Delegated Audit of Cloud Provider Chains Using Provider Provisioned Mobile Evidence Collection

Christoph Reich(✉) and Thomas Rübsamen

Institute for Cloud Computing and IT Security,
Furtwangen University of a Applied Science, Furtwangen, Germany
{christoph.reich,thomas.ruebsamen}@hs-furtwangen.de
http://wolke.hs-furtwangen.de

Abstract. Businesses, especially SMEs, increasingly integrate cloud services in their IT infrastructure. The assurance of the correct and effective implementation of security controls is required by businesses to attenuate the loss of control that is inherently associated with using cloud services. Giving this kind of assurance, is traditionally the task of audits and certification done by auditors. Cloud auditing becomes increasingly challenging for the auditor, if you be aware, that today cloud services are often distributed across many cloud providers. There are Software as a Service (SaaS) providers that do not own dedicated hardware anymore for operating their services, but rely solely on other cloud providers of the lower layers, such Infrastructure as a Service (IaaS) providers. Cloud audit of provider chains, that is cloud auditing of cloud service provisioned across different providers, is challenging and complex for the auditor.

The main contributions of this paper are: An approach to automated auditing of cloud provider chains with the goal of providing evidence-based assurance about the correct handling of data according to predefined policies. A concepts of individual and delegated audits, discuss policy distribution and applicability aspects and propose a lifecycle model. The delegated auditing of cloud provider chains using a provider provisioned platform for mobile evidence collection is the policy to collect evidence data on demand. Further, the extension of Cloud Security Alliance's (CSA) CloudTrust Protocol form the basis for the proposed system for provider chain auditing.

1 Introduction

As cloud computing becomes more accepted by mainstream businesses and replaces more and more on-premise IT installations, compliance with regulation, industry best-practices and customer requirements becomes increasingly important. The main inhibitor for even more widespread adoption of cloud services still remain security and privacy concerns of cloud customers [3]. In Germany, a preference for cloud providers that fall under German jurisdiction and also run

M. Helfert et al. (Eds.): CLOSER 2016, CCIS 740, pp. 43–64, 2017.
DOI: 10.1007/978-3-319-62594-2_3

their own data centers in Germany or at least inside the European Union can be observed recently [2]. This comes as no surprise when privacy violations that have become known to the general population in recent years are considered (e.g., NSA and Snowden revelations). A feasible way to assess and ensure compliance of cloud services regularly is by using audits. For any technical audit, information has to be collected in order to assess compliance. This automated process is called evidence collection in our system. In our previous work on cloud auditing, the focus was put on automating the three major parts of an audit system, (i) evidence collection and handling, (ii) evaluation against machine-readable policies and (iii) presentation of audit results [15,17,18].

Today, it is common to not only have a single cloud provider to provision a service to its customers, but multiple. The composition of multiple services provided by different providers can already be observed where Software as a Service (SaaS) providers host their offering on top of the computing resources provided by an Infrastructure as a Service (IaaS) provider. For instance, Dropbox and Netflix both host their services using Amazon's infrastructure. These composed services - they can be considered to form a chain of cloud providers, therefore cloud provider chain - can become very complex and opaque with respect to the flow of data between providers. Several new challenges for the auditing of such cloud provider chains can be identified, which will be discussed in this paper. The other major contribution of this paper is a proposed solution to auditing of cloud provider chains, which is an extension to our previous work in this area.

This paper is structured as follows: in Sect. 2, related research projects and industrial approaches are discussed. In Sect. 3 the authors elaborate on the definition and properties of cloud provider chains and auditing. Afterwards, a discussion of three different approaches of auditing provider chains in Sect. 4 is presented. In Sect. 5, the lifecycle of the delegated audit of cloud providers is described. How the evidence data is collected in a provider chain, how it is integrated, and automated is shown in Sect. 6. To realize the delegated audit by CloudTrust Protocol (CTP) it has to be extended (see Sect. 7). Section 8 evaluates the proposed approaches and a conclusion of the paper is drawn in Sect. 9.

2 Related Work

Standards and catalogues such as ISO27001, Control Objectives for Information and Related Technology (COBIT) or NIST 800-53 define information security controls. There are some extensions to the previous frameworks such as the Cloud Controls Matrix [4]. However, conducting audits based on these standards is mostly a manual process, still. Our proposed approach supports the automatic collection and evaluation of evidence based on policies that may stem from these frameworks and therefore could enable continuous certification.

Auditing and monitoring in cloud computing has gained more momentum in recent years and a growing number of research projects is addressing their unique challenges. Povedano et al. [13] propose Distributed Architecture for Resource

manaGement and mOnitoring in cloudS (DARGOS) that enables efficient distributed monitoring of virtual resources based on the publish/subscribe paradigm. While they introduce isolation of tenants in cloud environments, they do not at this stage show how their system would work in a multi-provider scenario.

Massonet et al. [10] propose an approach to monitoring data location compliance in federated cloud scenarios, where an infrastructure provider is chained with a service provider (i.e., the service provider uses resource provided by the infrastructure provider). A key requirement of their approach is the collaboration of both providers with respect to collecting monitoring data. Infrastructure monitoring data (from the IaaS provider) is shared with the service provider (SaaS provider) that uses it to generate audit trails. However, their main focus is to monitor virtual execution environments (VEE) that "are fully isolated runtime modules that abstract away the physical characteristics of the resource", which roughly translates to virtual machines.

Reference [8] follow the idea of tightly integrating monitoring into their management system for federated clouds, inorder to facilitate provider selection on the basis of availability and reliability metrics. They introduce service monitoring by reusing SALMonADA [12]. Their approach is geared towards provider decision making for stateless services based on performance metrics and does neither include protection mechanisms and dynamic collector distribution that are required in a system for evidence collection in the cloud.

Security and privacy auditing are increasingly important topics in cloud auditing. They demonstrate that security controls are put in place by the provider and also that they are functioning correctly (i.e., data protection mechanisms are working correctly and effectively). There are some projects working on the architectural and interface level regarding the automation of security audits, such as the Security Audit as a Service (SAaaS) project [7]. SAaaS is specifically designed to detect incidents in the cloud and thereby consider the dynamic nature of such ecosystems, where resources are rapidly provisioned and removed. However, SAaaS does not address provider chain setups or treat gathered data as evidence.

ABTiCI (Agent-Based Trust in Cloud Infrastructure) describes a system used for monitoring [19]. All relevant parts of a cloud infrastructure are monitored to be able to detect and verify unauthorized access. Integrity checks are done at boot-time, using Trusted Platform Module (TPM) boot or at runtime. Monitoring hardware and software configurations allow the system to detect changes at runtime. The aforementioned system is similar to our approach. Instead of using agents we utilize CTP. Furthermore, our approach relies on evidence collection through audits with pull and trigger mechanisms.

A centralized trust model is introduced by Rizvi et al. [14]. Trust between consumer and provider is established by using an independent third-party auditor. With the adoption of a third-party auditing system, consumers are able to create baseline evaluation for providers they have never worked with to generate initial trust. The model acts as a feedback mechanism providing valuable insight into the providers processes. After initial trust was generated the third-party

auditor continues to obtain trust values for the consumer. We see initial trust in the provider as a given factor and focus on obtaining trust values based on evidence within a multi-provider scenario.

The DMTF is also working on cloud auditing with the Cloud Audit Data Federation (CADF) working group. They focus on developing standardized interfaces and data formats to enable cloud security auditing [6]. A similar project is the Cloud Security Alliance's Cloud Trust Protocol (CTP), which defines an interface for enabling cloud users to "generate confidence that everything that is claimed to be happening in the cloud is indeed happening as described, ..., and nothing else" [5], which indicates an additional focus on providing additional transparency of cloud services. The latter two projects, however, do not elaborate on how the interfaces should be implemented nor do they describe explicitly focus on privacy and accountability. We use CTP as a basis and propose its extension and use in our proposed auditing system to enable automated auditing of cloud provider chains.

3 Complex Cloud Provider Chains for Service Provision

While a lot of today's cloud use cases only involve one service provider for service provision, there are also many cases where multiple providers are involved. A prominent example is Dropbox that heavily uses Amazon's S3 and EC2 services to provide its own SaaS offering.

There are several terms for the concept of provider chains such as *federated cloud*, *inter-cloud* and *cloud service composition*. In this work these terms are used synonymously. The concept of a provider chain is defined as follows:

1. At least two cloud providers (either IaaS, PaaS or SaaS) are involved in the provision of a service to a consumer (who can be an individual or business).
2. One cloud provider acts as a primary service provider to the cloud consumer.
3. Subsequent cloud providers do not have a direct relationship with the consumer.
4. The primary provider *must* be and the subsequent providers *can* be cloud consumers themselves, if they use services provided by other clouds.
5. The data handling policies agreed between the cloud consumer and the primary service provider must not be relaxed if data is processed by a subsequent provider.

The terms *cloud consumer* and *cloud customer* are used synonymously as well, while relying on the definition of a cloud consumer and auditor provided by NIST [9].

Figure 1 depicts a simplified scenario where three cloud service providers are involved in provisioning of a seemingly single service to a cloud consumer. The SaaS provider acts as the primary service provider, while it uses the PaaS provider's platform for hosting its service. The PaaS provider in turn does not have its own data center but uses resources provided by an IaaS provider.

The data handling policy applies to the whole chain (depicted by the dashed rectangle in Fig. 1). Data handling policies thereby govern the treatment of data such as data retention (the deletion of data after a certain time), location (geographical restrictions) and security requirements (access control rules and protection of systems that handle the data).

All cloud provider produce evidence of their cloud operations.

3.1 Auditing Cloud Provider Chains

According to NIST, a cloud auditor is defined as "A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation" [9]. In our proposed system, the auditor is supported by a system for automated evidence collection and assessment. Evidence in the audit system is any kind of information that is indicative of compliance with policies or a violation of those. Typically, evidence is collected at the auditee. In general, an auditee is an organization that is being audited, which in this paper, is always a cloud provider.

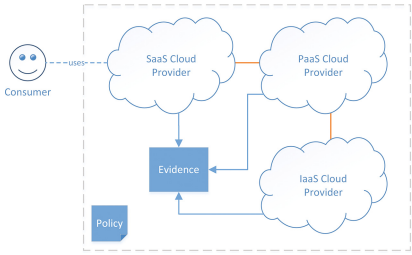

Fig. 1. Cloud provider chains.

Complex cloud service provision scenarios introduce new challenges with respect to auditing. While in a typical scenario, where there is one cloud provider and one cloud consumer, policies can be agreed upon relatively easily between the two, this is not as easy in a provider chain. In fact, the cloud consumer might not be aware of or even interested in the fact that there is an unknown third-party that might have access to his data as long as his expectations regarding the protection and processing of his data are fulfilled. However, to assert compliance, the whole chain of providers, including data flows that are governed by the previously mentioned policies, have to be considered. This means that an audit with respect to a single policy rule may need to be split into several smaller evidence collection and evaluation tasks that are distributed among the providers.

For instance: assuming there is a restriction on data retention put in place that states that certain types of data (e.g., Personal Identifiable Information PII) has to deleted by the provider after a certain fixed period of time and no copies may be left over. This restriction can stem from regulatory framework such as the European Data Protection Directive or simply preferences that were stated by the data subject, whose data is being processed in the cloud. Such requirements can be formulated and enforced in for example the Accountability PrimeLife Policy Language (A-PPL) and its enforcement engine (A-PPL-E) [1].

Auditing for compliance with such a policy requires, on a higher level, the check for the implementation of appropriate mechanisms and controls at each provider where the data itself or a copy thereof could have been stored. On a lower-level,

the correct enforcement of the data retention rule could be evaluated in an audit by using evidence of data deletion that is being collected from all the cloud providers. In the overview depicted in Fig. 1, that evidence could comprise of:

- Data deletion enforcement events generated by the service at the primary service provider as a reaction to the retention period being reached,
- Database delete log events produced by a database management system at the PaaS provider,
- And scan results on the IaaS level for data that may be still available outside of the running service in a backup subsystem provided by the IaaS provider.

The importance of widening the scope of audits in such dynamic scenarios is apparent, especially if at the same time the depth of analysis is widened beyond checking whether or not security and privacy controls are put in place.

3.2 Evidence of Compliance in Cloud Provider Chains

At the core of any audit is evidence of compliance or non-compliance that is being analyzed. The types of evidence are closely linked to the type of audit (e.g., security audit, financial audit etc.) and are - from a technological perspective - especially diverse in the cloud due to the heterogeneity of its subsystems, architectures, layers and services. The notion of evidence for cloud audits was discussed in our previous work in more detail [15].

In general, we follow the definition of digital evidence that is "information of probative value that is stored or transmitted in binary form" [20]. This means, that the types of evidence are diverse and include for example logs, traces, files, monitoring and history data from cloud management system like OpenStack's Nova service.

Evidence collection at a single cloud provider is already a complex task due to the diverse types of evidence sources and sheer amount of potentially required data that is being produced continuously. In a provider chain, these problems are intensified by the introduction of administrative domains and the lack of transparency regarding the number of involved providers and their relationships.

Another problem that is introduced with the concept of provider chains are changing regulatory domains. In a single-provider scenario, there are typically only two regulatory domains to be considered: (i) the one that applies to the cloud consumer and (ii) the one that applies to the cloud provider. With the addition of more cloud providers, the complexity of achieving regulatory compliance increases tremendously.

A simple example for such a case is the recent decision of the European Court in 2015 to declare Safe Harbor invalid, which leads to data transfers outside the European Union that are only governed by Safe Harbor to be invalid. In a provider chain, where a European Cloud provider transfers data about European individuals to another provider in the US, regulatory compliance could have been lost overnight. Here, it can be seen that regulatory domains can have a tremendous impact on how a compliance audit may have to look like, and on the type of evidence that may need to be collected at the different providers.

As previously suggested, the third major challenge for evidence collection in cloud provider chains is their inherent technological heterogeneity. APIs, protocols and data formats differ by provider and typically cannot be integrated easily (e.g., providers offering proprietary APIs). There are some approaches to homogenize some of the technologies, such as for example CSA CloudTrust Protocol [5] that aims to provide a well-defined API that enables cloud providers to export transparency-enhancing information to auditors and cloud consumers.

4 Audit Frameworks for Cloud Provider Chains

In this chapter, we illustrate different variations of auditing cloud provider chains. We thereby focus on traditional individual audits and delegated provider audits.

4.1 Audit Frameworks and Audit Automation

Policy compliance assessment and validation is the main goal of our audit system. Policies can be of various kinds, for instance, a data protection policy is a typical tool used by cloud providers to frame their data protection and handling practices. In such policies, limits and obligations that a provider has to fulfill are defined in well-known standards, frameworks and industry best practices, like CSA's Cloud Controls Matrix (CCM) [4] mentioned in "related work" (Sect. 2). Typically, these documents are not machine-readable and are geared towards limiting liability of the provider, but for automation this is necessary precondition. The nature of the obligations defined in standards are general and the expertise of an auditor is then to design machine-readable policies, which can be monitored.

4.2 Individual Provider Audits

Figure 2 describes the process of auditing individual providers in a service provision chain. All policies will be distributed to each provider (as seen in Fig. 2). Policy distribution can either be:

1. Manual policy evaluation: This approach is based on the specified policy documents (e.g., terms of service in human-readable form) given by the provider. The auditor manually maps statements of such documents to information requests for the providers (e.g., asking for specific process documentation or monitoring data and audit logs).
2. Deploying machine-readable policies: In this approach the auditor deploys a machine-readable policy document (XML, JSON) onto the provider. The provider will then automatically audit the tasks specified within the document. The auditor can request the results for the audited policy rules to verify if everything is fulfilled. The policy needs to be deployed to each involved provider. Within this approach new policies can easily be added and deployed for automated auditing.

The audit results are used to assure the consumer that policy and rule compliance is given or not. As previously described, a service provision chain contains at least one provider. In this case, two providers - a primary and a 2nd-level provider. To audit the service as a whole, it is necessary to audit each provider separately and then aggregate the results to form a complete picture of the service from an audit perspective. This means, that regarding data handling policies (e.g., location restrictions, access control etc.), each provider that holds data is audited. The same is true for the auditing of security and privacy controls that are put in place at the providers. Obviously, the consumer-facing provider has to transparently disclose all his sub-providers and notify auditors about every sub-provider his data was stored at and where his data is currently stored. Even though not every provider will get the consumer's data, the auditing process gains more complexity with an increasing number of Nth-level providers. Requests must be sent to each provider separately and each provider will deliver audit reports to the auditor.

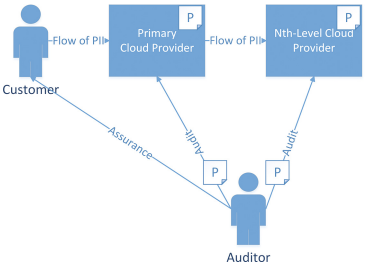

Fig. 2. Individual audit.

4.3 Delegated Provider Audits

An alternative to individual audits are what we call delegated audits, where the auditor only interfaces with the primary service provider that in turn takes over the auditing of its sub-provider(s). Therefore the auditor only has to audit the primary service provider to obtain policy compliance results of all involved service providers.

This allows less influential stakeholders such as the cloud consumer to act as an auditor towards the primary provider while not having the same rights towards the Nth-level provider(s). Whereas the individual audit scenario is an example of how chain audits could be performed with more influential stakeholders, such as data protection authorities. Figure 3 depicts the delegated provider audit scenario. Every audit request is sent to the primary provider who will then extract CTP calls from a previously deployed policy document (machine-readable document deployed by the auditor). Since the primary provider is acting as a mediator he has to delegate requests and communication. Existing problems regarding policy compliance is of major concern for the primary provider because complaints will always be addressed to him, even if he is not responsible for a failed audit. For the case a given audit response did not satisfy policy compliance the consumer will contact the primary provider with a complaint (e.g., data was transferred outside valid location).

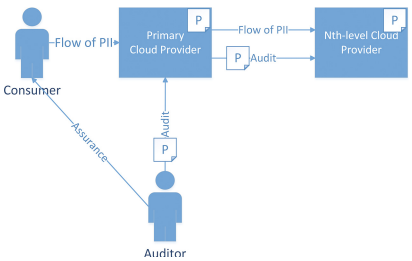

Fig. 3. Delegated audit.

On the other hand the consumer's payoff can be much higher due to the centralized structure using a mediator ensuing low complexity for the auditor. Therefore, he can always rely on the data controller to forward his request to the data holding sub-provider without the need of adaption (send requests to different entities, use different API-calls).

5 Audit Lifecycle in Delegated Provider Audits

In the following section the audit system lifecycle in delegated provider audits is described in three phases: **(i)** *Preparation*, **(ii)** *Processing* and **(iii)** *Presentation*. In the following, we describe each phase in more detail:

(i) Preparation Phase: The most important task during the *Preparation Phase* is resource identifier distribution, which is required for request handling. Request handling is done using unique resource identifiers (URI), which are used to identify any kind of resource that is part of an audit. A URI unambiguously identifies an object within a provider's domain. In our approach, each provider has its own namespace in which identifiers can be assigned arbitrarily.

The preparation process *Policy Adding* allows the auditor to create new machine-readable rules based on already existing policies, like specifying a new data location rule to ensure that his data will not leave his countries jurisdiction. From the new rule, auditable elements are derived, that an automated audit process provides all necessary information to enable the possibility of policy compliance assessing. Auditable elements include for example the location of data, logs and configurations.

The *Policy Mapping* process, maps each new added rule or policy to transparency elements and associated requests. If a newly added rule cannot be mapped to an already existing transparency element a new element needs to be created. The mapping is done based on the specified policies. For this reason the policy adding process is limited to already defined policies and the associated rules within the contract. During the mapping each non-standard policy (i.e., a policy that requires a transparency element that is not part of CTP) will receive an URI and all necessary data sources needed to answer a request. The mapping process generates URIs and defines all auditable attributes for an element.

The preparation process *Policy Distribution* propagates the resource identifiers throughout the system. Each sub-provider sends his resource identifiers to the primary provider. Afterwards, when all identifiers are known by the primary provider, they will be forwarded to the auditor.

(ii) Processing Phase: The information defined in *Preparation Phase* will be collected and stored in the evidence store. Is the primary cloud provider requesting evidence using the CTP, the secondary cloud provider retrieves the information from its evidence store and replies to the CTP response. A policy evaluation is done to determine the policy compliance.

(iii) Presentation: Within this phase the results are presented to the auditor. Thereby, each requested element will be presented containing the name of the policy rule as well as its achieved compliance state.

Depending on the auditor, the lifecycle can continue with *Preparation Phase* again. Returning to the *Preparation Phase* is necessary if new policies/rules were added or in a continuous auditing scenario, where policy compliance is audited in short intervals or event-driven (e.g., on new or changed policy, on infrastructure change or on custom triggers defined by the auditor). During the new cycle only newly generated URIs will be distributed.

6 Approaches for Collecting Evidence in Cloud Provider Chain Audits

In provider chains the problem of collecting evidence for audit purposes in a service provider chain differ in the following aspects:

1. The level of control an auditor has over the extent of the data that is being published, i.e. whether the auditor is limited to information that a provider is already providing or if he has more fine-grained control and access to a provider's infrastructure.
2. Technical limitations imposed by the technological environment, i.e. the extent to which cloud providers have to implement additional evidence collection mechanisms.
3. The expected willingness or acceptance to provide such mechanisms by the publishing service provider, i.e. the potential disclosure of confidential provider information and required level of access to the provider's systems.

In the remainder of this chapter, three approaches are described and rated by the above-mentioned factors.

The focus lies thereby on inspecting common components at two exemplary cloud providers that form a provider chain for the provision of a service. These components are:

AuditSys. An audit system that enables automated, policy-based collection of evidence as well as the continuous and periodic evaluation of said evidence during audits.

Collector. A component that enables the collection of evidentiary data such as logs at various architectural layers of the cloud, while addressing the heterogeneous nature of said evidence sources by acting as an adapter.

Source. A location where evidence of cloud operations is generated.

Now we consider three approaches how to collect the evidence. The first approach focuses on reusing already existing evidence sources by collecting via remote APIs of a cloud system. The second approach uses provider-provisioned evidence collectors and the third approach leverages the mobility of software agents (as used in the prototype implementation of our system) for evidence collection.

6.1 Remote API Evidence Collector

The first approach for collecting evidence that is relevant to automated auditing, leverages already existing APIs in cloud ecosystems. Several cloud providers, such as Amazon (CloudWatch) or Rackspace (Monitoring), already provide improved transparency over their cloud operations by providing their customers with access to proprietary monitoring and logging APIs. The extent to which data is shared is typically limited to information that is already produced by the cloud provider's system (e.g., events in the cloud management system) and restricted to information that immediately affects the cloud customer (e.g., events that are directly linked to a tenant).

Data such as logs that are generated by the underlying systems are very important sources of evidence, since they expose a lot of information about the operation of cloud services. A specific example of such evidence are for instance: VM lifecycle events (created, suspended, snapshotted etc.) including timestamp of the operation and who performed. This can be requested from OpenStack's Nova service via its REST interface. The type of information is highly dependent on the actual system, the granularity of the produced logs and the scope of the provided APIs. For instance, on the infrastructure level, there are log events produced and shared that provide insight on virtual resource lifecycle (e.g., start/stop events of virtual machine).

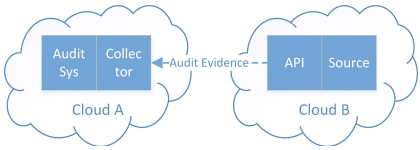

Fig. 4. Remote API evidence collector.

Figure 4 depicts such a scenario. The AuditSys at Cloud A operates a collector that implements the API of the remote data source at Cloud B. It is configured with the access credentials of Cloud A, thus enabling the collector to request evidence from Cloud B. Furthermore, since different services may provide different APIs (e.g., OpenStack vs. OpenNebula API), the collector is service-specific. For instance, a collector implements the data formats and protocols as defined in the OpenStack Nova API to collect evidence about the images that are owned or otherwise associated with Cloud A as a customer of Cloud B.

Level of Auditor Control: The amount of evidence that can be collected is severely limited by the actual APIs that are provided by a cloud provider. It is either: (i) the evidence that an auditor is looking for is immediately available because the provider already monitors all relevant data sources and makes that data accessible via the API or (ii) the data is not available. Since a lot of the cloud provider's systems expose remote APIs anyway, they have to be considered. However, the completeness of the exposed APIs and therefore the completeness of the collected evidence is questionable due to the aforementioned reasons.

If an auditee for some reason does not implement or provide access to the audit system, an auditor may still collect evidence to a limited degree using this approach.

Technical limitations: If lower-level access to the providers infrastructure is required to collect evidence (e.g., log events generated on the network layer or block storage-level access to data), an auditor might not be able to gain access to that information.

Acceptance: This approach poses some challenges with respect to security, privacy and trust required by the auditee. Since the auditee is already exposing the APIs publicly, it can be expected that they will be used for auditing and monitoring purposes. The implementation of security and privacy-preserving mechanisms on the API-level is therefore assumed. However, the extent to which such mechanisms are implemented highly depends on the actual implementation of the APIs on the provider side.

While this way of providing evidence to auditors is likely to be accepted by cloud providers, it may be too limited with respect to the extent to which evidence can be collected at lower architectural levels.

6.2 Provider Provisioned Evidence Collector

In this approach, the audit system still is the main component for evidence collection. All cloud providers that are part of the service provision chain are running a dedicated system for auditing. However, the instantiation and configuration of the collector is delegated to the auditee. The auditee assumes full control over the collector and merely grants the auditor access to interact with the collector for evidence collection. The auditee (see Cloud B in Fig. 5) provisions evidence collectors and provides access to them to the auditor. The auditor (who is using AuditSys at Cloud A) configures evidence collection for the audit to connect to the collectors at Cloud B.

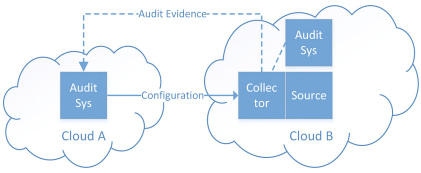

Fig. 5. Provider-provisioned evidence collector.

Level of Auditor Control: The configuration of the evidence collector can be adjusted by the auditor to a degree that is controlled by the auditee (e.g., applying filters to logs). He is provided limited means to configure a collector but no direct, low-level access such as freely migrating the collector in the auditee's infrastructure. At any time, the auditee can disconnect, change or otherwise control the collector. An auditor may be put off by the limitations posed by this approach, since he is effectively giving up control over the central part of evidence collection and is relying solely on the cooperation of the auditee. For instance, simple tasks such as reconfiguring or restarting a collector may require extensive interaction between the two audit systems and potentially intervention by a human (e.g., an administrator).

Technical limitations: This approach is only limited by the availability of collectors for evidence sources.

Acceptance: In this approach, the auditee retains full control over the collector and the potential evidence that can be collected by it. The auditor can take some influence on the filtering of data that is collected from the evidence source and on general parameters, such as whether evidence is pushed by or pulled from the collector. Most of the baseline configuration though, is performed by the auditee (such as access restrictions and deployment of the collector). The auditor's ability to influence the collector is severely limited by the restriction of interactions to a well-defined set of configuration parameters and the evidence exchange protocol. This level of control that the auditee has over the evidence collection process may have positive influence on provider acceptance.

6.3 Provider Provisioned Platform for Mobile Evidence Collectors

This approach is specific to a central characteristic of software agent systems, which is the ability to migrate over a network between runtime environments. In this approach, the migration of evidence collectors between separate instances of the audit system running at both Cloud A and B is proposed.

In our implementation, we opted for the well-known Java Agent Development framework (JADE) for implementing collectors. The migration of collectors between providers is thereby performed by using JADE's mobile agent capabilities.

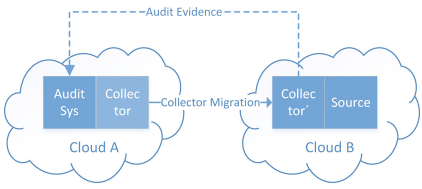

Fig. 6. Mobile evidence collector.

As depicted in Fig. 6, the auditor prepares the required collector fully (i.e., agent instantiation and configuration) and then migrates the collector (shaded box named *Collector*) to the auditee (*Collector'*). There, the collector gathers evidence that is sent back to the auditor for evaluation. Generally however, agents do not cross from one particular administrative domain to another, but remain at one. In this case, the collector crosses from Cloud A's administrative domain to Cloud B. This may have significant impact on the acceptance of the approach.

Level of Auditor Control: The auditor retains full control over the type of collector and its configuration. The auditee may not in any way change or otherwise influence the collector since this could be deemed a potentially malicious manipulation.

Technical limitations: Since the auditor knows most about the actual configuration required for a collector, it is logical to take this approach and simply hand-over a fully prepared collector to the auditee. However, this only works if both run the same audit system, or the auditee at the very least provides a runtime environment for the collector. In any case, this approach offers the most complete and most flexible way of collecting evidence at an auditee due to the comprehensive evidence collection capabilities.

Acceptance: The main problem with this approach is required trust by the auditee. Since the collector that is being handed over to him by the auditor is in fact software that the auditee is supposed to run on its infrastructure, several security, privacy and trust-related issues associated with such cross-domain agent mobility need to be addressed. Several security controls need to be implemented in order to make cloud providers consider the implementation of an audit system including the proposed approach of using mobile collectors.

The main security concerns of this approach stem from the fact that the auditee is expected to execute software on his infrastructure over which he does not have any control. He cannot tell for certain whether or not the agent is accessing only those evidence sources which he expects it to.

Without any additional security measures, it cannot be expected that any cloud provider is willing to accept this approach. However, with the addition of security measures such as ensuring authenticity of the collector (e.g., using collector code reviews and code signing) this approach becomes more feasible. The discussion of such measures depends on the technology used by the implementation and is out of scope of this paper. Without any additional measures, it can be assumed that this approach is only feasible, if the auditor is completely trusted by the auditee. In that case, this approach is very powerful and flexible.

6.4 Critical Review

All three approaches for evidence collection in provider chains have their distinct advantages and disadvantages. Using remote API evidence collectors is simple, quickly implemented, securely and readily available, but severely limited regarding access to evidence sources. Using provider-provisioned evidence collectors is more powerful with respect to access to evidence sources, but requires more effort in the configuration phase and leaves full control to the auditee. Using mobile evidence collectors is the most flexible approach that allows broad access to evidence sources at the auditee's infrastructure and leaves full control over the evidence collection to the auditor. Therefore, a balance has to be struck between broad access to evidence sources when using mobile collectors (effectively having low-level access to logs and other files for evidence collection) and more limited access when using remote APIs (evidence limited to what the system that exposes the API provides).

In the audit system, the use of remote APIs is integrated due to its simplicity and mobile collectors due to their flexibility and powerfulness as the main approaches to evidence collection.

7 Extending CloudTrust Protocol (CTP) for Provider Chain Auditing

So far there is no intention of permit other providers a more flexible evidence collection by defining audit evidence collectors on demand as described in "Provider Provisioned Platform for Mobile Evidence Collectors" (Sect. 6.3). Therefore auditors depend on standards like the CloudTrust Protocol (CTP).

7.1 CloudTrust Protocol (CTP)

The CloudTrust Protocol [5] establishes a mechanism that allows users to audit a CSP. An auditor can choose from a set of transparency elements for instance: geographic location of data objects and affirmation or results of latest vulnerability assessments. CTP has 23 pre-defined transparency elements and supports user-specified elements on which cloud consumer and provider agreed on. The purpose of the CTP is to transparently provide the user with important information about the cloud to show that processing is done as promised. By providing information about the inner-workings of the cloud service (with respect to the transparency elements), trust between the cloud provider and the consumer is supposed to be strengthened. For automation purpose of the audit the CTP had to be extended.

7.2 CloudTrust Protocol (CTP) Extended

In our approach, we leverage CTP as a means for evidence exchange between cloud providers in complex cloud auditing scenarios [16]. Additional functions and components are located above the protocol (as seen in Fig. 7) and are responsible to exchange requests and responses with the CTP. This structure enables us to utilize the benefits of CTP out of the protocol's operational area without changing the protocol itself. Although the operational structure of CTP remains unchanged some optimisations for audit reports are required to be able to transfer additional information e.g. more detailed user access lists. In this case, the additional information would give the auditor not only authorized users but also since when they have authorization and who authorized user permissions.

Figure 7 illustrates the systems architecture in a two provider scenario. Within the figure it is assumed, that the Preparation Phase did end and all for the audit request necessary policies and rules were already mapped and distributed. Incoming transparency element requests will directly go to the Remote Evidence Collection component. In the following paragraph the system components of our proposed approach are described:

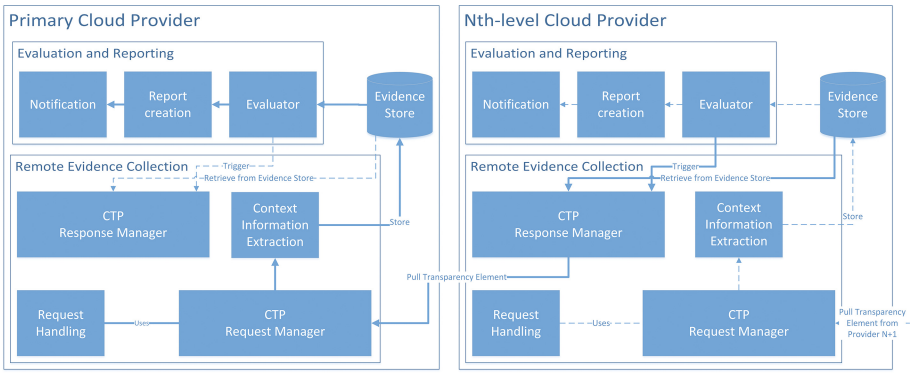

Fig. 7. Multi-provider audit system architecture.

Component: Remote Evidence Collection Consists of 4 Modules (see Fig. 7):

Request handling: Every incoming audit request will arrive at the Request handler of the primary provider. A decision is made which resource identifier should be used based on current data location (Nth-level provider, primary provider). The resource identifiers are used to set up CTP-calls. Therefore, it will forward each request to the CTP Request Manager. Each request is processed separately to guarantee that context information or states do not get mixed up.

CTP Request Manager: The Request Manager, sends each given request to the CTP Response Manager of a Nth-level provider (solid line in Fig. 7 between both providers) using a pull pattern (see [16]). Inter-provider communication is initiated by the Request Manager.

Context information extraction: An incoming CTP-response contains the general response (specified in [5]) as well as the corresponding context information and the compliance state for the requested element. The context information are extracted from the response and securely stored inside the evidence store. The remaining information which are used for report creation are stored as well for the audit report creation.

CTP Response Manager: After receiving a request the Response Manger pulls data from the evidence store if the Nth-level provider is not able to determine the compliance state of his own or receives the data from the Evaluator. Obtained results are packed into a CTP-response and sent back to the primary providers CTP Request Manager. In case a trigger is fired the Response Manager will push the response immediately to the primary providers CTP Request Manager even in the absence of an audit request for the triggered element. A primary provider might be a Nth-level provider of another provider and thus needs a Response Manager. Requests for context information are send from the auditor to the primary providers Response Manager. Like a normal response the Manager pulls the context information for the requested object from the evidence store and writes them into a CTP-Response.

Component: Evidence Store is a database containing all audit results (including context information) for the primary and its Nth-level providers. Each participating provider has its own evidence store where his achieved audit results are stored and can be pulled from by pending requests. The main purpose of the evidence store is to provide audit evidence and to make them accessible to the auditor.

Component: Evaluation and Reporting with 3 Modules:

Evaluator: The Evaluator runs policy compliance checks on all obtained results used for report creation. Achieved results can get one of three possible compliance states depending on the level of fulfillment:

- **State 1 successful:** The results obtained fulfills the policy.
- **State 2 partially:** A policy is partially fulfilled.

– **State 3 failed:** No records for this policy were found or the given results were unsatisfying.

Configured triggers (see Rübsamen [16]) are fired if a compliance check for a request failed or a deviation from the trigger specification is identified.

Report creation: The stored content (state, CTP transparency elements results) is used to create the final report. The report can be of different types, for instance a representation of the results on a web dashboard or in a auto-generated document.

Notification: An audit can take some time to finish. This largely depends on the size and scope of the audit. Therefore, asynchronous mechanisms are required to present audit results. An auditor can be notified via mail when his audit is finished and his audit report is available.

Implementation of each above described part is mandatory for every provider. It is possible that a primary provider is a Nth-level provider in a different audit-chain, whereas a Nth-level provider might be a primary provider in another audit chain.

A functional scenario based evaluation, a STRIDE [11] analysis, and a general security analysis can be found in the paper Rübsamen [16].

8 Scenario-Based Provider Chain Auditing Evaluation

In the previous Sect. 6, the approaches that can be taken when collecting evidence for auditing purposes in cloud provider chains were described. In this section, it is demonstrated how to incorporate the feasible approaches into an extension of the proposed audit system to enable automated, policy-driven auditing of cloud provider chains. The focus is put on the *Remote API Evidence Collector* and *Provider Provisioned Platform for Mobile Evidence Collectors* approaches (see Sects. 6.1 and 6.3 respectively). The approach is validated by discussing a fictitious use case.

8.1 Audit Agent System

In Fig. 8, an example deployment of the automated audit system is depicted. This deployment is not necessarily representative of real-world cloud environments but used to highlight possible combinations of services and data flows that can happen in a multi-cloud scenario. There are four cloud providers, which are directly or indirectly involved in the service provision. The SaaS provider A1 uses the platform provided by a PaaS provider B1, who does not have its own data center but uses computing resources provided by yet another IaaS provider C1. The IaaS provider C2 provides a low-level backup as a service solution that is used by provider C1. To enable auditing of the whole provider chain, each provider is running its own instance of the audit system (AuditSys, see Sect. 6).

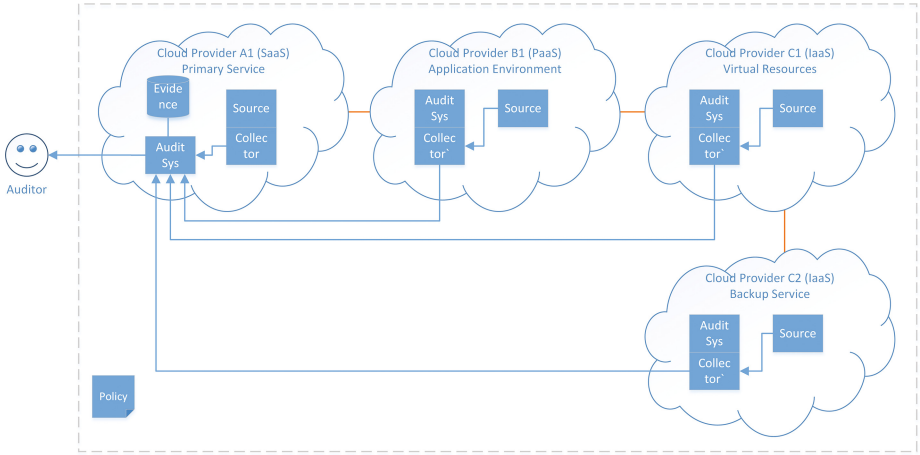

Fig. 8. Provider chain auditing architecture.

8.2 Provider Chain Auditing Extension

The auditor that uses AuditSys at the primary service provider A1 defines and configures continuous audits based on data protection and handling policy statements. Since these policy statements do not include any information about the service architecture, the auditor introduces his knowledge about the cloud deployment into the audit task, by defining evidence collection tasks that gather data on the PaaS and IaaS layer and also at the primary service provider. An audit task consists of collector, evaluator and notification agents. The type of evidence collection approach that has to be taken (as described in Sect. 6) is also defined by the auditor.

In this scenario it is assumed that all providers allow the auditor at A1 to collect evidence using the mobile evidence collectors and that the infrastructure providers also provide the auditor with access to their management system's APIs. As previously mentioned, the auditor is assumed trustworthy by all parties, which enables broad access to all cloud providers. Additionally, it is assumed that all cloud providers are acting in good faith and see the audit process as an opportunity to transparently demonstrate that they are acting in compliance with data handling policies.

As depicted in Fig. 8, the auditor uses A1's AuditSys to define and audit task based on the data handling policy that is in effect. That task refers to the data retention obligation that was described earlier in Sect. 3.1. The retention time is defined as 6 months for every PII data record that is gathered about the users of provider A1. If the retention time is reached, the following delete process is executed as part of the normal operation of the service A1 provides:

1. The delete event fires at A1 due to max retention time being reached and the event is propagated to B1.

2. The data record is deleted from the database at B1.
3. The database is hosted on virtual machines provided by C1 and therefore does not require any delete actions.
4. A backup of the B1's database is available in C2's backup system and the delete action was not triggered in C2.

As part of the delete event, the following evidences are collected by the mobile evidence collectors to build an evidence trail for compliance evaluation at A1.

1. The data retention event is recorded by the collector running at A1.
2. The delete action of the database is recorded by the collector running at B1.
3. No evidence is recorded by the collector at C1 since there are no leftover copies such as virtual machine snapshots available.
4. The backup's meta-data such as creation timestamps are recorded as evidence by the collector at C2.

The evidence from all collectors (A1, B1, C2) is sent to the AuditSys at A1, where it is evaluated and a policy compliance statement is generated for the auditor. In this particular case, a policy violation is detected, because the audit trail shows that the record that should have been deleted is still available in a backup at C2. Provider A1, and B1 acted compliant by deleting the data, whereas C1 never stored a copy outside of B1's database.

8.3 Pre-processing and Intermediate Results of Audit Evidence Evaluation

The audit system uses a component at the AuditSys that is responsible for storing evidence records that are collected by the collector agents. Externally collecting evidence and merging it at a central evidence store that is only reachable via the network, can easily become a bottle-neck in audit scenarios where either a lot of evidence records are produced externally or where the record size is big. This obviously has significant impact on the scalability of the whole system.

The problem can be addressed by making the evidence store (which is just a specialized form of an agent with a secure storage mechanism) distributable and also by de-centralizing parts of the evidence evaluation process. There are generally two concepts:

Pre-processing allows the evidence collector agent to apply filtering and other types of evidence pre-processing. The goal is to reduce the amount of collected evidence to a manageable degree (without negatively impacting the completeness of the audit trail) and to reasonably reduce the amount of network operations by grouping evidence records and storing them in bulk. For example, by filtering the raw data at the evidence source for certain operations, subjects, tenants, or time frames. Data that is not immediately required for the audit is filtered out.

Intermediate Result Production: A second pre-processing strategy is to move (parts of) the evaluation process near the collector. This means that the collected evidence is already reduced to the significant portions that indicate partial compliance or violation of policies. However, this strategy requires specific audit task types (e.g. audit result, that can be produced by combining intermediate results).

The three concepts bring several implications with them with respect to privacy and security. Pre-processing can be considered a manipulation of evidence. Therefore, the unaltered source upon which the pre-processing happened should be protected to later be able to trace pre-processed evidence back to its unaltered form. Immediate result production effectively moves the evaluation of evidence step of the audit into the domain of the auditee, where it would be easy for him to manipulate the result. However, the same applies to the collection of evidence as well where an auditor can intentionally manipulate the evidence source or the collector.

This case is not considered in the current iteration of the system but it is assumed that cloud providers (auditees) are acting in good faith. This assumption can be justified by the potential increase in transparency and the associated strengthening of trust in the cloud provider that can mean a competitive advantage. On the other hand, intentional manipulation of evidence or intermediate results can have disastrous impact on a provider‘s credibility, reputation and trustworthiness upon detection.

9 Conclusions

Cloud auditing is becoming increasingly important as cloud adoption increases and compliance of data processing is put into focus of the cloud consumer. While there are many systems for monitoring cloud providers (with varying level of completeness), there are fewer systems that automate audit tasks and even fewer still that enable continuous auditing, which is a key enabler of continuous certification. In this paper we discussed two different types of chain audits: (i) individual provider audits where the auditor has to audit each Nth-level provider separately and (ii) delegated provider audits where the primary cloud provider acts as an mediator. A powerful approach, that provides a provider prvisioned platform for mobile evidence collection allows on demand and policy-driven evidence collection, has been shown. We also discussed the applicability of data handling and privacy policies and how they apply in complex scenarios where multiple providers share a cloud consumer's data. In the latter part of this paper, we focused on the architectural integration of the CloudTrust Protocol in the evidence collection and transport of our audit system. Finally, their implementation in an audit system was presented and validated using a scenario-based approach. It was shown how automated cloud audits can be extended to scenarios, where more than one cloud provider is involved in the service provision. We evaluated the functional soundness by demonstrating an audit scenario that involves a cloud consumer using a service that is intransparently provided

by two different cloud providers. Additionally, we evaluated our approach by defining a threat model using threat scenarios and addressing those threats.

In our future work, we focus on even more complex service provision scenarios, where even more layers of service providers are involved. We will also put special focus on ensuring the scalability of our approach. Another interesting topic emerges, when any of the cloud providers is considered untrustworthy. This can be the case when a malicious insider tries to intrude in our system. We consider ensuring the integrity of evidence in the whole chain of providers to be a major challenge.

Acknowledgements. This work has been partly funded: EC:FP7/2007-2013, 317550, A4Cloud.

References

1. Azraoui, M., Elkhiyaoui, K., Önen, M., Bernsmed, K., Oliveira, A.S., Sendor, J.: A-PPL: an accountability policy language. In: Garcia-Alfaro, J., Herrera-Joancomartí, J., Lupu, E., Posegga, J., Aldini, A., Martinelli, F., Suri, N. (eds.) DPM/QASA/SETOP -2014. LNCS, vol. 8872, pp. 319–326. Springer, Cham (2015). doi:10.1007/978-3-319-17016-9_21
2. Bitkom Research GmbH: Cloud Monitor 2015 (2015). https://www.kpmg.com/DE/de/Documents/cloudmonitor%202015_copyright%20_sec_neu.pdf
3. Cloud Security Alliance: Top threats to cloud computing survey results update 2012 (2013). https://downloads.cloudsecurityalliance.org/initiatives/top_threats/Top_Threats_Cloud_Computing_Survey_2012.pdf
4. Cloud Security Alliance: Cloud Controls Matrix (2014). https://cloudsecurityalliance.org/research/ccm/
5. Cloud Security Alliance: CloudTrust Protocol (2016). https://cloudsecurityalliance.org/research/ctp
6. Distributed Management Task Force, Inc. (DMTF): Cloud auditing data federation (CADF) - data format and interface definitions specification (2014). http://www.dmtf.org/sites/default/files/standards/documents/DSP0262_1.0.0.pdf
7. Doelitzscher, F., Rübsamen, T., Karbe, T., Reich, C., Clarke, N.: Sun behind clouds - on automatic cloud security audits and a cloud audit policy language. Int. J. Adv. Netw. Serv. **6**(1&2) (2013)
8. Kertesz, A., Kecskemeti, G., Oriol, M., Kotcauer, P., Acs, S., Rodríguez, M., Mercè, O., Marosi, A., Marco, J., Franch, X.: Enhancing federated cloud management with an integrated service monitoring approach. J. Grid Comput. **11**(4), 699–720 (2013). http://dx.doi.org/10.1007/s10723-013-9269-0
9. Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., Leaf, D.: Nist cloud computing reference architecture (2011). http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505
10. Massonet, P., Naqvi, S., Ponsard, C., Latanicki, J., Rochwerger, B., Villari, M.: A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In: 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), pp. 1510–1517, May 2011
11. Microsoft Developer Network: The Stride Threat Model (2014). https://msdn.microsoft.com/en-US/library/ee823878(v=cs.20).aspx

12. Muller, C., Oriol, M., Rodriguez, M., Franch, X., Marco, J., Resinas, M., Ruiz-Cortes, A.: Salmonada: a platform for monitoring and explaining violations of WS-agreement-compliant documents. In: 2012 ICSE Workshop on Principles of Engineering Service Oriented Systems (PESOS), pp. 43–49, June 2012
13. Povedano-Molina, J., Lopez-Vega, J.M., Lopez-Soler, J.M., Corradi, A., Foschini, L.: dargos: a highly adaptable and scalable monitoring architecture for multi-tenant clouds. Future Gener. Comput. Syst. **29**(8), 2041–2056 (2013). http://www.sciencedirect.com/science/article/pii/S0167739X13000824
14. Rizvi, S., Ryoo, J., Liu, Y., Zazworsky, D., Cappeta, A.: A centralized trust model approach for cloud computing. In: 2014 23rd Wireless and Optical Communication Conference (WOCC), pp. 1–6, May 2014
15. Rübsamen, T., Reich, C.: Supporting cloud accountability by collecting evidence using audit agents. In: 2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), vol. 1, pp. 185–190, December 2013
16. Rübsamen, T., Hölscher, D., Reich, C.: Towards auditing of cloud provider chains using cloudtrust protocol. In: Proceedings of the 6th International Conference on Cloud Computing and Service Science (CLOSER 2016), pp. 83–94. SciTePress (2016)
17. Rübsamen, T., Pulls, T., Reich, C.: Secure evidence collection and storage for cloud accountability audits. In: CLOSER 2015 - Proceedings of the 5th International Conference on Cloud Computing and Services Science, Lisbon, Portugal, 20–22 May 2015. SciTePress (2015)
18. Rübsamen, T., Reich, C.: An architecture for cloud accountability audits. In: Baden-Württemberg Center of Applied Research Symposium on Information and Communication Systems, SInCom 2014 (2014)
19. Saleh, M.: Construction of agent-based trust in cloud infrastructure. In: 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC), pp. 941–946, December 2014
20. Scientific Working Groups on Digital Evidence, Imaging Technology: SWGDE and SWGIT Digital and Multimedia Evidence Glossary (2015). https://www.swgde.org/documents/Current%20Documents/2015-05-27%20SWGDE-SWGIT%20Glossary%20v2.8