Computing and Informatics, Vol. 33, 2014, 1333–1355

AGENT-BASED CLOUD RESOURCE MANAGEMENT FOR SECURE CLOUD INFRASTRUCTURES

Zoltán Balogh, Emil Gatial, Ladislav Hluchý

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: {zoltan.balogh, emil.gatial, ladislav.hluchy}@savba.sk

Ronald Toegl, Martin Pirker, Daniel Hein

Institute for Applied Information Processing and Communications
Inffeldgasse 16a
8010 Graz, Austria
e-mail: {ronald.toegl, martin.pirker, daniel.hein}@iaik.tugraz.at

Abstract. The cloud offers clear benefits for computations as well as for storage for diverse application areas. Security concerns are by far the greatest barriers to the wider uptake of cloud computing, particularly for privacy-sensitive applications. The aim of this article is to propose an approach for establishing trust between users and providers of cloud infrastructures (IaaS model) based on certified trusted agents. Such approach would remove barriers that prevent security sensitive applications being moved to the cloud. The core technology encompasses a secure agent platform for providing the execution environment for agents and the secure attested software base which ensures the integrity of the host platform. In this article we describe the motivation, concept, design and initial implementation of these technologies.

Keywords: Cloud computing, data privacy, cloud confidentiality, computer security, cryptography, privacy, trusted computing

1 INTRODUCTION

A major challenge in the area of security in the cloud is to develop new technologies, systems and approaches that can provide the best possible guarantees for potential security concerned users to leverage the tremendous potential of cloud computing. According to a Gartner Field Study (Figure 1) security and privacy-related issues remain the main concern for those contemplating a move to the cloud. Security and reliability issues will have to be resolved in order to capture more users.

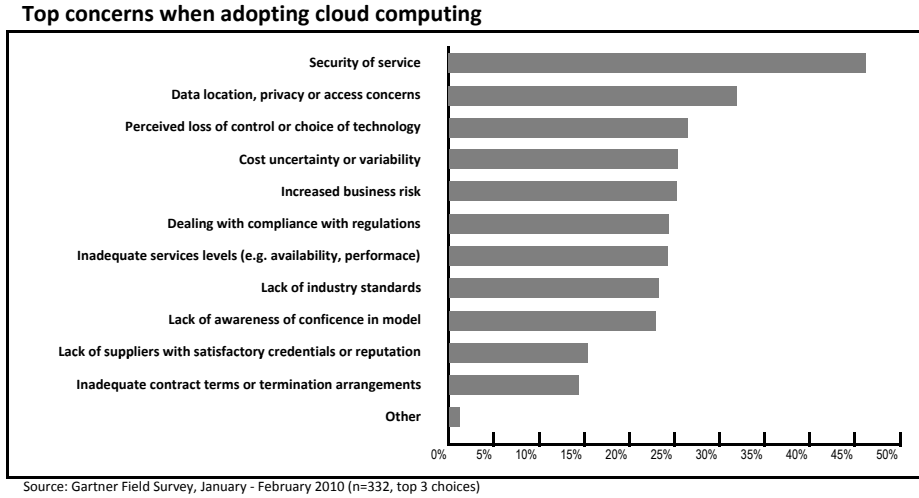


Figure 1. Top concerns when adopting cloud computing

The Security Guidance for Critical Areas of Focus in Cloud Computing [1] published by the Cloud Security Alliance (CSA) regards cloud computing to be a matter of "gracefully losing control while maintaining accountability even if the operational responsibility falls upon one or more third parties". However, many potential cloud users see losing control of security sensitive or private data or losing control of processing such confidential digital assets as a major roadblock to adopting cloud computing.

1.1 Motivation

Nowadays information security professionals must face a challenge on how to gain "trust" when outsourcing their IT infrastructure to cloud providers and to answer the question [2]:

"Do you trust an external third party with your sensitive data?"

This is the primary concern for anybody moving his operations into the cloud. The most common way to establish trust between a cloud user (*data controllers*)

and a cloud provider (*data processors*) is by establishing a Service Level Agreement (SLA), where the provider is bound to set and enforce policies for the provided infrastructure services. Figure 2 depicts a typical situation of a security concerned user who is considering migrating IT operations into the Cloud.

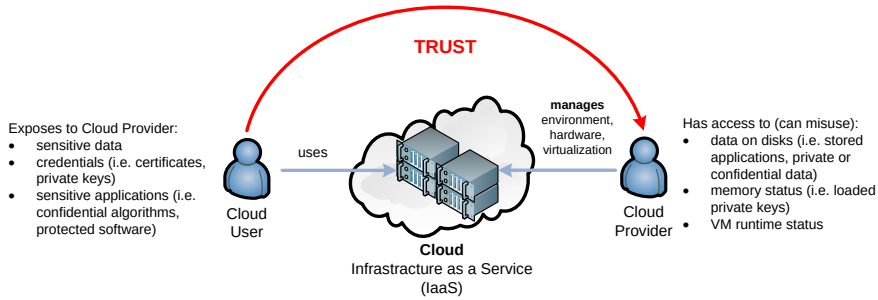


Figure 2. Current security challenge of a security concerned user migrating to Cloud

A trust relationship is established between a cloud user and a cloud provider by signing an SLA. Although policies are important and even if the provider is audited the confidential assets (data, computations) are still disclosed to the provider and its staff. The current cloud offerings pose serious security challenges. The major risks concerning cloud infrastructures which we target in this article are:

Losing control over critical information assets. As soon as the IT operations are moved into the cloud the user loses control over them. For instance, if data is uploaded into the cloud (i.e. into a virtual machine's file system), the user loses control over how many times the data is copied, backed up, used or otherwise processed.

Malicious insider attack. Even if the cloud provider is bound by an SLA, there are threats of disclosing a customer's security sensitive data and computations in the cloud infrastructure. Specifically, malicious insiders (e.g. cloud administrators) represent a significant concern.

Security of higher cloud layers. The security of the cloud infrastructure has a direct impact on the security of the upper cloud layers. Therefore, a compromise of the IaaS layer has direct effect on the security of platforms (PaaS) and software services (SaaS) build above the infrastructure.

1.2 Structure of the Article

The article is organized as follows. Section 2 reviews the current state of the art in related fields of study. Section 3 describes the concept of the introduced solution. Design of the overall solution is presented in Section 4. We describe the prototype implementation and provide validation of the designed approach in Section 5. The last section concludes the achievements presented in this paper.

2 STATE OF THE ART

Herein we provide review of current state-of-the-art in three key areas: introduction to basic cloud computing concepts, agent-based cloud management and trustworthy execution environments for the cloud.

2.1 Basic Cloud Computing Concepts

A computing cloud is a set of network enabled services, providing scalable, QoS guaranteed, inexpensive computing infrastructures on demand, which can be accessed in a simple and pervasive way [3]. Conceptually, users acquire computing platforms or IT infrastructures from computing clouds and then run their applications inside. Therefore, computing clouds render users with services to access hardware, software and data resources as an integrated computing platform in a transparent way. Users thus can on-demand subscribe to their favourite computing infrastructures with requirements of hardware configuration, software installation and data access demands. The architecture of the cloud is based on the following layers [4] (Figure 3):

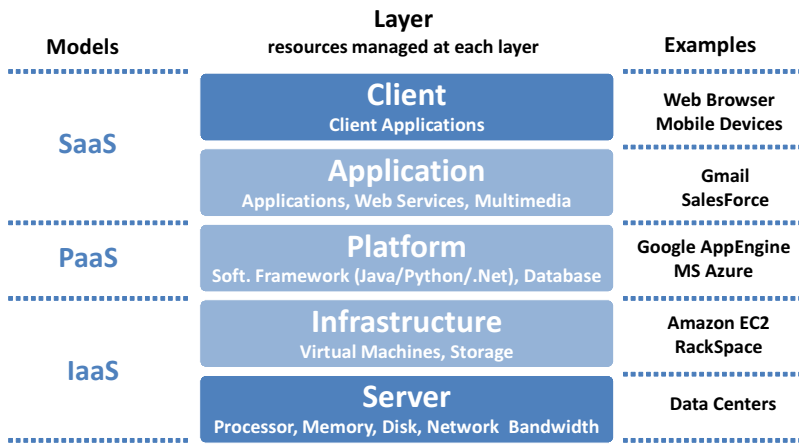


Figure 3. Cloud services architecture

The server level comprises of hardware and software resources, enabling the creation of Cloud infrastructure, based for example on multi-core processors, or special operating systems. The next layer is the computing infrastructure, mostly in the form of virtualized environments. Instead of procurement of physical infrastructure (such as servers, networking components, etc.) it is possible to rent computing capacity in the required amount and for the required time period. Payment for infrastructure, as it is with other consumer commodities (like electricity, water, gas),

is done according to consumption, i.e. the CPU time, disk space, or the use of network bandwidth (so called "utility computing"). The next layer is a platform that makes use of infrastructure and creates the environment for the execution of applications, which constitutes another layer of architecture. Such a platform enables to run applications without the need for purchase and operation of real hardware and software. Finally, the last layer is a client – terminal device through which the services are used. Examples of clients are personal computers, web browsers or mobile devices.

According to the National Institute of Standards and Technology (NIST) Cloud Computing Reference Architecture [5] services in a Cloud are provided in one of three basic models: In the *Infrastructure as a Service (IaaS)* model the virtualized computing resources are delivered (for a reward). The basis for the IaaS model are virtualization technologies such as Xen [6], KVM [7] or VMware [8] allowing the creation of virtual servers and run them over physical devices. A user specifies its hardware requirements using the application programming interface (API). A provider establishes a virtual computing space based on the requirements and provides the user with access to it. IaaS can be used as a standalone service, but also as a basis for higher levels of service. Examples of services provided by the IaaS model are Amazon EC2 [9] or ElasticHosts [10]. The service model providing a platform is known as a *Platform as a Service (PaaS)*. In case of *Software as a Service (SaaS)* model, the applications are provided as services mostly through a web interface. These models are layered above each other where IaaS is the bottom layer.

In this article we identify and propose a solution to security challenges for cloud infrastructures (the IaaS model). Excluding compromise of data or IT operations for cloud infrastructures is imperative, particularly for mission-critical and data-sensitive applications. Security related concerns prevent adoption in key innovative areas where new services could be created. Our objective is therefore to propose a solution which would resolve the security challenges of cloud infrastructures. In this article we propose a design and implementation of technologies which we believe can be used as a foundation for secure cloud infrastructures.

2.2 Management of Cloud Resources Using Agents

Use of agents in cloud computing is a relatively new research domain. Several authors suggested to exploit agents for management [11, 12, 13] or proactive monitoring [14] of computer networks. Several papers focus on approaches for cloud resource discovery and Service Level Agreement (SLA) negotiation. With the advent of new single-chip cloud computer (SCC) technology thermal optimization using agents was proposed [15]. A mobile agent-based service for cloud computing uses agents that can roam in the cloud between different platforms rather than using RPC/RMI service. They use mobile agents to implement the software and services for cloud users and make the cloud adaptable to the Internet environment [17]. The AgPSM (Agent-based Proactive System Management) system [18] involves active system resource monitoring. It proposes means to identify failures, provides fast

resolution of problems by following a sequence of events or activities. The AgPSM system is also capable to construct a warning pattern ahead of any potential outage, thus effectively managing the infrastructure in the long-run. The designed task of agents is primarily service property negotiation and service composition from autonomously selected services. In [16] authors propose a way to prevent the disclosure of confidential and private data by malicious attacks inside the cloud. They refer to paper [36] where three malicious insider attacks were described.

There are few vendors that offer pervasive approaches for handling the provisioning and managing metrics in hybrid environments, namely RightScale, Kaavo, Zeus, Scalr or Morph. Several cloud providers offer proprietary solutions such as CloudWatch from Amazon Web Services. IBM offers Tivoli for cloud management. OpenView is a complementary product by HP which allows management of cloud servers. Anyhow, neither of these products is based on agent technology and thus can not provide the advantages of agent-based computing such as mobile code deployment, reduction of network load, toleration to network failures or dynamic functional behaviour adaptation.

2.3 Trustworthy Execution Environments for the Cloud

Trusted Computing as it is available today is based on specifications of the Trusted Computing Group (TCG). The core hardware component involved is the Trusted Platform Module (TPM) [37]. Similarly to a smart card, the TPM features tamper-resilient cryptographic primitives, but is physically bound to its host device. It implements public-key cryptography, key generation, secure hashing, and random-number generation. Using these components, the TPM can enforce security policies on hierarchies of secret keys to protect them from software attacks by any remote attacker. Furthermore, the TPM helps to guarantee the integrity of measurements of software components. The Enforcer platform [38] and IBM's Integrity Measurement Architecture (IMA) [39] show how to integrate TCG-style static measurements into the Linux environment. While this collects precise information, it does not always allow identifying a limited number of possibly good configurations as the collected measurements will depend on hardware firmware and will consist of hundreds of files in different and variable order. As a workaround, file system images have been used to transport user software and data with SoulPads [40] or Secure Virtual Disk Images in grid services [41] between platforms.

Beyond just adding a TPM chip, modern platforms offer Intel Trusted Execution Technology (TXT), or AMD SVM. These provide CPU instructions and chipset modifications that allow switching a system to a well-known system state. If this state is used as starting point for a chain of trust, it is referred to as Dynamic Root of Trust for Measurement (DRTM). DRTM can be used to perform software measurements that do not depend on BIOS or other firmware. Another feature is hardware virtualization with strong isolation of partitions.

2.4 Trusted Virtualized Platforms

Platform virtualization has been used to leverage the security features of the TPM. Virtualization is a methodology of dividing the resources of a computer into multiple execution environments, by applying concepts such as time-sharing, hardware and software partitioning, machine simulation or emulation. Hardware architectures can be designed to offer complete virtualization [45] in hardware and thus host several unmodified operating systems in parallel.

Early examples of trusted virtualization platforms are PERSEUS [19] and Terra [20] or the Nizza [21] architecture. Microsoft's now apparently inactive NGSCB [22] project envisioned the security critical Nexus kernel to provide an environment for security critical services, while running a legacy OS in parallel. The EMSCB project demonstrates TPM-based Trusted Computing on an L4-based hypervisor. The FP6 OpenTC project demonstrated a system based on a static chain-of-trust from the BIOS to the bootloader via hypervisors, and into application partitions measured and loaded from CD images. Cocker et al. [23] describe a Xen-based platform which is focused on Remote Attestation. Schiman et al. [24] describe an all-layer (hypervisor to application) integrity enforcement and reporting architecture for distributed systems. Cabuk et al. [25] propose to use a software-based root of trust for measurement to enforce application integrity in federated virtual platforms, i.e. Trusted Virtual Domains [26].

Krishna et al. [27] propose a basic security architecture involving trusted virtualization and present a few security protocols. No practical implementation was reported. Krautheim et al. [28] propose the Trusted Virtual Environment Module, a software appliance that serves as virtual security module for IaaS cloud applications on virtualization platforms. As a cryptographic module the proposal shows a potential way to allow platform owner and Cloud user to share responsibility and control over data in the cloud. Brown and Chase [29] propose to use Remote Attestation so that users can gain insights and trust into SaaS service applications by leveraging trust in a neutral third party. They assume the Cloud platform and provider to be trustworthy, without actually relying on hardware security mechanisms. SICE [31] is a novel framework to provide hardware-level isolation and protection for sensitive workloads running on x86 platforms in compute clouds. It is not based on a traditional hypervisor, but it utilizes the System Management Mode (SMM) to isolate different CPU cores. The presented prototype therefore requires a customized platform firmware and currently does not integrate further trust mechanism such as the TPM. The IBM Trusted Virtual Data center (TVDc) [30] is designed to offer security guarantees in hosted data centers. It provides containment and trust guarantees based on virtualization. Isolation and TPM-based integrity are managed. It builds upon a Hypervisor derived from Xen and performs TPM-based measurements of software. The UK myTrustedCloud [32] project studies the integration of an IaaS cloud platform with KVM-based virtualization and hypervisor trust mechanisms built upon IBM IMA. Different levels of attestation are provided for the different layers in the software architecture.

Only a few very recent trusted platform proposals apply DRTM mechanisms; Vasudevan et al. [33] discuss general DRTM requirements, BIND uses AMD's Secure Virtual Machine (SVM) protection to collect fine grained measurements on both input and the small code modules that operate on it so that the computation results can be attested to. Flicker [34] isolates sensitive code by halting the main OS, switching into AMD SVM, and executing short-lived pieces of application logic (PALs). PALs may use the TPM to document their execution and handle results. As a trusted hypervisor, TrustVisor [35] is initiated via the DRTM process, assumes full control and allows managing, running and attesting multiple PALs in its protection mode, without the switch costs incurred by the Flicker approach.

3 PROPOSED SOLUTION

In order to address the above mentioned risks (mentioned in Section 1.1), we must solve the challenge how to enable utilization of the cloud infrastructures without disclosing security sensitive information assets of users (data, computations or applications) to a cloud provider but at the same time enable *cloud users* to exploit advantages of cloud computing in a secure and trusted way, and enable *cloud providers* to efficiently and securely manage cloud infrastructure for their customers without direct access to customer's information assets. Our proposed concept is depicted in Figure 4.

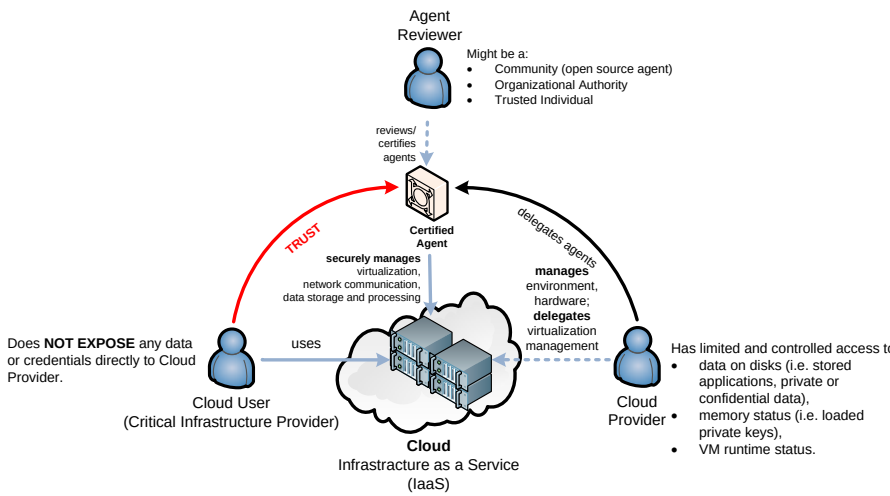


Figure 4. Addressing cloud security concerns using certified trusted agents

The proposed concept changes the way trust is established in the cloud by introducing the following concepts:

1. *Certified Trusted Agent (Agent)* – a piece of program code, which can be securely uploaded and managed, under very strict conditions, to a distant agent execution platform (point 2). Agents can be delegated to mediate specific operations such as virtual machine management (creation, launch, deletion or backup). Agents might be suitable also for other tasks such as computer network monitoring or dynamic reconfiguration.
2. *Agent Platform (AP)* – is the agent's execution environment and can be deployed directly on a host machine or in an dedicated virtual machine. AP enables secure deployment and execution of agents based on configurable policies.
3. *Secure Attested Software Base (SASB)* – ensures the integrity of the host platform. The AP is deployed on a server which is secured by SASB.

Agent Reviewer (AR) is an additional role introduced in our concept. In order to trust an agent both cloud users and cloud providers must be ensured that the agent code does exactly what it claims to do. The AR is responsible for agent code review and for ensuring agent's code integrity. Agents can be created and reviewed by one or several entities that act as warrantors of agent's functionality and execution safety. Thus AR can be any trusted entity including community (in case of open-source agents), individuals or dedicated organizations.

Additionally, to prevent physical attacks to the equipment (i.e. retrieving data from un-mounted hard drive) and to limit security attacks from the network to the infrastructure, our concept also suggests to encompass the following:

4. *Data protection* through encryption of both virtual containers' and the agent platform's file systems.
5. *Identity management, authentication, authorization and accounting of users and agents* for secure cloud access.
6. *Management of cloud infrastructure security breaches* mainly concerning detection, notification and reaction.
7. *Infrastructure availability* by remote synchronization of file systems of virtual containers via secure network to backup data centers and ensure resilient failover.

We believe such setup enables highly secure and trusted cloud infrastructure provisioning where certified trusted agents are the means by which we establish trust and security in cloud.

4 ADDRESSING THE IAAS SECURITY CHALLENGE

In the IaaS model several servers are operated by a cloud provider. These servers use a virtualization technique to execute and manage virtual machines (VM) on behalf of the cloud user. Figure 5 depicts one such server. Administrators of the cloud provider have full control over the hardware, operating system (OS) and hypervisor

layers of this infrastructure. Depending on the virtualization type administrators might have full (in a case of OS-level/container virtualization) or limited (in a case of hardware-based virtualization or paravirtualization) access to VM's file system, processes and memory statuses. Even in the case of limited access to users' VMs there are attacks possible [36] which enable an administrator to mount file systems or dump memory status of VMs. This poses a serious risk of *malicious insider attack* from the cloud provider side.

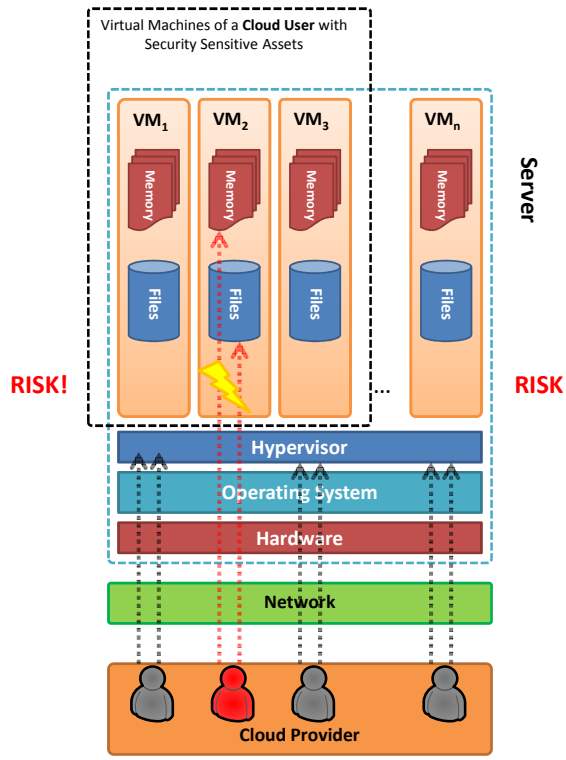


Figure 5. Infrastructure level cloud security challenge – risk of exposing confidential data and memory status of VMs to cloud provider administrators

It is important to note that only administrator with root access to the OS or to the management VM can execute such attacks. Regular cloud users cannot exploit the infrastructure in such a way.

4.1 Solution Design

Our solution to the raised IaaS cloud security challenge is depicted as a secure cloud server in Figure 6.

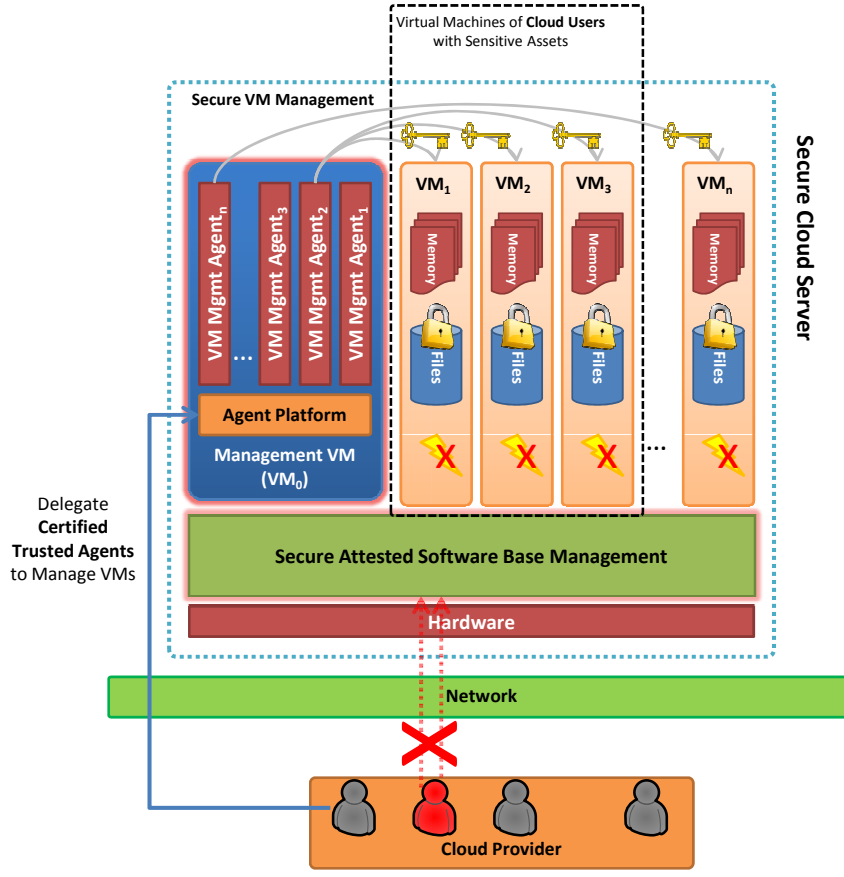


Figure 6. Secure cloud server schema

Compared to Figure 5 several modifications can be noted in Figure 6. Firstly we have introduced *the Secure Attested Software Base (SASB) management* layer. SASB is responsible for the measurement and attestation of the hardware, OS and hypervisor layers. Secondly *direct access of an administrator to the OS or to the hypervisor is excluded* and instead a special purpose VM is introduced which acts as a host platform for an Agent Platform. A physical Secure Cloud Server can be either pre-installed and pre-configured by a trusted third party or can be installed from a prepared secure installation media including all required software, namely a trusted OS, the SASB and the agent platform with the management VM. Lastly, the cloud provider, i.e. the *administrators* are enabled only to delegate Certified Trusted Agents into the Agent Platform to mediate virtual machine management

tasks; and lastly there are several *VM Management Agents* deployed in the Agent Platform where each one is managing one or several VMs on behalf of the cloud user. Moreover, disks where VMs are stored can be encrypted [42] (for example using dm-crypt [43]), so even physical access to server equipment does not disclose any data.

Such secure cloud server would provide both top level security and trust for cloud users and enable efficient management of cloud resources for the cloud provider's customers without direct access to customer's information assets.

4.2 The Secure Attested Software Base

From a technical point of view, the SASB takes advantage of the experience of the experimental acTvSM platform research prototype [44]. Consequently, the SABS design stands out from previous efforts because it makes actual use of the Intel TXT chipset's strong runtime isolation of virtual machines and also the DRTM mechanism for measuring software configurations into the TPM. Naturally, the SASB itself will also be covered by the deterministic chain-of-trust. Only this allows the hardware TPM to ensure the trust in a cloud infrastructure by enforcing binary integrity (e.g. TPM sealing) and providing strong platform identities.

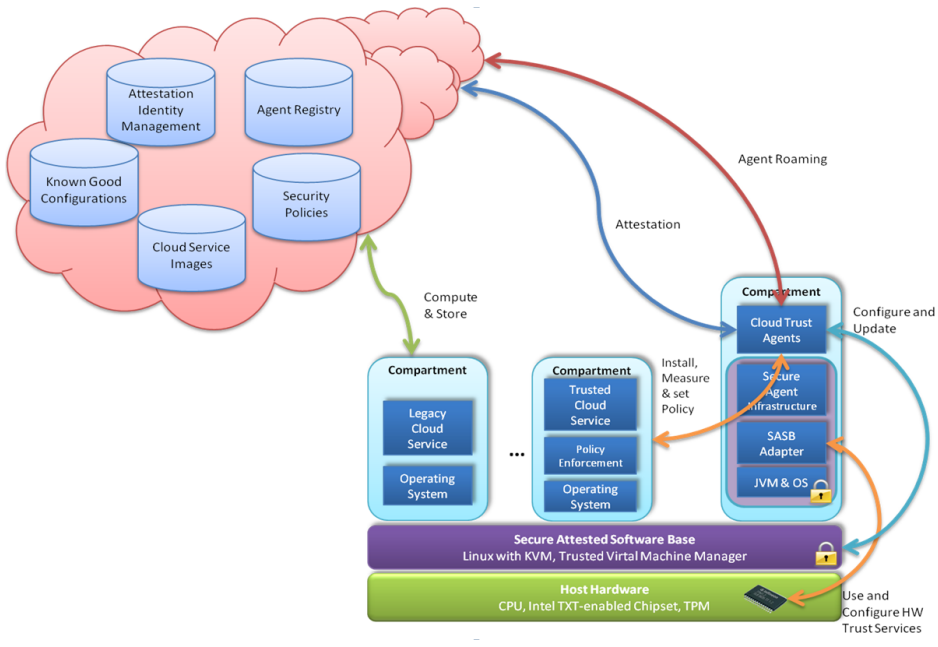


Figure 7. SASB schema

Also, on top of the Secure Attested Software Base, a Secure Agent Infrastructure (SAI) is provided as runtime environment for agents. A SASB Adapter module connects to the Trusted Virtual Machine Manager API and the TPM access interfaces of the SASB. The SAI is the core infrastructure for management of all nodes in the cloud and offers full control over each platform configuration to selected, trusted agents. These, so called certified trust agents, can also manage the configuration and security of a node throughout the cloud. Agents can be also specialized, roaming applications that automatically configure the cloud so that it fulfils and enforces the policies required by cloud services.

Among many other tasks for managing cloud operations and security, the certified trusted agents might be able to:

- Update the Secure Attested Software Base when needed. Updates will be transits from an old to a new trusted state without undefined intermediate steps.
- Acquire pseudonymous attestation identities from PrivacyCAs and revocation services as required by the TPM. They will therefore help automate the handling of node identities in compliance with TCG standards.
- Offer an attestation of service or the platform to allow remote agents to decide on the trustworthiness of a given node.
- Query known-good-software-configuration databases to determine the trustworthiness of TPM-signed attestation information. Decide whether a given node fulfils the needs of a given policy.
- Download, install, start and stop user services on the hosting node.
- Enforce that trusted cloud services do enforce policies. Each image that claims to subject its behaviour to a policy will come with a certificate that states that it is actually equipped with an active policy enforcement engine. This certificate will be checked before the service is run.
- Use the strong platform identities to determine the physical location and applicable data protection, security or homeland defence legislature of each node.

Therefore, the proposed architecture will improve the protection of agent execution environments, as agents are only deployed if SASB and SAI are in a trusted state. It also ensures that virtual containers are managed using certified trusted agents in an autonomous manner.

Furthermore, and perhaps most importantly, the powerful and flexible agents help overcome the additional complexity introduced by Trusted Computing by performing automatic updates and maintenance of nodes and enabling policy and trust decisions without user interactions.

5 VALIDATION AND PROTOTYPE IMPLEMENTATION

As a validation of the proposed approach we compare the advantage of agent-based virtual machine management in terms of security risk analysis and manageability

with other existing types of cloud infrastructure management. The main goal of the prototype implementation is to prove the viability of managing virtual machines using remotely deployable code, i.e. by certified trusted agents.

For our prototype we set up a testing environment comprising of an OpenVZ virtualization platform [47] installed on a TPM-enabled server. OpenVZ provides container-based virtualization which is similar to FreeBSD jails, Solaris zones (containers) or Linux-VServer. In addition, OpenVZ enables to execute emulated virtual machines using KVM [7]. In order to take advantage of virtualization management layer we used Proxmox (PVE) [48] which, among other advanced features, provides an API [49] to manage containers and virtual machines. According to our initial design we have deployed an agent platform (AP) into a separate container which we call the management container. The AP was configured to enable upload and execution of certified trusted agents.

For validation purposes let us consider a situation in which a malicious insider is trying to enact an illegal action against VMs under his/her supervision. We will compare three different approaches of cloud infrastructure management as depicted in Figure 8, namely:

Full root access (direct OS-level access) where the administrator has full root-shell access to the hypervisor and to the underlying OS,

Web service-based access (mediated access) where an administrator has access to limited set of services exposed through and API to manage VMs and

Agent-based management (delegated access) where an administrator is using certified trusted agents to manage an infrastructure of VMs.

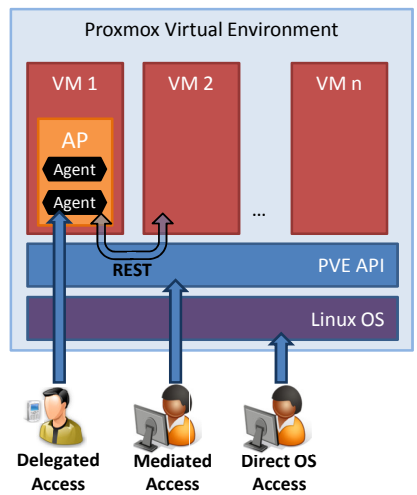


Figure 8. Three different levels of VM management

We compare the risks associated with each of these management approaches for different threat scenarios in Table 1. The risk is expressed as Low, Medium or High. Justification of the risk assessment is provided right below the table.

ID	Threat/Management	Full root	Web service-based	Agent-based
1	Illegal VM Monitoring	High	High	Low
2	Intercept VM communication	High	Low	Low
3	Unauthorized VM modification	High	Medium	Low
4	Copy data from VM or create unauthorized backup of a VM image	High	Medium	Low
5	Dump memory and copy contained VM-related information	High	Low	Low

Table 1. Security threats and the risk of misuse for different types of management approaches

From Table 1, we can see that the easiest way to misuse a VM-based cloud infrastructure for malicious insider is to have full root access to underlying OS and to the hypervisor layer. Such admin has a quite easy task to illegally monitor, modify or copy VM data, dump and extract useful information (such as RSA private keys) from the system memory or intercept VM communication. Example of such misuse is presented in [36] for the Xen virtualization platform.

In a case of web service-based access the level of risk highly depends on the type, offered functionality and security settings of the exposed API. The risk of illegal VM monitoring is high because a broad range of functionalities exist to monitor and control the resources of VMs [49]. In order to intercept VM's local or network communication or to dump system memory one needs a system level access to deploy a network sniffer or to execute a memory dump command. Without root-level access and without explicitly exposing such functionalities through a web service the risk of these two threats for web service-based management approach is low (we suppose that the malicious insider has no access to networking appliances such as switches or routers). The risk of unauthorized VM modification or backup is medium because of existing API functionalities supporting such tasks.

The risk of all the five threats was evaluated as low for the agent-based cloud infrastructure management. This assertion holds of course only for high quality secure agent code. By exploiting properly configured instances of basic agent types as proposed in Table 2 we can lower the risk of basic threats (Table 1) faced when dealing with malicious insider attacks. In order to address the basic required functionality for an administrator we have developed three different types of agents which are summarized in Table 2. For each agent type, this table presents an agent name, covered functionality and configuration options which are required in order to successfully execute a configured agent instance.

Agent Name	Functionality	Configuration
Backup Agent	Backup a container with VM_ID to a remote mounted file system through a secure channel, crypt using pre-stored private key using GPG [50].	VM_ID
Monitoring Agent	Monitor given system resource. Upon exceeding TRESHOLD (e.g. resource utilization) for more than ENDURE_TIME (in seconds) for container VM_ID notify administrator.	VM_ID, TRESHOLD, ENDURE_TIME
Resource Management Agent	Monitor certain resource, identify problem and autonomously enact correction action on container VM_ID. Example might be VM disk quota monitoring. Upon insufficient disk space more space can be comitted to the VM.	VM_ID

Table 2. List of proposed types of certified trusted agents

The secure agent code is a digitally signed piece of executable code. Since agents handle security critical tasks they must be verified and authorized in order to be executed. Agents in our implementation are Java-based so we take advantage of standard JAR (Java ARchive) signing and verification abilities. JAR archives enable to attach hashes or encoded representations of the contents of the files as they were at the time of signing. A file's digest will change if and only if the file itself changes. Additionally, a JAR file can be also digitally signed. Asymmetric cryptography is used where private and public keys are used complementarily to encrypt and decrypt a piece of text or to digitally sign a resource. The public key of the signer is also attached to the archive's so called manifest file. The manifest file can contain additional information such as certification data or policy specific information. The signing of the agent code is one of the the foundations of our approach for identity and access management of certified trusted agents in cloud. The code can be signed by several entities, i.e. by its author to certify its origin or by an agent reviewer to certify that the code was reviewed and certified for safe use. The AP enables upload of authorized agents. The AP contains an Authorized Agent Access List (AAAL) which is a list of certified trusted agents (i.e. their hashes) which are authorized to be executed on the respective AP. Each uploaded agent is matched with the AAAL before execution. Since several agents are executed on the same AP to manage several VMs it is required that each agent will be also signed by the respective VM user. The private keys used for encryption are managed and stored in the server's TPM and are managed by the SASB componentsnent. Private keys are released only in the case when the integrity of the system is ensured. The SASB checks individually the integrity of all the underlying layers (OS, hypervisor and AP).

Although we have identified that the risk of managing cloud IaaS is lowest by using the proposed agent-based management access, the shortcomings of our approach are manageability and relative initial complexity of the approach deployment. In order to implement our approach there are initial tasks required to be carried out such as agent platform deployment, PKI infrastructure set up, SASB configuration, server volume encryption or agent code review and verification. We believe these tasks can be automated and the complexity of the approach deployment eliminated in the future by developing suitable tools required for the initial set-up process.

6 CONCLUSION

In this article we have proposed a new approach for cloud infrastructures (IaaS model) based on certified trusted agents. We described the motivation, concept, design, initial implementation and validation of our approach.

For cloud users our concept would specifically enable operations of those applications in the cloud which could not be migrated to a cloud before due to serious security and data confidentiality concerns. For cloud providers our concept would attract more security conscious customers thus raising adoption of cloud computing as well as enabling a wider market share of the cloud services market.

In the future we want to develop a broader range of agents for cloud infrastructure management and to research security aspects of such agents.

Acknowledgement

This publication is the result of the following projects: Research of technologies for real-time heterogeneous distributed environments with multi-modal communication support (ITMS 26240220064), Research and development of new information technologies for forecasting and solving crisis situations and for the security of population (ITMS 26240220060) supported by Operational Programme Research & Development funded by the ERDF, VEGA No. 2/0054/12, FP7-607768 REDIRNET – Emergency Responder Data Interoperability Network and APVV-0809-11 CLAN – Cloud Computing for Big Data Analytics.

REFERENCES

[1] Cloud Security Alliance. Security Guidance for Critical Areas of Focus in Cloud Computing V2.1. Available on: `https://cloudsecurityalliance.org/csaguide.pdf`, 2009.

[2] Granneman, J.: Cloud Risk Assessment and ISO 27000 Standards. Available on: `http://searchcloudsecurity.techtarget.com/tip/Cloud-risk-assessment-and-ISO-27000-standards`, September 2011.

[3] Balogh, Z.—Gatial, E.—Hluchý, L.: Objectives for Migration and Operation Support of Legacy Applications in Cloud. In: Hluchý, L., Kurdel, P., Sebestyénová, J.

(Eds.): Proceedings of the 6th International Workshop on Grid Computing for Complex Problems (GCCP 2010), Bratislava 2010, pp. 124–131, ISBN 978-80-970145-3-7.

[4] Zhang, Q.—Cheng, L.—Boutaba, R.: Cloud Computing: State-of-the-Art and Research Challenges. Journal of Internet Services and Applications, Vol. 1, 2010, No. 1, pp. 7–18.

[5] Fang, L. et al.: NIST Cloud Computing Reference Architecture. NIST SP – 500-292, September 2011, 35 pp.

[6] Xen Project. Available on: `http://www.xen.org/`, accessed in April 2013.

[7] Kernel Based Virtual Machine. Available on: `http://http://www.linux-kvm.org/`, accessed in April 2013.

[8] VMware Virtualization for Desktop, Server, Public, Private Clouds. `http://www.vmware.com/`, accessed in April 2013.

[9] Amazon Elastic Compute Cloud. Available on: `http://aws.amazon.com/ec2/`, accessed in April 2013.

[10] ElasticHosts. Available on: `http://www.elastichosts.com/`, accessed in April 2013.

[11] Bieszczad, A.—Pagurek, B.—White, T.: Mobile Agents for Network Management. IEEE Communications Surveys & Tutorials, Vol. 1, 1998, No. 1, pp. 2–9. DOI: 10.1109/COMST.1998.5340400.

[12] Papavassiliou, S.—Puliafito, A.—Tomarchio, O.—Ye, J.: Mobile Agent-Based Approach for Efficient Network Management and Resource Allocation: Framework and Applications. IEEE Journal on Selected Areas in Communications, Vol. 20, 2002, No. 4, pp. 858–872. DOI: 10.1109/JSAC.2002.1003050.

[13] Du, T. C.—Li, E. Y.—Chang, A.-P.: Mobile Agents in Distributed Network Management. Communications of the ACM, Vol. 46, 2003, No. 7, pp. 127–132. DOI: 10.1145/792704.792710.

[14] Gavalas, D.—Greenwood, D.—Ghanbari, M.—O'Mahony, M.: Advanced Network Monitoring Applications Based on Mobile/Intelligent Agent Technology. Computer Communications, Vol. 23, 2000, No. 8, pp. 720–730, ISSN 0140-3664, DOI: 10.1016/S0140-3664(99)00232-7.

[15] Guang, L.—Nigussie, E.—Rantala, P.—Isoaho, J.—Tenhunen, H.: Hierarchical Agent Monitoring Design Approach Towards Self-Aware Systems. ACM Transactions on Embedded Computing Systems (TECS), Vol. 9, 2010, No. 3, 25 pp.

[16] Rocha, F.—Abreu, S.—Correia, M.: The Final Frontier: Confidentiality and Privacy in the Cloud. Computer, Vol. 44, 2011, No. 9, pp. 44–50.

[17] Chen, G.—Lu, J.—Huang, J.—Wu, Z.: SaaAS – The Mobile Agent Based Service for Cloud Computing in Internet Environment. IEEE Proceedings of the Sixth International Conference on Natural Computation, Yantai, Shandong 2010, pp. 2935–2939.

[18] Taneja, N.—Taneja, A.: An Agent Based Proactive System Management in the Cloud. 2011, AMCIS 2011 Proceedings, p. 441.

[19] Pfitzmann, B. et al.: The Perseus System Architecture. IBM Technical Report, 2001.

[20] Garfinkel, T.—Pfaff, B.—Chow, J.—Rosenblum, M.—Boneh, D.: A Virtual Machine-Based Platform for Trusted Computing. Proceedings of the 19th ACM

Symposium on Operating Systems Principles (SOSP '03), ACM, New York 2003, pp. 193–206.

[21] Singaravelu, L.—Pu, C.—Härtig, H.—Helmuth, C.: Reducing TCB Complexity for Security-Sensitive Applications: Three Case Studies. Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006 (EuroSys '06), ACM, New York 2006, pp. 161–174.

[22] England, P.—Lampson, B.—Manferdelli, J.—Peinado, M.—Willman, B.: A Trusted Open Platform. IEEE Computer, Vol. 36, 2003, No. 7, pp. 55–62.

[23] Coker, G.—Guttman, J.—Loscocco, P.—Sheehy, J.—Sniffen, B.: Attestation: Evidence and Trust. In: Chen, L., Ryan, M. D., Wang, G. (Eds.): Proceedings of the 10th International Conference on Information and Communications Security (ICICS '08), Springer-Verlag, Berlin, Heidelberg 2008, pp. 1–18, ISBN: 978-3-540-88624-2, DOI: 10.1007/978-3-540-88625-9-1.

[24] Schiffman, J.—Moyer, T.—Shal, C.—Jaeger, T.—McDaniel, P.: Justifying Integrity Using a Virtual Machine Verifier. Proceedings of the Computer Security Applications Conference (ACSAC '09), IEEE Computer Society, Washington, DC, USA 2009, pp. 83–92.

[25] Cabuk, S.—Chen, L.—Plaquin, D.—Ryan, M.: Trusted Integrity Measurement and Reporting for Virtualized Platforms. In: Chen, L., Yung, M. (Eds.): First International Conference on Trusted Systems (INTRUST 2009), LNCS, Vol. 6163, 2010, pp. 180–196.

[26] Catuogno, L.—Dmitrienko, A.—Eriksson, K.—Kuhlmann, D.—Ramunno, G.—Sadeghi, A.—Schulz, S.—Schunter, M.—Winandy, M.—Zhan, J.: Trusted Virtual Domains – Design, Implementation and Lessons Learned. Proceedings of the First International Conference on Trusted Systems (INTRUST 2009), LNCS, Vol. 6163, 2010, pp. 156–179.

[27] Santos, N.—Gummadi, K. P.—Rodrigues, R.: Towards Trusted Cloud Computing. Proceedings of the 4th USENIX Workshop on Hot Topics in Security (HotSec 09), August 2009, Montreal, Canada.

[28] Krautheim, F. J.—Phatak, D. S.—Sherman, A. T.: Introducing the Trusted Virtual Environment Module: A New Mechanism for Rooting Trust in Cloud Computing. In: Acquisti, A., Smith, S. W., Sadeghi, A.-R. (Eds.): TRUST 2010, Springer, LNCS, Vol. 6101, 2010, pp. 211–227.

[29] Brown, A.—Chase, J. S.: Trusted Platform-as-a-Service: A Foundation for Trustworthy Cloud-Hosted Applications. Proceedings of the 3rd ACM Workshop on Cloud Computing Security (CCSW '11), ACM, New York 2011, pp. 15–20, ISBN: 978-1-4503-1004-8, DOI: 10.1145/2046660.2046665.

[30] Berger, S.—Cáceres, R.—Pendarakis, D.—Sailer, R.—Valdez, E.—Perez, R.—Schildhauer, W.—Srinivasan, D.: TVDc: Managing Security in the Trusted Virtual Datacenter. ACM SIGOPS Operating Systems Review, Vol. 42, 2008, No. 1, pp. 40–47.

[31] Azab, A. M.—Ning, P.—Zhang, X.: SICE: A Hardware-Level Strongly Isolated Computing Environment for x86 Multi-Core Platforms. Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS 2011), pp. 375–388.

[32] Wallom, D.—Turilli, M.—Martin, A.—Raun, A.—Taylor, G.—Hargreaves, N.—McMoran, A.: MyTrustedCloud: Trusted Cloud Infrastructure for Security-Critical Computation and Data Management. Proceedings of the 2011 3rd IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2011), pp. 247–254, ISBN: 9780769546223, DOI: 10.1109/CloudCom.2011.41.

[33] Vasudevan, A.—McCune, J. M.—Qu, N.—van Doorn, L.—Perrig, A.: Requirements for an Integrity-Protected Hypervisor on the x86 Hardware Virtualized Architecture. Jun 2010, Proceedings of the 3rd International Conference on Trust and Trustworthy Computing (Trust 2010), 15 pp.

[34] McCune, J. M.—Parno, B.—Perrig, A.—Reiter, M. K.—Isozaki, H.: Flicker: An Execution Infrastructure for TCB Minimization. Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems 2008 (EuroSys '08), ACM, Glasgow, Scotland, UK 2008, pp. 315–328.

[35] McCune, J. M.—Li, Y.—Qu, N.—Zhou, Z.—Datta, A.—Gligor, V.—Perrig, A.: TrustVisor: Efficient TCB Reduction and Attestation. Proceedings of the IEEE Symposium on Security and Privacy, Oakland, May 2010, pp. 143–158.

[36] Rocha, F.—Correia, M.: Lucy in the Sky without Diamonds: Stealing Confidential Data in the Cloud. Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSNW '11), Hong Kong, China, June 2011, pp. 129–134.

[37] TCG TPM Specification Version 1.2 Revision 116. Trusted Computing Group, 2011, `http://www.trustedcomputinggroup.com`.

[38] Marchesini, J.—Smith, S.—Wild, O.—Macdonald, R.: Experimenting with TCPA/TCG Hardware, Or: How I Learned to Stop Worrying and Love the Bear. Technical Report TR2003-476, Department of Computer Science/Dartmouth PKI Lab, Dartmouth College, 2003.

[39] Sailer, R.—Zhang, X.—Jaeger, T.—van Doorn, L.: Design and Implementation of a TCG-Based Integrity Measurement Architecture. Proceedings of the 13th USENIX Security Symposium, San Diego, CA, USA 2004, 17 pp.

[40] Cáceres, R.—Carter, C.—Narayanaswami, C.—Raghunath, M.: Reincarnating PCs with Portable SoulPads. Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05), ACM, New York, NY, USA 2005, pp. 65–78, ISBN: 1-931971-31-5, http://doi.acm.org/10.1145/1067170.1067179.

[41] Gebhardt, C.—Tomlinson, A.: Secure Virtual Disk Images for Grid Computing. Proceedings of the 2008 Third Asia-Pacific Trusted Infrastructure Technologies Conference (APTC 2008), IEEE Computer Society, October 2008, pp. 19–29, DOI: 10.1109/APTC.2008.17.

[42] Fruhwirth, C.: New Methods in Hard Disk Encryption. Vienna University of Technology, July 18, 2005. Available on: `http://clemens.endorphin.org/nmihde/nmihde-A4-ds.pdf`.

[43] Peters, M.: Encrypting Partitions Using DM-Crypt and the 2.6 Series Kernel. Available on: http://archive09.linux.com/feature/36596.

[44] Toegl, R.—Pirker, M.—Gissing, M.: acTvSM: A Dynamic Virtualization Platform for Enforcement of Application Integrity. Proceedings of the Second International Conference on Trusted Systems (INTRUST 2010), LNCS, Vol. 6802, 2011, pp. 326–345.

[45] Popek, G. J.—Goldberg, R. P.: Formal Requirements for Virtualizable Third Generation Architectures. Communications of the ACM, Vol. 17, 1974, No. 7, pp. 412–421.

[46] Adams, K.—Agesen, O.: A Comparison of Software and Hardware Techniques for x86 Virtualization. Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS XII), ACM, San Jose, CA, USA, October 2006, pp. 2–13, ISBN: 1-59593-451-0, DOI: 10.1145/1168857.1168860.

[47] Kolyshkin, K.: Virtualization in Linux. September 1, 2006, http://download.openvz.org/doc/openvz-intro.pdf.

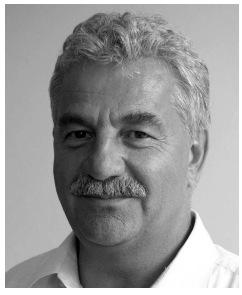
[48] Proxmox Virtual Environment (Server-Virtualization with KVM and Containers). Available on: http://www.proxmox.com/proxmox-ve.

[49] Proxmox VE API Documentation. Available on: http://pve.proxmox.com/pve2-api-doc/.

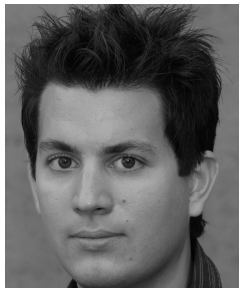
[50] The GNU Privacy Guard – GnuPG.org. Available on: http://www.gnupg.org/.

Zoltán Balogh is a senior researcher at II SAS. He received his Dipl.-Eng. (M. Sc.) in quantitative management and informatics. He defended his Ph. D. dissertation dealing with knowledge-based estimation of service performances in 2007. He is an author and co-author of more than 100 peer-reviewed publications. He has participated in several 5th, 6th and 7th EU FP projects as well as in national projects as a team member and a team leader. His R & D topics are cloud computing, knowledge engineering, service-based and agent-based systems. He is a co-author of the Secure Agent Infrastructure (SAI). He is also a member of the Research Council at II SAS and an industry consultant.

Emil Gatial is a researcher at II SAS since 2003. He received his Dipl.-Eng. (M. Sc.) in information and management systems. He defended his dissertation dealing with coordinating distributed execution of agents for trusted information collection in 2010. He is an author and co-author of many scientific papers. He is an author of Secure Agent Infrastructure (SAI) which was scientifically validated by successful Ph. D. defence and verified within scope of several projects. His R & D topics are agent-based systems and design of advance user interfaces. His interest is programming of mobile devices and single-board computers.



Ladislav Hluchý is the director of II SAS and also the Head of the Department of Parallel and Distributed Computing at the Institute. He received M. Sc. and Ph. D. degrees, both in the computer science. He is R & D Project Manager, Work-package Leader and Coordinator in a number of 4^{th}, 5^{th}, 6^{th} and 7^{th} EU FP projects as well as Slovak R & D projects (VEGA, APVT, SPVV). His R & D topics are focused on distributed computing, large scale applications and knowledge engineering. He is a member of IEEE, e-IRG, EGI Council, the Editor-in-Chief of the CC journal Computing and Informatics. He is also (co-)author of scientific books and more than 400 scientific papers, contributions and invited lectures at international scientific conferences and workshops. He is a supervisor and consultant for Ph. D. study at the Slovak University of Technology in Bratislava.



Ronald Toegl studied telematics at Graz University of Technology, wrote his Master's Thesis at the Institute for Communications and Navigation, German Aerospace Center (DLR) and received his Dipl.-Eng. (M. Sc.) with Distinction in 2006. From 2006 to 2007 he was Junior Researcher at the Virtual Vehicle Competence Center (VIF) in Graz and joined IAIK in 2007 as a Research Assistant for the Open-TC-project. Now a Senior Researcher, he is working on the design, application and verification of Interfaces and Protocols in Trusted Computing. He has (co-)authored two dozen peer-reviewed papers in international conferences and workshops and is also the Spec-Lead for JSR 321, where the API for Trusted Computing API in Java is being standardized.

Martin Pirker obtained his Dipl.-Eng. (M. Sc.) degree in telematics from Graz University of Technology. He experienced all the birth pangs of Trusted Computing technology when he joined the Institute for Applied Information Processing and Communications (IAIK) in 2005 for the FP6-IP OpenTC project. Since then he participated in several EU (Secricom, SEPIA) and national (Topas, acTvSM) research projects, (co-) authored and contributed to several Open Source security software packages and academic publications. Current research interests focus on enforcing data privacy and security properties with state-of-the-art security hardware support, on PC platforms as well as recent mobile devices.

Daniel Hein studied at Graz University of Technology and ETH Zurich, and received a master degree with distinction in computer science and electrical engineering from Graz University of Technology. Currently, he works as a university assistant at Graz University of Technology, where he contributed to several European and national research projects in addition to his teaching activities. His research interests include computer security, trusted computing, hardware security, and information flow security. He published more than 15 papers in these areas.