing quality measures using guidance from ISO/IEC 25020 and ISO/IEC 25021. This resulted in a customized quality model with the addition of a new quality sub-characteristic, namely, tenant network isolability, which we defined as *the capability of a product to meet tenant network isolation requirements*, and a new quality measure*,* namely, tenant isolation rules conformity.

## 8.3 Theoretical implications

As discussed in our literature review, Wollersheim & Krcmar (2014) argued that some specific technical quality “aspects” related to cloud computing services, such as those outlined by Repschlaeger et al. (2013) or Benlian et al. (2011), have not been addressed in detail because of the generic nature of the ISO/IEC 25010 standard. Such technical quality “aspects” include cloud computing service security.

This is consistent with our results that indicated that the application of ISO/IEC 25010 is insufficient to mitigate the CSA top threats to cloud computing.

More specifically, Repschlaeger et al. (2013) included “IT Security and Compliance” in their model to evaluate cloud computing service providers as a target dimension with “Data Center

Protection", "Network Protection", "Operations Protection", and "IT Compliance" as abstract requirements and "Building Safety", "Connection Opportunities", "Communication Security", "Application Access (Identity Management)", "Application Protection", "Data Center Location", and "Data Protection" as evaluation criteria. However, no measures were proposed for their evaluation.

These target dimensions, abstract requirements, and evaluation criteria can in fact be mapped to ISO/IEC 27001 and ISO/IEC 27002 security clauses, which comprise of security categories, where each security category contains a control objective that asserts what is to be achieved as well as one or more control that can be applied to achieve the control objective. For instance, the "Data Center Protection" abstract requirement and its associated evaluation criteria "Building Safety" can be mapped to the security clause "physical and environmental security". Other examples are the "Network Protection" abstract requirement and its associated evaluation criteria, namely, "Communication Security", which can be mapped to the security clause "communication security", and the "Operation Protection" abstract requirement and its associated evaluation criteria, namely, "Application Access (Identity Management)", which can be mapped to the security clause "access control".

As was also discussed in our literature review, Wen & Dong (2013) proposed a quality model for the quality evaluation of SaaS from the perspective of the service provider and customer separately. This proposed quality model includes ISO/IEC 25010 quality characteristics as well as what Wen & Dong (2013) refers to as the "security metrics" from ISO/IEC 27001. According to this study, by using their quality model, customers can evaluate the service provider, while the service provider can use these metrics for quality management. However, what is referred by Wen & Dong (2013) as security metrics, such as physical and environmental security and communication security (network security), are in fact ISO/IEC 27001 and ISO/IEC 27002 security clauses. For instance, the communication security clause contains the network security management and information transfer security categories. The security category "network security management" contains a control objective, and the network controls security of network services as well as segregation in networks. The security category “information transfer” contains a control objective as well as the controls information transfer policies and procedures, agreements on information transfer, electronic messaging, and confidentiality or non-disclosure agreements.

Although the Wollersheim & Krcmar (2014) model designed to evaluate cloud computing service providers includes the target dimension “IT Security and Compliance” and its associated requirements and evaluation criteria, we recommend applying information security standards, such as ISO/IEC 27005, ISO/IEC 27017, and ISO/IEC 27018, in selecting cloud computing services information security controls based on an application context. We also recommend the same approach in the case of Wen & Dong (2013) to complete and extend the proposed quality model for the evaluation of SaaS.

Additionally, in both cases, measures used for the evaluation of performance and effectiveness of the controls should be defined with guidance from ISO/IEC 27004 where performance measures are defined as “measures that express the planned results in terms of the characteristics of the planned activity, such as head counts, milestone accomplishment, or the degree to which information security controls have been implemented”, while effectiveness measures are defined as “measures that express the effect that realization of the planned activities has on the organization’s information security objectives”.

Furthermore, our research revealed that the ISO/IEC 27004 framework is grounded in the ISO/IEC 15939 measurement information model. The latter refers to INCOSE-TP-2003-020-01. Based on the definitions provided by INCOSE-TP-2003-020-01, technical performance measures are partly comprised of quality measures. Additionally, we have observed that while ISO/IEC 25020 is also grounded in the ISO/IEC 15939 measurement information model, it provides a specialization for the definition of quality measures.

These concepts, namely, technical performance measures and quality measures, as well as their interrelationships, were captured by our integrated ontology. As previously discussed, our integrated ontology extends the applicability of the ISO/IEC 25030 quality requirements definition framework to information security in cloud computing services through the integration of the ISO/IEC/IEEE 15288, NIST SP 800-160, ISO/IEC 25030, and ISO/IEC 27005 frameworks. Given that our integrated ontology includes ISO/IEC 27005 concepts used for the selection of applicable ISO/IEC 27017 and ISO/IEC 27018 cloud computing information security controls and can be used under ISO/IEC 27036-4 guidance, which provides recommendations on applicable controls for the different cloud computing services, it can also be used with any combination of cloud computing service capability type, service category, deployment model, or target customer profile. Additionally, our integrated ontology can be used from the perspective of both the cloud computing service customer and the cloud computing service provider.

Therefore, to complete and extend the model for use in the evaluation of cloud computing service providers as proposed by Wollersheim & Krcmar (2014) and the quality model for the evaluation of SaaS as proposed by Wen & Dong (2013), we recommend the application of our integrated ontology to support the definition of quality measures for the evaluation of cloud computing services information security controls.

Through our literature review, we also observed that quality models for cloud computing services have been proposed that do not make use of ISO/IEC 25000 quality models and quality measures. For example, Garg et al. (2011) proposed a quality model for IaaS based on SMI KPIs. Such KPIs include both qualitative and quantitative types. The objective is to support cloud computing service customers to determine the best cloud computing services to fulfill their QoS requirements. The proposed model includes the attribute “security and privacy”. Garg et al. (2011) indicated that this attribute is multi-dimensional in nature while also including many attributes, such as privacy, data loss, and integrity. Although their study defined quantifiable KPIs and proposed metrics for attributes, no KPIs or metrics were proposed for the attribute “security and privacy”.

On the one hand, attributes such as “security”, “privacy”, and “integrity” can be defined by ISO/IEC 25010, quality characteristics, quality sub-characteristics, and their associated ISO/IEC 25023 quality measures as quality requirements. Indeed, the attribute “security” can be specified by the ISO/IEC 25010 quality characteristic “security” as well as its associated ISO/IEC 25023 quality measures as a quality requirement. The attribute “integrity” can also be specified by the ISO/IEC 25010 quality sub-characteristic “integrity” as well as its associated ISO/IEC 25023 quality measures as a quality requirement. The attribute “privacy” can be defined by the ISO/IEC 25010 quality characteristic “freedom from risk” as a “quality in use requirement” where risks can arise from “…inadequate operational safety or protection of security or privacy” and from which product and data quality requirements can be derived from. The attribute “data loss” is not covered by ISO/IEC 25000 quality models and quality measures.

On the other hand, as previously discussed, our results indicated that the application of ISO/IEC 25010 is insufficient to mitigate the CSA top threats to cloud computing. However, our integrated ontology extends the applicability of the ISO/IEC 25030 quality requirements definition framework to information security in cloud computing services through the integration of the ISO/IEC/IEEE 15288, NIST SP 800-160, ISO/IEC 25030, and ISO/IEC 27005 frameworks.

For instance, through the application of our integrated ontology, the attribute “privacy” can be defined by the ISO/IEC 27018 control objectives and by the controls that implement them. The measures used to evaluate the performance and effectiveness of the said controls can be defined using guidance from ISO/IEC 27004. Regarding the attribute “data loss”, it can be defined by the ISO/IEC 27002 control objective “To protect against loss of data”, which is implemented by the control information backup. Again, the measures used to evaluate the performance and effectiveness of the said controls can be defined using guidance from ISO/IEC 27004.

Additionally, as previously discussed, the application of our integrated ontology supports the definition of quality measures for the evaluation of cloud computing services information security controls. More specifically, the application of our integrated ontology demonstrated that the control objectives can be implemented by controls to fulfill quality requirements specified by ISO/IEC 25000 quality characteristics and quality sub-characteristics as well as their associated quality measures. It also demonstrated that in some cases, these quality requirements are achieved by functional requirements that specify the security functions or are implemented by the security architecture extracted from ISO/IEC 25000 quality measures. Finally, it has further demonstrated that in some cases, these quality requirements are verified by the security-focused verification method extracted from ISO/IEC 25000 quality measures.

For instance, the control objective “To protect against loss of data” is implemented by the control information backup to fulfill the quality requirements specified by the ISO/IEC 25010 quality characteristic reliability, quality sub-characteristic recoverability, and the associated quality measure backup data completeness, as well as by the ISO/IEC 25012 quality characteristic recoverability and associated quality measures, namely, recoverability and architecture recoverability. These quality requirements are achieved by functional requirements that specify the security functions backup of data and back-up/restore procedures, both extracted from the said ISO/IEC 25000 quality measures.

Furthermore, based on its description, which stipulates that “Adequate backup facilities should be provided to ensure that all essential information and software can be recovered following a disaster or media failure”, the control information backup fulfills quality requirements specified by the ISO/IEC 25010 quality characteristic security and the quality sub-characteristic integrity and the associated quality measure, namely, data integrity, as well as by the ISO/IEC 25012 quality characteristics availability and recoverability and their associated quality measures, namely, data availability ratio and data recoverability ratio. However, in such cases, no security function, security architecture, or security-focused verification method can be extracted from the quality measures.

These results are consistent with the study by Garg et al. (2011), which stated that the attribute “security and privacy” is multi-dimensional in nature and includes many attributes. The application of our integrated ontology showed that such dimensions include quality requirements specified by ISO/IEC 25000 quality characteristics, quality sub-characteristics, and quality measures as well as ISO/IEC 27000 control objectives, their respective controls, and measures. Consequently, we recommend the application of our integrated ontology to define the “security and privacy” attribute and measures of the quality model for IaaS that was proposed by Garg et al. (2011).

Our analysis is consistent with that of Zheng et al. (2014), who argued that the quality model for IaaS proposed by Garg et al. (2011) does not define important quality dimensions, such as “security”. As a result, Zheng et al. (2014) proposed CLOUDQUAL, a quality model for cloud computing services, which specifies six quality dimensions, including “security” and its associated quality metric, where “security” is defined as “the assurance that cloud services are free from viruses, intrusions, spyware, attacks, and other security vulnerabilities that could put them at risk”. Zheng et al. (2014) also proposed the necessity of evaluating the level of cloud computing service security from an end-user perspective by applying historical information that is publicly available or obtained from a third-party.

On the one hand, Zheng et al. (2014) indicated that the objective of the proposed security dimension and its associated quality metric used for the evaluation of cloud computing services is not to identify security issues nor to enforce security mechanisms for cloud computing services. On the other hand, our integrated ontology showed that controls fulfill security requirements, system security requirements, and quality requirements, while modifying risks, mitigating threats and vulnerabilities, and enforcing security policies. Therefore, in this case, our integrated ontology can be used to extend the scope of the quality model and quality measures proposed by Zheng et al. (2014) to evaluate information security controls for cloud computing services from the perspective of the cloud service customer.

For example, as previously discussed, the application of our integrated ontology showed that the control information backup fulfills the quality requirements specified by the ISO/IEC 25010 quality characteristic security and the quality sub-characteristic integrity and the associated quality measure, namely, data integrity, as well as by the ISO/IEC 25012 quality characteristics availability and recoverability and their associated quality measures, namely, data availability ratio and data recoverability ratio. In this case, implementation guidance for cloud computing services of the control stipulates that the procedures to test backup capabilities should be provided by the cloud service provider to the cloud service customer.

Additionally, it also stipulates that the procedures used to verify the integrity of backup data should be provided by the cloud service provider to the cloud service customer. Consequently, the cloud service customer can verify the said quality requirements.

## 8.4 Potential benefits of this research project for industry

On the one hand, our research showed that the application of ISO/IEC 25000 quality models and quality measures is insufficient in mitigating threats and vulnerabilities to cloud computing services. On the other hand, our research project has shown that the application of our integrated ontology can support the definition of quality requirements related to controls that are applicable in mitigating threats and vulnerabilities to cloud computing services.

Additionally, the integrated ontology has been designed to avoid possible causes of IS failure, namely, lack of a systematic RE process, poor communication between people and lack of a shared understanding of the system being built, and poor management of the RE process (Lyytinen & Hirschheim, 1988; Macaulay, 1996). In the latter case, the integrated ontology has been designed more specifically for the establishment of the traceability of quality requirements and their sources.

It can therefore be concluded that our integrated ontology can support systems and software quality engineers who engage in the quality engineering of cloud-based IS.

## 8.5 Recommendation for future research

Findings from this research project highlight many gaps in our knowledge related to the application of the ISO/IEC 25000 series to information security in cloud computing services. Further research would help to fill these gaps, such as the following:

1. According to Wollersheim & Krcmar (2014), ISO/IEC 25010 ignores service quality. However, what their study does not mention is that the ISO/IEC 25000 series include a service quality model, namely, ISO/IEC 25011, as well as associated service quality

measures, namely, ISO/IEC 25025. While the service quality model has been previously published, the service quality measures were still under development at the time of this research project. Consequently, we were able to include the concepts extracted from the published service quality model in the construction of our proposed integrated ontology. On the other hand, given that the service quality measures have not yet been published, we did not evaluate the applicability of the said quality model and associated quality measures. Once published, however, our research design and integrated ontology can be used to evaluate the extent to which the service quality model and associated service quality measures support the definition of service quality requirements related to information security in cloud computing services.

2. In our analysis, we observed that quality measures are applicable under the condition that there is sufficient information and/or procedures and/or support provided by the cloud service provider to the cloud service customer to verify the quality requirement. This can potentially impact the applicability of ISO/IEC 25040 quality evaluation processes and the developement of a viable quality evaluation. Research is therefore necessary to evaluate the applicability of these processes to the evaluation of quality requirements related to information security in cloud computing services.
3. Information security in cloud computing services can be a component of the service contract negotiation between the cloud service customer and the cloud service provider where service level objectives (SLOs) and service qualitative objectives (SQOs) are defined and agreed upon. The ISO/IEC 19086 series provides a framework that aims to support the creation of such an agreement, namely, cloud Service Level Agreements (SLAs). Our research design can be used to extend our integrated ontology with the said framework, and consequenlty, extend the applicability of the ISO/IEC 25030 quality requirements framework to the definition of service quality requirements that can be used for the definititon of SLOs and SQOs.
4. The evaluation of the entire process resulting from the integration of the ISO/IEC/IEEE 15288, NIST SP 800-160, ISO/IEC 25030, ISO/IEC 27005, and ISO/IEC 19086 frameworks. Research can evaluate the extent to which the resulting process is

applicable to the industry by systems and software quality engineers for the quality engineering of cloud-based IS.

## 8.6 Conclusions

This research project aimed to extend the applicability the ISO/IEC 25030 quality requirements framework to information security in cloud computing services. Our integrated ontology, which extends and documents the extension of this framework, was used to evaluate the applicability of ISO/IEC 25010 and ISO/IEC 25012 quality models and associated ISO/IEC 25022, ISO/IEC 25023, and ISO/IEC 25024 quality measures to information security in cloud computing services. Our evaluation revealed that the application of the said quality models and quality measures is insufficient to mitigate threats and vulnerabilities to cloud computing services. Finally, the application of our integrated ontology revealed that it can be used to define quality requirements related to information security in cloud computing services.

To the best of our knowledge, our integrated ontology is the first artifact build to extend and document the applicability of the ISO/IEC 25030 quality requirements framework to information security in cloud computing services for the users of the framework. The addition of our integrated ontology to the systems and software quality engineering knowledgebase will help support systems and software quality engineers with the quality engineering of cloud-based IS as well as the ISO/IEC JTC 1/SC 7/WG 6 on systems and software product quality along with the further development of ISO/IEC 25000 quality models and quality measures.

# ANNEX I

# INTEGRATED ONTOLOGY

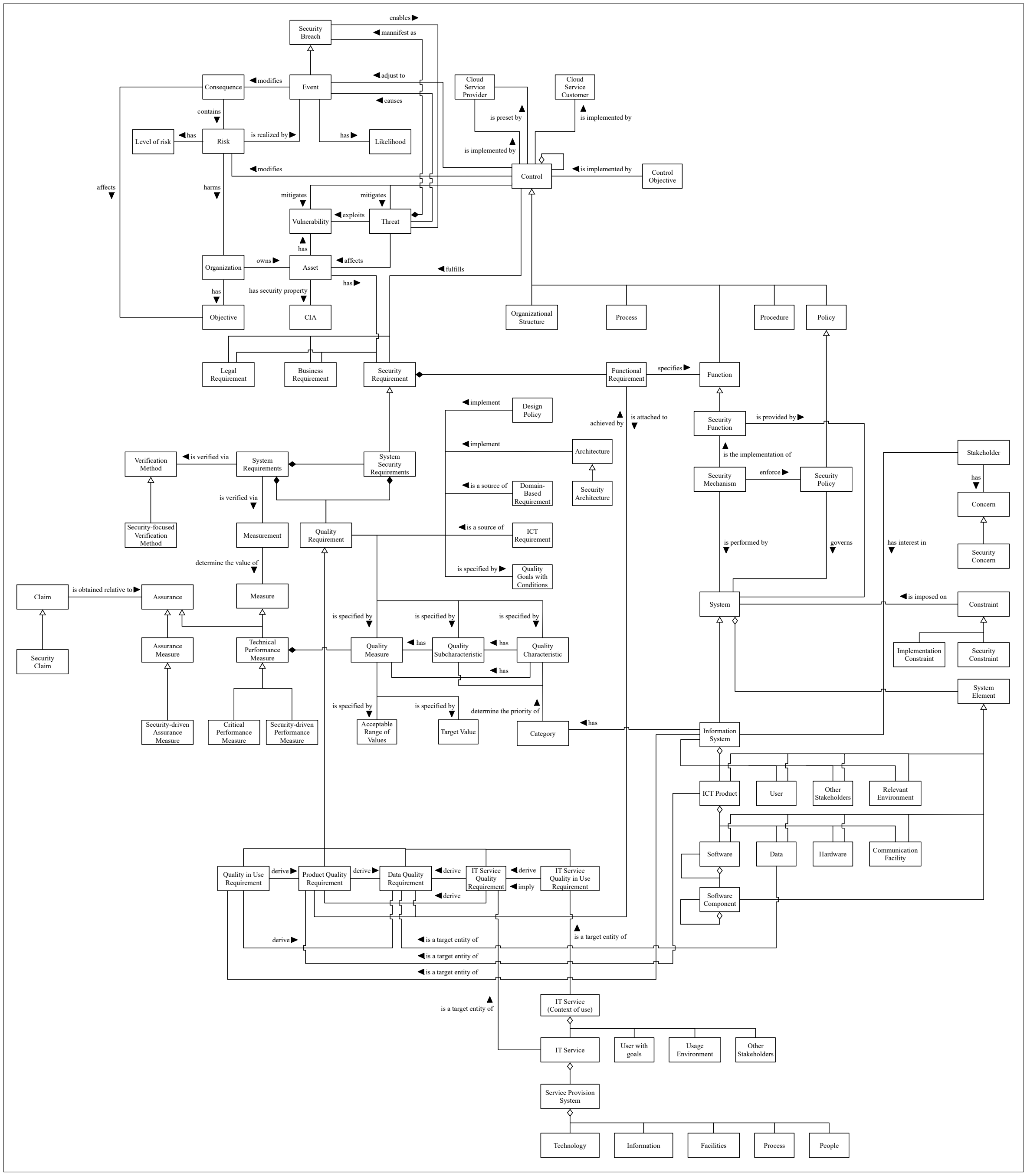


Figure-A I-1 Integrated ontology

# ANNEX II

## COMPETENCY QUESTION INSTANTIATIONS

### Security function: Digital signature

**Digital signature** is the security function extracted from the quality measure **digital signature usage**. Table-A II-1 provides the security function extracted from the quality measure.

Table-A II-1 Security function extracted from the quality measure **digital signature usage**

| **Extracted from ISO/IEC 25023 quality measures** | **ISO/IEC 25023 quality measures** | **ISO/IEC 25010 quality model** | |
|---|---|---|---|
| **Function** | **Quality measure** | **Quality sub-characteristic** | **Quality characteristic** |
| Digital signature | Digital signature usage | Non-repudiation | Security |

### Step 1: CQ1 instantiation

We found the control description excerpt provided in Table-A II-2 to instantiate our integrated ontology with the **security function** digital signature usage to answer CQ1.

Table-A II-2 Cryptography controls description excerpt from ISO/IEC 27002 (ISO/IEC, 2013)

10.1 Cryptography controls

Objective: To ensure proper and effective use of cryptography to protect the confidentiality, authenticity and/or integrity of information.

10.1.1 Policy on the use of cryptographic controls

Control
A policy on the use of cryptographic controls for protection of information should be developed and implemented.

Implementation guidance
When developing a cryptographic policy the following should be considered:

a) the management approach towards the use of cryptographic controls across the organization, including the general principles under which business information should be protected; […]

Cryptographic controls can be used to achieve different information security objectives, e.g.:

[…] b) integrity/authenticity: using digital signatures or message authentication codes to verify the authenticity or integrity of stored or transmitted sensitive or critical information; […]

c) non-repudiation: using cryptographic techniques to provide evidence of the occurrence or non-occurrence of an event or action; […]

According to its description, the **control objective** is "To ensure the proper and effective use of cryptography to protect the confidentiality, authenticity, and/or integrity of information…" and *IsImplementedBy* the **control** policy on the use of cryptographic controls to *Fulfill*:

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** security, **quality sub-characteristic** non-repudiation, and **quality measure** digital signature usage defined as the measurement function of values of the quality measure elements:
    a. Number of events that ensure non-repudiation using digital signature;
    b. Number of events requiring non-repudiation using a digital signature.

The **quality requirement** *IsAcheivedBy* a **functional requirement** that *Specifies* the **security function** digital signature.

### Step 2: CQ2 instantiation

Based on their descriptions, we found the **controls** provided in Table-A II-3 to instantiate our integrated ontology to answer CQ2 where the **control objective** *IsImplementedBy* the **control** policy on the use of cryptographic controls associated with the **security function** digital signature.

Table-A II-3 Control description excerpts from ISO/IEC 27002 (ISO/IEC, 2013)

| **Control objective instance** | **Control instance** | **Excerpts from the control instance description** |
|---|---|---|
| To prevent unauthorized access to systems and applications. | 9.4.5 Access control to program source code | "If the program source code is intended to be published, additional controls to help getting assurance on its integrity (e.g., digital signature) should be considered." |
| To ensure that information security is an integral part of information systems across the entire lifecycle. This also includes the requirements for information systems which provide services over public networks. | 14.1.2 Securing application services on public networks | "Application services can make use of secure authentication methods, e.g., using public key cryptography and digital signatures…" |
| | 14.1.3 Protecting application services transactions | "Information security considerations for application service transactions should include… the use of electronic signatures by each of the parties involved in the transaction…" |

### Step 3: CQ3 instantiation

We first determine if the **control** associated to the **security function** digital signature found by CQ1, namely, policy on the use of cryptographic controls, can be used to instantiate our integrated ontology to answer CQ3. We found the following instances, while the mapping is provided in Table-A II-4 and Table-A II-5.

1. The **control** policy on the use of cryptographic controls *Mitigates* the **threats**;

a. Data breaches;
b. Misconfiguration and inadequate change control;
c. Lack of cloud security architecture and strategy;
d. Insufficient identity, credential, access, and key management;
e. Insider threats;
f. Insecure interfaces and APIs;
g. Weak control plane;
h. Metastructure and applistructure failures;
i. Limited cloud usage visibility;
j. Abuse and nefarious use of cloud services.

Then, we determine if the **controls** associated to the **security function** digital signature found by CQ2, namely, access control to program source code, securing application services on public networks, and protecting application services transactions, can be used to instantiate our integrated ontology to answer CQ3. We found the following instances, while the mapping to the CSA top threats to cloud computing is provided in Tables II.6 and II.7.

1. The **control** access control to program source code *Mitigates* the **threats**;
   a. Misconfiguration and Inadequate Change Control;
   b. Insufficient Identity, Credential, Access, and Key Management;
   c. Account Hijacking;
   d. Insider Threat;
   e. Insecure Interfaces and APIs;
   f. Metastructure and Applistructure Failures;

2. The **control** securing application services on public networks *Mitigates* the **threats**;
   a. Data Breaches;
   b. Misconfiguration and Inadequate Change Control;
   c. Lack of Cloud Security Architecture and Strategy;
   d. Weak Control Plane;

e. Metastructure and Applistructure Failures;
f. Abuse and Nefarious Use of Cloud Services;

3. The **control** protecting application services transactions *Mitigates* the **threats**;
a. Data Breaches;
b. Misconfiguration and Inadequate Change Control;
c. Insufficient Identity, Credential, Access, and Key Management;
d. Insider Threat;
e. Metastructure and Applistructure Failures;
f. Limited Cloud Usage Visibility;
g. Abuse and Nefarious Use of Cloud Services.

**Step 4: Control and quality measure implementation guidance**

We analyze the available implementation guidance for cloud computing services of the **control** policy on the use of cryptographic control provided in section 6.3.1 to determine whether there is evidence to suggest that sufficient information and/or procedures and/or support is provided by the cloud service provider to the cloud service customer to verify the **quality requirement** *SpecifiedBy* the **quality measure** digital signature usage.

In the case where the cryptographic capability, namely, the **security function** digital signature, *IsImplementedBy* the **cloud service provider** or the **cloud service customer**, there is evidence to suggest that the information that should be provided by the cloud service provider to the cloud service customer is sufficient to verify the **quality requirement** *SpecifiedBy* the **quality measure** digital signature usage.

**Step 5: Additional instantiation**

No additional instantiations were found.

Table-A II-4 Digital signature applicability: CQ1 and CQ3 instantiations

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | **X** | | | | | | |
| | Misconfiguration and Inadequate Change Control | **X** | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | **X** | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | **X** | | | | | | |
| | Account Hijacking | | | | | | | |
| | Insider Threat | | | | **X** | | | |
| | Insecure Interfaces and APIs | | | | | | | |
| | Weak Control Plane | | | | | | | |
| | Metastructure and Applistructure Failures | | | | | **Quality characteristic** | | Security |
| | Limited Cloud Usage Visibility | | | | | **Quality sub-characteristic** | | Non-repudiation |
| | Abuse and Nefarious Use of Cloud Services | | | | | **Quality measure** | | Digital signature usage |
| | | Storage and Access | Wireless Security | Supply Chain Agreements | Third Party Audits | **Controls** / **Extracted function** | | Digital signature |
| | | EKM-04 | IVS-12 | STA-05 | STA-09 | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 10.1.1 | Policy on the use of cryptographic controls | **CQ1** |

Table-A II-5 Digital signature applicability: CQ1 and CQ3 instantiations

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | **X** | **X** | **X** | **X** | | | |
| | Misconfiguration and Inadequate Change Control | | **X** | | **X** | | | |
| | Lack of Cloud Security Architecture and Strategy | | **X** | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | **X** | **X** | | | |
| | Account Hijacking | | | | | | | |
| | Insider Threat | | | | **X** | | | |
| | Insecure Interfaces and APIs | **X** | **X** | | | | | |
| | Weak Control Plane | **X** | **X** | | | | | |
| | Metastructure and Applistructure Failures | **X** | **X** | | **X** | **Quality characteristic** | | Security |
| | Limited Cloud Usage Visibility | | | | **X** | **Quality sub-characteristic** | | Non-repudiation |
| | Abuse and Nefarious Use of Cloud Services | | | | **X** | **Quality measure** | | Digital signature usage |
| | | Data Integrity | Data Integrity/ Security | Entitlement | Sensitive Data Protection | **Controls** / **Extracted function** | | Digital signature |
| | | AIS-03 | AIS-04 | EKM-01 | EKM-03 | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 10.1.1 | Policy on the use of cryptographic controls | **CQ1** |

Table-A II-6 Digital signature applicability: CQ2 and CQ3 instantiations

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | |
| | Misconfiguration and Inadequate Change Control | | | **X** | **X** | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | **X** | | | | | | |
| | Account Hijacking | **X** | | | | | | |
| | Insider Threat | **X** | | | | | | |
| | Insecure Interfaces and APIs | **X** | | | | | | |
| | Weak Control Plane | | | | | | | |
| | Metastructure and Applistructure Failures | **X** | **X** | | | **Quality characteristic** | | Security |
| | Limited Cloud Usage Visibility | | | | | **Quality sub-characteristic** | | Non-repudiation |
| | Abuse and Nefarious Use of Cloud Services | | | **X** | | **Quality measure** | | Digital signature usage |
| | | Trusted Sources | New Development/ Acquisition | Outsourced Development | Quality Testing | **Controls** | **Extracted function** | Digital signature |
| | | IAM-08 | CCC-01 | CCC-02 | CCC-03 | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 9.4.5 | Access control to program source code | **10.1.1 (CQ2)** |
| | | | | | | 14.1.2 | Securing application services on public networks | **10.1.1 (CQ2)** |
| | | | | | | 14.1.3 | Protecting application services transactions | **10.1.1 (CQ2)** |

Table-A II-7 Digital signature applicability: CQ2 and CQ3 instantiations

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | **X** | | **X** | | | | |
| | Misconfiguration and Inadequate Change Control | **X** | **X** | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | **X** | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | **X** | | | **X** | | | |
| | Account Hijacking | | | | | | | |
| | Insider Threat | **X** | | | | | | |
| | Insecure Interfaces and APIs | | | | | | | |
| | Weak Control Plane | | **X** | | | | | |
| | Metastructure and Applistructure Failures | **X** | | **X** | | **Quality characteristic** | | Security |
| | Limited Cloud Usage Visibility | **X** | | | | **Quality sub-characteristic** | | Non-repudiation |
| | Abuse and Nefarious Use of Cloud Services | **X** | **X** | | | **Quality measure** | | Digital signature usage |
| | | Sensitive Data Protection | Network Security | Ecommerce Transaction | Source Code Access Restriction | **Controls** | **Extracted function** | Digital signature |
| | | EKM-03 | IVS-06 | DSI-03 | IAM-06 | | | |
| | | | | | **CQ3** | 9.4.5 | Access control to program source code | **10.1.1 (CQ2)** |
| | | | **CQ3** | **CQ3** | | 14.1.2 | Securing application services on public networks | **10.1.1 (CQ2)** |
| | | **CQ3** | | **CQ3** | | 14.1.3 | Protecting application services transactions | **10.1.1 (CQ2)** |

### Security function: Logging of user access to systems and data

**Logging of user access to system and data** is the security function extracted from the quality measures **user audit trail completeness** and **system log retention**. Table-A II-8 provides the security function extracted from the quality measures.

Table-A II-8 Security function extracted from the quality measures **user audit trail** and **system log retention**

| **Extracted from ISO/IEC 25023 quality measures** | **ISO/IEC 25023 quality measures** | **ISO/IEC 25010 quality model** | |
|---|---|---|---|
| **Function** | **Quality measure** | **Quality sub-characteristic** | **Quality characteristic** |
| Logging of user access to system and data | User audit trail completeness | Accountability | Security |
| | System log retention | | |

### Step 1: CQ1 instantiation

We found the control description excerpt provided in Table-A II-9 to instantiate our integrated ontology with the **security function** logging of user access to system and data to answer CQ1.

Table-A II-9 Logging and monitoring control description excerpt from ISO/IEC 27002 (ISO/IEC, 2013)

| |
|---|
| 12.4 Logging and monitoring<br>Objective<br>To record events and generate evidence.<br>12.4.1 Event logging<br>Control<br>Event logs recording user activities, exceptions, faults, and information security events should be produced, kept, and regularly reviewed. |

According to its description, the **control objective** is “To record events and generate evidence” and *IsImplementedBy* the **control** event logging to *Fulfill*:

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** security, **quality sub-characteristic** accountability, and **quality measure** user audit trail completeness defined as the measurement function of values of the quality measure elements:
    a. Number of accesses recorded in all logs;
    b. Number of accesses to system or data actually tested;
2. A **quality requirement** *SpecifiedBy* the **quality characteristic** security, **quality sub-characteristic** accountability, and **quality measure** system log retention defined as the measurement function of values of the quality measure elements:
    a. Duration for which the system log is actually retained in stable storage;
    b. Retention period specified for keeping the system log in stable storage.

Each **quality requirement** *IsAcheivedBy* a **functional requirement** that *Specifies* the **security function** logging of user access to system and data.

**Step 2: CQ2 instantiation**

Based on their descriptions, the **control** instances provided in Table-A II-10 and Table-A II-11 to instantiate our integrated ontology to answer CQ2 where the **control objective** *IsImplementedBy* the **control** event logging associated with the **security function** logging of user access to system and data.

Table-A II-10 Control description excerpts from ISO/IEC 27002 (ISO/IEC, 2013)

| **Control objective instance** | **Control instance** | **Excerpts from the control instance description** |
|---|---|---|
| To prevent unauthorized access to systems and applications. | 9.4.4 Use of privileged utility programs | "The following guidelines for the use of utility programs that might be capable of overriding system and application controls should be considered: …logging of all use of utility programs…" |
| To ensure proper and effective use of cryptography to protect the confidentiality, authenticity and/or integrity of information. | 10.1.2 Key management | "A key management system should be based on an agreed set of standards, procedures and secure methods for: …logging and auditing of key management related activities." |
| To ensure the protection of information in networks and its supporting information processing facilities. | 13.1.1 Network controls | "Controls should be implemented to ensure the security of information in networks and the protection of connected services from unauthorized access. In particular, the following items should be considered: …appropriate logging and monitoring should be applied to enable recording and detection of actions that may affect, or are relevant to, information security…" |

Table-A II-11 Control description excerpts from ISO/IEC 27018 (ISO/IEC, 2019)

| **Control objective instance** | **Control instance** | **Excerpts from the control instance description** |
|---|---|---|
| To protect PII. | A.11.3 Control and logging of data restoration | "There should be a procedure for, and a log of, data restoration efforts." |

### Step 3: CQ3 instantiation

We first determine if the **control** associated with the **security function** logging of user access to system and data found by the CQ1, namely, event logging, can be used to instantiate our integrated ontology to answer CQ3. We found the following instances, while the mapping is provided in Table-A II-12.

1. The **control** event logging *Mitigates* the **threats**;
    a. Misconfiguration and Inadequate Change Control;
    b. Lack of Cloud Security Architecture and Strategy;
    c. Account Hijacking;
    d. Weak Control Plane;
    e. Abuse and Nefarious Use of Cloud Services.

Then, we determine if the **controls** associated with the **security function** logging of user access to system and data found by CQ2, namely, use of privileged utility programs, key management, network controls, and control and logging of data restoration, can be used to instantiate our integrated ontology to answer CQ3. We found the following instances, while the mapping is provided in Table-A II-13 and Table-A II-14.

1. The **control** use of privileged utility programs *Mitigates* the **threats**;
    a. Insufficient Identity, Credential, Access, and Key Management;
    b. Account Hijacking;
    c. Insecure Interfaces and APIs;
    d. Weak Control Plane;
    e. Metastructure and Applistructure Failures;
    f. Abuse and Nefarious Use of Cloud Services;

2. The **control** key management *Mitigates* the **threats**;
    a. Data Breaches;
    b. Misconfiguration and Inadequate Change Control;
    c. Insufficient Identity, Credential, Access, and Key Management;

3. The **control** network controls *Mitigates* the **threats**;
    a. Data Breaches;
    b. Misconfiguration and Inadequate Change Control;
    c. Lack of Cloud Security Architecture and Strategy;

d. Insufficient Identity, Credential, Access, and Key Management;
e. Insider Threat;
f. Weak Control Plane;
g. Metastructure and Applistructure Failures;
h. Limited Cloud Usage Visibility;
i. Abuse and Nefarious Use of Cloud Services.

The **control** control and logging of data restoration is not mapped to CSA controls in the CSA CCM and, as a result, there is no relationship to the CSA top threats to cloud computing.

**Step 4: Control and quality measure implementation guidance**

We analyze the available implementation guidance for cloud computing services and for the protection of PII in public cloud computing services of the **control** event logging to determine whether there is evidence to suggest that sufficient information and/or procedures and/or support is provided by the cloud service provider to the cloud service customer to verify the **quality requirements** *SpecifiedBy* the **quality measures** user audit trail completeness, encryption usage, and the system log retention.

Implementation guidance of cloud computing services stipulates that "the cloud service customer should define its requirements for event logging and verify that the cloud service meets those requirements."

In the case where the **security function** logging of user access to system and data *IsImplementedBy* the **cloud service provider**, there is evidence to suggest that information and/or procedures and/or support should be provided by the cloud service provider to the cloud service customer to verify the **quality requirements** *SpecifiedBy* the **quality measures** user audit trail completeness and system log retention.

As it pertains to the implementation guidance for the protection of PII in public cloud computing services, it stipulates that "The public cloud PII processor should define criteria regarding if, when, and how log information can be made available to or usable by the cloud service customer. These procedures should be made available to the cloud service customer."

In this case, if the logs are made available to or usable by the cloud service customer, there is evidence to suggest that the procedures to access log records should be provided by the public cloud computing service PII processor to the cloud service customer to verify the **quality requirement** *SpecifiedBy* the **quality measures** user audit trail completeness and system log retention.

### Step 5: Additional instantiation

We found an additional instantiation with a **quality measure** for which no security function, security architecture, or security-focused verification method was extracted.

Based on its description which stipulates that "Event logs recording user activities, exceptions, faults and information security events should be produced, kept and regularly reviewed", the **control** event logging *Fulfills*:

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** traceability and **quality measure** user access traceability defined as the measurement function of values of the quality measure elements:
    a. Number of data items for which user access traceability is expected and realized;
    b. Number of data items for which user traceability is expected.

As previously discussed, we found evidence in the implementation guidance for cloud computing services and for the protection of PII in public cloud computing services of the **control** event logging to suggest that the cloud service customer can verify the **quality requirement** *SpecifiedBy* the **quality measure** user access traceability.

Table-A II-12 Logging of user access to systems and data applicability: CQ1 and CQ3 instantiations

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | | |
| | Misconfiguration and Inadequate Change Control | | **X** | **X** | **X** | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | **X** | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | |
| | Account Hijacking | **X** | | | | | | | |
| | Insider Threat | | | | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | |
| | Weak Control Plane | **X** | | **X** | | | | | |
| | Metastructure and Applistructure Failures | | | | | **Quality characteristic** | | Security | |
| | Limited Cloud Usage Visibility | | | | | **Quality sub-characteristic** | | Accountability | |
| | Abuse and Nefarious Use of Cloud Services | **X** | **X** | **X** | | **Quality measure** | | User audit trail completeness | System log retention |
| | | Audit Logging/Intrusion Detection | Change Detection | Network Security | OS Hardening and Base Control | **Controls** | **Extracted function** | Logging of user access to systems and data | |
| | | IVS-01 | IVS-02 | IVS-06 | IVS-07 | | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 12.4.1 | Event logging | **CQ1** | |

Table-A II-13 Logging of user access to systems and data applicability: CQ2 and CQ3 instantiations

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | **X** | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | **X** | **X** | | | | | | |
| | Account Hijacking | **X** | | | | | | | | |
| | Insider Threat | | | | | | | | | |
| | Insecure Interfaces and APIs | | | **X** | | | | | | |
| | Weak Control Plane | **X** | | | | | | | | |
| | Metastructure and Applistructure Failures | | | **X** | | **X** | **Quality characteristic** | | Security | |
| | Limited Cloud Usage Visibility | | | | | | **Quality sub-characteristic** | | Accountability | |
| | Abuse and Nefarious Use of Cloud Services | **X** | | | | | **Quality measure** | | User audit trail completeness | System log retention |
| | | Audit Logging/ Intrusion Detection | Diagnostic/ Configuration Ports Access | Utility Programs Access | Unauthorized Software Installations | Ecommerce Transactions | **Controls** / **Extracted function** | | Logging of user access to systems and data | |
| | | IVS-01 | IAM-03 | IAM-13 | CCC-04 | DSI-03 | | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | | 9.4.4 | Use of privileged utility programs | **12.4.1 (CQ2)** | |
| | | | | | | | 10.1.2 | Key management | **12.4.1 (CQ2)** | |
| | | | **CQ3** | | | **CQ3** | 13.1.1 | Network controls | **12.4.1 (CQ2)** | |

Table-A II-14 Logging of user access to systems and data applicability: CQ2 and CQ3 instantiations

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | **X** | | **X** | **X** | | | | | |
| | Misconfiguration and Inadequate Change Control | | | **X** | **X** | **X** | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | **X** | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | **X** | | **X** | **X** | | | | | |
| | Account Hijacking | | | | | | | | | |
| | Insider Threat | | | **X** | | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | | |
| | Weak Control Plane | | | | | **X** | | | | |
| | Metastructure and Applistructure Failures | | | **X** | | | **Quality characteristic** | | Security | |
| | Limited Cloud Usage Visibility | | | **X** | | | **Quality sub-characteristic** | | Accountability | |
| | Abuse and Nefarious Use of Cloud Services | | | **X** | | **X** | **Quality measure** | | User audit trail completeness | System log retention |
| | | Entitlement | Key Generation | Sensitive Data Protection | Storage Access | Network Security | **Extracted function**<br>**Controls** | | Logging of user access to systems and data | |
| | | EKM-01 | EKM-02 | EKM-03 | EKM-04 | IVS-06 | | | | |
| | | | | | | | 9.4.4 | Use of privileged utility programs | **12.4.1 (CQ2)** | |
| | | **CQ3** | **CQ3** | | **CQ3** | | 10.1.2 | Key management | **12.4.1 (CQ2)** | |
| | | | | **CQ3** | | **CQ3** | 13.1.1 | Network controls | **12.4.1 (CQ2)** | |

### Security function: Authentication

**Authentication** is the function extracted from the quality measures **authentication mechanism sufficiency** and **authentication rules conformity**. Table-A II-15 provides the security function extracted from the quality measures.

Table-A II-15 Security function extracted from the quality measures **authentication mechanism sufficiency** and **authentication rules conformity**

| **Extracted from ISO/IEC 25023 quality measures** | **ISO/IEC 25023 quality measures** | **ISO/IEC 25010 quality model** | |
|---|---|---|---|
| **Function** | **Quality measure** | **Quality sub-characteristic** | **Quality characteristic** |
| Authentication | Authentication mechanism sufficiency | Authenticity | Security |
| | Authentication rules conformity | | |

### Step 1: CQ1 instantiation

We found the control description excerpt provided in Table-A II-16 to instantiate our integrated ontology with the security function **authentication** to answer CQ1.

Table-A II-16 Access to networks and network services control description excerpt from ISO/IEC 27002 (ISO/IEC, 2013)

> 9.1 Business requirements for access control
>
> Objective: To limit access to information and information processing facilities.
>
> 9.1.2 Access to networks and network services
>
> Control
> Users should only be provided with access to the network and network services that they have been specifically authorized to use.
>
> Implementation guidance
> A policy should be formulated concerning the use of networks and network services. This policy should cover:
>
> e) user authentication requirements for accessing various network services;

According to its description, the **control objective** is "To limit access to information and information processing facilities" and *IsImplementedBy* the **control** access to networks and network services to *Fulfill*:

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** security, **quality sub-characteristic** authenticity, and **quality measure** authentication mechanism sufficiency defined as the measurement function of values of the quality measure elements:
   a. Number of authentication mechanisms provided (e.g., User ID/password or IC card);
   b. Number of authentication mechanisms specified;
2. A **quality requirement** *SpecifiedBy* the **quality characteristic** security, **quality sub-characteristic** authenticity, and **quality measure** authentication rules conformity defined as the measurement function of values of the quality measure elements:
   a. Number of authentication rules implemented;
   b. Number of authentication rules specified.

Each **quality requirement** *IsAcheivedBy* a **functional requirement** that *specifies* the **security function** authentication.

We also found the **control** description excerpt provided in Table-A II-17 to instantiate our integrated ontology with the **security function** authentication to answer CQ1.

Table-A II-17 Secure log-on procedures control description excerpt from ISO/IEC 27002 (ISO/IEC, 2013)

> 9.4 System and application access control
>
> Objective: To prevent unauthorized access to systems and applications.
>
> 9.4.2 Secure log-on procedures
>
> Control
> Where required by the access control policy, access to systems and applications should be controlled by a secure log-on procedure.
>
> Implementation guidance
> A suitable authentication technique should be chosen to substantiate the claimed identity of a user.
>
> Where strong authentication and identity verification is required, authentication methods alternative to passwords, such as cryptographic means, smart cards, tokens, or biometrics means, should be used. […]

According to its description, the **control objective** is "To prevent unauthorized access to systems and applications" and *IsImplementedBy* the **control** secure log-on procedures to *Fulfill*:

1. A **quality requirement** *SpecifiedBy* the **quality characteristic security**, **quality sub-characteristic** authenticity, and **quality measure** authentication mechanism sufficiency defined as the measurement function of values of the quality measure elements:
    a. Number of authentication mechanisms provided (e.g., User ID/password or IC card);

b. Number of authentication mechanisms specified.

The **quality requirement** *IsAcheivedBy* a **functional requirement** that *Specifies* the **security function** authentication.

**Step 2: CQ2 instantiation**

Based on their descriptions, we found the **controls** provided in Table-A II-18 to instantiate our integrated ontology to answer CQ2 where the **control objective** *IsImplementedBy* the **control** access to networks and network services and/or secure log-on procedures, both being associated with the **security function** authentication.

Table-A II-18 Control description excerpts from ISO/IEC 27002 (ISO/IEC, 2013)

| **Control objective** | **Control** | **Excerpts from the control description** |
|---|---|---|
| To prevent unauthorized access to systems and applications. | 9.4.4 Use of privileged utility programs | "The following guidelines for the use of utility programs that might be capable of overriding system and application controls should be considered: …use of identification, authentication and authorization procedures for utility programs…" |
| To ensure proper and effective use of cryptography to protect the confidentiality, authenticity and/or integrity of information. | 10.1.2 Key management | "In addition to securely managing secret and private keys, the authenticity of public keys should also be considered. This authentication process can be done using public key certificates, which are normally issued by a certification authority, which should be a recognized organization with suitable controls and procedures in place to provide the required degree of trust." |
| To prevent unauthorized physical access, damage and interference to the organization's information and information processing facilities. | 11.1.2 Physical entry controls | "…access to areas where confidential information is processed or stored should be restricted to authorized individuals only by implementing appropriate access controls, e.g., by implementing a two-factor authentication mechanism such as an access card and secret PIN…" |
| To prevent loss, damage, theft or compromise of assets and interruption to the organization's operations. | 11.2.9 Clear desk and clear screen policy | "computers and terminals should be left logged off or protected with a screen and keyboard locking mechanism controlled by a password, token or similar user authentication mechanism when unattended and should be protected by key locks, passwords or other controls when not in use…" |

| Control objective | Control | Excerpts from the control description |
|---|---|---|
| To ensure the protection of information in networks and its supporting information processing facilities. | 13.1.2 Security of network services | “Security mechanisms, service levels and management requirements of all network services should be identified and included in network services agreements, whether these services are provided in-house or outsourced.”<br>“Security features of network services could be… technology applied for security of network services, such as authentication, encryption and network connection controls…” |
| | 13.1.3 Segregation in networks | “The authentication, encryption and user level network access control technologies of modern standards based wireless networks may be sufficient for direct connection to the organization’s internal network when properly implemented. Stronger levels of authentication controlling access from publicly accessible networks. |
| To maintain the security of information transferred within an organization and with any external entity. | 13.2.3 Electronic messaging | “Information security considerations for electronic messaging should include the following: …stronger levels of authentication controlling access from publicly accessible networks.” |
| To ensure that information security is an integral part of information systems across the entire lifecycle. This also includes the requirements for information systems which provide services over public networks. | 14.1.2 Securing application services on public networks | “Information security considerations for application services passing over public networks should include the following: …the level of confidence each party requires in each other’s claimed identity, e.g. through authentication…” |
| | 14.1.3 Protecting application services transactions | “Information security considerations for application service transactions should include the following: …all aspects of the transaction, i.e., ensuring that: …user’s secret authentication information of all parties are valid and verified…” |

**Step 3: CQ3 instantiation**

We first determine if the **controls** associated with the **security function** authentication found by CQ1, namely, access to networks and network services and secure log-on procedures, can be used to instantiate our integrated ontology to answer CQ3. We found the following instances and the mapping to the CSA top threats to cloud computing is provided in Table-A II-19 and Table-A II-20.

1. The **control** access to networks and network services *Mitigates* the **threats**;
    a. Misconfiguration and Inadequate Change Control;
    b. Lack of Cloud Security Architecture and Strategy;
    c. Insufficient Identity, Credential, Access, and Key Management;
    d. Account Hijacking;
    e. Insider Threat;
    f. Insecure Interfaces and APIs;
    g. Weak Control Plane;
    h. Metastructure and Applistructure Failures;
    i. Abuse and Nefarious Use of Cloud Services;

2. The **control** secure log-on procedures *Mitigates* the **threats**;
    a. Data Breaches;
    b. Misconfiguration and Inadequate Change Control;
    c. Insufficient Identity, Credential, Access, and Key Management;
    d. Account Hijacking;
    e. Insider Threat;
    f. Insecure Interfaces and APIs;
    g. Metastructure and Applistructure Failures;
    h. Abuse and Nefarious Use of Cloud Services.

Then, we determine if the **controls** associated with the **security function** authentication found by CQ2, namely, use of privileged utility programs, key management, physical entry controls, clear desk and clear screen policy, security of network services, segregation in networks, electronic messaging, securing application services on public networks, and protecting application services transactions, can be used to instantiate our integrated ontology to answer CQ3. We found the following instances, while the mapping to the CSA top threats to cloud computing is provided in Table-A II-21, Table-A II-22, Table-A II-23, and Table-A II-24.

1. The **control** use of privileged utility programs *Mitigates* the **threats**;
    a. Insufficient Identity, Credential, Access, and Key Management;
    b. Account Hijacking;
    c. Insecure Interfaces and APIs;
    d. Weak Control Plane;
    e. Metastructure and Applistructure Failures;
    f. Abuse and Nefarious Use of Cloud Services;

2. The **control** key management *Mitigates* the **threats**;
    a. Data Breaches;
    b. Misconfiguration and Inadequate Change Control;
    c. Insufficient Identity, Credential, Access, and Key Management;
    d. Account Hijacking;
    e. Metastructure and Applistructure Failures;

3. The **control** security of network services *Mitigates* the **threats**;
    a. Data Breaches;
    b. Misconfiguration and Inadequate Change Control;
    c. Lack of Cloud Security Architecture and Strategy;
    d. Insider Threat;
    e. Insecure Interfaces and APIs;
    f. Metastructure and Applistructure Failures;

g. Abuse and Nefarious Use of Cloud Services;

4. The **control** segregation in networks *Mitigates* the **threats**;

a. Misconfiguration and Inadequate Change Control;
b. Lack of Cloud Security Architecture and Strategy;
c. Insider Threat;
d. Weak Control Plane;
e. Metastructure and Applistructure Failures;
f. Abuse and Nefarious Use of Cloud Services;

5. The **control** electronic messaging *Mitigates* the **threats**;

a. Data Breaches;
b. Misconfiguration and Inadequate Change Control;
c. Insufficient Identity, Credential, Access, and Key Management;
d. Insider Threat;
e. Metastructure and Applistructure Failures;
f. Limited Cloud Usage Visibility;
g. Abuse and Nefarious Use of Cloud Services;

6. The **control** securing application services on public networks *Mitigates* the **threats**;

a. Data Breaches;
b. Misconfiguration and Inadequate Change Control;
c. Lack of Cloud Security Architecture and Strategy;
d. Insufficient Identity, Credential, Access, and Key Management;
e. Insider Threat;
f. Weak Control Plane;
g. Metastructure and Applistructure Failures;
h. Limited Cloud Usage Visibility;
i. Abuse and Nefarious Use of Cloud Services;

7. The **control** protecting application services transactions *Mitigates* the **threats**;
    a. Data Breaches;
    b. Misconfiguration and Inadequate Change Control;
    c. Insufficient Identity, Credential, Access, and Key Management;
    d. Insider Threat;
    e. Metastructure and Applistructure Failures;
    f. Limited Cloud Usage Visibility;
    g. Abuse and Nefarious Use of Cloud Services.

However, in the case of the **controls** physical entry controls and clear desk and clear screen policy, although they are mapped to the CSA control's controlled access point, policy, and workspace, respectively, in the CSA CCM, the said CSA controls are not mapped to the CSA top threats to cloud computing.

**Step 4: Control and quality measure implementation guidance**

We analyze the available implementation guidance for cloud computing services and for the protection of PII in public cloud computing services of the **controls** access to networks and network services and secure log-on procedures to determine whether there is evidence to suggest that sufficient information and/or procedures and/or support should be provided by the cloud service provider to the cloud service customer to verify the **quality requirement** instances *SpecifiedBy* the **quality measures** authentication mechanism sufficiency and authentication rules conformity.

The implementation guidance for cloud computing services of the **control** access to networks and network services stipulates that "the cloud service customer's access control policy for the use of network services should specify requirements for user access to each separate cloud service that is used."

In the case where the **security function** authentication *IsImplementedBy* the **cloud service provider**, there is no evidence to suggest that information and/or procedures and/or support should be provided by the cloud service provider to the cloud service customer to verify the **quality requirements** *SpecifiedBy* the **quality measures** authentication mechanism sufficiency and authentication rules conformity.

Therefore, we consider the **quality measures** authentication mechanism sufficiency and authentication rules conformity applicable under the condition that there is sufficient information and/or procedures and/or support provided by the cloud service provider to the cloud service customer to verify the **quality requirements** *SpecifiedBy* the said **quality measures**.

The **control** secure log-on procedures do not include implementation guidance for cloud computing services. Therefore, we consider the **quality measures** authentication mechanism sufficiency and authentication rules conformity applicable under the condition that there is sufficient information and/or procedures and/or support provided by the cloud service provider to the cloud service customer to verify that **quality requiremen**ts are *SpecifiedBy* the said **quality measures**.

However, the **control** secure log-on procedures include implementation guidance for the protection of PII in public cloud computing services. It stipulates that "Where required, the public cloud PII processor should provide secure log-on procedures for any accounts requested by the cloud service customer for cloud service users under its control."

In this case, the **security function** authentication *IsImplementedBy* the **cloud service provider**. However, there is no evidence to suggest that information and/or procedures and/or support should be provided by the cloud service provider to the cloud service customer to verify the **quality requirements** *SpecifiedBy* the **quality measures** authentication mechanism sufficiency and authentication rules conformity.

Therefore, we consider the **quality measures** authentication mechanism sufficiency and authentication rules conformity applicable under the condition that there is sufficient information and/or procedures and/or support provided by the cloud service provider to the cloud service customer to verify the **quality requirements** *SpecifiedBy* the said **quality measures**.

### Step 5: Additional instantiation

No additional instantiations were found.

Table-A II-19 Authentication applicability: CQ1 and CQ3 instantiations

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | **X** | **X** | **X** | | | | |
| | Account Hijacking | | | | | | | |
| | Insider Threat | **X** | | | | | | |
| | Insecure Interfaces and APIs | **X** | **X** | **X** | | | | |
| | Weak Control Plane | | | | | | | |
| | Metastructure and Applistructure Failures | **X** | **X** | **X** | **Quality characteristic** | | Security | |
| | Limited Cloud Usage Visibility | | | | **Quality sub-characteristic** | | Authenticity | |
| | Abuse and Nefarious Use of Cloud Services | **X** | | **X** | **Quality measure** | | Authentication mechanism sufficiency | Authentication rules conformity |
| | | User Access Authorization | Utility Program Access | User ID Credentials | **Controls** | **Extracted function** | Authentication | |
| | | IAM-09 | IAM-13 | IAM-12 | | | | |
| | | **CQ3** | **CQ3** | | 9.1.2 | Access to networks and network services | **CQ1** | |
| | | | | **CQ3** | 9.4.2 | Secure log-on procedures | **CQ1** | |

Table-A II-20 Authentication applicability: CQ1 and CQ3 instantiations

| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches |  | X |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | Misconfiguration and Inadequate Change Control | X | X | X |  |  |  |  |  |
|  | Lack of Cloud Security Architecture and Strategy | X |  | X |  |  |  |  |  |
|  | Insufficient Identity, Credential, Access, and Key Management |  |  | X | X |  |  |  |  |
|  | Account Hijacking |  |  | X | X |  |  |  |  |
|  | Insider Threat |  |  |  | X |  |  |  |  |
|  | Insecure Interfaces and APIs |  | X |  | X |  |  |  |  |
|  | Weak Control Plane | X |  |  |  |  |  |  |  |
|  | Metastructure and Applistructure Failures |  | X | X | X | **Quality characteristic** |  | Security |  |
|  | Limited Cloud Usage Visibility |  |  |  |  | **Quality sub-characteristic** |  | Authenticity |  |
|  | Abuse and Nefarious Use of Cloud Services | X |  | X |  | **Quality measure** |  | Authentication mechanism sufficiency | Authentication rules conformity |
|  |  | Network Security | Application Security | Credential Lifecycle/Provision Management | Trusted Sources | **Controls** | **Extracted function** | Authentication |  |
|  |  | IVS-06 | AIS-01 | IAM-02 | IAM-08 |  |  |  |  |
|  |  | **CQ3** |  | **CQ3** |  | 9.1.2 | Access to networks and network services | **CQ1** |  |
|  |  |  | **CQ3** |  | **CQ3** | 9.4.2 | Secure log-on procedures | **CQ1** |  |

Table-A II-21 Authentication applicability: CQ2 and CQ3 instantiations

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | **X** | | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | **X** | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | |
| | Account Hijacking | | | | | | | | |
| | Insider Threat | | **X** | | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | |
| | Weak Control Plane | | | | | | | | |
| | Metastructure and Applistructure Failures | **X** | | **X** | | **Quality characteristic** | | Security | |
| | Limited Cloud Usage Visibility | | | | | **Quality sub-characteristic** | | Authenticity | |
| | Abuse and Nefarious Use of Cloud Services | | | | | **Quality measure** | | Authentication mechanism sufficiency | Authentication rules conformity |
| | | Network/ Infrastructure Services | Third-Party Audits | Ecommerce Transactions | Workspace | **Controls** / **Extracted function** | | Authentication | |
| | | STA-03 | STA-09 | DSI-03 | HRS-11 | | | | |
| | | | | | | 9.4.4 | Use of privileged utility programs | **CQ2** | |
| | | | | | | 10.1.2 | Key management | **CQ2** | |
| | | | | | | 11.1.2 | Physical entry controls | **CQ2** | |
| | | | | | **CQ3** | 11.2.9 | Clear desk and clear screen policy | **CQ2** | |
| | | **CQ3** | **CQ3** | **CQ3** | | 13.1.2 | Security of network services | **CQ2** | |
| | | | | | | 13.1.3 | Segregation in networks | **CQ2** | |
| | | | | | | 13.2.3 | Electronic messaging | **CQ2** | |
| | | | | **CQ3** | | 14.1.2 | Securing application services on public networks | **CQ2** | |
| | | | | **CQ3** | | 14.1.3 | Protecting application services transactions | **CQ2** | |

Table-A II-22 Authentication applicability: CQ2 and CQ3 instantiations

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | **X** | **X** | | | | | |
| | Account Hijacking | **X** | | | | | | | |
| | Insider Threat | | | | | | | | |
| | Insecure Interfaces and APIs | | | **X** | | | | | |
| | Weak Control Plane | **X** | | | | | | | |
| | Metastructure and Applistructure Failures | | | **X** | | **Quality characteristic** | | Security | |
| | Limited Cloud Usage Visibility | | | | | **Quality sub-characteristic** | | Authenticity | |
| | Abuse and Nefarious Use of Cloud Services | **X** | | | | **Quality measure** | | Authentication mechanism sufficiency | Authentication rules conformity |
| | | Audit Logging/ Intrusion Detection | Diagnostic/ Configuration Ports Access | Utility Programs Access | Unauthorized Software Installations | **Controls** | **Extracted function** | Authentication | |
| | | IVS-01 | IAM-03 | IAM-13 | CCC-04 | | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 9.4.4 | Use of privileged utility programs | **CQ2** | |
| | | | | | | 10.1.2 | Key management | **CQ2** | |
| | | | | | | 11.1.2 | Physical entry controls | **CQ2** | |
| | | | | | | 11.2.9 | Clear desk and clear screen policy | **CQ2** | |
| | | | | | | 13.1.2 | Security of network services | **CQ2** | |
| | | | | | | 13.1.3 | Segregation in networks | **CQ2** | |
| | | | | | | 13.2.3 | Electronic messaging | **CQ2** | |
| | | | | | | 14.1.2 | Securing application services on public networks | **CQ2** | |
| | | | | | | 14.1.3 | Protecting application services transactions | **CQ2** | |

Table-A II-23 Authentication applicability: CQ2 and CQ3 instantiations

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | **X** | | | | | | | |
| | Misconfiguration and Inadequate Change Control | **X** | **X** | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | **X** | **X** | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | **X** | | | | | | | |
| | Account Hijacking | | | | | | | | |
| | Insider Threat | | | **X** | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | |
| | Weak Control Plane | | **X** | **X** | | | | | |
| | Metastructure and Applistructure Failures | | | **X** | | **Quality characteristic** | | Security | |
| | Limited Cloud Usage Visibility | | | | | **Quality sub-characteristic** | | Authenticity | |
| | Abuse and Nefarious Use of Cloud Services | | **X** | | | **Quality measure** | | Authentication mechanism sufficiency | Authentication rules conformity |
| | | Storage and Access | Network Security | Segmentation | Wireless Security | **Controls** | **Extracted function** | Authentication | |
| | | EKM-04 | IVS-06 | IVS-09 | IVS-12 | | | | |
| | | | | | | 9.4.4 | Use of privileged utility programs | **CQ2** | |
| | | **CQ3** | | | **CQ3** | 10.1.2 | Key management | **CQ2** | |
| | | | | | | 11.1.2 | Physical entry controls | **CQ2** | |
| | | | | | | 11.2.9 | Clear desk and clear screen policy | **CQ2** | |
| | | | **CQ3** | | **CQ3** | 13.1.2 | Security of network services | **CQ2** | |
| | | | **CQ3** | **CQ3** | **CQ3** | 13.1.3 | Segregation in networks | **CQ2** | |
| | | | | | | 13.2.3 | Electronic messaging | **CQ2** | |
| | | | **CQ3** | | | 14.1.2 | Securing application services on public networks | **CQ2** | |
| | | | | | | 14.1.3 | Protecting application services transactions | **CQ2** | |

Table-A II-24 Authentication applicability: CQ2 and CQ3 instantiations

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | **X** | **X** | **X** | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | **X** | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | **X** | **X** | **X** | | | | |
| | Account Hijacking | | | | | | | | | |
| | Insider Threat | | | | **X** | **X** | | | | |
| | Insecure Interfaces and APIs | | | | | | | | | |
| | Weak Control Plane | | | | | | | | | |
| | Metastructure and Applistructure Failures | | | | **X** | **X** | **Quality characteristic** | | Security | |
| | Limited Cloud Usage Visibility | | | | | **X** | **Quality sub-characteristic** | | Authenticity | |
| | Abuse and Nefarious Use of Cloud Services | | | | | **X** | **Quality measure** | | Authentication mechanism sufficiency | Authentication rules conformity |
| | | Controlled Access Points | Policy | Entitlement | Key Generation | Sensitive Data Protection | **Extracted function**<br>**Controls** | | Authentication | |
| | | DCS-02 | DCS-06 | EKM-01 | EKM-02 | EKM-03 | | | | |
| | | | | | | | 9.4.4 | Use of privileged utility programs | **CQ2** | |
| | | | | **CQ3** | **CQ3** | | 10.1.2 | Key management | **CQ2** | |
| | | **CQ3** | **CQ3** | | | | 11.1.2 | Physical entry controls | **CQ2** | |
| | | | | | | | 11.2.9 | Clear desk and clear screen policy | **CQ2** | |
| | | | | | | | 13.1.2 | Security of network services | **CQ2** | |
| | | | | | | | 13.1.3 | Segregation in networks | **CQ2** | |
| | | | | | | **CQ3** | 13.2.3 | Electronic messaging | **CQ2** | |
| | | | | | | **CQ3** | 14.1.2 | Securing application services on public networks | **CQ2** | |
| | | | | | | **CQ3** | 14.1.3 | Protecting application services transactions | **CQ2** | |

### Security function: Backup of data and backup/restore procedures

**Backup of data** and **backup/restore procedures** are the security functions extracted from the quality measure **backup data completeness**, **periodical backup**, and **architecture recoverability**. Table-A II-25 provides the security functions extracted from the quality measures.

Table-A II-25 Security functions extracted from the quality measures, **backup data completeness**, **periodical backup**, and **architecture recoverability**

| **Extracted from ISO/IEC 25023 quality measure** | **ISO/IEC 25023 quality measures** | | **ISO/IEC 25010 quality model** | |
|---|---|---|---|---|
| **Function** | **Description** | **Quality measure** | **Quality sub-characteristic** | **Quality characteristic** |
| Backup of data | What proportion of data items is backed up regularly? | Backup data completeness | Recoverability | Reliability |

### Step 1: CQ1 instantiation

We found the control description excerpt provided in Table-A II-26 to instantiate our integrated ontology with the security functions **backup of data** and **backup/restore procedures** to answer CQ1.

Table-A II-26 Information backup control description excerpt from ISO/IEC 27002 (ISO/IEC, 2013)

> 12.3 Backup
>
> Objective: To protect against loss of data
>
> 12.3.1 Information backup
>
> Control
> Backup copies of information, software and system images should be taken and tested regularly in accordance with an agreed backup policy.
>
> Implementation guidance
> A backup policy should be established to define the organization's requirements for backup of information, software, and systems.
>
> The backup policy should define the retention and protection requirements.
>
> Adequate backup facilities should be provided to ensure that all essential information and software can be recovered following a disaster or media failure.
>
> When designing a backup plan, the following items should be taken into consideration:
>
> - accurate and complete records of the backup copies and documented restoration procedures should be produced; […]
> - the extent (e.g., full or differential backup) and frequency of backups should reflect the business requirements of the organization, the security requirements of the information involved and the criticality of the information to the continued operation of the organization; […]

According to its description, the **control objective** is "To protect against loss of data" and *IsImplementedBy* the **control** information backup to *Fulfill*:

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** reliability, **quality sub-characteristic** recoverability, and **quality measure** backup of data completeness defined as the measurement function of values of the quality measure elements:
    a. Number of data items actually backed up regularly;
    b. Number of data items requiring backup for error recovery;

2. A **quality requirement** *SpecifiedBy* the **quality characteristic** recoverability and **quality measure** periodical backup defined as the measurement function of values of the quality measure elements:
    a. Number of data items (or data file) successfully backed up periodically;
    b. Number of data items (or data file) to be backed up.

Each **quality requirement** *IsAcheivedBy* a **functional requirement** that *Specifies* the **security function** backup of data.

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** recoverability and **quality measure** architecture recoverability defined as the measurement function of values of the quality measure elements:
    a. Number of elements of the architecture successfully recovered;
    b. Number of elements of the architecture that shall be managed by backup/restore procedures.

The **quality requirement** *IsAcheivedBy* a **functional requirement** that *specifies* the **security function** backup/restore procedures.

**Step 2: CQ2 instantiation**

Based on their descriptions, we found the **controls** provided in Table-A II-27 to instantiate our integrated ontology to answer CQ2 where the **control objective** *IsImplementedBy* the **control** information backup associated with the **security function** backup of data or the **security function** backup/restore procedures.

Table-A II-27 Control description excerpts from ISO/IEC 27002 (ISO/IEC, 2013)

| Control objective | Control | Excerpts from the control description |
|---|---|---|
| To ensure the security of teleworking and use of mobile devices. | 6.2.1 Mobile device policy | "The mobile device policy should consider: …backups…" |
| | 6.2.2 Telenetworking policy | "The guideline and arrangements to be considered: …the procedures for backup and business continuity…" |
| To ensure correct and secure operations of information processing facilities. | 12.1.1 Documented operating procedures | "The operating procedures should specify the operational instructions, including: …backup…" |
| To ensure that information and information processing facilities are protected against malware. | 12.2.1 Controls against malware | "The following guidance should be considered: … preparing appropriate business continuity plans for recovering from malware attacks, including all necessary data and software backup and recovery arrangements…" |
| To ensure that information security is designed and implemented within the development lifecycle of information systems. | 14.2.6 Secure development environment | "Organizations should assess risks associated with individual system development efforts and establish secure development environments for specific system development efforts, considering: …backups are stored at secure offsite locations…" |

**Step 3: CQ3 instantiation**

We first determine if the **control** associated with the **security function** backup of data or the **security function** backup/restore procedures found by CQ1, namely, information backup, can be used to instantiate our integrated ontology to answer CQ3. In this case, although the **control** information backup is mapped to the CSA control retention policy in the CSA CCM, the said CSA control is not mapped to the CSA top threats to cloud computing. Results are provided in Table-A II-28 and Table-A II-29.

Then, we determine if the **controls** associated with the **security function** backup of data or the **security function** backup/restore procedures found by CQ2, namely, mobile device policy, telenetworking policy, documented operating procedures, controls against malware, and secure development environment, can be used to instantiate our integrated ontology to answer CQ3. We found the following instances and the mapping to the CSA top threats to cloud computing is provided in Table-A II-28 and Table-A II-29.

1. The **control** mobile device policy *Mitigates* the **threats**;
    a. Abuse and Nefarious Use of Cloud Services;

2. The **control** telenetworking policy *Mitigates* the **threats**;
    a. Abuse and Nefarious Use of Cloud Services;

3. The **control** documented operating procedures *Mitigates* the **threats**;
    a. Misconfiguration and Inadequate Change Control;
    b. Weak Control Plane;
    c. Metastructure and Applistructure Failures;

4. The **control** controls against malware *Mitigates* the **threats**;
    a. Misconfiguration and Inadequate Change Control.

The **control** secure development environment is not mapped to CSA controls in the CSA CCM and, as a result, there is no relationship to the CSA top threats to cloud computing.

**Step 4: Control and quality measure implementation guidance**

We analyze the available implementation guidance for cloud computing services and for the protection of PII in public cloud computing services of the **control** information backup to determine whether there is evidence to suggest that information and/or procedures and/or support should be provided by the cloud service provider to the cloud service customer to verify the **quality requirements** *SpecifiedBy* the **quality measures** backup data completeness, periodical backup, architecture recoverability.

The implementation guidance for cloud computing services of the **control** information backup stipulates that “where the cloud service provider provides backup capability as part of the cloud service, the cloud service customer should request the specifications of the backup capability from the cloud service provider.” It also stipulates that “the cloud service should also verify that they meet their backup requirements.”

As it pertains to cloud service provider, the implementation guidance stipulates that “the cloud service provider should provide the specifications of its backup capabilities to the cloud service customer.” Moreover, it stipulates that “The specifications should include the following information, as appropriate: scope and schedule of backups; procedures to test the backup capabilities; procedures for verifying integrity of backup data.”

In the case where the **security function** backup of data *IsImplementedBy* the **cloud service provider**, there is evidence to suggest that the information on the scope and schedule of backups and the procedures to test the backup capabilities should be provided by the cloud service provider to the cloud service customer to verify the **quality requirements** *SpecifiedBy* the **quality measures** backup data completeness, periodical backup, and architecture recoverability.

As it pertains to implementation guidance for the protection of PII in public cloud computing services of the **control** information backup, it stipulates that “Where the public cloud PII processor explicitly provides backup and restore services to the cloud service customer, the public cloud PII processor should provide clear information to the cloud service customer about the capabilities of the cloud service with respect to backup and restoration of the cloud service customer data.”

In this case, where the **security function** backup of data *IsImplementedBy* the public cloud computing service PII processor, the implementation guidance for cloud computing services also applies and therefore, there is evidence to suggest that the cloud service customer can verify the **quality requirements** *SpecifiedBy* the **quality measures** backup data completeness, periodical backup, and architecture recoverability.

**Step 5: Additional instantiation**

We found additional instantiations with **quality measures** for which no security function, security architecture, or security-focused verification method was extracted.

Based on its description which stipulates that "Adequate backup facilities should be provided to ensure that all essential information and software can be recovered following a disaster or media failure.", the **control** information backup *Fulfills*:

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** availability and the **quality measure** data recoverability ratio defined as the measurement function of values of the quality measure elements:
    a. Number of data items successfully and correctly recovered by the system;
    b. Number of data items that are required to be recovered;
2. A **quality requirement** *SpecifiedBy* the **quality characteristic** availability and **quality measure** data availability ratio defined as the measurement function of values of the quality measure elements:
    a. Number of data items available within a specific period of time;
    b. Number of data items requested within the same period of time.

As discussed in Step 4, there is evidence that to suggest that the procedures to test the backup capabilities should be provided by the cloud service provider to the cloud service customer to verify the **quality requirements** *SpecifiedBy* the **quality measures** data recoverability ratio and data availability ratio.

We also found evidence in the implementation guidance for cloud computing services to suggest that the procedures to verify the integrity of backup data should be provided by the cloud service provider to the cloud service customer to verify a **quality requirement** *SpecifiedBy* the **quality measure** data integrity defined as the measurement function of values of the quality measure elements:

1. Number of data items (or data file) successfully backed up periodically;
2. Number of data items (or data file) to be backed up.

Table-A II-28 Backup of data and backup/restore procedures applicability: CQ1, CQ2, and CQ3 instantiations

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | | |
| | Account Hijacking | | | | | | | | | |
| | Insider Threat | | | | | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | | |
| | Weak Control Plane | | | | | | | | | |
| | Metastructure and Applistructure Failures | | | | | **Quality characteristic** | | Reliability | Recoverability | |
| | Limited Cloud Usage Visibility | | | | | **Quality sub-characteristic** | | Recoverability | | |
| | Abuse and Nefarious Use of Cloud Services | | **X** | | | **Quality measure** | | Backup data completeness | Periodical backup | Architecture recoverability |
| | | Unauthorized Software Installations | Mobile Device Management | Anti-Virus/ Malicious Software | Mobile code | **Controls** | **Extracted function** | Backup of data | Backup of data | Back-up/restore procedures |
| | | CCC-04 | HRS-05 | TVM-01 | TVM-03 | | | | | |
| | | | | | | 12.3.1 | Information backup | **CQ1** | **CQ1** | **CQ1** |
| | | | **CQ3** | | | 6.2.1 | Mobile device policy | **CQ2** | **CQ2** | **CQ2** |
| | | | **CQ3** | | | 6.2.2 | Telenetworking policy | **CQ2** | **CQ2** | **CQ2** |
| | | | | | | 12.1.1 | Documented operating procedures | **CQ2** | **CQ2** | **CQ2** |
| | | **CQ3** | | **CQ3** | **CQ3** | 12.2.1 | Controls against malware | **CQ2** | **CQ2** | **CQ2** |
| | | | | | | 14.2.6 | Secure development environment | **CQ2** | **CQ2** | **CQ2** |

Table-A II-29 Backup of data and backup/restore procedures applicability: CQ1, CQ2, and CQ3 instantiations

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | | | | |
| | Misconfiguration and Inadequate Change Control | **X** | | | | **X** | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | | | |
| | Account Hijacking | | | | | | | | | | |
| | Insider Threat | | | | | | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | | | |
| | Weak Control Plane | | | **X** | | | | | | | |
| | Metastructure and Applistructure Failures | | | **X** | | | **Quality characteristic** | | Reliability | Recoverability | |
| | Limited Cloud Usage Visibility | | | | | | **Quality sub-characteristic** | | Recoverability | | |
| | Abuse and Nefarious Use of Cloud Services | | | | | | **Quality measure** | | Backup data completeness | Periodical backup | Architecture recoverability |
| | | OS Hardening and Base Control | Retention Policy | Documentation | Policy | Quality Testing | **Controls** / **Extracted function** | | Backup of data | Backup of data | Back-up/restore procedures |
| | | IVS-07 | BCR-11 | BCR-04 | BCR-10 | CCC-03 | | | | | |
| | | | **CQ3** | | | | 12.3.1 | Information backup | **CQ1** | **CQ1** | **CQ1** |
| | | | | | | | 6.2.1 | Mobile device policy | **CQ2** | **CQ2** | **CQ2** |
| | | | | | | | 6.2.2 | Telenetworking policy | **CQ2** | **CQ2** | **CQ2** |
| | | | | **CQ3** | **CQ3** | **CQ3** | 12.1.1 | Documented operating procedures | **CQ2** | **CQ2** | **CQ2** |
| | | **CQ3** | | | | | 12.2.1 | Controls against malware | **CQ2** | **CQ2** | **CQ2** |
| | | | | | | | 14.2.6 | Secure development environment | **CQ2** | **CQ2** | **CQ2** |

### Security architecture: Redundancy of components

**Redundancy of components** is a security architecture extracted from the quality measure **redundancy of components**. Table-A II-30 provides the security architecture extracted from the quality measure.

Table-A II-30 Security architecture extracted from the quality measure **redundancy of components**

| **Extracted from ISO/IEC 25023 quality measures** | **ISO/IEC 25023 quality measures** | **ISO/IEC 25010 quality model** | |
|---|---|---|---|
| **Architecture** | **Quality measure** | **Quality sub-characteristic** | **Quality characteristic** |
| Redundancy of components | Redundancy of components | Fault tolerance | Reliability |

### Step 1: CQ1 instantiation

We found the control description excerpt provided in Table-A II-31 to instantiate our integrated ontology with the security architecture **redundancy of components** to answer CQ1.

Table-A II-31 Availability of information processing facilities control description excerpt from ISO/IEC 27002 (ISO/IEC, 2013)

17.2 Redundancies

Objective: To ensure availability of information processing facilities.

17.2.1 Availability of information processing facilities

Control
Information processing facilities should be implemented with redundancy sufficient to meet availability requirements.

Implementation guidance
Organizations should identify business requirements for the availability of information systems. Where the availability cannot be guaranteed using the existing systems architecture, redundant components or architectures should be considered.

Where applicable, redundant information systems should be tested to ensure the failover from one component to another component works as intended.

According to its description, the **control objective** is "To ensure availability of information processing facilities" and *IsImplementedBy* the **control** availability of information processing facilities to *Fulfill*:

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** reliability, **quality sub-characteristic** fault tolerance, and **quality measure** redundancy of components defined as the measurement function of values of the quality measure elements:
    a. Number of system components redundantly installed;
    b. Number of system components.

**Step 2: CQ2 instantiation**

Based on their descriptions, we found the **control** provided in Table-A II-32 to instantiate our integrated ontology to answer CQ2 where the **control objective** *IsImplementedBy* the **control** availability of information processing facilities associated with the **security architecture** redundancy of components.

Table-A II-32 Control description excerpts from ISO/IEC 27002 (ISO/IEC, 2013)

| **Control objective** | **Control** | **Excerpts from the control description** |
|---|---|---|
| To prevent loss, damage, theft or compromise of assets and interruption to the organization's operations. | 11.2.2 Supporting utilities | "Additional redundancy for network connectivity can be obtained by means of multiple routes from more than one utility provider." |

**Step 3: CQ3 instantiation**

We first determine if the **control** associated with the **security architecture** redundancy of components found by CQ1, namely, availability of information processing facilities, can be used to instantiate our integrated ontology to answer CQ3. The said **control** could not be used to instantiate our integrated ontology to answer CQ3. Results are provided in Table-A II-33.

Then, we determine whether the **control** associated with the **security architecture** redundancy of components found by CQ2, namely, supporting utilities, can be used to instantiate our integrated ontology to answer CQ3. In this case, although the **control** supporting utilities is mapped to the CSA controls data center utilities/environmental conditions, environmental risks, and equipment power failures in the CSA CCM, the said CSA controls are not mapped to the CSA top threats to cloud computing. Results are provided in Table-A II-33.

**Step 4: Control and quality measure implementation guidance**

The **control** availability of information processing facilities does not include implementation guidance for cloud computing services or for the protection of PII in public cloud computing services. Therefore, we consider the **quality measures** redundancy of components under the condition that there is sufficient information and/or procedures and/or support provided by the cloud service provider to the cloud service customer to verify the **quality requirement** *SpecifiedBy* the said **quality measure**.

**Step 5: Additional instantiation**

We found additional instantiations with **quality measures** for which no security function, security architecture, or security-focused verification method was extracted.

Based on its description which stipulates that “Information processing facilities should be implemented with redundancy sufficient to meet availability requirements” and that “Where applicable, redundant information systems should be tested to ensure the failover from one component to another component works as intended”, the **control** availability of information processing facilities *Fulfills*:

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** reliability, **quality sub-characteristic** fault tolerance, and **quality measure** failure avoidance defined as the measurement function of values of the quality measure elements:
    a. Number of avoided critical and serious failure occurrences (based on test cases);
    b. Number of executed test cases of a fault pattern (almost causing failure) during testing;
2. A **quality requirement** *SpecifiedBy* the **quality characteristic** reliability, **quality sub-characteristic** availability, and **quality measure** system availability defined as the measurement function of values of the quality measure elements:

    a. System operation time actually provided;
    b. System operation time specified in the operation schedule;

3. A **quality requirement** *SpecifiedBy* the **quality characteristic** reliability, **quality sub-characteristic** availability, and **quality measure** mean down time defined as the measurement function of values of the quality measure elements:
    a. Total down time;
    b. Number of breakdowns observed.

As discussed in Step 4, the **control** availability of information processing facilities does not include implementation guidance for cloud computing services or for the protection of PII in public cloud computing services. However, in the case of availability measures, namely, system availability, and mean down time, ISO/IEC 25023 (ISO/IEC, 2016) implementation guidance stipulates that "Externally, availability can be assessed by the proportion of total time during which the system, product or component is in an up state." Therefore, we consider that the cloud service customer can verify the **quality requirements** *SpecifiedBy* the **quality measures** system availability and the mean down time.

As for the **quality measure** failure avoidance, there is no implementation guidance to suggest that the cloud service customer can verify the **quality requirement** *SpecifiedBy* the said **quality measure**. Therefore, we consider the **quality measure** failure avoidance applicable under the condition that there is sufficient information and/or procedures and/or support provided by the cloud service provider to the cloud service customer to verify the **quality requirement** *SpecifiedBy* the said **quality measure**.

Table-A II-33 Redundancy of components applicability: CQ2 and CQ3 instantiations scenarios

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | |
| | Account Hijacking | | | | | | |
| | Insider Threat | | | | | | |
| | Insecure Interfaces and APIs | | | | | | |
| | Weak Control Plane | | | | | | |
| | Metastructure and Applistructure Failures | | | | **Quality characteristic** | | Reliability |
| | Limited Cloud Usage Visibility | | | | **Quality sub-characteristic** | | Fault tolerance |
| | Abuse and Nefarious Use of Cloud Services | | | | **Quality measure** | | Redundancy of components |
| | | Datacenter Utilities/ Environmental Conditions | Environmental Risks | Equipment Power Failures | **Extracted function**<br>**Controls** | | Redundancy of components |
| | | BCR-03 | BCR-05 | BCR-08 | | | |
| | | | | | 17.2.1 | Availability of information processing facilities | **CQ1** |
| | | **CQ3** | **CQ3** | **CQ3** | 11.2.2 | Supporting utilities | **CQ2** |

### Security-focused verification method: Penetration testing

**Penetration testing** is the security-focused verification method extracted from the quality measure **non vulnerability**. Table-A II-34 provides the security-focused verification method extracted from the quality measure.

Table-A II-34 Security-focused verification method extracted from the quality measure **non vulnerability**

| **Extracted from ISO/IEC 25024 quality measures** | **ISO/IEC 25024 quality measures** | **ISO/IEC 25012 quality model** |
|---|---|---|
| **Test methodology** | **Quality measure** | **Quality characteristic** |
| Penetration test | Non vulnerability | Confidentiality |

### Step 1: CQ1 instantiation

We found the control description excerpt provided in Table-A II-35 to instantiate our integrated ontology with the security-focused verification method **penetration test** to answer CQ1.

Table-A II-35 Technical compliance control description excerpt from ISO/IEC 27002 (ISO/IEC, 2013)

> 18.2 Information security reviews
>
> Objective: To ensure that information security is implemented and operated in accordance with the organizational policies and procedures.
>
> 18.2.3 Technical compliance review
>
> Control
> Information systems should be regularly reviewed for compliance with the organization's information security policies and standards.
>
> Implementation guidance
> […] If penetration tests or vulnerability assessments are used, caution should be exercised as such activities could lead to a compromise of the security of the system. Such tests should be planned, documented and repeatable. […]

According to its description, the **control objective** is “To ensure that information security is implemented and operated in accordance with the organizational policies and procedures” and *IsImplementedBy* the **control** technical compliance review to *Fulfill*:

1. A **quality requirement** *SpecifiedBy* the **quality characteristic** confidentiality and the **quality measure** non vulnerability defined as the measurement function of values of the quality measure elements:
    a. Number of accesses successfully performed during formal penetration attempts by unauthorized users to reach a target data item within a specific period of time;
    b. Number of accesses attempted by unauthorized users to target a data item in a specific period of time.

The **quality requirement** *IsVerifiedVia* the **security-focused verification method** penetration test.

**Step 2: CQ2 instantiation**

We found no **controls** to instantiate our integrated ontology to answer CQ2 where the **control objective** *IsImplementedBy* the **control** technical compliance associated with the **security-focused verification method** penetration test.

**Step 3: CQ3 instantiation**

We determine whether the **control** associated with the **security-focused verification method** penetration test found by CQ1, namely, technical compliance review, can be used to instantiate our integrated ontology to answer CQ3. We found the following instances, while the mapping to the CSA top threats to cloud computing is provided in Table-A II-36, Table-A II-37, Table-A II-38, Table-A II-39, Table-A II-40, and Table-A II-41.

1. The **control** technical compliance review *Mitigates* the **threats**;
    a. Data Breaches;
    b. Misconfiguration and Inadequate Change Control;

c. Lack of Cloud Security Architecture and Strategy;
d. Account Hijacking;
e. Insider Threat;
f. Weak Control Plane;
g. Metastructure and Applistructure Failures;
h. Abuse and Nefarious Use of Cloud Services.

**Step 4: Control and quality measure implementation guidance**

The control technical compliance review does not include implementation guidance for cloud computing services or for the protection of PII in public cloud computing services. However, ISO/IEC 25024 stipulates that "Penetration tests can be performed to simulate an attack because such attack does not normally occur in the usual testing." As an attack on the cloud computing service, authorized by the cloud computing service provider, can be performed externally, we consider that the cloud service customer can verify the **quality requirement** *SpecifiedBy* the **quality measure** non vulnerability.

**Step 5: Additional instantiation**

No additional instantiations were found.

Table-A II-36 Penetration test applicability: CQ1 and CQ3 instantiations

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | | |
| | Account Hijacking | | | | | | | | | |
| | Insider Threat | | | | | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | | |
| | Weak Control Plane | | | | | | | | | |
| | Metastructure and Applistructure Failures | | | | | | | **Quality characteristic** | | Confidentiality |
| | Limited Cloud Usage Visibility | | | | | | | **Quality sub-characteristic** | | |
| | Abuse and Nefarious Use of Cloud Services | | | | | | | **Quality measure** | | Non vulnerability |
| | | Policy | Remote Wipe | Security Patches | Users | Data Quality and Integrity | Provider Internal Assessments | **Controls** / **Extracted function** | | Penetration test |
| | | MOS-17 | MOS-18 | MOS-19 | MOS-20 | STA-01 | STA-04 | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 18.2.3 | Technical compliance review | **CQ1** |

Table-A II-37 Penetration test applicability: CQ1 and CQ3 instantiations

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | | | |
| | Account Hijacking | | | | | | | | | | |
| | Insider Threat | | | | | | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | | | |
| | Weak Control Plane | | | | | | | | | | |
| | Metastructure and Applistructure Failures | | | | | | | | **Quality characteristic** | | Confidentiality |
| | Limited Cloud Usage Visibility | | | | | | | | **Quality sub-characteristic** | | |
| | Abuse and Nefarious Use of Cloud Services | | | | | | | | **Quality measure** | | Non vulnerability |
| | | Device Management | Device Encryption | Jailbreaking and Rooting | Legal | Lockout Screen | Operating Systems | Passwords | **Controls** | **Extracted function** | Penetration test |
| | | MOS-10 | MOS-11 | MOS-12 | MOS-13 | MOS-14 | MOS-15 | MOS-16 | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 18.2.3 | Technical compliance review | **CQ1** |

Table-A II-38 Penetration test applicability: CQ1 and CQ3 instantiations

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | | |
| | Account Hijacking | | | | | | | | | |
| | Insider Threat | | | | | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | | |
| | Weak Control Plane | | | | | | | | | |
| | Metastructure and Applistructure Failures | | | | | | | **Quality characteristic** | | Confidentiality |
| | Limited Cloud Usage Visibility | | | | | | | **Quality sub-characteristic** | | |
| | Abuse and Nefarious Use of Cloud Services | **X** | **X** | **X** | | | | **Quality measure** | | Non vulnerability |
| | | Approved Software for BYOD | Awareness and Training | Cloud Based Services | Compatibility | Device Eligibility | Device Inventory | **Controls** / **Extracted function** | | Penetration test |
| | | MOS-04 | MOS-05 | MOS-06 | MOS-07 | MOS-08 | MOS-09 | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 18.2.3 | Technical compliance review | **CQ1** |

Table-A II-39 Penetration test applicability: CQ1 and CQ3 instantiations

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | | |
| | Account Hijacking | | | | | | | | | |
| | Insider Threat | | | | | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | | |
| | Weak Control Plane | | | | | | | | | |
| | Metastructure and Applistructure Failures | | | | | | | **Quality characteristic** | | Confidentiality |
| | Limited Cloud Usage Visibility | | | | | | | **Quality sub-characteristic** | | |
| | Abuse and Nefarious Use of Cloud Services | | | | | **X** | **X** | **Quality measure** | | Non vulnerability |
| | | Policy & Legal | Standardized Network Protocols | Virtualization | Anti-Malware | Application Stores | Approved Applications | **Controls** | **Extracted function** | Penetration test |
| | | IPY-03 | IPY-04 | IPY-05 | MOS-01 | MOS-02 | MOS-03 | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 18.2.3 | Technical compliance review | **CQ1** |

Table-A II-40 Penetration test applicability: CQ1 and CQ3 instantiations

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | X | | | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | | | |
| | Lack of Cloud Security Architecture and Strategy | X | | | | | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | |
| | Account Hijacking | | | | | | | | |
| | Insider Threat | | X | | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | |
| | Weak Control Plane | X | X | X | | | | | |
| | Metastructure and Applistructure Failures | | | | X | | **Quality characteristic** | | Confidentiality |
| | Limited Cloud Usage Visibility | | | | | | **Quality sub-characteristic** | | |
| | Abuse and Nefarious Use of Cloud Services | | | | | | **Quality measure** | | Non vulnerability |
| | | Policy Impact on Risk Assessments | Risk Assessments | Risk Management Framework | APIs | Data Request | **Controls** / **Extracted function** | | Penetration test |
| | | GRM-08 | GRM-10 | GRM-11 | IPY-01 | IPY-02 | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 18.2.3 | Technical compliance review | **CQ1** |

Table-A II-41 Penetration test applicability: CQ1 and CQ3 instantiations

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cloud Security Alliance (CSA) Top Threats** | Data Breaches | | | | | | | | | |
| | Misconfiguration and Inadequate Change Control | | | | | | **X** | | | |
| | Lack of Cloud Security Architecture and Strategy | | | | | | **X** | | | |
| | Insufficient Identity, Credential, Access, and Key Management | | | | | | | | | |
| | Account Hijacking | **X** | | | | | | | | |
| | Insider Threat | | | | | | | | | |
| | Insecure Interfaces and APIs | | | | | | | | | |
| | Weak Control Plane | **X** | | | | | **X** | | | |
| | Metastructure and Applistructure Failures | | | | | | | **Quality characteristic** | | Confidentiality |
| | Limited Cloud Usage Visibility | | | | | | | **Quality sub-characteristic** | | |
| | Abuse and Nefarious Use of Cloud Services | **X** | | | | | **X** | **Quality measure** | | Non vulnerability |
| | | Audit Logging/ Intrusion Detection | Data Protection | Hypervisor Hardening | Outsourced Development | Quality Testing | Baseline Requirement | **Controls** / **Extracted function** | | Penetration test |
| | | IVS-01 | IVS-10 | IVS-11 | CCC-02 | CCC-03 | GRM-01 | | | |
| | | **CQ3** | **CQ3** | **CQ3** | **CQ3** | **CQ3** | **CQ3** | 18.2.3 | Technical compliance review | **CQ1** |

# LIST OF BIBLIOGRAPHICAL REFERENCES

Abdeladim, A., S. Baina and K. Baina (2014). “Elasticity and scalability centric quality model for the cloud.” 2014 Third IEEE International Colloquium in Information Science and Technology (CIST), IEEE.

Agrawal, V. (2016). “Towards the Ontology of ISO/IEC 27005: 2011 Risk Management Standard.” HAISA.

Alliance, C. S. (2019). "Top threats to cloud computing the egregious 11."

Arbanas, K. and M. Čubrilo (2015). "Ontology in information security." Journal of Information and Organizational Sciences **39**(2): 107-136.

Armbrust, M., A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin and I. Stoica (2009). "Above the Clouds: A Berkeley View of Cloud Computing."

Assawamekin, N., T. Sunetnanta and C. Pluempitiwiriyawej (2010). "Ontology-based multiperspective requirements traceability framework." Knowledge and Information Systems **25**(3): 493-522.

Atkinson, C. and T. Kuhne (2003). "Model-driven development: a metamodeling foundation." IEEE software **20**(5): 36-41.

Barbosa, F. P. and A. S. Charão (2012). “Impact of pay-as-you-go cloud platforms on software pricing and development: a review and case study.” International Conference on Computational Science and Its Applications, Springer.

Bautista, L., A. Abran and A. April (2012). "Design of a performance measurement framework for cloud computing."

Benlian, A., M. Koufaris and T. Hess (2011). "Service quality in software-as-a-service: Developing the SaaS-Qual measure and examining its role in usage continuance." Journal of management information systems **28**(3): 85-126.

Boehm, B. and V. R. Basili (2005). "Software defect reduction top 10 list." Foundations of empirical software engineering: the legacy of Victor R. Basili **426**(37): 426-431.

Castillo, I., F. Losavio, A. Matteo and J. Bøegh (2010). "REquirements, Aspects and Software Quality: the REASQ model." J. Object Technol. **9**(4): 69-91.

Cherfi, S. S.-s., J. Akoka and I. Comyn-Wattiau (2011). “Federating information system quality frameworks using a common Ontology.” ICIQ.

Choi, C.-R. and H.-Y. Jeong (2014). "Quality evaluation and best service choice for cloud computing based on user preference and weights of attributes using the analytic network process." Electronic Commerce Research **14**(3): 245-270.

Den Braber, F., I. Hogganvik, M. S. Lund, K. Stølen and F. Vraalsen (2007). "Model-based security analysis in seven steps—a guided tour to the CORAS method." BT Technology Journal **25**(1): 101-117.

dos Santos Moreira, E., L. A. F. Martimiano, A. J. dos Santos Brandao and M. C. Bernardes (2008). "Ontologies for information security management and governance." Information Management & Computer Security.

Fenz, S. and A. Ekelhart (2009). “Formalizing information security knowledge.” Proceedings of the 4th international Symposium on information, Computer, and Communications Security.

Fox, M. S., M. Barbuceanu and M. Gruninger (1995). “An organisation ontology for enterprise modelling: preliminary concepts for linking structure and behaviour.” Proceedings 4th IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'95), IEEE.

Garg, S. K., S. Versteeg and R. Buyya (2011). “Smicloud: A framework for comparing and ranking cloud services.” 2011 Fourth IEEE International Conference on Utility and Cloud Computing, IEEE.

Ghaisas, S. and N. Ajmeri (2013). “Knowledge-assisted ontology-based requirements evolution.” Managing requirements knowledge, Springer**:** 143-167.

Goeken, M. and S. Alter (2009). “Towards conceptual metamodeling of IT governance frameworks approach-use-benefits.” 2009 42nd Hawaii International Conference on System Sciences, IEEE.

Gómez-Pérez, A. (1996). "Towards a framework to verify knowledge sharing technology." Expert Systems with applications **11**(4): 519-529.

Gotel, O., J. Cleland-Huang, J. H. Hayes, A. Zisman, A. Egyed, P. Grünbacher, A. Dekhtyar, G. Antoniol, J. Maletic and P. Mäder (2012). “Traceability fundamentals.” Software and systems traceability, Springer**:** 3-22.

Grönroos, C. (2007). “Service management and marketing: customer management in service competition, John Wiley & Sons.

Herzog, A., N. Shahmehri and C. Duma (2007). "An ontology of information security." International Journal of Information Security and Privacy (IJISP) **1**(4): 1-23.

Hevner, A. R., S. T. March, J. Park and S. Ram (2004). "Design science in information systems research." MIS quarterly: 75-105.

ISO/IEC (2008). “ISO/IEC 25012 Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model. Geneva, International Organization for Standardization.

ISO/IEC (2012). “ISO/IEC 25021 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality measure elements. Geneva, International Organization for Standardization.

ISO/IEC (2013). “ISO/IEC 27002 Information technology — Security techniques — Code of practice for information security controls. Geneva, International Organization for Standardization.

ISO/IEC (2014). “ISO/IEC 17788 Information technology — Cloud computing — Overview and vocabulary. Geneva, International Organization for Standardization.

ISO/IEC (2014). “ISO/IEC 27036-1 Information technology — Security techniques — Information security for supplier relationships — Part 1: Overview and concepts. Geneva, International Organization for Standardization.

ISO/IEC (2015). “ISO/IEC 25024 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of data quality. Geneva, International Organization for Standardization.

ISO/IEC (2015). “ISO/IEC 27017 Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services. Geneva, International Organization for Standardization.

ISO/IEC (2016). “ISO/IEC 25022 Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Measurement of quality in use. Geneva, International Organization for Standardization.

ISO/IEC (2016). “ISO/IEC 25023 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality. Geneva, International Organization for Standardization.

ISO/IEC (2017). “ISO/IEC TS 25011 Information technology — Systems and software Quality Requirements and Evaluation (SQuaRE) — Service quality models. Geneva, International Organization for Standardization.

ISO/IEC (2018). “ISO/IEC 27000 Information technology — Security techniques — Information security management systems — Overview and vocabulary. Geneva, International Organization for Standardization.

ISO/IEC (2018). “ISO/IEC 27005 Information technology — Security techniques — Information security risk management. Geneva, International Organization for Standardization.

ISO/IEC (2019). “ISO/IEC 25010 CD Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Geneva, International Organization for Standardization.

ISO/IEC (2019). “ISO/IEC 25020 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality measurement framework. Geneva, International Organization for Standardization.

ISO/IEC (2019). “ISO/IEC 25030 Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Quality requirements framework. Geneva, International Organization for Standardization.

ISO/IEC (2019). “ISO/IEC 27018 Information technology — Security techniques — Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors. Geneva, International Organization for Standardization.

ISO/IEC (2020). “ISO/IEC AWI TS 25052-1 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE): cloud services — Part 1: Quality Model for SaaS (Software as a Service). Geneva, International Organization for Standardization.

ISO/IEC/IEEE (2015). “ISO/IEC/IEEE 15288 Systems and software engineering — System life cycle processes. Geneva, International Organization for Standardization.

Jarraya, Y., A. Eghtesadi, M. Debbabi, Y. Zhang and M. Pourzandi (2012). “Cloud calculus: Security verification in elastic cloud computing platform.” 2012 international conference on collaboration technologies and systems (CTS), IEEE.

Jeong, H. Y. and B. H. Hong (2013). “The Identification of Quality Attributes for SaaS in Cloud Computing.” Applied Mechanics and Materials, Trans Tech Publ.

Lee, J. Y., J. W. Lee, D. W. Cheun and S. D. Kim (2009). “A quality model for evaluating software-as-a-service in cloud computing.” Software Engineering Research, Management and Applications, 2009. SERA'09. 7th ACIS International Conference on, IEEE.

Lima, J. F., B. P. Garcia, C. M. G. Amaral and G. M. Caran (2011). “Building an ontological model for software requirements engineering.” International Conference on ENTERprise Information Systems, Springer.

Lyytinen, K. and R. Hirschheim (1988). “Information systems failures—a survey and classification of the empirical literature.” Oxford surveys in information technology**:** 257-309.

Macaulay, L. (1996). “Requirements for requirements engineering techniques.” Proceedings of the Second International Conference on Requirements Engineering, IEEE.

Madi, T., Y. Jarraya, A. Alimohammadifar, S. Majumdar, Y. Wang, M. Pourzandi, L. Wang and M. Debbabi (2018). "ISOTOP: auditing virtual networks isolation across cloud layers in OpenStack." ACM Transactions on Privacy and Security (TOPS) **22**(1): 1-35.

Madi, T., S. Majumdar, Y. Wang, Y. Jarraya, M. Pourzandi and L. Wang (2016). “Auditing security compliance of the virtualized infrastructure in the cloud: Application to openstack.” Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy.

Magnani, M., N. Rizopoulos, P. M. Brien and D. Montesi (2005). “Schema integration based on uncertain semantic mappings.” International Conference on Conceptual Modeling, Springer.

March, S. T. and V. C. Storey (2008). "Design science in the information systems discipline: an introduction to the special issue on design science research." MIS quarterly: 725-730.

Matthews, J., T. Garfinkel, C. Hoff and J. Wheeler (2009). “Virtual machine contracts for datacenter and cloud computing environments.” Proceedings of the 1st workshop on Automated control for datacenters and clouds.

Milicevic, D. and M. Goeken (2010). “Ontology-based evaluation of ISO 27001.” Conference on e-Business, e-Services and e-Society, Springer.

Nakajima, T. (2020). “Study group report on square future direction.” CEUR Workshop Proceedings, CEUR-WS.

NIST (2018). “NIST SP 800-160 Vol 1 Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems. U.S. Department of Commerce, National Institute of Standards and Technology.

Noy, N. F. and D. L. McGuinness (2001). “Ontology development 101: A guide to creating your first ontology, Stanford knowledge systems laboratory technical report KSL-01-05 and ….

Parasuraman, A., V. A. Zeithaml and L. L. Berry (1988). "Servqual: A multiple-item scale for measuring consumer perc." Journal of retailing **64**(1): 12.

Parkin, S. E., A. van Moorsel and R. Coles (2009). “An information security ontology incorporating human-behavioural implications.” Proceedings of the 2nd International Conference on Security of Information and Networks.

Pereira, T. S. M. and H. M. D. Santos (2012). “An Ontology Approach in Designing Security Information Systems to Support Organizational Security Risk Knowledge.” KEOD.

Perez-Botero, D., J. Szefer and R. B. Lee (2013). “Characterizing hypervisor vulnerabilities in cloud computing servers.” Proceedings of the 2013 international workshop on Security in cloud computing.

Ramesh, B. and M. Jarke (2001). "Toward reference models for requirements traceability." IEEE transactions on software engineering **27**(1): 58-93.

Rao, U. H. and U. Nayak (2014). “The InfoSec handbook: An introduction to information security, Springer.

Rehn, C. (2009). "Software architectural tactics and patterns for safety and security." TU Kaiserslautern **67663**.

Repschlaeger, J., S. Wind, R. Zarnekow and K. Turowski (2013). "Decision model for selecting a cloud provider: A study of service model decision priorities."

Rizopoulos, N. and P. Mçbrien (2005). "A general approach to the generation of conceptual model transformations." International Conference on Advanced Information Systems Engineering, Springer.

Roedler, G. J. and C. Jones (2005). "Technical Measurement. A Collaborative Project of PSM, INCOSE, and Industry, ARMY ARMAMENT RESEARCH DEVELOPMENT AND ENGINEERING CENTER PICATINNY ARSENAL NJ.

Ross, R., M. McEvilley and J. Oren (2018). "Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems, National Institute of Standards and Technology.

Schlauderer, S. and S. Overhage (2015). "Selecting Cloud Service Providers-Towards a Framework of Assessment Criteria and Requirements." Wirtschaftsinformatik.

Schuette, R. and T. Rotthowe (1998). "The guidelines of modeling–an approach to enhance the quality in information models." International Conference on Conceptual Modeling, Springer.

Siegemund, K. (2014). "Contributions to ontology-driven requirements engineering, Citeseer.

Sonnenberg, C. and J. Vom Brocke (2011). "Evaluation patterns for design science research artefacts." European Design Science Symposium, Springer.

Suryn, W. (2014). "Software quality engineering: a practitioner's approach, John Wiley & Sons.

Swaid, S. I. and R. T. Wigand (2009). "The customer perspective of E-Service quality: An empirical study." Electronic Markets, Springer**:** 36-61.

Tamura, N. and M. Banbara (2008). "Sugar: A CSP to SAT translator based on order encoding." Proceedings of the Second International CSP Solver Competition: 65-69.

Uschold, M. and M. Gruninger (1996). "Ontologies: Principles, methods and applications." TECHNICAL REPORT-UNIVERSITY OF EDINBURGH ARTIFICIAL INTELLIGENCE APPLICATIONS INSTITUTE AIAI TR.

Uschold, M. and R. Jasper (1999). “A framework for understanding and classifying ontology applications.” Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden.

Villalpando, L. E. B., A. April and A. Abran (2014). "Performance analysis model for big data applications in cloud computing." Journal of Cloud Computing **3**(1): 19.

Wen, P. X. and L. Dong (2013). “Quality model for evaluating SaaS service.” 2013 Fourth international conference on emerging intelligent data and web technologies, IEEE.

Wheatcraft, L. S., M. J. Ryan and J. Dick (2016). "On the use of attributes to manage requirements." Systems Engineering **19**(5): 448-458.

Wibowo, A. and J. Davis (2020). “Requirements Traceability Ontology to Support Requirements Management.” Proceedings of the Australasian Computer Science Week Multiconference.

Wollersheim, J. and H. Krcmar (2014). “Quality analysis approaches for cloud services-Towards a framework along the customer’s activity cycle.” Trusted Cloud Computing, Springer**:** 109-124.

Zhang, Y., A. Juels, A. Oprea and M. K. Reiter (2011). “Homealone: Co-residency detection in the cloud via side-channel analysis.” 2011 IEEE symposium on security and privacy, IEEE.

Zhang, Y., R. Witte, J. Rilling and V. Haarslev (2006). “An ontology-based approach for traceability recovery.” 3rd International Workshop on Metamodels, Schemas, Grammars, and Ontologies for Reverse Engineering (ATEM 2006), Genoa.

Zheng, X., P. Martin, K. Brohman and L. Da Xu (2014). "Cloudqual: A quality model for cloud services." IEEE transactions on industrial informatics **10**(2): 1527-1536.