INT J COMPUT COMMUN, ISSN 1841-9836
7(5):900-906, December, 2012.

Building a Cloud Governance Bus

V.I. Munteanu, T.-F. Fortiş, A. Copie

Victor Ion Munteanu, Teodor-Florin Fortiş, Adrian Copie
1. West University of Timişoara
Romania, Timişoara, bvd. V.Pârvan 4, and
2. Institute e-Austria, Timişoara
Romania, Timişoara, bvd. V.Pârvan 4
E-mail: vmunteanu@info.uvt.ro,
fortis@info.uvt.ro
adrian.copie@info.uvt.ro

Abstract: Thought still at its first steps, cloud governance lays the foundation upon which business innovations can be built. It fills in the gaps left by cloud providers and allows major players on the IT market to be challenged by small and medium-sized enterprises (SMEs) for their share.

At the core of cloud governance, its bus enables interaction and communication between various services and governance components. The cloud governance bus is a step forward for the enterprise service bus (ESB) into the cloud environment, addressing data integration and full implementation of enterprise integration patterns.

This paper covers current requirements for ESB migration to the cloud environment and proposes a cloud governance architecture that meets the given requirements.

Keywords: cloud governance, cloud management, cloud governance bus, enterprise integration patterns.

1 Introduction

Cloud migration is an ongoing process to which small and medium-sized enterprises (SMEs) must adhere such that they can benefit of the advantages given by its economic model. This adoption can enable them to challenge large enterprises by creating niche solutions or grouping themselves in order to provide complex applications that are tailored for their customers' needs.

According to [13], cloud computing can be summed up in five core characteristics: on-demand self-service, ubiquitous network access, location independent resource polling, rapid elasticity and pay-per-use. The last one, pay-per-use, is a clear incentive as to why cloud adoption is desired. Cloud adoption is also driven by technical characteristics like virtualization, service orientation, link with business models, strong fault tolerance, and loosely coupling, as identified in [10] .

The large amount of proprietary technologies used by cloud vendors and the lack of cloud standards has lead to the fragmentation of cloud environments making development hard. By having multiple deployment models (public, community, hybrid and private clouds), the gap is further enlarged because of the different type of policies that need to be implemented for each of them.

Several solutions that are built on-top of cloud infrastructures (IaaS) come in aid by offering flexible cloud-independent development environments and partially handling de facto things like resource provisioning, management and monitoring. Unfortunately, these platform-as-a-service (PaaS) solutions lack the functionality that is required to have a complete cloud management solution and require a set of complementary services, as exposed in [6–8].

Furthermore, current cloud applications run in isolation or in a small clusters [19] even though there is a demand for application integration at SaaS level [4, 5, 19]. This leads to the necessity of a central entity whose purpose is to enable both service and data integration and create a unitary ecosystem where applications can be easily created, managed, discovered and can easily interact one another, the necessity for cloud governance.

Cloud governance, a step forward for service oriented architecture (SOA) governance, is essential for full cloud adoption, and even the lack of a partial solution can lead to serious challenges [14]. While not part of SOA governance itself, an enterprise service bus (ESB) is a flexible connectivity infrastructure for integrating applications and services [1]. Similarly to SOA, cloud governance can benefit from the use of such a bus.

This paper focuses on defining the requirements for a cloud governance bus while providing partial solutions in the form of already available software. The remainder of the paper is organized as follows. Our motivation and related work is covered in Section 2. Section 3 introduces the mOSAIC project and its component, the Cloud Agency (CA). Our proposed cloud governance architecture is covered in section 4. The main results are presented in Section 5 and conclusions and future work are presented in Section 6.

2 Motivation and Related Work

2.1 Cloud management and governance

Cloud Management is covered in Distributed Management Task Force's (DMTF) white papers [7,8] which identify concerns and issues related to aspects of cloud service lifecycle, components in the architecture for managing clouds etc. The white papers describe management requirements in close relationship with governance ones.

The growing interest in cloud management solutions has lead to an abundance of PaaS solutions, like mOSAIC[1], OpenShift[2], Cloud Foundry[3], or Morfeo 4CaaSt[4]. However, little interest is payed o cloud governance related concerns like data and security management, logging and audit, event management and others.

A clear place for cloud governance in relation to a generic cloud management architecture is specified in [7]. Important information related to cloud governance covering Service Level Agreements (SLAs), security patterns and controls are covered in [6].

Several enterprises have taken interest in cloud governance and have integrated it as part of their PaaS solutions: enStratus[5], WSO2 Stratos[6] and Fiorano Cloud Platform[7].

2.2 Enterprise Service Bus in the cloud

Building an enterprise service bus is a challenge for any developer because of the complexity of integrating multiple services in one environment, the use of different technologies etc. There is a high variety available of ESBs ranging from commercial ones to open source ones like IBM WebSphere Message Broker[8], ORACLE ESB[9], Fuse ESB[10], Mule ESB[11], Petals ESB[12], JBoss ESB[13], OpenESB[14]. In his paper, GarcĂa-JimĂŠnez et. al. [9] compares some of the open source ESBs.

[1] http://www.mosaic-project.eu

[2] https://openshift.redhat.com/app/

[3] http://www.cloudfoundry.com/

[4] http://4caast.morfeoproject.org/

[5] http://www.enstratus.com/

[6] http://wso2.com/cloud/stratos/

[7] http://www.fiorano.com/products/ESB-enterprise-service-bus/Fiorano-Cloud-Platform.php

[8] http://www-01.ibm.com/software/integration/wbimessagebroker/

[9] http://www.oracle.com/technetwork/middleware/service-bus/overview/index.html

[10] http://fusesource.com/products/enterprise-servicemix/

[11] http://www.mulesoft.org/

[12] http://petals.ow2.org/

[13] http://www.jboss.org/jbossesb

[14] http://openesb-dev.org/

While not originally designed for the cloud, ESBs are slowly making their way into cloud environments. Some PaaS providers offer them alongside their products either built in or as a service. Some of the commercial ESB providing solutions are WSO2 Statos, Fiorano Cloud Platform, Netperspective Cloud ESB[15] and others, while open source ones are Mule ESB[16].

3 mOSAIC

mOSAIC[17] is an FP7-ICT project [12], which is developing a platform that promotes an open-source Cloud application programming interface (API) and a platform targeted for developing multi-Cloud oriented applications. Its goal is to provide enough freedom both at resource and programming level such that cloud-based services can be easily developed and deployed.

The architecture of the platform [16] is designed around the use of open and standard interfaces. Its main goal is to provide a unified Cloud programming interface which enables the flexibility needed to build inter-operable applications across different Cloud providers [15]. mOSAIC is comprised of the mOSAIC API and the Cloud Agency.

The Cloud Agency [2,17,18] is a multi-agent system that has been designed to handle resource provisioning and monitoring and also to handle reconfiguration of resources. The Cloud Agency is easily accessible to the mOSAIC platform through a REST interface. Built around a semantic engine, the Cloud Agency has capabilities that allow dynamic discovery and mapping of cloud providers. The Cloud Agency works at an IaaS level within the mOSAIC platform.

4 Cloud Governance Architecture

The proposed cloud governance architecture (Figure 1) is built in close relation with mOSAIC's Cloud Agency and is designed to offer a variety of services which complement it. This architecture closely follows DMTF's white paper [7] and is built as a multi-agent system. The Cloud Agency exposes itself within the ecosystem as services.

A clear representation of the system is depicted in Figure 1, and is composed of four subsystems: Service Management, Security Management, Audit Management and Governance Management. Each of the these subsystems is made of several agents, each agent being able to serve several of them.

The Service Management subsystem is in charge of service lifecycle management (publishing, brokering, instantiation/commissioning, etc.). Of the agents which compose it, the Service Management Agent is the most important one as it stores all service related information in the Service Datastore.

Security Management handles all security for our governance solution. The Service Management Agent is the core of this subsystem as it handles storing, retrieving, generating all the security information within the system.

The Audit Management subsystem covers all governance monitoring, ranging from cloud resource monitoring to service monitoring. It also uses a set of policies in order to notify the system or human administrator of possible errors/faults.

Governance Management manages the system based on setting and policies. It makes sure that all agents are running and that there is a sufficient number so that all systems work properly.

Several issues have been thought of when designing the system:

[15] http://www.netspective.com/netspective-cloud-esb-overview
[16] http://www.mulesoft.org/
[17] Open source API and Platform for multiple Clouds

- complete integration of cloud management (cloud resource management, scaling, monitoring, reconfiguration);
- complete service management and lifecycle related issues (including scaling, monitoring and reconfiguration);
- complete security and privacy management;
- compliance with business practices and standards.

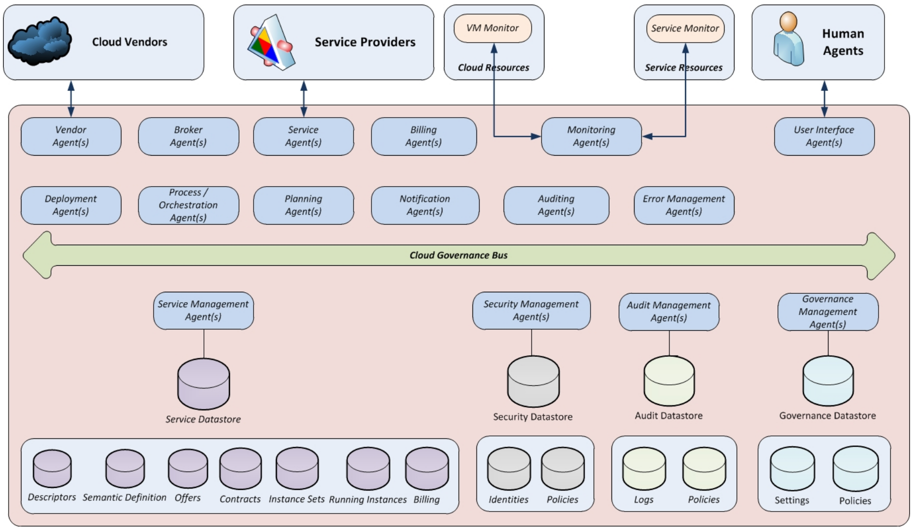

Figure 1: Cloud Governance

5 Cloud Governance Bus

The traditional role of an ESB in a SOA environment is to simplify access by hiding the complexity of the underlying system and providing a generic way for querying, accessing and interacting between services. This is achieved by handling the routing and monitoring of messages between services, handling service deployment and versioning etc.

Similarly to the ESB, a cloud governance bus (CGB) needs to be able to handle messages (queuing, sequencing), security, exceptions, protocol conversion and provide an adequate level of quality of services (QoS). Unlike traditional ESBs, our proposed CGB implements enterprise integration patterns (EIP) as well as data integration (extraction, transformation, loading, mapping) which enables easy access to datastores as well as other components like the integrated Scala implementation of ActiveMQ provided by Apache Apollo[18].

The CGB is first and foremost designed to handle the internal communication of our proposed cloud governance solution in a secure manner. Having ESB-like features is a secondary goal. As SOA allows for the development of both tightly coupled and loosely coupled services, having the opportunity to integrate them in our CGB is nice to have, but not a priority.

The following list summarizes what CGB features we would like/are a must having:

- Support both synchronous and asynchronous interaction between services
- Allow message operations like filtering, routing, translating

[18] http://activemq.apache.org/apollo/

- Allow various forms of message routing including, but not limited to, static routing, content-based routing, rules-based routing, policy-based routing
- Allow both statically and dynamically bound services
- Allow any type of data to be handled
- Handle semantic transformation if required
- Allow the possibility to define message channels
- Separate system messages from service messages
- Allow various ways in which endpoints can be defined

In his paper, Kiran Kanetkar discusses several functions that an ESB must handle [11]: routing, transformation, adaptation, messaging, orchestration, UDDI registry, security, consumer integration, service integration, metrics and management, and B2B. However, for building our cloud governance system, we only need to handle the most important functions as well as EIP.

In [3], Rob Barry identifies several problems an ESB has to face when being deployed in a cloud environment. Because of the various deployment environments (public, hybrid or private clouds) an ESB must adopt specific security policies (encryption) when dealing with the messages or authentication within the system. Another issue is the latency that can arise from sending messages between various clouds and the transport protocols the ESB needs to know.

5.1 Using Akka and Apache Camel

In order to address the issues related with building a cloud governance bus, two technologies that can cover them were identified, solutions that are event-driven and enable EIP and data integration. One of them, Akka[19], is an event-driven middleware in Scala[20]. While not a traditional, FIPA compliant, multi-agent system, Akka can be used successfully for building high performance and reliable distributed applications.

Akka's architecture allows easy mapping of agents to its Actor system. Its event driven system allows building reactive agents, facilitating them with mailboxes, another feature needed for an CGB. Akka's high-performance, self-healing, transparent-distributed system is complemented by features like support for various development libraries that enhance it like REST, Comet, Spring, Guice, Lift, Apache Came$^{\text{TM}}$, Persistence and AQMP libraries.

Akka's Apache Camel$^{\text{TM}}$ module allows easy integration with it. Apache Camel$^{\text{TM}}$ is a versatile open-source integration framework based on known Enterprise Integration Patterns.

It enhances our CGB by enabling the definition of routing and mediation rules in a variety of languages. It can be easily integrated into any kind of transport of messaging model that our CGB employs, and enhances or cloud governance architecture's ability to communicate with 3rd party applications and partners.

6 Conclusions and Future Work

Unlike large enterprises which have the resources (financial and otherwise) to build and maintain their own infrastructure, SMEs find themselves lacking and looking elsewhere for support. That support is found in the Cloud, where they can delegate infrastructure management to cloud

[19] http://akka.io/

[20] http://www.scala-lang.org/

providers benefiting from the given pay per use economic model. However they are somewhat limited and need a governance solution that enables them to group and provide complex, targeted services tailored for their customers' needs.

Cloud governance is complementary to cloud management through the services it provides. By having a cloud governance bus as the core of our cloud governance architecture, we enable a new approach to business integration and to building a highly complex and business oriented ecosystem.

This paper tried to cover requirements for a cloud governance bus from the perspective of our proposed governance architecture. Future work will cover patterns for building highly interdependent cloud services by using our bus to route and translate their messages.

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.

Bibliography

[1] Wohl Associates. SOA governance. An IBM white paper. White paper. 2006. Available on-line at: http://www-01.ibm.com/software/solutions/soa/Amy_Wohl_SOA_Governance_Analyst_White_Paper.p

[2] R. Aversa, B. Di Martino, M. Rak, and S. Venticinque. Cloud Agency: A Mobile Agent Based Cloud System. In *Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems*, CISIS '10, pages 132–137, Washington, DC, USA, 2010. IEEE Computer Society.

[3] R. Barry. ESBs in the cloud: Tricky in the early going. News. June 2010. Available on-line at: http://searchsoa.techtarget.com/news/1514427/ESBs-in-the-cloud-Tricky-in-the-early-going.

[4] S. Bennett, T. Erl, C. Gee, R. Laird, A. T. Manes, R. Schneider, L. Shuster, A. Tost, and C. Venable. SOA Governance: Governing Shared Services On-Premise & in the Cloud. Prentice Hall/PearsonPTR, 2011.

[5] T. Cecere. Five steps to creating a governance framework for cloud security. Cloud Computing Journal. November 2011. Available on-line at: http://cloudcomputing.sys-con.com/node/2073041.

[6] Cloud Computing Use Cases Group. Cloud computing use cases white paper. July 2010. Available on-line at: http://opencloudmanifesto.org/Cloud_Computing_Use_Cases_Whitepaper-4_0.pdf.

[7] DMTF. Architecture for Managing Clouds. June 2010. Available on-line at: http://dmtf.org/sites/default/files/standards/documents/DSP-IS0102_1.0.0.pdf.

[8] DMTF. Use Cases and Interactions for Managing Clouds. June 2010. Available on-line at: http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0103_1.0.0.pdf.

[9] F.J. Garcia-Jimenez, M.A. Martinez-Carreras, and A.F. Gomez-Skarmeta. Evaluating open source enterprise service bus. In *e-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on*, pages 284 –291, nov. 2010.

[10] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong. The characteristics of cloud computing. In Wang-Chien Lee and Xin Yuan, editors, *ICPP Workshops*, pages 275–279. IEEE Computer Society, 2010.

[11] K. Kanetkar. A roadmap to building an ESB. May 2006. Available on-line at: http://www.saterisystems.com/Docs/whitepapers/Roadmap to Building an ESB.pdf.

[12] mOSAIC Consortium. The mOSAIC project. 2010. Available on-line at: http://mosaic-cloud.eu/.

[13] A. Mulholland, J. Pyke, and P. Fingar. Enterprise Cloud Computing: A Strategy Guide for Business and Technology Leaders. Meghan-Kiffer Press, Tampa, FL, USA, 2010.

[14] P. Mynampati. Soa governance: Examples of service life cycle management processes. November 2008. Available on-line at: http://www.ibm.com/developerworks/webservices/library/ws-soa-governance/index.html.

[15] D. Petcu, C. Crăciun, M. Neagul, I. Lazkanotegi, and M. Rak. Building an interoperability API for sky computing. In *High Performance Computing and Simulation (HPCS), 2011 International Conference on*, pages 405–411, july 2011.

[16] D. Petcu, S. Panica, and M. Neagul. From grid computing towards sky computing. case study for earth observation. Proceedings Cracow Grid Workshop 2010, pages 11–20. Academic Computer Center, Poland, 2010.

[17] S. Venticinque, R. Aversa, B. Di Martino, M. Rak, and D. Petcu. A cloud agency for SLA negotiation and management. In *Proceedings of the 2010 conference on Parallel processing*, Euro-Par 2010, pages 587–594, Berlin, Heidelberg, 2011. Springer-Verlag.

[18] S. Venticinque, R. Aversa, B. Di Martino, and D. Petcu. Agent based Cloud Provisioning and Management - Design and Prototypal Implementation. In *CLOSER 2010*, pages 184–191, 2011.

[19] P. Wainewright. Time to think about cloud governance. August 2011. Available on-line at: http://www.zdnet.com/blog/saas/time-to-think-about-cloud-governance/1376.