# TrustCloud: A Framework for Accountability and Trust in Cloud Computing

Ryan K L Ko, Peter Jagadpramana, Miranda Mowbray, Siani Pearson, Markus Kirchberg, Qianhui Liang, Bu Sung Lee

HP Laboratories
HPL-2011-38

**Keyword(s):**

trust in cloud computing, logging, auditability, accountability, data provenance, continuous auditing and monitoring, governance

**Abstract:**

The key barrier to widespread uptake of cloud computing is the lack of trust in clouds by potential customers. While preventive controls for security and privacy measures are actively being researched, there is still little focus on detective controls related to cloud accountability and auditability. The complexity resulting from the sheer amount of virtualization and data distribution carried out in current clouds has also revealed an urgent need for research in cloud accountability, as has the shift in focus of customer concerns from server health and utilization to the integrity and safety of end-users' data. This paper discusses key challenges in achieving a trusted cloud through the use of detective controls, and presents the TrustCloud framework, which addresses accountability in cloud computing via technical and policy-based approaches.

External Posting Date: June 22, 2011 [Fulltext]. Approved for External Publication

Internal Posting Date: March 21, 2011 [Fulltext]

Additional Publication Information: To be published and presented at the 2nd IEEE Cloud Forum for Practitioners (IEEE ICFP 2011), Washington DC, USA, July 7-8, 2011.

# TrustCloud: A Framework for Accountability and Trust in Cloud Computing

Ryan K L Ko [1], Peter Jagadpramana [1], Miranda Mowbray [2], Siani Pearson [2],
Markus Kirchberg [1], Qianhui Liang [1], Bu Sung Lee [1]

[1] Cloud & Security Lab
Hewlett-Packard Laboratories
Fusionopolis, Singapore
{ryan.ko | peter.jagadpramana | markus.kirchberg |
qianhui.liang | francis.lee}@hp.com

[2] Cloud & Security Lab
Hewlett-Packard Laboratories
Bristol, United Kingdom
{miranda.mowbray | siani.pearson}@hp.com

***Abstract*— The key barrier to widespread uptake of cloud computing is the lack of trust in clouds by potential customers. While preventive controls for security and privacy measures are actively being researched, there is still little focus on detective controls related to cloud accountability and auditability. The complexity resulting from the sheer amount of virtualization and data distribution carried out in current clouds has also revealed an urgent need for research in cloud accountability, as has the shift in focus of customer concerns from server health and utilization to the integrity and safety of end-users' data. This paper discusses key challenges in achieving a trusted cloud through the use of detective controls, and presents the TrustCloud framework, which addresses accountability in cloud computing via technical and policy-based approaches.**

***Keywords- trust in cloud computing, logging, auditability, accountability, data provenance, continuous auditing and monitoring, governance.***

## I. INTRODUCTION

Cloud computing requires companies and individuals to transfer some or all control of computing resources to cloud service providers (CSPs). Such transfers naturally pose concerns for company decision makers. In a recent 2010 survey by Fujitsu Research Institute [1] on potential cloud customers, it was found that 88% of potential cloud consumers are worried about *who* has access to their data, and demanded more awareness of what goes on in the backend physical server. Such surveys demonstrate the urgency for practitioners and researchers to quickly address obstacles to trust.

While risks can be greatly mitigated via preventive controls for privacy and security (e.g. encryption, access control based on ID profiling, etc), they are not enough. There is a need to complement such measures with equally important measures that promote transparency, governance and accountability of the CSPs. This was also identified by The European Network and Information Security Agency (ENISA)'s cloud computing risk assessment report [2], which states that the 'loss of governance' as one of the top risks of cloud computing, especially Infrastructures as a Service (IaaS).

Despite auditability being a crucial component of improving trust, current prominent providers *(e.g. Amazon EC2/ S3 [3, 4], Microsoft Azure [5])* are still not providing full transparency and capabilities for the tracking and auditing of the file access history and data provenance [6] of both the physical and virtual servers utilized [1]. Currently, users can at best monitor the virtual hardware performance metrics and the system event logs of the services they engage. The cloud computing research community, particularly the Cloud Security Alliance, has recognized this. In its *Top Threats to Cloud Computing Report (Ver.1.0)* [7], it listed seven top threats to cloud computing:

1. Abuse and nefarious use of cloud computing
2. Insecure application programming interfaces
3. Malicious insiders
4. Shared technology vulnerabilities
5. Data loss or leakages
6. Account, service and traffic hijacking
7. Unknown risk profile.

Methods increasing the accountability and auditability of CSPs, such as the tracking of file access histories, will empower service providers and users to reduce five of the above seven threats: 1,2,3,5 and 7. As such, this paper identifies trust, *via* the addressing of accountability and auditability, as an urgent research area in cloud computing.

## II. TRUST IN CLOUD COMPUTING

While there is no universally accepted definition of trust in cloud computing, it is important to clarify its components and meaning. In dictionaries, *trust* is generally related to *"levels of confidence in something or someone"* [8, 9]. Hence we can view trust in the cloud as the customers' **level of confidence in using the cloud**, and try to increase this by mitigating technical and psychological barriers to using cloud services. For more analysis of the definitions of trust in cloud computing, see [10].

### A. Components of Trust in Cloud Computing

To best mitigate barriers to confidence, we need to understand the main components affecting cloud trust:

*1)* ***Security*** *[11, 12]* **-** Mechanisms (e.g. encryption) which make it extremely difficult or uneconomical for an unauthorised person to access some information.

*2)* ***Privacy*** *[13, 14]* **-** Protection against the exposure or leakage of personal or confidential data (e.g. personally identifiable information (PII)).

*3)* ***Accountability*** *[14, 15]* - Defined in [16] as "the obligation and/ or willingness to demonstrate and take

responsibility for performance in light of agreed-upon expectations", accountability goes beyond responsibility by obligating an organization to be answerable for its actions. Accountability has been established in guidance by organizations such as OECD, APEC, PIPEDA as placing a legal responsibility upon an organisation that uses personally identifiable information (PII) to ensure that contracted partners to whom it supplies the PII are compliant to privacy guidelines, wherever in the world they may be.

*4)* ***Auditability [2]*** **–** The relative ease of auditing a system or an environment. Poor auditability means that the system has poorly-maintained (or non-existent) records and systems that enable efficient auditing of processes within the cloud. Auditability is also an enabler of (retrospective) accountability: It allows an action to be reviewed against a pre-determined policy to decide if the action was compliant, and if it was not, to hold accountable the person or organization responsible for the action.

### *B. Preventive versus Detective Controls*

Trust components can be also classified as ***Preventive Controls*** or ***Detective Controls***.

Preventive controls are used to mitigate the occurrence of an action from continuing or taking place at all (e.g. an access list that governs who may read or modify a file or database, or network and host firewalls that block all but allowable activity).

Detective controls are used to identify the occurrence of a privacy or security risk that goes against the privacy or security policies and procedures (for example, an intrusion detection system on a host or network, or security audit trails, logs and analysis tools).

In addition, there are corrective controls, (e.g. an incident management plan) which are used to fix an undesired result that has already occurred. This paper focuses on detective controls for cloud computing.

Despite the lack of direct ability to stop irregularities from occurring, these controls are very important. They act as psychological obstacles to go against policies or procedures in the cloud, and also serves as a record for post-mortem investigations should any non-compliance occur. They act as in a similar way as speed cameras do for traffic control: the presence of speed cameras will deter law-abiding citizens from speeding, but their presence cannot prevent speeding from taking place. Detective controls hence complement preventive controls. A combination of the two is usually required for reasonable protection.

## III. COMPLEXITIES INTRODUCED IN CLOUD COMPUTING

Compared to traditional server architectures, the focus of monitoring and accountability now shifts from a server-health perspective to a user's data perspective. Companies who change from carrying out their computing in-house to using the public cloud are no longer concerned about the health of servers (since they no longer own or maintain them); they are more concerned about the integrity and safety of their data. However, with cloud computing's promise of elasticity empowered by virtualization [3, 17], comes several new complexities introduced into the area of accountability.

### *A. Challenges Introduced by Virtualisation*

*1) Tracking of virtual-to-physical mapping and vice versa*

The use of virtualization by CSPs allows them to use their server resources to be used more efficiently, and to adapt to peaks and troughs in individual users' computation and bandwidth requirements. However, the addition of virtualized layers also means that accountability might require the identification not only of the virtual server in which an events takes place,, but also the physical server.

Currently, there are only tools (e.g. HyTrust [18]) which are able to log virtual level logs and system health monitoring tools for virtual machines. There is still a lack of transparency of (1) the linkages between the virtual and physical operating systems, (2) relationships between virtual locations and physical static server locations, and (3) how the files are written into both virtual and physical memory addresses. These information are currently not available as a single-point-of-view for the customers.

*2) Multiple operating system environments to track*

Many different operating systems are available for virtual machines, and this potentially introduces the need to manage the logging of machines in the cloud which use a large number of different operating systems. Enforcing a single operating system for all virtual machines would solve this issue, but it would make the provider less competitive.

### *B. Logging from Operating System Perspective versus Logging from File-Centric Perspective*

Current tools focus on operating systems and system health monitoring (e.g. cloudstatus.com, [19], etc), but few emphasize the *file-centric perspective*. By the *file-centric perspective*, we mean that we need to trace data and files from the time they are created to the time they are destroyed. When we log from a file-centric perspective, we view data and information independent from the environmental constraints. This is reflective of the very elastic nature of cloud computing. With the transfer of control of data into the providers, the providers have the mandate to ease the minds of consumers by providing them the capabilities to track their data.

### *C. Scale, Scope and Size of logging*

The elasticity of cloud computing also increases the need for efficient logging techniques and a proper definition of scope and scale of logging. By efficient, we mean that the impending exponential increase in log size has to be manageable, and not quickly wipe out memory of servers hosting the cloud logging features. Detailed logs may reveal information that is private or sensitive, and there need to be adequate controls on *who* gets access to this information, and for *what* purposes. Thus the scope and scale of logging may need to be limited for reasons of security and privacy as well as for manageability. To scale and scope, we need policies that can help to clearly define the areas which loggers are assigned to log in. For example, a service provider may label its own network as a *safe zone*, while its suppliers or mirror sites *trusted zones*, and any other network outside of these are labeled as *unsafe zones*. Zonal planning will greatly reduce the complexities of network data transfer tracing within a cloud. Another way of reducing complexity will be the classification of data abstraction layers, e.g. crude data,

documents, and on a higher level, workflows. These are discussed further in Section V.

### D. Live and Dynamic Systems

While there are proposals for adoption of provenance-aware mechanisms (that allow tracing back the source or creator of data) in cloud computing, such proposals are unable to address all challenges in clouds, as cloud systems are live and dynamic in nature. Provenance techniques propose reports (e.g. audit trails) as the key to forensic investigations. However in reality, a snapshot of a running, or "live" system such as the VMs turned on within a cloud can be only reproduced up to its specific instance and cannot be reproduced in a later time-frame. As a result, with a live system, data from a probe up to one instance will be different from data from another probe say 15 minutes into the live system [20]. This means cloud accountability demands complex ***real-time*** accountability, where key suspected events are captured almost instantaneously.

Our team is currently working on research in these challenges through the framework presented later in Section V. It is important to note that the list of complexities mentioned in this section is not exhaustive. The reader may wish to refer to [10] for further discussions of complexities of cloud computing that may affect trust.

## IV. The Cloud Accountability Life Cycle


Figure 1. The Cloud Accountability Life Cycle (CALC)

The discussions in Section III show not only the scale and urgency of achieving cloud accountability but also exposed the need for reduction of complexity. Having an awareness of the key accountability phases will not only simplify the problem, but also allow tool makers and their customers to gauge the comprehensiveness of tools (i.e. whether there are any phases not covered by a tool). A classification of the different phases may also help researchers to focus on specific research sub-problems of the large cloud accountability problem. These phases are collectively known as the **Cloud Accountability Life Cycle** (CALC) [21], which consists of the following seven phases (see Figure 1):

*1) Policy Planning*

CSPs have to decide what information to log and which events to log on-the-fly. It is not the focus of this paper to claim or provide an exhaustive list of recommended data to be logged. However, in our observation, there are generally four important groups of data that must be logged: (1) Event data – a sequence of activities and relevant information, (2) Actor Data – the person or computer component (e.g. worm) which trigger the event, (3) Timestamp Data – the time and date the event took place, and (4) Location Data – both virtual and physical (network, memory, etc) server addresses at which the event took place.

*2) Sense and Trace*

The main aim of this phase is to act as a sensor and to trigger logging whenever an expected phenomenon occurs in the CSP's cloud (in real time). Accountability tools need to be able to track from the lowest-level system read/write calls all the way to the irregularities of high-level workflows hosted in virtual machines in disparate physical servers and locations. Also, there is a need to trace the routes of the network packets within the cloud [22].

*3) Logging*

File-centric perspective logging is performed on **both** virtual and physical layers in the cloud. Considerations include the lifespan of the logs within the cloud, the detail of data to be logged and the location of storage of the logs. . It may in some cases be necessary to pseudonymise or anonymize private data before it is recorded in logs.

*4) Safe-keeping of Logs*

After logging is done, we need to protect the integrity of the logs to prevent unauthorized access and ensure that they are tamper-free. Encryption may be applied to protect the logs. There should also be mechanisms to ensure proper backing up of logs and prevent loss or corruption of logs. Pseudonymisation of sensitive data within the logs may in some cases be appropriate.

*5) Reporting and Replaying*

Reporting tools generate from logs file-centric summaries and reports of the audit trails, access history of files and the life cycle of files in the cloud. Suspected irregularities are also flagged to the end-user. Reports may cover a large scope, for example recording virtual and physical server histories within the cloud; from OS-level read/write operations of sensitive data, or high-level workflow audit trails.

*6) Auditing*

Logs and reports are checked and potential irregularities highlighted. The checking can be performed by auditors or stakeholders. If automated, the process of auditing will become 'enforcement'. Automated enforcement is very feasible for the massive cloud environment, enabling cloud system administrators to detect irregularities more efficiently.

*7) Optimising and Rectifying*

Problem areas and security loopholes in the cloud are removed or rectified and control and governance of the cloud processes are improved.

## V. The TrustCloud Framework

Our team is currently focusing on addressing accountability in the cloud from all aspects, via five layers in the TrustCloud framework, keeping in mind CALC's phases.

### A. TrustCloud Accountability Abstraction Layers

Since log types range from a *system-level* log to a *workflow-level* audit trail transactional log, there needs to be a clear definition of abstraction layers to reduce ambiguity and increase research focus and impact. We propose the TrustCloud framework, which consists of the following **layers of accountability**:

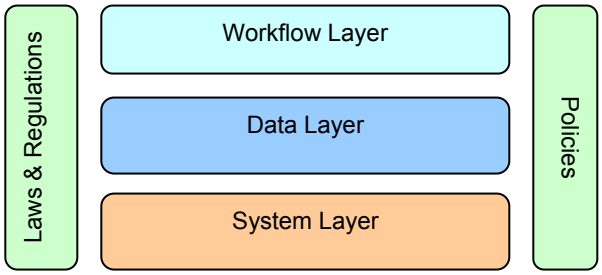

Figure 2. Abstraction Layers of Accountability in Cloud Computing

Figure 2 shows the abstraction layers for the type of logs needed for an accountable cloud. It extends the layers in our previous work [21], which stipulated three basic layers: workflow, data and system layers. It is important to note that the focus is on the abstraction layers of logs and not on architectural layers. Hence, the TrustCloud framework is independent of virtual or physical environments. The data and workflow abstraction layers are derived from related works in business process management [23], and data provenance [6, 24, 25] respectively, while the system layer is derived from related works in trusted computing platforms [26, 27] and system logging literature [28, 29].

Such explicit definition of layers allows us to efficiently identify the areas of their application and their focus areas. At a glance, the five layers look deceptively simple, but the problem is more complex than it looks. Each layer has a slightly different set of sub-components for each different context. Our model simplifies the problem and makes accountability more achievable. The usefulness of abstraction layers is also analogous to OSI [30] and TCP/IP [31] networking layers.

Let us now discuss the research issues, scope and scale of each layer in the TrustCloud framework:

### *B. System Layer*

The lowest TrustCloud layer is the system layer. The system layer performs file-centric logging within the following three components:

#### *1) Operating Systems (OS)*

OS system and event logs are the most common type of logs associated with cloud computing at the moment. However, these logs are not the main contributing factor to accountability of **data** in the cloud, but a supporting factor. This is because in traditional physical server environments housed within companies, the emphasis was on health and feedback on system status and ensuring uptime as server resources are limited and expensive to maintain. In cloud computing, resources like servers and memory are '*elastic*', and are no longer limited or expensive [3, 17]. Hence, OS logs, while important, are no longer the top concern of customers. Instead, the customers are more concerned about the integrity, security and management of their data stored in the cloud [1, 32].

#### *2) File Systems*

Even though the file system is technically part of the OS, we explicitly include it as a major component in a file-centric system layer. This is because, in order to know, trace and record the exact file life cycles, we often have to track system read/write calls to the file system. From the system read/write calls, we can also extract the files' virtual and physical memory locations, providing more information for further forensics. The file-centric perspective [33] is also the area which is less emphasized by current tools.

#### *3) Cloud's Internal Network*

As clouds are vast networks of physical and virtual servers over a large number of locations, we need to also monitor network logs within the cloud. Network logs [34, 35] are logs specific to data being sent and received over the network.

Our team is currently working on a technique which performs logging over the three above-mentioned components, and the tracing and logging of files' life cycles (i.e. *creation, modification, duplication, destruction*) within clouds.

#### *4) Why System Layer?*

One of the key problems of cloud computing environment is the "replay" of a snapshot, i.e. a reproduction of the exact state of the cloud at a particular moment and the machines turned on and off at that instance. With a large number of virtual machines turned on and off at different time periods, and executing several business applications at the same time, it is very difficult to replay the exact same snapshot of the Cloud from the past, e.g. 1 hour ago, so that one can track what actually went wrong [20]. There needs to be an effective and efficient method to do this, and our current work on a file-centric system layer logging mechanism across both virtual machines and physical machines fits into the role very well. Such system-layer mechanisms log the resources the VMs use and share when they are turned on. Evidently, such snapshots cannot be captured in the data and workflow layer as they are too high-level and dependent on the on and off status of their hosting machines. The only assured way to track the complete VM changes is actually to track the system layer of the Cloud.

### *C. Data Layer*

The data layer supports the data abstraction and facilitates data-centric logging through the following components:

#### *1) Provenance Logger*

In order to enable reasoning about the origins, collection or creation, evolution, and use of data, it is essential to track the history of data, i.e., its provenance. Provenance information has been described as *'the foundation for any reasonable model of privacy and trust*' in the context of the Semantic Web [36], and we believe it to be similarly central to trust in Cloud Computing. It enables validation of the processes involved in generating/obtaining the data and the detection of unusual behavior.

While these advantages are very promising, corresponding challenges are equally difficult to address/overcome. Common challenges include efficiently and effectively managing the sheer amount of provenance data that has to be maintained; ensuring consistency and completeness of provenance data; detecting malicious users who attempt to falsify provenance data; protecting data owner as well as data providers from exposing sensitive, confidential, proprietary or competitively important information indirectly through provenance logs; enabling efficient querying of provenance data; etc.

Considering past and current efforts, cloud computing-based provenance logging must fulfill the following criteria:

(1) be secure and privacy-aware (to ensure that the logs themselves cannot be tempered with or be a source for knowledge inference); (2) be (eventually) consistent and complete (similar to the ACID properties known from database transaction processing); (3) be transparent/non-invasive; (4) be scalable, e.g. avoid exponential explosion of provenance data through application of summarization techniques (5) be persistent over the long term; (6) allow for multiple tailored views (to permit access based on roles with different access privileges); and (7) be efficiently accessible.

*2) Consistency Logger*

While current cloud providers typically support a weaker notion of consistency, i.e., eventual consistency, it is important to have mechanisms to allow for rollback, recovery, replay, backup, and restoring of data. Such functionality is usually enabled by using operational and/or transactional logs, which assist with ensuring atomicity, consistency, and durability properties. Logs have also been proven useful for monitoring of operational anomalies. While these concepts are well established in the database domain, cloud computing's characteristics such as eventual consistency, "unlimited" scale, and multi-tenancy pose new challenges. In addition, secure and privacy-aware mechanisms must be devised not only for consistency logs but also for their backups, which are commonly used for media recovery.

### *D. Workflow Layer*

The workflow layer focuses on the audit trails and the audit-related data found in the software services in the cloud. High level fraudulent risks such as procurement approval routes, decision making flows and role management in software services run within the cloud has to be monitored and controlled. In a service oriented architecture [37], services from several sources are composed to perform higher-level, more complex business functions. The accountability of the services and their providers within the clouds also have to be managed.

TrustCloud's workflow layer aims to ensure proper governance of cloud applications, empower continuous auditing and manage the accountability of services composed as business processes via the following components:

*1) Governance in the cloud*

When cloud computing experiences an increase in uptake and usage, there will be mandated needs for the auditability, proper prevention and tracking of fraudulent activities, irregularities and control loopholes in the business processes in the cloud. At the workflow layer, the TrustCloud framework explores how clouds can achieve high auditability via compliance to regulations such as Sarbanes-Oxley (SOX) [38] and Health and Human Services Health Insurance Portability and Accountability Act (HIPAA) (e.g. Title II: Preventing Healthcare Fraud and Abuse) regulations [39], and/ or benchmarking against information security standards such as the ISO 27000 suite [40, 41].

*2) Automated Auditing*

With the promise of high performance computing power from cloud architectures, TrustCloud envisions the realization of automated auditing of financial and business process transactions in the cloud. Auditability is a prerequisite for such a step. However, achieving auditability via methods such as continuous auditing [42] within a highly virtualized environment is a very difficult and complex task. There needs to be considerations for not only the auditing of the business logic and control flows, but also the applications implementing them.

*3) Patch Management Auditing*

There is also a need for auditing of the management of virtual machine image bug fixes, patching and upgrades in a cloud environment [43, 44]. The scale of patching and deployment within the cloud environment is massive, and the associated logs need to be highly auditable for proper troubleshooting, playbacks and accountability of the technical staff performing these activities.

*4) Accountability of Services*

Accountability is also required in service oriented architectures in cloud environments. When composing services from existing service components, we also face the problem of trust. With cloud computing, service components can proliferate and their access is virtualized. This makes composition easier and practical. Meanwhile, the source of services may or may not be trustworthy, which presents a major problem in cloud computing. This can be explained using the following example.

Let us assume that we are developing a Web portal and we are designing this by integration of the services into a portal. Some of the services may be malicious (for example they manipulate data passing through). Therefore, the portal may or may not be a valid software and perform according to the expected design or according to the contractual agreement. In this scenario, the achievement of accountability of services can help us to investigate such scenarios.

We believe that the logging approach is also applicable to help achieve the accountability of services. Logging should take care of the following concerns on a component:

a) *Input or pre-processing*, whether the component takes in more than enough input to perform the required function. It is usually a sign of maliciousness if the input is more than what is needed. Additional information from the user may be used to do something undesirable.

b) *Processing*, whether the component is designed to actually do what is expected. Is there any extra and unexpected processing that has occurred during the production of the requested result?

c) *Post processing*, whether the component has deleted the input and the intermediary results of the processing. Proper actions need to keep the input and the whole processing confidential and no traces of processing should have been recorded.

Our logging solution should achieve the purpose of deterring the service component providers from making malicious components and encourage the proper behavior and execution of the components.

### *E. Policy, Law and Regulations*

Policies and laws require information to be logged on what data items are processed, accessed, stored or transmitted. They may also require information on why, when, where, how and by whom this processing takes place.

**What:** Data classification is important, as in general there will be different policies and legal rules affecting

different classes of data items. Classes to consider might include non-PII data, anonymised data, pseudonymised data, PII, sensitive PII, and Payment Card Industry (PCI)-regulated data. When a new data item is created (either by a user, or as the result of automated copying or processing of already-existing data) this creation may need to be logged together with the classification of the item and/or the policies associated with it. In addition to records about individual data items, there may be audit needs associated with higher-level information. For example policy changes should be recorded, and there may be audit requirements for process flows within and between organizations in the cloud.

**Why:** The OECD's Purpose Specification and Use Limitation principles legally restrict the use of PII/sensitive PII to purposes compatible with ones specified before or at the time of data collection, required by law, or consented to by the data subject. Therefore the purpose of a data processing action, and the purposes for which the processing of a given PII data item is permitted, may need to be recorded.

**When:** Logs usually include timestamps. Timing information is also necessary for compliance to laws and policies concerned with data retention: it is necessary to have a data retention and destruction plan for all data storage systems. (Data retention compliance also requires information to be kept on which records or data items are duplicates, and on the location of backup copies, to ensure that all copies of an item can be destroyed.) Timing considerations may also reduce the information that needs to be recorded, as transient data that is only stored for the purpose of the current transaction and then deleted has minimal privacy implications.

**Where:** Geographical location matters from a legal point of view – different laws may apply depending on where information exists, and there are some restrictions on trans-border data flows. It can be difficult to ascertain within the cloud where data is, and there may be multiple copies. So the physical location of storage and the occurrence of cross-border data transfers (for example to partners, sub-contractors, or cloud service providers) may need to be recorded. A related question is "where from?", that is, the source of data. Was PII data collected directly from the data subject or from a third party provider?

**How:** Some laws and policies restrict how data is handled. For example, the processing of PCI-regulated data may require encryption and other safeguards. Information on how such data has been handled therefore needs to be recorded for auditability. The OECD's Collection Limitation principle requires PII to be collected with the knowledge of the data subject where appropriate; if it has been collected without the subject's knowledge, this may need to be logged. Similarly, auditability may require the logging of unplanned data disclosures and the reasons for them (internal requests, e-discovery process, compelled by court order, compelled by law enforcement investigation).

**Who:** Policies may restrict access to a data item to a particular set of authorized users, identified either as individuals or by role. There may also be a need to record the corporate identity of partners or cloud service providers to which data is transmitted, as part of due diligence about cloud service provisioning, and to assist actions required by policies if a provider goes out of business or is acquired, or has a data breach.

## *VI.* RELATED WORK

### *A. Governance, Risk Management and Compliance (GRC) Stack of the Cloud Security Alliance (CSA) [45]*

The Cloud Security Alliance (CSA) is a non-profit organization formed to promote the use of best practices for providing security assurance within Cloud Computing, and provide education on the uses of Cloud Computing [46]. The CSA is comprised of many subject matter experts from academia and leading organizations *(Hewlett-Packard, Dell, Intel, RSA, Microsoft, Cisco, Oracle, DMTF, ENISA, AT&T, IBM, Google, etc)*. Two projects from the CSA's Governance, Risk Management and Compliance (GRC) Stack [46] are very relevant to our paper:

- *CloudAudit* [47] – An ongoing API project hosted on Google Code; CloudAudit aims to provide the technical foundation to enable transparency and trust in private and public cloud systems.
- *Trusted Cloud Initiative* [48] – An initiative which aims to promote education, research and certification of secure and interoperable identity in the cloud. Most significant and related to our paper will be their movement towards the certification of 'trusted clouds'.

### *B. HP Labs – Cloud and Security Lab*

Pearson and Mowbray, two of the co-authors of this paper, have done research on technical and procedural methods for promoting cloud privacy [14, 49]. Their previous work has focused on the higher level accountability layers. This paper includes the lower layers identified in Section V. Recently in 2011, Ko, Lee and Pearson highlighted and established the case for accountability in [21], via a short paper covering scenarios and high level concerns of accountability within the cloud.

### *C. University of Pennsylvania/ Max Planck Institute for Software Systems*

Haeberlen et al. were one of the first researchers to call for awareness in an accountable cloud [15]. In [15], they assumed a primitive *AUDIT* with considerations of *agreement*, *service* and *timestamps*. However, *AUDIT* did not have a clear explanation of the scope, scale, phases and layers of abstraction of accountability. It is our aim to complement their work. Their team has also proposed an approach for accountable virtual machines [50], and discussed a case study on the application to detect cheats in an online multi-player game Counterstrike. The scenario of this non-cloud based game was not a practical business scenario for accountability, and did not address the needs of logging virtual-to-physical mapping.

### *D. HyTrust Appliance [18]*

Recently in the industry, HyTrust, a startup focusing on cloud auditing and accountability, has released a hypervisor

consolidated log report and policy enforcement tool *(i.e. HyTrust Appliance)* for virtual machine accountability management in clouds. In the context of Section V, HyTrust Appliance addresses the *System layer* of accountability in the cloud. Despite this, it focuses on the virtual layers and did not mention capabilities for virtual-to-physical complexities. Also, it views logging for accountability from system perspective and not a file-centric perspective.

### E. Accountability of Services by CSIRO

Chen and Wang of CSIRO currently have a team looking at "accountability as a service" for the cloud [51, 52]. Their work presented a prototype which enforces accountability of service providers whose services are deployed in the cloud. This is achieved by making the service providers responsible for faulty services and a technique which allows identification of the cause of faults in binding Web services.

### F. Provenance in Databases and the Web of Data

The concept of provenance has mainly been researched in the context of databases, the Web, and workflow systems – resulting in a myriad of notions and interpretations. Generally, provenance of a data item refers to information about its origin, its creation/collection, and the ways in which it was altered and/or accessed. Buneman et al. [24] consider the notion of data provenance in the context of data management systems and propose a sub-classification into why- and where-provenance. *Why-provenance* captures why a data item is in a query result, while *Where-provenance* explains where the data item came from. This two-sided perspective was later extended by Green et al. [53] to also include *How-provenance* describing how the data item's origin(s) where involved in the computations/query processing. Tan [25] takes a slightly different perspective, which is less database-centric. Provenance is defined for workflows (i.e., a coarse-grained view focusing on the entire history of change of the final result of a workflow) and for data (i.e., a fine-grained view focusing on how a single data item was derived). Building on top of provenance notions from the database domain, Hartig [54] and Halpin [36] have discussed challenges and proposed solutions for provenance adoption in the Semantic Web. The former work mainly focuses on enabling provenance in the Web of Data (which constitutes a key part of the Semantic Web effort), while the latter work positions provenance as the missing building block of the Semantic Web to enable privacy and trust.

### G. Provenance in Clouds

The emergence and rapid adoption of cloud computing has seen a significant increase in research on provenance as it is regarded as the foundation for any model capturing privacy and/or trust. Muniswamy-Reddy et al. [55] discuss the main challenges of provenance adoption for cloud computing and suggest four properties (i.e. data coupling, multi-object casual ordering, data-independent persistence, and efficient querying) that make provenance systems truly useful. While our approach is coherent with these views, we strongly advocate the need for secure and privacy-aware properties. Corresponding notions of secure provenance [56] and privacy-aware provenance [57] have been proposed for cloud computing systems as provenance information may contain or expose sensitive, confidential or proprietary information directly or indirectly.

## VII. Concluding Remarks

In this paper, we establish the urgent need for research in accountability in the cloud, and outline the risks of not achieving it. We propose detective rather than preventive approaches to increasing accountability. Detective approaches complement preventive approaches as they enable the investigation not only of external risks, but also risks from within the CSP. Detective approaches can also be applied in a less invasive manner than preventive approaches. We have argued that the shift in end-users' concerns from *system health and performance* to the *integrity and accountability of data* stored in the Cloud requires a file-centric perspective, on top of the usual system-centric perspective for logging.

Using concepts from the Cloud Accountability Life Cycle and the abstraction layers of logs, we have identified the importance of both real-time and post-mortem approaches to address the nature of cloud computing at different levels of granularity. Our conceptual model potentially can be used to give cloud users a single point of view for accountability of the CSP.

We are currently researching and developing solutions for each layer, with one example being a logging mechanism for the system layer of cloud accountability.

## References


1. Fujitsu Research Institute, "Personal data in the cloud: A global survey of consumer attitudes," 2010; http://www.fujitsu.com/downloads/SOL/fai/reports/fujitsu_personal-data-in-the-cloud.pdf.
2. D. Catteddu and G. Hogben, *Cloud Computing Risk Assessment*, European Network and Information Security Agency (ENISA) 2009.
3. M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin and I. Stoica, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, 2010, pp. 50-58.
4. S. Garfinkel, *An Evaluation of Amazon's Grid Computing Services: EC2, S3, and SQS*, Center for Research on Computation and Society, Harvard University, 2007.
5. D. Chappell, "Introducing windows azure," 2009; http://www.microsoft.com/windowsazure/Whitepapers/IntroducingWindowsAzure/default.aspx.
6. P. Buneman, S. Khanna and W. Tan, "Data provenance: Some basic issues," *FST TCS 2000: Foundations of Software Technology and Theoretical Computer Science*, 2000, pp. 87-93.
7. Cloud Security Alliance, "Top Threats to Cloud Computing (V1.0)," 2010; https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf.
8. Oxford University Press, "Concise Oxford English Dictionary," *Retrieved December*, vol. 5, 2005, pp. 2005.
9. H. Woolf, *The Merriam-Webster Dictionary*, Pocket Books New York, 1974.
10. S. Pearson and A. Benameur, "Privacy, Security and Trust Issues Arising from Cloud Computing," *Proc. The 2nd International Conference on Cloud Computing 2010*, IEEE, 2010, pp. 693-702.
11. J. Brodkin, "Gartner: Seven cloud-computing security risks," *Infoworld*, 2008, pp. 1–3.
12. M. Vouk, "Cloud computing—Issues, research and implementations," *Proc. 30th International Conference on Information Technology Interfaces, 2008 (ITI 2008)* IEEE, 2008, pp. 31-40.
13. S. Pearson, "Taking account of privacy when designing cloud computing services," *Proc. 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, IEEE Computer Society, 2009, pp. 44-52.

14. S. Pearson and A. Charlesworth, "Accountability as a way forward for privacy protection in the cloud," *Cloud Computing*, 2009, pp. 131-144.
15. A. Haeberlen, "A case for the accountable cloud," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, 2010, pp. 52-57.
16. US House of Representatives, *The Best Practices Act of 2010 and Other Privacy Legislation*, T. Sub-Committee on Commerce, and Consumer Protection, 2010.
17. A. Baldwin, S. Shiu and Y. Beres, "Auditing in shared distributed virtualized environments," *HP Technical Reports*, 2008.
18. HyTrust, "HyTrust Appliance," 2010; http://www.hytrust.com/product/overview/.
19. Hyperic, "CloudStatus," 2010; http://www.cloudstatus.com/.
20. J. Shende, "Live Forensics and the Cloud - Part 1," 2010; http://cloudcomputing.sys-con.com/node/1547944.
21. R.K.L. Ko, B.S. Lee and S. Pearson, "Towards Achieving Accountability, Auditability and Trust in Cloud Computing," *Proc. International workshop on Cloud Computing: Architecture, Algorithms and Applications (CloudComp2011)*, Springer, 2011, pp. 5.
22. W. Zhou, M. Sherr, T. Tao, X. Li, B.T. Loo and Y. Mao, "Efficient querying and maintenance of network provenance at internet-scale," *Proc. 2010 International Conference on Management of Data (SIGMOD 2010)*, ACM, 2010, pp. 615-626.
23. R.K.L. Ko, S.S.G. Lee and E.W. Lee, "Business Process Management (BPM) Standards: A Survey," *Business Process Management Journal*, vol. 15, no. 5, 2009.
24. P. Buneman, S. Khanna and T. Wang-Chiew, "Why and where: A characterization of data provenance," *Database Theory—ICDT 2001*, 2001, pp. 316-330.
25. W. Tan, "Provenance in databases: Past, current, and future," *IEEE Data Engineering*, 2007, pp. 3.
26. S. Pearson and B. Balacheff, *Trusted computing platforms: TCPA technology in context*, Prentice Hall PTR, 2003.
27. G. Proudler, "Concepts of trusted computing," *Trusted Computing*IEE Professional Applications of Computing Series, C. J. Mitchell, ed., The Institute of Electrical Engineers (IEE), 2005, pp. 11–27.
28. S. Hansen and E. Atkins, "Automated system monitoring and notification with swatch," USENIX Association's Proceedings of the Seventh Systems Administration (LISA VII) Conference, 1993.
29. M. Roesch, "Snort-lightweight intrusion detection for networks," *Proc. 13th Large Installation System Administration Conference (LISA)*, 1999, pp. 229–238.
30. H. Zimmermann, "OSI reference model--The ISO model of architecture for open systems interconnection," *Communications, IEEE Transactions on*, vol. 28, no. 4, 2002, pp. 425-432.
31. W. Stevens, *TCP/IP Illustrated Vol. I: The Protocols*, Pearson Education India, 1994.
32. R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka and J. Molina, "Controlling data in the cloud: outsourcing computation without outsourcing control," *Proc. 2009 ACM workshop on Cloud computing security (CCSW 2009)*, ACM, 2009, pp. 85-90.
33. M. Rosenblum and J. Ousterhout, "The design and implementation of a log-structured file system," *ACM Transactions on Computer Systems (TOCS)*, vol. 10, no. 1, 1992, pp. 26-52.
34. A. Slagell, J. Wang and W. Yurcik, "Network log anonymization: Application of crypto-pan to cisco netflows," *Proc. NSF/AFRL Workshop on Secure Knowledge Management (SKM '04)*, 2004.
35. A. Slagell and W. Yurcik, "Sharing computer network logs for security and privacy: A motivation for new methodologies of anonymization," *Proc. Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks, 2005.* , IEEE, 2006, pp. 80-89.
36. H. Halpin, "Provenance: The Missing Component of the Semantic Web for Privacy and Trust," *Proc. Proceedings of the Trust and Privacy on the Social and Semantic Web (SPOT) Workshop at ESWC 2009*, Citeseer, 2009.
37. T. Erl, *Service-oriented architecture: concepts, technology, and design*, Prentice Hall PTR Upper Saddle River, NJ, USA, 2005.
38. Sarbanes-Oxley Act, "Public Law No. 107-204," *Book Public Law No. 107-204*, Series Public Law No. 107-204 107th US Congress ed., Editor ed.^eds., 2002, pp.
39. *Health Insurance Portability and Accountability Act (HIPAA) of 1996 (P.L. 104-191)*.
40. A. Calder, *Information Security Based on ISO 27001/ISO 17799: A Management Guide*, The Stationery Office/Tso, 2006.
41. A. Calder and S. Watkins, *IT Governance: A Manager's Guide to Data Security and ISO 27001/ISO 27002*, Kogan Page Ltd. London, UK, UK, 2008.
42. Z. Rezaee, A. Sharbatoghlie, R. Elam and P.L. McMickle, "Continuous auditing: Building automated auditing capability," *Auditing*, vol. 21, no. 1, 2002, pp. 147-164.
43. W.Z.P. Ning, X.Z.G. Ammons, R. Wang and V. Bala, "Always Up-to-date–Scalable Offline Patching of VM Images in a Compute Cloud," *IBM Technical Papers*, no. RC24956, 2010.
44. J. Wei, X. Zhang, G. Ammons, V. Bala and P. Ning, "Managing security of virtual machine images in a cloud environment," ACM, 2009, pp. 91-96.
45. Cloud Security Alliance, "Cloud Security Alliance Governance, Risk Management and Compliance (GRC) Stack," 2010; http://www.cloudsecurityalliance.org/grcstack.html.
46. Cloud Security Alliance, "Cloud Security Alliance Homepage," 2010; http://www.cloudsecurityalliance.org/.
47. Cloud Security Alliance, "CloudAudit (A6 - The Automated Audit, Assertion, Assessment, and Assurance API) " 2010; http://cloudaudit.org/.
48. Cloud Security Alliance, "Trusted Cloud Initiative," 2010; http://www.cloudsecurityalliance.org/trustedcloud.html.
49. M. Mowbray, S. Pearson and Y. Shen, "Enhancing privacy in cloud computing via policy-based obfuscation," *The Journal of Supercomputing*, 2010, pp. 1-25.
50. A. Haeberlen, P. Aditya, R. Rodrigues and P. Druschel, "Accountable virtual machines," *9th OSDI*, 2010.
51. S. Chen and C. Wang, "Accountability as a Service for the Cloud: From Concept to Implementation with BPEL," *Proc. 6th IEEE World Congress on Services (SERVICES-1)*, IEEE, 2010, pp. 91-98.
52. J. Yao, S. Chen, C. Wang, D. Levy and J. Zic, "Accountability as a Service for the Cloud," *Proc. IEEE Service Computing Conference 2010 (SCC 2010)*, IEEE, 2010, pp. 81-88.
53. T.J. Green, G. Karvounarakis and V. Tannen, "Provenance semirings," *Proc. Proceedings of the 26th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PoDS)*, ACM, 2007, pp. 31-40.
54. O. Hartig, "Provenance information in the web of data," *Proc. Proceedings of the Linked Data on the Web (LDOW) Workshop at WWW 2009*, 2009, pp. 1-9.
55. K.K. Muniswamy-Reddy, P. Macko and M. Seltzer, "Provenance for the Cloud," *Proc. Proceedings of the 8th USENIX Conference on File and Storage Technologies*, USENIX Association, 2010, pp. 197-210.
56. R. Lu, X. Lin, X. Liang and X.S. Shen, "Secure provenance: the essential of bread and butter of data forensics in cloud computing," *Proc. Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, ACM, 2010, pp. 282-292.
57. S.B. Davidson, S. Khanna, S. Roy, J. Stoyanovich, V. Tannen and Y. Chen, "On provenance and privacy," *Proc. Proceedings of the 14th International Conference on Database Theory (ICDT)*, ACM, 2011, pp. 3-10.