Interoperability Standards for Cloud Architecture

Claus Pahl, Li Zhang and Frank Fowley
IC4, Dublin City University, Dublin, Ireland

Keywords: Cloud Computing, Cloud Architecture, Cloud Interoperability, Cloud Standards.

Abstract: Enabling cloud infrastructures to evolve into a transparent platform raises interoperability issues. Interoperability requires standard data models and communication technologies compatible with the existing Internet infrastructure. To reduce vendor lock-in situations, cloud computing must implement common strategies regarding standards, interoperability and portability. Open standards are of critical importance and need to be embedded into interoperability solutions. Interoperability is determined at the data level as well as the service level. Relevant modelling standards and integration solutions shall be analysed in the context of clouds.

1 INTRODUCTION

Enabling clouds to be merged and connected creates interoperability problems (Armbrust et al., 2009; Buyya et al., 2011). Interoperability requires standard data models and communication technologies compatible with existing Internet infrastructure. Public cloud services are available to the public. Hybrid clouds are an integrated cloud services arrangement that includes provision of compute resources from more than one source. Hybrid architectural models may be vertically partitioned (e.g. data stored privately) or horizontally partitioned (e.g. using public cloud to prototype a new device view of a service in parallel with an existing implementation). Interoperability solutions for flexible composition, migration and portability are sought (451 Group, 2010). To reduce vendor lock-in, cloud computing must implement universal strategies regarding standards, interoperability and portability. Open standards are of critical importance and need to be embedded into interoperability solutions (DMTF, 2010). Interoperability is determined at the data level as well as the service level (OMG, 2009; Cloud Standards, 2013). The objectives here include the review of relevant standards for cloud architecture interoperability and analysing the overall maturity of the technology and determining current trends and shortfalls.

Typically, cloud computing consists of the three cloud layers infrastructure (IaaS), platform (PaaS) and software (SaaS) as service (Armbrust et al., 2009; Buyya et al., 2011). In addition, we can differentiate between (hardware or software) resources provided in a traditional way, and the … *as a Service* version of them, which considers virtualization, multitenancy and elasticity as the main properties (Wang et al., 2009). Different usage models can be derived from the combinations of the layers. For instance, an application over a platform: a software provider can develop an application (e.g., a content management or delivery system) and publish it. That allows it to be deployed using some platform software (Web application container or database) and offered as a service (PaaS). The different usage models rely on interoperability solutions. Service-based abstractions (APIs) need to be aligned with common (Web) service description, modelling and composition standards.

Interoperability concerns arise in different situations: Firstly, interoperability between layers - standardised APIs are necessary to allow higher cloud layers to link to a range of services provided at the lower layers, e.g. platform implementations to uniformly link to IaaS offerings. Roles of importance are service provider and service user. Secondly, interoperability within layers - standards are required to allow components in a layer to interact and be exchangeable. Non-service interactions need to be supported, e.g. where a software developer combines different platforms in the development of a new system.

2 INTEROPERABILITY STANDARDS

Standards are necessary to consolidate efforts in a technology domain and to enable interoperability in

Pahl C., Zhang L. and Fowley F..
Interoperability Standards for Cloud Architecture.
DOI: 10.5220/0004366901230126
In *Proceedings of the 3rd International Conference on Cloud Computing and Services Science* (CLOSER-2013), pages 123-126
ISBN: 978-989-8565-52-5

any technology domain. An overview and a categorisation of standards relevant to interoperability in the cloud computing context that we cover here is:

- Service modelling: Open-SCA (service composition and interaction), USDL/SoaML/CloudML (multi-view services), EMML (mashups)
- Service interfaces: OCCI (infrastructure management), CIMI (infrastructure management), EC2 (de-facto standard), TOSCA (portability), CDMI (data management)
- Infrastructure: OVF (virtual machines)

For each standard, we provide background about origins, support and purpose, the intended usage, and an analysis of the relevance for interoperability considerations. Providing a comprehensive overview of all standards is not the objective. We have singled out those that represent specific aspects well. Well-known services standards and security standards shall not be covered due to space reasons.

The OASIS Open Composite Services Architecture (CSA) aims to simplify SOA application development (Open CSA, 2012) connected to the Service Component Architecture (SCA) and Service Data Objects (SDO) families of specifications. The SCA Assembly Model is a framework to describe service coordination and interaction that ties in service composition with common software architecture concerns. The CSA standards can be utilised as is or can serve as input for any composition and assembly language. The specifications on the SCA Assembly Model are very relevant for interoperability. Data interoperability is an equally important concern in Open-CSA. Data could be represented in Service Data Objects.

Enterprise Mashups combine and remix data from databases, spreadsheets, websites, Web Services, RSS/Atom feeds, and unstructured sources that deliver actionable information. The Open Mashup Alliance (OMA) is in charge of the Enterprise Mashup Markup Language (EMML) (EMML, 2012). It can support enterprise mashup implementations, improve mashup portability of mashup designs, and increase the interoperability of mashup solutions. EMML is designed to be complimentary to and integrated with languages like JavaScript, Java, Groovy, and Ruby via scripting. It is particularly suited for interoperability issues related to mashup creation. It supports lightweight and integrative service assembly and, therefore, represents specific modelling and integration concerns.

The aim of the Unified Service Description Language (USDL) is describing general parts of technical and business services to allow services to become tradable and consumable (USDL, 2012). Technical services are considered services based on WSDL, REST or other technical specifications. Business services are defined as business activities that are provided by a service provider to a service consumer to create value for the consumer. The USDL definition aims at complementing the technical language stack by adding required business and operational information. The targeted cloud stakeholders for USDL are service providers, infrastructure providers, service assemblers and service consumers. USDL defines an interoperability-centric language that enables its users to model arbitrary services and to integrate with existing standards. The aim to address service modelling and support mappings to different standards makes this interesting for interoperable cloud modelling. This enables new business models for service brokerage because services can automatically be offered, delivered, executed, and composed from services of different providers.

Another development to be considered in this context includes SoaML (standardised by OMG, see http://www.omg.org/spec/SoaML/), which falls into the same category as USDL in our categorisation as a service description and modelling language. While still under development (and thus far from being standardised), CloudML (http://www.cloudml.org/) is a language more specific to clouds.

OCCI (infrastructure lifecycle management) is now the first of four cloud-specific standards, also including CIMI (like OCCI on infrastructure management), TOSCA (portability and cloud-bursting), and CDMI (data management), interleaved with a note on EC2 as a proprietary solution that has become a de-facto standard. Providers offer IaaS solutions to enhance elastic capacity, where server instances are executed in their proprietary infrastructure and billed on a utility computing basis. For the infrastructure layer this means that typically virtual machines on a per-instance per-hour basis are the units. The OGF OCCI working group provides an API specification for the management of cloud computing infrastructure to capture these concerns (OCCI, 2012). The scope is a comprehensive range of high-level functionality required for the life-cycle management of virtual machines (or workloads) running on virtualization technologies (such as containers) supporting service elasticity. OCCI provides an API for interfacing IaaS cloud computing facilities, which is sufficiently complete to facilitate the implementation of interoperable implementations. While targeting IaaS concerns, it can foster interoperability endeavours at higher levels. The scope of OCCI is high-level functionality required for lifecycle management. This is in part realised through coverage of existing proprietary APIs. While the focus is on the upper cloud stack layers for the section presented in Figure 1, it is nonetheless a suitable framework for interoperability concerns at the interface of infrastructure services.

Similar to OCCI, the CIMI - Cloud Infrastructure Management Interface from DMTF addresses infrastructure management. CIMI which addresses the runtime maintenance and provisioning of cloud services. The scope of the CIMI standard covers core IaaS functionality, addressing deploying and managing virtual machines and other artifacts such as volumes, networks, or monitoring. Once interfaced to the IaaS provider, the information that needs to be processed to manage a cloud service can be discovered iteratively, including the metadata describing capabilities and resource constraints. Most developers use with the CIMI REST/HTTP-based protocol, the current interface binding to the model (others are expected later). This delivers standard HTTP status codes and supports JSON and XML serialization formats. CIMI, if widely used, would allow organisations to design cloud-based business solutions being assured that management (and governance) processes will not be compromised if the business solution is moved to another (standards-based) IaaS provider.

While OCCI and CIMI are similar standards, Amazon EC2 as a proprietary solution needs to be considered as well. Amazon Elastic Compute Cloud (Amazon EC2) provides resizeable computing capacity servers in Amazon's data centers to build and host software systems. They allow access to components and features using a web-based GUI, command line tools and APIs . EC2 comes with a rich ecosystem. Some open-source standards are pushed by Amazon competitors to regain market shares.

Supported by OASIS, the TOSCA framework aims to enhance the portability of cloud applications and services. TOSCA enables interoperable description of application and infrastructure cloud services, the relationships between parts of the service, and the operational behaviour of these services (such as deploy, patch, shutdown) independent of the supplier creating the service, and any particular cloud provider or hosting technology. TOSCA also aims to support higher-level operational behaviour to be associated with cloud infrastructure management. The core concept behind TOSCA is cloud bursting, which is the ability to move workloads between public and private cloud infrastructures in a transparent way. There seems to be some discussion about the prospect of success, with large IaaS providers not having joined the consortium yet. The core to the solution would be a hypervisor-agnostic portability mechanism, which requires IaaS compliance. TOSCA also needs to be observed as a vendor initiative in the context of open-source activities like OpenStack gaining momentum.

The CDMI, the Cloud Data Management Interface by SNAI targets cloud storage. CDMI is a standard for self-provisioning, administering and accessing cloud storage. CDMI defines RESTful HTTP operations for accessing the capabilities of the Cloud storage system. CDMI defines the functional interface that applications use to create, retrieve, update and delete data elements from the cloud. As part of this interface, a client can discover the capabilities of the cloud storage offering and use this interface to manage containers and the data that is placed in them. In addition, metadata can be set on containers and their contained data elements through this interface. Compared to OCCI and OVF, CDMI specifically targets data migration and format immigration. Although CDMI can also be used for task management, this would need extensive rules to be defined.

OVF is the most critical and successful standard specific to cloud infrastructure. The Open Virtualization Format (OVF) describes an open, secure, portable, efficient, and flexible format for the packaging and distribution of one or more virtual machines (OVF, 2012). OVF features include optimized distribution and portability of virtual appliances. It supports compression for more efficient package transfers, content verification and integrity checking, and provides a basic scheme for the management of software licensing. It supports vendor and platform independence as it does not rely on the use of a specific host platform, virtualization platform, or host/guest operating system. It is also designed to be extended as the industry moves forward with virtual appliance technology. The OVF format provides a complete specification of a virtual machine, which may include multiple virtual disks.OVF is a portable format that allows users to deploy VMs in any hypervisor that supports OVF. As such, it sits at the core of resource management in the infrastructure provisioning layer, overcoming previous deficiencies in standardised solutions such as VMDK. For supporting interoperability as a standard for this technical context, other features like localisation are important.

3 DISCUSSION AND CONCLUSIONS

Interoperability between clouds is vital for the further development of the cloud ecosystem and market. While standards for the Web Services context are abundant, more specific standards for the cloud computing domain reflect the current maturity.

- A number of standards for the lower infrastructure layers apply to respective cloud computing technologies. They address interoperability solutions for specific aspects like virtual machine management or data management.
- For platforms and services, the respective (Web) service standards are still of relevance. Standards exist, beyond the core Web services platform, that can further the development of platform and software services from existing offers. Generic service solution can provide a starting point where cloud-specific standards are lacking.

In addition, it is worth looking at a number of different concerns that help us to judge the state of standardisation and it's impact on interoperability: organisations behind standards and their domain, stakeholders involved through standards, and standards and open-source/proprietary solutions

Firstly, by looking at the organisations behind the standards, we can also observe that while the Web services domain is primarily dominated by W3C and OASIS in terms of standardisation, the situation in cloud computing is more diverse. Some of the organisations active include DMTF (management of distributed IT systems), the OGF (grid computing), the OMG (middleware), SNIA (storage), OASIS (services), OCC (cloud), as well as national (e.g. NIST) and sector-specific (e.g. ETSI - telecoms) organisations. Currently, there is a dominance of infrastructure and lower-level management. Secondly, stakeholders are yet another perspective that we can look at. We have referred to stakeholders in the review and discussion of standards where relevant to differentiate the different interoperability needs of stakeholders in clouds as multi-organisational, multi-role environments. While the infrastructure standards target clearly software developers, the more generic service-oriented standards are more at the interface (as-a-service) level, targeting service providers and consumers. Particularly combined roles, such as prosumers or aggregators that are providers and consumers benefit from the recent service description and modelling standards. Thirdly, while standards can achieve interoperability, often de-facto standards emerge from open-source or proprietary solutions. We dicussed OCCI and CIMI as standards in a context where OpenStack is a compliant open-source framework, all competing with Amazon EC2.

By looking at the standards we reviewed here for indications of future standardisation needs, emerging from the categorisation of standards are the following observations: (i) Modelling under incorporation of a variety of standards can support migration and, consequently, the uptake of cloud computing solutions. (ii) Composition, e.g. through mashups, is becoming of increasing importance to provide a market for basic and composite offering where providers and aggregators compete. (iii) Quality of Service and Service Level Agreement standardisations beyond security concerns in the cloud are actually largely lacking.

REFERENCES

451 Group (2010). *Cloud Computing 'As-a-service' market sizing*. Report II

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M. (2009). A view of cloud computing. *Comm. ACM* 53(4), 50-58.

Buyya, R., Broberg, J., and Goscinski, A.(2011). *Cloud Computing - Principles and Paradigms*. Wiley.

CloudCom (2011). *Market Implementation of Cloud Interoperability and Portability Research in IaaS and PaaS*. Retrieved from: http://www.cloud4soa.eu/workshop2011

Cloud Standards (2013). Retrieved from: http://cloud-standards.org/wiki/index.php?title=Main_Page

Distributed Management Task Force (2010). *Interoperable Clouds*. Retrieved from: http://www.dmtf.org/sites/default/files/standards/documents/ DSP-IS0101 _1. 0. 0.pdf

Enterprise Mashup Markup Language (2012). Retrieved from: http://www.openmashup.org/omadocs/v1.0/index.html

OCCI Open Cloud Computing Interface (2012). Retrieved from: http://occi-wg.org/

OMG Object Management Group (2009). *Cloud Interoperability Roadmaps Session*. Retrieved from: http:// www.omg.org/ news/ meetings/ tc/ ca/ special-events/ Cloud_Interop_Roadmaps.htm

Open CSA - Open Composite Services Architecture (2012). Retrieved from: http://www.oasis-opencsa.org/

OVF Open Virtualization Format (2012.) Retrieved from: http:// www.dmtf.org/ sites/ default/ files/ standards/ documents/DSP0243_1.1.0.pdf

Pahl, C. (2005) Layered ontological modelling for web service-oriented model-driven architecture. *European Conference on Model Driven Architecture Foundations and Applications ECMDA05*, 88-102.

Pahl, C., Giesecke, S. and Hasselbring, W. (2009). An ontology-based approach for modelling architectural styles. *European Conference on Software Architecture ECSA'2007*, 60-75.

Unified Service Description Language (2012.) Retrieved from: http://www.w3.org/2005/Incubator/usdl/

Wang, M.X., Bandara, K.Y. and Pahl, C. (2009). Integrated constraint violation handling for dynamic service composition. *IEEE International Conference on Services Computing SCC'09.*