Towards Achieving Accountability, Auditability and Trust in Cloud Computing

Ryan K.L. Ko[1], Bu Sung Lee[1], and Siani Pearson[2]

[1] Cloud and Security Lab, HP Labs, Fusionopolis, Singapore
[2] Cloud and Security Lab, HP Labs, Bristol, United Kingdom
{ryan.ko,francis.lee,siani.pearson}hp.com

Abstract. The lack of confidence in entrusting sensitive information to cloud computing service providers (CSPs) is one of the primary obstacles to widespread adoption of cloud computing, as reported by a number of surveys. From the CSPs' perspective, their long-term return-on-investment in cloud infrastructure hinges on overcoming this obstacle. Encryption and privacy protection techniques only solve part of this problem: in addition, research is needed to increase the accountability and auditability of CSPs. However, achieving cloud accountability is a complex challenge; as we now have to consider large-scale virtual and physical distributed server environments to achieve (1) real-time tracing of source and duplicate file locations, (2) logging of a file's life cycle, and (3) logging of content modification and access history. This position paper considers related research challenges and lays a foundation towards addressing these via three main abstraction layers of cloud accountability and a Cloud Accountability Life Cycle.

Keywords: Accountable cloud computing, trusted computing platform, accountability, logging, continuous auditing, audit trails.

1 Introduction

In a recent survey by Fujitsu Research Institute [1], it was revealed that 88% of potential cloud consumers surveyed are worried about who has access to their data within the cloud, and would like to have more awareness of what "goes on" in the cloud's backend physical servers. Such surveys have not only identified trust as the key barrier to cloud computing uptake, but also enhanced the urgency for researchers to quickly address key obstacles to trust [1-3].

From a system design perspective, the notion of trust can be increased via reducing risk when using the cloud. While risk can be greatly mitigated via privacy protection and security measures such as encryption, they are not enough, particularly as full encryption of data in the cloud is at present not a practical solution.

There is a need to complement such ***preventative controls*** with equally important ***detective controls*** that promote transparency, governance and accountability of the service providers. This paper focuses on the detective controls of tracing data and file movements in the cloud.

A. Abraham et al. (Eds.): ACC 2011, Part IV, CCIS 193, pp. 432–444, 2011.

Despite accountability being a crucial component of improving trust and confidence [4, 5], current prominent providers (e.g. Amazon EC2/ S3 [6, 7], Microsoft Azure [8]) are still not providing full transparency or capabilities for the tracking and auditing of the file access history and data provenance [9] of both the physical and virtual servers utilized [1]. Currently, users can at best monitor the virtual hardware performance metrics and system event logs of the services in which they engage. The cloud computing research community, particularly the Cloud Security Alliance, has recognized this. In its Top Threats to Cloud Computing Report [10], 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 cloud service providers, such as tracing of file access histories, will allow service providers and users to reduce five of the above seven threats: 1,2,3,5 and 7.

In this position paper, we (1) identify accountability and auditability as urgent research areas for the promotion of trust in cloud computing, (2) discuss the complexities of achieving accountability as a result of cloud computing's promise of elastic resources, (3) propose a conceptual foundation that promotes system designs which fully address the different cloud accountability phases and abstraction layers and (4) discuss related and further work.

2 A Trust-Related Scenario

Figure 1 shows a typical trust-related scenario which many potential cloud customers fear [1]. A customer stores some sensitive data in a file (see Fig. 1 top-left; red icon) within a virtual machine (VM) hosted by a provider s/he has subscribed to. Upon uploading the data, failsafe mechanisms within the cloud will typically back it up, and perform load balancing by creating redundancies across several virtual servers and physical servers in the service provider's trusted domain. From the file's creation to the backup processes, large numbers of data transfers occur across virtual and physical servers (black solid-line arcs; Fig. 1), and several memory read/write transactions to both virtual and physical memories are involved (blue dotted-line arcs; Fig. 1). If all such transactions and the creation of new duplicate files are logged, monitored and accounted for, we would be able to trace the file history and log the access history and content modifications, i.e. achieving cloud accountability and auditability.

Even if a malicious insider of the CSP attempts to transfer the sensitive file/ data to a target outside the cloud (e.g. in Fig. 1, 'via email'), we will be well-equipped to know when, where, how and what was being leaked, and by whom. This empowers both the CSP and the consumers, as problematic processes and even insider jobs may be investigated. This also removes some barriers to confidence in the cloud.

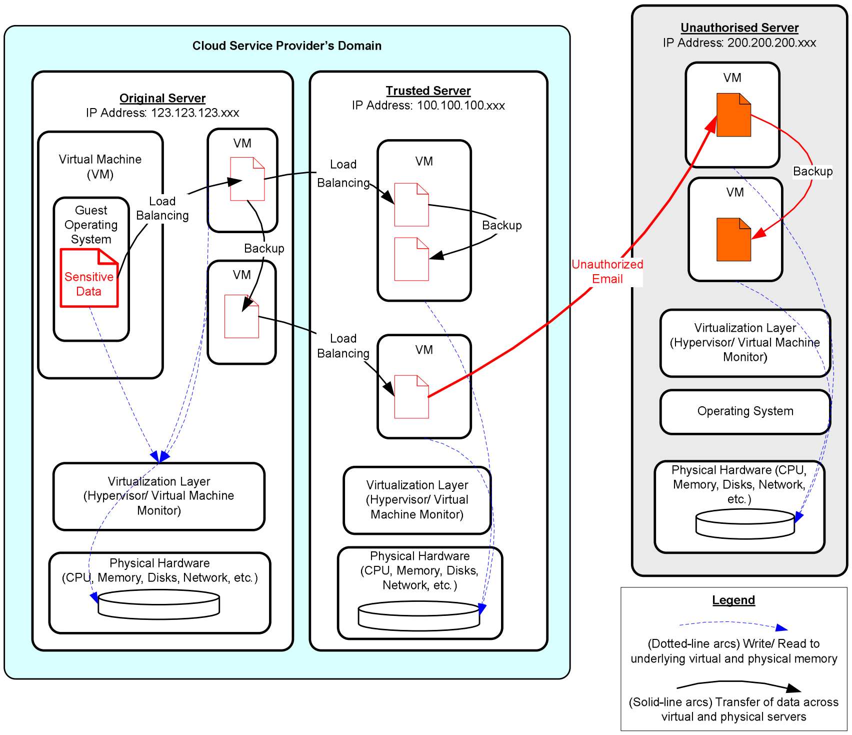
Cloud Service Provider's Domain
Original Server
IP Address: 123.123.123.xxx
Virtual Machine (VM)
Guest Operating System
Sensitive Data
Load Balancing
VM
Backup
VM
Virtualization Layer (Hypervisor/ Virtual Machine Monitor)
Physical Hardware (CPU, Memory, Disks, Network, etc.)
Trusted Server
IP Address: 100.100.100.xxx
Load Balancing
VM
Backup
Load Balancing
VM
Virtualization Layer (Hypervisor/ Virtual Machine Monitor)
Physical Hardware (CPU, Memory, Disks, Network, etc.)
Unauthorized Email
Unauthorised Server
IP Address: 200.200.200.xxx
VM
Backup
VM
Virtualization Layer (Hypervisor/ Virtual Machine Monitor)
Operating System
Physical Hardware (CPU, Memory, Disks, Network, etc.)
Legend
(Dotted-line arcs) Write/ Read to underlying virtual and physical memory
(Solid-line arcs) Transfer of data across virtual and physical servers

Fig. 1. An example scenario in cloud computing, showing the importance of accountability and auditability

3 Complexities Introduced by Cloud's Elasticity

With cloud computing's feature of elasticity empowered by virtualization [6, 11] comes several new complexities introduced related to the area of accountability.

3.1 Challenges Introduced by Virtualisation

3.1.1 Tracking of Virtual-to-Physical Mapping and Vice versa

With the introduction of virtualization, server resources are utilized more efficiently. However, the addition of virtualized layers means that not only the events in each individual virtual server need to be tracked, but also in the physical servers [11]. Currently, there are only tools (e.g. HyTrust [12]) 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. This information is currently disparate and not available as a single-point-of-view for cloud users.

3.1.2 Multiple Operating System Environments

With the ease of choosing myriad operating systems for virtual machines comes the complexity of managing logging of a very large possibility of operating systems (OSs) within the cloud. Enforcing a homogeneous OS for all virtual machines would solve this issue, but makes the provider less competitive. This means that we cannot focus on system health logging, and also existing OS-based logging tools [13], but need a new perspective for logging, as explained in the following section.

3.2 Logging from Operating System Perspective vs. Logging from File-Centric Perspective

Current tools focus on OSs and system health monitoring (e.g. cloudstatus.com, [14], etc), but few emphasize the file-centric perspective. By this, we mean that we need to trace data and files from the time of creation to the time of destruction. When we log from a file-centric perspective, we view data and information independently from environmental constraints. This reflects the elastic nature of cloud computing. With the transfer of control of data to CSPs, the latter should ease the minds of consumers by providing them with the capabilities to track their data (just like those shown in Figure 1).

3.3 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 [15]. This means cloud accountability demands complex real-time accountability, where key suspected events are captured almost instantaneously.

3.4 Scale, Scope and Size of logging

The elasticity concept 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. By scale and scope, we mean policies that can help to clearly define the areas in which loggers are assigned to log. For example, a CSP 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 the level of data abstraction, e.g. crude data, documents, and on a higher level, workflows. These are discussed further in Section 5.

4 Achieving an Accountable Cloud

4.1 Phases of Cloud Accountability

The discussions in Section III and the scenario in Figure 1 have not only revealed the scale and urgency of the problem 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 to gauge the comprehensiveness of their tool (i.e. if there are any phases not covered by their product). Phases can help researchers focus on specific research sub-problems of the large cloud accountability problem. Consumers can also understand if the cloud accountability tool has a real coverage of all phases of cloud accountability. These phases are collectively known as the Cloud Accountability Life Cycle (CALC). We propose CALC as the following seven phases (see Figure 2):

1) Policy Planning

In the beginning, 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

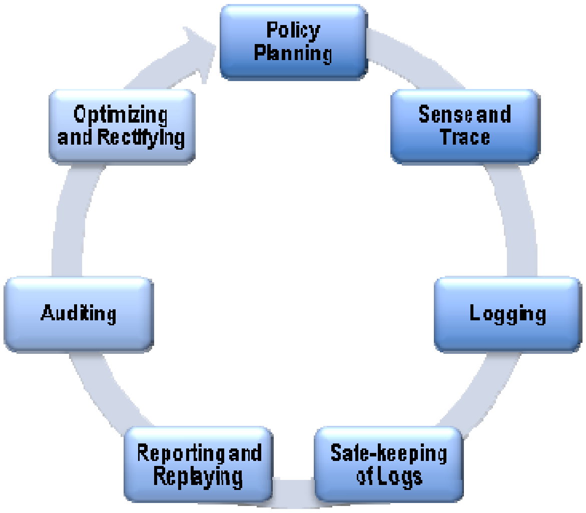

Fig. 2. The Cloud Accountability Life Cycle

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.

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.

4) Safe-keeping of Logs

After logging is done, we need to protect the integrity of the logs 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 cover a large scope: virtual and physical server histories within the cloud; from OS-level read/write operations of sensitive data to high-level workflow audit trails.

6) Auditing

Logs and reports are checked and potential fraud-causing loopholes 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 and end-users 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.

4.2 Cloud Accountability Abstraction Layers

Next we address the important question: what data to log? The answer ranges from a system-level log to a workflow-level audit trail transactional log. Such a range shows that there are many abstraction layers of data, and a framework is needed to reduce this kind of ambiguity and increase research focus and impact. As such, we propose the following layers of accountability in a cloud:

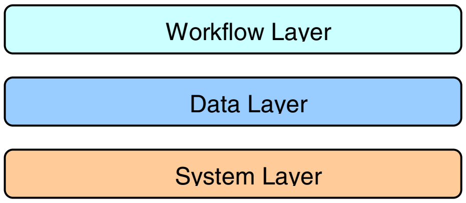

Fig. 3. Abstraction Layers of Accountability in Cloud Computing

Figure 3 shows the abstraction layers for the type of logs needed for an accountable cloud. It is important to note that the focus is on the abstraction layers of logs and not on architectural layers. Hence, it is independent of virtual or physical environments. The data and workflow abstraction layers are derived from related works in data and workflow provenance [9, 16, 17], and the system layer is derived from related works in trusted computing platforms [18, 19] and system logging literature [20, 21].

Such explicit definition of layers in Figure 3 allows us to efficiently identify the areas of their application and their focus areas. At a glance, the three 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 layers is also analogous to OSI [22] and TCP/IP [23] networking layers. Let us now discuss the scope and scale of each layer:

4.2.1 System Layer

At the lowest level lie the system layer logs. The system layer consists of logging within the following components:

1) Operating System (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 [6, 11]. 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, 24].

2) File System

Even though the file system is technically part of the OS, we explicitly include it as a major component in this system layer. This is because, in order to know, trace and record the exact file life cycle and history, we often have to track system read/write calls to the file system. From the system read/write calls, we can also extract the virtual and physical memory locations of the file, providing more information for further forensic investigations. The file-centric perspective [25] is also the area which is less emphasized by current tools. Cloud computing needs to have more emphasis on file-centric logging, and the tracing and logging of a file’s life cycle (i.e. creation, modification, duplication, destruction).

3) Network Logs

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 [26, 27] are logs specific to data being sent and received over the network.

4.2.2 Data Layer

This layer contains the logging of data transactions and the life cycle of data. The difference with the system layer is that the system layer's file system logs track the life cycle of files, whereas the data layer actually tracks the life cycle of data and the contents of files. The same file can contain drastically different sets of data over time. Some examples of the data layer are: (1) data provenance, which records the so-called chains of custody [9, 16, 17] (e.g. the history of owners and authorized users) of the data found in the cloud and (2) database logs [28, 29] (i.e. histories of updates and actions executed by a database management system to the database).

4.2.3 Workflow Layer

This layer primarily contains logs which reveal the robustness or weaknesses of the governance and controls of a workflow or business process [30, 31] in an organization. It correlates with an organization's strategic and management levels [32]. It is the key layer audited by most IT auditors and internal audits. Examples include: (1) audit trails from transactions in business process and workflow management systems (2) audit trails from information systems for the customer organizations (e.g. ERP systems, Human Resource systems, etc) (3) continuous auditing and monitoring tools.

4.3 Foreseeable Research Challenges

4.3.1 Growth of Log Size

Without a doubt, this will be an obstacle to efficient auditability and will most likely be the major problem in cloud accountability. Some of the main areas of research will be on the optimal period of storage of logs, how to shrink logs over time, and how and where to store logs efficiently.

4.3.2 Security and Integrity of Logs (No Tampering)

With an increased awareness (from customers and also hackers) of the cloud's ability in accountability, there will be attempts to tamper with logs to escape detection of fraudulent activities. One of the ways to minimize compromise of security of logs is to make the logging silent within the cloud. The integrity and security of logs must never be compromised or there will be leakage of user access history and possibilities of competitors studying customers' usage behavior.

4.3.3 Privacy vs. Accountability

With logs available in the cloud, hackers and corporate spies can benefit from studying these logs. Key competitive advantages may be lost, and it is of the utmost important that CSPs place strong focus on privacy and secrecy of logs.

5 Technical Approaches to Increasing Accountability

With the definition of CALC and the abstraction layers of the type of data to log, we are primed to create tools and software which will achieve cloud accountability. Currently, we envision three possible technical approaches:

1) Central Watchdog/ Manager Service

In this approach, a watchdog service manages a certain set of nodes, and watches over the physical and virtual logs of all layers and stores the logs centrally. While this is more economical and easier to maintain, such a watchdog service would undoubtedly be vulnerable to network routing problems, interference or use of false identities.

2) Local File Tracking Embedment

In this approach, we envision that a file is designed to dedicate some of its memory for storage of bite-sized local logs and provenance data. Currently, this is very difficult to achieve in current file extensions as they are usually predefined without much consideration of local logging.

3) Domain segregation

Accountability in cloud computing will be more achievable if there is a clear design of different domains from the perspective of CSPs or customers. Internal Zones can depict the CSP's own network, with Trusted Zones for its Collaborators, and External Zones for networks outside these two zones. If the data leaves authorized zones, the event will be flagged.

6 Policy Approaches to Increasing Accountability

Primarily, the aim is to improve trust. While technical approaches may be efficient, we foresee that they must be accompanied by the following policy approaches.

1) Certification of "trusted clouds"

Cloud service providers have to conform and be audited to a set of rules and guidelines to be labeled as a "trusted cloud". A recent movement by the CSA related to this concept is the Trusted Cloud Initiative [33].

2) Cloud Trust Track Record

Cloud service providers are subject to customer ratings, and each record of 'breach in trust' will be tabulated into a central, neutral database, which will advise consumers on the particular strength and security standards of the ranked clouds. This is likened to airline safety track records and serves as a standard for cloud service providers to uphold.

3) Legislation

Governmental bodies must control and impose penalties on breach of trust in the cloud, much like antitrust and privacy protection laws.

7 Related Research

Cloud accountability and auditing are growing areas of active research. We summarize some key elements below:

7.1 Governance, Risk Management and Compliance (GRC) Stack of the Cloud Security Alliance (CSA) [34]

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

Cloud Computing uses [35]. Two projects from the CSA's GRC Stack [35] are very relevant to our paper:

- CloudAudit [36] – An ongoing API project hosted on Google Code, and aims to provide the technical foundation to enable transparency and trust in private and public cloud systems.
- Trusted Cloud Initiative [33] – 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'.

7.2 CSC and National Institute of Standards and Technology

In mid-2010, at 6th Annual Information Technology Security Automation Conference (ITSAC) hosted by the National Institute of Standards and Technology (NIST), a representative from the technology provider CSC presented the "CloudTrust Protocol (CTP) with Security Content Automation Protocol (SCAP)" [37]. The CTP with SCAP was claimed to offer a simple way to request and receive the fundamental information needed to address cloud transparency. At the time of writing, there is no public release of the proposed tool.

7.3 HP Labs – Cloud and Security Lab

Pearson [4, 5, 38, 39] and Mowbray [38, 40] were two of the first researchers to aim to promote privacy protection via procedural and technical solutions encouraging the increase of accountability in the cloud [5, 39]. Their work on cloud privacy has addressed the high levels of the accountability layers, and this paper aims to complement their work with the inclusion of the lower system layers identified in Section VI.

7.4 University of Pennsylvania

Haeberlen et al were one of the first researchers to call for awareness in an accountable cloud [41]. In [41], 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 abstraction layers of accountability. It is our aim to complement their work. Their team has also proposed an approach for accountable virtual machines [42], and discussed a case study on the application to detect cheats in an online multi-player game Counterstrike. In our opinion, the scenario of a non-cloud based game was not a practical business scenario for cloud accountability.

7.5 HyTrust Appliance [12]

Recently in 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 VM accountability management in clouds. HyTrust Appliance addresses the System layer of accountability in the cloud. Despite this, it focuses only on virtual layers and is not virtual-to-physical complexities.

7.6 Data and Workflow Provenance Research [17]

From the field of databases, data and workflow provenance research focuses on recording histories of derivation of final outputs of data at different levels of abstraction within databases. Provenance research may offer clues to recording logs in the workflow and data layers of cloud accountability.

8 Concluding Remarks

We highlighted accountability and auditability as an important perspective towards increasing trust in cloud computing. Several complexities introduced by the cloud's nature of elasticity were discussed. Some examples include (1) tracking of virtual-to-physical mapping and vice versa, (2) multiple operating system environments, (3) logging from file-centric perspective, (4) live and dynamic systems, and (5) the scale, scope and size of logging.

Achieving accountability and auditability in cloud computing will also empower: automated monitoring and enforcement; Sarbanes-Oxley (SOX) audits in Clouds; cloud security forensics; learning and analytics of usage behavior.

To simplify and enable efficient scoping of this complex problem, we proposed the Cloud Accountability Life Cycle (CALC) and three abstraction layers. With these conceptual foundations, researchers and practitioners can design tools and approaches which address all areas of cloud accountability. This paper also discussed imminent roadblocks to achieving accountability. In addition to related work discussions, technical and policy approaches were suggested.

Moving forward, we are developing the different modules in the CALC, eg. logging and mapping of virtual machines to physical machines. We believe that with CALC, we would have a model that enables us to have a Trusted Cloud environment where there is accountability and auditability.

Acknowledgments. The authors would like to thank their HP Labs colleagues Peter Jagadpramana and Miranda Mowbray for their input.

References

1. Fujitsu Research Institute: Personal data in the cloud: A global survey of consumer attitudes (2010)
2. Gross, G.: Microsoft presses for cloud computing transparency (2010), `http://www.infoworld.com/d/cloud-computing/microsoft-presses-cloud-computing-transparency-799`
3. Strukhoff, R.: Cloud Computing Vendors Need More Transparency (2010), `http://cloudcomputing.sys-con.com/node/1308929`

4. Pearson, S., Benameur, A.: Privacy, Security and Trust Issues Arising from Cloud Computing. In: The 2nd International Conference on Cloud Computing. IEEE, Indiana (2010)
5. Pearson, S., Charlesworth, A.: Accountability as a way forward for privacy protection in the cloud. In: Cloud Computing 2009, pp. 131–144 (2009)
6. Armbrust, M., et al.: A view of cloud computing. Communications of the ACM 53(4), 50–58 (2010)
7. Garfinkel, S.: An Evaluation of Amazon's Grid Computing Services: EC2, S3, and SQS (2007)
8. Chappell, D.: Introducing windows azure. Microsoft (2009)
9. Buneman, P., Khanna, S., Tan, W.: Data provenance: Some basic issues. In: Foundations of Software Technology and Theoretical Computer Science, pp. 87–93 (2000)
10. Cloud Security Alliance: Top Threats to to Cloud Computing Report, Ver.1.0 (2010)
11. Baldwin, A., Shiu, S., Beres, Y.: Auditing in shared distributed virtualized environments. HP Technical Reports (2008)
12. HyTrust. HyTrust Appliance (2010), http://www.hytrust.com/product/overview/
13. Silberschatz, A., Galvin, P., Gagne, G.: Operating system concepts. Addison-Wesley, New York (1991)
14. Hyperic: CloudStatus (2010), http://www.cloudstatus.com/
15. Shende, J.: Live Forensics and the Cloud - Part 1. Cloud Computing Journal (2010), http://cloudcomputing.sys-con.com/node/1547944
16. Buneman, P., Khanna, S., Wang-Chiew, T.: Why and where: A characterization of data provenance. In: International Conference on Database Theory—ICDT 2001, pp. 316–330 (2001)
17. Tan, W.: Provenance in databases: Past, current, and future. Data Engineering 2007, 3 (2007)
18. Pearson, S., Balacheff, B.: Trusted computing platforms: TCPA technology in context. Prentice Hall PTR, Upper Saddle River (2003)
19. Proudler, G.: Concepts of trusted computing. In: Mitchell, C.J. (ed.) Trusted Computing. IEE Professional Applications of Computing Series, vol. 6, pp. 11–27. The Institute of Electrical Engineers (IEE), London (2005)
20. Hansen, S., Atkins, E.: Automated system monitoring and notification with swatch. In: USENIX Association's Proceedings of the Seventh Systems Administration (LISA VII) Conference (1993)
21. Roesch, M.: Snort-lightweight intrusion detection for networks. In: Proceedings of the 13th USENIX Conference on System Administration, LISA 1999, Seattle, Washington (1999)
22. Zimmermann, H.: OSI reference model–The ISO model of architecture for open systems interconnection. IEEE Transactions on Communications 28(4), 425–432 (2002)
23. Stevens, W.: TCP/IP Illustrated: The Protocols, vol. I. Pearson Education, India (2004)
24. Chow, R., et al.: Controlling data in the cloud: outsourcing computation without outsourcing control. In CCSW 2009: Proceedings of the 2009 ACM Workshop on Cloud Computing Security. ACM, New York (2009)
25. Rosenblum, M., Ousterhout, J.: The design and implementation of a log-structured file system. ACM Transactions on Computer Systems (TOCS) 10(1), 26–52 (1992)
26. Slagell, A., Wang, J., Yurcik, W.: Network Log Anonymization: Application of Crypto-PAn to Cisco NetFlows. In: NSF/AFRL Workshop on Secure Knowledge Management (SKM 2004), Buffalo, NY (2004)

27. Slagell, A., Yurcik, W.: Sharing computer network logs for security and privacy: A motivation for new methodologies of anonymization. In: Proceedings of SECOVAL: The Workshop on the Value of Security Through Collaboration (August 2005)
28. Gray, J., Reuter, A.: Transaction processing: concepts and techniques. Morgan Kaufmann, San Francisco (1993)
29. Peters, T.: The history and development of transaction log analysis. Library Hi Tech. 11(2), 41–66 (1993)
30. Ko, R.: A computer scientist's introductory guide to business process management (BPM). ACM Crossroads 15(4), 11–18 (2009)
31. Ko, R., Lee, S., Lee, E.: Business process management (BPM) standards: a survey. Business Process Management Journal 15(5), 744–791 (2009)
32. Anthony, R.: Planning and control systems: a framework for analysis. Division of Research, Graduate School of Business Administration, Harvard University (1965)
33. Cloud Security Alliance: Trusted Cloud Initiative (2010), `http://www.cloudsecurityalliance.org/trustedcloud.html`
34. Cloud Security Alliance: Cloud Security Alliance Governance, Risk Management and Compliance (GRC) Stack (2010), `http://www.cloudsecurityalliance.org/grcstack.html`
35. Cloud Security Alliance (2010), `http://www.cloudsecurityalliance.org/`
36. Cloud Security Alliance: CloudAudit (A6 - The Automated Audit, Assertion, Assessment, and Assurance API) (2010), `http://cloudaudit.org/`
37. Knode, R.: CloudTrust 2.0 (2010), `http://scap.nist.gov/events/2010/itsac/presentations/day2/Security_Automation_for_Cloud_Computing-CloudTrust_2.0.pdf`
38. Mowbray, M., Pearson, S., Shen, Y.: Enhancing privacy in cloud computing via policy-based obfuscation. The Journal of Supercomputing, 1–25 (2010)
39. Pearson, S.: Taking account of privacy when designing cloud computing services. In: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing. IEEE, Los Alamitos (2009)
40. Mowbray, M., Pearson, S.: A client-based privacy manager for cloud computing. In: Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE, COMSWARE 2009. ACM, New York (2009)
41. Haeberlen, A.: A case for the accountable cloud. ACM SIGOPS Operating Systems Review 44(2), 52–57 (2010)
42. Haeberlen, A., et al.: Accountable virtual machines. In: Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2010 (2010)