Evidence Collection in Cloud Provider Chains

Thomas Rübsamen[1], Christoph Reich[1], Nathan Clarke[2] and Martin Knahl[3]
[1]*Institute for Cloud Computing and IT Security, Furtwangen University, Robert-Gerwig-Platz 1, Furtwangen, Germany*
[2]*Centre for Security, Communications and Network Research, Plymouth University, Portland Square, Plymouth, U.K.*
[3]*Furtwangen University, Robert-Gerwig-Platz 1, Furtwangen, Germany*

Keywords: Cloud Computing, Audit, Federated Cloud, Security, Digital Evidence

Abstract: With the increasing importance of cloud computing, compliance concerns get into the focus of businesses more often. Furthermore, businesses still consider security and privacy related issues to be the most prominent inhibitors for an even more widespread adoption of cloud computing services. Several frameworks try to address these concerns by building comprehensive guidelines for security controls for the use of cloud services. However, assurance of the correct and effective implementation of such 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. Cloud auditing becomes increasingly challenging for the auditor the more complex the cloud service provision chain becomes. There are many examples for 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 as platform as a service (PaaS) or infrastructure as a service (IaaS) providers. The collection of data (evidence) for the assessment of policy compliance during a technical audit is aggravated the more complex the combination of cloud providers becomes. Nevertheless, the collection at all participating providers is required to assess policy compliance in the whole chain. The main contribution of this paper is an analysis of potential ways of collecting evidence in an automated way across cloud provider boundaries to facilitate cloud audits. Furthermore, a way of integrating the most suitable approaches in the system for automated evidence collection and auditing is proposed.

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 (Cloud Security Alliance, 2013). In Germany, a preference for cloud providers that fall under German jurisdiction and also run their own data centers in Germany or at least inside the European Union can be observed recently (Bitkom Research GmbH, 2015). 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 (Rübsamen and Reich, 2013; Rübsamen et al., 2013; Rübsamen and Reich, 2014; Rübsamen et al., 2015).

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

Rübsamen, T., Reich, C., Clarke, N. and Knahl, M.
Evidence Collection in Cloud Provider Chains.
In *Proceedings of the 6th International Conference on Cloud Computing and Services Science (CLOSER 2016) - Volume 1*, pages 59-70
ISBN: 978-989-758-182-3

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 Section 2, related research projects and industrial approaches are discussed. Following that, in Section 3 the authors elaborate on the definition and properties of cloud provider chains and auditing. Afterwards, a discussion of three different approaches to evidence collection for provider chain auditing in Section 4 is presented. In Section 5, the architectural integration of the approaches in a system for automating cloud audits is presented and evaluated for their effectiveness using a fictitious scenario. Section 6 concludes this paper.

2 RELATED WORK

Standards and catalogues such as ISO27001 (ISO, 2005), Control Objectives for Information and Related Technology (COBIT) (Information Systems Audit and Control Association, 2012) or NIST 800-53 (National Institute of Standards and Technology), 2013) define information security controls. A major part of these frameworks is auditing, both in regular auditing as an control itself and by using audits to ensure the correct and effective implementation of the controls. They are typically generic and target information systems in general and do not address the specifics of cloud computing.

There are some extensions to the previous frameworks such as the Cloud Controls Matrix (Cloud Security Alliance, 2014). It aims at the integration of aspects from ISO and COBIT, and NIST's more cloud-focused security and privacy protection recommendations 800-144 (National Institute of Standards and Technology, 2011), as well as domain-specific frameworks such as PCI-DSS (PCI Security Standards Council, 2015) or FedRAMP (U.S. General Services Administration, 2014), into a common controls framework for cloud computing that facilitates the risk assessment of using cloud services. CSA's Security, Trust & Assurance Registry (Cloud Security Alliance, 2015) enables comparison of cloud providers based on self-certification of cloud providers using the Cloud Controls Matrix. 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.

Monitoring systems provide similar functionality to audit systems with respect to the collection of data and synthesizing metrics that are compared against defined thresholds. There are several solutions for IT monitoring such as Nagios (Nagios Enterprises, LLC, 2014) or Ganglia (Ganglia, 2015) and several big commercial solutions. However, they often have a very distinct heritage in data center, cluster and grid monitoring and are therefore not necessarily suitable for the cloud due its dynamic infrastructure and potential chaining of cloud providers. More specialized monitoring systems such as Amazon's CloudWatch (Amazon Web Services, 2016) or Rackspace's monitoring (Rackspace, 2016) are naturally proprietary and do not support chaining outside of the providers own set of services. The integration of an evidence collection system with such widely used monitoring systems is of great importance, since they provide deep insight into cloud services and therefore are considered valuable sources of evidence.

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-Molina et al. (2013) 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. They utilize monitor agents to gather information for their centralized collector node. Katsaros et al. (2012) describe a similar approach to cloud monitoring with virtual machine units (VMU) that contain data collectors (scripts). Their focus is on self-adaptation of the monitoring system by adjusting monitoring intervals and other parameters. 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. (2011) 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. The actual infrastructure layer is out of scope. Also, opposed to our ap-

proach, monitoring probes (data collectors) do not have a way to be dynamically deployed where needed, but rather are included in the VEE on deployment time.

Kertesz et al. (2013) 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 (Muller et al., 2012). 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.

Montes et al. (2013) introduce an important aspect to cloud monitoring by also including the client-side in the data collection in addition to the cloud provider. However, they do not consider integrating third-party cloud providers as well.

Xie and Gamble (2012, 2013); Xie et al. (2014) describe an approach to inter-cloud auditing on the web service level, where audit assets are requested from a federated service.

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 (Tom Cook, 2015).

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 (characterized by being either IaaS, PaaS or SaaS providers) are involved in the provision of a service to a cloud consumer (who can be an individual or business).
2. One of the cloud providers acts as a primary service provider to the cloud consumer.
3. Subsequent cloud providers do not have a direct relationship with the cloud consumer.
4. The primary service provider *must* be and the subsequent providers *can* be cloud consumers themselves, if they use services provided by other cloud providers.
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 (Liu et al., 2011).

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 Figure 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.

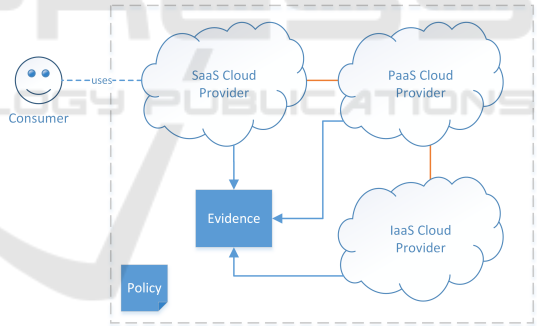

Figure 1: Cloud Provider Chains for Service Provision.

3.1 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 (Rübsamen and Reich, 2013).

In general, we follow the definition of digital evidence that is "information of probative value that

is stored or transmitted in binary form" (Scientific Working Groups on Digital Evidence and Imaging Technology, 2015). 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 (Cloud Security Alliance, 2016) that aims to provide a well-defined API that enables cloud providers to export transparency-enhancing information to auditors and cloud consumers. In this approach, the technological heterogeneity on the architectural level of the system is addressed by ensuring flexibility and extensibility and enabling the easy development of adapters for different evidence sources.

3.2 Audit Frameworks

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. Typically, these documents are not machine-readable and are geared towards limiting liability of the provider.

Additionally, there are well-known standards, frameworks and industry best practices, which define various aspects of how data handling and protection should be implemented in practice. Such frameworks are for instance the well-known ISO27001 for information security management in general, COBIT for IT governance and CSA's Cloud Controls Matrix (CCM) (Cloud Security Alliance, 2014) for cloud-specific risk assessment. However, requirements and obligations stated by these frameworks are typically not available in a machine-readable format. There are approaches to making these requirements and obligations explicit in a machine-readable way, for example Accountability Primelife Policy Language (Azraoui et al., 2014) for defining data protection and data handling-related obligations for data processing in the cloud.

Traditionally, policy compliance is evaluated using audits and asserted with a certification of compliance (e.g., ISO27001 compliance certification). Typically, the intervals in which an audit is repeated are quite long (often yearly or longer). In the meantime, policy violations can potentially remain undetected for extended periods of time. One of our main goals is to address these periods of uncertainty by enabling the continuous assessment of cloud operations with respect to policy compliance. This is an important step towards continuous certification.

3.3 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" (Liu et al., 2011). 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.

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) Azraoui et al. (2015).

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 Figure 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.

4 APPROACHES FOR COLLECTING EVIDENCE IN CLOUD PROVIDER CHAIN AUDITS

There are several approaches available when it comes to collecting evidence for audit purposes in a service provider chain. These approaches 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.

Implementation details of these components are discussed in our previous work. The following discussion focuses on the different approaches to extend the system for cloud provider chains.

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.

4.1 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 or Rackspace, already provide improved transparency over their cloud operations by providing their customers with access to proprietary monitoring and logging APIs (see (Amazon Web Services, 2016; Rackspace, 2016)). The extent to which data is shared is typically limited to information that is already produced by the clod 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).

Figure 2 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.

4.1.1 Level of Auditor Control

a 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

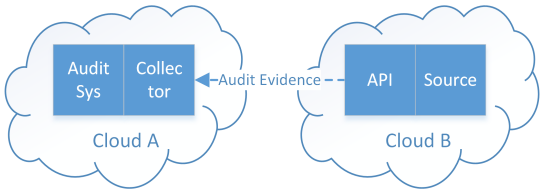

Figure 2: Remote API Evidence Collector.

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.

4.1.2 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.

4.1.3 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.

4.2 Provider Provisioned Evidence Collector

In this approach, the audit system still is the main component for evidence collection. Here, 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 Figure 3) 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.

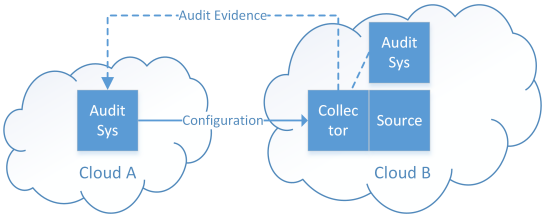

Figure 3: Provider-provisioned Evidence Collector.

4.2.1 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).

4.2.2 Technical Limitations

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

4.2.3 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.

4.3 Mobile Evidence Collector

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) JADE (2014) for implementing collectors. The migration of collectors between providers is thereby performed by using JADE's mobile agent capabilities.

As depicted in Figure 4, 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.

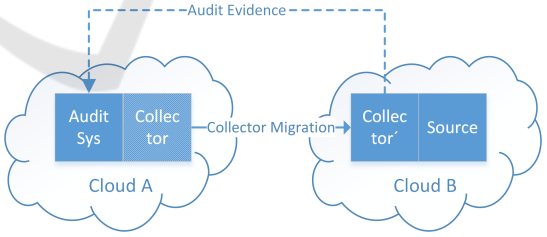

Figure 4: Mobile Evidence Collector.

4.3.1 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.

4.3.2 Technical Limitations

Since the auditor knows most about the actual con-

figuration 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.

4.3.3 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.

4.4 Round-up

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.

5 SCENARIO-BASED PROVIDER CHAIN AUDITING EVALUATION

In the previous Section 4, 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 *Mobile Evidence Collector* approaches (see Section 4.1 and 4.3 respectively). The approach is validated by discussing a fictitious use case.

5.1 Audit Agent System

In Figure 5, 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, as described in Section 4).

5.2 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 Section 4) 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 Figure 5, 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 Section 3.3. 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 evidence is collected by the mobile evidence collectors as part of building an evidence trail for compliance evaluation at A1.

1. The data retention event is recorded as evidence by the collector running at A1.
2. The delete action of the database is recorded as evidence 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.

5.3 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:

1. Pre-processing: 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.
2. Intermediate Result Production: A second preprocessing 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 (where an audit result can be produced by combining several intermediate results).

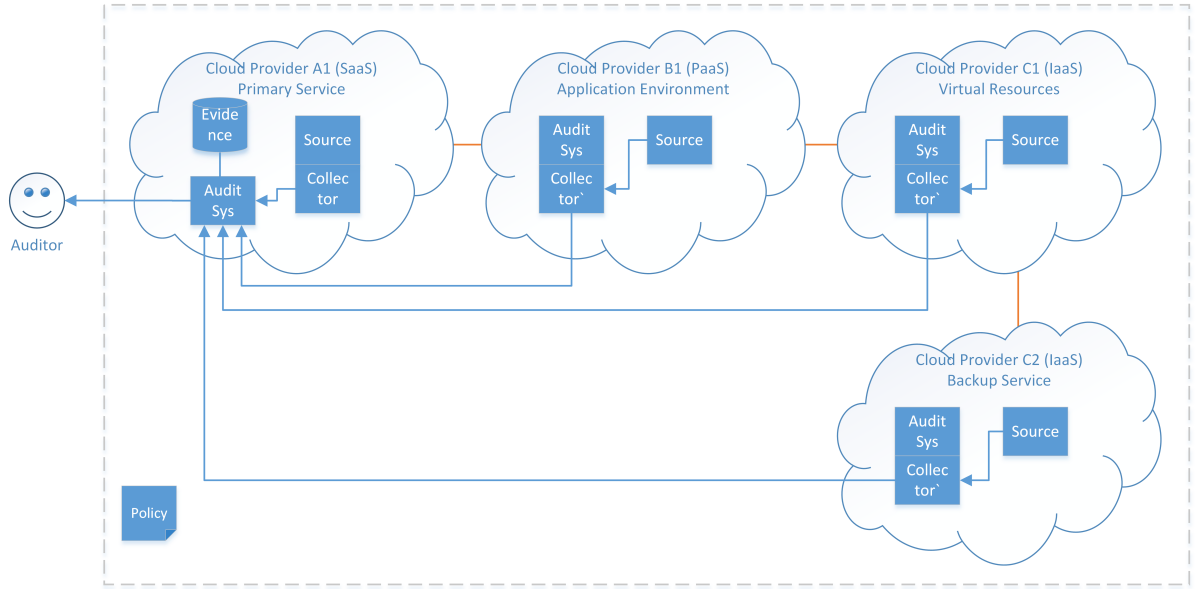

Figure 5: Provider Chain Auditing Architecture.

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.

6 CONCLUSIONS

Cloud auditing is becoming increasingly important as cloud adoption increases and compliance of data processing is put into focus of the cloud consumer. The key to making cloud audits a useful tool is the effectiveness of collection process that is used to build the basis for the evaluation of policy compliance or lack thereof.

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. As long as there is only one cloud provider involved in service provisioning to the cloud consumer, monitoring and auditing is relatively simple (with the above mentioned restrictions). However, in more complex scenarios where there are chains of providers (or federations of cloud providers), current approaches are severely limited.

In this paper an extension to our previous work on automating continuous cloud audits that enables the collection of evidence across the boundaries of multiple cloud providers in a cloud provider chain was presented. The concept of cloud provider chains and three different approaches to evidence collection with their advantages and disadvantages were discussed. Furthermore, 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.

In the future, the analysis of the different approaches and their integration in our system will be expanded in two main areas: i) expanding the security mechanisms that are already present to account for the notion of provider chains and ii) demonstrating the scalability and efficiency of the system.

ACKNOWLEDGEMENTS

This work has been partly funded from the European Commission's Seventh Framework Programme (FP7/2007-2013), grant agreement 317550, Cloud Accountability Project - http://www.a4cloud.eu/ - (A4CLOUD).

REFERENCES

Amazon Web Services (2016). Amazon cloudwatch. https://aws.amazon.com/de/cloudwatch/.

Azraoui, M., Elkhiyaoui, K., Önen, M., Bernsmed, K., De Oliveira, A., and Sendor, J. (2015). A-ppl: An accountability policy language. In Garcia-Alfaro, J., Herrera-Joancomartí, J., Lupu, E., Posegga, J., Aldini, A., Martinelli, F., and Suri, N., editors, *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance*, volume 8872 of *Lecture Notes in Computer Science*, pages 319–326. Springer International Publishing.

Azraoui, M., Elkhiyaoui, K., Önen, M., Bernsmed, K., Santana De Oliveira, A., and Sendor, J. (2014). A-PPL: An accountability policy language. In *DPM 2014, 9th International Workshop on Data Privacy Management, September 10, 2014, Wroclaw, Poland*, Wroclaw, POLAND.

Bitkom Research GmbH (2015). Cloud Monitor 2015. https://www.kpmg.com/DE/de/Documents/cloudmonitor%202015_copyright%20_sec_neu.pdf.

Cloud Security Alliance (2013). Top threats to cloud computing survey results update 2012. https://downloads.cloudsecurityalliance.org/initiatives/top_threats/Top_Threats_Cloud_Computing_Survey_2012.pdf.

Cloud Security Alliance (2014). Cloud Controls Matrix. https://cloudsecurityalliance.org/research/ccm/.

Cloud Security Alliance (2015). Security, Trust & Assurance Registry. https://cloudsecurityalliance.org/star/.

Cloud Security Alliance (2016). Cloud Trust Protocol. https://cloudsecurityalliance.org/research/ctp.

Ganglia (2015). Ganglia. http://ganglia.sourceforge.net/.

Information Systems Audit and Control Association (2012). Control Objectives for Information and Related Technology (COBIT) 5. http://www.isaca.org/cobit/.

ISO (2005). ISO27001:2005. http://www.iso.org/iso/catalogue_detail?csnumber=42103.

JADE (2014). Java Agent DEvelopement framework. http://jade.tilab.com.

Katsaros, G., Kousiouris, G., Gogouvitis, S. V., Kyriazis, D., Menychtas, A., and Varvarigou, T. (2012). A self-adaptive hierarchical monitoring mechanism for clouds. *Journal of Systems and Software*, 85(5):1029 – 1041.

Kertesz, A., Kecskemeti, G., Oriol, M., Kotcauer, P., Acs, S., Rodríguez, M., Mercè, O., Marosi, A., Marco, J., and Franch, X. (2013). Enhancing federated cloud management with an integrated service monitoring approach. *Journal of Grid Computing*, 11(4):699–720.

Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., and Leaf, D. (2011). Nist cloud computing reference architecture. http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505.

Massonet, P., Naqvi, S., Ponsard, C., Latanicki, J., Rochwerger, B., and Villari, M. (2011). A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, pages 1510–1517.

Montes, J., Sánchez, A., Memishi, B., Pérez, M. S., and Antoniu, G. (2013). Gmone: A complete approach to cloud monitoring. *Future Generation Computer Systems*, 29(8):2026 – 2040.

Muller, C., Oriol, M., Rodriguez, M., Franch, X., Marco, J., Resinas, M., and Ruiz-Cortes, A. (2012). Salmonada: A platform for monitoring and explaining violations of ws-agreement-compliant documents. In *Principles of Engineering Service Oriented Systems (PESOS), 2012 ICSE Workshop on*, pages 43–49.

Nagios Enterprises, LLC (2014). Nagios. http://www.nagios.org/.

National Institute of Standards and Technology (2011). Guidelines on security and privacy in public cloud computing. http://csrc.nist.gov/publications/nistpubs/800-144/SP800-144.pdf.

National Institute of Standards and Technology) (2013). Security and privacy controls for federal information systems and organizations. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf.

PCI Security Standards Council (2015). Payment Card Industry Data Security Standard (PCI-DSS). https://www.pcisecuritystandards.org/.

Povedano-Molina, J., Lopez-Vega, J. M., Lopez-Soler, J. M., Corradi, A., and Foschini, L. (2013). Dargos: A highly adaptable and scalable monitoring architecture for multi-tenant clouds. *Future Generation Computer Systems*, 29(8):2041 – 2056.

Rackspace (2016). Rackspace cloud monitoring. http://www.rackspace.com/cloud/monitoring.

Rübsamen, T., Pulls, T., and Reich, C. (2015). 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, May 20 - 22, 2015*. SciTePress.

Rübsamen, T. and Reich, C. (2013). Supporting cloud accountability by collecting evidence using audit agents. In *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, volume 1, pages 185–190.

Rübsamen, T. and Reich, C. (2014). An Architecture for Cloud Accountability Audits. In *1. Baden-Württemberg Center of Applied Research Symposium on Information and Communication Systems SInCom 2014*.

Rübsamen, T., Reich, C., Wlodarczyk, T., and Rong, C. (2013). Evidence for accountable cloud computing services. http://dimacs.rutgers.edu/Workshops/TAFC/TAFC_a4cloud.pdf.

Scientific Working Groups on Digital Evidence and Imaging Technology (2015). SWGDE and SWGIT Digital & Multimedia Evidence Glossary. https://www.swgde.org/documents/Current%20Documents/2015-05-27%20SWGDE-SWGIT%20Glossary%20v2.8.

Tom Cook (2015). Dropbox at AWS re:Invent 2014. https://blogs.dropbox.com/tech/2014/12/aws-reinvent-2014/.

U.S. General Services Administration (2014). Federal Risk and Authorization Program. http://www.fedramp.gov.

Xie, R. and Gamble, R. (2012). A tiered strategy for auditing in the cloud. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 945–946.

Xie, R. and Gamble, R. (2013). An architecture for cross-cloud auditing. In *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, CSIIRW '13, pages 4:1–4:4, New York, NY, USA. ACM.

Xie, R., Gamble, R., and Ahmed, N. (2014). Diagnosing vulnerability patterns in cloud audit logs. In Han, K. J., Choi, B.-Y., and Song, S., editors, *High Performance Cloud Auditing and Applications*, pages 119–146. Springer New York.