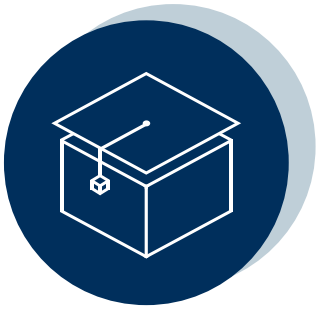

BISE Student

https://bise-student.io

MASTER'S THESIS

Automated Cloud Service Certification

Publication Date: 2022-02-22

Author
Sebastian LINS
University of Cologne
Cologne, Germany
lins@kit.edu
0x4bFa4752323dB7d2E8B0F774288A6474650be046

Abstract

Cloud service certifications (CSC) are good means to establish trust, increase transparency of the cloud market, and allow providers to improve their processes and systems. Several CSCs, such as 'CSA STAR', 'EuroCloud Star Audit' or 'TÜV Cloud Security', have recently evolved. However, cloud services (CSs) are part of an ever-changing environment, resulting from fast technology life cycles and inherent cloud computing characteristics, like on-demand provisioning and entangled supply chains. Hence, such long validity periods may put in doubt the reliability of issued certificates. Conditions and requirements of CSCs may no longer be met throughout these periods, for instance, due to configuration changes or major security incidents. Continuous monitoring and auditing of selected certification criteria (henceforth defined as dynamic certification) is required to assure continuously reliable and secure CSs, and to establish a trustworthy CSC, after the initial certification process is accomplished. Dynamic certification...

Keywords: cloud computing, continuous certification, certification
Methods: literature review, interview, field observation

Submission Date: 2022-02-22
Submission Contract: 0x53787C2b2E69feaB1712512f70Ae50D47fa66630

Sebastian Lins

Masterarbeit
im Fach Allgemeine Wirtschaftsinformatik

Automated Cloud Service Certification

Themensteller: Prof. Dr. Ali Sunyaev

Vorgelegt in der Masterprüfung
im Studiengang Information Systems
der Wirtschafts- und Sozialwissenschaftlichen Fakultät
der Universität zu Köln

Köln, Oktober 2014

Table of Contents

Index of Abbreviations

ADM	Audit Data Mart
CA	Continuous auditing
CAATTs	Computer-assisted auditing technologies and tools
CM	Continuous monitoring
CORBA	Common Object Request Broker Architecture
CS	Cloud Service
CSC	Cloud Service Certification
DA	Digital Agent
DSS	Decision Support System
EAM	Embedded Audit Module
GAS	Generalized Audit Software
IaaS	Infrastructure as a Service
iM&A-Department	Internal Monitoring and Auditing Department
IPS	Intra-platform Surveillance
JADE	Java Agent Development Framework
LAS	Local Application Surveillance
MCL	Monitoring and Control Layer
PaaS	Platform as a Service
SaaS	Software as a Service
SLA	Service Level Agreement

SOAP	Simple Object Access Protocol
VM	Virtual Machine
XBRL	Extensible Business Reporting Language
XBRL GL	Extensible Business Reporting Language Global Ledger

Index of Illustrations

Index of Tables

1. Introduction

An increasing number of organizations outsource their data, applications and business processes to the cloud, empowering them to achieve financial and technical benefits.[1] Cloud computing enables ubiquitous, on-demand provisioning of up to date computing resources and services, like networks, applications, or storage on a pay-per-use basis.[2] However, some organizations are still hesitant to adopt cloud services (CS) because of security, privacy, and reliability concerns regarding provisioned CSs as well as doubts about the trustworthiness of their cloud service provider (CSP).[3] Thus, CSPs have to address these concerns and prove their credibility to increase adoption of potential customers.

Cloud service certifications (CSC) are good means to establish trust, increase transparency of the cloud market, and allow providers to improve their processes and systems.[4] Several CSCs, such as 'CSA STAR'[5], 'EuroCloud Star Audit'[6] or 'TÜV Cloud Security'[7], have recently evolved.[8] These CSCs attempt to assure a high level of security, reliability, and legal compliance, for a validity period of one to three years. However, CSs are part of an ever-changing environment, resulting from fast technology life cycles and inherent cloud computing characteristics, like on-demand provisioning and entangled supply chains.[9] Hence, such long validity periods may put in doubt the reliability of issued

---

[1] Cf. Cimato et al. (2013), p. 101.

[2] Cf. Mell, Grance (2011), p. 2.

[3] Cf. concerning security Kalloniatis, Mouratidis, Islam (2013), p. 300, concerning privacy and reliability Subashini, Kavitha (2011), p. 2, 6, 7, and regarding trustworthiness Khan, Malluhi (2010), p. 20.

[4] Cf. Schneider et al. (2014), p. 1, Cimato et al. (2013), p. 100, concerning transparency Sunyaev, Schneider (2013), p. 34, and regarding process and system improvements Federal Office for Information Security (2011), p. 21.

[5] See Cloud Security Alliance (n.y.).

[6] See EuroCloud Europe (n.y.).

[7] See TÜV Rheinland (n.y.).

[8] Cf. this and the following sentence Schneider et al. (2014), p. 1 and Schneider, Lansing, Sunyaev (2013), p. 13-14.

[9] Cf. Kunz, Niehues, Waldmann (2013), p. 522, Bezzi, Kaluvuri, Sabetta (2011), p. 40-41, Cimato et al. (2013), p. 100, and European Network and Information Security Agency (2013), p. 32-33.

certificates.[10] Conditions and requirements of CSCs may no longer be met throughout these periods, for instance, due to configuration changes or major security incidents.[11] Thus, continuous monitoring and auditing of selected certification criteria (henceforth defined as dynamic certification) is required to assure continuously reliable and secure CSs, and to establish a trustworthy CSC, after the initial certification process is accomplished.

Dynamic certification is still in its beginning, so in a first step CSC criteria must be evaluated and classified in order to assess, whether or not a continuous monitoring and auditing is required. For instance, a criterion might be audited on a high frequency when a CSP can benefit (e.g., cost reductions) from discontinuing adherence to it. Moreover, dynamic certifications cannot be carried out solely manually due to high costs and considerable expenditures, thus requiring (semi) automated methods.[12] However, literature concerning the usage of (semi) automated methods is scarce.[13] For example, Chieu et al. (2012) show how to automatically validate the configuration of activated CSs regarding security requirements,[14] and Liu et al. (2013a) present an approach to audit and validate the integrity of data stored in a cloud.[15] Nonetheless, the automation of processes is limited due to their complexity and interconnectedness.[16] Likewise, unstructured and human-driven interactions may complicate the automation of privacy and data security certifications. Thus, automated methods have to be evaluated regarding their practical feasibility.

---

[10] Cf. also Cimato et al. (2013), p. 101, Kunz, Niehues, Waldmann (2013), p. 522, European Network and Information Security Agency (2013), p. 5,24, and Schneider, Lansing, Sunyaev (2013), p. 16.

[11] Cf. Windhorst, Sunyaev (2013), p. 414, and European Network and Information Security Agency (2013), p.18-19.

[12] Cf. Kunz, Niehues, Waldmann (2013), p. 522, Bezzi, Kaluvuri, Sabetta (2011), p. 41, Brown, Wong, Baldwin (2007), p. 21, Schneider, Lansing, Sunyaev (2013), p. 16, and Woodroof, Searcy (2001), p. 1.

[13] Cf. Aceto et al. (2013), p. 2094, Bernnat et al. (2012), p. 13, Accorsi, Lowis, Sato (2011), p. 145, and Schneider et al. (2014), p. 2.

[14] Cf. Chieu et al. (2012), p. 286-289.

[15] Cf. Liu et al. (2013a), p. 1.

[16] Cf. this and the following sentence Schneider et al. (2014), p. 3, Doganata, Curbera (2009), p. 310-311, and Kunz, Niehues, Waldmann (2013), p. 522.

To address these research gaps, this thesis aims to answer the question *Which automated monitoring and auditing methods can be used in practice to assure ongoing CSC adherence?* (RQ 1). To answer this question, this thesis first focuses on the question *Which CSC criteria should be continuously monitored and audited?* (RQ 2). Moreover, for each CSC criterion an auditing frequency is determined (e.g., monthly or quarterly). After defining a set of CSC criteria appropriate, automated monitoring and auditing methods are identified. In addition, these methods are evaluated regarding their applicability in CS contexts, hence, answering the question *Which (semi) automated monitoring and auditing methods exist and are applicable in the context of cloud computing?* (RQ 3). Identified methods are discussed with practitioners involved in conducting CSC audits, to ensure applicability in auditing practice, therefore answering the question *Which monitoring and auditing methods can be applied in practice?* (RQ 4). Based upon these discussions and assessments, design recommendations and guidelines for dynamic CSCs are derived, and a first conceptual model of dynamic CSC is developed to answer the research question *What needs to be considered when designing dynamic certifications?* (RQ 5). Finally, on the basis of this model of dynamic certification, CSC criteria are mapped to applicable methods, hence, answering the question *Which CSC criteria can be monitored and audited by which methods?* (RQ 6).

To answer these questions, existing CSC criteria catalogs are assessed and aggregated, a systematic literature review is performed to identify existing methods, interviews with practitioners are performed to assess applicability of methods and to derive design recommendations, and finally CSC criteria and applicable methods are mapped.

The remainder of this thesis is structured as follows. Section 2 proceeds with a theoretical background on cloud computing, CSCs, and a differentiation between continuous monitoring and auditing. Section 3 outlines the applied research approach. Further on, section 4 presents the results of the assessment which CSC criteria should be continuously monitored and audited. Section 5 and 6 summarize identified auditing and monitoring methods, and present insights regarding their practical applicability. Finally, section 7 depicts derived design recommendations and guidelines for dynamic certification, and presents a first conceptual model of dynamic certification, followed by a conclusion in section 8.

2. Background

2.1 Cloud Computing

Cloud Computing enables "ubiquitous, on-demand network access to a shared pool of configurable computing resources […] that can be rapidly provisioned and released with minimal management effort or service provider interaction."[17] These resources refer, for instance, to hardware, development platforms, and applications.[18] Cloud computing entails five essential characteristics, that are: the provision of (i) on-demand self-service access to (ii) virtualized, shared, and managed IT resources that are (iii) scalable on-demand, (iv) available over a network, and (v) priced on a pay-per-use basis. These characteristics challenge current assessment and certification processes, and make it difficult to certify CSs.[19]

Cloud computing is composed of three service models: Infrastructure as a Service (IaaS), Platform as a service (PaaS), and Software as a Service (SaaS).[20] IaaS refers to offering processing, storage, networks, and other basic computing resources. The CS customer is able to deploy and run operating systems and applications on these resources. Likewise, PaaS refers to providing CS customers the capability to deploy individual or acquired applications. Lastly, SaaS enables CS customers to use applications running in the cloud. Furthermore, deployments models of CS can be differentiated into private, public, community and hybrid cloud. In a private cloud, the "cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers […] [,] may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises."[21] Similar, a public cloud infrastructure is provisioned for open use by the general public, a community cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations.[22] Lastly, a hybrid cloud is a composition of at least two previously described deployment models.

---

[17] Mell, Grance (2011), p. 2.

[18] Cf. this and the following sentence Mell, Grance (2011), p. 2-3.

[19] Cf. Windhorst, Sunyaev (2013), p. 413, and Kaliski, Pauley (2010), p. 2-4.

[20] Cf. this and the following sentences Mell, Grance (2011), p. 2-3.

[21] Mell, Grance (2011), p. 3.

[22] Cf. this and the following sentence Mell, Grance (2011), p. 3.

2.2 Cloud Service Certification

Several CSCs have emerged and cloud certification schemes in particular (e.g., ISO 27017) are currently under development.[23] A *certification* is defined as a third party attestation of products, processes, systems or persons that verifies the conformity to specified requirements.[24] As a result of this certification process, a formal written certificate is awarded.[25]

The adoption of CSCs can support (potential) cloud customers by providing additional information about CSs, thus customers don't have to rely solely on information from a CSP.[26] Customers can compare the certification results of different CSPs to gain a market overview and make a better provider selection, as well.[27] In general, CSCs are good means to establish trust in dynamic cloud environments.[28] Furthermore, CSCs allow providers to improve their processes and systems.[29] The European Union has issued the cloud strategy 'Unleashing the Potential of Cloud Computing in Europe' in order to accelerate and increase the usage of cloud computing in Europe.[30] According to Gartner Inc., cloud certification will become the norm for cloud offerings.[31] However, existing CSCs are not yet fully matured, and an established and recognized CSC standard is missing.[32]

Nonetheless, existing CSCs represent a backward look at the fulfillment of technical and organizational measures at the time of their issuing.[33] Since CSs are part of an ever-changing environment, resulting from fast technology life cycles and inherent cloud computing characteristics, their ongoing reliability might be questioned. To increase the

---

[23] Cf. European Network and Information Security Agency (2013), p. 3-4, 6-7, and International Organization for Standardization ISO/IEC 27017 (n.y.).

[24] Cf. International Organization for Standardization ISO/IEC 17000:2004 (n.y.), p. 3-4.

[25] Cf. Bruhn (2008), p. 424.

[26] Cf. Windhorst, Sunyaev (2013), p. 412.

[27] Cf. Rannenberg (2000), p. 2.

[28] Cf. Khan, Malluhi (2010), p. 24-25, Schneider et al. (2014), p. 1, and Cimato et al. (2013), p. 100.

[29] Cf. Federal Office for Information Security (2011), p. 21.

[30] Cf. European Network and Information Security Agency (2013), p. 2.

[31] Cf. Heiser, Nicolett (2008), p. 5.

[32] Cf. Schneider et al. (2014), p. 1, and Schneider, Lansing, Sunyaev (2013), p. 13-14, and concerning missing standard Bernnat et al. (2012), p. 2.

[33] Cf. this and the following two sentences Windhorst, Sunyaev (2013), p. 414.

trustworthiness of issued certifications, and to assure continuously reliable and secure CSs, dynamic certifications are introduced. A dynamic certification comprises a variety of mechanisms, techniques, and activities. First, an auditor certifies a CS according to a CSC catalog. Second, continuous monitoring and auditing (see section 2.3 for a detailed description) have to be performed, to assure ongoing adherence of selected certification criteria. Third, methods and mechanisms have to be implemented to cope with identified deviations, triggered alerts or cases of non-adherence. Finally, auditors have to provide cloud customers with ongoing information about certification (non-) adherence.

2.3 Differentiation between Continuous Monitoring and Auditing

Continuous monitoring (CM) and continuous auditing (CA) are important processes in the context of dynamic certification. Both terms are often used interchangeably in literature,[34] however, throughout this thesis a precise distinction is made between these concepts.

Continuous monitoring is defined as ongoing "observance and analysis of the operational states of systems [and applications] to provide decision support regarding situational awareness and deviations from expectations."[35] CM comprises several domains, for instance, network, configuration, vulnerability and incident management as well as malware detection.[36] Throughout this thesis, CM is performed by CSPs or a third party (e.g., a third party providing monitoring as a service capability) to continuously track, measure, and assess system and application behavior to detect and diagnose problems, and to provide information for further analyses.

In contrast, a *continuous audit* is defined as "a methodology that enables independent auditors to provide written assurance on a subject matter, for which an entity's management is responsible, 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."[37] Thus, CA enables auditors to react to changes or events concerning the subject

---

[34] Cf. Brown, Wong, Baldwin (2007), p. 1, and Hardy (2011), p. 2.

[35] Mell et al. (2012), p. 9.

[36] Cf. Mell et al. (2012), p. 10.

[37] CICA/AICPA (1999), p. xiii.

matter, and to adjust their auditing reports based on the assessment of these changes and events.

Performing CM by CSPs forms a prerequisite for auditors to perform efficient CA, since monitoring capabilities of auditors might be limited due to technical, organizational and legal reasons. First, technical limitations and barriers might hamper auditors to gather necessary certification information by themselves. Integration of additional monitoring methods requires extensive modifications to the auditees' systems, which can be quite expensive to realize, especially post hoc.[38] Matching auditees' heterogonous data formats and legacy systems can be complex and expensive for auditors as well.[39] Furthermore, integrated modules, developed for one CSP, might not be easily utilized for another provider.[40] Second, efficient CM requires extensive knowledge about organizational processes and structures as well as system architectures. Moreover, auditees are not necessarily willing or obligated, and may be even resisting to integrate auditors' techniques into their systems.[41] Third, due to privacy, regulatory and legal requirements gaining access to required data and systems might be limited for external auditors.[42] Hence, dynamic certification requires on the one hand CSPs to implement suitable CM methods to provide necessary monitoring information. On the other hand, it requires auditors to implement CA methods to continuously analyze and assess provided, and additional gathered information to assure certification adherence.

The frequency of performing CM and CA depends on the observed subject matter. Thus, the frequency of performing monitoring and auditing operations might range from real-time to daily, weekly, or monthly.[43] To be efficient and cost effective, CM and CA require a high degree of standardization and automation.[44] In addition, automation enables

---

[38] Cf. Murthy, Groomer (2004), p. 148-149.

[39] Cf. Du, Roohani (2007), p. 137, and Flowerday, Blundell, Von Solms (2006), p. 328.

[40] Cf. Du, Roohani (2007), p. 136.

[41] Cf. Alles et al. (2006), p. 146, Du, Roohani (2007), p. 136, and Groomer, Murthy (1989), p. 68.

[42] Cf. Du, Roohani (2007), p. 136-137.

[43] Cf. Marques, Santos, Santos (2013), p. 305 cited by Vasarhelyi, Alles, Williams (2010).

[44] Cf. Kunz, Niehues, Waldmann (2013), p. 522, Bezzi, Kaluvuri, Sabetta (2011), p. 41, Brown, Wong, Baldwin (2007), p. 21, Schneider, Lansing, Sunyaev (2013), p. 16, Chan, Vasarhelyi (2011), p. 155, and Woodroof, Searcy (2001), p. 1.

efficient validation of specific requirements (e.g., data location compliance),[45] the automated management of certificates for CSPs,[46] and increases monitoring and auditing efficiency.[47]

CSP as well as auditors can realize several benefits, when adopting and implementing CM and CA. First of all, internal processes and systems can be improved when implementing suitable CM techniques.[48] Moreover, a CSP can actively detect and investigate exceptions as they occur rather than to react after the exception has long occurred.[49] Hence, CM can be considered as proactive and enables corrective action to be taken as soon as a problem is detected. Likewise, automated CA is more cost-effective, by enabling auditors to test larger samples, and examine data faster and more efficiently, compared to their manual predecessors.[50] More importantly, through timely detection and continuous assurance of certification adherence, CA can improve the trustworthiness of auditors' CSC.[51] Notwithstanding the benefits of CA, recent surveys reveal that from a practical perspective, CA is still maturing and wide adoption is missing.[52]

---

[45] Cf. Kunz, Niehues, Waldmann (2013), p. 522.

[46] Cf. Bezzi, Kaluvuri, Sabetta (2011), p. 41.

[47] Cf. Woodroof, Searcy (2001), p. 1, and Manson, McCartney, Sherer (2001), p. 126.

[48] Cf. Brown, Wong, Baldwin (2007), p. 21.

[49] Cf. this and the following sentence Chan, Vasarhelyi (2011), p. 154-155, and Flowerday, Blundell, Von Solms (2006), p. 326.

[50] Cf. Brown, Wong, Baldwin (2007), p. 2, 21, Rezaee et al. (2002), p. 151, and Woodroof, Searcy (2001), p. 1-2.

[51] Cf. Windhorst, Sunyaev (2013), p. 414.

[52] Cf. Vasarhelyi et al. (2012), p. 268, 279, and Lin, Lin, Liang (2010), p. 415.

3. Research Approach

The applied research approach is divided into five steps, accompanied by a cooperation with *Certify*[53] to gain practical insights, and access to practical knowledge. Figure 3-1 illustrates the research approach.

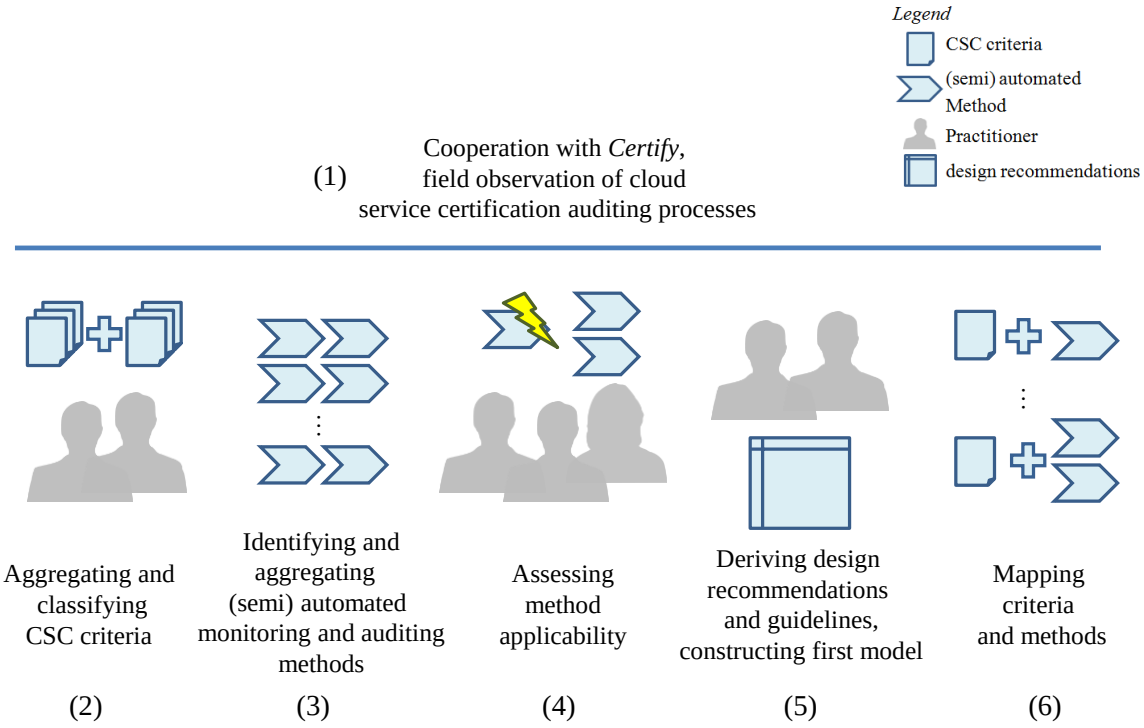

Figure 3-1 Illustration of applied research approach.

3.1 Cooperation with Certify

To drive the adoption and application of dynamic certification in practice, it has to be ensured that dynamic certification is applicable in practice. Only practitioners can assess possible obstacles regarding the automation of CSC audits, and application of CM and CA methods due to their experience in conducting CSC audits. Hence, to gain practical insights into CSC auditing processes, and to get access to knowledge and experience of practitioners, the author cooperated with *Certify*.

*** further information on Certify was deleted to preserve confidentiality***

Due to the cooperation, the author was able to gain practical insights into CSC processes, and to get access to knowledge and experience of practitioners. First, several workshops with practitioners were conducted (see section 3.2). Second, the author was able to

[53] Company is pseudonymized to preserve confidentiality.

participate in an one-site CSC audit. During this two-day audit, technical analyses by means of interviews with auditee's were observed. This observation deepened the author´s understanding of how CSC audits are performed in practice. Moreover, insights concerning the design of dynamic certifications were gained and will be discussed in section 7. Third, to evaluate the practical applicability of identified (semi) automated methods, interviews with practitioners from *Certify* were conducted (see section 3.4). Practitioners who participated in workshops and interviews have several years of experience in conducting cloud certification audits. Table 3-1 provides an anonymized overview of practitioners participating in the workshops and interviews. Throughout this thesis, statements and opinions from practitioners will be marked with their corresponding identifier '[i0X]'.

Identifier	**Position**	**Participated in**
i01	Principal Consultant	Workshops
i02	Security Analyst	Workshop
i03	Security Analyst	Workshop, Interview
i04	Research Analyst	Interviews

Table 3-1 Anonymized overview of practitioners participating in workshops and interviews.

3.2 Aggregating and Classifying Cloud Service Certification Criteria

When analyzing the application of dynamic certification in cloud computing context, first a collection of CSC criteria has to be defined and further analyzed. For each criterion of this collection it has to be determined whether or not a high frequency monitoring and auditing is required after the initial certification process was accomplished. To take a variety of CSC criteria into consideration, the *Certify* requirements catalog and a CSC criteria taxonomy developed by Schneider et al. (2014) were included in this classification step. The taxonomy for CSC criteria is based upon seven well-known and established security standards, cloud computing frameworks (e.g., Badger et al. (2012) and Federal Office for Information Security (2011)), and expert interviews.[54] The *Certify*

[54] Cf. Schneider et al. (2014), p. 1.

requirements catalog comprises 273 CSC criteria, whereas the taxonomy by Schneider et al. (2014) consists of 328 CSC criteria.[55]

To construct a comprehensive collection of CSC criteria for the succeeding analyses, the author compared and merged the *Certify* requirements catalog and CSC criteria taxonomy according to the following steps. First, in an iterative process, the entire criteria contained in the taxonomy were compared with a *Certify* criterion (see Figure 3-2). A taxonomy criterion was assigned to a *Certify* criterion, if both criteria descriptions matched. Matched criteria descriptions were analyzed in detail, to aggregate them or to extend the corresponding *Certify* criterion description. Once the entire taxonomy criteria were compared, a further iteration started with the next criterion from the *Certify* catalog. Second, each taxonomy criterion that was not assigned to a *Certify* criterion was added to the *Certify* requirements catalog. Finally, each assignment, aggregation and addition was reviewed to ensure validity and integrity. As a result, the revised *Certify* requirements catalog comprised 414 certification criteria.

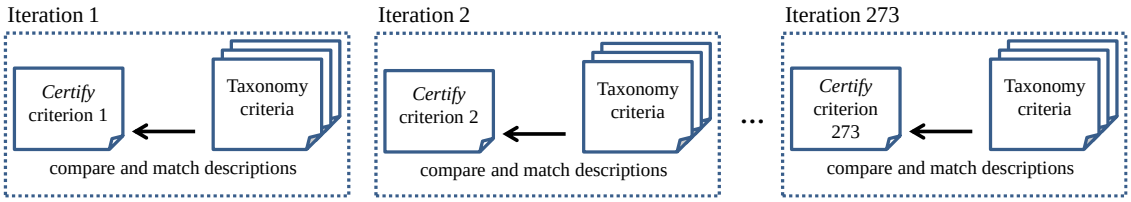

Figure 3-2 Iterative comparison and matching of certification criteria.

The author further classified each criterion of this collection whether or not a high frequency monitoring and auditing is required. This classification is based upon a checklist listed in table 3-2. This checklist was built upon analyzing the criteria definitions and the experience in classifying criteria gained in previous work (see Lins (2014)). It consists of questions to evaluate whether or not a criterion needs a high frequency monitoring and auditing, and corresponding attributes. Each criterion was individually assessed in regard to the checklist questions. If a checklist question could be affirmed, the corresponding attribute(s) were assigned to the criterion, and the criterion was marked as a candidate for continuous monitoring and auditing. Consequently, not every CSC criterion needs to be

[55] Cf. Schneider et al. (2014), p. 5.

continuously monitored or audited. Moreover, for each criterion an estimated auditing frequency (e.g., monthly or quarterly) was notated.

Attribute	Question
Regularity	Does the criterion imply actions, which have to be performed on a regular basis (e.g., monthly review of firewall rules)?
Internal Changes	Is the criterion affected by internal changes (e.g., cloud configuration changes or aging of components)?
External Changes	Is the criterion affected by external changes (e.g., new customers or supplier changes)?
Critical Cloud Characteristic	Does the criterion require that critical cloud characteristics (e.g., availability, integrity, or scalability) are assured?
Critical Security Criterion	Does the criterion necessitate managing critical security measures and issues (e.g., performing a security check when integrating new cloud components or vulnerability scans)?
Benefits due to discontinuity	Can the cloud provider benefit (e.g., cost reductions) from discontinuing adherence to the criterion?
Transparency	Does the criterion require cloud service providers to notify cloud customers or third parties on emerged events or performed actions (e.g., informing about security incidents or providing information about updating business continuity plans)?

Table 3-2 Checklist to assess whether or not a criterion has to be continuously monitored and audited.

To increase validity and reduce subjectivity of the initial criteria classification performed by the author, three workshops were held with experts from *Certify*. In these workshops, practitioners [i01, i02, i03] and the author discussed the criteria aggregation, classification and estimated auditing frequency. The workshops lasted about two hours on average. In these workshops each criterion that was changed or inserted during the criteria aggregation process was discussed in conjunction. This discussion led to a reduction of certification criteria, since some newly added criteria were already covered by other included criteria, some criteria were merged, or assessed as not being feasible. As a result, the final *Certify* requirements catalog contains 326 certification criteria in total. A list of the criteria is outlined in appendix F.

Furthermore, practitioners [i01, i03] and the author jointly assessed the initial classifications whether or not a high frequency monitoring and auditing is required. First, the classification checklist was presented and discussed. Workshop participants agreed on the

validity and integrity of the checklist [i01, i03]. Afterwards, each classified criteria was jointly discussed. In total, 78 out of 326 certification criteria were marked as a candidate for continuous monitoring and auditing. Finally, the proposed auditing frequency was adjusted based upon their experience from conducting CS audits and technical knowledge. The results of the joint assessments and classification as well as classification examples are presented in section 4.

3.3 Identifying and Aggregating (semi) automated Monitoring and Auditing Methods

This thesis extends previous work regarding the identification of (semi) automated monitoring and auditing methods. Lins (2014) and Thiebes (2014) performed a systematic literature review to identify (semi) automated monitoring and auditing methods. In total, 35 methods were identified and further grouped into six clusters. Moreover, a first evaluation concerning their applicability in the context of cloud computing was made. These methods and evaluations were included in this paper. Previously identified methods and corresponding publications are listed in appendix A, in order to differentiate them precisely against new methods identified in this thesis. The corresponding publications were re-read, and methods were further analyzed concerning their applicability in cloud computing contexts during this work.

Based upon the experience gained in the previous research, the author performed a new review with an adjusted search string, and search parameters to further extend the set of identified methods. Hence, to identify publications addressing (semi) automated monitoring and auditing methods, a systematic scientific database search in the following databases that cover a wide range of journals and conferences was applied (i.e., they cover the top computer science and information systems journals and conferences): AIS Electronic Library, ACM Digital Library, EBSCOhost, Emerald Insight, IEEE Xplore, ProQuest, and ScienceDirect. There are a variety of terms (interchangeably) used to describe activities in this research area, such as continuous assurance, continuous auditing, continuous monitoring, and real-time auditing.[56] Therefore, and to cover a broad set of publications, each database was searched with the following string in title and keywords:

[56] Cf. Hardy (2011), p. 2, and Brown, Wong, Baldwin (2007), p. 1.

(certif OR audit* OR monitor* OR assur*) AND (continuous* OR permanent* OR dynamic* OR automat* OR realtime OR computerized OR (machine AND readable) OR computer AND (assisted OR aided))*

To assure transferability to the cloud computing context, the search was limited to sources published after 1980, because in 1981 the concept of TCP/IP was introduced.[57] Furthermore, screening of randomly sampled articles that matched the keywords and were published before 1980 did not yield relevant articles. The search was limited to peer-reviewed articles, when possible. Because of this broad search string, 10,142 articles were identified in the initial search.

The relevancy of each article was assessed by analyzing title, abstract, and keywords in order to identify possibly relevant publications. If any indication for relevancy appeared, the article was marked for further processing. A large number of publications from medical (e.g., glucose and heart rate monitoring), environmental (e.g., water and vegetation monitoring), sensor network (e.g., energy certification of wireless sensor networks), speech recognition (e.g., continuous speech analysis), and power supply contexts (e.g., power monitoring) were identified through the broad initial search and were then excluded, leading to a remaining set of 151 possibly relevant articles. Afterwards, the relevance of the remaining 151 articles was validated in detail. Inclusion and exclusion criteria for this relevance validation are listed in table 3-3. Research that does not propose (semi) automated methods (15), is not applicable to cloud computing (13), or off-topic (56) were excluded. Moreover, duplicates (6) and non-research articles were excluded (5). Furthermore, a number of publications were marked exclusively for backward analysis (12). These publications are reviewing literature on automated monitoring and auditing methods. Nonetheless, reviewed and cited sources seemed to be relevant for this thesis, thus these publications were included in the backward analysis. The relevancy assessment led to a set of 44 relevant articles. Furthermore, 25 relevant articles were added from Lins (2014) and Thiebes (2014), leading to at total set of 69 articles.

[57] Cf. Postel (1981), and Badach, Hoffmann (2007), p. 94, 107.

Inclusion criteria	Exclusion criteria
(semi) automated monitoring methods	Published before 1981
(semi) automated auditing methods	No (semi) automated methods
	Not applicable to cloud computing
	Work-in-progress, editorials, forums
	Not written in English or German

Table 3-3 Applied exclusion and inclusion criteria.

Following the recommendation of Webster, Watson (2002), a backward and forward analysis on the set of relevant articles was made using Google Scholar. In addition, a backward analysis was made on articles that were earlier marked for backward analysis only. This backward search resulted in 1941 articles and the forward search yielded 2536 articles. Again, a relevancy validation was made, which led to 30 additional relevant articles. Hence, 99 relevant articles were included in the final set.

To identify (semi) automated monitoring and auditing methods, each relevant publication was read and analyzed. In total, 23 (semi) automated monitoring and 18 (semi) automated auditing methods were extracted and are presented in section 5 and 6. Appendix B and C provide an overview of identified methods as well as corresponding sources. Furthermore, monitoring and auditing methods were separately clustered. Methods were clustered regarding their objectives and application contexts. Additionally, some clusters were adapted from Lins (2014) and Thiebes (2014). This clustering resulted in five auditing clusters (see table 3-4) and six monitoring clusters (see table 3-5).

Computer-Assisted Auditing Technologies and Tools (CAATTs)
Various CAATTs exist to support continuous auditing by enabling auditors to extract, sample and analyze auditee´s data as well as to perform technical assessments.
Evidence Gathering Mechanisms
Mechanisms to gather and store electronic evidence and information, for example embedded auditing components, independent digital agents, data marts and databases.
Auditing System Architectures
Architectural concepts and systems to support and perform continuous auditing.

Data Integrity Validation
Methods to audit and validate the integrity of customer data stored in a cloud.
Automated Analysis of Processes and System Models
Process mining techniques and semi automated model evaluation algorithms to support auditing operations.

Table 3-4 Clusters of identified auditing methods.

Cloud Monitoring Tools and Architectures
Basic architectural concepts as well as components to continuously monitor cloud service systems and gather necessary data.
Logging and Inspection
Logging frameworks and methods to create and analyze logs, which contain e.g. information about system operation.
Monitoring of virtualized Environments
Methods to monitor virtual machines, virtual environments, and to detect attacks on virtualized applications.
Intrusion, Anomaly and Behavior of Malware Detection
Methods for monitoring cloud infrastructure and networks to detect intrusions, anomalies, and behavior of malware.
Service Level Agreements Monitoring
Methods for (dynamic) monitoring of service level agreements adherence.
Compliance Monitoring
Methods to ensure contractual and regulatory compliance, for example, data protection and location compliance.
Network Monitoring
Methods to gather information about network operations and to ensure network reliability.

Table 3-5 Clusters of identified monitoring methods.

3.4 Assessing practical applicability of (semi) automated Methods

After identifying methods, their practical applicability in context of dynamic certification needed to be assessed. Therefore, semi-structured one-to-one interviews with practitioners from *Certify* were conducted. Interviews allow gathering of rich data from people in

different roles.[58] Furthermore semi-structured interviews involve the use of pre-formulated questions, but allow improvisation for emerging topics during the conversation as well. Nonetheless, there is some consistency across interviews, because of a similar set of initial questions.

In total, three semi-structured interviews were conducted with practitioners [i03, i04], lasting about 60 minutes in one case and 90 minutes in the other two cases. Since [i04] did not participate in the preceding workshops, an email was send in advance, including an introduction to dynamic certification and an interview guideline. Interview guidelines were prepared individually beforehand and are summarized in the appendix D. Interviews started with general questions concerning the execution of certification processes, followed by descriptions of selected methods, and questions regarding their general feasibility and applicability. All interviews were approved to be recorded and transcribed afterwards. Practical applicability and feasibility assessments were analyzed and are presented in section 5 and 6.

3.5 Deriving Design Recommendations and Guidelines for Dynamic Certification

The concept of dynamic certification was discussed during the interviews to gather first insights into how to design dynamic certifications. Based upon insights gained by analyzing relevant publications in the research area of CM and CA, participating in workshops, accompanying a CSC audit, and discussing the concept of dynamic certification during the interviews, design recommendations and guidelines for dynamic certifications were derived. These recommendations and guidelines are presented and discussed in section 7.1. Further on, an initial conceptual model of dynamic certification was developed (see section 7.2), comprising necessary processes and components to assure ongoing certification adherence. Finally, this conceptual model was reviewed by [i01] to increase model validity. This review led to minor model adjustments and extensions.

3.6 Mapping of Methods and Certification Criteria

Finally, the author mapped the identified CSC criteria and methods. This mapping was performed in accordance to the constructed model of dynamic certification. Mapping was based on a ‘best-fit effort’ considering the gathered information of the method, method

[58] Cf. this and the following two sentences Myers (2013), p. 119,122.

applicability assessments, and the requirements of the criteria. Remaining CSC criteria were individually assessed whether they have potential for automation, although no suitable method could be identified. Appendix E presents the criteria and method mapping.

4. Cloud Service Certification Criteria Assessment

4.1 Criteria Delineation

Assessments during the workshop revealed that 78 of 326 certification criteria should be continuously monitored and audited. These criteria are listed in appendix E. Due to size limitations and confidentially issues, in the following only a few exemplary criterion assessments are presented.

One criterion states that source code reviews should be performed regularly to identify possible vulnerabilities and security issues when developing software. Since this criterion implies actions, which have to be performed on a regular basis, the ‘Regularity’ checklist attribute was assigned, and the criterion was marked as a candidate for continuous monitoring and auditing. Another criterion requires that a service desk has to have appropriate capabilities to cope with the current amount of cloud customers. Hence, the service desk would have to be enlarged in case of a major growth of cloud customers (assigned attribute ‘External Change’). However, a CSP might neglect this enlargement to realize cost savings (assigned attribute ‘Benefits due to discontinuity’). Furthermore, the catalog requires CSPs to implement secure and reliable multi-tenancy capabilities. Since multi-tenancy is a critical cloud characteristic, and multi-tenancy security vulnerabilities might have major effects on CSs, both attributes ‘Critical Cloud Characteristic’ and ‘Critical Security Criterion’ were assigned. As a last example, one criterion demands that cloud customers are to be informed about major security incidents. This criterion has to be continuously audited to assure ongoing notification. Hence, the attribute ‘Transparency’ was assigned. Figure 4-1 shows the distribution of assigned checklist attributes and points out that attribute ‘Regularity’ and ‘Critical Security Criterion’ were assigned topmost. In addition, appendix E lists criteria and assigned attributes.

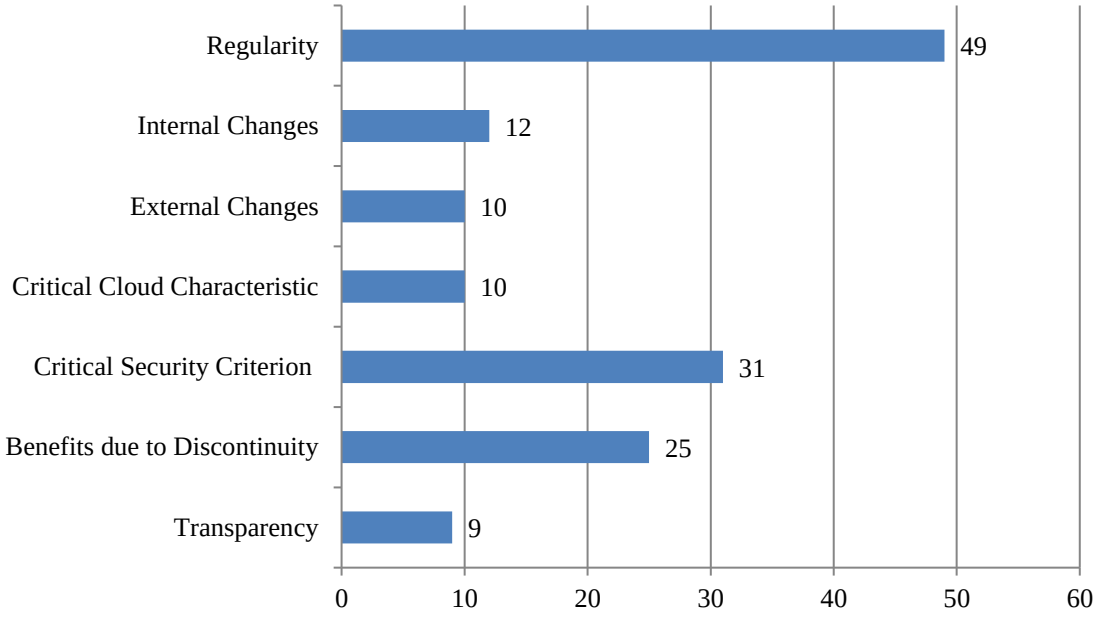

Figure 4-1 Distribution of assigned checklist attributes.

To briefly delineate criteria that were marked as candidates for continuous monitoring and auditing, they were further grouped into different categories based upon their requirement contexts. First, criteria of category *Cloud Architecture* ensures ongoing network security, performing backups and assure secure multi-tenancy capabilities. Second, criteria in category *Security Architecture and Management* necessitate performing vulnerability analysis and assuring encrypted storage of data, data confidentiality and integrity. Moreover, *Monitoring* compromises criteria, which require ongoing monitoring of cloud components, networks and availability of services. *Incident Response Management* contains criteria that require providers to receive and process incident messages in a timely manner. Further criteria belonging to the category *IT Service Continuity Management* require providers to test, extend and update service and business continuity plans regularly. Criteria assigned to the category *Internal Audit Management* necessitate providers to audit potential sub-providers, perform technical audits as well as to implement internal audit findings. Concerning *Development Processes*, documenting code, performing code reviews and assuring secure development processes should be continuously monitored and audited. Ongoing *Compliance Management* assures compliant data location, improvements of existing service directives, and service adjustments due to changes of legal or regulatory requirements. *Change management* implies performing (security) tests before integrating new hardware components and software as well as performing patch management processes. Furthermore, *Risk Management* requires providers to perform

ongoing risks analyses, reviews and updates of risk management plans. The criteria contained in category *Administration* ensure performing regular administration tasks, for example, deletion of inactive user accounts. *Service Level Management* implies monitoring of service level agreements (SLAs) adherence and reporting this adherence to customers. Finally, *Employee Management* contains criteria that recommend performing regular trainings of employees. Figure 4-2 lists these categories and the number of criteria contained.

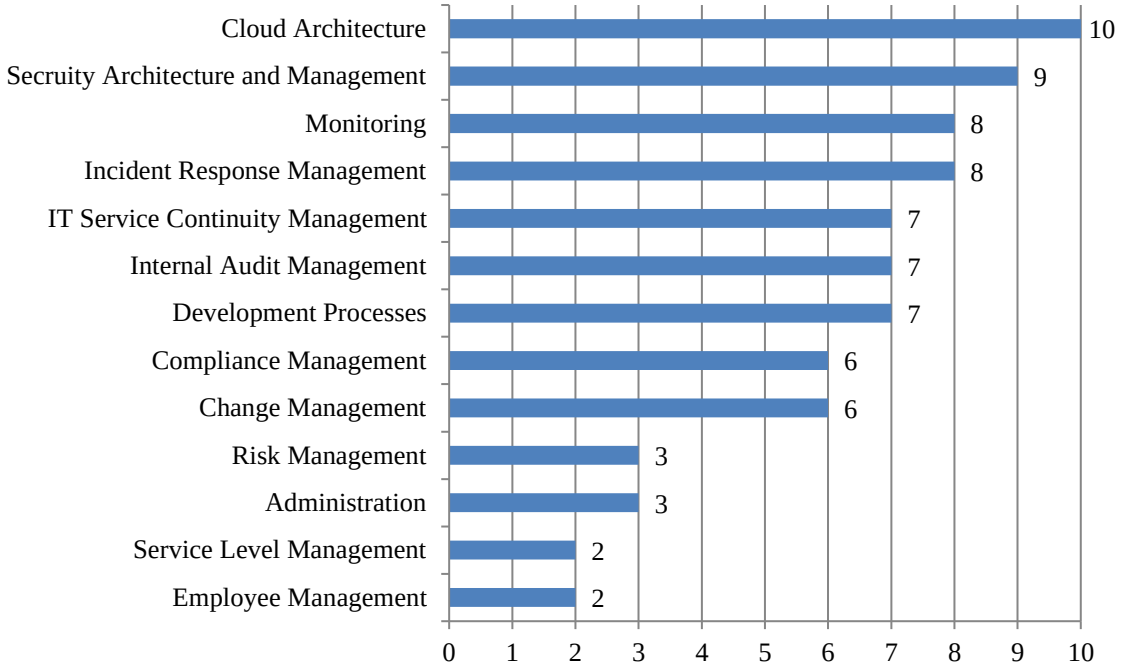

Figure 4-2 Categories of criteria that were marked for continuous monitoring and auditing.

4.2 Monitoring and Auditing Frequencies

After assessing criteria for continuous monitoring and auditing, one has to specify the frequency of operations. The frequency of CM operations is determined by the certification criterion requirements, internal CSP processes and operations (i.e., frequency of performing internal audits), as well as by the frequency of CA operations (upper frequency boundary). In general, CA has a lower frequency compared to CM to ensure economic feasibility for auditors. For example, a criterion requires CSPs to continuously monitor, document and report capacity utilization, thus the CM process has to be performed in real-time. In contrast, an auditor verifies monthly that this CM process is actually performed according to the certification requirements.

During the workshops, the CA frequency of each certification criterion was individually discussed and determined. Figure 4-3 presents the proposed frequencies and shows that their frequencies are greatly varying across different certification criteria. In addition, figure 4-3 shows that for some criteria a period of time (i.e., monthly to quarterly) was determined as frequency.

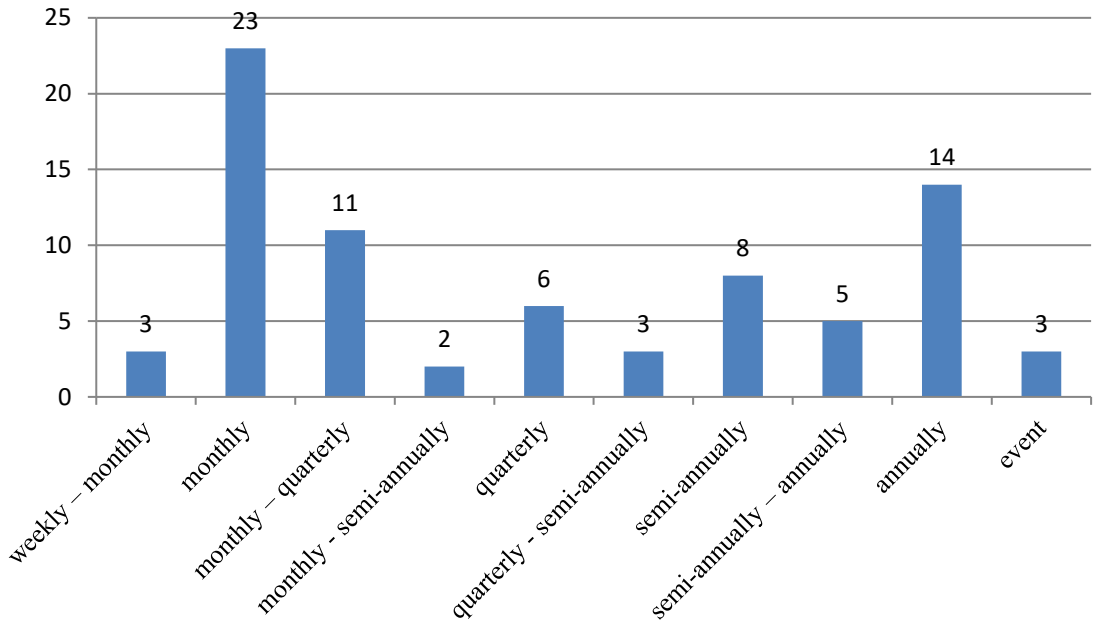

Figure 4-3 Distribution of auditing frequencies.

The required auditing frequency is influenced by several factors. First, it depends on the CS type [i03, i04]. A highly dynamic CS requires a higher frequency compared to a static one [i03]. For instance, a CS with a fairly consistent number of customers and a stable function portfolio is less affected by external and internal changes, thus a lower frequency of CA is required. Second, auditee´s operations and processes influence the frequency [i01, i03]. A CSP might review their firewall guidelines on a semi-annually basis for example. Hence an auditor has to align his CA frequency to their reviewing processes. Finally, one has to evaluate the economic feasibility of CA. This was emphasized by a practitioner as well: "*You have to keep in mind, what a provider is able to achieve on a monthly or quarterly basis. A continuous audit must always be economically achievable for him*" [i03]. In conclusion, the frequency of CA has to be individually adjusted based upon the auditee´s context.

In addition, emerging external or internal events might trigger further audits as well. Such external events comprise, for instance, announced software and hardware vulnerabilities

(e.g., Heartbleed vulnerability[59]), and might require auditors to verify certification requirements, which were not specified as CA candidates initially. Internal events might comprise major security incidents, major architectural changes or adjusted SLAs, among others.

5. Continuous Auditing Methods

The preceding criteria assessment emphasizes the need for dynamic certification, since a variety of criteria should be continuously monitored and audited. To be efficient and cost effective, CM and CA require a high degree of automation.[60] In this section, identified (semi) automated CA methods are presented and discussed according to their applicability in cloud computing contexts. In addition, assessments of practitioners about whether or not these methods can be used by external auditors to continuously audit CSC criteria are presented.

5.1 Computer-Assisted Auditing Technologies and Tools

Since the 1980s, researchers and major accounting organizations have developed various computer-assisted auditing technologies and tools (CAATTs) which might support CA.[61] CAATTs can be used by an auditor as part of their audit procedures to automatically process necessary data contained in auditee´s information systems to improve the efficiency and efficacy of these audit procedures.[62] CAATTs comprise generalized auditing software, electronic working papers, tools for fraud detection, network security testing, audit reporting, as well as databases of audit history among others.[63] Pedrosa, Costa (2014) propose a classification for CAATTs which is shown in table 5-1.

---

[59] United States Computer Emergency Readiness Team (2014).

[60] Cf. Kunz, Niehues, Waldmann (2013), p. 522, Bezzi, Kaluvuri, Sabetta (2011), p. 41, Brown, Wong, Baldwin (2007), p. 21, Schneider, Lansing, Sunyaev (2013), p. 16, Chan, Vasarhelyi (2011), p. 155 and Woodroof, Searcy (2001), p. 1.

[61] Cf. Chou, Du, Lai (2007), p. 2275, and Ahmi, Kent (2012), p. 88-89.

[62] Cf. Lungu, Vătuiu (2007), p. 217, and Singleton, Flesher (2003), p. 49, 52.

[63] Cf. Mahzan, Lymer (2014), p. 328.

Features	Description	
	Aim	Software
Data Analysis and Extraction	Data analysis, import and join data from distinct file formats, extracting registers	IDEA, ACL
Ratio analysis	Financial ratio and trend analysis	IDEA, DRAI 3, ACD Auditor
Audit sampling	Obtain a representative sample of the population	Attribute Sampling, PPS Sampling, IDEA, ACL
Digital analysis	Includes new data mining techniques to classification and association on emails, structured and unstructured texts	Benford's Law, Text Mining, Data Mining, Log Analytics
Data mining: regression/ANOVA	Define linear regression models to understand how variables are related	SAS, SPSS
Working Papers on auditing	Plan, document and share (in a collaborative perspective) all the audit process	Working Papers, ACD Auditor, DRAI 3, SIPTA
Big Data Analytics	Audit Big Data	IDEA, ACL, Hadoop
Cloud Analytics	Use online tools to audit the work in the cloud	Audit Applications
Security and Privacy Tools	Generate bring your own device and data privacy alerts	-

Table 5-1 Classification of CAATTs adapted by Pedrosa, Costa (2014).[64]

According to literature, generalized audit software (GAS) is one of the most commonly used types of CAATTs.[65] GAS provides user-friendly, packaged and automated auditing processes (e.g., data extraction, sampling and analysis tools), and is offered by software

[64] Cf. Pedrosa, Costa (2014), p. 140-141.

[65] Cf. Braun, Davis (2003), p. 727 and Ahmi, Kent (2012), p. 88-89.

vendors.[66] The ‘Audit Command Language’ and ‘Interactive Data Extraction and Analysis’ software are widely used in the context of GAS.[67] ‘Audit Command Languages’ allow auditors to connect their own laptops to the auditee´s system to download data for further analysis, provide data integrity and fraud detection tests as well as detailed journals of the accomplished audit.[68] Similarly, ‘Interactive Data Extraction and Analysis’ software provides functions to accomplish audit objectives concerning financial situations, offering verification and calculation instruments, crossed data verification, fraud investigation and testing of security norms among others.[69] In addition, CM and CA functions were recently added to both software packages, and are currently under further development.[70] However, existing CAATTs are mainly used in, and developed for accounting contexts.[71] Hence, their applicability in cloud computing contexts might be questionable. Likewise, interviews revealed that in CSC auditing practice CAATTs are mainly used to support technical security analyses [i01, i03, i04]. Auditors predominantly use customized notebooks with a variety of different software, and virtualized operating systems [i01, i04]. Furthermore, an isolated server room can be utilized to access higher computing power and to perform different types of security analyses [i04].

Regularly performing penetration tests is recommended to validate adequate security mechanisms, and to identify system vulnerabilities [i01, i03, i04]. To support efficient penetration testing, a broad variety of corporate and open-source tools exist [i03, i04].[72] Such tools were typically developed first by hackers, and then sold or distributed by different companies [i04]. Penetration testing tools and vulnerability scanners can be used to gather information about cloud systems, and to identify vulnerabilities in implemented cloud components [i01, i03, i04]. For example, ‘Nmap Security Scanner’[73] is an auditee independent tool, which can be used to scan network ports, identify running services, and

---

[66] Cf. Chou, Du, Lai (2007), p. 2275, Braun, Davis (2003), p. 727 and Ahmi, Kent (2012), p. 88-89.

[67] Cf. Braun, Davis (2003), p. 727, Ahmi, Kent (2012), p. 90 and Lungu, Vătuiu (2007), p. 219.

[68] Cf. Lungu, Vătuiu (2007), p. 219, 221.

[69] Cf. Lungu, Vătuiu (2007), p. 221.

[70] Cf. ACL Services Ltd. (n.y.), and CaseWare IDEA Inc. (2008).

[71] Cf. Pedrosa, Costa (2014), p. 138, Chou, Du, Lai (2007), p. 2275, Ahmi, Kent (2012), p. 88-89, and Braun, Davis (2003), p. 727.

[72] See for example Dalziel (2013), and tenable network security (n.y.).

[73] Lyon (n.y.).

gather information about identified services (e.g., name and version) [i04]. This information can be used to test well-known or identified security vulnerabilities [i04]. By attempting to execute prohibited behavior or attacks on vulnerabilities, auditors can verify that such behavior is prevented, or detected and compensated. Typically, penetration and vulnerability tests are carefully conducted on production, live or staging systems after consultation with auditees [i04]. Such testing provides strong evidence for comprehensive and exhaustive protection mechanisms. To support these penetration and vulnerability tests, several computerized tools are used in practice, for instance, 'Nessus'[74] and 'Qualys'[75]. These tools provide a variety of (semi) automated functions (e.g., analyze and test SQL injection vulnerability), and can be configured based upon an auditee´s context [i03, i04]. During the interviews it was noted that performing penetration tests does not require a high amount of computing power [i04]. The scope and duration of these tests is depended on the scope of the auditee ´s systems, CS type and configured testing parameters. Performing extensive penetration tests on a continuous basis (e.g., weekly) might be limited in cloud environments, since those tests might affect multiple customers (e.g., temporary performance losses or operational disturbances), even though some customers may not insist on continuous penetration tests.

Aside from the emergence of a broad variety of computer-aided auditing tools, important technological advances enhanced the technological feasibility of CA.[76] The introduction of XML in 1996 enabled a platform independent, efficient and effective exchange of data over the internet.[77] By using the XML specification, information can be exchanged and processed without modification. Practitioners recommend using XML-coded data when exchanging data between an auditee and an auditor, because XML-coded data is standardized, well-structured, and can be processed quickly [i03, i04]. Based on XML, the Extensible Business Reporting Language (XBRL) was established as a standard way of preparing and exchanging business information.[78] Users can attach financial data to XML-tags, and extract or analyze the data with analytical applications. Moreover, the related XBRL Global Ledger (XBRL GL) specification can be used for representing both

[74] Tenable network security (n.y.).

[75] Qualys (n.y.).

[76] Cf. Gao (2010), p. 2142.

[77] Cf. this and the following sentence Boritz, No (2005), p. 13, and Gao (2010), p. 2143.

[78] Cf. this and the following sentence Gao (2010), p. 2143, and Murthy, Groomer (2004), p. 144.

financial and nonfinancial information.[79] "It is both extensible, given its roots in XML, and standards-based, enabling cross-platform information exchange around the globe."[80] Similar, a broad variety of other languages can be used to facilitate CA. For example, Koschorreck (2011) analyzed the 'Open Vulnerability and Assessment Language' and the 'Extensible Configuration Checklist Description Format' to enable (semi-) automated IT controls checks, patch and configuration management validation, and vulnerability assessments. Table 5-2 summarizes methods and concepts of this cluster.

Generalized Audit Software (GAS)
GAS provides user-friendly, packaged and automated auditing processes (e.g., data extraction, sampling and analysis tools), and is offered by different software vendors.
Penetration Testing
Using a variety of corporate and open-source tools to perform penetration tests to validate adequate security mechanisms, and to identify system vulnerabilities.
Formal Languages
By using formal languages, for example XML or XBRL GL, exchanging data between an auditee and an auditor can be improved. Likewise, a variety of other languages can be used to facilitate CA and CM, and to perform automated checks.

Table 5-2 Summary of methods and concepts in cluster CAATTs.

5.2 Evidence Gathering Mechanisms

Several automated mechanisms have been identified to enable auditors to gather electronic evidence and information. Continuous evidence extraction and transmission require a (permanent) communication connection between the auditing object (i.e., the CS) and the auditor´s systems. To enable a permanent communication, Du, Roohani (2007) propose a CA model, which effectively connects auditor's systems to the auditee's systems. They classify auditee systems into two categories, XML-ready systems and non-XML-ready system.[81] If the auditee´s system is XML-ready, the Simple Object Access Protocol (SOAP) is used to send messages or requests between the auditing system and the auditee´s system. Otherwise, the Common Object Request Broker Architecture

[79] Cf. Gao (2010), p. 2143.

[80] Gao (2010), p. 2143.

[81] Cf. this and the following two sentences Du, Roohani (2007), p. 139.

(CORBA) is used as a middleware to gather information from a variety of heterogeneous auditee's applications.

The most frequently mentioned mechanism to gather audit evidence and information is an embedded audit module (EAM). EAMs are special purpose functions, programs, or other code objects that are embedded into the auditees' information systems and supervise all of the audit-related data in real-time.[82] One of the most important advantages of EAMs is that they automatically act as triggers and inform the auditor when suspicious events appear, thus eliminating the need for high frequency assurance queries.[83] Recently, organizations have begun to combine artificial intelligence with EAMs to expand their capabilities, and to reduce the number of necessary modules.[84] On the contrary, EAMs are more vulnerable to manipulation, especially by the auditees' employees who have necessary access privileges to interfere with the EAM.[85] In addition, EAMs are proprietary software solutions, hence they might not be portable to other auditing contexts.[86] When assessing the applicability of EAMs in cloud computing contexts, an EAM might be used to monitor the availability of a CS for example.[87] However, the usage of EAMs for CA in cloud computing contexts may be limited, because the incorporation of EAMs into a cloud architecture that is distributed across different datacenters and locations requires a complicated, expensive development and customization process.[88] More importantly, practitioners assess that the use of EAMs is not practically feasible. First of all, most auditees are not willing to permit auditors' to integrate EAMs due to security and privacy concerns [i01, i03, i04]. An external EAM might cause new security vulnerabilities or might disturb operating cloud systems [i04]. Auditees might even fear data theft or corporate espionage [i04]. Moreover, integrating EAMs might violate internal compliance

---

[82] Cf. Alles et al. (2006), p. 146, Chen (2004), p. 34, Schroeder (1995), p. 73-74, Chou, Du, Lai (2007), p. 2276, Groomer, Murthy (1989), p. 54, 57, Rezaee et al. (2002), p. 152, Hunton, Rose (2010), p. 303, and Braun, Davis (2003), p. 726-727.

[83] Cf. Chou, Du, Lai (2007), p. 2276, Groomer, Murthy (1989), p. 65-67, Schroeder (1995), p. 73-74, and Alles et al. (2006), p. 145-146.

[84] Hunton, Rose (2010), p. 303 cited by Hermanson et al. (2006).

[85] Cf. Alles et al. (2006), p. 146, and Groomer, Murthy (1989), p. 67.

[86] Cf. Lin, Lin, Liang (2010), p. 419.

[87] Cf. Ardagna et al. (2012), p. 1-6.

[88] Cf. concerning the development and customization process Alles et al. (2006), p. 146, Groomer, Murthy (1989), p. 68, and Lin, Lin, Liang (2010), p. 419.

requirements or corporate security policies [i04]. Integrating EAMs into the auditee´s systems seems to be exclusively feasible, if these EAMs require minimal access privileges, and analyze non-confidential data [i03].Hence, the integration of EAMs into the auditee´s systems has to be viewed critically.

An interceptor can be applied as a wrapper that is used to wrap information systems or IT components.[89] They can monitor data flowing into and out of systems, therefore enabling CA. Interceptors can be configured to validate the accordance with implemented business logics and certification requirements.[90] Contrary to EAMs, interceptors usually operate independently of information system. Hence, they can be implemented in any phase of a software life cycle, and detailed knowledge about auditee´s information systems are not necessary to initiate an interceptor. Interceptors can be installed in the application, middleware, and operating system or on the network layer, to capture any messages flowing into or out of the information system.[91] Especially the middleware layer is the most appropriate layer for implementing the interceptor approach. Currently, CORBA and SOAP provide portable interceptors,[92] and different providers offer a variety of tools to implement interceptors, for example, 'Windows Hook'[93] for the application layer, 'Apache Axis handler'[94] for the middleware layer, 'Microsoft Spy++'[95] for the operating system layer, and the 'Microsoft network monitor'[96] for the network layer.[97] Due to their independence ability, auditors can modify an interceptor without interrupting the operations of the auditee´s information system if any modification or customization is required (e.g., changing an audit rule).[98] CSC auditors are currently using interceptor tools (e.g., 'Burp Suite'[99] or 'OWASP'[100]) to intercept data streams between cloud servers and their web

---

[89] Cf. this and the following sentence Lin, Lin, Liang (2010), p. 418, and Fang et al. (2006), p. 1.

[90] Cf. this and the following two sentences Żmuda, Psiuk, Zieliński (2010), p. 128-129.

[91] Cf. this and the following sentence Lin, Lin, Liang (2010), p. 420.

[92] Cf. Lin, Lin, Liang (2010), p. 420, and Fang et al. (2006), p. 1.

[93] Microsoft (b) (n.y.).

[94] The Apache Software Foundation (n.y.).

[95] Microsoft (a) (n.y.).

[96] Microsoft (c) (n.y.).

[97] Cf. Lin, Lin, Liang (2010), p. 420.

[98] Cf. Lin, Lin, Liang (2010), p. 420, and Fang et al. (2006), p. 3.

[99] Portswigger Web Security (n.y.).

[100] OWASP Foundation Inc. (n.y.).

browser [i04]. By intercepting client requests and server responses, auditors identify and test security vulnerabilities [i04]. From a technical perspective, this interceptor concept can be further extended such that each customer request and the corresponding service response can be intercepted in general [i04]. For instance, a physical interceptor component can be placed inside the auditee´s network or data can be routed to corresponding interceptors [i04]. However, analyzing the entire data traffic is practically not feasible due to performance losses, privacy and security concerns, and legal requirements (e.g., fear of monitoring employees) [i04]. Hence, when implementing interceptors in cloud contexts, one has to filter and adjust the amount of data that is actually analyzed.

Furthermore, CA models that use multiple digital agents to support auditing processes are suggested.[101] Digital agents (DAs) (also referred to software, autonomous, or intelligent agents) are software objects that achieve individual goals by autonomously performing actions and reacting to events in a dynamic environment. Furthermore, they are characterized by having different degrees of artificial intelligence and mobility (the ability to travel from one platform to another).[102] Moreover, they can be added, removed, reconfigured and updated during runtime without altering and influencing an auditee´s or auditors system.[103] DAs are supposed to automatically perform activities that are traditionally undertaken by human auditors, for example, collecting and evaluating information and audit evidence, validating certification requirements as well as asset examination when using RFID technologies.[104] Typically, audit tasks are performed by a team of DAs, which are hierarchically structured.[105] For instance, each audit agent team consists of one captain agent, M mediator agents and N operator agents (with $0<M<N$). The captain and mediator agent are mainly responsible for coordination and aggregation, operator agents are dispatched to different information system to collect the necessary audit evidence. Through their artificial intelligence, mobility and individual, autonomous acting, they seem to be

---

[101] Cf. this and the following sentence Du, Li, Wei (2005), p. 372, Fuggetta, Picco, Vigna (1998), p. 334, Chou, Du, Lai (2007), p. 2276, Shaikh (2005), p. 410-414, Woodroof, Searcy (2001), p. 4, and Doelitzscher et al. (2012a), p. 6.

[102] Cf. Chou, Du, Lai (2007), p. 2276, and Shaikh (2005), p. 410-414.

[103] Cf. Doelitzscher et al. (2012a), p. 6-7.

[104] Cf. Chou, Du, Lai (2007), p. 2276, Shaikh (2005), p. 410-414, and Woodroof, Searcy (2001), p. 4.

[105] Cf. this and the following two sentences Ye, Yang, Gan (2012), p. 224, and Doelitzscher et al. (2012b), p. 7.

very suitable for CA of CSs, especially when comparing DAs to EAMs. However, high efforts and expenses for DA development and implementation as well as possible negative impacts on system performance have to be considered.[106] Similar to EAMs, the usage of DAs has been evaluated critically by practitioners: "*I believe that the customer acceptance to permit digital agents to perform actions is very low, because with these agents you implement untrustworthy software into your cloud systems*" [i03]. Especially agent deployment interfaces might be a highly valuable target for attackers to compromise the auditee´s and auditor´s systems [i03]. Thus, potential security vulnerabilities when using DAs bear high risks for both auditees and auditors.

Auditing mechanisms might have a negative impact on auditee´s system performance.[107] To counteract this issue, audit-related processing can be performed outside the source system by applying the concept of 'ghosting'.[108] Ghosting entails operating a 'copy' of an entire system on separate hardware, including data and system settings. Many companies utilize copies of the production environment or staging systems for development processes and change management.[109] Auditors prefer testing on these staging systems, if staging systems are technically similar to production or live systems [i01, i04]. However, in most cases staging systems are not connected to the internet due to potential open security issues or vulnerabilities [i03]. Thus, auditors are not able to access these systems externally to perform auditing operations.

Moreover, audit relevant data can be transferred at predetermined intervals to supplementary databases, for instance, in audit data marts (ADM).[110] ADMs are small, mostly auditee-independent data repositories in which relevant data from all application systems are automatically integrated.[111] By using extract, transform and load tools, audit-relevant data can be extracted from the requisite systems, and transformed to facilitate audit reporting

---

[106] Cf. Chou, Du, Lai (2007), p. 2284-2285.

[107] Cf. Singh et al. (2013), p. 302.

[108] Cf. this and the following sentence Kuhn Jr., Sutton (2010), p. 95.

[109] Cf. Braun, Davis (2003), p. 727.

[110] Cf. Singh et al. (2013), p. 302, Ye, Yang, Gan (2012), p. 221, and Rezaee et al. (2002), p. 153.

[111] Cf. Chou, Du, Lai (2007), p. 2276, Rezaee et al. (2002), p. 152, 155, and David, Steinbart (1999), p. 30.

and analytics.[112] Thus, ADMs enable a real-time data access on which continuous (semi) automated analysis and auditing can be performed. ADMs may be used for CA of CSs, if appropriate data formats are available and a secure access to the collected data is guaranteed.

Besides gathering information from the auditee´s system, auditors need to identify and evaluate external changes, for example, emergence of security threats or vulnerabilities[113]. External information can be gathered and evaluated in decision support system (see section 5.3), to trigger re-auditing events or alerts. Open vulnerability databases, like the Common Vulnerability and Exposures[114] and Common Configuration Enumeration[115] database can be assessed, to expose unknown vulnerabilities and system weakness configurations that may cause system crashes and malfunctions.[116] These databases offer information about vulnerabilities in open data formats, like XML. Additionally a vulnerability scoring system[117] has been developed, which proposes a vulnerability rating method from zero to ten, indicating the risk height, and can be accessed via internet. Using automated scanners and questionnaires, information about the auditee´s system can be gathered and then compared to the information retrieved from the databases and scoring systems, thus, enabling (semi) automated vulnerability analysis.[118] Table 5-3 summarizes evidence gathering mechanisms, and databases to store or gather audit evidence.

Embedded Audit Module (EAM)
EAMs are special purpose code objects (e.g., programs) that are embedded into the auditees' information systems and supervise all of the audit-related data in real-time.
Interceptor
Interceptors can be applied as a wrapper that is used to wrap information systems or IT components. They can monitor data flowing into and out of systems.
Digital Agent (DA)

[112] Cf. Baksa, Turoff (2011), p. 240.

[113] See for instance Shahriar, Zulkernine (2011), for a classification of vulnerabilities.

[114] The MITRE Corporation (n.y.).

[115] National Institute of Standards and Technology (n.y.).

[116] Cf. this and the following two sentences Kuo et al. (2011), p. 643-644.

[117] Forum of Incident Response and Security Teams (n.y.).

[118] Cf. Kuo et al. (2011), p. 645-646.

DAs are intelligent and mobile software objects that achieve individual goals by autonomously performing actions that are traditionally undertaken by human auditors.
Audit Data Mart (ADM)
ADMs are small, mostly auditee-independent data repositories in which relevant data from all application systems are automatically integrated and analyzed.
Vulnerability Databases
Assessing external vulnerability databases (e.g., Common Vulnerability and Exposures database) to exposure unknown vulnerabilities and system weakness configurations which might re-trigger audits.

Table 5-3 Overview of evidence gathering mechanisms and corresponding databases.

5.3 Auditing System Architectures

Aside from individual components and mechanisms to gather audit evidence, researchers have developed several comprehensive CA systems and system architectures. A monitoring and control layer (MCL) can be implemented as an independent computer system, which is usually owned and operated by the auditor.[119] It forms an overlay on top of a set of existing systems and utilizes a middleware layer to provide integration between loosely coupled applications such as the auditee´s service applications and legacy systems.[120] The concept of a MCL was first proposed by Vasarhelyi, Halper (1991).[121] In general, the MCL architecture comprises several layers: (1) data capture layer, (2) data filtering layer, (3) relational storage, (4) measurement standards layer, (5) inference engine, (6) analytic layer, (7) alarms and alerting layer, and (8) reporting platform.[122] A MCL can query data from integrated applications or receive periodic data from them.[123] Thus, a read-only access to the auditee´s system is exclusively required.[124] Extracted data is compared to a predefined rule-set of audit procedures inside the analytical layer.[125] Any violations, as defined by the rule-set, are stored and might automatically trigger an alert to the auditor.

[119] Cf. Alles et al. (2006), p. 145-146, and Kuhn Jr., Sutton (2010), p. 95.

[120] Cf. Vasarhelyi et al. (2004), p. 10, and Kuhn Jr., Sutton (2010), p. 95.

[121] Cf. Perols, Murthy (2012), p. 37.

[122] Cf. Vasarhelyi et al. (2004), p. 10, and Kuhn Jr., Sutton (2010), p. 95.

[123] Cf. Kuhn Jr., Sutton (2010), p. 95, Alles et al. (2006), p. 146, and Perols, Murthy (2012), p. 37-38.

[124] Cf. Alles et al. (2006), p. 145-146.

[125] Cf. this and the following sentence Vasarhelyi et al. (2004), p. 12, Kuhn Jr., Sutton (2010), p. 95, and Perols, Murthy (2012), p. 37-38.

Alles et al. (2006) provide an insight into the implementation of a MCL at Siemens that focused on detecting control exceptions in business processes using control information from SAP applications.[126] Similar, Singh et al. (2013) reviews CA/CM systems that are based on the MCL architecture. In contrast to embedded components, data retrieved by a MCL can be presumed to be safe to manipulation by the auditee´s employees because of the independency of the MCL.[127] However, Alles et al. (2006) identify the need for management of audit alarms and the prevention of possible alarm floods.[128] The usage of MCL in cloud computing contexts might be limited due to distributed cloud infrastructures.

Beside MCLs, agent-based CA architectures are common in literature. Under this architecture, a DA is initiated to represent a certain audit procedure and dispatched to different auditee´s systems.[129] A flexible (e.g., platform independent) and adaptable (e.g., agents can be deployed as required) agent-based architecture facilitates gathering audit evidence in distributed and heterogeneous auditee´s systems.[130] In general, it consists of several components: organizing and planning modules, scheduling modules, agent repositories and dispatcher, audit evidence and knowledge databases. These components are implemented into the auditor´s systems and only agents are dispatched to the auditee´s sites. Typically, audit organizer modules provide auditors with an interface to invoke functions, such as planning, analyzing and reporting. Moreover, audit planning modules are in charge of generating an audit plan, which may include objectives, metrics, and audit rules for a particular DA.[131] A scheduling module determines which agent needs to be instantiated as well as a dispatching destination.[132] Furthermore, an agent repository possesses the source code of each kind of DA. The agent dispatcher deploys DAs of various functions to auditee´s sites according to the audit and scheduling plan.[133] Hierarchically structured teams consisting of captain, mediator and operator agents will then perform the

---

[126] Cf. Alles et al. (2006), p. 138-139, and Perols, Murthy (2012), p. 37-38.

[127] Cf. Alles et al. (2006), p. 145-146.

[128] Cf. Alles et al. (2006), p. 157, and Perols, Murthy (2012), p. 37-38.

[129] Cf. Chou, Du, Lai (2007), p. 2276, and Ye, Yang, Gan (2012), p. 221.

[130] Cf. this and the following three sentences Wu et al. (2008), p. 355, 357.

[131] Cf. Wu et al. (2008), p. 357, and Ye, Yang, Gan (2012), p. 222-223.

[132] Cf. this and the following sentence Ye, Yang, Gan (2012), p. 222-223.

[133] Cf. Wu et al. (2008), p. 357, and Ye, Yang, Gan (2012), p. 222-223.

planned audit operations, for example, interacting with the auditee´s system and retrieving necessary audit evidence, testing effectiveness of business processes, mining data to analyze and identify fraud behavior.[134] To secure the movement of DAs, a concept of using shared keys between agents and data sources to authenticate operations are proposed.[135] Finally, information and audit evidence gathered by DAs are stored in knowledge and audit evidence databases to support the audit report.[136] In contrast to the usage of MCL in cloud contexts, agent-based CA architectures enable a flexible deployment and transmission of DAs across different cloud infrastructures and locations. In addition, DAs are able to perform a broad variety of auditing operations. Thus, agent-based architectures seem to be suitable in the context of CSC. The Java Agent Development Framework (JADE) might be used to implement DAs in cloud contexts.[137] It is one of the widely used frameworks for developing multi-agent systems that enables Java agents to be easily deployed regardless of auditee´s operating system platforms.

CA can be realized as a set of web services that reside within the auditor´s computing environment.[138] Each auditing function is therefore represented as a web service which can be invoked to continuously audit an auditee´s system.[139] Such a web service model can be realized by implementing wrappers for each business process within the auditee´s system. For instance, by integrating web services description language wrappers at the auditee´s system, SOAP and HTTP communication is facilitated between the auditor's web service and auditee´s systems. Alternatively, an XBRL GL data hub can be integrated into the auditee´s system, which converts system data into the XBRL GL format and transmits formatted data to the auditor´s web services. The use of web services for auditing enables new businesses model for auditing firms.[140] In such business models, cloud customers can invoke auditor´s web services to continuously retrieve assurance reports. Each time a web service is invoked, an auditor can receive a service fee. The literature

---

[134] Cf. this and the following two sentences Ye, Yang, Gan (2012), p. 224, and Doelitzscher et al. (2012b), p. 7.

[135] Cf. Zhang, Wan (2011), p. 1-4.

[136] Cf. Ye, Yang, Gan (2012), p. 222-223.

[137] Cf. this and the following sentence Wu et al. (2008), p. 358-359.

[138] Cf. Yeh, Chang, Shen (2008), p. 1013, Murthy, Groomer (2004), p. 149, and Gao (2010), p. 2143.

[139] Cf. this and the following three sentences Murthy, Groomer (2004), p. 147, 149-150.

[140] Cf. this and the following two sentences Murthy, Groomer (2004), p. 147, 149-150.

review revealed that the concept of auditing web services has already been applied in context of cloud computing. The Cloud Research Lab at Furtwangen University in Germany is developing an incident detection web service for cloud computing.[141] A cloud customer is able to define security SLAs that regulate which cloud components should be monitored and how.[142] To validate defined security SLAs, an agent framework is used, whereas DAs are deployed at all core components of a cloud infrastructure (e.g., running VMs of cloud users, data storage, network transition points).[143] These DAs are able to detect security incidents, for instance, account misuses, distributed denial of service attacks, VM breakouts, and cloud resource misusage, among others.[144] Cloud customers can track the status of monitored components through a web portal.[145]

Finally, it is recommended to implement Decision Support Systems (DSS) to support CA processes.[146] In general, DSS are intended to improve decision quality, expedite decision-making processes and decrease the amount of effort required for effective performance. To support CA, these DSSs may incorporate different mining techniques, which are performed on a regularly basis. Data mining involves many different techniques (e.g., neural networks, distributions of numbers) for discovering patterns in large sets of data and to detect irregularities in these patterns. Text mining involves discerning patterns from text to detect deception and fraud, and can be applied for example to email, discussion groups, media, and in general to the internet. These mining techniques enable an efficient employment of scarce auditing resources. Nevertheless, for auditors, mining techniques are difficult to develop. Especially when employing mining techniques, a huge volume of data, exceptions and reports may be generated, thus threatening audit efficiency. In these cases, DSSs can be used to aggregate information from many different sources, for instance from EAMs, minded external and internal data, and efficiently and automatically decide to take actions or to alert the auditor, based on the aggregated and analyzed

---

[141] Cf. Doelitzscher et al. (2013), p. 150.

[142] Cf. Doelitzscher et al. (2012a), p. 379.

[143] Cf. Doelitzscher et al. (2012a), p. 379, and Doelitzscher et al. (2012b), p. 8.

[144] Cf. Doelitzscher et al. (2012b), p. 13-14.

[145] Cf. Doelitzscher et al. (2013), p. 158.

[146] Cf. this and the following sentences Hunton, Rose (2010), p. 297, 301, 303.

evidence.[147] Future DSSs may even evolve to intelligent and adaptive audit process systems, which collect large volumes of data from the audit environment, automatically adjust audit plans based upon data analysis and environmental triggers, and automatically generate new audit tests for unexpected events.[148] DSSs can be used to support CA in contexts of CS and reduce the workload of auditors, by aggregating information and promote decisions automatically. Especially concerning decisions, about whether or not to re-audit several criteria, based upon auditee´s information, DSSs can be used to reduce auditor´s manual judgment. Thus, in combination with interfaces or websites, in which an auditee reports information and changes, a DSS can be used to automatically react on changes, perform actions, or promote alerts. Additionally, mining techniques can be used to inform auditors about changes in the auditee´s system, which may be relevant for the certification process. Also mining of external sources, for example reports about major security risks or incidents, new viruses or threats, can provide the auditors with timely information. Currently, DSSs are not used during CSC processes [i03]. Nonetheless, auditors endorse the concept of using DSSs to support and to automate their auditing practices [i03]. Implementation efforts depend on the scope of such DSSs, hence simple systems might be easily developed [i03]. Table 5-4 outlines identified continuous auditing system architectures.

Monitoring and Control Layer (MCL)
The MCL forms an overlay on top of a set of existing systems and utilizes a middleware layer to provide integration between loosely coupled applications, such as the auditee's service applications and legacy systems.
Agent-based continuous Auditing Architectures
Digital Agents are initiated to represent a certain audit procedure and dispatched to different auditee´s systems. A flexible (e.g., platform independent) and adaptable (e.g., agents can be deployed as required) agent-based architecture facilitates gathering audit evidence in distributed and heterogeneous auditee systems.
Auditing Web Services

[147] See Kleinmuntz (1990) for a discussion, about DSSs are better aggregators for information cues than human decision makers.

[148] Cf. Hunton, Rose (2010), p. 304-305.

Auditing functions can be represented as web services that reside within the auditor´s computing environment, and can be invoked to continuously audit an auditee´s system.
Decision Support System (DSS)
DSSs can expedite decision-making processes and decrease efforts by incorporating mining techniques (e.g., data and text mining) and monitoring technologies, and by aggregating and analyzing data from different information sources.

Table 5-4 Overview of identified continuous auditing system architectures.

5.4 Data Integrity Validation

As CS customers do not longer possess their data locally, assuring that their data is being correctly stored and maintained in cloud environments is of critical importance [i03]. Data integrity may be threatened by malicious insiders, data loss due to management errors, technical or byzantine failures, and by external attackers.[149] Ensuring data integrity in cloud environments is a challenging task due to multitenant architectures and distributed systems.[150] In addition, validating data integrity of outsourced customer data and customer´s meta data is more challenging compared to traditional integrity checks of in-house stored data [i03]. To ensure data integrity proactively, auditors are analyzing auditee´s processes and technical arrangements [i03]. For instance, appropriate data backup processes, and security mechanisms to prevent malicious data modification have to be in place to ensure data integrity [i03]. However, to create trustworthy CSs, auditors might continuously validate that data integrity is maintained as well. A wide range of research currently addresses the question on how to assure data integrity in cloud computing contexts. Recently, Liu et al. (2013b) and Yang, Jia (2012) analyze and provide a survey on main aspects of this research problem, summarize methodologies as well as present achievements of selected integrity validation approaches.

A broad variety of methods enables a third party to audit and validate the integrity of data stored in a cloud.[151] Especially hashing techniques have been identified as adequate

[149] Cf. Nithiavathy (2013), p. 125-126.

[150] Cf. Subashini, Kavitha (2011), p. 5.

[151] See Liu et al. (2013a), Wang et al. (2013), Wang et al. (2011), Yang, Jia (2013), Zhu et al. (2013), Sujana, Revathi (2012), Nithiavathy (2013), Wang, Li, Li (2013b), Rajkumar, Kumar, Sivaramakrishnan (2013), Liu et al. (2014), and Shah, Swaminathan, Baker (2008).

methods for monitoring the integrity of large amounts of data.[152] These methods enable auditors to simultaneously verify the integrity of multiple users' data, which is important in multitenant cloud environments with many users operating at the same time. Moreover, simultaneous monitoring of multiple and hybrid clouds, and multiple owners is feasible.[153] Auditors are able to detect anomalous behavior of data operations as well.[154] Aside from that, some methods support dynamic data operations on a fine-grained level, thus, minor data changes are considered when validating data integrity.[155] Data security and privacy has to be ensured when validating data integrity in cloud environments, for example, by implementing cryptography,[156] authentication,[157] or authorization techniques.[158] Furthermore, using periodic sampling audits or moving computational operations onto the cloud server, auditors can reduce communication and computation cost, which leads to increased audit efficiency.[159] In addition, by computing verification tokens, auditors are able to locate data errors.[160]

Moreover, contexts in which cloud users are sharing data as a group require adjusted integrity validation checks.[161] In these contexts, initiating users and other group users are able to concurrently access and modify shared data.[162] Furthermore, new users may be added, or existing users may be revoked from the group, hence creating dynamic group settings. Commonly, auditors will validate private user signatures of used data to validate data integrity, however, in shared data contexts, auditors may be able to reveal

---

[152] Cf. this and the following sentence, for example, Yang, Jia (2013), p. 1721, Zhu et al. (2013), p. 230, and Liu et al. (2013a), p. 4

[153] Cf. Yang, Jia (2013), p. 1717, Zhu et al. (2012), p. 2231, and He et al. (2013), p. 51.

[154] Cf. Zhu et al. (2013), p. 227.

[155] Cf. Liu et al. (2013a), p. 1, and Wang et al. (2009), p. 1.

[156] Cf. Yang, Jia (2013), p. 1718, and see Ni et al. (2013) for a proposed security patch.

[157] Cf. Wang et al. (2013), p. 2-3, Sujana, Revathi (2012), p. 96, He et al. (2013), p. 53, and Nithiavathy (2013), p. 126.

[158] Cf. considering authorization techniques Liu et al. (2013a), p. 2, and Zhu et al. (2013), p. 229.

[159] Cf. for periodic sampling Zhu et al. (2013), p. 229, and Kwon et al. (2014), p. 2, and for moving computational operations onto the cloud server Yang, Jia (2013), p. 1717.

[160] Cf. Nithiavathy (2013), p. 127.

[161] Cf. Kwon et al. (2014), p. 2, Wang, Li, Li (2012), p. 1-2, Wang, Li, Li (2014), p. 295, and Wang, Li, Li (2013a), p. 2904.

[162] Cf. this and the following sentences Wang, Li, Li (2013b), p. 1946, 1947, 1948.

confidential information of the group (e.g., which user in the group is modifying data most) by this common approaches. Thus, these shared data contexts require adjusted integrity checks, which can preserve privacy of group users. One solution for this problem is using private group keys that can be used as additional file signatures, and as an indicator for data integrity. Likewise, an auditor can use index tables or signatures to ensure data integrity in shared data contexts.[163]

When stored data is archived, it remains necessary to ensure its integrity for disaster recovery, or to assure compliance with legal requirements.[164] To perform automatic integrity checks, and to ensure the recovery of corrupted files under a multi-server setting, a data integrity protection scheme is proposed. This scheme requires only a thin-cloud interface (an interface offering standard read and write functionalities), thus it can be deployed to general types of storage services and no CS implementation changes are required.[165] Given an archive file, it can be encoded into code chunks, which are distributed over and stored on a number of servers. In cases of server failures, a file can be reconstructed by reading a set of chunks smaller than the original file from other surviving servers, and by reconstructing only lost (or corrupted) data chunks, instead of recovering a complete file from one server. A client or an auditor can ask for randomly chosen parts of remotely stored data, and run a probability checking protocol to verify the data integrity.[166]

The identified methods form a comprehensive sample for enabling continuous, secure, and privacy-preserving auditing of cloud storage data integrity, with low computational overhead. Table 5-5 categorizes the presented approaches.

Auditing of Data Integrity
A variety of methods are proposed to enable third parties to audit and validate the integrity of multiple users´ data stored in a cloud.
Auditing of Shared Data Integrity

[163] Cf. Kwon et al. (2014), p. 2, Wang, Li, Li (2012), p. 1-2, Wang, Li, Li (2014), p. 295, and Wang, Li, Li (2013a), p. 2904.

[164] Cf. this and the following sentence Chen, Lee (2014), p. 407.

[165] Cf. this and the following sentences Chen, Lee (2014), p. 408, 410.

[166] Cf. Chen, Lee (2014), p. 409-412.

Contexts in which cloud users are sharing data as a group require adjusted integrity checks due to dynamic group settings. One solution for this problem is using private group keys, which can be used as additional file signatures, and as an indicator for data integrity.
Validating Backup Integrity
To perform automatic backup integrity checks, and to ensure the recovery of corrupted files, a backup file can be encoded into code chunks, which are distributed over and stored in a number of servers.

Table 5-5 Summary of data integrity validation mechanisms.

5.5 Automated Analysis of Processes and System Models

Several certification criteria require auditors to analyze business processes and system architectures to identify security vulnerabilities or compliance issues. Process mining techniques and semi automated model evaluation algorithms can be used to support these auditing operations.

Process mining describes a technique of systematically analyzing data recorded by information systems.[167] It enables auditors to gain insights into how processes are being undertaken by analyzing a vast amount of data that is routinely gathered and stored in event logs. “The scope and power of process mining is dependent on how comprehensive the event log is including data on all activities relevant to the process being analyzed.”[168] Thus, comprehensive event logs must be created.[169] Three fundamental process mining perspectives can be applied when analyzing event logs: the process perspective, the organizational perspective and the case perspective. The process perspective can be used to compare the actually logged process with a designed process model to identify control failures and weaknesses. Such process discovery and conformance checks are carried out by examining log timestamps to systematically establish a flow of activities through the process from beginning to end, and compare these to prescriptive models. Adopting the organizational perspective enables the auditor to identify how the process was undertaken, hence checking segregation of duty controls. “The case perspective focuses on a

[167] Cf. this and the following sentence Jans, Alles, Vasarhelyi (2013), p. 2,4.

[168] Jans, Alles, Vasarhelyi (2013), p. 5.

[169] Cf. this and the following sentences Jans, Alles, Vasarhelyi (2013), p. 5, 11, 12.

single process instance, tracing back its history and relationships of users that are involved in that history."[170] Such process mining techniques might be used in CA contexts, to derive process models, or to assess process executions.

It must be ensured that moving (parts of) business processes into the cloud does not violate compliance or security rules.[171] To verify business processes, workflow models can be designed and analyzed. "A workflow is a discrete and case-based business process, i.e., it has a defined start and end point, and handles a specific instance of a business process."[172] It comprises a control flow that describes activities that happen in what order, and an information flow that describes which data and resources are exchanged between these activities.[173] For further analysis, such workflow models have to be transformed into Petri nets.[174] A Petri net is an abstract, formal model of an information flow that can be used to model systems of events, and is illustrated by using graphs.[175] The transformation process from workflow models to Petri nets can be automated, when processes are described in WS-Business Process Execution Language or Business Process Model and Notation.[176] Additionally, Petri nets can be generated based on event logs, in case the start and the completion time of relevant events are logged.[177] After this transformation, workflows can be analyzed regarding information flow risks and compliance adherence.[178] In addition to the workflow Petri net, a Petri net for compliance rules has to be defined.[179] A compliance rule describes which activities might, must or must not be performed on what objects by which roles at what time. Such compliance Petri nets are automatically analyzed and compared to the workflow Petri net, and rule violations are marked. Likewise, when (partially) outsourcing workflows into the cloud, providers have

---

[170] Jans, Alles, Vasarhelyi (2013), p. 11.

[171] Cf. this and the following sentence Accorsi, Lowis, Sato (2011), p. 145, and Accorsi (2011), p. 1-5.

[172] Accorsi, Lowis, Sato (2011), p. 146.

[173] Cf. Accorsi, Lowis, Sato (2011), p. 146.

[174] Cf. Accorsi, Lowis, Sato (2011), p. 149, and see Business Process Technology Group (n.y.), and Saha (2008) for transformation tools.

[175] Cf. Peterson (1977), p. 223.

[176] Cf. Accorsi (2011), p. 1-5.

[177] Cf. Wen et al. (2009), p. 1-2.

[178] Cf. Accorsi (2011), p. 1-5.

[179] Cf. this and the following two sentences Accorsi, Lowis, Sato (2011), p. 148-149.

to ensure that data and information are securely handled.[180] Security leaks and risks of workflow models can be detected semi automatically by analyzing information flows based upon Petri nets. After the workflow transformation to Petri nets, security requirements concerning information flows between activities and objects inside the workflow have to be defined.[181] For example, information flows can be classified as secret or public, thus it has to be ensured that secret information flows do not interfere with public flows to prevent information leakage. Finally, automated statistical validation algorithms can be used on these Petri nets to check whether risks or information leakages exist.

Service-oriented architectures and web services are one of the most important enabling technologies for cloud computing.[182] Similar to the analyzing of workflows to detect failures, web service designs can be checked to identify and address design problems at early stages. Therefore, web service behaviors are divided into two types: operational behaviors and control behaviors. Operational behavior illustrates the business logic that is represented by the functioning of a web service. Control behavior acts as a controller over the operational behavior, and guides its execution progress. To model these behaviors, state charts and Petri nets are recommended. The interactions between control and operational behaviors are modeled as conversation sessions (i.e., sequences of messages exchanged between the control and operational behaviors). By analyzing conversational messages and checking service behavior specifications, it is possible to verify the service design. This approach has been tested on several web services, and has been validated to detect service design problems. Table 5-6 presents methods contained in this cluster.

The presented methods can be used to automatically analyze processes, models and system architectures. However, interviews and field observation revealed that most auditees do not provide models and architectures in suitable, machine-readable formats. Typically, auditee´s are presenting and describing their systems and processes during one-site audits in order to enable auditors to identify vulnerabilities and errors as well as to evaluate the adherence to certification requirements. Hence, the presented methods were rated as not relevant for certification processes [i01, i03, i04]. Moreover, it was noted that usually

---

[180] Cf. this and the following sentence Accorsi (2011), p. 1-5.

[181] Cf. this and the following two sentences Accorsi (2011), p. 1-5.

[182] Cf. this paragraph Sheng et al. (2014), p. 416, 417, 423, 429, 431.

cloud systems are getting certified after reaching a certain service maturity stage, thus, system designs and process models might be outdated [i04].

Process Mining
Process mining describes a technique to systematically analyze data recorded by information systems to gain insights into how processes are being undertaken.
Workflow Model Analysis
Workflow models can be analyzed regarding information flow risks, compliance adherence as well as security leaks.
Web Service Design Analysis
Web service designs can be semi automatically checked to identify and address design problems at early stages.

Table 5-6 Overview of cluster ‘Automated Analysis of Processes and System Models’.

6. Continuous Monitoring Methods

Performing CM by CSP forms a prerequisite for auditors to perform efficient CA, since monitoring capabilities of auditors may be limited due to technical, organizational and legal reasons (see section 2.3). In the following sections, identified (semi) automated CM methods are presented and discussed according to their applicability in cloud computing contexts. Moreover, supplements by practitioners were added to corresponding methods. It was agreed upon that the following monitoring clusters cover the most important cloud monitoring methods and concepts [i04].

6.1 Cloud Monitoring Tools and Architectures

From a CSP´s point of view, operating and maintaining a cloud infrastructure is more challenging compared to a classic datacenter due to inherent cloud computing characteristics.[183] A CSP has to prove that he is capable of dealing with a variety of requirements, for instance, secure isolation and adequate segregation of shared computing and storage resources, measuring availability, service and data protection, and compliance to laws and customer requirements. Therefore, a provider has to implement and operate different monitoring tools, mechanisms and architectures.

[183] Cf. this and the following sentence Doelitzscher et al. (2013), p. 149.

Different types of monitoring mechanisms and tools were identified to collect runtime information from cloud components on different architectural layers.[184] First, the cloud infrastructure and platform might be monitored in real-time by using native libraries, for instance, using the cross-platform API 'Hyperic's System Information Gatherer and Reporter'[185]. Second, filters and interceptors can gather information out of messages to and from monitored services. Third, service probes are widely used to monitor different cloud components. A service probe is a special purpose monitoring code that is manually embedded inside the target code. Lastly, http-detectors can monitor the performance of cloud applications by simulating http-requests sent from end users. Aside from abstract monitoring mechanisms, a broad range of commercial (e.g., 'Amazon CloudWatch'[186], 'AzureWatch'[187] for Windows Azure-based cloud applications) and open-source cloud monitoring software and tools exist.[188] One of the most popular open source monitoring solutions is 'Nagios'.[189] Nagios monitors hosts and services, and alerts users when issues occur or are resolved. Typically, Nagios is operated on a central server that remotely executes monitoring operations. Likewise, 'OpenNebula'[190] is an open source toolkit for the management of distributed and heterogeneous cloud infrastructures, and 'Nimbus'[191] provides an integrated set of monitoring tools.[192]

Besides cloud monitoring mechanisms and tools, a variety of publications were identified that describe and analyze general cloud monitoring architectures,[193] and monitoring

---

[184] Cf. this and the following sentences Shao et al. (2010), p. 316-317, and Shao, Wang (2011), p. 29.

[185] HYPERIC (n.y.).

[186] Amazon Web Services, Inc. (n.y.).

[187] Paraleap Technologies (n.y.).

[188] Cf. Aceto et al. (2013), p. 2103-2108, Fatema et al. (2014), p. 2920-2921, and Alhamazani et al. (2014), p. 11.

[189] See for Nagios Nagios Enterprises (n.y.), and cf. this and the following two sentences Katsaros, Kübert, Gallizo (2011), p. 427.

[190] OpenNebula Project (n.y.).

[191] University of Chicago (n.y.).

[192] Cf. Aceto et al. (2013), p. 2105.

[193] See for example Katsaros, Kübert, Gallizo (2011), Povedano-Molina et al. (2013), Montes et al. (2013), Kutare et al. (2010), Hasselmeyer, d'Heureuse (2010), Tovarnak, Pitner (2012), and Shao et al. (2010).

architectures for virtualized environments.[194] However, due to size limitations and a focus on CA in this thesis, only an exemplary open-source monitoring platform will be described in the following. Aguado, Calero (2014) developed a monitoring PaaS for providers and consumers to monitor cloud infrastructures. Hence, their PaaS addresses the lack of control of customers with regards to monitoring in cloud environments.[195] Through the use of this monitoring PaaS, providers are able to see a complete overview of their infrastructure, whereas cloud customers are able to see and monitor their provisioned cloud resources. Furthermore, cloud customers can configure and customize what information is gathered about their monitored resources. Additionally, this monitoring information is kept private, thus they can neither be accessed by other cloud customers nor by the CSP. The proposed monitoring architecture is based on Nagios, and extends the cloud infrastructure by means of inserting a new service attached to the communication middleware of the cloud infrastructure.[196] Monitoring services are provided via additional VMs, created per cloud customer. Thereby, cloud customers can access their own monitoring platform by accessing a web interface. Additionally, cloud customers can use this web interface to define new services and metrics to be monitored. This monitoring PaaS has been implemented and released to the community as an open source project under GPL license. The architecture has been successfully validated in an intensive test, and it has been empirically proven that the proposed monitoring architecture only imposed a negligible performance overhead and scales well under different stressing workloads. Table 6-1 summarizes this cluster.

---

[194] See for example Clayman, Galis, Mamatas (2010), Clayman et al. (2011), and Xiang et al. (2010).

[195] Cf. this and the following sentences Aguado, Calero (2014), p. 1, 2,5.

[196] Cf. this and the following sentences Aguado, Calero (2014), p. 4,6-8, 13.

Cloud Monitoring Mechanisms and Tools
Different types of monitoring mechanisms and tools were identified to collect runtime information from cloud components on different architectural layers, e.g., native libraries, interceptors, service probes as well as commercial and open-source monitoring software and tools.
Cloud Monitoring Architectures
A variety of monitoring architectures for clouds and virtualized environments are proposed, e.g., a monitoring platform-as-a-service that attaches new monitoring services to the communication middleware of cloud infrastructures.

Table 6-1 Overview of cloud monitoring tools and architectures.

6.2 Logging and Inspection

To ensure reliable and comprehensive CM, an extensive logging of occurred events and corresponding information is essential.[197] Thus, a CSP has to implement appropriate logging facilities and mechanisms. However, to be useful and credible, log data must fulfill the properties integrity and confidentiality [i01, i04].[198] Log integrity states that log data is accurate (entries have not been modified), complete (entries have not been deleted), and compact (entries have not been illegally added to the log file). Confidentiality states that log entries cannot be stored in clear-text to prevent manipulation. To ensure these properties, log data must be encrypted by means of cryptographic techniques, for example hashing techniques.[199] Nonetheless, a provider has to evaluate which log data is of critical importance, since securing and encrypting logs cause's additional efforts [i04].

A suggested and widely accepted solution to implement efficient logging structures in cloud environments is a layered logging framework to increase accountability of CSs.[200] It consists of different logging layers: system layer, data layer, and workflow layer. First, the system layer creates logs, which contain information about the operating system, (file) system events, virtual and physical memory, and network traffic. A technique addressing this layer has been proposed and implemented.[201] It intercepts every file access in virtual

[197] Cf. Accorsi, Stocker (2008), p. 4.

[198] Cf. this and the following two sentences Accorsi (2007), p. 6-7.

[199] Cf. Accorsi (2007), p. 6.

[200] Cf. this and the following two sentences Ko et al. (2011), p. 585-587.

[201] Cf. this and the following sentence Ko, Jagadpramana, Lee (2011), p. 765, 770, 771.

and physical machines to enable system administrators and end-users to audit file life cycles, access and transfer histories as well as to determine the physical location of files. Second, the data layer produces logs about the data storage system of a CS.[202] Data layer logs can be subdivided into logs recording the provenance of data, and logs documenting the consistency of stored data. Lastly, the workflow layer is concerned with how clouds can achieve high auditability. For example, logs are required for auditing the patch management process or to increase the accountability of CSs by logging processes in detail. Additionally, policies, laws, and regulations require further information to be logged. This framework may serve as a foundation for future CM of CSs regarding logging and log inspection.

Moreover, logs can be analyzed to assure protection of customer´s privacy.[203] Current techniques aim at a preventive protection of privacy, for example by using identity management systems. However, posteriori techniques to verify compliance with privacy policies can be used as well. To realize such posteriori techniques, a policy language for the expression of privacy preferences, a secure logging to ensure confidentiality and integrity of recorded data, and an automated monitoring process for checking adherence to policies have to be implemented.[204] A policy language allows providers to specify a set of rules (i.e., a policy to regulate access). To validate that user´s and object´s (e.g., computer monitors or threads) actions adhere to defined privacy policy rules, monitoring logs can be automatically analyzed.[205] The monitoring is carried out in two steps: first, the log file is transformed into a tree structure, and second the resulting tree is pruned according to defined policies. Tree pruning refers to successively removing tree nodes when they are compliant with policy rules. Thus, the remaining tree comprises the policy violations found during the monitoring.

Aside from that, abstract execution logs to monitor the execution of applications are a suitable solution to enable CA of CS applications.[206] Such an approach enables heuristics-based log inspection techniques, which can inspect log lines with limited format

---

[202] Cf. this and the following four sentences Ko et al. (2011), p. 585-587.

[203] Cf. this and the following two sentences Accorsi (2007), p. 2.

[204] Cf. this and the following sentence Accorsi (2007), p. 3.

[205] Cf. this and the following sentences Accorsi, Stocker (2008), p. 1,3,5,8.

[206] Cf. this and the following two sentences Jiang et al. (2008), p. 181-185.

requirements and can scale up to process log files, which contain thousands or millions of log lines. This approach has been tested on a large enterprise application and provided evidence that log lines with high precision and recall can be abstracted. Supposedly, such a method can be used to automatically and continuously check whether different applications are actually running on a cloud infrastructure, for example, malware protection or antivirus software. Additionally, it may be used to automatically identify prohibited application execution (e.g., restricted access).

Likewise, unstructured logs can be automatically analyzed by using data mining techniques, for example, to detect system anomalies.[207] The technique consists of two processes, the learning and the detection process. During the learning process, models that represent the normal executional behavior of the system are derived. Therefore, training log files, which represent a normal system usage, are used as input in the learning process. Afterwards, these logs are analyzed, and finite state automaton models are automatically created. In the detection process, new input logs are compared to the learned models to automatically detect anomalies. This mining technique was implemented in two distributed systems and has shown efficiency.

To ensure integrity, confidentiality and auditability of log files, several concepts can be implemented [i04]. First, a central log server with a restrictive access model that gathers log files can be deployed [i01, i04]. Second, a central logging component that comprises encryption techniques can be implemented.[208] Instead of adjusting and customizing existing log mechanisms, an appropriate log adapter can be implemented. These adapters are mechanisms that extract and transfer log entries from different logging sources (e.g., hypervisor) to a central logging component. This central logging component transforms log entries into a secure, encrypted and uniform log type. To prevent internal log manipulation from insiders, a trusted third party hardware security module can be implemented that provides secure log encryption functions [i01, i04]. Such a trusted third party manages and stores encryption and meta data (e.g., encryption keys, certificates, authentication data) to prevent provider manipulation, and to enable external auditability.[209] Likewise, decryption keys can be divided into chunks and distributed to different locations, or

---

[207] Cf. this paragraph Fu et al. (2009), p. 149-150, 156-157.

[208] Cf. this and the following three sentences Kunz, Niehues, Waldmann (2013), p. 524.

[209] Cf. Kunz, Niehues, Waldmann (2013), p. 524-525.

handed over to different employees [i04]. In large distributed systems (e.g., cloud systems), it may be impractical to assume a real-time communication between a trusted third party and a logging facility.[210] Therefore, an untrusted logging machine has to accumulate monitoring logs, which are threatened by attackers and manipulation. Thus, they have to be properly secured by ensuring external auditability at the same time. To solve that problem, a strategy that enables signers to log a large number of log entries with little computational, storage, and communication costs in a publicly verifiable way was identified.[211] First, an individual signature is computed for each accumulated log item, which cannot be forged without knowing its associated secret keys. Second, signature keys are updated and the old ones are deleted. Third, the newly generated signature for the last log item is aggregated to the existing signatures. Through the use of public keys, auditors are able to decode the logs and to verify the data. Finally, to prevent logs from internal manipulation, monitoring and log administration employees can be divided into different teams with distinct responsibilities and entitlements, for instance, the monitoring team supervises monitoring components that create logs but only log administrators have access to these logs [i04]. Table 6-2 summarizes logging methods and concept contained in this cluster.

Layered Cloud Logging Framework
A cloud logging framework to increase accountability of CSs. The framework consists of different logging layers: system layer (e.g., logs about operating system, virtual and physical memory, and network traffic), data layer (e.g., logs recording the provenance of data and documenting the consistency of stored data), and workflow layer (e.g., logging to achieve high auditability).
Privacy Protection based upon Log Analysis
A posteriori log analysis to verify the compliance with defined privacy policies.

[210] Cf. this and the following two sentences Yavuz, Ning (2009), p. 219, 222.

[211] Cf. this and the following four sentences Yavuz, Ning (2009), p. 219, 222.

Abstract Execution Log Inspection
Using abstract execution logs to monitor the execution of applications with limited log format requirements.
Unstructured Logs Analysis
Automatically analyzing unstructured logs by using data mining techniques to detect system anomalies.
Securing Logs
To ensure integrity, confidentiality and auditability of log files, several concepts can be implemented, for example, implementing a central log server with a restrictive access model, encryption techniques, and hardware security modules.

Table 6-2 Overview of logging methods and concepts contained in cluster 'Logging and Inspection'.

6.3 Monitoring of virtualized Environments

A cloud infrastructure builds on multi-tenancy and virtualized environments, thus appropriate monitoring methods have to be implemented, to assure security of virtual machines (VM) and virtualized applications [i04]. Different types of methods can be distinguished: monitoring of VMs, monitoring of interactions between applications, VMs, and virtual environments, and finally monitoring of virtualized applications.

When monitoring VMs one can differentiate between In-VM-Monitoring and Out-of-VM-Monitoring.[212] When a monitoring component resides in the same VM environment, it is called In-VM-Monitoring. On the contrary the Out-of-VM-Monitoring approach is used, when a monitor is located and isolated in separate VMs. At the high-level, the In-VM monitoring approach provides higher performance, and the Out-of-VM approach provides higher security. To overcome such lower security levels of In-VM-Monitoring approaches, an In-VM-Monitoring framework is proposed, in which a security monitor can reside inside a guest VM but still enjoys the same security benefits of Out-of-VM-Monitoring.[213] This framework introduces a separate hypervisor-protected virtual address space in the guest VM to place the security monitor. The virtual memory is mapped in such a way that it has a one-way view of the guest VM's original virtual address space. This means that the security monitor can view the address space of the operating system,

[212] Cf. this and the following three sentences Sharif et al. (2009), p. 478.

[213] Cf. this and the following sentences Sharif et al. (2009), p. 478.

but no code executing in the operating system can view the security monitor's address space, thus ensuring the security of the monitor.[214] This framework was implemented and compared to an Out-Of-VM-Monitoring technique. It shows that the overhead was significantly reduced while preserving security.

To address the lack of control for cloud customers, an architecture for dynamic management and monitoring of VMs is proposed.[215] This architecture enables cloud users to participate in the monitoring of their VMs by deploying a series of security policies according to their outsourced workload. This architecture consists of three modules. First, a monitoring agent module aims to monitor the guest operating system by deploying an agent into each VM. Second, a VM management console provides cloud users a graphical interface to participate in the management and monitoring (e.g., managing security policies, view VM states). Lastly, a privileged monitoring model runs directly in the hypervisor, to support the other two modules and necessary operations. Through this architecture a set of security validation functions can be performed, for example validating the integrity of guest OS kennel code and data, monitoring the state of VMs, and providing cloud user with an operating platform.

Moreover, the secure and flawless interaction of application instances running on different VMs in different virtualized cloud environments has to be validated as well. Therefore, an automated model is proposed that consists of three layers: local application surveillance (LAS), intra-platform surveillance (IPS), and global application surveillance.[216] Each application instance is monitored by an LAS component, to examine if the instance violates any established monitoring rule and to detect malicious behavior or implementation flaws. [217] Furthermore, to monitor interaction problems between different virtualized environments, IPS components are allocated to each VM and are also inter-connected with other IPS components of the same virtualized environment. IPS components evaluate the results of the LAS components from their allocated VM and check for security risks that might arise through interaction of different applications or VMs. Lastly, global

---

[214] Cf. this and the following two sentences Sharif et al. (2009), p. 479, 485-486.

[215] Cf. this paragraph Chen, Wen (2012), p. 444-446.

[216] Cf. Gonzalez, Munoz, Mana (2011), p. 293-295, and Mana, Munoz, Gonzalez (2011), p. 3-5.

[217] Cf. this and the following sentences Gonzalez, Munoz, Mana (2011), p. 293-295, and Mana, Munoz, Gonzalez (2011), p. 3-5.

application surveillance components analyze data from different VMs referred to the same application. Therefore, these components receive and analyze information from several IPS components and have a global view of an application behavior in different virtualized environments.

To automatically determine the status of applications, a framework consisting of specific VM and analyzer modules for virtualized cloud environments is proposed.[218] By using this framework, providers can detect attacks on executables by noticing measurement changes, thus increasing the security of VMs. A prototype of this framework was implemented and tested, and has demonstrated performance efficiency. However, this framework lacks the ability to detect dynamic attacks on running applications in virtualized environments.[219] To detect dynamic attacks a control module in the privileged VM, and measurement modules are located in each guest VM.[220] This privileged VM is started by the hypervisor and runs the host operating system. For each VM a light measurement module is constructed and is responsible for receiving requests from the guest VMs, measuring running applications on demand. This monitoring data is transferred to a control center. This control center analysis data received from measurement modules to detect dynamic attacks on applications. Table 6-3 recapitulates the presented monitoring methods and techniques.

In-VM- and Out-of-VM-Monitoring
In-VM-Monitoring (i.e., monitoring component resides in the same VM environment) and Out-of-VM-Monitoring (i.e., a monitor is located and isolated in a separate VM) can be implemented to monitor and assure secure virtualized environments.
Cloud User Monitoring VM Approach
To enable cloud users to participate in the monitoring of their VMs, a series of security policies, and corresponding monitoring agents can be deployed.
Application Monitoring Model

[218] Cf. this and the following two sentences Liu et al. (2010), p. 56-62.

[219] Cf. Wang, Mao, Luo (2012), p. 761.

[220] Cf. this and the following four sentences Wang, Mao, Luo (2012), p. 762.

Monitors the secure and flawless interaction of application instances running on different VMs in different virtualized cloud environments by using a layered VM monitoring architecture.
Framework for increasing VM Security
Determines the status of applications and detects attacks on executables in virtualized cloud environments.
Detect Dynamic Attacks on virtualized Applications
A control module in the privileged VM, and measurement modules in each guest VM can be located to detect dynamic attacks on running applications in virtualized environments.

Table 6-3 Overview of methods and techniques to monitor virtualized environments.

6.4 Intrusion, Anomaly and Behavior of Malware Detection

Security is one of the most important and most discussed topics concerning cloud computing. Especially by establishing CSCs, auditors verify the implementation of proper security mechanisms. CSs form a highly valuable target for attackers and are particularly exposed to risks of malicious behavior from external attackers, CS customers as well as malicious employees.[221] Thus, auditors need to continuously verify, whether a CSP has established and operates mechanisms to prevent intruders from performing malicious operations.

Intrusion detection systems have been a research area of security monitoring since the beginning of the 1980s.[222] Lunt (1993) provides one of the first surveys on automated and real-time intrusion detection techniques and systems, including the use of neural networks, expert systems, and model-based reasoning for intrusion detection.[223] Such techniques monitor and analyze the behavior and actions of users, compare them to established norms and past behavior, and check for suspicious events (e.g., sudden late hour accesses) to provide evidence by interpreting monitoring logs.[224] Likewise, process mining techniques can be used to analyze monitoring logs to detect low-level intrusions and

[221] Cf. European Network and Information Security Agency (2009), p. 10, Subashini, Kavitha (2011), p. 7,9, and Kaufman (2009), p. 63.

[222] Cf. Hasan, Stiller (2005), p. 122.

[223] Cf. Lunt (1993), p. 409-413.

[224] Cf. Lunt (1993), p. 409-413.

to prevent high-level fraud.[225] Network traffic exchange can be monitored as well, to detect anomalies and intrusions.[226] The presented techniques mainly differ in matching user behavior as well as detection of suspicious events. More recently, machine-independent approaches for intrusion and anomaly detection using a knowledge-based system have been proposed.[227] Knowledge-based systems perform, for instance, intelligent analyses of operating system audit trails and assess unauthorized user activity in multi-user computer systems.

Traditional intrusion detection and prevention techniques need to be adjusted to deal with the challenges of CS characteristics. First, a large number of customers using (different) CSs will greatly increase the number of event records and the complexity of monitoring logs, compared to traditional in-house application usage. Secondly, due to virtualized and distributed cloud architectures, the detection of malicious behavior will be more difficult. Modi et al. (2013) provide a comprehensive overview of intrusion detection and protection system in cloud contexts. They present for example network and host based intrusion detection systems, which are installed on external or virtual networks, and on each VM, or host systems to identify intrusions by monitoring the host's file system, system calls or network traffic.[228] Likewise, hypervisor based intrusion detection systems monitor and analyze communications between VMs, between the hypervisor and VMs, and within the hypervisor based virtual network. Based upon the auditee´s context, auditors recommend implementing web application firewalls to improve the security of networks by analyzing data traffic and blocking attackers [i01, i04]. Furthermore, next generation advanced persistent threat control[229] is recommended for large CSPs [i01, i04].

CSs are not only exposed by risks of external attacks, but also are confronted with malicious insider and cloud customer behavior.[230] Authorized employees of the CS can cause harm to the cloud infrastructure or to the assets of the company in general. Since employees are trusted they can move easily within the organizations compared to outsiders, and

---

[225] Cf. van der Aalst, de Medeiros (2005), p. 4.

[226] Cf. Zachary, McEachen, Ettlich (2004), p. 1.

[227] Cf. this and the following sentence Best, Mohay, Anderson (2004), p. 85-86.

[228] Cf. this and the following sentence Modi et al. (2013), p. 54.

[229] See for example FireEye Inc. (n.y.).

[230] Cf. this and the following two sentences Ghulam, Shaikh, Shaikh (2008), p. 1.

have access to confidential and customer data. Besides the usage of personalized user accounts, the principle of least privilege, or jump servers which enable administrators to manage different security zones in networks [i01, i04], CSPs can continuously record and profile employee behavior, to avoid, detect and recover malicious insider threats. These profiles can be analyzed and compared with an organization´s pre-defined (access) policy.[231] To automate this profiling and recoding process, an agent-based model is proposed. This model consists of three main components: system profile agents, manager agents, and a database. DAs are deployed to machines when an employee logs into the system, build a profile of the user, and monitor activities. A manager agent collects information from these agents monitoring different employees on the entire network, and sends these reports to a database. This database stores up-dated profiles of employees, which can be further automatically analyzed and compared to defined (access) policies.[232] When an insider intends to carry out a malicious act, an alarm is triggered so that potential threats might be avoided even before occurring. "The organization may also declare a threshold that will help to make decision regarding the acceptability or unacceptability of the behavior of the employee."[233] However, because of employee privacy concerns, organizations have to determine whether they want to monitor every activity or only a defined set of activities (e.g., critical administrator activities). Furthermore the organization has to evaluate whether or not employees are informed about being monitored.[234]

Moreover, when different users interact with system files simultaneously, malicious modification of such system files affects all users. For that reason, file system integrity must be continuously ensured, especially in the context of multitenant CSs. Several file system integrity tools have already been developed and allow administrators to automatically detect system changes and malicious file modifications.[235] However, the concept of file system integrity validation has to be adjusted for virtualized cloud environments. Implementing monitoring processes on VMs and modules into hypervisor levels is one proposed approach for matching requirements of virtualized environments. Due to low

---

[231] Cf. this and the following sentences Ghulam, Shaikh, Shaikh (2008), p. 1.

[232] Cf. this and the following sentence Ghulam, Shaikh, Shaikh (2008), p. 2.

[233] Ghulam, Shaikh, Shaikh (2008), p. 3.

[234] Cf. Ghulam, Shaikh, Shaikh (2008), p. 2.

[235] Cf. this and the following three sentences Kim, Kim, Eom (2010), p. 335-336.

performance overhead, this approach enables real-time file system monitoring, which is particularly suitable for virtualized CS contexts.

When intrusions and anomalies are automatically and continuously detected, information overload is likely to appear, thus leading to limited decision making and action taking.[236] For that reason, an architecture was proposed that automatically and continuously detects, aggregates and evaluates detected anomalies. This architecture consist of several layers: a monitoring layer for detecting anomalies, an aggregation layer for grouping anomalies, an evaluation layer for drawing conclusions and finally a decision layer, to conduct system wide decisions. Furthermore, to relieve and support cloud administrators, and to improve detection efficiency, several intrusion detection visualization techniques are proposed.[237] Table 6-4 summarizes presented methods of this cluster.

Intrusion Detection Systems
Techniques to monitor and analyze behavior and actions of users to check for suspicious events and detect intrusions. In the context of cloud computing network, host based, and hypervisor based intrusion detection systems are recommended.
Insider Monitoring
Authorized employees can cause harm to the cloud infrastructure or to the assets of the company, hence, CSPs can continuously record and profile employee behavior, to avoid, detect and recover the malicious insider threat.
File System Integrity Tool
A tool which allow administrators to automatically detect system changes and malicious file modifications.
Anomaly Detection and Aggregation Architecture
An architecture that automatically and continuously detects anomalies and automatically aggregates and evaluates detected anomalies to reduce information overload.

Table 6-4 Overview of methods to detect intrusions, anomalies and malicious behavior.

[236] Cf. this and the following two sentences Perols, Murthy (2012), p. 35,36, 43-48.

[237] Cf. Zhao, Zhou, Fan (2012), p. 11.

6.5 Service Level Agreements Monitoring

Adherence to SLAs has to be continuously monitored, especially in context of cloud computing, in which customers have a lack of control.[238] Auditors might be informed about ongoing SLA violations to adjust their certification reports [i01, i04].

To evaluate and validate adherence to SLAs, measurable requirements or service level objectives have to be specified, for example, expected availability, throughput, or response time.[239] Thus, a description of SLAs containing information about key performance indicators of interest, and conditions which define compliance/non-compliance are necessary.[240] Afterwards, requirements have to be transformed into a formal, machine-understandable representation, for instance, an ontology-based representation, [241] Domain Specific Languages,[242] using WS-Agreements,[243] or using a combination of XML, formulas, and logics.[244] To automatically monitor the adherence of specified criteria, mechanisms (e.g., DAs) have to be incorporated into the cloud environment to make assertions, ask queries, or gather necessary information.[245]

When services are replaced at runtime or terms of SLAs are changing dynamically, applied monitoring mechanisms have to be adjusted.[246] In some cases, applied mechanisms may not be any longer applicable, for example, when a replaced service might not be able to provide runtime events required for monitoring SLAs. Hence, in these dynamic environments, it is necessary to check whether or not the ability to monitor SLA terms and conditions is affected by the changes.[247] Moreover, the deployed monitoring infrastructure has to be modified in order to ensure the continuous execution of the required runtime checks. An additional monitoring management layer can be implemented to perform these

---

[238] Cf. European Network and Information Security Agency (2009), p. 9.

[239] Cf. Goel, Kumar, Shyamasundar (2011), p. 110, and Lamparter, Luckner, Mutschler (2007), p. 3-4.

[240] Cf. Romano et al. (2011), p. 46, 48.

[241] Cf. Lamparter, Luckner, Mutschler (2007), p. 3, and Romano et al. (2011), p. 48.

[242] Cf. Emeakaroha et al. (2012), p. 1021-1022.

[243] Cf. Romano et al. (2011), p. 48.

[244] Cf. Goel, Kumar, Shyamasundar (2011), p. 110-113.

[245] Cf. Goel, Kumar, Shyamasundar (2011), p. 114-116, Lamparter, Luckner, Mutschler (2007), p. 2-3, Romano et al. (2011), p. 48, and Emeakaroha et al. (2012), p. 1021.

[246] Cf. this and the following sentences Comuzzi, Spanoudakis (2010), p. 2414.

[247] Cf. this and the following sentence Comuzzi, Spanoudakis (2010), p. 2414.

checks and modify the monitoring infrastructure accordingly.[248] First, SLA terms, specifying the functional and non-functional properties that a service should provide, have to be matched with the monitoring capabilities of services that are currently deployed or can be deployed. These capabilities include event reporting (e.g., which service events are reported) and the SLA checking capabilities of the service (e.g., supported SLA term specification languages), and can be represented as XML based schemes. Based upon this matching, it can be decided, whether a SLA term can be monitored. In case a term cannot be monitored, the monitoring is delegated to local or external service monitor (e.g., monitor services on the network).[249]

The identified approaches were developed in the context of web services and internet standards,[250] thus they seem to be suitable to be used for semi automated monitoring of CSs. Table 6-5 categorizes these SLA monitoring approaches.

SLA Monitoring
To evaluate and validate adherence to SLAs, service level objectives have to be specified and transformed into a machine-understandable representation. To automatically monitor the adherence, mechanisms (e.g., digital agents) have to be incorporated into the cloud environment to make assertions, ask queries, or gather necessary information.
Dynamic SLA Monitoring
When services are changing dynamically, applied monitoring mechanisms have to be adjusted. An additional monitoring management layer can be implemented, to modify the SLA monitoring infrastructure accordingly.

Table 6-5 Summary of SLA monitoring concepts.

6.6 Compliance Monitoring

CSPs have to assure that the execution of their business processes is in accordance with a multitude of requirements, for example with laws and regulations (e.g., Sarbanes Oxley Act, Basel II), standards (e.g., ISO/IEC 27000-series), commercial contracts (e.g.,

[248] Cf. this and the following three sentences Comuzzi, Spanoudakis (2010), p. 2414, 2415.

[249] Cf. Comuzzi, Spanoudakis (2010), p. 2416, 2419.

[250] Cf. Goel, Kumar, Shyamasundar (2011), p. 109, Comuzzi, Spanoudakis (2010), p. 2414, and Lamparter, Luckner, Mutschler (2007), p. 1.

nondisclosure agreements), or organizational policies.[251] In this context, being compliant refers to showing that business process executions and data accesses adhere to this multitude of requirements. Similar to SLAs, laws, regulations and standards that are described in a textual form have to be interpreted for a business domain and transformed into compliance policies.[252] The adherence to such policies can then be observed by monitoring technologies,[253] for instance, by using event monitoring technologies that allow the monitoring of distributed and heterogeneous IT systems.[254]

Other compliance rules impose requirements on CSs, for instance, regarding their configuration and security. To assure compliance to such rules, CSs can be automatically analyzed, and compliance can be validated at the time services are created.[255] Such a validation system might use DAs and scripts to check, for example, if anti-virus software is running with the latest signature file and all available security patches are applied. Such an automation solution has already been implemented and deployed in a private enterprise cloud and in several customer dedicated private clouds.

However, monitoring of compliance requirements might be not feasible in some cases.[256] Hence, only compliance violations can be detected. Practitioners emphasize practical limitations of compliance monitoring as well [i04]. Especially business processes that rely heavily on human interactions, are highly unstructured, or lack proper documentation hamper (semi) automated compliance monitoring.[257] However, Doganata, Curbera (2009) present how to track compliance of an unmanaged business process by using a monitoring tool based on business provenance technology.[258] Furthermore, it was recommended to

---

[251] Cf. this and the following sentence Sackmann et al. (2008), p. 79, and Sackmann, Kähmer (2008), p. 366.

[252] Cf. Sackmann, Kähmer (2008), p. 366 and regarding compliance policies see for instance Ashley et al. (2002), and Giblin, Mueller, Pfitzmann (2006).

[253] Cf. Sackmann, Kähmer (2008), p. 367.

[254] Cf. Giblin, Mueller, Pfitzmann (2006), p. 2, 3.

[255] Cf. this paragraph Chieu et al. (2012), p. 285-288, 290.

[256] Cf. this and the following sentence Giblin, Mueller, Pfitzmann (2006), p. 3.

[257] Cf. Doganata, Curbera (2009), p. 310.

[258] See Doganata, Curbera (2009), p. 310-311, and Curbera et al. (2008), p. 100-101.

request reports concerning compliance adherence, accompanied by random validation tests to improve report reliability [i04].

Data protection issues regarding security and privacy aspects, such as preventing user's sensitive data from illegal disclosure or malicious violation, are hampering widespread adoption of cloud computing.[259] Thus, assuring compliance to privacy policies is of critical importance in cloud computing contexts. By using a role based access control model and an active monitoring scheme, data protection can be improved in distributed cloud scenarios.[260] A role based access control model validates that cloud resources are being accessed or managed legally according to predefined data protection policy. However, some security attacks still may deploy bugs or vulnerabilities of the system to illegally bypass this access control layer. Therefore, a monitoring scheme is implemented that provides ongoing monitoring and is "capable of tracing, tracking, and triggering an alert on any operation, data or policy violations in […] cloud environment."[261] Invalid behavior violating policies will be automatically alerted to a specific data owner or administrator.[262]

Similar to general data protection policies, potential cloud customers might place restrictions on transborder data flows and data location.[263] Thus, especially in cloud federations, adherence to these restrictions has to be enforced. Cloud federations are interoperable heterogeneous cloud environments that interact together, for example by sharing and cross-managing VMs. A VM can be deployed from one member of the federation onto the infrastructure of another site of the federation in different geographical locations. In general, the infrastructure provider can decide whether using such a federation architecture is made transparent and communicated to its users. Therefore it must be possible for clients and auditors to monitor the management of VMs. An architecture to enable data location monitoring is proposed. First, CS users have to express their data locations requirements, for example by using XML definitions.[264] When deploying VMs,

---

[259] Cf. Chen, Hoang (2011), p. 550.

[260] Cf. this and the following two sentences Chen, Hoang (2011), p. 550,553.

[261] Chen, Hoang (2011), p. 551, 553.

[262] Cf. Chen, Hoang (2011), p. 553-554.

[263] Cf. this paragraph Massonet et al. (2011), p. 1511.

[264] Cf. this and the following sentences Massonet et al. (2011), p. 1512, 1513.

additional service probes are initiated to monitor VM operations, e.g., start, shut down, and migration to other physical machines. This monitoring information is published to CS users, who in turn can log necessary information in the appropriate format for auditing. When VMs are immigrated to new physical machines, (e.g., because of performance scales), new federation candidates are evaluated regarding the users requirements in advance. However, regular external auditing is still required to assure that the monitoring captures all relevant information.[265] External auditors may provide services to automatically check such audit logs based upon defined requirements, or users can ensure the adherence by analyzing logs themselves. Table 6-6 summarizes compliance validations concepts.

Cloud Service Configuration Validation
Method to validate the configuration of CSs including security and compliance adherence, at the time services are created.
Compliance Validation of unstructured Business Processes
Compliance adherence of business processes that rely on human interactions, and are unstructured might be monitored based upon business provenance technology.
Data Protection Compliance
By using a role based access control model and an active monitoring scheme, data protection can be improved in distributed cloud scenarios.
Data Location Compliance
Deploying additional probes onto VMs that are monitoring VM operations (e.g., start, shut down, and migration to other physical machines) to assure restrictions on data locations.

Table 6-6 Overview of methods to monitor and assure adherence to compliance requirements.

6.7 Network Monitoring

Furthermore, a dynamic network monitoring method to ensure network reliability and gather network information was identified. This method is based on the incorporation of DAs and was developed based on the cross-platform language Python.[266] It enables automatic network monitoring as well as manual intervention (e.g., on emergency events).

[265] Cf. Massonet et al. (2011), p. 1514.

[266] Cf. this and the following two sentences Wu, Zhao, Ye (2008), p. 637.

Experimental results show that the method can be used within various network environments and topologies. In addition, network monitoring through the usage of DAs can be realized as a multilevel architecture. For example, probes are deployed on network nodes, which report gathered information to higher-level agents.[267] These higher level agents then aggregate and analyze data, and transmit their results to higher agents, to create a global monitoring. This agent communication can be realized through XML based files or data. Thus, this method may be used to automatically monitor cloud computing networks. Table 6-7 depicts this network monitoring technique.

Network Monitoring
Gathering information about network operations and ensure network reliability by using (layers) of digital agents and service probes.

Table 6-7 Network monitoring technique.

7. Recommendations for Dynamic Certification

In the following sections, design recommendations and guidelines that were derived during this work will be presented and discussed. Moreover, a first model of dynamic certification is presented, comprising processes and components to assure ongoing certification adherence. In addition, examples regarding how to continuously assess CSC criteria are outlined. Finally, open issues that need to be addressed in further research are presented.

7.1 Design Recommendations and Guidelines for Dynamic Certification

By analyzing extant literature on CM and CA, interviewing CSC auditors as well as observing a CSC audit, a variety of design recommendations and guidelines for dynamic CSC were derived and will be discussed in the following. The derived design recommendations and guidelines are summarized at the end of this section in table 7-1.

When applying dynamic cloud certifications, an adequate and *individual certification scope* has to be defined based upon the certified CS and auditee´s context [i01, i03, i04]. On average, current *Certify* CSCs require about 30-50 man-days of work for initially issuing the CSC [i01, i03]. However, when designing dynamic certifications, CS complexity, extent of implemented cloud systems, and offered service functions vary greatly

[267] Cf. this and the following two sentences Nowak, Bagrij (2007), p. 4-5.

among different CSP and have to be considered [i03]. In addition, the size of the auditee´s enterprise, number of employees as well as the level of technical knowledge and skills influence and limit the scope of dynamic certifications [i04]. Ensuring *economic feasibility* of dynamic certification is of critical importance: "*You have to keep in mind, what a provider is able to achieve on a monthly or quarterly basis. A continuous audit must always be economically achievable for him*" [i03].

Furthermore, several challenges have to be faced when defining and specifying CM and CA processes. First, a differentiation between performing CM and CA has to be made. Performing CM by a CSP (or specialized third parties offering monitoring services) forms a prerequisite for auditors to perform efficient CA. More importantly, when performing CA, it has to be ensured that CSPs do not outsource their monitoring and assessment processes to the auditor [i03]. Thus, a precise *distinction between monitoring and auditing responsibilities* is required [i03, i04], which is as well important for reliable and secure day-to-day operations [i01]. For instance, a CA of system vulnerabilities on a monthly basis might be viewed as a substitute for internal vulnerability management by a CSP. As a consequence, practitioners recommend that an auditor gathers the results of vulnerability analysis from CSP on a monthly basis and assesses certification adherence on a quarterly or semiannually basis in the context of vulnerability management. Current CSCs are mostly based upon manual auditing operations, for example, performing interviews and analyzing documents. However, dynamic certification cannot be realized solely manually due to continuous costs and expenditures, hence, *process automation* is required.[268] Alles, Kogan, Vasarhelyi (2008) suggests that such an automation of processes is likely to be incremental rather than disruptive, since auditors will likely attempt to first automate existing processes rather than developing technology enabled auditing processes.[269] Nonetheless, a dynamic certification requires auditors to build up extensive knowledge about auditee´s systems and contexts to validate ongoing certification adherence [i04]. Hence, relying only on automated auditing methods is not feasible. Instead different *data sources*,

[268] Cf. Kunz, Niehues, Waldmann (2013), p. 522, Bezzi, Kaluvuri, Sabetta (2011), p. 41, Brown, Wong, Baldwin (2007), p. 21, Schneider, Lansing, Sunyaev (2013), p. 16, and Woodroof, Searcy (2001), p. 1.

[269] Cf. Alles, Kogan, Vasarhelyi (2008), p. 2-3.

for instance, interviewing employees, analyzing auditee´s processes, and external information repositories need to be incorporated into the concept of dynamic certification.

Practitioners emphasize that the auditee environment is characterized by a great heterogeneity. Typically, individual, customized or legacy systems are analyzed and certified [i03, i04]. In some cases a CSP might even outsource (parts of) their IT department, leading to entangled supply chains that have to be faced in certification processes [i04]. Hence, *suitable auditing and monitoring methods* have to be implemented and adjusted based upon the auditee´s context. In addition, the frequency of performing CA operations depends on the CS type and is influenced by auditee´s operations and processes, hence the frequency should be aligned to the certification context as well. It is important that a dynamic certification does *not inhibit* this *individualism* of auditees [i01, i04].

Furthermore, it was noted that auditors do not recommend specific technical solutions that have to be implemented by the auditee for certification adherence [i01, i04]. Instead, they analyze existing system architectures and processes, identify and evaluate problems and vulnerabilities, and recommend potential solutions as well as enhancements [i01, i04]. Thus, when certifying IT systems and architectures a variety of feasible and valid solutions have to be taken into consideration: "*This in one of the biggest challenges for technical auditors. […] Technical audits are more relative. […] [An auditee] might implement a different solution, you have never thought of, but his solution is still valid [according to the certification requirements]*" [i04]. As a consequence, the concept of dynamic certification should incorporate different processes and solutions, but should be detached from specific technologies (i.e., *technological abstraction*) [i01, i04]. Likewise, when designing dynamic CSCs it has to be ensured that recommended and implemented CM and internal CA structures and processes can be used without having to rely on specific auditors to *prevent auditor lock-in* [i01, i04]. Hence, such structures and processes should be mostly standardized and portable to reduce auditor dependency, potential sunk and switching costs [i04].

Moreover, several *legal and regulatory requirements* have to be considered when designing dynamic certifications. Such requirements might limit data gathering or impose additional efforts and expenditures.

Finally, *assuring security, privacy, confidentiality, and integrity* is of critical importance when designing dynamic certification processes and systems [i01, i04].[270] One should always act according to the maxim to reduce risks and potential threats for auditee´s and auditor´s operating systems [i03]. Similarly, monitoring and auditing systems should be maintainable (e.g., modify existing modules), reliable (e.g., low performance impacts and high availability),[271] and adaptable (e.g., updating and adjusting modules to changes).[272]

Design Recommendation and Guidelines	**Description**
Individual certification scope	When applying dynamic CSCs, an adequate and individual certification scope has to be defined based upon the certified CS and auditee´s context.
Ensuring economic feasibility	When designing dynamic certifications, economic feasibility for auditors and CSPs has to be ensured.
Distinction of responsibilities	A precise distinction between monitoring and auditing responsibilities is required, to avoid outsourcing of monitoring and assessment processes to auditors.
Process automation	Dynamic certification cannot be realized solely manually due to continuous costs and expenditures, hence process automation is required.
Incorporate different data sources	Different data sources, for instance, interviewing employees, analyzing auditee´s processes, and external information repositories need to be incorporated into the concept of dynamic certification.
Method and frequency adjustments	Suitable auditing and monitoring methods as well as corresponding operation frequencies have to be settled and adjusted based upon the auditee's context, since auditee environment is characterized by a great heterogeneity.
Respect auditee´s individualism	Individual, customized or legacy systems are being analyzed and certified. Therefore, it is important that a

[270] See Sujana, Revathi (2012), p. 97, and Woodroof, Searcy (2001), p. 5 as well.

[271] Cf. Lin, Lin, Liang (2010), p. 417, and Woodroof, Searcy (2001), p. 5.

[272] Cf. Vasarhelyi et al. (2004), p. 12, and Alles et al. (2006), p. 151.

	dynamic certification does not inhibit this individualism of auditees.
Technological abstraction	When certifying IT systems and architectures a variety of technical feasible and valid solutions have to be taken into consideration. As a consequence, the concept of dynamic certification should incorporate different processes and solutions, but should be detached from specific technologies.
Prevent auditor lock-in	When designing dynamic CSCs it has to be ensured that recommended and implemented CM, and internal CA structures and processes can be used without specific auditors to prevent auditor lock-in.
Consider legal and regulatory requirements	Legal and regulatory requirements have to be considered when designing dynamic certifications. Such requirements might limit data gathering or impose additional efforts and expenditures.
Assuring security, privacy, confidentiality, and integrity	When designing dynamic certification processes and systems security, privacy, confidentiality, and integrity have to be assured.

Table 7-1 Design recommendations and guidelines for dynamic certification.

7.2 A conceptual Model of Dynamic Cloud Service Certification

A dynamic CSC comprises a broad range of different processes, operations and computer-assisted systems. Based upon insights gained by analyzing relevant publications in the research area of CM and CA, participating in workshops, accompanying a CSC audit, and discussing the concept of dynamic certification during the interviews, an initial conceptual model of dynamic certification was developed, comprising necessary processes and components to assure ongoing certification adherence. Figure 7-1 illustrates this model.

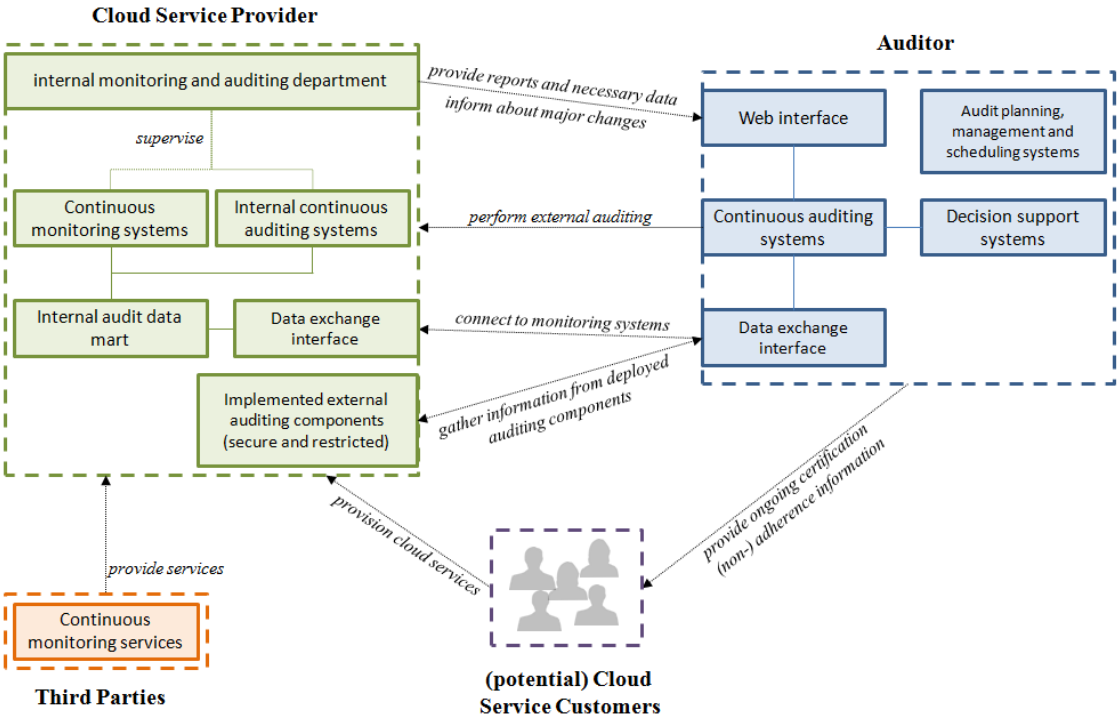

Figure 7-1 Conceptual model of dynamic certification.

Interviews and field observations revealed that external auditing capabilities are limited due to technical, organizational and legal reasons. First, most auditees are not willing to permit auditors' to integrate external auditing modules or software, or to allow external system access due to security and privacy concerns [i01, i03, i04]. Second, auditor´s knowledge about auditee´s system and processes is limited due to the nature of focusing on potential security problems [i01, i04]. Third, auditors are hesitant to externally interfere with auditees system to prevent security vulnerabilities, as well [i04]. Fourth, implementing and developing auditing modules and software requires high expenditures and efforts due to heterogeneous auditee´s systems. Finally, legal requirements or organizational policies might prohibit or limit auditors auditing operations. To cope with these challenges, CSPs need to establish an internal monitoring and auditing department (iM&A-Department). Consequently, this department has to perform extensive CM operations, comprising monitoring of virtualized environments, intrusion detection and prevention, SLAs and compliance monitoring as well as network monitoring. Likewise suitable cloud monitoring tools and architectures as well as appropriate logging facilities have to be implemented (see section 6). More importantly, this iM&A-Department forms a bridge between the CSP and auditor when performing a dynamic certification. On the one hand, an auditor can communicate and interact with this iM&A-Department when performing CA operations. On the other hand, the iM&A-Department manages and

supervises CM operations, gathers, processes and provides audit relevant information to the auditor. In addition to establishing CM processes, the iM&A-Department has to implement internal auditing processes and systems to meet dynamic certification requirements and challenges. Hence, an iM&A-Department can implement presented CA methods (see section 5), since these are mostly developed for internal auditing contexts. For instance, by deploying internally a team of DAs or implementing a MCL, an iM&A-Department can gather data and information across implemented cloud monitoring tools to prepare monthly reports, which are requested by auditors. Moreover, an internal audit data mart can be implemented, which stores audit-relevant data.

When establishing an iM&A-Department several advantages can be achieved compared to dynamic certification contexts that solely rely on external auditing. First, auditee resistance will decrease and acceptance will increase when auditors do not interfere directly with auditee´s systems [i01, i04]. Second, employees of the iM&A-Department possess, or can easily access the required knowledge about internal processes and cloud systems [i04]. Third, audit relevant data and information can be gathered and processed internally, hence reducing security and privacy concerns [i04]. Fourth, instead of implementing standardized or inappropriate external modules and software, an auditee can implement proprietary and customized internal auditing techniques aligned to their customized cloud architecture [i04]. Finally, efforts and expenses for auditors are reduced when CSPs perform internal audits. However, a CSP has to ensure that appropriate monitoring and internal auditing resources are allocated and integrated into daily operational management, and employee responsibilities are settled [i01, i04]. Thus, organizational structures have to be adjusted to meet dynamic certification requirements [i01, i04]. Preserving autonomy and auditor independence have to be considered to prevent auditor lock-in [i04], when establishing an iM&A-Department, and corresponding processes and systems. Moreover, auditees might incorporate external monitoring services and software from specialized third parties (e.g., using an automated vulnerability scanner as a service) [i04].

To assess ongoing certification adherence, auditors request auditees to provide reports and data according to defined frequencies. Therefore, auditors should offer, for instance, a web interface to upload and enter auditee´s data, or to inform auditors about major changes. "*[Auditees] should produce standardized reports which the auditor receives and automatically assesses. I think this would be a secure solution. [An auditor can] offer a*

special service where people can upload their data" [i03]. Likewise, auditees might transfer monitoring logs that auditors analyze to assess criteria adherence [i01, i03]. When receiving and storing auditee´s reports and data, confidentiality and privacy have to be assured, thus, auditor´s systems have to be securely designed to minimize risks and potential threats [i03]. To prevent report manipulation, it is recommended to randomly perform validation tests on regularly basis [i03].

For example, when validating adherence to criterion 'regularly performing reviews of firewall rules' an auditee can upload a short report to a web server that comprises, for instance, the following information: date, firewall policy version x.x, number of offending firewall rules, initiated operations and changes made [i04]. On a quarterly basis an auditor analyzes these reports and performs random validation checks (e.g., assessing latest firewall policy) [i01, i04]. Likewise, adherence to criterion 'regularly performing vulnerability tests' can be assessed based upon receiving vulnerability reports [i01, i04]. These reports should not only provide information about identified vulnerabilities but also contain information about initiated and performed operations to fix the identified vulnerabilities [i01, i04].

Aside from receiving regularly reports, auditors might realize synergy effects when connecting to existing CM systems [i03]. However, auditee´s resistance due to security and privacy concerns as well as auditee heterogeneity will restrict and limit possible implementation solutions [i01, i03, i04]. Still, during the interviews and field observation three potential solutions were identified. First, standardized monitoring tools that provide export functionalities (e.g., Nagios) seem to be suitable [i03, i04]. Second, standardized reports from vulnerability scanners can be exported and transferred to the auditor. It was noted that especially vulnerability information provides strong auditing evidence [i01, i03, i04]. Lastly, a CSP might offer external monitoring services for their cloud customers. Such existing services might be modified according to certification contexts and used by auditors. Likewise, limited external CA can be performed. In general, cloud components that are connected to the internet can be (automatically) tested and scanned [i03]. *"Of course you can automatically scan components that are reachable from the outside. You can scan and check everything that is connected to the internet"* [i03]. Hence, performing external vulnerability scans and interceptor tools can be used to analyze cloud systems [i04], and service availability and encryption can be assessed externally as well

[i03]. In addition, external operations can be performed based upon criteria requirements. For example, a criterion claims that a ‘security incident handling team has to be available 24h, 7 days a week’. It can be externally audited by performing automated telephone calls or automatically sending predefined and computerized trouble-shooting tickets, and assessing CSP responses [i04].

To improve audit efficiency, expedite decision-making processes, and to cope with potential alarm floods, auditors should implement a suitable DSS. DSSs can be used to aggregate gathered information, and efficiently and automatically decide to take actions or to alert the auditor, based on the aggregated and analyzed evidence (see section 6.3). However, manual assessing and decision making is still required, because, for instance, a new security vulnerability might have no impact on cloud systems since other security mechanisms prevent attackers from exploiting the identified vulnerability [i01, i04]. Thus, extensive knowledge about auditee´s systems is necessary as well [i04]. Furthermore, these DSSs might trigger additional auditing operations based upon external changes, for instance, announcement of new viruses or software vulnerabilities (e.g., Heartbleed vulnerability).

Aside from processes that deal with assessing ongoing certification adherence, dynamic certification comprises other mechanisms and concepts, which are out of scope of this thesis. For instance, incorporating continuous updating and improvement capabilities, managing certification violations, and informing (potential) cloud customers about certification (non-) adherence.

7.3 Open Issues

By interviewing CSC auditors and observing a CSC audit, a variety of design recommendations and guidelines for, and a first conceptual model of dynamic CSC was derived. Nonetheless, further research needs to consider the CSP´s perspective to identify additional technical, organizational and legal requirements as well as possible solutions for dynamic CSCs.

Moreover, a framework of dynamic certifications has to be developed, comprising precise descriptions of participating entities, processes (e.g., determining process inputs and outputs), interfaces and information flows, and standards as well as implementation recommendations [i01, i04]. Furthermore, different metrics have to be developed based upon

certification criteria and corresponding processes. As well, a framework and guidelines have to be specified, to handle violations of CSC criteria on a continuous basis.

For the dynamic certification of CSs to become widely adopted, it must be technologically and economically feasible. CSPs as well as auditors, must be motivated and have the expertise to participate in dynamic certification. To motivate them to participate, perceived benefits must be higher than perceived expenditures. In general, benefits related to dynamic certification will be difficult to specify and to quantify.[273] Still, CSPs and auditors might realize several advantages. First, internal CS processes and systems can be improved by implementing suitable monitoring techniques and evaluating continuous feedback about how they are performing.[274] In addition, CSPs receive ongoing expert assessments about their systems [i01, i03]. Second, improvements and enhancements of cloud infrastructure and processes after the initial certification can be considered earlier and reflected in the certification report due to ongoing assessment. Finally, CSPs can differentiate themselves in the cloud market by making their CSs more transparent to customers. Thus, they may gain competitive advantages. Moreover, through timely detection and continuous assurance for certification adherence, dynamic certification can improve the trustworthiness of auditors' CSCs [i03].[275] Auditors can counteract the lack of cloud customers' control in cloud computing environments by increasing the transparency regarding operations of CSPs.[276] Further research should focus on evaluations regarding acceptance and benefits of CSPs when participating in dynamic certification as well as drivers and inhibitors for CS customers' demand for CA.

8. Conclusion

The ever-changing cloud environment, fast update cycles, and the increasing adoption of business-critical applications from CSPs demand for highly reliable CSs. Dynamic certification of CSs can assure a high level of reliability to (potential) CS adopters. However, methods to efficiently and continuously audit CSs are still in their infancy. Existing work of academics and practitioners concerning continuous methods for monitoring and auditing information systems provide a useful basis for future research to develop continuous

[273] Cf. Brown, Wong, Baldwin (2007), p. 21.

[274] Cf. Brown, Wong, Baldwin (2007), p. 21-23.

[275] Cf. Windhorst, Sunyaev (2013), p. 414.

[276] Cf. concerning the lack of control European Network and Information Security Agency (2009), p. 9.

monitoring and auditing methods for CSs. With this work, a first step to increase trustworthiness of CSCs is provided, by identifying methods to continuously monitor and audit CSs and evaluating their practical applicability, and by deriving design recommendations and guidelines as well as developing a first conceptual model of dynamic certifications.

This work contributes to business practice by extending the *Certify* CSC requirements catalog and classifying contained CSC criteria to decide whether or not a high frequency auditing is required, after the initial certification process is accomplished. More importantly, this work illustrates (semi) automated methods, which can be used in practice to enable continuous monitoring and auditing of CSs as well as (distributed) information systems in general. Additionally, some of these methods have already shown to be efficient in productive use. Finally, the mapping of CSC criteria and identified methods provides a first starting point for auditors and providers to implement corresponding methods to assure criteria adherence. Furthermore, new business models, for instance, monitoring as a service, might emerge out of the context of dynamic certification to manage the demand for internal auditing and monitoring systems.

With this thesis, further contributions for research are made. First, the concept of continuous monitoring and auditing is transferred in a new context. Second, the taxonomy developed by Schneider et al. (2014) was improved by incorporating feedback that was gathered during the criteria assessment workshops. Third, a comprehensive overview of (semi) automated monitoring and auditing is presented and can be used for future research. Further on, the applicability of these methods for CSPs to continuously monitor their CSs, and for (internal and external) auditors to enable CA of CSs is evaluated. In addition, challenges, limitations and benefits of dynamic CSCs are demonstrated. More importantly, a first set of design recommendations and guidelines was derived, which should be considered and incorporated in future research when planning to design and implement dynamic certification. Finally, a first conceptual model of dynamic certification is presented which incorporates several processes and concepts, hence, forming a basis for future research.

However, as the preceding discussion about open issues reveals, there is still plenty of research to do. It was recommended that future research should focus on developing a

dynamic certification process library similar to 'ITIL'[277] to guide auditors and providers [i04]. Likewise, future research should clarify how to manage certification violations, and if to inform (potential) cloud customers about certification (non-) adherence in context of dynamic certification. Moreover, mapping of criteria and methods revealed that some methods are still missing to (efficiently) assure ongoing criteria adherence, hence, further research is required.

[277] See ITIL (n.y.).

Bibliography

Accorsi (2011)

Rafael Accorsi: Anwenden struktureller Nicht-Interferenz zur Sicherheitsanalyse von Workflow-Modellen. 2011

Accorsi (2007)

Rafael Accorsi: Automated Privacy Audits to Complement the Notion of Control for Identity Management. Freiburg, Germany 2007

Accorsi, Lowis, Sato (2011)

Rafael Accorsi, Lutz Lowis, Yoshinori Sato: Automated Certification for Compliant Cloud-based Business Processes. In: Business & Information Systems Engineering. Iss. 3, Vol. 3, 2011, p. 145-154

Accorsi, Stocker (2008)

Rafael Accorsi, Thomas Stocker: Automated Privacy Audits Based on Pruning of Log Data. In: IEEE Computer Society (Ed.): 12th IEEE International Conference on Enterprise Distributed Object Computing (EDOC 2008), 15-19 September, Munich, Germany. New York, NY, USA 2008, p. 175-182

Aceto et al. (2013)

Giuseppe Aceto, Alessio Botta, Walter d. Donato, Antonio Pescapè: Cloud monitoring: A survey. In: Computer Networks. Iss. 9, Vol. 57, 2013, p. 2093-2115

ACL Services Ltd. (n.y.)

ACL Services Ltd.: ACL Solutions - Data-driven Insight for Better Assurance. http://www.acl.com/solutions/products/, last retrieval 10.10.2014

Aguado, Calero (2014)

Juan G. Aguado, Jose M. Alcaraz Calero: MonPaaS: An Adaptive Monitoring Platform as a Service for Cloud Computing Infrastructures and Services. In: IEEE Transactions on Services Computing. (in Press), 2014, p. 1-14

Ahmi, Kent (2012)

Aidi Ahmi, Simon Kent: The utilisation of generalized audit software (GAS) by external auditors. In: Managerial Auditing Journal. Iss. 2, Vol. 28, 2012, p. 88-113

Alhamazani et al. (2014)

Khalid Alhamazani, Rajiv Ranjan, Karan Mitra, Fethi Rabhi, Prem P. Jayaraman, Samee U. Khan, Adnene Guabtni, Vasudha Bhatnagar: An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art. In: Computing. April, 2014, p. 1-21

Alles et al. (2006)

Michael Alles, Gerard Brennan, Alexander Kogan, Miklos A. Vasarhelyi: Continuous monitoring of business process controls: A pilot implementation of a continuous auditing system at Siemens. In: International Journal of Accounting Information Systems. Iss. 2, Vol. 7, 2006, p. 137-161

Alles, Kogan, Vasarhelyi (2008)

Michael Alles, Alexander Kogan, M. Vasarhelyi: Audit Automation for Implementing Continuous Auditing: Principles and Problems. 2008

Amazon Web Services, Inc. (n.y.)

Amazon Web Services, Inc.: Amazon CloudWatch. http://aws.amazon.com/de/cloudwatch/, last retrieval 23.10.2014

Ardagna et al. (2012)

C.A Ardagna, E. Damiani, R. Jhawar, V. Piuri: A model-based approach to reliability certification of services. In: IEEE Computer Society (Ed.): 6th IEEE International Conference on Digital Ecosystems Technologies (DEST), 18-20 June, Campione d'Italia, Italy. New York, NY, USA 2012, p. 1-6

Ashley et al. (2002)

Paul Ashley, Satoshi Hada, Günter Karjoth, Matthias Schunter: E-P3P Privacy Policies and Privacy Authorization. In: ACM (Ed.): Proceedings of the 2002 ACM Workshop on Privacy in the Electronic Society, 21 November, Washington, DC, USA. New York, NY, USA 2002, p. 103-109

Badach, Hoffmann (2007)

Anatol Badach, Erwin Hoffmann: Technik der IP-Netze. TCP/IP incl. IPv6; Funktionsweise, Protokolle und Dienste. 2. ed., München 2007

Badger et al. (2012)

Lee Badger, Tim Grance, Robert Patt-Corner, Jeff Voas: Cloud Computing Synopsis and Recommendations. Recommendations of the National Institute of Standards and Technology. Gaithersburg, MD, U.S. 2012

Baksa, Turoff (2011)

Robert Baksa, Murray Turoff: Continuous Auditing as a Foundation for Real Time Decision Support: Implementation Challenges and Successes. In: Frada Burstein, Patrick Brézillon, Arkady Zaslavsky (Ed.): Supporting Real Time Decision-Making 2011, p. 237-252

Bernnat et al. (2012)

Rainer Bernnat, Wolfgang Zink, Nicolai Bieber, Joachim Strach: Das Normungs- und Standardisierungsumfeld von Cloud Computing. Eine Untersuchung aus europäischer und deutscher Sicht unter Einbeziehung des Technologieprogramms „Trusted Cloud“. Berlin, Germany 2012

Best, Mohay, Anderson (2004)

Peter J. Best, George Mohay, Alison Anderson: Machine-independent audit trail analysis-a tool for continuous audit assurance. In: Intelligent Systems in Accounting, Finance and Management. Iss. 2, Vol. 12, 2004, p. 85-102

Bezzi, Kaluvuri, Sabetta (2011)

Michele Bezzi, Samuel Kaluvuri, Antonino Sabetta: Ensuring trust in service consumption through security certification. In: ACM (Ed.): Proceedings of the International Workshop on Quality Assurance for Service-Based Applications (QASBA 2011), 14 September, Lugano, Schweiz. New York, NY, USA 2011, p. 40-43

Boritz, No (2005)

J. Efrim Boritz, Won G. No: Security in XML-based financial reporting services on the Internet. In: Journal of Accounting and Public Policy. Iss. 1, Vol. 24, 2005, p. 11-35

Braun, Davis (2003)

Robert L. Braun, Harold E. Davis: Computer-assisted audit tools and techniques: analysis and perspectives. In: Managerial Auditing Journal. Iss. 9, Vol. 18, 2003, p. 725-731

Brown, Wong, Baldwin (2007)

Carol E. Brown, Jeffrey A. Wong, Amelia A. Baldwin: A Review and Analysis of the Existing Research Streams in Continuous Auditing. In: Journal of Emerging Technologies in Accounting. Iss. 1, Vol. 4, 2007, p. 1-28

Bruhn (2008)

Manfred Bruhn: Qualitätsmanagement für Dienstleistungen. Grundlagen, Konzepte, Methoden. 7. ed., Heidelberg 2008

Business Process Technology Group (n.y.)

Business Process Technology Group: The Oryx Project. http://bpt.hpi.uni-potsdam.de/Oryx/WebHome, last retrieval 06.10.2014

CaseWare IDEA Inc. (2008)

CaseWare IDEA Inc.: CONTINUOUS AUDITING: A STRATEGIC APPROACH TO IMPLEMENTATION. Toronto, Canada 2008

Chan, Vasarhelyi (2011)

David Y. Chan, Miklos A. Vasarhelyi: Innovation and practice of continuous auditing. In: International Journal of Accounting Information Systems. Iss. 2, Vol. 12, 2011, p. 152-160

Chen (2004)

Yining Chen: CONTINUOUS AUDITING USING A STRATEGIC-SYSTEMS APPROACH. In: Internal Auditing. Iss. 3, Vol. 19, 2004, p. 31-36

Chen, Hoang (2011)

Lingfeng Chen, D. Hoang: Novel Data Protection Model in Healthcare Cloud. In: IEEE Computer Society (Ed.): 13th International Conference on High Performance Computing and Communications (HPCC), 2-4 September, Banff, Alberta, Canada. New York, NY, USA 2011, p. 550-555

Chen, Lee (2014)

Henry C. H. Chen, Patrick P. C. Lee: Enabling Data Integrity Protection in Regenerating-Coding-Based Cloud Storage: Theory and Implementation. In: IEEE Trans. Parallel Distrib. Syst. Iss. 2, Vol. 25, 2014, p. 407-416

Chen, Wen (2012)

Wei Chen, Qiaoyan Wen: An architecture for dynamic management and monitoring of virtual machines. In: IEEE Computer Society (Ed.): 2nd International Conference on Cloud Computing and Intelligent Systems (CCIS), 30 October - 1 November, Hangzhou, China. New York, NY, USA 2012, p. 444-448

Chieu et al. (2012)

T.C Chieu, M. Singh, Chunqiang Tang, M. Viswanathan, A. Gupta: Automation System for Validation of Configuration and Security Compliance in Managed Cloud Services. In: IEEE Computer Society (Ed.): Ninth International Conference on e-Business Engineering (ICEBE), Sep 9-11, Hangzhou, China. New York, NY, USA 2012, p. 285-291

Chou, Du, Lai (2007)

Charles L.-y. Chou, Timon Du, Vincent S. Lai: Continuous auditing with a multi-agent system. In: Decision Support Systems. Iss. 4, Vol. 42, 2007, p. 2274-2292

CICA/AICPA (1999)

CICA/AICPA: Continuous auditing. Research Report. The Canadian Institute of Chartered, Toronto, Canada 1999

Cimato et al. (2013)

Stelvio Cimato, Ernesto Damiani, Renato Menicocci, Francesco Zavatarelli: Towards the certification of cloud services. In: IEEE Computer Society (Ed.): Proceedings of the 2013 IEEE Ninth World Congress on Services (SERVICES 2013), 28 June - 3 July, Santa Clara, California, USA. Washington, DC, USA 2013, p. 100-105

Clayman et al. (2011)

S. Clayman, R. Clegg, L. Mamatas, G. Pavlou, A. Galis: Monitoring, aggregation and filtering for efficient management of virtual networks. In: IEEE Computer Society (Ed.): 7th International Conference on Network and Service Management (CNSM), 24-28 October, Paris, France. New York, NY, USA 2011, p. 1-7

Clayman, Galis, Mamatas (2010)

S. Clayman, A. Galis, L. Mamatas: Monitoring virtual networks with Lattice. In: IEEE Computer Society (Ed.): IEEE/IFIP Network Operations and Management Symposium Workshops (NOMS Wksps), 19-23 April, Osaka, Japan. New York, NY, USA 2010, p. 239-246

Cloud Security Alliance (n.y.)

Cloud Security Alliance: CSA Security, Trust & Assurance Registry (STAR). https://cloudsecurityalliance.org/star/, last retrieval 08.09.2014

Comuzzi, Spanoudakis (2010)

Marco Comuzzi, George Spanoudakis: Dynamic Set-up of Monitoring Infrastructures for Service Based Systems. In: ACM (Ed.): Proceedings of the 2010 ACM Symposium on Applied Computing, 22-26 March, Sierre, Switzerland. New York, NY, USA 2010, p. 2414-2421

Curbera et al. (2008)

Francisco Curbera, Yurdaer Doganata, Axel Martens, Nirmal K. Mukhi, Aleksander Slominski: Business Provenance – A Technology to Increase Traceability of End-to-End Operations. In: Robert Meersman, Zahir Tari (Ed.): On the Move to Meaningful Internet Systems (OTM 2008) 2008, p. 100-119

Dalziel (2013)

Henry Dalziel: Our 2013 recommended penetration testing tools. http://www.concise-courses.com/security/top-ten-pentesting-tools/, last retrieval 05.10.2014

David, Steinbart (1999)

J. S. David, P. J. Steinbart: Drowning in Data: HOW YOU CAN TURN OCEANS OF DATA INTO USEFUL INFORMATION. In: Strategic Finance. Iss. 6, Vol. 81, 1999, p. 30-36

Doelitzscher et al. (2013)

Frank Doelitzscher, Christoph Reich, Martin Knahl, Nathan Clarke: Understanding Cloud Audits. In: Siani Pearson, George Yee (Ed.): Privacy and Security for Cloud Computing 2013, p. 125-163

Doelitzscher et al. (2012a)

F. Doelitzscher, C. Fischer, D. Moskal, C. Reich, M. Knahl, N. Clarke: Validating Cloud Infrastructure Changes by Cloud Audits. In: IEEE Computer Society (Ed.): IEEE Eighth World Congress on Services (SERVICES), 24-29 June, Honolulu, HI, USA. New York, NY, USA 2012, p. 377-384

Doelitzscher et al. (2012b)

Frank Doelitzscher, Christoph Reich, Martin Knahl, Alexander Passfall, Nathan

Clarke: An agent based business aware incident detection system for cloud environments. In: Journal of Cloud Computing: Advances, Systems and Applications. Iss. 1, Vol. 1, 2012, p. 1-9

Doganata, Curbera (2009)

Yurdaer Doganata, Francisco Curbera: Effect of Using Automated Auditing Tools on Detecting Compliance Failures in Unmanaged Processes. In: Umeshwar Dayal, Johann Eder, Jana Koehler, HajoA Reijers (Ed.): Business Process Management 2009, p. 310-326

Du, Li, Wei (2005)

Timon C. Du, Eldon Y. Li, Eric Wei: Mobile agents for a brokering service in the electronic marketplace. In: Decision Support Systems. Iss. 3, Vol. 39, 2005, p. 371-383

Du, Roohani (2007)

Hui Du, Saeed Roohani: Meeting Challenges and Expectations of Continuous Auditing in the Context of Independent Audits of Financial Statements. In: International Journal of Auditing. Iss. 2, Vol. 11, 2007, p. 133-146

Emeakaroha et al. (2012)

Vincent C. Emeakaroha, Marco A.S. Netto, Rodrigo N. Calheiros, Ivona Brandic, Rajkumar Buyya, César A.F. De Rose: Towards autonomic detection of SLA violations in Cloud infrastructures. In: Future Generation Computer Systems. Iss. 7, Vol. 28, 2012, p. 1017-1029

EuroCloud Europe (n.y.)

EuroCloud Europe: EuroCloud Star Audit (ECSA). http://eurocloud-staraudit.eu/, last retrieval 08.09.2014

European Network and Information Security Agency (2013)

European Network and Information Security Agency: Incident Reporting for Cloud Computing. 2013

European Network and Information Security Agency (2009)

European Network and Information Security Agency: Cloud Computing. Benefits, risks and recommendations for information security. 2009

Fang et al. (2006)

Chen-Liang Fang, Deron Liang, Fengyi Lin, Chien-Cheng Lin, W.C.-C. Chu: A Portable Interceptor Mechanism on SOAP for Continuous Audit. In: IEEE Computer Society (Ed.): 13th Asia Pacific Software Engineering Conference (APSEC 2006), 6-8 December, Bangalore, India. New York, NY, USA 2006, p. 95-104

Fatema et al. (2014)

Kaniz Fatema, Vincent C. Emeakaroha, Philip D. Healy, John P. Morrison, Theo Lynn: A survey of Cloud monitoring tools: Taxonomy, capabilities and objectives. In: Journal of Parallel and Distributed Computing. Iss. 10, Vol. 74, 2014, p. 2918-2933

Federal Office for Information Security (2011)

Federal Office for Information Security: Security Recommendations for Cloud Computing Providers. (Minimum information security requirements). Bonn, Germany 2011

FireEye Inc. (n.y.)

FireEye Inc.: FireEye. http://www.fireeye.com/, last retrieval 28.10.2014

Flowerday, Blundell, Von Solms (2006)

S. Flowerday, A. W. Blundell, R. Von Solms: Continuous auditing technologies and models: A discussion. In: Computers & Security. Iss. 5, Vol. 25, 2006, p. 325

Forum of Incident Response and Security Teams (n.y.)

Forum of Incident Response and Security Teams: Common Vulnerability Scoring System. http://www.first.org/cvss, last retrieval 10.10.2014

Fu et al. (2009)

Qiang Fu, Jian-Guang Lou, Yi Wang, Jiang Li: Execution Anomaly Detection in Distributed Systems through Unstructured Log Analysis. In: IEEE Computer Society (Ed.): Ninth IEEE International Conference on Data Mining (ICDM ’09), 6-9 December, Miami, Florida, USA. New York, NY, USA 2009, p. 149-158

Fuggetta, Picco, Vigna (1998)

A. Fuggetta, G.P Picco, Giovanni Vigna: Understanding code mobility. In: IEEE Transactions on Software Engineering. Iss. 5, Vol. 24, 1998, p. 342-361

Gao (2010)

Jinping Gao: Technical Framework Model of Continuous Online Assurance. In: IEEE Computer Society (Ed.): 2010 International Conference on E-Business and E-Government (ICEE), 7-9 May, Guangzhou, China. New York, NY, USA 2010, p. 2141-2144

Ghulam, Shaikh, Shaikh (2008)

Ali Ghulam, N. Shaikh, Z. Shaikh: Towards an automated multiagent system to monitor user activities against insider threat. In: IEEE Computer Society (Ed.): International Symposium on Biometrics and Security Technologies (ISBAST), 23-24 April, Islamabad, Pakistan. New York, NY, USA 2008, p. 1-5

Giblin, Mueller, Pfitzmann (2006)

Christopher J. Giblin, Samuel Mueller, Birgit Pfitzmann: From regulatory policies to event monitoring rules: Towards model-driven compliance automation. 2006

Goel, Kumar, Shyamasundar (2011)

N. Goel, N.V.N Kumar, R. Shyamasundar: SLA Monitor: A System for Dynamic Monitoring of Adaptive Web Services. In: IEEE Computer Society (Ed.): Ninth IEEE European Conference on Web Services (ECOWS), 14-16 September, Lugano, Switzerland. New York, NY, USA 2011, p. 109-116

Gonzalez, Munoz, Mana (2011)

J. Gonzalez, A. Munoz, A. Mana: Multi-layer Monitoring for Cloud Computing. In: IEEE Computer Society (Ed.): IEEE 13th International Symposium on High-Assurance Systems Engineering (HASE), 10-12 November, Boca Raton, FL, USA. Washington and DC and USA 2011, p. 291-298

Groomer, Murthy (1989)

S. M. Groomer, Uday S. Murthy: Continuous Auditing of Database Applications: An Embedded Audit Module Approach. In: Journal of Information Systems. Iss. 2, Vol. 3, 1989, p. 53

Hardy (2011)

Catherine Hardy: Exploring Continuous Assurance In Practice: Preliminary Insights. In: AISEL (Ed.): Proceedings of 15th Pacific Asia Conference on Information Systems (PACIS 2011), 7-11 July, Brisbane, Australia 2011, p. 1-15

Hasan, Stiller (2005)

Hasan, Burkhard Stiller: A Generic Model and Architecture for Automated Auditing. In: ACM (Ed.): Proceedings of the 16th IFIP/IEEE Ambient Networks International Conference on Distributed Systems: Operations and Management. Berlin, Heidelberg 2005, p. 121-132

Hasselmeyer, d'Heureuse (2010)

P. Hasselmeyer, N. d'Heureuse: Towards holistic multi-tenant monitoring for virtual data centers. In: IEEE Computer Society (Ed.): IEEE/IFIP Network Operations and Management Symposium Workshops (NOMS Wksps), 19-23 April, Osaka, Japan. New York, NY, USA 2010, p. 350-356

He et al. (2013)

Kai He, Chuanhe Huang, Jinhai Wang, Hao Zhou, Xi Chen, Yilong Lu, Lianzhen Zhang, Bin Wang: An Efficient Public Batch Auditing Protocol for Data Security in Multi-cloud Storage. In: 8th ChinaGrid Annual Conference (ChinaGrid), 22-23 August, Changchun, China 2013, p. 51-56

Heiser, Nicolett (2008)

J. Heiser, M. Nicolett: Assessing the security risks of cloud computing. http://www.globalcloudbusiness.com/SharedFiles/Download.aspx?pageid=138&mid=220&fileid=12, last retrieval 09.09.2014

Hermanson et al. (2006)

D. Hermanson, B. Moran, C. Rossie, D. Wolfe: Continuous monitoring of transactions to reduce fraud, misuse, and errors. In: R. T. Edwards (Ed.): Journal of Forensic Accounting: Auditing, Fraud & Taxation 2006, p. 17-30

Hunton, Rose (2010)

James E. Hunton, Jacob M. Rose: 21st Century Auditing: Advancing Decision Support Systems to Achieve Continuous Auditing. In: Accounting Horizons. Iss. 2, Vol. 24, 2010, p. 297-312

HYPERIC (n.y.)

HYPERIC: Hyperic SIGAR API. http://www.hyperic.com/products/sigar, last retrieval 23.10.2014

International Organization for Standardization

International Organization for Standardization, Conformity assessment -- Vocabulary and general principles. ISO/IEC: 17000:2004

International Organization for Standardization

International Organization for Standardization, Information technology - Security techniques - Guidelines on information security controls for the use of cloud computing services based on ISO/IEC 27002. ISO/IEC: 27017

ITIL (n.y.)

ITIL: What is ITIL®? http://www.itil-officialsite.com/AboutITIL/Whatis-ITIL.aspx., last retrieval 28.10.2014

Jans, Alles, Vasarhelyi (2013)

Mieke Jans, Michael Alles, Miklos Vasarhelyi: The case for process mining in auditing: Sources of value added and areas of application. In: Methodologies in AIS Research. Iss. 1, Vol. 14, 2013, p. 1-20

Jiang et al. (2008)

Zhen Jiang, A.E Hassan, P. Flora, G. Hamann: Abstracting Execution Logs to Execution Events for Enterprise Applications. In: IEEE Computer Society (Ed.): The Eighth International Conference on Quality Software (QSIC '08), 12-13 August, Oxford, England. New York, NY, USA 2008, p. 181-186

Kaliski, Pauley (2010)

Jr. Burton S. Kaliski, Wayne Pauley: Toward Risk Assessment As a Service in Cloud Environments. In: ACM (Ed.): Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing, 22-25 June, Boston, MA, USA. Berkeley, CA, USA 2010, p. 1-7

Kalloniatis, Mouratidis, Islam (2013)

Christos Kalloniatis, Haralambos Mouratidis, Shareeful Islam: Evaluating cloud deployment scenarios based on security and privacy requirements. In: Requirements Engineering. Iss. 4, Vol. 18, 2013, p. 299-319

Katsaros, Kübert, Gallizo (2011)

G. Katsaros, R. Kübert, G. Gallizo: Building a Service-Oriented Monitoring Framework with REST and Nagios. In: IEEE Computer Society (Ed.): IEEE International Conference on Services Computing (SCC), 4-9 July, Washington, DC, USA. New York, NY, USA 2011, p. 426-431

Kaufman (2009)

Lori M. Kaufman: Data Security in the World of Cloud Computing. In: IEEE Security and Privacy. Iss. 4, Vol. 7, 2009, p. 61-64

Khan, Malluhi (2010)

K.M Khan, Q. Malluhi: Establishing Trust in Cloud Computing. In: IT Professional. Iss. 5, Vol. 12, 2010, p. 20-27

Kim, Kim, Eom (2010)

Junghan Kim, Inhyuk Kim, Young Eom: NOPFIT: File System Integrity Tool for Virtual Machine Using Multi-byte NOP Injection. In: IEEE Computer Society (Ed.): International Conference on Computational Science and Its Applications (ICCSA), 23-26 March, Fukuoka, Japan. New York, NY, USA 2010, p. 335-338

Kleinmuntz (1990)

B. Kleinmuntz: Why we still use our heads instead of formulas: toward an integrative approach. In: Psychological bulletin. Iss. 3, Vol. 107, 1990, p. 296-310

Ko et al. (2011)

R.K.L Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Qianhui Liang, Bu Sung Lee: TrustCloud: A Framework for Accountability and Trust in Cloud Computing. In: IEEE Computer Society (Ed.): IEEE World Congress on Services (SERVICES), 4-9 July, Washington, DC, USA. New York, NY, USA 2011, p. 584-588

Ko, Jagadpramana, Lee (2011)

R.K.L. Ko, P. Jagadpramana, Bu-Sung Lee: Flogger: A File-Centric Logger for Monitoring File Access and Transfers within Cloud Computing Environments. In: IEEE Computer Society (Ed.): IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 16-18 November, Changsha, China. New York, NY, USA 2011, p. 765-771

Koschorreck (2011)

G. Koschorreck: Automated Audit of Compliance and Security Controls. In: IEEE Computer Society (Ed.): Sixth International Conference on IT Security Incident Management and IT Forensics (IMF), 10-12 May, Stuttgart, Germany. Washington, DC, USA 2011, p. 137-148

Kuhn Jr., Sutton (2010)

John R Kuhn Jr., Steve G. Sutton: Continuous Auditing in ERP System Environments: The Current State and Future Directions. In: Journal of Information Systems. Iss. 1, Vol. 24, 2010, p. 91-112

Kunz, Niehues, Waldmann (2013)

Thomas Kunz, Peter Niehues, Ulrich Waldmann: Technische Unterstützung von Audits bei Cloud-Betreibern. In: Datenschutz und Datensicherheit - DuD. Iss. 8, Vol. 37, 2013, p. 521-525

Kuo et al. (2011)

Chien-Ting Kuo, He-Ming Ruan, Chin-Laung Lei, Shih-Jen Chen: A Mechanism on Risk Analysis of Information Security with Dynamic Assessment. In: IEEE Computer Society (Ed.): Third International Conference on Intelligent Networking and Collaborative Systems (INCoS 2011), 30 November - 2 December, Fukuoka, Japan. New York, NY, USA 2011, p. 643-646

Kutare et al. (2010)

Mahendra Kutare, Greg Eisenhauer, Chengwei Wang, Karsten Schwan, Vanish Talwar, Matthew Wolf: Monalytics: Online Monitoring and Analytics for Managing Large Scale Data Centers. In: ACM (Ed.): Proceedings of the 7th International Conference on Autonomic Computing, 7-11 June, Washington, DC, USA. New York, NY, USA 2010, p. 141-150

Kwon et al. (2014)

Ohmin Kwon, Dongyoung Koo, Yongjoo Shin, Hyunsoo Yoon: A Secure and Efficient Audit Mechanism for Dynamic Shared Data in Cloud Storage. In: The Scientific World Journal. 2014, p. 1-11

Lamparter, Luckner, Mutschler (2007)

S. Lamparter, S. Luckner, S. Mutschler: Formal Specification of Web Service Contracts for Automated Contracting and Monitoring. In: IEEE Computer Society (Ed.): 40th Annual Hawaii International Conference on System Sciences (HICSS 2007), 3-6 January, Waikoloa, Hawaii. New York, NY, USA 2007, p. 1-10

Li, Huang, Lin (2007)

Shing-Han Li, Shi-Ming Huang, Yuah-Chiao G. Lin: DEVELOPING A CONTINUOUS AUDITING ASSISTANCE SYSTEM BASED ON INFORMATION PROCESS MODELS. In: Journal of Computer Information Systems. Iss. 1, Vol. 48, 2007, p. 2-13

Lin, Lin, Liang (2010)

Chien-Cheng Lin, Fengyi Lin, Deron Liang: An Analysis of Using State of the Art Technologies to Implement Real-Time Continuous Assurance. In: IEEE Computer Society (Ed.): 6th World Congress on Services (SERVICES-1), 5-10 July, Miami, FL, USA. New York, NY, USA 2010, p. 415-422

Lins (2014)

Sebastian Lins: Methoden und Modelle zur kontinuierlichen Auditierung von Cloud-Services. Köln, Germany 2014

Liu et al. (2014)

Chang Liu, Rajiv Ranjan, Chi Yang, Xuyun Zhang, Lizhe Wang, Jinjun Chen: MuR-DPA: Top-down Levelled Multi-replica Merkle Hash Tree Based Secure Public Auditing for Dynamic Big Data Storage on Cloud. In: IACR Cryptology ePrint Archive. 2014, p. 1-12

Liu et al. (2013a)

C. Liu, J. Chen, L. Yang, X. Zhang, C. Yang, R. Ranjan, K. Ramamohanarao: Authorized Public Auditing of Dynamic Big Data Storage on Cloud with Efficient Verifiable Fine-grained Updates. In: IEEE Transactions on Parallel and Distributed Systems. (in Press), 2013, p. 1-11

Liu et al. (2013b)

Chang Liu, R. Ranjan, Xuyun Zhang, Chi Yang, D. Georgakopoulos, Jinjun Chen: Public Auditing for Big Data Storage in Cloud Computing - A Survey. In: IEEE Computer Society (Ed.): 16th International Conference on Computational Science and Engineering (CSE), 3-5 December, Sydney, NSW, Australia. New York, NY, USA 2013, p. 1128-1135

Liu et al. (2010)

Qian Liu, Chuliang Weng, Minglu Li, Yuan Luo: An In-VM Measuring Framework for Increasing Virtual Machine Security in Clouds. In: IEEE Security & Privacy. Iss. 6, Vol. 8, 2010, p. 56-62

Lungu, Vătuiu (2007)

Ion Lungu, Teodora Vătuiu: COMPUTER ASSISTED AUDIT TECHNIQUES. In: Annals of the University of Petrosani Economics. Vol. 7, 2007, p. 217-224

Lunt (1993)

Teresa F. Lunt: A survey of intrusion detection techniques. In: Computers & Security. Iss. 4, Vol. 12, 1993, p. 405-418

Lyon (n.y.)

Gordon Lyon: NMap. http://nmap.org/, last retrieval 20.10.2014

Mahzan, Lymer (2014)

Nurmazilah Mahzan, Andy Lymer: Examining the adoption of computer-assisted audit tools and techniques. In: Managerial Auditing Journal. Iss. 4, Vol. 29, 2014, p. 327-349

Mana, Munoz, Gonzalez (2011)

A. Mana, A. Munoz, J. Gonzalez: Dynamic security monitoring for Virtualized Environments in Cloud computing. In: IEEE Computer Society (Ed.): 1st International Workshop on Securing Services on the Cloud (IWSSC), 6-8 September, Milan, Italy 2011, p. 1-6

Manson, McCartney, Sherer (2001)

Stuart Manson, Sean McCartney, Michael Sherer: Audit automation as control within audit firms. In: Accounting, Auditing & Accountability Journal. Iss. 1, Vol. 14, 2001, p. 109-130

Marques, Santos, Santos (2013)

Rui Marques, Henrique Santos, Carlos Santos: A Conceptual Model for Evaluating Systems with Continuous Assurance Services. In: "Petru Maior" University Press (Ed.): The 7th International Conference Interdisciplinarity in Engineering (INTER-ENG 2013), 10-11 October, Petru Maior University of Tirgu Mures, Romania 2013, p. 304-309

Massonet et al. (2011)

P. Massonet, S. Naqvi, C. Ponsard, J. Latanicki, B. Rochwerger, M. Villari: A Monitoring and Audit Logging Architecture for Data Location Compliance in Federated Cloud Infrastructures. In: IEEE Computer Society (Ed.): IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 16-20 May, Shanghai, China. New York, NY, USA 2011, p. 1510-1517

Mell et al. (2012)

Peter Mell, David Waltermire, Larry Feldman, Harold Booth, Alfred Ouyang, Zach Ragland, Timothy McBride: CAESARS Framework Extension: An Enterprise Continuous Monitoring Technical Reference Architecture (Second Draft). Gaithersburg, MD, U.S. 2012

Mell, Grance (2011)

Peter Mell, Timothy Grance: The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology. Gaithersburg, Montgomery, USA 2011

Microsoft (a) (n.y.)

Microsoft (a): Home Page: Spy++. http://msdn.microsoft.com/en-us/library/aa264396%28v=vs.60%29.aspx, last retrieval 28.10.2014

Microsoft (b) (n.y.)

Microsoft (b): Hooks Overview. http://msdn.microsoft.com/en-us/library/windows/desktop/ms644959%28v=vs.85%29.aspx, last retrieval 28.10.2014

Microsoft (c) (n.y.)

Microsoft (c): Microsoft Network Monitor 3.4. http://www.microsoft.com/en-us/download/details.aspx?id=4865, last retrieval 28.10.2014

Modi et al. (2013)

Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel, Muttukrishnan Rajarajan: A survey of intrusion detection techniques in Cloud. In: Journal of Network and Computer Applications. Iss. 1, Vol. 36, 2013, p. 42-57

Montes et al. (2013)

Jesús Montes, Alberto Sánchez, Bunjamin Memishi, María S. Pérez, Gabriel Antoniu: GMonE: A complete approach to cloud monitoring. In: Future Generation Computer Systems. Iss. 8, Vol. 29, 2013, p. 2026-2040

Murthy, Groomer (2004)

Uday S. Murthy, S. M. Groomer: A continuous auditing web services model for XML-based accounting systems. In: International Journal of Accounting Information Systems. Iss. 2, Vol. 5, 2004, p. 139-163

Myers (2013)

Michael Myers: Qualitative research in business & management. 2. ed., London 2013

Nagios Enterprises (n.y.)

Nagios Enterprises: Nagios. http://www.nagios.com/products, last retrieval 23.10.2014

National Institute of Standards and Technology (n.y.)

National Institute of Standards and Technology: Common Configuration Enumeration (CCE). http://nvd.nist.gov/cce/index.cfm, last retrieval 10.10.2014

Ni et al. (2013)

J. Ni, Y. Yu, Y. Mu, Q. Xia: On the Security of an Efficient Dynamic Auditing Protocol in Cloud Storage. In: Parallel and Distributed Systems, IEEE Transactions on. (in Press), 2013, p. 1-3

Nithiavathy (2013)

R. Nithiavathy: Data integrity and data dynamics with secure storage service in cloud. In: IEEE Computer Society (Ed.): International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME), 21-22 February, Salem, Germany. New York, NY, USA 2013, p. 125-130

Nowak, Bagrij (2007)

K. Nowak, L. Bagrij: Using Distributed Multilevel Agent-based Monitoring Technique for Automated Network Modelling Approach. In: IEEE Computer Society (Ed.): 2nd International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX ’07), 14-16 June, Szklarska, Poland. New York, NY, USA 2007, p. 61-72

OpenNebula Project (n.y.)

OpenNebula Project: OpenNebula. http://opennebula.org/, last retrieval 23.10.2014

OWASP Foundation Inc. (n.y.)

OWASP Foundation Inc.: Open Web Application Security Project (OWASP). https://www.owasp.org/, last retrieval 23.10.2014

Paraleap Technologies (n.y.)

Paraleap Technologies: About AzureWatch. https://www.paraleap.com/AzureWatch, last retrieval 23.10.2014

Pedrosa, Costa (2014)

Isabel Pedrosa, Carlos Costa: New Trends on CAATTs: What Are the Chartered Accountants' New Challenges? In: ACM (Ed.): Proceedings of the International Conference on Information Systems and Design of Communication, 16-17 May, Lisboa, Portugal. New York, NY, USA 2014, p. 138-142

Perols, Murthy (2012)

Johan L. Perols, Uday S. Murthy: Information Fusion in Continuous Assurance. In: Journal of Information Systems. Iss. 2, Vol. 26, 2012, p. 35-52

Peterson (1977)

James L. Peterson: Petri Nets. In: ACM Computing Surveys. Iss. 3, Vol. 9, 1977, p. 223-252

Portswigger Web Security (n.y.)

Portswigger Web Security: Burp Suite. http://portswigger.net/burp/, last retrieval 23.10.2014

Postel (1981)

J. Postel: RFC: 791 - Internet Protocol. DARPA Inernet Programm, Protocol Specification. 1981

Povedano-Molina et al. (2013)

Javier Povedano-Molina, Jose M. Lopez-Vega, Juan M. Lopez-Soler, Antonio Corradi, Luca Foschini: DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant Clouds. In: Future Generation Computer Systems. Iss. 8, Vol. 29, 2013, p. 2041-2056

Qualys (n.y.)

Inc. Qualys: Qualys. https://www.qualys.com/, last retrieval 20.10.2014

Rajkumar, Kumar, Sivaramakrishnan (2013)

M. Rajkumar, V. Kumar, R. Sivaramakrishnan: Efficient Integrity Auditing Services for Cloud Computing Using Raptor Codes. In: ACM (Ed.): Proceedings of the 2013 Research in Adaptive and Convergent Systems, 1-4 October, Montreal, Canada. New York, NY, USA 2013, p. 75-78

Rannenberg (2000)

Kai Rannenberg: IT Security Certification and Criteria. In: Sihan Qing, Jan H.P Eloff (Ed.): Information Security for Global Information Infrastructures 2000, p. 1-10

Rezaee et al. (2002)

Zabihollah Rezaee, Ahmad Sharbatoghlie, Rick Elam, Peter L. McMickle: Continuous auditing: Building automated auditing capability. In: Auditing. Iss. 1, Vol. 21, 2002, p. 147-163

Romano et al. (2011)

L. Romano, D. de Mari, Z. Jerzak, C. Fetzer: A Novel Approach to QoS Monitoring in the Cloud. In: IEEE Computer Society (Ed.): First International Conference on Data Compression, Communications and Processing (CCP), 21-24 June, Palinuro, Italy. New York, NY, USA 2011, p. 45-51

Sackmann et al. (2008)

S. Sackmann, M. Kahmer, M. Gilliot, L. Lowis: A Classification Model for Automating Compliance. In: IEEE Computer Society (Ed.): 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 21-24 July, Washington, DC, USA. New York, NY, USA 2008, p. 79-86

Sackmann, Kähmer (2008)

Stefan Sackmann, Martin Kähmer: ExPDT: Ein Policy-basierter Ansatz zur Automatisierung von Compliance. In: WIRTSCHAFTSINFORMATIK. Iss. 5, Vol. 50, 2008, p. 366-374

Saha (2008)

Dipankar Saha: A Hitchhiker's Guide to Galaxy a.k.a Netweaver BPM - Part 1. http://scn.sap.com/people/dipankar.saha3/blog/2008/09/08/a-hitchhikers-guide-to-galaxy-aka-netweaver-bpm--part-1, last retrieval 06.10.2014

Schneider et al. (2014)

S. Schneider, J. Lansing, F. Gao, A. Sunyaev: A Taxonomic Perspective on Certification Schemes: Development of a Taxonomy for Cloud Service Certification Criteria. In: IEEE Computer Society (Ed.): Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS 2014), January 6-9, Big Island, Hawaii, USA. New York, NY, USA 2014, p. 1-10

Schneider, Lansing, Sunyaev (2013)

S. Schneider, J. Lansing, A. Sunyaev: Empfehlungen zur Gestaltung von Cloud-Service-Zertifizierungen. In: Industrie Management. Iss. 4, Vol. 29, 2013, p. 13-17

Schroeder (1995)

B.A Schroeder: On-line monitoring: a tutorial. In: Computer. Iss. 6, Vol. 28, 1995, p. 72-78

Shah, Swaminathan, Baker (2008)

Mehul A. Shah, Ram Swaminathan, Mary Baker: Privacy-preserving audit and extraction of digital contents. 2008

Shahriar, Zulkernine (2011)

Hossain Shahriar, Mohammad Zulkernine: Taxonomy and classification of automatic monitoring of program security vulnerability exploitations. In: Journal of Systems & Software. Iss. 2, Vol. 84, 2011, p. 250-269

Shaikh (2005)

Junaid M. Shaikh: E-commerce impact: emerging technology - electronic auditing. In: Managerial Auditing Journal. Iss. 4, Vol. 20, 2005, p. 408-421

Shao et al. (2010)

Jin Shao, Hao Wei, Qianxiang Wang, Hong Mei: A Runtime Model Based Monitoring Approach for Cloud. In: IEEE Computer Society (Ed.): IEEE 3rd International Conference on Cloud Computing (CLOUD), 5-10 July, Miami, FL, USA. New York, NY, USA 2010, p. 313-320

Shao, Wang (2011)

Jin Shao, Qianxiang Wang: A Performance Guarantee Approach for Cloud Applications Based on Monitoring. In: IEEE Computer Society (Ed.): IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW), 18-22 July, Munich, Germany. New York, NY, USA 2011, p. 25-30

Sharif et al. (2009)

Monirul Sharif, Wenke Lee, Weidong Cui, Andrea Lanzi: Secure in-VM Monitoring Using Hardware Virtualization. In: ACM (Ed.): Proceedings of the 16th ACM Conference on Computer and Communications Security, 9-13 November, Chicago, IL, USA. New York, NY, USA 2009, p. 477-487

Sheng et al. (2014)

Quan Z. Sheng, Zakaria Maamar, Lina Yao, Claudia Szabo, Scott Bourne: Behavior modeling and automated verification of Web services. In: Information Sciences. Vol. 258, 2014, p. 416-433

Singh et al. (2013)

Kishore Singh, Peter J. Best, Mario Bojilov, Catherine Blunt: Continuous Auditing and Continuous Monitoring in ERP Environments: Case Studies of Application Implementations. In: Journal of Information Systems. Iss. 1, Vol. 28, 2013, p. 287-310

Singleton, Flesher (2003)

Tommie Singleton, Dale L. Flesher: A 25-year retrospective on the IIA's SAC projects. In: Managerial Auditing Journal. Iss. 1, Vol. 18, 2003, p. 39-53

Subashini, Kavitha (2011)

S. Subashini, V. Kavitha: A survey on security issues in service delivery models of cloud computing. In: Journal of Network and Computer Applications. Iss. 1, Vol. 34, 2011, p. 1-11

Sujana, Revathi (2012)

J.A.J. Sujana, T. Revathi: Ensuring Privacy in Data Storage as a Service for Educational Institution in Cloud Computing. In: IEEE Computer Society (Ed.): International Symposium on Cloud and Services Computing (ISCOS), 17-18 December, Mangalore, India. New York, NY, USA 2012, p. 96-100

Sunyaev, Schneider (2013)

Ali Sunyaev, Stephan Schneider: Cloud services certification. In: Communications of the ACM. Iss. 2, Vol. 56, 2013, p. 33-36

Tenable network security (n.y.)

Tenable network security: Nessus. nessus.org, last retrieval 05.10.2014

The Apache Software Foundation (n.y.)

The Apache Software Foundation: Package org.apache.axis.handlers. https://axis.apache.org/axis/java/apiDocs/org/apache/axis/handlers/package-summary.html, last retrieval 28.10.2014

The MITRE Corporation (n.y.)

The MITRE Corporation: Common Vulnerabilities and Exposure List. https://cve.mitre.org/, last retrieval 10.10.2014

Thiebes (2014)

Scott Thiebes: Methoden und Modelle zur kontinuierlichen Auditierung von Cloud-Services. Köln, Germany 2014

Tovarnak, Pitner (2012)

D. Tovarnak, T. Pitner: Towards multi-tenant and interoperable monitoring of virtual machines in cloud. In: IEEE Computer Society (Ed.): 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 26-29 September, Timisoara, Romania. New York, NY, USA 2012, p. 436-442

TÜV Rheinland (n.y.)

TÜV Rheinland: Cloud Security Certification. http://www.tuv.com/en/corporate/business_customers/information_security_cw/strategic_information_security/cloud_security_certification/cloud_security_certification.html, last retrieval 08.09.2014

United States Computer Emergency Readiness Team (2014)

United States Computer Emergency Readiness Team: OpenSSL 'Heartbleed' vulnerability (CVE-2014-0160). https://www.us-cert.gov/ncas/alerts/TA14-098A, last retrieval 28.10.2014

University of Chicago (n.y.)

University of Chicago: Nimbus. http://www.nimbusproject.org/, last retrieval 23.10.2014

van der Aalst, de Medeiros (2005)

W.M.P. van der Aalst, A.K.A. de Medeiros: Process Mining and Security: Detecting Anomalous Process Executions and Checking Process Conformance. In: Electronic Notes in Theoretical Computer Science. Vol. 121, 2005, p. 3-21

Vasarhelyi et al. (2012)

Miklos A. Vasarhelyi, Michael Alles, Siripan Kuenkaikaew, James Littley: The acceptance and adoption of continuous auditing by internal auditors: A micro analysis. In: Methodologies in AIS Research. Iss. 3, Vol. 13, 2012, p. 267-281

Vasarhelyi et al. (2004)

Miklos A. Vasarhelyi, Michael G. Alles, Alexander Kogan, Dan O'Leary: Principles of Analytic Monitoring for Continuous Assurance. In: Journal of Emerging Technologies in Accounting. Vol. 1, 2004, p. 1-21

Vasarhelyi, Alles, Williams (2010)

Miklos Vasarhelyi, Michael Alles, Katie Williams: Continuous assurance for the now economy. Sydney 2010

Vasarhelyi, Halper (1991)

Miklos A. Vasarhelyi, Fern B. Halper: The Continuous Audit of Online Systems. In: Auditing. Iss. 1, Vol. 10, 1991, p. 110-125

Wang et al. (2013)

Cong Wang, Chow, Sherman S. M., Qian Wang, Kui Ren, Wenjing Lou: Privacy-Preserving Public Auditing for Secure Cloud Storage. In: IEEE Transactions on Computers. Iss. 2, Vol. 62, 2013, p. 362-375

Wang et al. (2011)

Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, Jin Li: Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing. In: IEEE Transactions on Parallel & Distributed Systems. Iss. 5, Vol. 22, 2011, p. 847-859

Wang et al. (2009)

Cong Wang, Qian Wang, Kui Ren, Wenjing Lou: Ensuring data storage security in Cloud Computing. In: IEEE Computer Society (Ed.): 17th International Workshop on Quality of Service (IWQoS), 13-15 July, Charleston, SC, USA. New York, NY, USA 2009, p. 1-9

Wang, Li, Li (2014)

Boyang Wang, Baochun Li, Hui Li: Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud. In: IEEE Transactions on Cloud Computing. Iss. 1, Vol. 2, 2014, p. 43-56

Wang, Li, Li (2013a)

Boyang Wang, Baochun Li, Hui Li: Panda: Public auditing for shared data with efficient user revocation in the cloud. In: IEEE Computer Society (Ed.): Proceedings of IEEE INFOCOM 2013 Conference, 14-19 April, Turin, Italy. New York, NY, USA 2013, p. 2904-2912

Wang, Li, Li (2013b)

Boyang Wang, Hui Li, Ming Li: Privacy-preserving public auditing for shared cloud data supporting group dynamics. In: IEEE Computer Society (Ed.): IEEE International Conference on Communications (ICC), 9-13 June, Budapest, Hungary. New York, NY, USA 2013, p. 1946-1950

Wang, Li, Li (2012)

Boyang Wang, Baochun Li, Hui Li: Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud. In: Feng Bao, Pierangela Samarati, Jianying Zhou (Ed.): Applied Cryptography and Network Security 2012, p. 507-525

Wang, Mao, Luo (2012)

Yao Wang, Yaqiang Mao, Yuan Luo: An In-Out-VM measurement architecture against dynamic attacks in clouds. In: IEEE Computer Society (Ed.): IEEE 14th International Conference on Communication Technology (ICCT), 9-11 November, Chengdu, China. New York, NY, USA 2012, p. 761-767

Webster, Watson (2002)

Jane Webster, Richard T. Watson: Analyzing the Past to Prepare for the Future: Writing a Literature Review. In: MIS Q. Iss. 2, Vol. 26, 2002, p. xiii-xxiii

Wen et al. (2009)

Lijie Wen, Jianmin Wang, Wil M. Aalst, Biqing Huang, Jiaguang Sun: A Novel Approach for Process Mining Based on Event Types. In: Journal of Intelligent Information Systems. Iss. 2, Vol. 32, 2009, p. 163-190

Windhorst, Sunyaev (2013)

I. Windhorst, A. Sunyaev: Dynamic Certification of Cloud Services. In: IEEE Computer Society (Ed.): Eighth International Conference on Availability, Reliability and Security (ARES), 2-6 September, Regensburg, Germany. New York, NY, USA 2013, p. 412-417

Woodroof, Searcy (2001)

J. Woodroof, D. Searcy: Continuous audit implications of Internet technology: triggering agents over the Web in the domain of debt covenant compliance. In: IEEE Computer Society (Ed.): Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS), 3-6 January, Outrigger Wailea Resort, Island of Maui. New York, NY, USA 2001, p. 1-8

Wu et al. (2008)

Chien-Ho Wu, Y. Shao, Bih-Yih Ho, Tsair-Yuan Chang: On an agent-based architecture for collaborative continuous auditing. In: IEEE Computer Society (Ed.): 12th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2008), 16-18 April, Xi´an, China. New York, NY, USA 2008, p. 355-360

Wu, Zhao, Ye (2008)

Fang Wu, Zhijin Zhao, Xueyi Ye: A New Dynamic Network Monitoring Based on IA. In: IEEE Computer Society (Ed.): International Symposium on Computer Science and Computational Technology (ISCSCT ’08), 20-22 December, Shanghai, China. New York, NY, USA 2008, p. 637-640

Xiang et al. (2010)

Guofu Xiang, Hai Jin, Deqing Zou, Xinwen Zhang, Sha Wen, Feng Zhao: VMDriver: A Driver-Based Monitoring Mechanism for Virtualization. In: IEEE Computer Society (Ed.): Reliable Distributed Systems, 2010 29th IEEE Symposium on, 31 October - 3 November, New Delhi, India. New York, NY, USA 2010, p. 72-81

Yang, Jia (2013)

Kan Yang, Xiaohua Jia: An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing. In: IEEE Transactions on Parallel & Distributed Systems. Iss. 9, Vol. 24, 2013, p. 1717-1726

Yang, Jia (2012)

Kan Yang, Xiaohua Jia: Data storage auditing service in cloud computing: challenges, methods and opportunities. In: World Wide Web. Iss. 4, Vol. 15, 2012, p. 409-428

Yavuz, Ning (2009)

A. Yavuz, Peng Ning: BAF: An Efficient Publicly Verifiable Secure Audit Logging Scheme for Distributed Systems. In: IEEE Computer Society (Ed.): Annual Computer Security Applications Conference (ACSAC ’09), 7-11 December, Honolulu, HI, Hawaii. New York, NY, USA 2009, p. 219-228

Ye, Yang, Gan (2012)

Huanzhuo Ye, Jingwei Yang, Yanping Gan: Research on Continuous Auditing Based on Multi-agent and Web Services. In: IEEE Computer Society (Ed.): International Conference on Management of e-Commerce and e-Government (ICMeCG), 20-21 October, Beijing, China. New York, NY, USA 2012, p. 220-225

Yeh, Chang, Shen (2008)

Chun-Hsiu Yeh, Tsui-Ping Chang, Wei-Cheng Shen: Developing Continuous Audit and Integrating Information Technology in E-business. In: IEEE Computer Society (Ed.): IEEE Asia-Pacific Services Computing Conference (APSCC ’08), 9-12 December, Yilan, Taiwan. New York, NY, USA 2008, p. 1013-1018

Zachary, McEachen, Ettlich (2004)

J. Zachary, J. McEachen, D. Ettlich: Conversation exchange dynamics for real-time network monitoring and anomaly detection. In: IEEE Computer Society (Ed.): Second IEEE International Information Assurance Workshop, 8-9 April, New York, NY, USA. New York, NY, USA 2004, p. 59-70

Zhang, Wan (2011)

Juan Zhang, Changsheng Wan: Securing continuous auditing in wireless network. In: IEEE Computer Society (Ed.): International Conference on E -Business and E -Government (ICEE), 6-8 May, Shanghai, China. New York, NY, USA 2011, p. 1-4

Zhao, Zhou, Fan (2012)

Ying Zhao, Fangfang Zhou, Xiaoping Fan: A Real-time Visualization Framework for IDS Alerts. In: ACM (Ed.): Proceedings of the 5th International Symposium on Visual Information Communication and Interaction, 27-28 September, Hangzhou, China. New York, NY, USA 2012, p. 11-17

Zhu et al. (2013)

Yan Zhu, Gail-Joon Ahn, Hongxin Hu, S.S Yau, H.G An, Chang-Jun Hu: Dynamic Audit Services for Outsourced Storages in Clouds. In: IEEE Transactions on Services Computing. Iss. 2, Vol. 6, 2013, p. 227-238

Zhu et al. (2012)

Yan Zhu, Hongxin Hu, Gail-Joon Ahn, Mengyang Yu: Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage. In: IEEE Transactions on Parallel and Distributed Systems. Iss. 12, Vol. 23, 2012, p. 2231-2244

Żmuda, Psiuk, Zieliński (2010)

Daniel Żmuda, Marek Psiuk, Krzysztof Zieliński: Dynamic monitoring framework for the SOA execution environment. In: 2014 Conference on Systems Engineering Research. Iss. 1, Vol. 1, 2010, p. 125-133

Appendix

Appendix A – Previous identified Methods

This paper extends previous work regarding the identification of (semi) automated monitoring and auditing methods. Lins (2014) and Thiebes (2014) performed a systematic literature review to identify (semi) automated monitoring and auditing methods. These methods were included in this paper. Previously identified methods and corresponding publications are listed in the following table, in order to differentiate them precisely against new methods identified in this thesis.

Method description	**Source**
System Architecture	
Monitoring and Control Layer	Perols, Murthy (2012), Alles et al. (2006).
Continuous Process Auditing System	Du, Roohani (2007).
XML-based Independent Auditing System	Du, Roohani (2007).
Standard Interfaces via Middleware	Shaikh (2005).
Continuous Auditing Web Services	Murthy, Groomer (2004).
Embedded Audit Modules	Li, Huang, Lin (2007), Rezaee et al. (2002), Perols, Murthy (2012), Alles et al. (2006), Du, Roohani (2007), Chou, Du, Lai (2007), Shaikh (2005).
Audit Data Marts	Rezaee et al. (2002), Chou, Du, Lai (2007).
Automated Audit of Compliance and Security Controls	Koschorreck (2011).
Digital Agents	Woodroof, Searcy (2001).
Intelligent Agents	Shaikh (2005).
Agent-based Continuous Audit Model	Chou, Du, Lai (2007).

Application, Virtualization and Network	
Local Application Surveillance (LAS)	Gonzalez, Munoz, Mana (2011).
Intra Platform Surveillance (IPS)	Gonzalez, Munoz, Mana (2011).
Global Application Surveillance (GAS)	Gonzalez, Munoz, Mana (2011).
Automatic Network Monitoring	Wu, Zhao, Ye (2008).
Framework for Increasing Virtual Machine Security	Liu et al. (2010).
Compliance	
Ontology Based Contracts	Lamparter, Luckner, Mutschler (2007).
SLA-Monitor	Goel, Kumar, Shyamasundar (2011).
Automation Engine	Chieu et al. (2012).
Logging and Inspection	
Abstract Execution Log Inspection	Jiang et al. (2008).
System Layer Logging	Ko et al. (2011).
Data Layer Logging	Ko et al. (2011).
Workflow Layer Logging	Ko et al. (2011).
Application, Virtualization and Network	
Audit Trail Analysis System for Intrusion Detection	Best, Mohay, Anderson (2004).
Intrusion Detection Expert System	Lunt (1993).
Neural Networks for Intrusion Detection	Lunt (1993).
Model-based Reasoning for Intrusion Detection	Lunt (1993).
Continuous Assurance Fusion Architecture	Perols, Murthy (2012).

Penetration Testing	Alles et al. (2006).
Data Integrity	
Multicloud Batch Auditing Protocol	Yang, Jia (2013).
Periodic Sampling Audit	Zhu et al. (2013).
Public Auditability and Data Dynamics Scheme	Wang et al. (2011).
Authorized Auditing Scheme	Liu et al. (2013a).
File System Integrity Tool for Virtual Machine	Kim, Kim, Eom (2010).
Auditing Scheme for Data Integrity	Wang et al. (2013).

Table Appendix A Previous identified methods by Lins (2014) and Thiebes (2014).

Appendix B – Identified Continuous Auditing Methods

Method description	Source
Computer-Assisted Auditing Technologies and Tools	
Generalized Audit Software (GAS)	Chou, Du, Lai (2007), Ahmi, Kent (2012), Lungu, Vătuiu (2007), Singleton, Flesher (2003), Mahzan, Lymer (2014), Pedrosa, Costa (2014), Braun, Davis (2003).
Penetration Testing	[Interviews]
Formal Languages	Gao (2010), Boritz, No (2005), Murthy, Groomer (2004).
Evidence Gathering Mechanisms	
Embedded Audit Module (EAM)	Alles et al. (2006), Chen (2004), Schroeder (1995), Chou, Du, Lai (2007), Groomer, Murthy (1989), Rezaee et al. (2002), Hunton, Rose (2010), and Braun, Davis (2003), Hunton, Rose (2010), Lin, Lin, Liang (2010), Ardagna et al. (2012).
Interceptor	Lin, Lin, Liang (2010), Fang et al. (2006), Żmuda, Psiuk, Zieliński (2010).
Digital Agent (DA)	Du, Li, Wei (2005), Fuggetta, Picco, Vigna (1998), Chou, Du, Lai (2007), Shaikh (2005), Woodroof, Searcy (2001), Ye, Yang, Gan (2012), Doelitzscher et al. (2012b).
Audit Data Mart (ADM)	Singh et al. (2013), Ye, Yang, Gan (2012), Rezaee et al. (2002), Chou, Du, Lai (2007), David, Steinbart (1999), Baksa, Turoff (2011).
Vulnerability Databases	Kuo et al. (2011).
Auditing System Architectures	
Monitoring and Control Layer (MCL)	Alles et al. (2006), Kuhn Jr., Sutton (2010), Vasarhelyi et al. (2004), Perols, Murthy (2012).

Agent-based continuous Auditing Architectures	Chou, Du, Lai (2007), Ye, Yang, Gan (2012), Wu et al. (2008), Doelitzscher et al. (2012b), Zhang, Wan (2011).
Auditing Web Services	Yeh, Chang, Shen (2008), Murthy, Groomer (2004), Gao (2010), Doelitzscher et al. (2013), Doelitzscher et al. (2012a), Doelitzscher et al. (2012b).
Decision Support System (DSS)	Hunton, Rose (2010).
Data Integrity Validation	
Auditing of Data Integrity	Liu et al. (2013a), Wang et al. (2013), Wang et al. (2011), Yang, Jia (2013), Zhu et al. (2013), Sujana, Revathi (2012), Nithiavathy (2013), Wang, Li, Li (2013b), Rajkumar, Kumar, Sivaramakrishnan (2013), Liu et al. (2014), Shah, Swaminathan, Baker (2008), Zhu et al. (2012), He et al. (2013), Wang et al. (2009).
Auditing of Shared Data Integrity	Kwon et al. (2014), Wang, Li, Li (2012), Wang, Li, Li (2014), Wang, Li, Li (2013a).
Validating Backup Integrity	Chen, Lee (2014).
Automated Analysis of Processeses and System Models	
Process Mining	Jans, Alles, Vasarhelyi (2013).
Workflow Model Analysis	Accorsi, Lowis, Sato (2011), Accorsi (2011), Peterson (1977),Wen et al. (2009), Accorsi (2011).
Web Service Design Analysis	Sheng et al. (2014).

Table Appendix B Overview of identified continuous auditing methods and corresponding sources.

Appendix C – Identified Continuous Monitoring Methods

Method description	Source
Cloud Monitoring Tools and Architectures	
Cloud Monitoring Mechanisms and Tools	Shao et al. (2010), Shao, Wang (2011), Aceto et al. (2013), Fatema et al. (2014), Alhamazani et al. (2014), Katsaros, Kübert, Gallizo (2011).
Cloud Monitoring Architectures	Katsaros, Kübert, Gallizo (2011), Povedano-Molina et al. (2013), Montes et al. (2013), Kutare et al. (2010), Hasselmeyer, d'Heureuse (2010), Tovarnak, Pitner (2012), and Shao et al. (2010), Clayman, Galis, Mamatas (2010), Clayman et al. (2011), and Xiang et al. (2010), Aguado, Calero (2014).
Logging and Inspection	
Layered Cloud Logging Framework	Ko et al. (2011).
Privacy Protection based upon Log Analysis	Accorsi (2007).
Abstract Execution Log Inspection	Jiang et al. (2008).
Unstructured Logs Analysis	Fu et al. (2009).
Securing Logs	Kunz, Niehues, Waldmann (2013), Yavuz, Ning (2009).
Monitoring of virtualized Environments	
In-VM- and Out-of-VM-Monitoring	Sharif et al. (2009).
Cloud User Monitoring VM Approach	Chen, Wen (2012).
Application Monitoring Model	Gonzalez, Munoz, Mana (2011), Mana, Munoz, Gonzalez (2011).
Framework for increasing VM Security	Liu et al. (2010).
Detect Dynamic Attacks on virtualized Applications	Wang, Mao, Luo (2012).

Intrusion, Anomaly and Behavior of Malware Detection	
Intrusion Detection Systems	Lunt (1993), van der Aalst, de Medeiros (2005), Zachary, McEachen, Ettlich (2004), Best, Mohay, Anderson (2004), Modi et al. (2013).
Insider Monitoring	Ghulam, Shaikh, Shaikh (2008).
File System Integrity Tool	Kim, Kim, Eom (2010).
Anomaly Detection and Aggregation Architecture	Perols, Murthy (2012), Zhao, Zhou, Fan (2012).
Service Level Agreements Monitoring	
SLA Monitoring	Goel, Kumar, Shyamasundar (2011), Lamparter, Luckner, Mutschler (2007), Romano et al. (2011), Emeakaroha et al. (2012).
Dynamic SLA Monitoring	Comuzzi, Spanoudakis (2010).
Compliance Monitoring	
Cloud Service Configuration Validation	Chieu et al. (2012).
Compliance Validation of unstructured Business Processes	Doganata, Curbera (2009), Curbera et al. (2008).
Data Protection Compliance	Chen, Hoang (2011).
Data Location Compliance	Massonet et al. (2011).
Network Monitoring	
Network Monitoring	Wu, Zhao, Ye (2008), Nowak, Bagrij (2007).

Table Appendix C Overview of identified continuous monitoring methods and corresponding sources.

Appendix D – Interview Guidelines

Interview Leitfaden

Projekt-Hintergrund

Aktuelle Cloud-Service-Zertifizierungen suggerieren ein hohes Maß an Sicherheit, Verfügbarkeit und Compliance von Cloud-Services, bei einer Gültigkeit von ein bis drei Jahren. Aufgrund der inhärenten Dynamik und der ständigen (technischen) Weiterentwicklung von Cloud-Services, werden jedoch hohe Anforderungen an Zertifizierungen gestellt. Daher ist eine langjährige Gültigkeit im Cloud-Computing Umfeld kritisch zu betrachten. Die Einhaltung bestimmter Anforderungen und Kriterien kann über diesen Zeitraum gefährdet sein, bspw. durch das Auftreten von schwerwiegenden Sicherheitsvorfällen oder Änderungen an der Cloud-Konfiguration.

Um die Glaubwürdigkeit und das Vertrauen in ausgestellte Zertifikate zu erhöhen, und um kontinuierlich sicherzustellen, dass Cloud-Services sicher und zuverlässig angeboten werden, beschäftigt sich u. a. die Universität zu Köln mit der Forschung und Entwicklung dynamischer Zertifizierungen für Cloud-Services, die es ermöglichen kritische Anforderungen an Cloud-Services kontinuierlich und (teil-)automatisiert zu überprüfen.

Ziele der Masterarbeit

Im Rahmen der Masterarbeit wurde zunächst der aktuelle Prüfkatalog zu dem Zertifikat ‚Certified Cloud Service' überarbeitet. Zudem wurden die darin enthaltenen Controls in Zusammenarbeit mit [i01] und [i03] hinsichtlich einer kontinuierlichen Überprüfung klassifiziert. Dabei wurde festgestellt, dass es sinnvoll sein könnte, 78 Controls des Prüfkatalogs nach dem eigentlichen Zertifizierungsprozess bspw. monatlich, quartalsweise oder halbjährlich erneut zu prüfen. Um diese Vielzahl von Controls kontinuierlich und insbesondere wirtschaftlich tragbar zu überprüfen, werden (teil-) automatisierte Verfahren benötigt. Im Rahmen dieser Masterarbeit wurde eine umfangreiche Literaturanalyse durchgeführt, um (teil-) automatisierte Verfahren und Konzepte zu identifizieren. Im Anschluss werden nun Interviews durchgeführt, um Einblicke über bestehende (teil-) automatisierte Konzepte in der Praxis zu erhalten sowie die in der Literatur identifizierten Konzepte aus der praktischen Sicht zu evaluieren.

Voraussichtlicher Interviewablauf

Nach der Erläuterung des Projekthintergrunds und den aktuellen Stand der Masterarbeit, werden verschiedene Bereiche im Interview diskutiert. Die nachfolgenden Fragen dienen als mögliche Beispielfragen und zur Unterstützung. Im Interview soll aber ganz bewusst frei und detailliert über verschiedene Themen gesprochen werden, sodass einige Fragen unbeantwortet bleiben können. Ziel ist es, durch die Praxissicht verschiedene Probleme und Chancen zu identifizieren.

Mögliche Fragestellungen

- Wie sind jährliche Monitoringaudits gestaltet?
- Werden bereits computergestützte Systeme zur Unterstützung der Auditierung eingesetzt?
- Wie werden beim *Certify* Penetrationstests durchgeführt?
- Eine Vielzahl von Kriterien fordern, dass bestimmte Prozesse durchgeführt werden. Wie wird dies aktuell überprüft und wie könnte man dies automatisieren?
- Scheint es möglich, bestimmte technische Auditierungsmodule in den Provider-Systemen zu platzieren?
- Eine Vielzahl von Kriterien erwarten das (Prozess)- Dokumentationen aufrecht erhalten werden. Wie wird das bisher überprüft, Siehst du dort automatisierungspotential?
- Ein viel versprechendes Verfahren ist der Einsatz von Digitalen Agenten zur Sammlung von Daten und Analyse. Wie ist die Einschätzung hinsichtlich des Einsatzes solcher Tools?
- Können kont. Überprüfungen allein auf Staging-Systemen durchgeführt werden, um Live- und Produktionssysteme nicht zu belasten?
- Es existiert eine Vielzahl von Verfahren, die einen Auditor oder eine dritte Partei befähigen, die Datenintegrität von Clouddaten sicherzustellen. Wie wichtig werden diese Verfahren im Vergleich zu anderen Verfahren angesehen?
- Werden Entscheidungsunterstützungssysteme verwendet?
- Werden während der Auditierung konkrete Prozessmodelle oder Architekturmodelle (bspw. in UML) analysiert? Scheinen automatisierte Analysen von solchen Modellen einen Anwendungsfall in der Praxis dar?

- Wie werden der Aufwand und die Möglichkeit sich an bestehende Monitoring-dienste eines Providers anzuschließen eingeschätzt, um Audit-relevante Daten zu erhalten?
- Wie könnte man die Auditierungssysteme mit den Systemen eines Providers verbinden / ein Kommunikationskanal einrichten?
- Wie reagiert der *Certify*, wenn neue kritische Schwachstellen (bspw. Heartbleed) oder Viren bekannt werden?

Beispiel kontinuierliche Kriterien

- Entwickelte Programme oder Module müssen durch z. B. Reviews, automatisierte Tests, Vulnerability-Tests etc. gesichert sein. Hierbei sind nicht nur neu entwickelte Module zu überprüfen, sondern auch in regelmäßigen Abständen bereits existierende Module. Ziel der regelmäßigen Reviews ist es, Sicherheitsprobleme zu erkennen, die durch geänderte Anforderungen auftreten. – Monatlich
- Es muss ein wirksames Berechtigungskonzept existieren, dass den unberechtigten Zugriff auf Informationen anderer Mandaten verhindert. – Monatlich
- Vor der Integration neuer Systeme in ein produktives Netzwerksegment muss die Sicherheit des zu integrierenden Systems überprüft werden.

Appendix E – Dynamic Certification Criteria and Method Mapping

The following table presents CSC criteria that were marked as candidate for continuous monitoring and auditing. Furthermore, for each criterion assigned checklist attributes (see section 3.2) are listed. Finally, the table illustrates the mapping of dynamic certification criteria and identified methods. Due to confidentiality limitations criteria descriptions are removed, thus, only headlines for the corresponding criterion are included.

In accordance to the developed model of dynamic certification, methods that can be performed by a CSP´s internal auditing and monitoring department are mapped to the criteria. In addition, for each criterion potential auditing operations for external auditors were matched. Based upon the dynamic certification model, operations were differentiated between receiving and assessing reports (‘Report’), performing interviews (‘Interview’), penetration testing (‘Penetration Testing’), general technical analyses of cloud architectures (‘Technical Analyses’), and analyzing transmitted logs from CSP´s (‘Log Analyses’), and finally connecting to auditee´s system to exchange data (‘Connection’).

This appendix was removed to preserve confidentiality.

Appendix F – Cloud Service Certification Criteria

A list of the final set of CSC criteria is presented in original thesis. Due do confidentiality limitations this appendix is deleted.

Erklärung

Hiermit versichere ich an Eides Statt, dass ich die vorliegende Arbeit selbstständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Sebastian Lins

Köln, den 28.10.2014