Communications of the Association for Information Systems

Volume 44 Article 25

4-2019

Designing Monitoring Systems for Continuous Certification of Cloud Services: Deriving Meta-requirements and Design Guidelines

Sebastian Lins
Karlsruhe Institute of Technology, sebastian.lins@kit.edu

Stephan Schneider
Karlsruhe Institute of Technology

Jakub Szefer
Yale University

Shafeeq Ibraheem
Yale University

Ali Sunyaev
Karlsruhe Institute of Technology

Recommended Citation

Lins, S., Schneider, S., Szefer, J., Ibraheem, S., & Sunyaev, A. (2019). Designing Monitoring Systems for Continuous Certification of Cloud Services: Deriving Meta-requirements and Design Guidelines. Communications of the Association for Information Systems, 44, pp-pp. https://doi.org/10.17705/1CAIS.04425

Communications of the Association for Information Systems

Research Paper DOI: 10.17705/1CAIS.04425 ISSN: 1529-3181

Designing Monitoring Systems for Continuous Certification of Cloud Services: Deriving Meta-requirements and Design Guidelines

Sebastian Lins
Research Group Critical Information Infrastructures
Institute of Applied Informatics and Formal Description Methods
Karlsruhe Institute of Technology
Germany
sebastian.lins@kit.edu

Stephan Schneider
Research Group Critical Information Infrastructures
Institute of Applied Informatics and Formal Description Methods
Karlsruhe Institute of Technology
Germany

Jakub Szefer
Department of Electrical Engineering,
Yale University
USA

Shafeeq Ibraheem
Department of Electrical Engineering
Yale University
USA

Ali Sunyaev
Research Group Critical Information Infrastructures
Institute of Applied Informatics and Formal Description Methods
Karlsruhe Institute of Technology
Germany

Abstract:

Continuous service certification (CSC) involves the consistently gathering and assessing certification-relevant information about cloud service operations to validate whether they continue to adhere to certification criteria. Previous research has proposed test-based CSC methodologies that directly assess the components of cloud service infrastructures. However, test-based certification requires that certification authorities can access the cloud infrastructure, which various issues may limit. To address these challenges, cloud service providers need to conduct monitoring-based CSC; that is, monitor their cloud service infrastructure to gather certification-relevant data by themselves and then provide these data to certification authorities. Nevertheless, we need to better understand how to design monitoring systems to enable cloud service providers to perform such monitoring. By taking a design science perspective, we derive universal meta-requirements and design guidelines for CSC monitoring systems based on findings from five expert focus group interviews with 33 cloud experts and 10 one-to-one interviews with cloud customers. With this study, we expand the current knowledge base regarding CSC and monitoring-based CSC. Our derived design guidelines contribute to the development of CSC monitoring systems and enable monitoring-based CSC that overcomes issues of prior test-based approaches.

Keywords: Cloud Computing, Continuous Certification, Monitoring Based, Continuous Monitoring, Security, Privacy.

This manuscript underwent peer review. It was received 09/09/2016 and was with the authors for 14 months for 3 revisions. Tilo Böhmann served as Associate Editor.

1 Introduction

Recently, researchers and practitioners have started to investigate how to innovate the process of certifying cloud services to address the challenges of an ever-changing cloud service infrastructure and to increase the reliability of issued cloud service certifications (e.g., *EuroCloud's* StarAudit or *CSA's* Security, Trust & Assurance Registry). These efforts have generated innovative specifications of architectures, processes, and prototypes that enable certification authorities to continually certify cloud services (Anisetti, Ardagna, & Damiani, 2015; Lins, Schneider, & Sunyaev, 2018; Stephanow & Fallenbeck, 2015). Continuous service certification (CSC) involves consistently gathering and assessing certification-relevant information about cloud service operations to validate whether they adhere to ongoing certification criteria. CSC represents a disruptive change because it provides cloud customers with ongoing, up-to-date feedback about a cloud service's security and privacy levels compared with conventional certifications that assess a cloud service only at a specific point in time. Although the innovative CSC concept has recently gained importance, it remains underexplored and has been test-marketed and evaluated only in trials (Krotsiani, Spanoudakis, & Kloukinas, 2015; Teigeler, Lins, & Sunyaev, 2018). Certification authorities and cloud service providers continue to struggle with implementing CSC processes and systems due to high complexity and a challenging interplay between both parties. In particular, certification authorities encounter the problem of how to effectively and consistently collect and assess comprehensive, reliable information about cloud service operations (Lins et al., 2018; Lins, Teigeler, & Sunyaev, 2016b).

Prior research has already proposed conceptual architectures and techniques to continuously audit and certify cloud service providers that one can categorize as either test-based or monitoring-based CSC (Lins et al., 2018; Stephanow, Banse, & Schütte, 2016b). Certification authorities operate test-based methodologies and directly access the cloud service infrastructure to examine cloud service components and operations (Stephanow & Banse, 2017), such as verifying the integrity of multiple cloud users' data (Wang, Li, & Li, 2014), or to validate whether they adhere to security criteria (Stephanow & Khajehmoogahi, 2017). However, certification authorities cannot easily apply test-based certification in practice because they need access to the cloud infrastructure. Cloud service providers may refuse required infrastructure access due to organizational issues (e.g., resistance to integrate untrustworthy techniques of certification authorities) and regulatory issues (e.g., data protection laws) (Lins et al., 2018). In addition, performing test-based CSC requires certification authorities to configure and adjust applied techniques in accordance with the cloud infrastructure and heterogonous data formats. This adaptation is challenging in cloud service environments because cloud infrastructures exhibit dynamic characteristics (i.e., dynamic reassignment of resources) and feature fast technology lifecycles and ongoing technical changes (i.e., due to agile software development), which ultimately limits the extent to which certification authorities can apply test-based CSC.

Monitoring-based CSC strategies provide auspicious means to overcome these challenges. When conducting monitoring-based CSC, cloud service providers monitor their cloud service infrastructure to collect data by themselves and then provide certification-relevant information to certification authorities (Kunz & Stephanow, 2017). Consequently, monitoring-based CSC does not require invasive cloud infrastructure access from certification authorities in contrast to test-based CSC. More importantly, unlike in test-based CSC scenarios, cloud service providers can independently alter their cloud infrastructure while ensuring that they still transmit certification-relevant data to certification authorities. Despite these benefits of monitoring-based over test-based CSC, cloud service providers' providing certification-relevant data has one challenging drawback: the risk that they will deliberately manipulate data. Providers may euphemize provided data to assure certification criteria adherence; therefore, providers and certification authorities must prove that malicious providers do not manipulate monitoring data. Likewise, cloud service providers must set up sophisticated CSC monitoring systems that aggregate certification-relevant data across implemented monitoring technologies and format certification-relevant data in accordance with certification authorities' needs. Because previous research has mostly focused on achieving and applying test-based CSC, we need to better understand how to design such CSC monitoring systems to address these challenges and to enable cloud service providers and certification authorities to conduct monitoring-based CSC. To address this gap, we answer the research question:

RQ: How should cloud service providers design a CSC monitoring system to enable certification authorities to conduct monitoring-based CSC?

To counteract current the shortcomings in test-based CSC and foster the diffusion and application of CSC, we followed Kuechler and Vaishnavi's (2008) design science approach and derived universal meta-

requirements and design guidelines for CSC monitoring systems. Because CSC has recently gained research importance (Stephanow & Banse, 2017; Teigeler et al., 2018), we applied an explorative and inductive research approach. First, we derived meta-requirements that help cloud service providers design CSC monitoring systems by conducting comprehensive interviews with cloud service providers, certification authorities, consultants, and customers. Second, we reviewed related literature and surveyed available monitoring technologies to propose design guidelines that fulfill the derived meta-requirements.

Our findings reveal that CSC monitoring systems must fulfill various requirements, such as monitoring and transmitting information in an aggregated and anonymized manner. To address these requirements, we show how cloud service providers should properly design CSC monitoring systems by, for example, applying an agent-based system architecture and integrating existing IT infrastructure monitoring systems and corresponding plugins. We provide several contributions to practice and research. We identify meta-requirements for designing a CSC monitoring system based on exhaustive discussions with industry experts and, thus, expand the current design knowledge base regarding CSC and monitoring-based CSC. Our derived design guidelines contribute to developing CSC monitoring systems and enable monitoring-based CSC that overcomes the issues that prior test-based approaches experience in highly dynamic cloud environments.

This paper proceeds as follows. In Section 2, we provide background information on cloud computing, CSC, and related work. In Section 3, we describe our design science research approach. In Section 4, we outline how we derived the meta-requirements in detail and present our findings. In Section 5, we propose design guidelines to fulfill these meta-requirements. In Section 6, we discuss the benefits and challenges of monitoring-based CSC, implications, and directions for future research. Finally, in Section 7, we conclude the paper.

2 Theoretical Background

2.1 Cloud Computing

Cloud computing enables organizations to ubiquitously and on demand access a shared pool of configurable and rapidly scalable computing resources without significant management effort or interaction with the service provider (Mell & Grance, 2011). These computing resources typically involve hardware (infrastructure as a service (IaaS)), development platforms (platform as a service (PaaS)), and applications (software as a service (SaaS)). Cloud computing entails five essential characteristics: 1) on-demand self-service access to 2) virtualized, shared, and managed IT resources that are 3) scalable on-demand, 4) available over a network, and 5) priced on a pay-per-use basis. These characteristics render cloud computing an attractive alternative to traditional information technologies for organizations in diverse industries (i.e., healthcare (Gao, Thiebes, & Sunyaev, 2018; Thiebes, Kleiber, & Sunyaev, 2017)) while challenging contemporary security and privacy risk-assessment approaches (Benlian, Kettinger, Sunyaev, & Winkler, 2018; Hentschel, Leyh, & Petznick, 2018). For example, a multi-tenant and virtualized approach seems promising from a cloud provider's perspective in terms of profit but increases the risk of co-location attacks due to inappropriate logical and virtual isolation.

An increasing amount of research and industry reports has focused on identifying and addressing security and privacy risks, such as Internet, network, and access security issues and risks regarding non-compliance with regulatory requirements (refer to Fernandes, Soares, Gomes, Freire, & Inácio (2014) for an excellent review of security issues in cloud environments). For example, cloud computing can experience software security risks due to software bugs or faults, such as when a cloud provider insufficiently isolates cloud users on a virtual machine layer (Fernandes et al., 2014). In addition, outsourced data may fall victim to insiders' and/or outsiders' tampering (Sood, 2012), and cloud service providers must address security issues related to data storage, unreliable computing, availability, cryptography, sanitization, malware, and others (Fernandes et al., 2014; Subashini & Kavitha, 2011). Consequently, cloud service providers require diverse strategies to prove that they have satisfactorily mitigated these risks.

Cloud service infrastructures and their surrounding environments feature constant change and high dynamism due to the various technological, organizational, and environmental conditions, which make it challenging for cloud service providers to mitigate cloud service risks. From a technological perspective, a cloud service infrastructure exhibits dynamic characteristics, such as high scalability and on-demand access, which require a cloud infrastructure that expands dynamically to provide sufficient resources to

meet load deviations (Grozev & Buyya, 2014). To guarantee service quality, cloud infrastructures currently rely on dynamic resource reassignments and workload transfers through interconnected cloud systems from different cloud data centers (i.e., cloud federations, hybrid or multi clouds) (Buyya, Ranjan, & Calheiros, 2010; Grozev & Buyya, 2014). Likewise, cloud infrastructures feature fast technology lifecycles and ongoing architectural changes due to inherent architectural patterns (i.e., decoupling of systems and micro services) and agile software development (Venters & Whitley, 2012; Weinhardt et al., 2009). For example, PaaS and IaaS technologies offer an adequate sandbox for cloud developers to quickly experiment with alternative options and ideas and to continuously integrate and deploy new software (fragments) (Wei & Blake, 2010). Similarly, cloud computing has exploited and spurred increasing decoupling and popularized the reuse of third-party services (Benlian et al., 2018; Tan, Fan, Ghoneim, Hossain, & Dustdar, 2016). Cloud applications use functions and resources from remote (micro) services (e.g., to retrieve geolocation information from Google maps) and, thereby, allow for flexible, recombined services. Consequently, substantial changes in the service infrastructure or even in a company's strategic thrust now occur more often in cloud service contexts.

From an organizational perspective, cloud service providers operate in complex and dynamic ecosystems and supply chains that involve various stakeholders with responsibilities that quickly change in regard to, for instance, processing customers' data (Felici, Koulouris, & Pearson, 2013; Hentschel et al., 2018; Ngo, Demchenko, & Laat, 2012). Cloud providers might quickly change how they operate business processes during their day-to-day operations despite initial process specifications. Cloud computing and IT environment changes, such as the emergence of new vulnerabilities, require cloud providers to adapt their organizational processes and train their employees accordingly; otherwise, major security incidents or harmful vulnerabilities may threaten the cloud service.

Cloud services also constantly change due to the surrounding environment (Lins, Schneider, Benlian, & Sunyaev, 2017; Schneider & Sunyaev, 2016). For example, new market entries, competitive pressures, and steady changes in consumers' preferences require cloud service providers to sense and respond to emerging environmental changes quickly and affect providers' strategies, structures, and day-to-day operations (Hentschel et al., 2018; Lee, Sambamurthy, Lim, & Wei, 2015; Schneider, Wollersheim, Krcmar, & Sunyaev, 2018). In addition, cloud services have highly dynamic legal and regulatory landscapes (Lins, Grochol, Schneider, & Sunyaev, 2016a). Governments continually adjust existing laws and propose new ones to address the challenges resulting from society's digital transformation and IT growth. For example, the E.U. General Data Protection Regulation (GDPR) focuses on giving control back to citizens and residents over their personal data and, thus, imposes new requirements on cloud service operations and demands that cloud service providers adjust their cloud infrastructure and organizational processes.

These technological, organizational, and environmental changes pose major challenges for cloud service providers in ensuring their services' ongoing security and privacy and call for innovative strategies to prove and communicate ongoing, adequate risk prevention for cloud customers.

2.2 Continuous Certification of Cloud Services

A common strategy (and a particularly important one for small- and medium-sized cloud service providers) to reduce customers' security, privacy, and reliability uncertainty and to signal trustworthiness and adequate risk prevention involves acquiring certifications (Khan & Malluhi, 2013; Lins & Sunyaev, 2017; Sunyaev & Schneider, 2013). A certification refers to a third-party's verification that products, processes, systems, or persons conform to specified criteria (International Organization for Standardization, 2004). During a certification process, certification authorities employ provider-independent and accredited auditors to perform comprehensive, manual checks to assess adherence according to a defined set of certification criteria (Lansing, Benlian, & Sunyaev, 2018)[1]. If a provider adheres to specified criteria, then a certification authority awards a formal written certificate and providers can embed a graphical Web assurance seal on their websites. Cloud service certifications typically focus on ensuring cloud services' availability, integrity, and confidentiality for a one- to three-year validity period (Lins et al., 2018; Schneider, Lansing, Gao, & Sunyaev, 2014).

[1] In this study, we focus on certification authorities who collect and analyze certification-relevant data for clarity. However, certification authorities typically employ (independent) auditors to perform certification-relevant activities (i.e., data collection and analysis) in practice. Therefore, auditors may actively participate in the CSC process.

However, existing cloud service certifications represent only a retrospective view about whether cloud providers fulfill technical and organizational measures when the certification authorities issue the certifications (Lins et al., 2016a; Schneider et al., 2014). Typically, certification authorities evaluate adherence to certification criteria during a comprehensive certification audit that they perform once. Throughout the one- to three-year validity period, certification authorities may not detect certification deviations or breaches until long after their occurrence because they validate certification adherence via sport checks only during annual surveillance audits. Thereafter, conventional certifications cannot support dynamic changes in cloud infrastructures' structure, deployment, or configuration, such as when data and software dynamically migrate across different computational nodes in cloud federations (Krotsiani et al., 2015). In addition, a cloud service provider may deliberately stop adhering to security and privacy requirements to achieve benefits (e.g., reducing required incident response staff to save costs) (Lins et al., 2017, 2018). Considering the highly dynamic cloud infrastructure that results from the aforementioned technological, organizational, and environmental changes, long validity periods may cause cloud customers to question certifications' reliability and trustworthiness and, ultimately, threaten cloud certification's ability to prove adequately prevent risk.

To address the juxtaposition of static certifications in a dynamic cloud service environment, researchers have started to investigate how to innovate the process of certifying cloud services (Windhorst & Sunyaev, 2013). These research efforts have resulted in innovative architectural and processual specifications and processes and CSC prototypes that allow certification authorities to continuously certify cloud services. A CSC monitoring system includes automated monitoring and auditing techniques and mechanisms to transparently provide certification-relevant information to continuously monitor whether cloud services adhere to certification criteria (Lins et al., 2016a). The process to do so includes four major dimensions: 1) collecting and transmitting data in a (semi-)automated manner, 2) analyzing data in a (semi-)automated manner, 3) presenting up-to-date certification information, and 4) adjusting processes (see Figure 1 and Lins et al., 2016a).

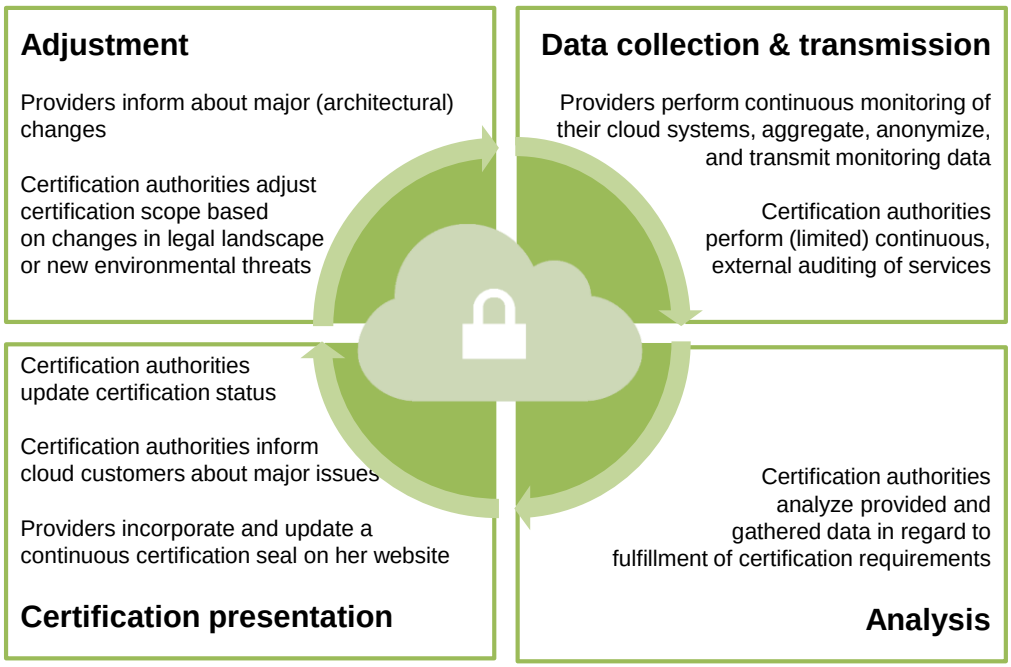

Figure 1. Continuous Certification Process (Adapted from Lins et al., 2016a)

In contrast to annual surveillance audits of conventional certifications, by collecting and analyzing certification-relevant data in a (semi-)automated manner, certification authorities can actively detect and investigate critical defects as they occur, immediately react to changes or events concerning a cloud service, and adjust their certification reports based on continuously assessing these defects, changes, and events. By continuously detecting and assuring that cloud providers actually adhere to certifications in highly dynamic cloud service environments, CSC can improve consumers' trust in the certifications that certification authorities issue. In cases when environmental threats emerge (i.e., new software vulnerabilities) or the regulatory landscape changes (i.e., the E.U. GDPR), certification authorities that

perform CSC can adjust their auditing scope by, for example, checking whether a cloud provider adheres to new certification criteria, whereas they can conventionally only change the certification's scope after the validity period has ended. Likewise, cloud providers and certification authorities can adjust their CSC-monitoring and -auditing processes to cope with architectural changes to cloud services, such as new hardware components or service functionalities. Consequently, in contrast to conventional certification, CSC considers the actual status quo of the cloud infrastructure when assessing whether cloud service providers adhere to certifications and, ultimately, informs cloud customers on both infrastructure improvements (i.e., better service quality) or failures (i.e., data losses) through a transparent and up-to-date certification representation.

Because CSC represents a promising strategy to address the challenges in conventional certification in dynamic cloud service contexts, an increasing amount of research has focused on analyzing how to certify cloud services on an ongoing and automated basis, which emphasizes the need for interminably secure and reliable cloud services. In particular, research on CSC analyzes the need and reasons for CSC (Lins et al., 2016a; Stephanow & Gall, 2015), examines the theoretical rationale underlying CSC to understand it (Lins et al., 2017), and discusses factors that influence stakeholders to participate in CSC (Lins et al., 2016b; Quinting, Lins, Szefer, & Sunyaev, 2017; Teigeler et al., 2018). More importantly, related research discusses how to perform CSC, which we discuss in Section 2.3 in detail.

2.3 Related Research on How to Perform Continuous Certification

As Table 1 summarizes, prior research has mainly focused on developing conceptual CSC architectures and processes and, more importantly, on analyzing two distinct but complementary types of CSC: test-based CSC and monitoring-based CSC (Anisetti, Ardagna, Damiani, & Gaudenzi, 2017; Kunz & Stephanow, 2017; Lins et al., 2018; Stephanow et al., 2016b).

Table 1. Overview on How to Perform Continuous Certification

Literature stream	Description	Example sources
Architectures and processes	Conceptual architectures and processes depicting how to perform CSC.	Anisetti et al. (2017), Krotsiani et al. (2015), Kunz & Stephanow (2017), Lins et al. (2018), Stephanow et al. (2016b)
Test-based CSC	Certification authorities access the cloud service infrastructure and test cloud service components directly.	Anisetti et al. (2017), Ardagna et al. (2018), Katopodis, Spanoudakis, & Mahbub (2014), Lins, Thiebes, Schneider, & Sunyaev (2015), Stephanow & Banse (2017), Stephanow, Srivastava, & Schütte (2016a)
Monitoring-based CSC	Cloud service providers monitor their service infrastructure to collect and provide certification-relevant information to certification authorities.	Krotsiani et al. (2015), Krotsiani (2016), Stephanow & Fallenbeck (2015) **This study**

Certification authorities operate test-based CSC methodologies to collect certification-relevant information by directly accessing the cloud service infrastructure and testing cloud service components (Stephanow & Banse, 2017). Typically, test-based certification techniques produce evidence by controlling some input to the cloud service and evaluating the output, such as calling a cloud service's RESTful API and comparing responses with expected results (Kunz & Stephanow, 2017). Prior research has shown that certification authorities can apply test-based CSC to verify the integrity of multiple cloud users' data (Wang, Li, & Li, 2014), assess data location (Doelitzscher et al., 2012a), validate adherence to security criteria (Stephanow & Khajehmoogahi, 2017), and so on.

In contrast, monitoring-based certification techniques use monitoring data as evidence collected from service-delivery components when they execute the cloud service (Kunz & Stephanow, 2017). Implementing a CSC monitoring system enables cloud service providers to (continuously) monitor their cloud infrastructures and collect and then transmit certification-relevant information to certification authorities (Lins et al., 2018). For example, researchers developed a prototypical monitoring-based CSC infrastructure (called *"CUMULUS"*) to, for instance, verify database user identification to validate certification criteria (Krotsiani et al., 2015; Krotsiani, 2016). Likewise, prior research has shown that certification authorities can use various monitoring metrics and key performance indicators (e.g., availability and resource management indicators and hypervisor security metrics) for monitoring-based CSC purposes (Stephanow & Fallenbeck, 2015).

While test- and monitoring-based CSC methodologies complement each other because certification authorities can use them in parallel to collect diverse evidence about certification adherence, both CSC methodologies have advantages and disadvantages. First, certification authorities produce evidence in test-based CSC via independent tests, whereas monitoring-based CSC bears the risk that cloud service providers may manipulate data to assure certification criteria adherence (Stephanow & Banse, 2017). Second, in contrast to monitoring-based CSC, test-based methodologies are invasive because they require access to the cloud service components in contrast to monitoring-based CSC (Kunz & Stephanow, 2017). Technical limitations and barriers may hinder certification authorities from collecting necessary information in test-based CSC (e.g., because integrating additional auditing systems and accessing software interfaces require extensive modifications to cloud systems, which can be quite expensive to implement) (Lins et al., 2018). More importantly, most service providers do not want and have no obligation to integrate certification authorities' test-based techniques into their systems. Indeed, some providers may even resist doing so. Third, efficiently collecting test-based data requires extensive knowledge about organizational processes, structures, and cloud service architecture unlike monitoring-based CSC. However, certification authorities have limited knowledge about specific cloud infrastructures and processes since they operate independently from providers (Lins et al., 2018). Finally, performing test-based CSC requires certification authorities to configure and adjust applied testing techniques in accordance with the individual cloud infrastructure and respective heterogonous data formats. Thus, in cases of highly dynamic cloud service infrastructures, certification authorities need to adjust applied test-based methodologies constantly, which ultimately leads to high operation costs and expenditures and limits their ability to practically apply test-based CSC. In contrast, when performing monitoring-based CSC, cloud service providers can independently alter their cloud infrastructure when ensuring that they transmit certification-relevant data to certification authorities according to their needs.

Despite the benefits associated with monitoring-based CSC, previous research has mostly focused on achieving and applying test-based CSC (see Table 1). While prior research has proven the feasibility of monitoring-based CSC by developing a prototype (Krotsiani et al., 2015; Krotsiani, 2016) and provided recommendations for items to monitor (Stephanow & Fallenbeck, 2015), we need to more deeply understand how to design CSC monitoring systems to address the risk of data manipulation and the challenging interplay between certification authorities and cloud service providers. To address this research gap, we apply a design science research approach and derive meta-requirements and design guidelines for CSC monitoring systems.

3 Research Approach

3.1 Design Science Research

Design science research involves both creating new knowledge via designing novel IT artifacts and evaluating how one use them or how the artifacts perform. It also involves reflection and abstraction to improve and understand how the artifacts behave (Hevner, March, Park, & Ram, 2004; Vaishnavi & Kuechler, 2015). In the design-creation process, one conducts a sequence of activities to produce "something new"—a novel IT artifact. In the evaluation process, the created IT artifact undergoes an evaluation to produce feedback and generate new knowledge about the problem (Beck, Weber, & Gregory, 2013). Design science offers a rigorous and meaningful contribution to practice and theory in the form of the design knowledge, an IT artifact, and its evaluation.

We currently lack a comprehensive (theoretical) understanding about how to design CSC monitoring systems to perform monitoring-based CSC because the innovative idea of continuous cloud service certification has recently gained research importance and remains underexplored, test marketed, and evaluated only in trials (Stephanow & Fallenbeck, 2015; Teigeler et al., 2018). We address this lack of design knowledge by following Kuechler and Vaishnavi's (2008) design science approach and deriving universal requirements and design guidelines for designing CSC monitoring systems (refer to Figure 2).

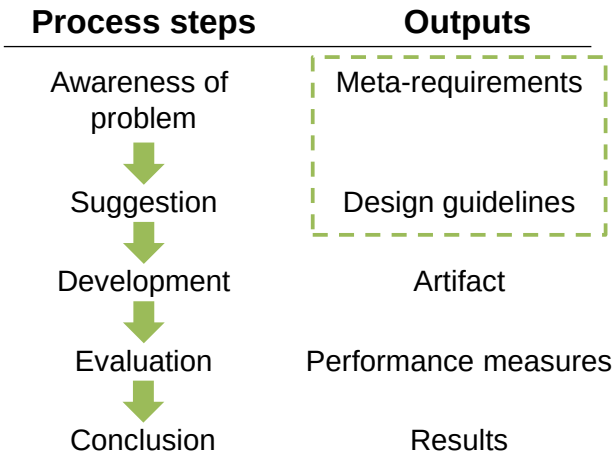

Figure 2. Design Research Cycle (Adapted from Kuechler & Vaishnavi, 2008)[2]

According to Kuechler and Vaishnavi's (2008) model, design science research starts with problem awareness, which an initial proposal that depicts a problem that one has to solve reflects. The next phase, the suggestion phase, tests whether one can convert the formulated proposal into a tentative design. One subsequently develops and evaluates the IT artifact and draws conclusions from it due to the problem-solving process.

In this study, we focus on the first two phases: problem awareness and suggestion. One needs to understand the problem to design a satisfactory IT artifact (Amrou & Böhmann, 2016; Beck et al., 2013). Thereafter, in design science research, one needs to derive well-defined requirements and design guidelines; they ensure an IT artifact's relevance to a real-world problem and provide the basis for creating and evaluating it (Amrou & Böhmann, 2016; Beck et al., 2013; Hevner et al., 2004; Pries-Heje & Baskerville, 2008; Walls, Widmeyer, & Sawy, 1992). Without clearly defined requirements, the IT artifact will not be useful and cannot offer a satisfying solution (Albert, Goes, & Gupta, 2004; Beck et al., 2013; Markus, Majchrzak, & Gasser, 2002). Based on this study's findings, future research can develop and evaluate a monitoring technology to perform continuous cloud service certification to complete the design cycle.

3.2 Data Collection

We applied an explorative and inductive design science approach as Beck, Weber, and Gregory (2013) and Kuechler, Park, and Vaishnavi (2009) suggest to collect the necessary data to derive rich meta-requirements and design guidelines for the monitoring-based CSC phenomenon. We complement and specify this design science approach with grounded theory techniques, such as theoretical sampling and a constant comparative technique (Beck et al., 2013; Kuechler et al., 2009). More specifically, we applied techniques of the "Straussian" version of grounded theory (Corbin & Strauss, 2015) because this version provides unambiguous process guidance and IS research widely adopts it (Urquhart, Lehmann, & Myers, 2010; Wiesche, Jurisch, Yetton, & Krcmar, 2017). We applied an inductive approach so we could investigate complex circumstances that the literature has not yet explored (Kuechler et al., 2009). In addition, research has shown grounded theory approaches to be useful when addressing new or poorly understood phenomena (Smolander, Rossi, & Purao, 2008), to allow theoretical concepts that have close ties to the phenomena under observation to emerge, and to allow one to consider the contexts that embed those phenomena (Volkoff, Strong, & Elmes, 2005).

Theoretical sampling guided the data-collection process; that is, the concepts that emerged from data analysis greatly influenced the respondents we chose and the questions we posed (Abraham, Boudreau, Junglas, & Watson, 2013; Corbin & Strauss, 2015). Thus, insights that we gained from initially collecting and analyzing data guided how we subsequently collected and analyzed data. Theoretical sampling

[2] Note: dashed outline summarizes this study's outputs.

enables researchers to discover the concepts in terms of their properties and dimensions that pertain to a problem and to uncover variations and identify relationships between concepts (Corbin & Strauss, 2015).

To collect data to better understand the problem domain and derive meta-requirements and design guidelines, we conducted five focus group interviews with certification authorities, cloud service providers, and consultants and ten one-to-one interviews with cloud customers (refer to Table 2). Because we used an inductive approach, we did not select theories to test or incorporate prior to collecting and analyzing data (Abraham et al., 2013).

Table 2. Summary of Conducted Interviews

No.	Interview type	Focus	Participants	Sampling reason
1	Focus group	Objectives and scope of CSC	Four cloud service providers One cloud service consultant	Starting point
2	Focus group	How to collect and provide information	Four cloud service providers Four certification authorities Two cloud service consultants	Include certification authority representatives
3	Focus group	How to collect and provide information	Six cloud service providers Seven certification authorities Two cloud service consultants	Joint discussion between providers and authority representatives
4	One-to-one interviews	Identify customer requirements	Ten cloud service customers	Consider customer perspective
5	Focus group	Implementing monitoring-based CSC	Six cloud service providers	Validate applicability from provider perspective
6	Focus group	Using monitoring results in certification processes	Five certification authorities	Validate applicability from certification authority perspective

By conducting focus group interviews, one can collect viewpoints about a certain defined topic of interest from a group of people who have certain experiences (Myers, 2013). Focus groups also enable participants to engage in thoughtful discussions and generate practical and extensive data. We conducted the first focus group interview in November, 2014, with four cloud service providers and one cloud service consultant. In this first interview, we discussed the objectives and scope of performing monitoring-based CSC. However, discussions and data analyses revealed that we needed to more deeply understand certification authorities' needs and perspectives on monitoring-based CSC. Therefore, we invited representatives of certification authorities to join the discussions in the second and third focus group interviews, which we held in December, 2014, and April, 2015. Subsequent data analyses revealed that monitoring-based CSC may also affect cloud customers to a certain degree (e.g., certification authorities may analyze customers' data during certification processes). Thus, we conducted 10 semi-structured interviews with cloud service customers between June and July, 2015, to identify requirements that customers imposed on performing monitoring-based CSC. Finally, we conducted a focus group interview in January, 2017, with six cloud service providers first and then with five certification authority representatives to discuss our data analysis findings and open issues and reach theoretical saturation.

During these interviews, the participants actively discussed the CSC concept and reflected their individual use cases. In total, 16 cloud service providers, 12 certification authority representatives, five cloud service consultants, and 10 cloud service customers participated (refer to Appendix A for information on the interviewees). The cloud service providers operated on national and global scales to provide infrastructure, platform, and software cloud services. The size of each provider ranged from medium- to large-sized enterprises. The certification authority representatives had years of experience in cloud services, infrastructure, data security, and protection certification audits. Large certification or auditing organizations employed them or they worked as independent auditors. The participating consultants advised providers about their decision to obtain certification. The cloud customers, all IT managers, came from medium- to large-sized enterprises and different sectors, including IT, health, and finance. The practitioners who participated in our interviews had not adopted CSC but expressed interest in or advocated for it. We distributed the interview partners to gain as much insight as possible and triangulate data from different perspectives (i.e., certification authorities and cloud service providers) as previous work recommends (Patton, 2015). This highly diverse group of practitioners helped us understand the problem domain and establish various meta-requirements to design CSC monitoring systems. Each focus group interview lasted four hours on average. Each customer interview lasted 53 minutes on average.

We conducted each interview, which we recorded, based on an interview guide (Yin, 2014). The interview guide kept the interactions focused while allowing individual experiences to emerge during the limited interview period (Gorden, 1980). Therefore, the interview guide served as a reminder of the information that we needed to collect (Yin, 2014). After each interview, we adapted the interview guide if new concepts emerged and to validate prior findings following the theoretical-sampling process. We asked interview partners questions about potential CSC use cases, CSC's scope, potential architectures and processes to provide certification-relevant information, and CSC's risks and limitations (refer to Appendix B for example interview questions). Although we have conducted research regarding CSC for several years, we applied a nonjudgmental form of listening (Walsham, 1995), maintained distance (Patton, 2015), and strived to maintain an open and non-directive style of conversation during the interviews to ensure impartiality as required when applying grounded theory techniques to avoid bias in theory development (Heath, 2006). Finally, we followed established methodological guidelines to collect and analyze data (see Appendix C for summary).

3.3 Design Creation and Evaluation

Building on our iterative data-collection and theoretical-sampling approach, we analyzed the initial data we collected in 2014 to derive meta-requirements (see Section 4.1 for details on how we analyzed the data to derive meta-requirements and Section 5.1 for design guidelines, respectively). We then used the sequenced interviews to constantly evaluate our derived meta-requirements and design guidelines. Therefore, our design science process occurred in multiple iterations of meta-requirement and design guideline derivation and refinement to generate a design that fully satisfies the researchers and practitioners who subsequently use it (Beck et al., 2013; Hevner et al., 2004). Thus, the entire design process involves much repetition since, in each step, we repeated and improved on previous steps (as Figure 3 depicts).

Specifically, we analyzed the data that we collected in the first and second focus group interviews in 2014 to derive an initial set of meta-requirements and corresponding design guidelines, which resulted in 13 meta-requirements and 15 design guidelines. To evaluate our initial findings, we presented and discussed this set with practitioners during the third focus group interview in 2015. Then, we refined the meta-requirements and design guidelines accordingly and also identified one additional meta-requirement and design guideline. As a second evaluation, we then validated these results during the one-to-one interviews with cloud customers in 2015 based on which we further refined our previous findings. Finally, in the last evaluation phase, we discussed the derived design concepts of CSC monitoring systems and jointly assessed their suitability and appropriateness when continuously assessing whether cloud service operations adhere to certification criteria adherence during both focus groups in 2017. Appendix D summarizes how we derived and refined the meta-requirements and design guidelines.

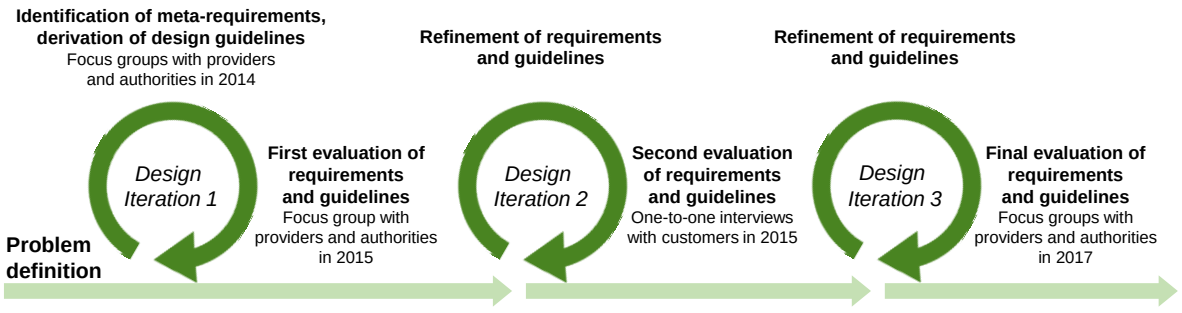

Figure 3. Evaluation Process of the Meta-requirements and Design Guidelines

4 Problem Awareness Phase: Deriving Meta-requirements

4.1 Research Steps to Derive Meta-requirements

As we state in Section 3.1, design science research starts with problem awareness, which an initial proposal that depicts a problem that one has to solve reflects (Beck et al., 2013; Kuechler & Vaishnavi, 2008). Although the innovative idea of continuous cloud service certification has recently gained research awareness, the majority of research has focused on developing test-based CSC methodologies, and we

lack comprehensive design knowledge about how to perform monitoring-based CSC. To better understand different stakeholders' needs (i.e., cloud service providers and certification authorities) and the issues related to performing monitoring-based CSC, we derived and constantly refined meta-requirements that can help cloud service providers design CSC monitoring systems. Meta-requirements define what an IT artifact is for and typically describe its goals, scope, and boundaries (Gregor & Jones, 2007). Thus, identifying meta-requirements helps one to understand how to design and implement CSC monitoring systems and determine the challenges that design guidelines need to meet in the suggestion phase. Meta-requirements do not refer to the requirements for one instance of an IT artifact (i.e., a specific CSC monitoring system) as would be the case if industry needed to build a single system. Instead, design science research focuses on deriving requirements that pertain to a whole class of IT artifacts. Consequently, meta-requirements provide insights and knowledge for upcoming design science research projects.

Following the concept of theoretical sampling, we began analyzing data to derive meta-requirements after we collected our first data in 2014 during the focus group with cloud service providers (Corbin & Strauss, 2015). We used the coding paradigm typical of the "Straussian" version of grounded theory to analyze our data (Corbin & Strauss, 2015; Wiesche et al., 2017). Accordingly, in analyzing the data, we conducted open, axial, and selective coding (with each type at a higher, more abstract data-analysis level than the preceding one) (Abraham et al., 2013; Corbin & Strauss, 2015). Coding refers to a process in which one annotates and labels interview transcripts with a piece of text (Jones & Hughes, 2001). To determine the labels, we used words that the interviews suggested (Volkoff, Strong, & Elmes, 2007). We used Atlas.ti 7, a tool that helps one code qualitative data, to facilitate this process.

4.1.1 Open Coding

Open coding entails fracturing the data by describing concepts in it that may define a significant occurrence or incident about a phenomenon (Abraham et al., 2013; Corbin & Strauss, 2015). Through this type of coding, we created 94 codes related to 438 textual segments that we obtained from the 15 interviews. As an example of a coded textual segment, we highlighted a portion of a customer interview (i.e., "For me, it is obvious that no customer data are checked") and coded it as "data protection during monitoring".

4.1.2 Axial Coding

Axial coding involves coding for causes, actions-interactions, and consequences (Corbin & Strauss, 2015; Strong & Volkoff, 2010). Causes answer questions about why, when, and how come and, thus, refer to the perceived reasons that persons give for why things happen. Action-interaction refers to the actual responses people or groups make to the events or problematic situations that occur. Consequences refer to anticipated or actual outcomes of action and interaction. Thus, with axial coding, we could understand what requirements stakeholders impose when designing CSC monitoring systems, why they do so, and the consequences when stakeholders do or do not adhere to these requirements. Through axial coding, we created 23 codes that relate to text segments that describe causes for requirements and 13 codes that correspond to text segments that depict consequences from (not) adhering to requirements, while the majority of previous open codes relate to actions-interactions. After understanding causes, actions, and consequences, axial coding enables one to compare codes to classify them under common themes and, thus, create hierarchical classifications (Abraham et al., 2013; Corbin & Strauss, 2015). For example, we combined the code "availability criterion" and "access management criterion" to form the theme "security criteria". Later on, we grouped the themes "security criteria" and "organizational criteria" as "certification criteria" to classify data into larger categories. By comparing codes, we created 14 categories on the highest level that form our meta-requirements and that we present in the Section 4.2.

4.1.3 Selective Coding

During selective coding, we integrated all findings into one "core category"; that is, we formulated a story line that coherent conceptualized the main phenomenon. By doing so, we could move beyond description to a more abstract conceptualization level (Urquhart et al., 2010). We turned to the literature to identify a suitable core category. Because we focused on designing CSC monitoring systems, we relied on system architectural patterns to structure our meta-requirements. Architectural patterns capture the essence of an architecture that different software systems have used (Sommerville, 2012). Traditional monitoring system architectures use a layered client-server architectural pattern (refer to Figure 4) (Fatema, Emeakaroha,

Healy, Morrison, & Lynn, 2014). The layered architecture pattern organizes the system into layers with related functionality associated with each layer (Sommerville, 2012).

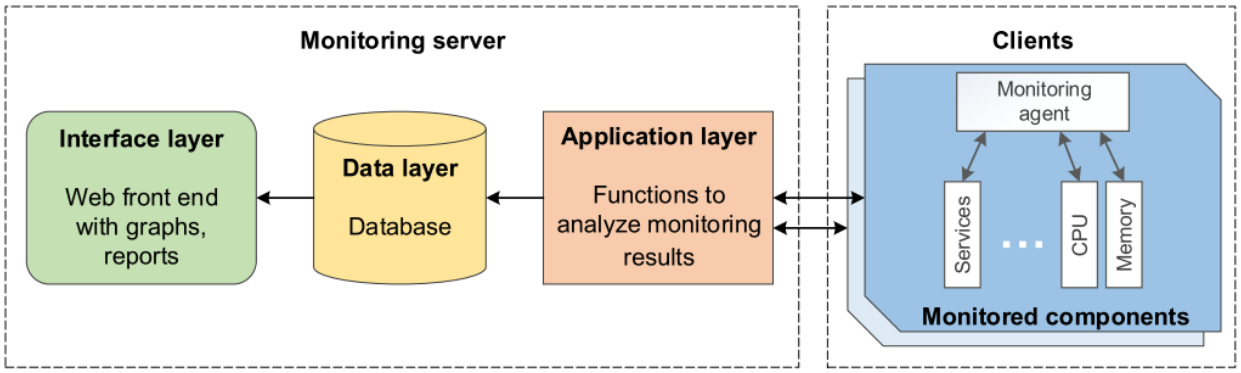

Figure 4. Traditional Monitoring System Architecture (Adapted from Fatema et al., 2014)

The clients refer to cloud hardware and software components (i.e., storage servers and virtual machines). Monitoring agents are deployed from a monitoring server on these components to collect relevant monitoring data. The monitoring server typically embeds application, data, and interface layers. On the application layer, the monitoring system manages agent deployment, analyzes the collected data and metrics, and stores the results in a database on the database layer. On the interface layer of a monitoring system, a Web frontend displays the stored data or provides administrators with the means to generate graphs or service level agreements reports.

In comparing our interview findings with the layered client-server architectural pattern from the literature, we found a means to organize our results in a conceptually meaningful way. Thus, we assigned our derived meta-requirements to related architectural layers. In addition, we identified non-functional requirements that serve as constraints on the system's design across the different layers. A non-functional requirement refers to a requirement that specifies criteria that one can use to judge how a system operates rather than specific behaviors (Sommerville, 2012).

To guide our data-analysis approach, we applied constant comparison (Corbin & Strauss, 2015; Wiesche et al., 2017). Constant comparison focuses on assessing whether the data support and continue to support emerging categories (Bryant & Charmaz, 2007). Thus, we compared any unit of data with another unit to ensure that we grounded the discovery in rigorous coding and systematic procedures. First, we constantly compared who said what because we interviewed practitioners from different domains, including cloud service providers, certification authorities and cloud customers. Furthermore, we compared the individual statements that relate to a certain category to identify their properties and dimensions. For instance, one meta-requirement demands that CSC monitoring systems need to aggregate monitoring data. While comparing statements on data aggregation, we identified several aggregation dimensions, including aggregating data across monitoring systems, plugins and tools, and cloud data centers. More importantly, we compared findings from each interview with subsequent interviews to triangulate our data. For example, cloud service providers reported that they would integrate third-party hardware to enable CSC during the first focus group interview. However, interviewees in the second and third focus groups expressed resistance due to high security concerns when integrating third-party hardware. We compared the corresponding statements to understand the reasons for and consequences of the contrasting opinions and, thus, deepened our understanding. Appendix D summarizes the emergence of meta-requirements, causes, and consequences and how we iteratively refined meta-requirements throughout our research approach given the iterative nature in which we collected data and applied constant comparison.

Finally, we applied memoing; that is, a technique that one can use to note theoretical ideas during data analysis and communicate insights from the data analysis (Gasson & Waters, 2013). Memoing ensures that one does not forget or leave underdeveloped novel insights triggered during the long process of collecting and analyzing the data (Wiesche et al., 2017). We opted for a relatively fluid memoing scheme (Abraham et al., 2013) and noted insights that emerged while we coded data. For example, we took notes while assessing data-aggregation statements because it became apparent that we should conduct data aggregation from a service perspective.

4.2 Meta-Requirements for Continuous Service Certification Monitoring Systems

In this section, we present meta-requirements for designing CSC monitoring systems. Table 3 summarizes the elicited requirements. While we present the meta-requirements in isolation, they depend on and might even complement one another. For example, achieving a high degree of monitoring automation (relating to REQ-NF1) facilitates traceability of operations (relating to REQ-NF3) because a CSC monitoring system can easily record the execution of automated operations in log files.

Table 3. Elicited Meta-requirements

Req-ID	Requirement	Description
Data-gathering layer (client)		
DGL1	Gather relevant data for cloud service certifications	Gather evidence about data availability, confidentiality, integrity, and so on.
DGL2	Gather data by relying on existing monitoring technologies	Leverage existing monitoring technologies to gather relevant data.
DGL3	Consider monitoring the cloud supply chain	Evaluate whether providers can gather additional information about their subproviders' performance.
DGL4	Ensure data confidentiality during data collection	Determine which monitoring data the certification authorities require and whether collecting these data causes a conflict with any privacy, legal, or company regulations.
Application layer (server)		
AL1	Enable data aggregation	Aggregate low-level monitoring data of individual resources to construct meaningful and significant status indicators of respective cloud services.
AL2	Ensure data protection	Ensure protection of monitoring data to prevent leakage of sensitive or security-relevant information.
AL3	Enable data filtering	Filter monitoring data according to authorities' needs.
Data layer (server)		
DL1	Enable data archiving	Archive collected and processed monitoring data for certain periods.
DL2	Ensure data integrity	Prevent manipulation of monitoring data.
Interface layer (server)		
IL1	Provide certification-relevant data	Enable flawless and continuous data exchange with certification authorities.
IL2	Ensure data security when exchanging data	Implement appropriate security mechanisms to safeguard data exchange.
Non-functional requirements for client and server layers		
NF1	Achieve a high degree of automation	Perform monitoring actions with a high degree of automation.
NF2	Achieve a high degree of adaptability	Achieve a high degree of adaptability that involves the ability to adjust to new conditions or change due to a new use or purpose.
NF3	Enable traceability of operations	Inform about how and when certification-relevant data were gathered and analyzed.

4.2.1 Client: Data-Gathering Layer

Gather relevant data for cloud service certifications (REQ-DGL1): when performing CSC, certification authorities require diverse data sets to evaluate adherence to certification criteria. Prior to designing a CSC monitoring system, a cloud service provider and certification authority must specify a set of certification criteria that they will constantly check because cloud service providers cannot provide (monitoring) data to assess every certification criterion. As one cloud service provider said: "There are no hard and fast rules about which criterion can be monitored. Availability for example. Availability can be assessed automatically". Cloud service (infrastructures) are highly heterogenic; therefore, authorities will differ in the extent to which they focus on the items that they should monitor and audit. As one certification authority representative said: "We need to identify the lowest common denominator of criteria that can be checked".

The most frequently demanded criterion during our interviews involved ensuring service availability; therefore, a CSC monitoring system must gather data on cloud service availability, maintaining system redundancy, and operating data recovery mechanisms. As one cloud service customer said:

> *If I outsource my data to the cloud, I need to be sure that I can access my data at any time. Service availability plays a major role because, with availability, I ensure that my business is able to work*.

The second most important criterion involved ensuring that a cloud service provider processes and stores data in locations that comply with legal requirements. As one cloud service customer said:

> *For us, it is important that data that we outsource to the cloud will stay in [country X]. We have made some special agreements with our cloud service providers to ensure that they do not set up a new data center in [country Y] to save costs. I imagine that a continuous certification will validate that these conditions are still true*.

Finally, cloud experts demanded penetration testing to ensure that a certification authority identifies any potential system vulnerabilities at an early stage. As one cloud service customer said: "To be sure that you cannot be attacked from the outside. That is the most important thing for providers". Other certification criteria that certification authorities may audit during CSC include validating access management, testing the correctness of provided service functionalities, ensuring the longevity of a cloud service provider, assessing exit-management processes, and proving adherence to service-level agreements. Table 4 summarizes example certification criteria that the cloud experts discussed often. These findings concur with the findings in previous research that has analyzed certification criteria catalogues to identify criteria that require continuous validation (Lins et al., 2018).

Table 4. Certification Criteria that Should Be Monitored

Certification criteria	Description
Availability	Gather data on cloud service availability and information on maintaining system redundancy and operating data-recovery mechanisms.
Data location	Track data processing and storage locations to validate that data complies with legal requirements.
Penetration testing	Perform regular penetration tests to reveal potential system vulnerabilities.
Access management	Provide data about securing access to cloud resources, including prevention of unauthorized access to resources, limited access of administrators, and information on managing passwords and assessing and adjusting access rules.
Process operation	Verify that cloud providers operate organizational processes as defined in process specifications.
Data encryption	Analyze whether a cloud provider maintains encryption and whether such encryption achieves a high level of security. In addition, provide evidence that cloud providers operate encryption-management processes (e.g., key management processes).
Data loss	Provide information on maintaining and operating mechanisms that prevent (customer) data loss.
Data handling	Monitor who has accessed and modified data.
Continuity management	Verify that cloud providers operate continuity-management processes as defined in process specifications.
Exit management	Analyze whether data-exports mechanisms remain in place to ensure data portability in case of switching the cloud service provider.
Functionality	Assess that offered functions work properly.
Patch management	Provide evidence that cloud providers patch software and systems regularly.

Gather data by relying on existing monitoring technologies (REQ-DGL2): to fulfill a certification authority's data needs, cloud service providers have to gather most certification-relevant monitoring information themselves and subsequently make these data accessible because external certification authorities lack comprehensive access to a provider's systems. As one certification authority representative said: "The monitoring is under control of the cloud service provider. In regard to data

provisioning and what is gathered and aggregated, it resides at the provider's side". To prevent multifarious cloud risks, providers continuously monitor their service infrastructures and equip their data centers with sophisticated monitoring technologies that can rapidly detect malicious behavior, failures, and outages. As two separate cloud service providers said:

> *It is called [monitoring tool X]. With this tool, you can simulate application interactions, for example, an ordering process in the online portal.*
>
> *Yes, we check on which location the virtual machines of our customers are currently running*.

Leveraging these monitoring technologies for CSC can provide certification authorities with detailed insights about service operation that surpass what they can measure themselves. Available data sets and required data must align with each other, which may mean that a cloud service provider needs to extend monitoring technologies' current data-collection capabilities or implement additional data-collection mechanisms. Then, a CSC monitoring system should gather relevant data from existing monitoring technologies.

Consider monitoring the cloud supply chain (REQ-DGL3): cloud experts emphasized that cloud services typically rely on entangled supply chains because they integrate different subproviders. As one cloud service provider said: "We do not have the knowledge in our own company, and we, therefore, have several special purpose companies that we have to integrate into our cloud". For example, a SaaS provider may rely on a PaaS provider to offer its cloud services in an efficient and cost-effective manner. Conversely, the offered cloud service depends on the underlying platform and its quality. Thus, unsurprisingly, the interviewed cloud customers demanded CSC processes that include or at least consider integrated subproviders. One cloud service customer said: "This is indispensable. …For me as a customer, it is irrelevant whether the software of the provider or the infrastructure of the subprovider is unavailable". In return, cloud service providers doubt that one can easily integrate subproviders into the CSC process. One cloud service provider said: "If I tell them [the subproviders] they have to be certified, I really limit myself when selecting some subproviders". Thus, an entangled supply chain may limit data collection and CSC processes. Cloud service providers should evaluate whether they can gather additional information about their subproviders' performance as their customers and the certification criteria require.

Ensure data confidentiality during data collection (REQ-DGL4): when collecting certification-relevant data, cloud service providers should carefully determine the monitoring data that the certification authorities require and whether gathering these data causes a conflict with any data protection, legal, or company regulations. As one cloud service provider said: "We often have to check whether log files and monitoring files are in accordance with data protection laws. …We often have problems when arguing why we need all these data and to map data to specific purposes". Specifically, cloud experts argue that gathering data about user actions should be limited to prevent employee surveillance. As one cloud service provider said: "An administrator does not see the exact user who has generated a certain event". In addition, CSC monitoring systems should not gather and analyze customer data because revealing sensitive customer data may breach service-level agreements, which may require financial compensation. As one cloud service customer said: "For me, it is obvious that no customer data are checked but cloud service functions are examined to prove that the service is running and operates as expected". Consequently, CSC monitoring systems must ensure data confidentiality to comply with data-protection regulations.

4.2.2 Server: Application Layer

Enable data aggregation (REQ-AL1): typically, monitoring technologies gather data for each computing resource individually (e.g., information on availability for each server). However, certification authorities focus on consolidated information about general cloud service behavior. As one certification authority representative said: "[Monitoring technology X] generates 20,000, 30,000 messages per day. These messages have to be interpreted, aggregated, weighted, and then assessed". Therefore, a CSC monitoring system should aggregate monitoring data from individual resources to construct meaningful and significant status indicators about respective cloud services. Because cloud services comprise hundreds of interconnected systems that interact to create the service, data aggregation reduces a service evaluation's complexity, scope, and depth and generates high-level indicators that certification authorities can employ to perform efficient assessments. As one cloud service provider said: "A service

request goes through the login service, identification service, across the web service and application server, and to the database, creating a wealth of information".

Ensure data protection (REQ-AL2): when cloud service providers transfer data to a certification authority, they must ensure they protect the monitoring data to prevent sensitive information (e.g., personal information) from leaking. As one certification authority representative said: "Which data will be anonymized? It depends. We will analyze personal data that are associated with audit-relevant data. Yet, which employee has worked on which ticket will be obfuscated". Likewise, a CSC monitoring system must obfuscate security-relevant information that may reveal any service vulnerabilities. As one cloud service customer said: "I have no interest in communicating any service vulnerabilities that have been revealed during the certification process. Malicious attackers might misuse this information and threaten my data". Cloud experts also expressed concerns regarding blindly trusting the certification authority and the intentions of their employees as one cloud service consultant noted: "I have some problems with transmitting data because I am not able to verify that the certification authority employs trustworthy people".

Enable data filtering (REQ-AL3): in addition to the need for data aggregation, the interviews revealed that a CSC monitoring system must filter data before a cloud service provider transmits the data to a certification authority for several reasons. Certification authorities have different certification and auditing scopes and require variable amounts of information. As one certification authority representative said:

> *If the certification criteria will be checked by two different groups of auditors, for example, a criterion concerning the access control. As an IT service management auditor, I have to make a simple audit regarding access control. While somebody, for example, in Germany, a legal auditing company that has to do access control, they have to go even further…and deeper. If I do not determine the auditor group before, I might show an insufficient amount of information or too much*.

While an insufficient amount of certification-relevant information may hamper a certification authority's ability to evaluate certification adherence, an excessive amount of information may create risks for cloud service providers and certification authorities. As two separate certification authority representatives said:

> *I can imagine certain information and several metrics that you do not want to analyze because you are then exposed to liability risks*.

> *Your customers will call you crazy if you provide auditors with the means to inspect everything*.

4.2.3 Server: Data Layer

Enable data archiving (REQ-DL1): certification authorities care about comparing current data with historic data to identify any previous criteria deviations, increase CSC process traceability, or conduct trend analyses. As one cloud service provider said:

> *What does a certification tell you? On average, 90% of all tests were successful for the last year. And then, you have to store the monitoring data for the whole year. Otherwise, I am not able to provide evidence…. As a customer, I am interested in what the cloud service looked like on every day for the past year*.

Further, a cloud service consultant said: "Why should cloud service providers backup historic monitoring data? To conduct any trend analysis or identify issues".

Consequently, a CSC monitoring system should archive collected and processed monitoring data for certain periods. Archiving data may force cloud service providers to store a vast amount of data over time. As one cloud service provider said: "Access logs, change logs, and changes in management systems for 10 years retrospectively. Meanwhile, we have a remarkable amount of data stored. This should not be underestimated!".

Ensure data integrity (REQ-DL2): despite the benefits of monitoring-based CSC, a cloud service provider's providing certification-relevant data has one drawback: the risk of low data integrity due to deliberate data manipulation. Providers may modify or euphemize provided data to assure certification criteria adherence. As two cloud service providers said:

> *If I am a software developer and I am aware that the certification depends on the results that my software generates, of course, service providers or software producers are tempted to manipulate the data. In keeping with the motto, transmitting everything is okay*.
>
> *I have worked with so many log files. Everyone can easily manipulate these. Say, in case I transmit data, I inspect the log files before and check whether the data are in accordance with the certification criteria*.

Therefore, preventing cloud service providers from manipulating certification-relevant data is an important prerequisite to ensure trustworthy and reliable CSC.

4.2.4 Server: Interface Layer

Providing certification-relevant data (REQ-IL1): in addition to gathering data, providers must manage how they provide certification-relevant information to ensure ongoing data exchange with certification authorities. Service providers must establish an internal certification department that manages and supervises when the provider collects, processes, provides, and transmits certification-relevant information. A CSC monitoring system should provide respective functionalities that enable cloud service providers to flawlessly and continuously exchange data with certification authorities.

Ensure data security when exchanging data (REQ-IL2): providers face novel security- and data-protection issues when providing monitoring data to external certification authorities. Most of these issues arise due to existing monitoring technologies that cloud service providers deploy only for organizational monitoring purposes. Thus, interactions with certification authorities do not fall in the scope of traditional monitoring technologies. As one cloud service provider said:

> *That I have to provide various data to the auditor is a risk for me. In case data are leaked or not, it is different whether the auditor is in my company and I have full control versus I have to provide data automatically. It is definitely a risk for me*.

Attackers may be interested in targeting data transmissions to retrieve or modify exchanged data. Modified data may affect how a certification authority assesses criteria adherence and may cause certification non-adherence.

4.2.5 Non-functional Requirements for Client and Server Layers

Achieve a high degree of automation (REQ-NF1): performing CSC requires certification authorities to frequently collect and analyze data. To be efficient and cost-effective, CSC requires a high degree of standardization and automation. As one cloud service provider said: “Full automation. Because what is not automated won’t be done properly”. Further, a cloud service customer said: “It should be automated as manual work requires time and might result in failures”. Subsequently, CSC imposes high requirements for CSC monitoring systems, such as performing data-collection, data-analysis, and data-aggregation mechanisms automatically. Nevertheless, cloud service experts expressed concerns regarding fully automating monitoring operations for each certification criteria. As one cloud service provider said: “Automation, it depends on the [certification] criterion. What you can automatically check is availability. …You have to differentiate between what can be automated and what cannot. There are definitely things that you cannot automate”.

In addition to reducing costs and errors, a high degree of automation enables on-demand auditing that increases the trustworthiness and transparency of CSC processes. As two separate cloud service providers said:

> *Okay, I press this button, and then I see the results. This creates trust for our customers*.
>
> *I am instantly able to communicate with customers, internal departments, or whoever is interested. Everything is okay or there are some problems here and there*.

Certification authorities require on-demand auditing to assess whether a cloud service still adheres to certification criteria or when they want to initiate a (quick) separate validation due to major changes in the cloud service. As one cloud service customer said: “I imagine in regard to higher flexibility; I change the cloud infrastructure and depending on the change, I initiate an additional check”. Consequently, CSC monitoring systems may provide the functionality to collect and analyze data on demand.

Achieve a high degree of adaptability (REQ-NF2): while certification authorities and cloud service providers frequently adjust the CSC process to cope with an ever-changing environment, a CSC monitoring system should achieve a high degree of adaptability that involves the ability to adjust to new conditions or change due to a new use or purpose. Emerging environmental threats or legal and regulatory landscape changes may induce certification authorities to adjust their auditing scope by, for example, adding new certification criteria. In these cases, a CSC monitoring system should be easily extendable to enable certification authorities to monitor these new criteria. As one cloud service provider said: "New parameters can be included dynamically and then integrated into continuous monitoring operations". Architectural changes to cloud services (e.g., adding hardware components or new service functionalities) can cause providers to adjust their monitoring processes and, thus, the CSC monitoring system. As one cloud service consultant said: "I mean that [the CSC monitoring system] is a living object. In line with technological or other changes, it has to be adjusted and maintained". Ensuring a high degree of adaptability provides a basis for an up-to-date monitoring system that can address emerging challenges. As one cloud service provider said: "It is important for me that we are compliant with the current situation, security levels, and requirements".

Enable traceability of operations (REQ-NF3): the interviewed cloud experts emphasized that they needed to inform customers about how and when they gathered and analyzed certification-relevant data in addition to presenting information regarding certification adherence. One cloud service customer said: "I wish that one is able to gain insight into the certification process". Further, a cloud service consultant said:

> *Based on my experience with customers, there are two or three parties. The first party wants to join in and have a say in the matter. The others want to be left alone in line with the slogan: you should ensure that everything is okay, and I do not need to know anything else. Between these parties are endless interim stages*.

Certification authorities especially need to ensure traceability in cases where cloud service customers must prove that their outsourced IT resources are secure. As one cloud service customer said: "In the end, I also require some evidence in case of emerging questions, for example, when I'm audited by a legal auditing company". When achieving a high degree of traceability, CSC monitoring systems can increase the comprehensibility and trustworthiness of CSC processes. As one cloud service customer said: "Trust will then be increased. I do not have to solely trust that there is actually something going on. Instead, I am able to check by myself or at least see the results".

5 Suggestion Phase: Proposing Design Guidelines

5.1 Research Steps to Derive Design Guidelines

In the suggestion phase, one tests whether one can transfer the formulated meta-requirements into a tentative design or not (Beck et al., 2013; Kuechler & Vaishnavi, 2008). Thus, we derived design guidelines for CSC monitoring systems for addressing meta-requirements. Researchers typically draw suggestions for a problem solution from the problem domain's existing knowledge base (Vaishnavi & Kuechler, 2015). Grounding information artifacts' design in justificatory knowledge not only increases how well the designer understands the problem domain but also helps the designer formulate high-level design guidelines independent of technological constraints and specific implementation details (Arazy, Kumar, & Shapira, 2010). In addition, grounding a design on justificatory knowledge explains why an artifact takes the form that it does and why it works (Gregor & Jones, 2007).

Because both the environment (people, organizations, and existing technologies) and the knowledge base (Hevner et al., 2004) influence IS research, we looked into related literature, surveyed existing monitoring technologies, and analyzed interview findings to incorporate practical domain knowledge. IS theorists widely include extant literature alongside the empirical data as a means to raise their overall analyses to a higher conceptual level (Beck et al., 2013; Fernandez, 2004; Levina & Vaast, 2005). Figure 5 summarizes the knowledge base that we used as a guide to design CSC monitoring systems.

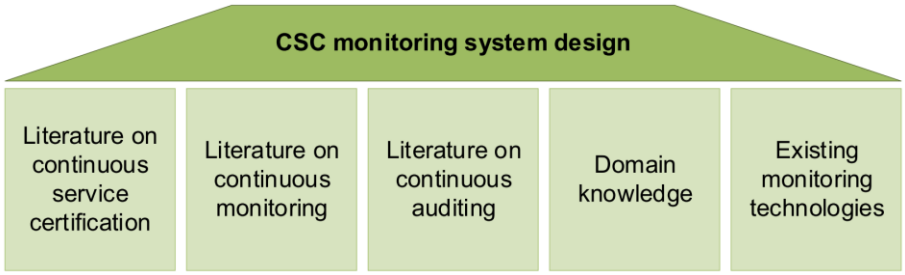

Figure 5. Knowledge Base that Guide the Design of CSC Monitoring Systems

We first reviewed literature on CSC (refer to Sections 2.2 and 2.3) and related concepts (namely, continuous monitoring and auditing) (refer to Lins et al., 2018, 2015). We define *continuous monitoring*, which service providers perform, as the ongoing observance and analysis of operational states of systems and applications to provide decision support, detect and diagnose problems, and provide information for subsequent analyses (Mell et al., 2012). We define *continuous auditing*, which certification authorities perform, as a methodology that enables independent auditors to provide written assurance on a subject matter using a series of auditors' reports issued virtually simultaneously with, or a short period of time after, the occurrence of events underlying the subject matter (Canadian Institute of Chartered Accountants, 1999). Second, we assessed a comprehensive set of monitoring technologies to gather information on their architecture, data-gathering capabilities, and interfaces to deepen our understanding about designing CSC monitoring systems (refer to Appendix E for an overview). Finally, we reexamined interview statements for each meta-requirement to exploit interviewees' practical domain knowledge when deriving design guidelines. By grounding the design approach on this knowledge base, we could derive a set of design guidelines for CSC monitoring systems that can address meta-requirements. In Section 5.2, we summarize the derived design guidelines.

5.2 Design Guidelines for CSC Monitoring Systems

Figure 6 depicts the tentative design of a CSC monitoring system architecture that embeds derived guidelines, and Appendix F summarizes the derived design guidelines. Researchers and practitioners may use this tentative design to implement an IT artifact in the future.

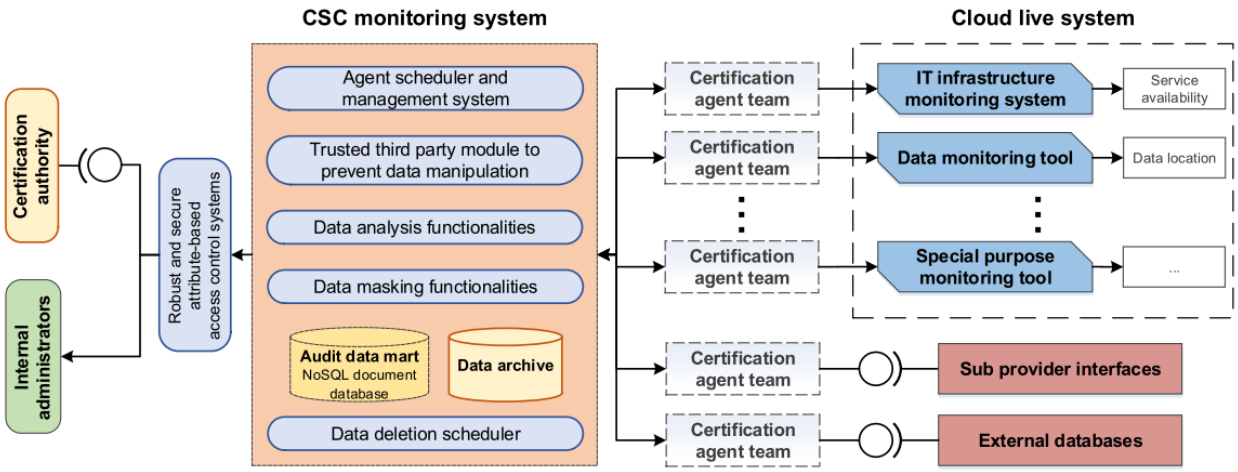

Figure 6. Tentative Design of a CSC Monitoring System

5.2.1 Leverage Existing Monitoring Technologies to Gather Certification-relevant Data

A CSC monitoring system can leverage existing monitoring technologies to gather certification-relevant data (achieving REQ-DGL2), such as 1) IT infrastructure-monitoring systems, 2) special-purpose monitoring tools, and 3) monitoring plugins (refer to Table 5).

Table 5. Design Guidelines to Leverage Existing Monitoring Technologies

ID	Design guideline	Description	Corresponding meta-requirements
DG-1	Leveraging IT infrastructure-monitoring systems	Integrate IT infrastructure monitoring systems as they typically monitor applications, services, operating systems, system metrics, and network infrastructures.	REQ-DGL1 REQ-DGL2 REQ-NF1 REQ-NF2
DG-2	Leveraging monitoring tools	Leverage monitoring tools to gain detailed insights about cloud service and process operations that surpass the capabilities of infrastructure monitoring systems.	
DG-3	Leveraging monitoring plugins	Leverage monitoring plugins to extend monitoring technologies' capabilities.	REQ-DGL1 REQ-NF2

These monitoring technologies provide certification authorities with detailed insights into cloud system performance, security, and reliability (achieving REQ-DGL1). Table 6 summarizes monitoring technologies that researchers have developed (and evaluated) that support collecting data for representative certification criteria. Furthermore, IT infrastructure-monitoring systems and monitoring tools exhibit unique characteristics that increase their relevance for being leveraged to perform monitoring-based CSC. First, most monitoring technologies embed functions to export gathered data to ASCII coded text files, HTML, or CSV log files or provide an API for requesting files that enable post-processing with external software (i.e., the CSC monitoring system; achieving REQ-DGL2). Second, they exhibit a high degree of automation by using automated agent schedulers, event-driven patterns, and automated (resource) discovery functions, and by allowing execution of external automation scripts *(*achieving REQ-NF1). Finally, monitoring technologies prominently feature an extendable architecture that allows one to easily integrate them with third-party monitoring plugins *(*achieving REQ-NF2).

Table 6. Certification Criteria that a CSC Monitoring System can Monitor by Leveraging Existing Monitoring Technologies

Certification criteria	Example monitoring technologies	Source
Availability	Adaptive distributed monitoring architectures that automatically monitor availability of resources, servers, and services.	Calero & Aguado (2015), Xiang et al. (2010)
Data location	Monitoring architecture for checking data location compliance in federated cloud infrastructures.	Massonet et al. (2011)
Penetration testing	Web application security-testing methodologies to test for evidence of vulnerabilities in the application due to deficiencies with identified security controls.	Chang & Ramachandran (2016), LaBarge & McGuire (2012)
Access management	Monitoring infrastructures that monitor users' behavior to detect anomalies and account misuse and posteriori techniques to verify compliance with privacy policies.	Doelitzscher, Reich, Knahl, Passfall, & Clarke (2012), Leeuw, Fischer-Hübner, Tseng, & Borking (2008)
Process operation	Applying process mining to gain insights into how processes are being undertaken by analyzing workflow models and a vast amount of data that is routinely gathered and stored in event logs.	Accorsi, Lowis, & Sato (2011), Jans, Alles, & Vasarhelyi (2013)
Data loss	Performing automatic backup integrity checks to ensure the recovery of corrupted files.	Chen & Lee (2014)
Data handling	Simultaneously monitoring data integrity in cases of multiple and hybrid clouds with multiple owners.	Yang & Jia (2013), Zhu et al. (2013)
Patch management	Applying formal languages to identify which patch has been applied and which is missing.	Koschorreck (2011)

Leveraging IT infrastructure monitoring systems: IT infrastructure-monitoring systems form an indispensable core for monitoring tasks when cloud service providers operate a cloud service infrastructure. A broad range of commercial (e.g., Amazon CloudWatch and AzureWatch) and open-source cloud monitoring systems exist (e.g., Nagios, Zabbix and Icinga) (Aceto, Botta, Donato, & Pescapè, 2013; Fatema et al., 2014). Accordingly, research on continuous monitoring proposes and evaluates general cloud monitoring architectures (Katsaros, Kübert, & Gallizo, 2011; Povedano-Molina, Lopez-Vega, Lopez-Soler, Corradi, & Foschini, 2013) and monitoring architectures for virtualized

environments (Clayman, Clegg, Mamatas, Pavlou, & Galis, 2011; Xiang et al., 2010). Refer to Syed, Gani, Ahmad, Khan, and Ahmed (2017) for a recent overview of systems and related research. These IT infrastructure monitoring systems can gather an extensive range of data because they monitor applications, services, operating systems, system metrics, and network infrastructures and provide key performance indicators for both platforms and applications (achieving REQ-DGL1) (Aceto et al., 2013). For example, Nagios provides functions to gather relevant monitoring data actively (i.e., Nagios executes a plugin on a host or service to poll information) and passively (i.e., external applications submit information to Nagios) that Nagios then uses to analyze the availability of infrastructure components and to perform capacity planning, among others.

Leveraging monitoring tools: in addition to infrastructure-monitoring systems, providers often implement and operate monitoring tools for special purposes. These monitoring tools provide various monitoring functions, such as network scanning (e.g., Nmap, Angry IP Scanner), network packet analysis (e.g., Wireshark, Bro Network Security Monitor), vulnerability scanning (e.g., Lynis, Qualys), penetration testing (e.g., OWASP Zed Attack Proxy), database monitoring (e.g., DBAmon), and process or workflow management (e.g., Widen Collective, Nintex), and special cloud monitoring (e.g., Hyperic HQ). Similarly, prior research on continuous monitoring proposes specific tools and techniques to detect intrusions (Modi et al., 2013), monitor service-level agreements (Comuzzi & Spanoudakis, 2010; Emeakaroha et al., 2012), and so on. While infrastructure-monitoring systems currently have limited data-analysis capabilities, monitoring tools typically contain rich data-analysis functions for specific purposes. For example, the IT monitoring system Zabbix evaluates monitoring data by continuously observing user-defined thresholds and evaluating logical definitions of problem states only, whereas the monitoring tool Qualys (continuously) scans complex internal networks to collect comprehensive information that it analyzes in detail to detect security, compliance, and other issues. Consequently, leveraging these monitoring tools for CSC can provide certification authorities with detailed insights about cloud service and process operations that surpass the capabilities of infrastructure monitoring systems (achieving REQ-DGL1).

Leveraging monitoring plugins: given the extendable architecture of monitoring technologies that allows one to easily integrate third-party monitoring plugins into them (achieving REQ-NF2*)*, large communities have formed that focus on steadily extending monitoring technologies' functionalities. Typically, one can access monitoring plugins over sharing platforms, and these plugins fall under open-source licenses that enable programmers to modify or adapt plugins to a particular monitoring scenario. For example, the Nagios' Exchange platform includes more than 5,700 listings of various add-ons and extensions that enable administrators to extend core functionalities, use alternative user interfaces, and integrate new components for the monitoring system Nagios (achieving REQ-DGL1). Therefore, cloud service providers can leverage monitoring plugins to easily extend monitoring technologies' capabilities (refer to Appendix G for example plugins that may be useful in the monitoring-based CSC context).

5.2.2 Access External Interfaces

Besides leveraging existing monitoring technologies, a CSC monitoring system may also connect to external sources that provide valuable information *(*achieving REQ-DGL1), which Table 7 summarizes.

Table 7. Design Guidelines to Access External Interfaces

ID	Design guideline	Description	Corresponding meta-requirements
DG-4	Integrate external databases	Connect to databases that provide valuable information.	REQ-DGL1
DG-5	Access interfaces of subproviders	Access service interfaces of subproviders to gather information about certification adherence that subproviders provide.	REQ-DGL3
DG-6	Monitor services of subproviders	Incorporate means to measure services that subproviders provide (e.g., test-based CSC methodologies).	

Integrate external databases: a CSC monitoring system may gather additional information from external databases (achieving REQ-DGL1). As a cloud service consultant said: "It would be very nice if you have a public database…that tells you which versions you have and which versions are threatened by certain vulnerabilities". For example, a CSC monitoring system can integrate the CVE security vulnerability database (refer to www.cvedetails.com) to gather information about current software vulnerabilities. This

and related databases provide detailed definitions of identified vulnerabilities, the measures that one should take to verify a vulnerability, and a score that ranks the vulnerability's severity.

Access or monitor interfaces of subproviders: few studies have explored the concept of cloud computing as a supply chain in detail even though monitoring the cloud supply chain pertains highly to cloud customers (Herrera & Janczewski, 2016). To achieve REQ-DGL3, cloud providers can distribute monitoring-based CSC across their supply chain partners. Thus, a supply chain partner may also implement a CSC monitoring system that gathers monitoring data to prove adherence to certification criteria. As one cloud service provider said: "I would obligate my subproviders to provide respective information or reports…that enable me to prove adherence to certification criteria". The CSC system then offers an interface that partners can access to provide required evidence (achieving REQ-DGL3). In addition, a CSC monitoring system can implement test-based CSC methodologies because they measure a cloud service from the outside (e.g., refer to Stephanow and Banse (2017) and Stephanow and Khajehmoogahi (2017) for representative test-based procedures). As one certification authority representative said: "At least one can monitor interfaces that connect both services". Nevertheless, the current lack of research about monitoring supply chain partners creates an opportunity for further research to apply theoretical concepts from supply chain coordination mechanisms adjusted to CSC's specific features and challenges to develop new conceptual models and methodologies.

5.2.3 Apply and Operate an Agent-based Architectural Model

To gather certification-relevant data across different monitoring technologies and external interfaces, a CSC monitoring system should exhibit an agent-based architecture (refer to Table 8).

Table 8. Design Guidelines to Apply and Operate an Agent-based Architectural Model

ID	Design guideline	Description	Corresponding meta-requirements
DG-7	Apply an agent-based architecture model	Dispatch certification agents to different monitoring technologies to gather certain certification-relevant data.	REQ-DGL1 REQ-DGL2 REQ-NF1 REQ-NF2
DG-8	Equip agents with security policies	Agents can receive security policies to ensure that they comply with data-protection regulations or customer requirements.	REQ-DGL4
DG-9	Perform service-focused aggregation by using agent teams	Organize agents as hierarchically structured teams to aggregate data across monitoring technologies.	REQ-AL1
DG-10	Store meta-information about agent operations	Provided meta-information should comprise data on 1) what was monitored, 2) how it was monitored, 3) when it was monitored, 4) who performed the monitoring, and 5) the monitoring results.	REQ-NF3

Apply an agent-based architectural model: researchers and practitioners have already developed several architectural patterns and mechanisms to achieve an agent-based architecture, such as the JADE framework (Bellifemine, Poggi, & Rimassa, 2001). Under this architecture, a CSC monitoring system initiates a certification agent and dispatches it to different monitoring technologies to gather certain certification-relevant data *(*achieving REQ-DGL1 and REQ-DGL2*)*. Agents are software objects that achieve individual goals by autonomously performing actions and reacting to events in a dynamic environment (Chou, Du, & Lai, 2007). Typically, an agent has mobility and intelligence. Mobility means that the agent can travel from one platform to another, and intelligence refers to the agent's artificial intelligence that enables it to use sophisticated computation or behavioral models when working on data resources. A flexible agent-based architecture (e.g., platform independent) and an adaptable agent-based architecture (e.g., an agent can be deployed as required) facilitate data collection in a distributed and heterogeneous monitoring technology landscape (Wu, Shao, Ho, & Chang, 2008).

A CSC monitoring system may build on automated agent schedulers, event-driven patterns, and automated (resource) discovery functions to achieve a high degree of automation *(achieving REQ-NF1)*. Likewise, in terms of achieving a high degree of adaptability to address an ever-changing environment, an agent-based architecture allows the CSC monitoring system to flexibly and quickly integrate new agents to gather additional information because they are loosely coupled with monitoring technologies and can also

be added, removed, or reconfigured during runtime (achieving REQ-NF2) (Chou et al., 2007; Doelitzscher et al., 2012b).

Equip agents with security policies: CSC monitoring systems must ensure data confidentiality when gathering required data. Assuring confidentiality involves preserving authorized restrictions on access and disclosure, which includes protecting personal privacy and proprietary information (National Institutes of Standards and Technology, 2002). In the context of monitoring-based CSC, cloud experts argue that a CSC monitoring system should not be able to gather data about user actions to prevent employee surveillance or customer data to adhere to service-level agreements. To ensure that dispatched agents comply with data-protection regulations or customer requirements, agents can receive security policies (achieving REQ-DGL4). Through security policies, each agent receives a rule set (its "intelligence") that specifies allowed actions to limit the agent's data-collection capabilities (Doelitzscher et al., 2012b). For example, one can equip an agent with a Jess rule engine (see http://www.jessrules.com/) and a knowledge base that contains gathered evidence (e.g., monitoring data) (Bellifemine et al., 2001). A set of action rules (i.e., in the form of Jess production rules), such as rules that derive new data by inserting them into the knowledge base and that lead an agent to execute a special action, represent the agent's behavior. One may apply self-learning algorithms to further improve an agent's intelligence (Doelitzscher et al., 2012v).

Perform service-focused aggregation by using agent teams: a CSC monitoring system must aggregate monitoring data from individual resources to construct meaningful and significant status indicators of respective cloud services. Typically, a team of hierarchically structured agents perform aggregation tasks (achieving REQ-AL1) (Doelitzscher et al., 2012b; Ye, Yang, & Gan, 2012). For instance, each agent team comprises one captain agent, M mediator agents, and N operator agents (with $0 < M < N$). The captain and mediator agents mainly focus on coordination and aggregation, and operator agents collect necessary evidence from different monitoring technologies. By ordering agents in a hierarchical structure, agents can preprocess gathered information and share information, which leads to a reduced network load (Doelitzscher et al., 2012b). Furthermore, this structure increases the system's scalability by reducing data sent to upper system layers.

When aggregating data, cloud experts emphasize that a CSC monitoring system should aggregate data from a service perspective: data should be aggregated across computing resources (i.e., database and application servers) to create meaningful indicators that pertain to a particular service, such as a service availability indicator. As one cloud service provider said:

> *We have to specify availability or performance indicators or related service indicators. …The certification authority has to say what features and attributes the service has to fulfill. Then, I [as a service provider] have to check whether I am able to monitor these*.

Therefore, an agent team must aggregate gathered data across integrated monitoring systems, plugins, tools, and cloud data centers to create meaningful service indicators. Similarly, agent teams can aggregate data in a temporal dimension to consolidate data that a CSC monitoring system and respective agent gather, for example, every minute or hour to reduce the complexity and the amount of information that certification authorities must analyze.

When designing agent teams' aggregation functionalities, researchers and practitioners can build on research findings that address (automated) monitoring of service-level agreement parameters; for instance, prior research proposes strategies to apply low-level metrics to high-level service-level agreement parameters (Emeakaroha, Brandic, Maurer, & Dustdar, 2010). In the CSC context, Stephanow and Fallenbeck (2015) show examples of how one can use low-level metrics (i.e., the number of logins and the number of terminating instances) to construct complex metrics to help validate certification criteria, such as metrics for service scalability and availability and a metric that describes anomalous behavior. Building on their findings, a CSC monitoring system may also embed complex event-processing engines (Cugola & Margara, 2012) where complex metrics can be represented as queries that agents perform and applied to event streams (i.e., values of low-level monitoring metrics) (Stephanow & Fallenbeck, 2015).

Store meta-information about agent operations: providers must log, store, and transmit meta-information about CSC monitoring system operations because certification authorities and customers demand traceable monitoring results. As one cloud service customer said: "It should be evident which controls were done and when these controls were done". This meta-information should include data on 1) what was monitored, 2) how it was monitored, 3) when it was monitored, 4) who performed the monitoring,

and 5) the monitoring results. Consequently, a CSC monitoring system should store information on agent operations. For example, an agent that a cloud service provider operates (4) collects intrusion information (1) that contains certain evidence (5) by relying on an intrusion detection system (2) that analyzes incoming data packages in real-time (3). To gather relevant information, agents may rely on the corresponding monitoring technology's internal databases because monitoring technologies typically gather information on their own performance and operations *(*achieving REQ-NF3). For instance, Zabbix provides a function for internal checks that allows one to monitor its internal operations, such as when it executes (i.e., every second) and who its individual monitoring processes target (i.e., a particular server).

5.2.4 Incorporate and Secure Flexible Data Storages

A CSC monitoring system must store and archive gathered evidence so it can later archive it for processing and transmit it to certification authorities. In addition, CSC monitoring systems must ensure the stored data's integrity. Table 9 summarizes respective design guidelines.

Table 9. Guidelines to Incorporate and Secure Flexible Data Storages

ID	Design guideline	Description	Corresponding meta-requirements
DG-11	Incorporate flexible data storage technologies to store and archive data	Incorporate flexible data storage technologies to easily adjust data schemes and store additional data or results from new data analysis operations in, for example, a NoSQL document database.	REQ-DL1 REQ-NF2
DG-12	Implement means to guard data against improper modification	Integrate a trusted third-party module that provides secure log encryption functions, establish a chain of custody for digital evidence, or apply other techniques from the cloud forensics domain to prevent internal log manipulation.	REQ-DL2

Incorporate flexible data storage technologies to store and archive data: (captain) agents can gather and store certification-relevant data in supplementary databases, such as audit data marts (Rezaee, Sharbatoghlie, Elam, & McMickle, 2002). Audit data marts refer to small data repositories in the form of log files, historic tables, or data warehouses that store relevant data and enable real-time data access and automated data analyses. Traditional monitoring technologies typically incorporate data storage units built on rigorous data schemes and cannot address the scalability and flexibility challenges that contemporary applications encounter. Conversely, a CSC monitoring system should incorporate flexible data storage technologies to easily adjust data schemes and store additional data or results from new data-analysis operations (achieving REQ-NF2). Recently, NoSQL database technologies have gained momentum because they possess greater flexibility and scalability than traditional SQL databases (Madison, Barnhill, Napier, & Godin, 2015). Research of NoSQL database technologies has revealed many NoSQL database varities (e.g., key-value stores, document databases, wide-column stores, and graph databases) that each serve specific functions (Madison et al., 2015; Meijer & Bierman, 2011; Moniruzzaman & Hossain, 2013). NoSQL document databases seem to suit CSC monitoring systems because they can manage and store encoded documents, such as XML or JSON files, that existing monitoring technologies typically generate (Moniruzzaman & Hossain, 2013).

In addition to storing data, a CSC monitoring system must archive gathered and processed monitoring data for certain periods to ensure it can access historic data (achieving REQ-DL1). For data-analysis purposes, comparing current data with historic data can help agents to learn and configure exceptions and alert patterns (e.g., rule-based configurations based on deviations from historic data). To reduce the storage load of archived data, the CSC monitoring system must specify retention periods (e.g., based on a certification authority's needs, customers' needs, or regulatory requirements). Then, a CSC monitoring system must implement appropriate mechanisms to securely and automatically delete outdated data (refer to Kissel, Regenscheid, Scholl, and Stine (2014) for guidelines on media sanitization).

Implement means to guard data against improper modification: CSC systems must implement mechanisms that ensure data integrity because monitoring-based CSC bears the risk that the provider may manipulate data. Ensuring integrity involves guarding against improper information modification or destruction and includes ensuring information nonrepudiation and authenticity (National Institutes of Standards and Technology, 2002). First, cloud experts empathize that a monitoring system should collect data only from live systems; it should not collect data from test or backup systems. As one cloud service consultant said: "Continuous monitoring of live systems. We are only interested in live systems, not in any

test systems". Further, a certification authority representative said: "Nothing else is meaningful". The interviewed cloud experts jointly recommended integrating a trusted third-party module (e.g., hardware or virtual module) that provides secure log encryption functions to prevent internal log manipulation (achieving REQ-DL2). As one cloud service provider said: "You have to implement some certified or signed module to prevent any manipulation". Further, a certification authority representative said: "One might implement a foreign machine or hardware security module, something like that, that uses cryptographic techniques to show that the data are correct". A trusted third-party security module provides secure encryption functions, and a trusted third-party manages and stores encryption and metadata (e.g., encryption keys, certificates, and authentication data) to prevent provider manipulation and enable external auditability (Kunz, Niehues, & Waldmann, 2013).

To prevent manipulation, CSC monitoring system designers can build on findings from research on cloud forensics. Cloud forensics refers to applying scientific principles and technological practices to reconstruct past cloud computing events by identifying, collecting, preserving, examining, interpreting, and reporting digital evidence (National Institute of Standards and Technology, 2014). Researchers have proposed various procedures to address the challenges with cloud forensics (i.e., malicious cloud service providers that manipulate log files). These cloud forensic procedures enable third-party investigators to collect and analyze reliable data (Pichan, Lazarescu, & Soh, 2015). For example, one method for revealing data manipulation involves establishing a chain of custody for digital evidence, which represents a roadmap showing how data were collected, analyzed, and preserved for presentation as evidence in court (Lin, Lee, & Wu, 2012). Researchers have proposed several procedures to gather trusted certification-relevant data, such as remotely acquiring data over trusted and secure channels, using management planes, performing live forensics on systems in a running state, and snapshotting a clone of a virtual image (refer to Pichan et al. (2015) for a detailed comparison). Consequently, a CSC monitoring system may contain various remedial techniques to prevent manipulation in monitoring-based CSC (achieving REQ-DL2).

5.2.5 Secure Data Exchange

Prior to cloud service providers' transferring data to a certification authority, they must mask and filter gathered monitoring data to limit access to sensitive information and prevent it from leaking. Then, cloud service providers can either actively transmit certification-relevant data or provide it via an interface when the certification authority requests it. Regardless of the transmission modality, certification authorities recommend that cloud service providers provide meta-data containing information about deployed cloud service technologies, monitoring operations, monitoring system configurations, and data-collection frequencies (achieving REQ-NF3). These metadata help certification authorities to better understand and trace conducted monitoring operations, which they need to do to evaluate certification criteria adherence. Table 10 summarizes guidelines for data exchange.

Table 10. Guidelines for Secure Data Exchange

ID	Design guideline	Description	Corresponding meta-requirements
DG-13	Embed data-masking techniques	Embed data-masking techniques such as encryption, substitution, and nulling out.	REQ-DGL4 REQ-AL2 REQ-IL2
DG-14	Implement attribute-based access control and define access policies	Implement an attribute-based access control to filter data according to an authority's needs and privileges. Specify access policies that define the access rules for the allowable subjects, operations, and environmental conditions to the object.	REQ-AL2 REQ-AL3 REQ-IL2
DG-15	Implement encrypted data-transmission means	Provide functionalities to automatically generate reports based on analyzed data and automatically transmit encrypted reports about defined points in time.	REQ-IL1 REQ-IL2
DG-16	Implement secure data-providing interfaces	Implement passive interfaces that enable certification authorities to access data.	

Embed data-masking techniques: IS researchers have extensively employed the data–masking concept to increase the level of information security and protect data that organizations share with third parties (Baranchikov, Gromov, Gurov, Grinchenko, & Babaev, 2016; Domingo-Ferrer & Mateo-Sanz, 2002; Ravikumar, Rabi, Manjunath, Hegadi, & Archana, 2011). Data masking (also referred to as data

obfuscation and scrambling) refers to the process in which one obscures specific data elements in data stores. Data masking ensures that realistic (but not real) data replaces sensitive data to prevent the availability of sensitive information outside the authorized environment.

Certification agents can employ several data-masking techniques when gathering data (achieving REQ-DGL4 and REQ-AL2), such as substitution (replacing existing data with random values) and nulling out (deleting sensitive data and replacing a data field with NULL values) (Li & Motiwalla, 2009; Sarada, Abitha, Manikandan, & Sairam, 2015). Furthermore, a CSC monitoring system may apply encryption techniques that suit the CSC context because encryption offers the option of leaving data in place and visible to people with an appropriate key (i.e., employees of the cloud service provider) while remaining effectively useless to anybody without a key (i.e., employees of the certification authority). Therefore, a CSC monitoring system may encrypt data fragments that third parties should not read (achieving REQ-AL2 and REQ-IL2).

Cloud service providers can apply data-masking techniques in a CSC context when, for example, transmitting results from penetration tools. The interviewed experts regarded providing comprehensive results from a penetration test as critical because malicious certification authority employees may leak and/or misuse identified system vulnerabilities. For certification authorities, however, the results from penetration tests serve as excellent indicators to evaluate certification adherence and confirm secure services. In this case, a CSC monitoring system must mask the detailed results from penetration-testing tools and provide only abstract but certification-relevant information for certification authorities, such as "two minor security vulnerabilities and one major security vulnerability".

Implement attribute-based access control and define access policies: certification authorities require different amounts of information because they have different certification and auditing scopes. Consequently, a CSC monitoring system should provide a function to filter or limit access to gathered and stored data according to what a certification authority needs prior to data transmission. Data filtering involves redefining data sets to remove redundant or irrelevant data.

A CSC monitoring system may implement an attribute-based access control to filter data according to an authority's needs and privileges *(*achieving REQ-AL2, REQ-AL3, and REQ-IL2). Attribute-based access control represents a popular approach to access control that provides flexibility suitable for current dynamic distributed systems (Hu, Ferraiolo, Kuhn, Kacker, & Lei, 2015). In this approach, one grants or denies data-access or transmission requests based on arbitrary attributes of users, objects, and optionally environmental conditions that may be globally recognized and tailored to current policies. When setting up a CSC monitoring system, a cloud service provider must define policies that outline access rules for allowable subjects (i.e., authorities), operations (i.e., view monitoring data), and environmental conditions (i.e., a specific certification) with respect to an object (i.e., a monitoring file). A cloud service provider can derive policies depending on a certification authority's data needs, such as the criteria that the latter will continuously check. Likewise, CSC monitoring systems and corresponding access policies should consider the individual audit focus of the respective certification authority. As one certification authority representative said: "The data should be limited to a specific perspective, like an operating or compliance management or audit management perspective".

Implement encrypted data transmission means: practitioners suggested that cloud service providers may actively transmit filtered and masked data (achieving REQ-IL1). As two separate cloud service providers said:

> *Password security is ensured. Every week, we will generate a report from a password manager tool and send it to the auditor*.
>
> *Every day, I get a report about process operation; is there any deviation or issue with processes? […] I can provide this report to an auditor*.

For example, a certification authority needs to validate adherence to certification criterion, "a cloud service provider should regularly perform reviews of firewall rules" (Lins, Schneider, & Sunyaev, 2019; Schneider et al., 2014). To do so, a cloud service provider can transmit a short report that contains various data, such as the date, firewall policy version, number of offending firewall rules, initiated operations, and completed changes. To increase transmission efficiency, a CSC monitoring system should provide functions to automatically generate reports based on analyzed data and automatically transmit these reports about defined points in time (achieving REQ-IL1).

When transmitting data to certification authorities, cloud service providers must ensure that the data do not leak (e.g., due to a man-in-the-middle attack). In this scenario, an attacker secretly relays and possibly alters the communicated material between the certification authority and the provider who believe that they are directly communicating with each other. Therefore, a CSC system must encrypt data prior to sending the data to certification authorities *(achieving REQ-IL2)*.

Implement secure data-providing interfaces: a CSC monitoring system can implement passive interfaces that enable certification authorities to access data (achieving REQ-IL1). As one certification authority representative said: "[CSC monitoring system] provides an API, I can then access metrics that are provided and authorized. That would be nice". Certification authorities may implement different types of data-providing interfaces, such as a standardized data-exchange interface that provides XML-formatted log files (i.e., allows direct data access or export of data). In addition, certification authorities state that a graphical user interface (i.e., a simple Web frontend that presents certification-relevant data) can support CSC processes. Existing monitoring technologies already provide a graphical user interface that allows users to inspect monitoring data, adjust monitoring configurations (e.g., monitoring frequency), invoke graphs and charts, and perform additional administrative actions. However, only administrators can use most user interfaces, whereas some monitoring systems, such as CA Unified Infrastructure Management, provide means to share dashboards with internal and external stakeholders with granular control, which enables certification authorities to inspect gathered and analyzed data.

Providing relevant information via interfaces to certification authorities requires that providers implement robust and secure access control systems (achieving REQ-IL2). Similarly, providers must ensure that certification authorizes can access CSC monitoring systems and their interfaces. As a cloud service provider said: "This is prone to faults. Catchword: interface is shortly unavailable and I don't have any data anymore". Ensuring availability involves ensuring that a certification authority can access and use information in a timely and reliable manner (National Institutes of Standards and Technology, 2002). To disrupt the availability and process of CSC, attackers may target interfaces by, for example, performing a distributed denial-of-service attack. Therefore, cloud service providers must implement appropriate countermeasures for potential attacks, such as limiting the number of failed login attempts or introducing time delays between successive attempts (achieving REQ-IL2).

6 Discussion

6.1 Advantages and Boundaries of Monitoring-Based Certification

Although prior research has focused on developing and evaluating test-based CSC methodologies, test-based certification requires that certification authorities can access the cloud infrastructure (which various issues may limit) and fails to deal with an ever-changing cloud infrastructure. Monitoring-based certification that leverages existing monitoring technologies to collect and provide certification-relevant data overcomes the limitations with test-based CSC because it does not require direct access to cloud infrastructure components. In this section, we discuss advantages and boundaries of monitoring-based certification based on the insights we gained from the interviews with cloud experts.

Performing monitoring-based CSC benefits cloud service providers, certification authorities, and cloud service customers alike (Lins et al., 2016b; Teigeler et al., 2018). Cloud service providers receive ongoing third-party expert assessments about their systems. In addition, implementing a CSC monitoring system and evaluating monitoring results about how cloud services have performed improves the quality of internal processes and systems. Therefrom, providers can detect potential flaws and (security) incidents earlier and can save costs due to successive service improvements. In contrast to test-based CSC, cloud service providers can analyze data across different monitoring technologies and may identify any errors in the data before they transmit it to a certification authority and, hence, prevent false positives. As one cloud service provider said: "Because of a single negative event in the monitoring system that was caused by faulty operation or maintenance window or whatever, I do not want to communicate a negative service status even though there was no issue".

Certification authorities actively detect and investigate critical certification deviations as they occur, which increases certification reliability. Certification authorities can increase their auditing efficiency, achieve savings in their budgets, reduce operation times, and reduce operation fees by reducing auditing time and errors due to automated auditing processes. Likewise, monitoring-based CSC is more cost effective since

it enables certification authorities to test larger samples and examine data faster and more efficiently compared to their manual predecessors.

Cloud service customers can benefit from CSC as well. Typically, cloud environments lack control since cloud customers cede governance to cloud service providers (ENISA, 2012). CSC can counteract this lack of control by increasing transparency about providers' operations and by providing assurance about requirements (e.g., ensuring encryption, data integrity, and location), which can ultimately enhance cloud services' trustworthiness. Informing customers in cases of critical certification violations or major security incidents has become even more important because organizations, individuals, and even societies and economies now highly depend on cloud services during their day-to-day activities (Benlian et al., 2018).

Nevertheless, monitoring-based CSC has some drawbacks. Performing monitoring-based involves a high degree of complexity because it requires cloud service providers to set up comprehensive monitoring infrastructures. Providers need to implement large-scale (continuous) monitoring technologies to ensure that they provide available, current, and accurate certification-relevant data and that, most importantly, they can transmit such data to certification authorities. Organizational employees can find integrating (expensive) hardware and software into their organizations' existing IT infrastructures intimidating, particularly if it requires them to change their existing business practices or acquire new skills (Beatty, Shim, & Jones, 2001; Rogers, 2005). Consequently, decreasing system complexity and ensuring economic feasibility has critical importance when designing CSC monitoring systems. Nevertheless, interviewed cloud customers appreciated cloud service providers' efforts when participating in CSC. As a cloud service customer said: "If a certification is like a toothless tiger, I does not provide any value. There have to be a certain level of suffering for the cloud service provider".

Similarly, continuously collecting, analyzing, and preparing data might have a substantial performance impact on cloud services. Failures in these operations may even disrupt the cloud service operation. Hence, a monitoring-based CSC might threaten cloud service availability. Therefore, cloud experts recommended performing CSC operations on parallel systems on a separate infrastructure (although collecting data from live systems) and defining economically viable data collection and analysis frequencies.

Interviewees also expressed concerns regarding comparability of certification results when performing monitoring-based CSC. As one cloud service consultant said: "If we only focus on data that is provided by a service provider, comparability of monitoring results and certifications are limited". Further, a cloud service customer said: "A certification has to be comparable. Comparability can only be achieve if monitoring algorithms are identical".

The most debated drawback with monitoring-based CSC relates to the risk that cloud service providers will manipulate data to assure they continue to adhere to certification criteria. Although we derived design guidelines to prevent cloud service providers from manipulating certification-relevant data, customers' perceived risk of data manipulation may threaten trustworthiness in and reliability of monitoring-based CSC. Focus group participants and customers assessed a low likelihood that cloud service providers would internally modify data because continuous modification constitutes high expenditures. As one cloud service provider said:

> *Let's talk about a real world example. It is easy for every cloud service provider to modify its availability monitoring to prove that the cloud service is 100% available. They do not do this. Because the efforts are not outweighing the benefits*.

Data manipulation requires a provider to store data volume twice (i.e., the unmodified data for internal evaluations first and the modified data for certification authorities second). Customers might also reveal tampered data when using the service (e.g., tampered availability rate). As two separate cloud service providers said:

> *Customers will notice manipulations as well. Let us say the service is available 100%. A customer might easily verify this by developing a script that checks availability*.

> *If you do not send honest data, at some point in time customers will notice it, and then the community will go crazy*.

Yet, customers and providers recommended that certification authorities should randomly perform validation tests on regularly basis to prevent data manipulation or reveal tampered data. As one cloud service provider said: "You might validate this on a regularly basis. For example, during yearly spot

checks". Further, a cloud service customer said: "Apart from that you will trust the provider that he is sending the true values".

6.2 Implications

To counteract current shortcomings of test-based CSC and enable monitoring-based CSC, we applied a design science approach and derived universal meta-requirements and design guidelines for CSC monitoring systems. Our findings reveal that CSC monitoring systems have to fulfill various requirements, such as transmitting monitoring information in an aggregated and confidential way. To address these requirements, we show what design CSC monitoring systems should follow by, for example, integrating existing IT infrastructure monitoring systems and corresponding plugins and applying research findings from the related continuous-monitoring and -auditing domain.

We contribute to practice and research in several ways. While prior research has proven the feasibility of monitoring-based CSC by developing a prototype (Krotsiani et al., 2015) and providing recommendations on what to monitor (Stephanow & Fallenbeck, 2015), we conduct the first work that identifies meta-requirements for designing a CSC monitoring system based on exhaustive discussions with industry experts. Thereby, we have expanded the current knowledge base regarding CSC in general and monitoring-based CSC in particular. Thoroughly defined requirements are important in design science research because they ensure an IT artifact's relevance to a real-world problem and provide the basis for creating and evaluating an IT artifact (Beck et al., 2013; Hevner et al., 2004; Pries-Heje & Baskerville, 2008; Walls et al., 1992). Without clearly defined requirements, the IT artifact will not be useful and cannot offer a satisfying solution (Albert et al., 2004; Beck et al., 2013; Markus et al., 2002). Furthermore, we derive several design guidelines to address meta-requirements. Researchers can use these guidelines to implement an IT artifact and evaluate proposed concepts. Hence, our derived design guidelines contribute to the development of CSC monitoring systems and enable monitoring-based CSC that overcomes issues of prior test-based approaches.

As we grounded the CSC monitoring system's design in justificatory knowledge, we have not only increased the extent to which designers understand the problem domain (Arazy et al., 2010) and explained why a CSC monitoring system takes the form that it does and why it works (Gregor & Jones, 2007) but also formulated high-level design guidelines independent of technological constraints and specific implementation details (Arazy et al., 2010). Consequently, in regard to artifact mutability (Gregor & Jones, 2007), our derived meta-requirements and design guidelines apply to both cloud service contexts and to continuous assessments of IT infrastructures in related contexts. For instance, although we discuss certification criteria that certification authorities should continuously audit in a cloud service context, requirements and corresponding guidelines for aggregating, securing, filtering, and providing certification-relevant data also apply to related IT assessments domains.

For practice, we provide grounding for developing systems that enable monitoring-based CSC and, thereby, provide a way to overcome practical issues with test-based CSC that prior research has focused on (e.g., Kunz & Stephanow, 2017; Stephanow & Khajehmoogahi, 2017; Wang et al., 2014). In particular, our design guidelines show how one can leverage existing monitoring technologies by implementing an agent-based certification server to enable monitoring-based CSC. One can use our derived meta-requirements as evaluation framework to assess suitability and readiness of existing monitoring technologies in regard to CSC purposes. By providing meta-requirements and design guidelines that depict how to design CSC monitoring systems, we hope to encourage cloud service providers and certification authorities to participate in CSC processes and, ultimately, create trustworthy certifications and cloud services.

6.3 Limitations

As with any study, our study has some limitations. First, although we propose meta-requirements and design guidelines by following the first two phases in Kuechler and Vaishnavi's (2008) design science approach, we do not develop a CSC monitoring system and, more importantly, evaluate it afterwards. With this study, we focus on giving insights into current state and issues of CSC to motive researchers and practitioners to engage in these topics. We believe that CSC constitutes one possible way to address current gaps and issues in cloud computing. We leave it to future research to develop and evaluate a monitoring technology to perform continuous cloud service certification and, thus, complete the first design cycle.

Second, our study has limitations concerning the number and depth of interviews we conducted to gather necessary data. While we conducted five expert focus group interviews with 33 cloud experts and 10 one-to-one interviews and derived and validated meta-requirements and design guidelines based on those interviews, future research might focus on gathering more information on specific findings to increase understanding. In addition, we chose a key informant method for data collection and focused on informants on a managerial level. This approach, while having advantages for our exploratory work, has the limitations that the data reflects perceptions of one person per company. Likewise, interviewees may have found it difficult to verbalize some requirements about designing CSC monitoring systems, particularly since they had not yet adopted such system. Future research might observe providers and authorities interacting with CSC monitoring systems to derive further or refine existing meta-requirements. Moreover, we analyzed data based on how we interpreted it. Nevertheless, we confidently believe that we reduced potential interpretation bias by constantly validating our interim findings throughout the three design cycles.

The third limitation refers to generalizability. The data-collection methodology we applied had advantages for our exploratory work, and the theoretical sampling strategy we followed suggests that we recruited highly qualified individuals to participate in the study. Nonetheless, one cannot extrapolate our results to other populations (Lee & Baskerville, 2003). Future research should investigate whether and to what extent our results generalize to other contexts than cloud computing and other types of certifications, particularly mandatory certifications.

Finally, we provide a set of design guidelines based on related work, interview findings, and knowledge on existing monitoring technologies. Designers and researchers may come up with additional design guidelines that fulfill meta-requirements. We analyzed only a limited amount of monitoring technologies in detail and, thus, might have neglected promising monitoring technologies that already fulfill most of our requirements and, thus, that certification authorities could leverage CSC purposes.

6.4 Future Work

Future work might implement a prototype system according to derived design guidelines to test our meta-requirements. Therefore, researchers first need to define a set of testable design propositions to verify whether design guidelines satisfy meta-requirements (Walls et al., 1992). One can articulate numerous testable design propositions to address the extent to which a guideline satisfies meta-requirements. However, in the context of a single study, one can articulate and test only a few propositions (Arazy et al., 2010; Walls et al., 1992). Table 11 articulates example propositions to guide future work.

Table 11. Testable Design Propositions

No.	Testable design proposition description	Design guidelines	Meta-requirements
TDP1	One can feasibly design a CSC monitoring system that applies an agent-based architecture model to gather certification-relevant data from different monitoring technologies.	DG-7	REQ-DGL1 REQ-DGL2 REQ-NF1 REQ-NF2
TDP2	One can feasibly design a CSC monitoring system that integrates a trusted third-party module to prevent internal log manipulation.	DG-12	REQ-DL2
TDP3	One can feasibly design a CSC monitoring system that implements data-masking techniques such as encryption, substitution, and nulling out to ensure data confidentiality.	DG-13	REQ-DGL4 REQ-AL2 REQ-IL2
TDP4	One can feasibly design a CSC monitoring system that provides a data interface to exchange monitoring data with certification authorities.	DG-16	REQ-IL1

Further, future research should evaluate how certification authorities and cloud service providers can exchange monitoring data while considering taking security challenges. With this study, we hope to encourage further researchers to address these issues and, thereby, create continuously secure and reliable cloud services.

7 Conclusion

Given that organizations have begun to increasingly rely on cloud service providers to support their daily IT needs, the necessity for continuous, highly reliable, and secure services has increased in importance. CSC represents a disruptive change because it provides cloud customers with ongoing, up-to-date feedback about a cloud service's security and privacy levels compared with conventional certifications that assess a cloud service only at a specific point in time. Nevertheless, certification authorities and cloud service providers continue to struggle with implementing CSC processes and systems due to their high complexity and the challenging interplay between both parties. While practitioners have scarcely applied test-based methodologies to achieve CSC, monitoring-based CSC strategies represent auspicious means because they reuse data that cloud service providers routinely gather while monitoring their cloud service infrastructure. However, previous research has mostly focused on achieving and applying test-based CSC. Thus, we need to better understand how to design CSC monitoring systems to enable certification authorities to use monitoring-based CSC to monitor cloud services. To counteract the current shortcomings with test-based CSC and foster the diffusion and application of CSC, we derived universal meta-requirements and design guidelines for designing CSC monitoring systems by conducting comprehensive interviews with various stakeholders, reviewing related literature, and surveying available monitoring technologies. Our findings reveal that CSC monitoring systems must fulfill various requirements, such as transmitting and monitoring information in an aggregated and anonymized manner. To address these requirements, we show how to properly design CSC monitoring systems (e.g., by applying anagent-based system architecture and integrating existing IT infrastructure monitoring systems and corresponding plugins).

Acknowledgments

This research was funded by the German Federal Ministry for Education and Research (grant no. 16KIS0079). We thank the associate editor and anonymous reviewers for their careful reading and their many insightful and valuable comments and suggestions that helped to greatly improve this paper.

References

Abraham, C., Boudreau, M.-C., Junglas, I., & Watson, R. (2013). Enriching our theoretical repertoire: The role of evolutionary psychology in technology acceptance. *European Journal of Information Systems*, *22*(1), 56-75.

Accorsi, R., Lowis, L., & Sato, Y. (2011). Automated certification for compliant cloud-based business processes. *Business & Information Systems Engineering*, *3*(3), 145-154.

Aceto, G., Botta, A., Donato, W. de, & Pescapè, A. (2013). Cloud monitoring: A survey. *Computer Networks*, *57*(9), 2093-2115.

Albert, T. C., Goes, P. B., & Gupta, A. (2004). GIST: A model for design and management of content and interactivity of customer-centric web sites. *MIS Quarterly*, *28*(2), 161-182.

Amrou, S., & Böhmann, T. (2016). Design and evaluation of transfer-supporting IT components for corporate training services. In *Proceedings of the 47th International Conference on Information Systems.*

Anisetti, M., Ardagna, C. A., & Damiani, E. (2015). A test-based incremental security certification scheme for cloud-based systems. In *Proceedings of the 2015 IEEE International Conference on Services Computing.*

Anisetti, M., Ardagna, C., Damiani, E., & Gaudenzi, F. (2017). A semi-automatic and trustworthy scheme for continuous cloud service certification. *IEEE Transactions on Services Computing*.

Arazy, O., Kumar, N., & Shapira, B. (2010). A theory-driven design framework for social recommender systems. *Journal of the Association for Information Systems*, *11*(9), 455-490.

Ardagna, C. A., Asal, R., Damiani, E., Dimitrakos, T., El Ioini, N., & Pahl, C. (2018). Certification-based cloud adaptation. *IEEE Transactions on Services Computing*.

Baranchikov, A. I., Gromov, A. Y., Gurov, V. S., Grinchenko, N. N., & Babaev, S. I. (2016). The technique of dynamic data masking in information systems. In *Proceedings of the 5th Mediterranean Conference on Embedded Computing.*

Beatty, R. C., Shim, J. P., & Jones, M. C. (2001). Factors influencing corporate web site adoption: A time-based assessment. *Information & Management*, *38*(6), 337-354.

Beck, R., Weber, S., & Gregory, R. W. (2013). Theory-generating design science research. *Information Systems Frontiers*, *15*(4), 637-651.

Bellifemine, F., Poggi, A., & Rimassa, G. (2001). Developing multi-agent Systems with a FIPA-compliant agent framework. *Software: Practice and Experience*, *31*(2), 103-128.

Benlian, A., Kettinger, W. J., Sunyaev, A., & Winkler, T. J. (2018). The transformative value of cloud computing: A decoupling, platformization, and recombination theoretical framework. *Journal of Management Information Systems*, *35*(3), 1-24.

Bryant, A., & Charmaz, K. (2007). *The SAGE handbook of grounded theory*. Thousand Oaks, CA: Sage.

Buyya, R., Ranjan, R., & Calheiros, R. N. (2010). InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services. In C.-H. Hsu, L. T. Yang, J. H. Park, & S.-S. Yeo (Eds.), *Algorithms and architectures for parallel processing* (pp. 13-31). Berlin: Springer.

Calero, J. A. M., & Aguado, J. G. (2015). MonPaaS: An adaptive monitoring platformas a service for cloud computing infrastructures and services. *IEEE Transactions on Services Computing*, *8*(1), 65-78.

Canadian Institute of Chartered Accountants. (1999). *Continuous auditing* (research report). Toronto, Canada.

Chang, V., & Ramachandran, M. (2016). Towards achieving data security with the cloud computing adoption framework. *IEEE Transactions on Services Computing*, *9*(1), 138-151.

Charmaz, K. (2000). Grounded theory: Objectivist and constructivist methods. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of qualitative research* (pp. 509-535). Thousand Oaks, CA: Sage.

Chen, H. C. H., & Lee, P. P. C. (2014). Enabling data integrity protection in regenerating-coding-based cloud storage: Theory and implementation. *IEEE Transactions on Parallel and Distributed Systems*, *25*(2), 407-416.

Chou, C. L.-y., Du, T., & Lai, V. S. (2007). Continuous auditing with a multi-agent system. *Decision Support Systems*, *42*(4), 2274-2292.

Clayman, S., Clegg, R., Mamatas, L., Pavlou, G., & Galis, A. (2011). Monitoring, aggregation and filtering for efficient management of virtual networks. In *Proceedings of the 7th International Conference on Network and Service Management.*

Comuzzi, M., & Spanoudakis, G. (2010). Dynamic set-up of monitoring infrastructures for service based systems. In *Proceedings of the ACM Symposium on Applied Computing.*

Corbin, J., & Strauss, A. (2015). *Basics of qualitative research: Techniques and procedures for developing grounded theory* (4th ed.). Thousand Oaks, CA: Sage.

Cugola, G., & Margara, A. (2012). Processing flows of information. *ACM Computing Surveys*, *44*(3), 1-62.

Doelitzscher, F., Fischer, C., Moskal, D., Reich, C., Knahl, M., & Clarke, N. (2012a). Validating cloud infrastructure changes by cloud audits. In *Proceedings of the 8th IEEE World Congress on Services.*

Doelitzscher, F., Reich, C., Knahl, M., Passfall, A., & Clarke, N. (2012b). An agent based business aware incident detection system for cloud environments. *Journal of Cloud Computing: Advances, Systems and Applications*, *1*(1), 1-19.

Domingo-Ferrer, J., & Mateo-Sanz, J. M. (2002). Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering*, *14*(1), 189-201.

Donoghue, S. (2000). Projective techniques in consumer research. *Journal of Family Ecology and Consumer Sciences*, *28*(1), 47-53.

Emeakaroha, V. C., Brandic, I., Maurer, M., & Dustdar, S. (2010). Low level metrics to high level SLAs—LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments. In *Proceedings of the International Conference on High Performance Computing and Simulation.*

Emeakaroha, V. C., Netto, M. A.S., Calheiros, R. N., Brandic, I., Buyya, R., & de Rose, C. A.F. (2012). Towards autonomic detection of SLA violations in cloud infrastructures. *Future Generation Computer Systems*, *28*(7), 1017-1029.

ENISA. (2012). *Cloud computing—benefits, risks and recommendations for information security.* Retrieved from https://resilience.enisa.europa.eu/cloud-security-and-resilience/publications/cloud-computing-benefits-risks-and-recommendations-for-information-security

Fatema, K., Emeakaroha, V. C., Healy, P. D., Morrison, J. P., & Lynn, T. (2014). A survey of cloud monitoring tools: Taxonomy, capabilities and objectives. *Journal of Parallel and Distributed Computing*, *74*(10), 2918-2933.

Felici, M., Koulouris, T., & Pearson, S. (2013). Accountability for data governance in cloud ecosystems. In *Proceedings of the 5th International Conference on Cloud Computing Technology and Science.*

Fernandes, D. B., Soares, L. B., Gomes, J., Freire, M., & Inácio, P. M. (2014). Security issues in cloud environments: A survey. *International Journal of Information Security*, *13*(2), 113-170.

Fernandez, W. D. (2004). The grounded theory method and case study data in IS research: Issues and design. In *Proceedings of Information Systems Foundations Workshop: Constructing and Criticising.*

Flick, U. (2014). *An introduction to qualitative research* (5th ed.). Thousand, CA: Sage.

Gao, F., Thiebes, S., & Sunyaev, A. (2018). Rethinking the meaning of cloud computing for health care: A taxonomic perspective and future research directions. *Journal of Medical Internet Research*, *20*(7), e10041.

Gasson, S., & Waters, J. (2013). Using a grounded theory approach to study online collaboration behaviors. *European Journal of Information Systems*, *22*(1), 95-118.

Gorden, R. L. (1980). *Interviewing: Strategy, techniques, and tactics* (3rd ed.). Homewood, IL: Dorsey Press.

Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association for Information Systems*, *8*(5), 312-335.

Grozev, N., & Buyya, R. (2014). Inter-cloud architectures and application brokering: Taxonomy and survey. *Software: Practice and Experience*, *44*(3), 369-390.

Heath, H. (2006). Exploring the influences and use of the literature during a grounded theory study. *Journal of Research in Nursing*, *11*(6), 519-528.

Hentschel, R., Leyh, C., & Petznick, A. (2018). Current cloud challenges in Germany: The perspective of cloud service providers. *Journal of Cloud Computing*, *7*(5), 1-12.

Herrera, A., & Janczewski, L. (2016). Cloud supply chain resilience model: Development and validation. In *Proceedings of the 49th Hawaii International Conference on System Sciences.*

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75-105.

Hu, V., Ferraiolo, D. F., Kuhn, D. R., Kacker, R. N., & Lei, Y. (2015). Implementing and managing policy rules in attribute based access control. In *Proceedings of the IEEE International Conference on Information Reuse and Integration.*

International Organization for Standardization. (2004). *Conformity assessment—vocabulary and general principles* (ISO/IEC 17000:2004). Retrieved from https://www.iso.org/standard/29316.html

Jans, M., Alles, M., & Vasarhelyi, M. (2013). The case for process mining in auditing: Sources of value added and areas of application. *International Journal of Accounting Information Systems*, *14*(1), 1-20.

Jones, S., & Hughes, J. (2001). Understanding IS evaluation as a complex social process: A case study of a UK local authority. *European Journal of Information Systems*, *10*(4), 189-203.

Katopodis, S., Spanoudakis, G., & Mahbub, K. (2014). Towards hybrid cloud service certification models. In *Proceedings of the IEEE International Conference on Services Computing.*

Katsaros, G., Kübert, R., & Gallizo, G. (2011). Building a service-oriented monitoring framework with REST and Nagios. In *Proceedings of the IEEE International Conference on Services Computing.*

Khan, K. M., & Malluhi, Q. (2013). Trust in cloud services: Providing more controls to clients. *Computer*, *46*(7), 94-96.

Kissel, R., Regenscheid, A., Scholl, M., & Stine, K. (2014). Guidelines for media sanitization. *National Institute of Standards and Technology.* Retrieved from https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-88r1.pdf

Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly*, *23*(1), 67-93.

Koschorreck, G. (2011). Automated audit of compliance and security controls. In *Proceedings of the 6th International Conference on IT Security Incident Management and IT Forensics.*

Krotsiani, M. (2016). *Model driven certification of cloud service security based on continuous monitoring* (doctoral dissertation). City University London. Retrieved from http://openaccess.city.ac.uk/15044/

Krotsiani, M., Spanoudakis, G., & Kloukinas, C. (2015). Monitoring-based certification of cloud service security. In *Proceedings of OTM Confederated International Conferences* (LNCS vol. 9415).

Kuechler, B., & Vaishnavi, V. (2008). On theory development in design science research: Anatomy of a research project. *European Journal of Information Systems*, *17*(5), 489-504.

Kuechler, W., Park, E. H., & Vaishnavi, V. (2009). Formalizing theory development in IS design science research: Learning from qualitative research. In *Proceedings of the 15th Americas Conference on Information Systems.*

Kunz, I., & Stephanow, P. (2017). A process model to support continuous certification of cloud services. In *Proceedings of the 31st International Conference on Advanced Information Networking and Applications.*

Kunz, T., Niehues, P., & Waldmann, U. (2013). Technische unterstützung von audits bei cloud-betreibern. *Datenschutz Und Datensicherheit*, *37*(8), 521-525.

LaBarge, R., & McGuire, T. (2012). Cloud penetration testing. *International Journal on Cloud Computing: Services and Architecture*, *2*(2), 43-62.

Lansing, J., Benlian, A., & Sunyaev, A. (2018). "Unblackboxing" decision makers' interpretations of IS certifications in the context of cloud service certifications. *Journal of the Association for Information Systems*, *19*(11), 1064-1096.

Lee, A. S., & Baskerville, R. L. (2003). Generalizing generalizability in information systems research. *Information Systems Research*, *14*(3), 221-243.

Lee, O.-K., Sambamurthy, V., Lim, K. H., & Wei, K. K. (2015). How does IT ambidexterity impact organizational agility? *Information Systems Research*, *26*(2), 398-417.

Leeuw, E., Fischer-Hübner, S., Tseng, J., & Borking, J. (Eds.). (2008). *Policies and research in identity management*. Boston, MA: Springer.

Levina, N., & Vaast, E. (2005). The emergence of boundary spanning competence in practice: Implications for implementation and use of information systems. *MIS Quarterly*, *29*(2), 335-363.

Li, X.-B., & Motiwalla, L. (2009). Protecting patient privacy with data masking. In *Proceedings of the AIS SIGSEC Workshop on Information Security and Privacy.*

Lin, C.-H., Lee, C.-Y., & Wu, T.-W. (2012). A cloud-aided RSA signature scheme for sealing and storing the digital evidences in computer forensics. *International Journal of Security and Applications*, *6*(2), 241-244.

Lins, S., Grochol, P., Schneider, S., & Sunyaev, A. (2016a). Dynamic certification of cloud services: Trust, but verify! *IEEE Security and Privacy* , *14*(2), 67-71.

Lins, S., Schneider, D., Benlian, A., & Sunyaev, A. (2017). The shifts of fortune test the reliability of friends—the brittle nature of signal reliability in cloud service markets. In *Proceedings of the 38th International Conference on Information Systems.*

Lins, S., Schneider, S., & Sunyaev, A. (2018). Trust is good, control is better: Creating secure clouds by continuous auditing. *IEEE Transactions on Cloud Computing*, *6*(3), 890-903.

Lins, S., & Sunyaev, A. (2017). Unblackboxing IT certifications: A theoretical model explaining IT certification effectiveness. In *Proceedings of the 38th International Conference on Information Systems.*

Lins, S., Teigeler, H., & Sunyaev, A. (2016b). Towards a bright future: Enhancing diffusion of continuous cloud service auditing by third parties. In *Proceedings of 24th European Conference on Information Systems.*

Lins, S., Thiebes, S., Schneider, S., & Sunyaev, A. (2015). What is really going on at your cloud service provider? In *Proceedings of the 48th Hawaii International Conference on System Science.*

Lins, S., Schneider, S., & Sunyaev, A. (2019). *Cloud-Service-Zertifizierung* (2nd ed.). Berlin: Springer.

Madison, M., Barnhill, M., Napier, C., & Godin, J. (2015). NoSQL database technologies. *Journal of International Technology & Information Management*, *24*(1), 1-13.

Markus, M. L., Majchrzak, A., & Gasser, L. (2002). A design theory for systems that support emergent knowledge processes. *MIS Quarterly*, *26*(3), 179-212.

Massonet, P., Naqvi, S., Ponsard, C., Latanicki, J., Rochwerger, B., & Villari, M. (2011). A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum.*

Meijer, E., & Bierman, G. (2011). A co-relational model of data for large shared data banks. *Communications of the ACM*, *54*(4), 49-58.

Mell, P., Waltermire, D., Feldman, L., Booth, H., Ouyang, A., Ragland, Z., & McBride, T. (2012). *CAESARS framework extension: An enterprise continuous monitoring technical reference architecture* (second draft). Gaithersburg, Montgomery: National Institute of Standards and Technology. Retrieved from http://csrc.nist.gov/publications/drafts/nistir-7756/Draft-NISTIR-7756_second-public-draft.pdf

Mell, P. M., & Grance, T. (2011). *The NIST definition of cloud computing: Recommendations of the National Institute of Standards and Technology*. Gaithersburg, Montgomery: National Institute of Standards and Technology. Retrieved from https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., & Rajarajan, M. (2013). A survey of intrusion detection techniques in cloud. *Journal of Network and Computer Applications*, *36*(1), 42-57.

Moniruzzaman, A. B. M., & Hossain, S. A. (2013). NoSQL database: New era of databases for big data analytics—classification, characteristics and comparison. *International Journal of Database Theory and Application*, *6*(4), 1-14.

Myers, M. D. (2013). *Qualitative research in business & management* (2nd ed.). Thousand Oaks, CA: Sage.

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization*, *17*(1), 2-26.

Nagios Enterprises. (2016). *NRPE—Nagios Remote Plugin Executor.* Retrieved from https://exchange.nagios.org/directory/image/93

National Institute of Standards and Technology. (2014). *NIST cloud computing forensic science challenges: Draft NISTIR 8006.*

National Institutes of Standards and Technology. (2002). *Federal Information Security Management Act of 2002.* Retrieved from http://csrc.nist.gov/drivers/documents/FISMA-final.pdf

Ngo, C., Demchenko, Y., & Laat, C. de (2012). Toward a dynamic trust establishment approach for multi-provider intercloud environment. In *Proceedings of the 4th International Conference on Cloud Computing Technology and Science.*

Patton, M. Q. (2015). *Qualitative research & evaluation methods: Integrating theory and practice* (4th ed.). Thousand Oaks, CA: Sage.

Pichan, A., Lazarescu, M., & Soh, S. T. (2015). Cloud forensics. *Digital Investigation*, *13*, 38-57.

Povedano-Molina, J., Lopez-Vega, J. M., Lopez-Soler, J. M., Corradi, A., & Foschini, L. (2013). DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant clouds. *Future Generation Computer Systems*, *29*, 2041-2056.

Pries-Heje, J., & Baskerville, R. (2008). The design theory nexus. *MIS Quarterly*, *32*(4), 731-755.

Quinting, A., Lins, S., Szefer, J., & Sunyaev, A. (2017). Advancing the adoption of a new generation of certifications—a theoretical model to explain the adoption of continuous cloud service certification by certification authorities. In *Proceedings of the 13th Internationale Tagung Wirtschaftsinformatik.*

Ravikumar, G. K., Rabi, J. B., Manjunath, T. N., Hegadi, R., & Archana, R. A. (2011). Design of data masking architecture and analysis of data masking techniques for testing. *International Journal of Engineering Science and Technology*, *3*(6), 5150-5159.

Rezaee, Z., Sharbatoghlie, A., Elam, R., & McMickle, P. L. (2002). Continuous auditing: Building automated auditing capability. *Auditing*, *21*(1), 147-163.

Rogers, E. M. (2005). *Diffusion of innovations* (5th ed). New York, NY: Free Press.

Sarada, G., Abitha, N., Manikandan, G., & Sairam, N. (2015). A few new approaches for data masking. In *Proceedings of the International Conference on Circuit, Power and Computing Technologies.*

Sarker, S., & Sarker, S. (2009). Exploring agility in distributed information systems development teams: An interpretive study in an offshoring context. *Information Systems Research*, *20*(3), 440-461.

Schneider, S., Lansing, J., Gao, F., & Sunyaev, A. (2014). A taxonomic perspective on certification schemes: Development of a taxonomy for cloud service certification criteria. In *Proceedings of the 47th Hawaii International Conference on System Sciences.*

Schneider, S., & Sunyaev, A. (2016). Determinant factors of cloud-sourcing decisions: Reflecting on the IT outsourcing literature in the era of cloud computing. *Journal of Information Technology*, *31*(1), 1-31.

Schneider, S., Wollersheim, J., Krcmar, H., & Sunyaev, A. (2018). How do requirements evolve over time? A case study investigating the role of context and experiences in the evolution of enterprise software requirements. *Journal of Information Technology*, *33*(2), 151-170.

Smolander, K., Rossi, M., & Purao, S. (2008). Software architectures: Blueprint, literature, language or decision? *European Journal of Information Systems*, *17*(6), 575-588.

Sommerville, I. (2012). *Software engineering* (9th ed.). Amsterdam: Pearson.

Sood, S. K. (2012). A combined approach to ensure data security in cloud computing. *Journal of Network and Computer Applications*, *35*(6), 1831-1838.

Stephanow, P., & Fallenbeck, N. (2015). Towards continuous certification of infrastructure-as-a-service using low-level metrics. In *Proceedings of the 12th IEEE International Conference on Advanced and Trusted Computing.*

Stephanow, P., & Gall, M. (2015). Language classes for cloud service certification systems. In *Proceedings of the 11th IEEE World Congress on Services.*

Stephanow, P., & Khajehmoogahi, K. (2017). Towards continuous security certification of software-as-a-service applications using Web application testing techniques. In *Proceedings of the IEEE 31st International Conference on Advanced Information Networking and Applications.*

Stephanow, P., Srivastava, G., & Schütte, J. (2016a). Test-based cloud service certification of opportunistic providers. In *Proceedings of the 8th IEEE International Conference on Cloud Computing.*

Stephanow, P., & Banse, C. (2017). Evaluating the performance of continuous test-based cloud service certification. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing.*

Stephanow, P., Banse, C., & Schütte, J. (2016b). Generating threat profiles for cloud service certification systems. In *Proceedings of the 17th International Symposium on High Assurance Systems Engineering.*

Strong, D. M., & Volkoff, O. (2010). Understanding organization—enterprise system fit: A path to theorizing the information technology artifact. *MIS Quarterly*, *34*(4), 731-756.

Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, *34*(1), 1-11.

Sunyaev, A., & Schneider, S. (2013). Cloud services certification. *Communications of the ACM*, *56*(2), 33-36.

Syed, H. J., Gani, A., Ahmad, R. W., Khan, M. K., & Ahmed, A. I. A. (2017). Cloud monitoring: A review, taxonomy, and open research issues. *Journal of Network and Computer Applications*, *98*, 11-26.

Tan, W., Fan, Y., Ghoneim, A., Hossain, M. A., & Dustdar, S. (2016). From the service-oriented architecture to the Web API economy. *IEEE Internet Computing*, *20*(4), 64-68.

Teigeler, H., Lins, S., & Sunyaev, A. (2018). Drivers vs. inhibitors—what clinches continuous service certification adoption by cloud service providers? In *Proceedings of the 51th Hawaii International Conference on System Sciences.*

Thiebes, S., Kleiber, G., & Sunyaev, A. (2017). Cancer genomics research in the cloud: A taxonomy of genome data sets. In *Proceedings of the 4th International Workshop on Genome Privacy and Security.*

Urquhart, C., Lehmann, H., & Myers, M. D. (2010). Putting the "theory" back into grounded theory: Guidelines for grounded theory studies in information systems. *Information Systems Journal*, *20*(4), 357-381.

Vaishnavi, V., & Kuechler, W. (2015). *Design science research in information systems*. Retrieved from www.desrist.org/design-research-in-information-systems/

Venters, W., & Whitley, E. A. (2012). A critical review of cloud computing: Researching desires and realities. *Journal of Information Technology*, *27*(3), 179-197.

Volkoff, O., Strong, D. M., & Elmes, M. B. (2005). Understanding enterprise systems-enabled integration. *European Journal of Information Systems*, *14*(2), 110-120.

Volkoff, O., Strong, D. M., & Elmes, M. B. (2007). Technological embeddedness and organizational change. *Organization Science*, *18*(5), 832-848.

Walls, J. G., Widmeyer, G. R., & Sawy, O. A. E. (1992). Building an information system design theory for vigilant EIS. *Information Systems Research*, *3*(1), 36-59.

Walsham, G. (1995). Interpretive case studies in IS research: Nature and method. *European Journal of Information Systems*, *4*(2), 74-81.

Wang, B., Li, B., & Li, H. (2014). Oruta: Privacy-preserving public auditing for shared data in the cloud. *IEEE Transactions on Cloud Computing*, *2*(1), 43-56.

Wei, Y., & Blake, M. B. (2010). Service-oriented computing and cloud computing: Challenges and opportunities. *IEEE Internet Computing*, *14*(6), 72-75.

Weinhardt, C., Anandasivam, A., Blau, B., Borissov, N., Meinl, T., Michalk, W., & Stößer, J. (2009). Cloud computing—a classification, business models, and research directions. *Business & Information Systems Engineering*, *1*(5), 391-399.

Wiesche, M., Jurisch, M. C., Yetton, P. W., & Krcmar, H. (2017). Grounded theory methodoogy in information systems research. *MIS Quarterly*, *41*(3), 685-701.

Windhorst, I., & Sunyaev, A. (2013). Dynamic certification of cloud services. In *Proceedings of the 8th International Conference on Availability, Reliability and Security.*

Wu, C.-H., Shao, Y. E., Ho, B.-Y., & Chang, T.-Y. (2008). On an agent-based architecture for collaborative continuous auditing. In *Proceedings of the 12th International Conference on Computer Supported Cooperative Work in Design.*

Xiang, G., Jin, H., Zou, D., Zhang, X., Wen, S., & Zhao, F. (2010). VMDriver: A driver-based monitoring mechanism for virtualization. In *Proceedings of the 29th IEEE Symposium on Reliable Distributed Systems.*

Yang, K., & Jia, X. (2013). An efficient and secure dynamic auditing protocol for data storage in cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, *24*(9), 1717-1726.

Ye, H., Yang, J., & Gan, Y. (2012). Research on continuous auditing based on multi-agent and Web services. In *Proceedings of the International Conference on Management of e-Commerce and e-Government.*

Yin, R. K. (2014). *Case study research: Design and methods* (5th ed.). Thousand Oaks, CA: Sage.

Zhu, Y., Ahn, G.-J., Hu, H., Yau, S. S., An, H. G., & Hu, C.-J. (2013). Dynamic audit services for outsourced storages in clouds. *IEEE Transactions on Services Computing*, *6*(2), 227-238.

Appendix A: Information on Interviewees

Table A1 provides information on interviewees participated in the focus group interviews. Table A2 summarizes information on interviewed cloud customers.

Table A1. Information about Focus Group Interview Participants

Area of expertise	Position	Years in company	Participated in focus group				
			1	2	3	4	5
Certification authorities representatives							
Cloud service auditing	Acting director	6		X	X	X	
Cloud service auditing	Management	3		X	X	X	
Cloud service auditing	Management	2		X	X	X	
Security analysis & auditing	Operational	5		X			
IT-Infrastructure auditing	Freelancer	14			X		
ISO 27001 auditing	Freelancer	unknown			X		
Cloud service auditing	Operational	9			X		X
Data protection and security auditing	Management	6			X		
ISO 27001 auditing	Management	20					X
ISO 27001 auditing	Operational	5					X
Data protection and security auditing	Freelancer	15					X
ISO 27001 auditing	Management	17					X
Cloud service providers							
Cloud operation, SaaS, PaaS	Management	6	X				
Cloud operation, SaaS	Management	5	X				
Cloud management, SaaS	Management	6	X				
Cloud management, SaaS, PaaS, IaaS	Management	8	X	X	X		
Cloud management, SaaS	CEO	2			X		
Cloud management, PaaS, IaaS	CEO	18			X		
Cloud management, IaaS	CEO	13			X		
Cloud management, SaaS	CEO	2			X		
Managing cloud security services	Management	8		X	X	X	
Cloud management, SaaS	Management	11		X			
Cloud management, SaaS, PaaS, IaaS	Management	6		X			
Cloud management, SaaS, PaaS	Management	17				X	
Cloud management, SaaS, PaaS	Management	20				X	
Cloud management, SaaS, PaaS, IaaS	Management	15				X	
Cloud operation, SaaS, PaaS, IaaS	Operational	7				X	
Cloud management, SaaS, PaaS, IaaS	Management	16				X	
Cloud service consultants							
Cloud operation consulting	Management	9	X				
Quality management consulting	Management	18		X			
Quality management consulting	Management	21		X			
Regulations and law consulting	Operational	4			X		
Business development consulting	Management	7			X		

Table A2. Information about Interviewed Cloud Service Customers

ID	Position	Years in company	Cloud experience	Certification experience	Industry	Company size (no. employees)	Company cloud usage	Importance of certifications	Interview duration
i01	Consultant	7	High	High (ISO)	IT	130	SaaS	High	40 min
i02	Head of IT	8	High	High (ISO)	IT	130	SaaS	High	66 min
i03	Head of IT	21	Low	High (ISO)	Metal	750	None	High	49 min
i04	Team manger	5	High	High (ISO)	IT	130	SaaS	High	58 min
i05	Head of IT	16	Low	High (ISO)	Health	9000	None	High	69 min
i06	CTO/ Founder	6	High	None	IT Software	18	Yes	Low	48 min
i07	IT-Service manager	4	High	High (ISO)	Engineering	2500	Yes	High	64 min
i08	Head of IT	10	Low	High (ISO)	Health assurance	300-750	Yes	High	46 min
i09	Head of IT	19	High	High (ISO)	Finance	700	None	High	44 min
i10	Head of IT	14	Low	None	Agribusiness	2700	None	High	48 min

Appendix B: Example Interview Guidelines

In this section, we provide an example overview of (translated) questions that we asked practitioners during the interviews.

B1 Focus Group Interviews

1) CSC use cases
   a) Why should you change from a conventional certification to CSC?
   b) Are there additional stakeholders in the context of CSC compared to conventional certification contexts?
   c) How could these stakeholders jointly participate in a CSC process?
   d) Describe example situations how your company can take part in CSC processes.
2) Certification scope
   a) Which characteristics of a cloud-service are important for you and should be certified continuously?
   b) What are upper and lower boundaries of the certification / auditing frequency?
3) Architectural concepts
   a) Which components and processes are required to perform monitoring-based CSC?
   b) How can certification-relevant data be gathered and exchanged between parties?
   c) Which monitoring systems and tools do you have to monitor security, privacy, and reliability issues?
   d) Which monitoring data might be relevant for certification purposes?
   e) How can we leverage existing monitoring technologies for CSC?
   f) How should certification-relevant data be presented or transferred to certification authorities?
4) Risks and limitations
   a) Which risks and challenges bears integrating extern monitoring services or third-party services or providing certification-relevant data?

B2 Customer Interviews

1) Criteria and certification frequency
   a) Which characteristics of a cloud-service are important for you, and should be certified continuously?
   b) How often should these criteria be checked for adherence?
2) Customer integration
   a) How would you, as a customer, like to be involved in the processes of CSC?
   b) Should the process of CSC be customizable for you (e.g., defining thresholds)?
   c) Which requirements do you have concerning privacy and security of your data during CSC processes?
3) Information transfer to the customer
   a) Which information should be delivered continuously to you?
   b) Through which channels (e.g., e-mail) would you like to receive certification information?
   c) On which events would you like to be notified?

Appendix C: Methodological Guidelines for Interviews

Table C1 summarizes how we followed methodological guidelines from Sarker and Sarker (2009).

Table C1. Methodological Guidelines with Illustration (Sarker & Sarker, 2009)

Aspect of the study	Methodological considerations	Additional description	Illustration (where applicable)
Organization choice	Selecting suitable organizations to intensively study the phenomenon	We sought to interview employees from representative organizations about continuous cloud service certification. We chose organizations based on their quality/reputation to derive normative implications for future practice (Flick, 2014; Patton, 2015).	We interviewed 16 cloud service providers, twelve certification authority representatives, five cloud service consultants, and 10 cloud service customers. The practitioners who participated in our interviews had not adopted CSC when we interviewed them but had interest in or strove for innovatively developing CSC. We distributed the interview partners to gain as many insights as possible and triangulate data from different sources and perspectives on the phenomenon as Patton (2015) recommends.
Data collection	Choosing interviewees	The acting director of a participating certification authority invited suitable respondents to join the focus group interviews. We employed "snowballing" techniques to acquire additional interviewees (Patton, 2015). We acquired customer interviewees via business-oriented social networking services.	
	Conducting the interviews	Sensitivity to the following principles: 1) flexibility 2) non-direction, and 3) range (Flick, 2014).	1) We followed a semi-structured interview approach to foster discussions among participants (Myers, 2013). We rescheduled or shortened the interview agenda items to suit emerging and highly discussed themes. 2) We strived to maintain an open and nondirective style of conversation during the interviews. We applied projective techniques to uncover participants' innermost thoughts and feelings (Donoghue, 2000). 3) We varied how we conducted the interviews depending on the distribution of interviewees concerning their perspectives and interviewees' motivation and ability to elaborate on CSC issues.
	Maintaining empathetic neutrality	"Nonjudgmental form of listening" (Walsham, 1995); empathizing with interviewees' frustrations but simultaneously maintaining distance (Patton, 2015).	We strived to be patient and sympathetic listeners when interviewees expressed dissatisfaction with current certification approaches or certification authorities without elaborating on organizational stories that did not pertain to the CSC contexts.
Data analysis and representation	Triangulation	"Data triangulation" (Flick, 2014; Patton, 2015); comparing responses across respondents and time as part of the constant comparison process (Charmaz, 2000).	Whenever possible, we strived to ensure that multiple respondents suggested derived requirements (at least across different interviews).

Table C1. Methodological Guidelines with Illustration (Sarker & Sarker, 2009)

	Being suspicious about evidence	Sensitivity to possible biases in interviews (Klein & Myers, 1999).	We ensured that we recognized individuals in different positions and situations who may have brought different biases to the interview. For example, we treated some recommendations from a cloud service provider that focused on CSC's economic feasibility with caution. We also sought to validate issues that the interviewees raised with data from other interviews.
	Member checking	Validating/checking researchers' interpretations with interviewees (Flick, 2014).	We adapted interview guides if new concepts emerged and to validate derived requirements. For example, we conducted focus group interviews between November, 2014, and January, 2017. Thus, we had sufficient time to analyze the results and adjust future interview guidelines.
	Being sensitive to ethical concerns	1) Balancing anonymity and disclosure (Flick, 2014). 2) Ensuring that transcripts and other data were kept secure (Myers & Newman, 2007). 3) Respecting respondents' knowledge and time (Myers & Newman, 2007).	1) We ensured that we would not disclose the following information: organizations' identity and interviewees' personal information, identities of cloud technologies and methodologies, and specifics about interviewees' opinions about CSC. 2) We ensured that only the authors and transcribers had access to the empirical material. 3) We scheduled meetings to fit the interviewees' schedules and frequently acknowledged their efforts.

Appendix D: Overview of Meta-requirement and Guideline Refinement

Design science processes typically occur in multiple iterations, which enables one to generate a design or the IT artifact such that it fully satisfies the researchers and practitioners who subsequently use it (Beck et al., 2013; Hevner et al., 2004). Thus, the entire design process involved much repetition since, in each step, we repeated and improved on previous steps. Building on our iterative data-gathering approach, we used the interviews to evaluate our derived meta-requirements and design guidelines (see Table D1 for summary).

Table D1. Overview of Meta-requirement Refinement in Iterative Interview Phases

Interview No.	Type	Meta-requirements													
		DGL1	DGL2	DGL3	DGL4	AL1	AL2	AL3	DL1	DL2	IL1	IL2	NF1	NF2	NF3
1	Req.	new	new	new	/	new	new	new	/	new	new	new	new	/	new
	Cause	/	/	new	/	new	new	/	/	/	new	/	new	/	/
	Cons.	/	/	/	/	/	/	/	/	/	new	/	new	/	/
2	Req.	ref	ref	ref	/	ref	/	ref	new	ref	ref	ref	ref	new	/
	Cause	/	/	/	/	new	/	new	new	/	/	/	ref	new	/
	Cons.	/	/	/	/	/	/	/	new	new	new	/	new	new	/
3	Req.	ref	ref	/	new	ref	ref	/	ref	ref	ref	/	ref	/	ref
	Cause	new	/	/	new	ref	/	/	/	/	ref	/	new	/	/
	Cons.	/	/	/	/	/	/	/	new	ref	new	/	/	/	/
4	Req.	ref	ref	ref	ref	/	ref	ref	ref	ref	ref	ref	ref	/	ref
	Cause	/	/	/	new	/	ref	/	ref	/	/	/	new	/	/
	Cons.	/	/	/	/	/	/	/	/	/	/	/	/	/	new
5	Req.	ref	ref	ref	/	ref	ref	ref	/	ref	ref	ref	ref	/	/
	Cause	/	/	/	/	/	ref	/	/	/	/	/	/	/	/
	Cons.	/	/	/	/	/	/	/	/	/	new	/	/	/	/
6	Req.	ref	ref	/	/	/	/	ref	/	ref	ref	ref	/	/	ref
	Cause	/	/	/	/	/	/	new	/	/	/	/	/	/	/
	Cons.	/	/	/	/	/	/	/	/	/	ref	/	/	/	/

Req. = meta-requirement, cause = causes of requirement, cons. = consequence of (non-)adherence to requirement, new = requirement / cause / consequence was first identified, ref = additional data lead to refinement, / = no data gathered.

First, we analyzed the data gathered in the first and second focus group interviews in 2014 to derive a first set of meta-requirements and corresponding design guidelines. This set comprised 13 meta-requirements and 15 design guidelines. In addition, we identified 15 causes for meta-requirements and eight consequences of (non-)adherence to meta-requirements. For example, we derived the meta-requirement "enable data aggregation" (REQ-AL1) and developed the corresponding design requirement "perform service-focused aggregation by using agent teams" (DG-9), which a CSC monitoring system needs to "reduce data complexity" and "prevent false positives" (coded as causes).

To evaluate our initial findings, we presented and discussed this set with practitioners during the third focus group interview in 2015. Afterwards, we refined the meta-requirements and design guidelines accordingly and also identified one additional meta-requirement, one design guideline, three causes, and two consequences of meta-requirements. For example, we added the new meta-requirement "ensure data confidentiality during data gathering" (REQ-DGL4) because cloud service providers expressed concerns that one might misuse gathered data to surveil employees and, thus, violate employees' privacy or regulations.

As a second evaluation, we validated these results during the one-to-one interviews with cloud customers in 2015. Based on these interviews, we further refined our previous findings and identified an additional three causes and two consequences. For example, we refined the meta-requirement "ensure data confidentiality during data gathering" (REQ-DGL4) as interviewed cloud customers expressed their concerns about cloud service providers and certification authorities' monitoring their data. In addition, we coded a new cause named "data confidentiality during monitoring cause: customer monitoring".

Finally, in the last evaluation phase, we discussed the design concepts about CSC monitoring systems that we derived and jointly assessed on their suitability for performing continuous assessments during our focus groups in 2017. We refined our meta-requirements and design guidelines and identified two causes and one consequence when analyzing the data from these final focus groups. For example, we revised the design guideline “implement attribute-based access control” (DG-14) because focus group participants emphasized that certification authorities have different assessment scopes depending on the specific certification. Consequently, we extended agent access policies by including environmental conditions that, for instance, limit information exchange based on a specific certification and auditing scope to prevent the issue that a certification authority can inspect too much evidence.

Appendix E: Assessing Monitoring Systems

During the suggestion phase, we surveyed existing monitoring technologies to better understand monitoring systems and to determine what design CSC monitoring systems should adopt. IS theorists widely include empirical data alongside the extant literature as a means to raise the overall analysis to a higher conceptual level (Beck et al., 2013; Fernandez, 2004; Levina & Vaast, 2005). We searched for monitoring systems and tools using the Google and Bing search engines. By applying the search terms "monitoring system" and "monitoring tool", we identified 174 monitoring technologies. We found prominent monitoring technologies that the practitioners we interviewed used as well.

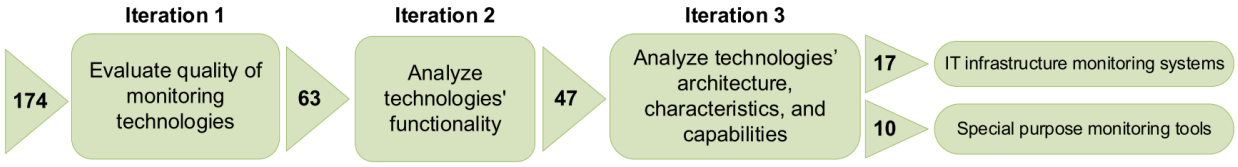

Figure E1. The Approach We Followed to Assess Existing Monitoring Systems

We assessed the monitoring technology in three iterations (refer to Figure E1). First, we assessed all 174 technologies based on different quality criteria, such as the degree of diffusion, community size, reputation, age, and maturity. These assessments reduced the relevant technologies by 63 for the next iteration. In the second iteration, we carefully analyzed the functionality that each technology provided. As a result, we removed 16 technologies because their functions had a limited scope or they constituted niche products. In the third iteration, we carefully analyzed and assessed the remaining 47 technologies regarding, for example, their architecture, data-collection capabilities, interfaces, and support for distributed monitoring. From this last assessment, we chose 17 monitoring systems and 10 monitoring tools to more deeply analyze. We assessed the documentation and system specifications of each system and deployed some technologies in test environments.

Appendix F: Design Guidelines Overview

Table F1. Guidelines for Designing CSC Monitoring Systems

ID	Design guideline	Description	Corresponding meta-requirements
Design guidelines to leverage existing monitoring technologies			
DG-1	Leveraging IT infrastructure-monitoring systems	Integrate IT infrastructure monitoring systems as they typically monitor applications, services, operating systems, system metrics, and network infrastructures.	REQ-DGL1 REQ-DGL2 REQ-NF1 REQ-NF2
DG-2	Leveraging monitoring tools	Leverage monitoring tools to gain detailed insights about cloud service and process operations that surpass the capabilities of infrastructure monitoring systems.	
DG-3	Leveraging monitoring plugins	Leverage monitoring plugins to extend monitoring technologies' capabilities.	REQ-DGL1 REQ-NF2
Design guidelines to access external interfaces			
DG-4	Integrate external databases	Connect to databases that provide valuable information.	REQ-DGL1
DG-5	Access interfaces of subproviders	Access service interfaces of subproviders to gather information about certification adherence that subproviders provide.	REQ-DGL3
DG-6	Monitor services of subproviders	Incorporate means to measure services that subproviders provide (e.g., test-based CSC methodologies).	
Guidelines to apply and operate an agent-based architectural model			
DG-7	Apply an agent-based architecture model	Dispatch certification agents to different monitoring technologies to gather certain certification-relevant data.	REQ-DGL1 REQ-DGL2 REQ-NF1 REQ-NF2
DG-8	Equip agents with security policies	Agents can receive security policies to ensure that they comply with data-protection regulations or customer requirements.	REQ-DGL4
DG-9	Perform service-focused aggregation by using agent teams	Organize agents as hierarchically structured teams to aggregate data across monitoring technologies.	REQ-AL1
DG-10	Store meta-information about agent operations	Provided meta-information should comprise data on 1) what was monitored, 2) how it was monitored, 3) when it was monitored, 4) who performed the monitoring, and 5) the monitoring results.	REQ-NF3
Guidelines to incorporate and secure flexible data storages			
DG-11	Incorporate flexible data storage technologies to store and archive data	Incorporate flexible data storage technologies to easily adjust data schemes and store additional data or results from new data analysis operations in, for example, a NoSQL document database.	REQ-DL1 REQ-NF2
DG-12	Implement means to guard data against improper modification	Integrate a trusted third-party module that provides secure log encryption functions, establish a chain of custody for digital evidence, or apply other techniques from the cloud forensics domain to prevent internal log manipulation.	REQ-DL2

Table F1. Guidelines for Designing CSC Monitoring Systems

Guidelines for secure data exchange			
DG-13	Embed data-masking techniques	Embed data-masking techniques such as encryption, substitution, and nulling out.	REQ-DGL4 REQ-AL2 REQ-IL2
DG-14	Implement attribute-based access control and define access policies	Implement an attribute-based access control to filter data according to an authority's needs and privileges. Specify access policies that define the access rules for the allowable subjects, operations, and environmental conditions to the object.	REQ-AL2 REQ-AL3 REQ-IL2
DG-15	Implement encrypted data-transmission means	Provide functionalities to automatically generate reports based on analyzed data and automatically transmit encrypted reports about defined points in time.	REQ-IL1 REQ-IL2
DG-16	Implement secure data-providing interfaces	Implement passive interfaces that enable certification authorities to access data.	

Appendix G: Monitoring Plugin Evaluation

Monitoring technologies prominently feature an extendable architecture, which provides easy integration with third-party plugins. Typically, one can write plugins in various programming languages (e.g., Perl, bash script, Java, PHP or Python). One can access plugin source code over sharing platforms, and these plugins fall under open source license, which allows programmers to modify or adapt them to a particular monitoring scenario. Most monitoring systems work with a broad range of plugins from different platforms (i.e., Icinga 2, Zabbix and Opsview work with most available plugins for Nagios). In this section, we briefly evaluate available plugins at Nagios' Exchange Platform in their relation to support monitoring-based CSC. Available plugins on Nagios' Exchange platform enable administrators to extend core functionalities, to use alternative user interfaces, to integrate new components (e.g., monitoring agents), and so on. Table G1 further illustrates example plugins from Nagios' Exchange Platform that might be useful in the CSC context.

Table G1. Outline of Nagios Plugin and Add-on Examples

Subcategory (no. of plugins)	Description	Example plugin
Cloud (38)	Nagios plugins for monitoring a cloud infrastructure.	Nagios check scripts to monitor the cloudstack.
Network and systems management (131)	Nagios plugins to monitor network and systems management software.	Check the state and bandwidth of specified interface of Cisco devices.
Security (107)	Nagios plugins for monitoring security software.	Check for the Heartbleed vulnerability.
System metrics (373)	Nagios plugins for monitoring different types of system metrics (e.g., disk, memory, CPU, etc.).	Check CPU performance statistics.
Monitoring agents (40)	Agents that allow Nagios to monitor remote systems.	A cross-platform monitoring agent that runs on Windows, Linux/Unix, and Mac OS/X machines.
Active checks (48)	Add-ons for managing active checks.	Check ping time.
Reactor add-ons (54)	Various add-ons (conditions, actions, etc.) for Nagios Reactor.	Compare results of the output returned from a script to a specified exit code.
Notifications (69)	Add-ons for sending notifications via mail, phone, etc.	Extend Nagios with voice, SMS, and push-notification capabilities.
APIs (46)	Nagios access and data export interfaces.	Send status of hosts and services using JSON format.
Frontends (123)	Alternative interfaces—GUIs and CLIs for Nagios.	Allow external company users to view infrastructure status information without giving them access to the CGI interface.

CSC requires gathering certification-relevant data; hence, organizations can select a set of plugins from the Nagios Exchange Platform and install them in their Nagios deployment to collect such information. For example, using the plugin Nagios Remote Plugin Executor (NRPE) enables cloud service providers to monitor remote machine metrics (e.g., disk usage or CPU load) (see Figure G1). A SSL connection secures data between monitoring host and remote plugins. Plugins on the remote host can gather arbitrary information about the certification. The monitoring system can then evaluate and process this information provide it to the certification authority.

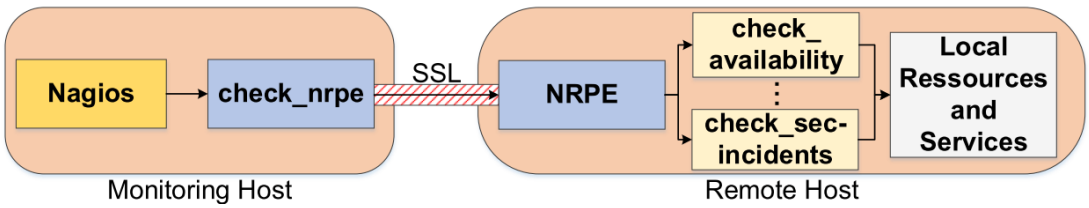

Figure G1. Gather Certification-relevant Data Using NRPE (Adapted from Nagios Enterprises, 2016)

As another example, cloud service providers can provide certification-relevant data to certification authorities by providing extended graphical user interfaces through plugins such as Nagex. The Nagex plugin provides a Nagios Web dashboard that enables internal and external stakeholders to view infrastructure status information without giving them access to the CGI interface. Furthermore, providers can integrate available plugins into a CSC monitoring system to provide data transmission interfaces for certification authorities as well. More than 48 plugins currently offer various functionalities to export data (e.g., in JSON or XML format) and to use SOAP or RESTful interfaces for exchanging data. Similarly, available plugins allow remote agents and applications to submit commands and passive checks on the monitoring server. Nonetheless, data interfaces are designed for only internal data communication; thus, providers need to further adjust these interfaces so that they can securely and confidentially transmit data with third-party authorities.

About the Authors

Sebastian Lins is a PhD student at the Research Group Critical Information Infrastructures, Institute of Applied Informatics and Formal Description Methods, Karlsruhe Institute of Technology, Germany. His main interests in the field of information systems research are the (continuous) certification of cloud services and distributed ledger technology as well as understanding and enhancing the effectiveness of IT certifications.

Stephan Schneider is a postdoctoral researcher at the Research Group Critical Information Infrastructures, Institute of Applied Informatics and Formal Description Methods, Karlsruhe Institute of Technology, Germany. His research focuses on strategic decision-making in cloud sourcing projects as well as on cloud security and certification of cloud computing infrastructures.

Jakub Szefer is a professor at the Department of Electrical Engineering, Yale University, CT, USA. His research focuses on computer architecture and security related topics, including secure hardware-software architectures for servers and mobile devices, cloud computing security and many-core processor architectures with security features.

Shafeeq Ibraheem is a student researcher at the Department of Electrical Engineering, Yale University, CT, USA. His research focuses on computer architecture and security related topics.

Ali Sunyaev holds the Research Group Critical Information Infrastructures in the Institute of Applied Informatics and Formal Description Methods, Karlsruhe Institute of Technology, Germany. His research interests include design and management of information systems, cloud computing, health IT and information privacy. His work has been published in international journals such as *Journal of Information Technology*, *Communications of the ACM*, *IEEE Software*, *IEEE Transactions on Cloud Computing* and *Communications of the Association for Information Systems*.