Data Security Perspectives in the Framework of Cloud Governance

Adrian Copie[1,2], Teodor-Florin Fortiş[1,2], and Victor Ion Munteanu[1,2]

[1] Institute e-Austria, Timişoara
bvd. V.Pârvan 4, Timişoara, Romania
[2] West University of Timişoara,
Faculty of Mathematics and Informatics,
Department of Computer Science
bvd. V.Pârvan 4, Timişoara, Romania
{adrian.copie,fortis,vmunteanu}@info.uvt.ro

Abstract. The adoption of Cloud Computing paradigm by the Small and Medium Enterprises allows them to associate and to create virtualized forms of enterprises or clusters that better sustain the competition with large enterprises sharing the same markets. In the same time the lack of security standards in Cloud Computing generates reluctance from the Small and Medium Enterprises in fully move their activities in the Cloud. We have proposed a Cloud Governance architecture which relies on mOSAIC project's cloud management solution called Cloud Agency, implemented as a multi-agent system. The Cloud Governance solution is based on various datastores that manage the data produced and consumed during the services lifecycle. This paper focuses on determining the requirements that must be met by the various databases that compound the most complex datastore from the proposed architecture, called Service Datastore, together with emphasizing the threats and security risks that the individual database entities must face.

Keywords: Cloud Computing, Cloud Governance, Datastores, Databases, Security in the Cloud.

1 Introduction

Recently, Cloud Computing has become an omnipresent paradigm which offers a plethora of advantages and opportunities from the technological and economical points of view. It has the potential to revolutionize the way we see and do business by providing clear economic incentives (e.g. the *pay per use* economic model) [1, 2]. Its approach on selling and renting cloud resources and services is enabled through essential characteristics like on-demand self-service, broad network access, resource pooling (multi tenancy), rapid elasticity, measured services, service orientation, strong fault tolerance and loosely coupled services [3–5].

The diversity in proprietary technologies deployed by the different cloud vendors has caused what is known as *vendor lock-in*, slowing down the adoption

I. Caragiannis et al. (Eds.): Euro-Par 2012 Workshops, LNCS 7640, pp. 24–33, 2013.

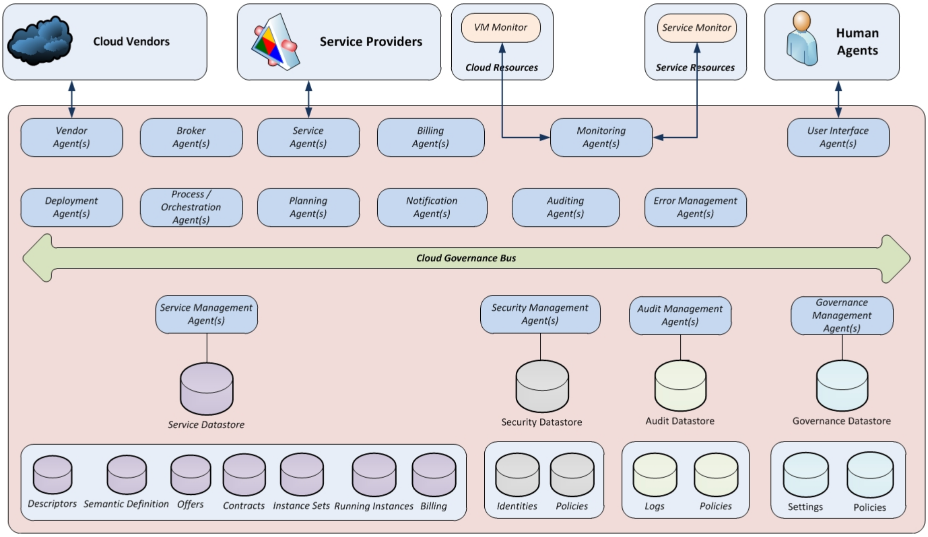

Fig. 1. Multi-Agent Governance Architecture

process. Different platform-as-a-service (PaaS) solutions, including mOSAIC[1], Cloud Foundry[2], OpenShift[3], Morfeo 4CaaSt[4], ActiveState's Stackato[5], WSO2 Stratos[6], attempt to mitigate this problem.

Available PaaS solutions have enhanced cloud adoption by allowing the development of multi-cloud services but, because of the lack of features of current PaaS solutions like service lifecycle management, service discovery, contract brokering, and others, there is a high degree of fragmentation, most of the services being run rather in isolation [6].

Cloud adoption is an ongoing process which gives to the SMEs the opportunity to develop highly specialized solutions or to cooperate and work together as a virtual enterprise in order to offer complex solutions, tailored for their customers' needs. To build effective solutions inside virtual enterprises, it is necessary to implement appropriate cloud management and cloud government policies [7] acting together as distinct components of a cloud infrastructure.

In the papers from Distributed Management Task Force (DMTF, [8, 9]) there are emphasized the cloud management and cloud governance requirements, having them in close relation and describing their roles, which allow us to propose a security oriented cloud governance architecture.

Our paper focuses on security aspects regarding lifecycle support for the proposed cloud governance architecture and it is structured as follows.

[1] http://www.mosaic-cloud.eu/
[2] http://cloudfoundry.org/
[3] https://openshift.redhat.com/app/
[4] http://4caast.morfeo-project.org/
[5] http://www.activestate.com/stackato
[6] http://wso2.com/cloud/stratos/

The results of our paper are covered in Sections 2 and 3. Finally, Section 4 draws conclusions and discusses future work.

2 Governance Datastores

A multi-agent cloud governance architecture, as depicted in Figure 1,

targets service management, security and privacy management, service lifecycle management and monitoring by using agent technologies. It is composed of four interdependent subsystems: *Service Management*, *Security Management*, *Audit Management* and *Governance Management.*

This architecture was designed with several issues in mind:

- Compliance with business standards;
- Complete service management (registering, updating, searching, removing);
- Automated service lifecycle (instantiation/commissioning, contracting, retirement etc.);
- Security and privacy management that is compliant with government laws.

The cloud management component relies on mOSAIC's Cloud Agency, implemented in the framework of the 'Open Source API and platform for multiple Clouds' (mOSAIC) project. The Cloud Agency is a multi-agent system that performs resource provisioning, monitoring and reconfiguration and is accessible through a REpresentational State Transfer (REST) interface [10–12].

Table 1. Components of the Service Datastores

Component	Description	Stored data type	Implemented as
Service Descriptor	Functional and non-functional requirements for the services	Structured, searchable	RDBMS
Semantic Definitions	Semantic description of the services	Formatted data, searchable through SPARQL	RDF datastore
Offers	Describes the service parameters and the price	Unstructured information	Graph database
Contracts	The agreement between the service provider and service consumer	Unstructured information	Graph database
Instance Sets	Aggregation of trading, billing and deploying information	Unstructured information	Graph database
Running Instances	Information about the live service instances	Unstructured information	Graph database
Billing	Billing information generated as a result of service usage	Unstructured information	Graph database

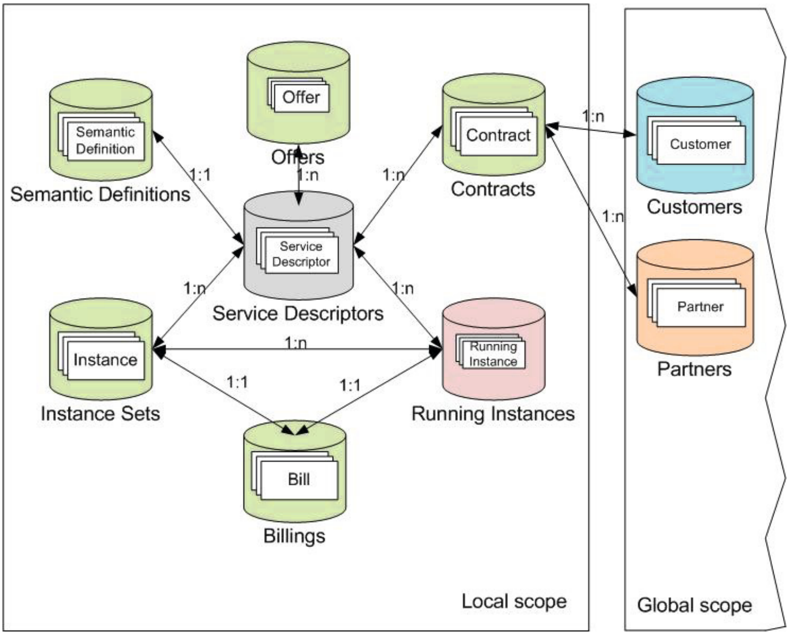

Fig. 2. Service Datastore

The information persisted inside the Cloud Governance system is extremely heterogeneous due to the fact that every agency manipulates different data structures, leading to a complex mesh of databases. By data integration through Enterprise Integration Patterns, all the information is combined and presented to the governance component in a consumable format. The Cloud Governance Bus (as identified in Figure 1) allows the messages exchange and the data aggregation through the agencies related to the different datastores. More than being a complex storage in terms of number of databases, also their types are varying in order to offer best performances for specific operations.

2.1 Service Management Datastore

This storage system has the role to maintain a coherent information about all the services registered inside the Cloud Governance environment. The information stored in this datastore describes the services from syntactic and semantic point of view, but it is also needed by other processes like offers consulting, contract establishment, billing generation or maintaining accurate data about the running services instances.

Because Cloud Governance is an evolution of the Service Oriented Architecture (SOA) governance, it relies also on a Service Repository model involving methods called by the providers to register their services and methods called by the consumers in order to discover the existing services. The structure and the content of the information inside the Service Datastore is grouped in several components described in Table 1 along with the data types they store and also their practical implementation in terms of used database type. Figure 2 depicts the architecture and the relationship between all the components of the Service Datastore. All these components are interdependent, they have to be kept synchronized and up to date all the time to provide accurate data. More than this they are related also with two global catalogues namely *Customers* and *Partners* used in the same time by other Cloud Governance modules.

2.2 Security Management Datastore

The access to cloud services must be performed in a secured way, only authorized parts having the rights to use them fully or in a granular way, based on specific access control lists or policies. The Security Manager agent is responsible for the credentials management, the authentication and authorization processes and the generation of the security tokens that will be further used in accessing the resources by the other agents in the Cloud Governance environment.

All the credentials and security policies governing the cloud services are stored inside the Security Datastore. The identities of all the service consumers together with their representative credentials like passwords and secret keys used to access specific resources are kept in the Security database. This information must be optimized for reading and also is required to be highly searchable and fast, in order to speed up the authentication process, our approach being a RDBMS database.

2.3 Configuration Management Datastore

Every service provider has its policies and constraints and have to publish them in the services catalogue in order to be taken into account during the service execution phase. The information related to the policies and constraints is unstructured due to the multitude of service types that could cooperate. The data is usually related to the service functionality, guaranteed parameters of services, security policies, configuration parameters of the virtual machines on which agents are executed and many more.

In the absence of the governance policies or the functional constraints, the governance process can not happen, so this is a critical repository which requires high availability and reliability. Also it must be searchable and editable which finally lead us to a graph database.

2.4 Monitoring and Audit Management Datastore

The current cloud management solution based on mOSAICs Cloud Agency is not enough in order to monitor the registered cloud services because it acts at the infrastructure level. This is why the Cloud Governance system is responsible for this task through its Audit and Monitoring Management agent. The Audit Datastore must face to a high number of writes since all the services taking part in the cloud governance system are continuously generating data related to audit actions or information resulted from the monitoring process. The goal is to provide a highly available for writes datastore. All the data generated as the result of auditing or monitoring are stored as they come, the only operation at this phase is the insertion in the database.

In the same time, this datastore must offer the possibility of analysing the data and performing different predefined actions based on the result of the data analysis (e.g. intrusion detection, insufficient application resources, stopped services, etc). The nature of the stored data, the number of records and the complexity of the generated information qualify this datastore to be considered as a Big Data store. The data is unstructured and the volume depends basically on the number and type of monitored service parameters like the number of the connections to a database or the number of elements in a message queue at a specific moment, the sampling frequency and the number of actions that require auditing. Special strategies need to be defined in order to deal with the content of this storage, to search and process the data and produce valuable results.

Because the data volume is high, the store have to support scalability to accommodate with the amount of generated data. Also the store must be highly available for writes and must easily scale horizontally, that is the reason why we have chosen a key-value datastore for our implementation.

3 Requirements for Service Datastore Security

Security is the most important concern that prevents SMEs to fully adopt Cloud Computing. The Cloud Governance system handles extremely sensitive data belonging to the SMEs and their partners, therefore the security requirements are very strict and they must be implemented and respected regardless the functional component in the governance environment. The Service Datastore has a complex structure built from many different databases holding different data types. Every database type has inherent security issues that must be first understood and then addressed individually. In the following sections, the particular security threats will be discussed and solutions in order to mitigate the possible risks are proposed.

3.1 Service Datastore

Services Repository. This repository has to be highly readable as it must face a large number of requests related to the discovery process. The number of writes is smaller than the number of reads since new services registration occurs much seldom than usual interrogations.

The Service Descriptors must provide searchability, which must be enough flexible to respond to various search criteria. These requirements lead to a RDBMS solution which can be implemented in the public cloud through many existent NewSQL solutions, as well in the private cloud by using a relational database optimized for reading.

This kind of storage comes with some possible threats and risks that must be taken into account at the implementation phase. Wherever the database would be (in the public or private cloud), it is not directly exposed to attacks, but indirectly through carefully chosen malicious input. The best known attack of this type is the SQL injection which acts to the Data Manipulation Language (DML) as well to the Data Definition Language (DDL) and could alter or even destroy the entire database content or even compromise the entire service.

Semantic Definitions. The data items contain both data itself together with their representation schema and the XPath queries contain the values to be queried for and the query itself. Sometimes, this data and code mixture could be exploited by malicious text chunks intercalated among the XML items, just like in the SQL injection technique.

The XML parsers used to access the RDF repositories are also a source of possible security risks due to their vulnerabilities to various exploits. For example the SAX parsers are less prone to vulnerabilities due to the fact that they usually don't support the full XML implementations. However their serial reading, with data superseding, leads to less control over the invalid or unclosed XML.

Using the same analogy with the SQL injection, the prevention for the XML injection consists in a better input validation, looking for specific code patterns, checking not only the data type itself but other characteristics like format, length and content. Data validation at both endpoints is also required in order to be sure that the code that reach the server has no malicious potential.

The RDF stores are implemented as in-memory databases, native disk storages or support relational databases plugins. The users management for these persistence approaches could bring supplementary threats over the Service Datastore and they should be seriously treated at multiple levels.

Offers, Contracts, Instances, Running Instances and Bills Databases. The graph database, like other NoSQL databases, suffers from different kinds of risks and vulnerabilities that must be understood and properly combated. This type of database has a high level of horizontal scalability, so it carries with it the issues encountered in the distributed systems like connection pollution and Man-in-the-middle (MITM) attacks.

Even if the installation is behind a firewall, these risks are present and various defending techniques like Public Key Infrastructures (PKI), stronger mutual authentication between nodes or second channel for verification should be added. Because the information inside those databases is in some cases formatted using XML schemas, data manipulation is also a potential risk.

Every database support inputs from different components, which leads to supplementary data validation procedures to enhance the security and mitigate code injections. Many graph databases does not come with an individual authentication mechanisms, relaying in most cases on the host operating system user management. This is usually not enough because of the granularity lack, most permission being at the file level. The applications using the database often have to deal with the Access Control Lists or other access policies.

Table 2. Security Controls that apply to the Service Datastore

Security Control	Description
Assets Management	Service Datastore could reside partially in the private and in the public cloud. However, for the databases that are hosted in the private cloud it must be possible to manage all the infrastructure components (physical or virtual machines, networks, software)
Data/Storage Security	It must be possible to store data in an encrypted format. In the same time, it must be possible to store data in separate locations as special requirements came from the service consumers.
Endpoint Security	Every endpoint composing the Service Datastore must be fully securable, including restrictions related to protocols and access devices.
Network Security	It must be possible to secure the traffic at the switch, router and packet levels. Also if the public cloud is selected to host various storage systems, appropriate cloud providers must be chosen, that can make the proof of an adequate security support.
Workload and Service Management	The services must be configured, deployed and monitored according to defined security policies and customer license agreements

Table 3. Security Patterns that apply to the Service Datastore

Security Pattern	Description
Identity Management	Having many database components as building blocks of the Service Datastore and taking into account that they must communicate permanently, an identity management mechanism is required in order to provide security tokens that will be used internally, wherever necessary, to prove the requester's identity.
Access Management	The security tokens must be examined and determined what level access is allowed over a specified resource.
Configuration Management	The databases compounding the Service Datastore could be installed in physical or virtual configurations, running on-premise or in cloud. It is necessary a mechanism that is able to federate this configuration data and offer it as an aggregated information in order to be used across multiple domains.

Depending on the access level of a hypothetical attacker, at the database level or at the file-system level, the records could be corrupted, destroyed or stolen, the worst scenario being the corruption or destruction of the entire database.

3.2 Security Controls and Patterns in Service Datastore

According to [13], different security controls are necessary to secure the Cloud Governance system. This environment relies on different specialized databases and the Service Datastore is the most sophisticated. Due to its complexity, various cloud security controls could be applied. We have identified some of them, namely *Access Management*, *Data/Storage Security*, *Endpoint Security*, *Network Security*, *Workload and Service Management.*

Table 2 provides a synthesis of these security controls and details their connections to the concrete Service Datastore architecture. Every identified control is described from its functional point of view.

Service Datastore is a complex mesh of databases being seen like a federation of various resources for which we have identified different security patterns in conformity with [13]. The Service Datastore components could reside in the private cloud but also in the public cloud. This is why strong security requirements must be fulfilled to assure an appropriate protection of governance information together with sensitive data belonging to the consumers of the registered cloud services. Table 3 presents some of the security patterns identified inside the Service Datastore component.

4 Conclusions and Future Work

Cloud Computing adoption allows SMEs to access new markets where they can associate in virtual enterprises or virtual clusters, being able to better sustain the competition with the large enterprises acting on the same markets. However, the lack of standardization in what concerns the Cloud Computing security prevents the SMEs to fully embrace the cloud technologies and to move their activities entirely in the cloud.

We have proposed a Cloud Governance architecture that relies on the mOSAIC's Cloud management solution that aims to solve the management and governance of the services infrastructure. The proposed governance solution is based on various datastores that manage the data produced and consumed during the life cycle of the cloud services.

Our paper focused on the most complex datastore in our governance system, the Service Datastore, analysing the requirements related to the databases type selected to hold all the data produced and consumed during the services lifecycle. In the same time we have pointed the security threats and risks facing the various databases and also proposing solutions to mitigate them. This paper is a base for our future work consisting in implementing the security mechanisms in our proposed Cloud Governance architecture.

Acknowledgments. This work was partially supported by the grant of the European Commission FP7-ICT-2009- 5-256910 (mOSAIC), FP7-REGPOT-CT-2011-284595 (HOST), and Romanian national grant PN-II-ID-PCE-2011-3-0260 (AMICAS). The views expressed in this paper do not necessarily reflect those of the corresponding projects consortium members.

References

1. Weinhardt, C., Anandasivam, A., Blau, B., Borissov, N., Meinl, T., Michalk, W., Stößer, J.: Cloud Computing–A Classification, Business Models, and Research Directions. Business and Information Systems Engineering, BISE 1(5), 391–399 (2009) ISSN: 1867-0202
2. Weinhardt, C., Anandasivam, A., Blau, B., Stößer, J.: Business Models in the Service World. IEEE IT Professional, Special Issue on Cloud Computing 11(2), 28–33 (2009) ISSN: 1520-9202
3. Mulholland, A., Pyke, J., Fingar, P.: Enterprise Cloud Computing: A Strategy Guide for Business and Technology Leaders. Meghan-Kiffer Press, Tampa (2010)
4. Gong, C., Liu, J., Zhang, Q., Chen, H., Gong, Z.: The characteristics of cloud computing. In: ICPP Workshops, pp. 275–279 (2010)
5. Mell, P., Grance, T.: The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology (September 2011), http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf
6. Wainewright, P.: Time to think about cloud governance (August 2011), http://www.zdnet.com/blog/saas/time-to-think-about-cloud-governance/1376
7. ISACA: Cobit 5 introduction (February 2012), http://www.isaca.org/COBIT/Documents/An-Introduction.pdf
8. DMTF: Architecture for managing clouds (June 2010), http://dmtf.org/sites/default/files/standards/documents/DSP-IS01021.0.0.pdf
9. DMTF: Use cases and interactions for managing clouds (June 2010), http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0103_1.0.0.pdf
10. Venticinque, S., Negru, V., Munteanu, V.I., Sandru, C., Aversa, R., Rak, M.: Negotiation policies for provisioning of cloud resources. In: Proceedings of the 4th International Conference on Agents and Artificial Intelligence, pp. 347–350. SciTePress (February 2012)
11. Venticinque, S., Aversa, R., Di Martino, B., Rak, M., Petcu, D.: A Cloud Agency for SLA Negotiation and Management. In: Guarracino, M.R., Vivien, F., Träff, J.L., Cannatoro, M., Danelutto, M., Hast, A., Perla, F., Knüpfer, A., Di Martino, B., Alexander, M. (eds.) Euro-Par 2010 Workshop. LNCS, vol. 6586, pp. 587–594. Springer, Heidelberg (2011)
12. Aversa, R., Di Martino, B., Rak, M., Venticinque, S.: Cloud agency: A mobile agent based cloud system. In: Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems, CISIS 2010, pp. 132–137. IEEE Computer Society, Washington, DC (2010)
13. Cloud Computing Use Cases Group: Cloud computing use cases white paper (July 2010)