Software Engineering Institute

The Role of Standards in Cloud-Computing Interoperability

Grace A. Lewis

October 2012

TECHNICAL NOTE
CMU/SEI-2012-TN-012

Research, Technology, and System Solutions Program

http://www.sei.cmu.edu

Carnegie Mellon

This material is based upon work funded and supported by the United States Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

This report was prepared for the

SEI Administrative Agent
AFLCMC/PZE
20 Schilling Circle, Bldg 1305, 3rd floor
Hanscom AFB, MA 01731-2125

Table of Contents

List of Figures

List of Tables

Abstract

In cloud computing, interoperability typically refers to the ability to easily move workloads and data from one cloud provider to another or between private and public clouds. A common tactic for enabling interoperability is the use of open standards, and many cloud standardization projects are developing standards for the cloud. This report explores the role of standards in cloud-computing interoperability. It covers cloud-computing basics and standard-related efforts, discusses several cloud-interoperability use cases, and provides some recommendations for moving forward with cloud-computing adoption regardless of the maturity of standards for the cloud.

1 Introduction

There is currently a lot of discussion about the role of standards in the cloud, along with a large amount of activity in standards development for the cloud. While some parties see the cloud as something completely new that requires an entirely new set of standards, other parties see the cloud as a technology based on existing technologies that already have standards. Answers to questions about how standards can enable interoperability depend on the type of service model that a cloud provider uses and the level of interoperability that an organization expects.

The cloud-computing community typically uses the term *interoperability* to refer to the ability to easily move workloads and data from one cloud provider to another or between private and public clouds. Even though this definition corresponds to the meaning of the term *portability*—the ability to move a system from one platform to another—the community refers to this property as *interoperability*, and I will use this term in this report.

In general, the cloud-computing community sees the lack of cloud interoperability as a barrier to cloud-computing adoption because organizations fear "vendor lock-in." Vendor lock-in refers to a situation in which, once an organization has selected a cloud provider, either it cannot move to another provider or it can change providers but only at great cost [Armbrust 2009, Hinchcliffe 2009, Linthicum 2009, Ahronovitz 2010, Harding 2010, Badger 2011, Kundra 2011]. Risks of vendor lock-in include reduced negotiation power in reaction to price increases and service discontinuation because the provider goes out of business.

A common tactic for enabling interoperability is the use of open standards [ITU 2005]. A representative of the military, for example, recently urged industry to take a more open-standards approach to cloud computing to increase adoption [Perera 2011]. The Open Cloud Manifesto published a set of principles that its members suggest that the industry follow, including using open standards and "playing nice with others" [Open Cloud 2009]. Cerf emphasizes the need for "inter-cloud standards" to improve asset management in the cloud [Krill 2010]. However, other groups state that using standards is just "one piece of the cloud interoperability puzzle" [Lewis 2008, Hemsoth 2010, Linthicum 2010b, Considine 2011]. Achieving interoperability may also require sound architecture principles and dynamic negotiation between cloud providers and users.

This report explores the role of standards in cloud-computing interoperability. The goal of the report is to provide greater insight into areas of cloud computing in which standards would be useful for interoperability and areas in which standards would not help or would need to mature to provide any value.

2 Cloud-Computing Basics

Cloud computing, as defined by the National Institute of Standards and Technology (NIST), is "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [Mell 2011, p. 2].

Clouds use one of three main types of computing models, and providers deploy them either publicly or privately. The type of service model and deployment model affect how much the cloud can benefit from standardization. This section describes the main types of service and deployment models for cloud computing. Additionally, this section also identifies some of the drivers of and barriers to cloud-computing adoption.

2.1 Cloud-Computing Service Models

Based on the services that the cloud provides, there are three types of cloud-computing models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

IaaS consists mainly of computational infrastructure available over the internet, such as compute cycles and storage. IaaS allows organizations and developers to extend their IT infrastructure on demand. Examples of IaaS offerings in alphabetical order include

- Amazon Elastic Compute Cloud (EC2): special virtual machines, called Amazon Machine Images (AMI), that can be deployed and run on the EC2 infrastructure [Amazon 2012a]
- Amazon Simple Storage Solution (S3): dynamically scalable storage resources [Amazon 2012c]
- Amazon's other data-related offerings: Elastic Block Storage, which provides block-level storage volumes for use with Amazon EC2 instances; SimpleDB, which is a non-relational data store; and Relational Data Store, which is a relational data store
- GoGrid Cloud Servers: dynamically scalable computation and storage resources [GoGrid 2012]
- Rackspace Cloud Servers: dynamically scalable computing, storage, and load-balancing resources [Rackspace 2012]

PaaS is based on application development platforms that allow the use of external resources to create and host applications. Examples of PaaS offerings in alphabetical order include

- CloudBees: platform to build, deploy, and manage Java applications [CloudBees 2012]
- Engine Yard: platform to build and deploy Ruby and PHP applications that can be extended with add-ons [Engine Yard 2012]
- Google App Engine: platform to develop and run Java, Python, and Go applications on Google's infrastructure [Google 2012a]

- Heroku: platform to deploy Java, Ruby, Python, Clojure, node.js, and Scala applications that can be extended with add-on resources [Heroku 2012]
- Microsoft Windows Azure: on-demand compute and storage services as well as a development and deployment platform for applications that run on Windows [Microsoft 2012a]
- Salesforce Force.com: platform to build and run applications and components bought from AppExchange or custom applications [Salesforce 2012a]

SaaS is a model of software deployment in which a third party provides an application to customers for use as a service on demand. Examples of SaaS offerings in alphabetical order include

- Google Apps: web-based email, calendar, document management, and web site creation and management [Google 2012b]
- Microsoft Office 365: email, calendar, Office Web Apps, web conferencing, and file sharing [Microsoft 2012b]
- NetSuite: business-management software applications that include accounting, enterprise resource planning (ERP), inventory management, customer relationship management (CRM), and e-Commerce [NetSuite 2012]
- Salesforce: CRM software application [Salesforce 2012b]
- SurveyTool: web-based survey platform for collecting feedback from employees, customers, focus groups, or any active user base [SurveyTool 2012]
- Zoho: large suite of web-based applications, mostly for enterprise use [Zoho 2012]

2.2 Cloud-Computing Deployment Models

Based on where organizations deploy cloud services and who can access these services, there are two main types of cloud-computing models: public cloud and private cloud.

In public clouds, organizations offer resources as a service, usually over an internet connection, typically for a pay-per-usage fee. Users can scale their use on demand and do not need to purchase hardware to use the service. Public cloud providers manage the infrastructure and pool resources into the capacity required by its users.

In private clouds, the user organization deploys resources inside a firewall and manages those resources itself. The user organization owns the software and hardware infrastructure, manages the cloud, and controls access to its resources. Typically, those resources and services are not shared outside the organization. CloudStack, Eucalyptus, HP, Microsoft, OpenStack, Ubuntu, and VMWare provide tools for building private clouds [CloudStack 2012, Eucalyptus 2012, HP 2012, Microsoft 2012c, Ubuntu 2012, VMWare 2012].

NIST defines two additional types of cloud deployment models: (1) community clouds that are shared by multiple organizations and support the specific needs and concerns of a community and (2) hybrid clouds that are the combination of two or more public, private, and community clouds [Mell 2011]. However, community and hybrid clouds are specialties of public and private clouds.

2.3 Drivers for Cloud-Computing Adoption

Several attributes of cloud computing motivate organizations to adopt cloud computing:

- Availability: Users have access to data and applications from around the globe.
- Collaboration: Organizations see the cloud as a way for members to work simultaneously on common data and information.
- Elasticity: Organizations can request, use, and release as many resources as needed based on changing needs.
- Lower infrastructure costs: The pay-per-use model allows an organization to pay only for the resources that it needs with no minimal investment in physical resources (i.e., to move from fixed costs to variable costs). The organization incurs no infrastructure-maintenance or upgrade costs for these resources.
- Reliability: Cloud providers have much more robust reliability mechanisms for supporting service-level agreements (SLAs) than those that a single organization could cost-effectively provide. However, it is important to note that organizations often view reliability as a barrier because cloud providers tend to rely on commodity hardware that is known to fail.
- Risk reduction: Organizations can use the cloud to test ideas and concepts before making major investments in technology.
- Scalability: Organizations have access to many resources that scale based on user demand.

2.4 Barriers to Cloud-Computing Adoption

Some key organizational concerns can act as barriers to the adoption of cloud computing:

- Interoperability: The cloud-computing community has not yet defined a universal set of standards or interfaces, resulting in a significant risk of vendor lock-in.
- Latency: All access to the cloud occurs through a network (or the internet in the case of public clouds), introducing latency into every communication between the user and the environment.
- Legal issues: Because cloud vendors tend to locate server farms and data centers where it is cheaper to operate them, some cloud-computing users have concerns about jurisdiction, data protection, fair information practices, and international data transfer.
- Platform or language constraints: Some cloud environments provide support for specific platforms and languages only.
- Security: The key concern is data privacy; in most cases, organizations do not have control of or know where cloud providers store their data.

3 Standard-Related Efforts for Cloud Computing

There are many cloud standardization projects—maybe too many [Fogarty 2011]. Some of these projects focus on standardizing parts of a cloud-computing solution such as workloads, authentication, and data access. Other efforts focus on standardizing how the parts should work together as a solution. The Cloud Standards Coordination Wiki maintains a list of some of these projects [Cloud Standards 2012]. Table 1 presents an alphabetical list of cloud standardization efforts. While this list is not complete, it provides an indication of the variety, number, and overlap of current projects related to standards for cloud-computing interoperability.

Table 1: Cloud Standardization Efforts

Project Name	URL	Focus
CloudAudit, also known as Automated Audit, Assertion, Assessment, and Assurance API (A6)	http://www.cloudaudit.org	• Open, extensible, and secure interface, namespace, and methodology for cloud-computing providers and their authorized consumers to automate the audit, assertion, assessment, and assurance of their environments • Part of the Cloud Security Alliance since October 2010
Cloud Computing Interoperability Forum	http://www.cloudforum.org	• Common, agreed-on framework/ontology for cloud platforms to exchange information in a unified manner • Sponsors of the Unified Cloud Interface Project to create an open and standardized cloud interface for the unification of various cloud APIs
Cloud Security Alliance	http://cloudsecurityalliance.org	• Recommended practices for cloud-computing security • Working on Version 3 of the *Security Guidance for Critical Areas of Focus in Cloud Computing* • Nonprofit organization that includes Google, Microsoft, Rackspace, Terremark, and others
Cloud Standards Customer Council	http://cloudstandardscustomercouncil.org	• Standards, security, and interoperability issues related to migration to the cloud • End-user advocacy group sponsored by the Object Management Group (OMG) and creator of the Open Cloud Manifesto
Cloud Storage Initiative	http://www.snia.org/cloud	• Adoption of cloud storage as a new delivery model (Data-Storage-as-a-Service) • Initiative sponsored by the Storage Networking Industry Association (SNIA), the creator and promoter of the Cloud Data Management Interface (CDMI) • SNIA includes members from NetApp, Oracle, and EMC

DeltaCloud	http://incubator.apache.org/deltacloud	• Abstraction layer for dealing with differences among IaaS providers • API based on representational state transfer (REST) with a small number of operations for managing instances • Currently has libraries for seven providers including Amazon EC2, Eucalyptus, and Rackspace
Distributed Management Task Force (DMTF)	http://dmtf.org/standards/cloud	• Management interoperability for cloud systems • Developer of the Open Virtualization Framework (OVF) • Runs the Open Cloud Standards Incubator
IEEE P2301, Guide for Cloud Portability and Interoperability Profiles	http://standards.ieee.org/develop/project/2301.html	• Standards-based options for application interfaces, portability interfaces, management interfaces, interoperability interfaces, file formats, and operation conventions
IEEE P2302, Draft Standard for Intercloud Interoperability and Federation	http://standards.ieee.org/develop/project/2302.html	• Protocols for exchanging data, programmatic queries, functions, and governance for clouds sharing data or functions or for federating one cloud to another
OASIS Identity in the Cloud (IDCloud)	http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=id-cloud	• Profiles of open standards for identity deployment, provisioning, and management in cloud computing • Performs risk and threat analyses on collected use cases and produces guidelines for mitigating vulnerabilities
Open Cloud Computing Interface	http://occi-wg.org	• REST-based interfaces for management of cloud resources including computing, storage, and bandwidth • Working group of the Open Grid Forum
Open Cloud Consortium	http://opencloudconsortium.org	• Frameworks for interoperating between clouds and operation of the Open Cloud Testbed
Open Data Center Alliance	http://www.opendatacenteralliance.org	• Unified customer vision for long-term data-center requirements • Developing usage models for cloud vendors • Independent IT consortium
OpenStack	http://www.openstack.org	• Open-source software for running private clouds • Currently consists of three core software projects: OpenStack Compute (Nova), OpenStack Object Storage (Swift), and OpenStack Image Service (Glance) • Founded by Rackspace and NASA
Standards Acceleration to Jumpstart Adoption of Cloud Computing	http://www.nist.gov/itl/cloud/sajacc.cfm	• Drives the creation of cloud-computing standards by providing key use cases that can be supported on cloud systems that implement a set of documented and public cloud-system specifications • Sponsored by NIST
The Open Group Cloud Work Group	https://collaboration.opengroup.org/cloudcomputing/	• Works with other cloud standards organizations to show enterprises how to best incorporate cloud computing into their organizations

TM Forum Cloud Services Initiative	http://www.tmforum.org/community/groups/cloud_computing_services/default.aspx	• Common approaches to increase cloud-computing adoption such as common terminology, transparent movement among cloud providers, security issues, and benchmarking

4 Cloud-Computing Interoperability Use Cases

Use cases in the context of cloud computing refer to typical ways in which cloud consumers and providers interact. NIST, OMG, DMTF, and others—as part of their efforts related to standards for data portability, cloud interoperability, security, and management—have developed use cases for cloud computing.

NIST defines 21 use cases classified into three groups: cloud management, cloud interoperability, and cloud security [Badger 2010]. These use cases are listed below [Badger 2010]:

- Cloud Management Use Cases
  - Open an Account
  - Close an Account
  - Terminate an Account
  - Copy Data Objects into a Cloud
  - Copy Data Objects out of a Cloud
  - Erase Data Objects on a Cloud
  - VM [virtual machine] Control: Allocate VM Instance
  - VM Control: Manage Virtual Machine Instance State
  - Query Cloud-Provider Capabilities and Capacities
- Cloud Interoperability Use Cases
  - Copy Data Objects Between Cloud-Providers
  - Dynamic Operation Dispatch to IaaS Clouds
  - Cloud Burst from Data Center to Cloud
  - Migrate a Queuing-Based Application
  - Migrate (fully-stopped) VMs from One Cloud Provider to Another
- Cloud Security Use Cases
  - Identity Management: User Account Provisioning
  - Identity Management: User Authentication in the Cloud
  - Identity Management: Data Access Authorization Policy Management in the Cloud
  - Identity Management: User Credential Synchronization Between Enterprises and the Cloud
  - eDiscovery
  - Security Monitoring
  - Sharing of Access to Data in a Cloud

OMG presents a more abstract set of use cases as part of the Open Cloud Manifesto [Ahronovitz 2010]. These are much more generic than those published by NIST and relate more to deployment than to usage. The use cases “Changing Cloud Vendors” and “Hybrid Cloud” are the ones of interest from a standards perspective because they are the main drivers for standards in cloud-computing environments. “Changing Cloud Vendors” particularly motivates organizations that do not want to be in a vendor lock-in situation. The full list is presented below [Ahronovitz 2010]:

- End User to Cloud: applications running in the public cloud and accessed by end users
- Enterprise to Cloud to End User: applications running in the public cloud and accessed by employees and customers
- Enterprise to Cloud: applications running in the public cloud integrated with internal IT capabilities
- Enterprise to Cloud to Enterprise: applications running in the public cloud and interoperating with partner applications (supply chain)
- Private Cloud: a cloud hosted by an organization inside that organization's firewall
- Changing Cloud Vendors: an organization using cloud services decides to switch cloud providers or work with additional providers
- Hybrid Cloud: multiple clouds work together, coordinated by a cloud broker that federates data, applications, user identity, security, and other details

DMTF produced a list of 14 use cases specifically related to cloud management [DMTF 2010]:

- Establish Relationship
- Administer Relationship
- Establish Service Contract
- Update Service Contract
- Contract Reporting
- Contract Billing
- Terminate Service Contract
- Provision Resources
- Deploy Service Template
- Change Resource Capacity
- Monitor Service Resources
- Create Service Template
- Create Service Offering
- Notification of Service Condition or Event

Across the complete set of use cases proposed by NIST, OMG, and DMTF, four types of use cases concern consumer–provider interactions that would benefit from the existence of standards. These interactions relate to interoperability and can be mapped to the following four basic cloud interoperability use cases:

1. User Authentication: A user who has established an identity with a cloud provider can use the same identity with another cloud provider.
2. Workload Migration: A workload that executes in one cloud provider can be uploaded to another cloud provider.
3. Data Migration: Data that resides in one cloud provider can be moved to another cloud provider.

4. Workload Management: Custom tools developed for cloud workload management can be used to manage multiple cloud resources from different vendors.

The remainder of this section describes existing standards and specifications that support these four main types of use cases.

4.1 User Authentication

The use case for *user authentication* corresponds to a user or program that needs to be identified in the cloud environment. It is important to differentiate between two types of users of cloud environments: end users and cloud-resource users.

End users are users of applications deployed on cloud resources. Because these users register and identify with the application and not with the infrastructure resources, they are usually not aware that the application is running on cloud resources.

Cloud-resource users are typically administrators of the cloud resources. These users can also set permissions for the resources based on roles, access lists, IP addresses, domains, and so forth. This second type of user is of greater interest from an interoperability perspective.

Some of the standardization efforts, as well as technologies that are becoming de facto standards, that support this use case are

- Amazon Web Services Identity Access Management (AWS IAM): Amazon uses this mechanism for user authentication and management, and it is becoming a de facto standard [Amazon 2012d]. It supports the creation and the permissions management for multiple users within an AWS account. Each user has unique security credentials with which to access the services associated with an account. Eucalyptus also uses AWS IAM for user authentication and management.
- OAuth: OAuth is an open protocol by the Internet Engineering Task Force (IETF) [OAuth 2010]. It provides a method for clients to access server resources on behalf of the resource owner. It also provides a process for end users to authorize third-party access to their server resources without sharing their credentials. The current version is 1.0, and IETF's work continues for Version 2.0. Similarly to WS-Security, OAuth Version 2.0 will support user identification information in Simple Object Access Protocol (SOAP) messages. Cloud platforms that support OAuth include Force.com, Google App Engine, and Microsoft Azure.
- OpenID: OpenID is an open standard that enables users to be authenticated in a decentralized manner [OpenID 2012]. Users create accounts with an OpenID identity provider and then use those accounts (or identities) to authenticate with any web resource that accepts OpenID authentication. Cloud platforms that support OpenID include Google App Engine and Microsoft Azure. OpenStack has an ongoing project to support OpenID.
- WS-Security: WS-Security is an OASIS security standard specification [OASIS 2006]. The current release is Version 1.1. WS-Security describes how to secure SOAP messages using Extensible Markup Language (XML) Signature and XML Encryption and attach security tokens to SOAP messages. Cloud platforms that support WS-Security for message authentication include Amazon EC2 and Microsoft Azure.

4.2 Workload Migration

The use case for *workload migration* corresponds to the migration of a workload, typically represented as a virtual-machine image, from one cloud provider to a different cloud provider. The migration of a workload requires (1) the extraction of the workload from one cloud environment and (2) the upload of the workload to another cloud environment. Some of the standards that support this use case are

- Amazon Machine Image (AMI): An AMI is a special type of virtual machine that can be deployed within Amazon EC2 and is also becoming a de facto standard [Amazon 2012b]. Eucalyptus and OpenStack support AMI as well.
- Open Virtualization Framework (OVF): OVF is a virtual-machine packaging standard developed and supported by DMTF [DMTF 2012]. Cloud platforms that support OVF include Amazon EC2, Eucalyptus, and OpenStack.
- Virtual Hard Disk (VHD): VHD is a virtual-machine file format supported by Microsoft [Microsoft 2006]. Cloud platforms that support VHD include Amazon EC2 and Microsoft Azure.

4.3 Data Migration and Management

The use case for *data migration and management* corresponds to the migration of data from one cloud provider to another. As with workload migration, it requires (1) the extraction of the data from one cloud environment and (2) the upload of the data to another cloud environment. In addition, in an interoperability context, once the data has been moved to the new provider, any program that performed create, retrieve, update, or delete (CRUD) operations on that data in the original cloud provider should continue to work in the new cloud provider.

There are two types of cloud storage. Typed-data storage works similarly to an SQL-compatible database and enables CRUD operations on user-defined tables. Object storage enables CRUD operations of generic objects that range from data items (similar to a row of a table), to files, to virtual-machine images.

Some of the standards that support this use case, especially for object storage, are

- Cloud Data Management Interface (CDMI): CDMI is a standard supported by the Storage Networking Industry Association (SNIA) [SNIA 2011]. CDMI defines an API to CRUD data elements from a cloud-storage environment. It also defines an API for discovery of cloud-storage capabilities and management of data containers.
- SOAP: Even though SOAP is not a data-specific standard, multiple cloud-storage providers support data- and storage-management interfaces that use SOAP as a protocol. SOAP is a W3C specification that defines a framework to construct XML-based messages in a decentralized, networked environment [W3C 2007]. The current version is 1.2, and HTTP is the primary transport mechanism. Amazon S3 provides a SOAP-based interface that other cloud-storage environments, including Eucalyptus and OpenStack, also support.
- Representational State Transfer (REST): REST is not a data-specific standard either, but multiple cloud-storage providers support RESTful interfaces. REST is considered an architecture and not a protocol [IBM 2008]. In a REST implementation, every entity that can be identified,

named, addressed, or handled is considered a resource. Each resource is addressable via its universal resource identifier and provides the same interface, as defined by HTTP: GET, POST, PUT, DELETE. Amazon S3 provides a RESTful interface that Eucalyptus and OpenStack also support. Other providers with RESTful interfaces for data management include Salesforce.com's Force.com, Microsoft Windows Azure (Windows Azure Storage), OpenStack (Object Storage), and Rackspace (Cloud Files). The API defined by CDMI is a RESTful interface.

4.4 Workload Management

The use case for *workload management* corresponds to the management of a workload deployed in the cloud environment, such as starting, stopping, changing, or querying the state of a virtual instance. As with the data-management use case, in an interoperability context an organization can ideally use any workload-management program with any provider. Even though most environments provide a form of management console or command-line tools, they also provide APIs based on REST or SOAP. Providers that offer SOAP-based or RESTful APIs for workload management include Amazon EC2, Eucalyptus, GoGrid Cloud Servers, Google App Engine, Microsoft Windows Azure, and OpenStack (Image Service).

5 Role of Standards in Cloud-Computing Environments

Cloud users would particularly welcome standards that address the workload migration and data migration use cases because such standards would mitigate vendor lock-in concerns. This requires standardization of virtual-machine image file formats and APIs for cloud storage [Ahronovitz 2010]. Standardization for the user-authentication use case has the advantage that user identities based on OpenID or authentication protocols based on OAuth, for example, could be used across multiple providers that support these standards. Similarly, standardization to support the workload-management use case would leverage any existing efforts related to the construction of workload-management clients and scripts that could be used across multiple providers.

However, cloud providers use different types of service models, and some service models stand to benefit more from standardization than others. The remainder of this section looks at how IaaS, PaaS, and SaaS would benefit from standardization.

5.1 Infrastructure as a Service (IaaS)

IaaS is the service model that would most benefit from standardization because the main building blocks of IaaS are workloads represented as virtual-machine images and storage units that vary from typed data to raw data [Badger 2011].

For workload migration, standards efforts such as OVF and VHD would allow users to extract an image from one provider and upload it to another provider. Given that most IaaS providers allow consumers to install and run any operating system, a more manual and time-consuming form of migration would be to retrieve the image from the current provider, create a new image on a new provider, and reinstall software [Badger 2011]. This manual migration would not require standards as long as there is a way to retrieve the application state (e.g., application data, files, running processes) from the source image and move it to a new image.

For data migration, standards efforts such as CDMI and the Amazon S3 API, which multiple providers support, would enable users to extract data from one provider and upload it to a different provider. If a provider implements these standard interfaces using SOAP- or REST-based protocols, the cloud will offer the advantages of ease of development and tool availability. However, these standards are more useful for raw data that is not typed (e.g., virtual-machine images, files, blobs) because the cloud resource in this case simply acts as a container and usually does not require data transformation. For typed data, data migration would occur similarly to any other data-migration task: users must extract data from its original source, transform it to a format compatible with the target source, and upload it into the target source, which could be a complex process [Fogarty 2011]. The effort required for transformation will also depend on factors such as the similarity between the target's and source's data-storage technologies (e.g., moving from one SQL-compatible database to another will be easier than moving from an object database to a relational database or vice versa) and the similarity of the interface operations (e.g., two SOAP-based interfaces can have completely different operations).

5.2 Platform as a Service (PaaS)

The PaaS service model benefits less from standardization than IaaS. Organizations that buy into PaaS do it for the perceived advantages of the development platform. The platform provides many capabilities out of the box, such as managed application environments, user authentication, data storage, reliable messaging, and other functionality in the form of libraries that can be integrated into applications. This functionality is tied to a specific language and runtime environment. For example, Google App Engine supports applications written in Java, Python, and Go. Microsoft Azure supports applications written in .NET, and more recently applications written in Java, PHP, and Node.js.

The incentives for PaaS adoption are primarily rapid development and deployment and the potential for these applications to serve a greater number of clients. Buying into a PaaS provider means buying into a platform in the same way that organizations traditionally have, which is based on added value, skills, cost, and any other criteria.

Providers can make applications more interoperable by selecting platforms that support more standardized tools and languages, such as those based on the Java language or standard data-access interfaces, including Java Database Connectivity (JDBC), Open Database Connectivity (ODBC), and SQL. However, even among providers that support the same programming language, the interfaces to basic services such as authentication, files, queues, hash tables, and tasks may not be compatible [Badger 2011]. In addition, native options may be more powerful (i.e., have greater benefit that can motivate an adoption decision) than standardized options. For example, the default data store in Google App Engine is the High Replication data store that offers automatic replication of data across data centers. A user can access the data store with a standard API or a low-level API. The tradeoff is that the standard API makes an application more portable but offers less control and less provider-specific value-added features than the low-level API, resulting in a lowest common denominator for features [Badger 2011].

5.3 Software as a Service (SaaS)

SaaS is a somewhat different model than IaaS and PaaS because it is a licensing agreement to third-party software instead of a different deployment model for existing resources that range from data storage to applications.

Benefits of standardization for SaaS are even more limited than for PaaS. For SaaS offerings such as Salesforce.com CRM, the user is an end user. However, there are other SaaS offerings such as Google Maps or Yahoo Social in which the user can be a developer who is integrating functionality from these services into other applications [Google 2012c, Yahoo! 2012]. In the latter case, standardized APIs are useful because they facilitate the development process [Linthicum 2010a]. However, unless the APIs are identical from a functional perspective, this standardization helps little with migration.

Migration for the case when the SaaS user is an end user would occur in the same way as with any software migration because each SaaS provider has its own processing logic; it is simply a different way to license software [Harding 2010]. In this case, the only area where SaaS would benefit from standardization is data storage because the most important concern for SaaS consumers, especially for enterprise software SaaS such as CRM or human resources, is how to extract their

data. In one widely publicized incident, an online storage service shut down and a SaaS provider lost access to 45% of its customer data [Armbrust 2009]. In this case, the consumer would have to extract its data from the SaaS provider, write logic to perform data transformations, and then upload data to a new SaaS provider. The standardized APIs could potentially make this task easier.

5.4 Do Standards Make Sense Beyond IaaS?

The answer to this question is that they probably do not. A decision to adopt IaaS extends an organization's IT department mainly by adding resources (primarily computation and storage) that exist outside of the organization and for which there is a pay-per-use fee as opposed to acquisition, maintenance, and obsolescence costs. In this case, the advantage of standards is that an organization can move these basic resources if another provider offers better prices or the organization experiences problems with its provider. Also, there is not much differentiation among IaaS providers other than price and SLAs.

A decision to adopt a PaaS or SaaS provider goes beyond extending basic IT resources. The service model usually involves value-added features in the form of libraries and platforms in the case of PaaS and application software in the case of SaaS. An organization selects a PaaS or SaaS provider based on these value-added features, and the choice involves a commitment similar to the commitment to traditional development platforms, deployment platforms, and software packages. PaaS and SaaS providers' focus on offering precisely the best set of value-added features creates many differences among them. Expecting PaaS and SaaS providers to standardize feature sets is equivalent to asking ERP software vendors to standardize feature sets. This is not likely to happen because it is not in their best interest.

5.5 Can Existing Standards Support Cloud Interoperability Instead of Portability, or Do Clouds Require New Standards?

Interoperability refers to the ability of a collection of communicating entities to share specific information and operate on it according to agreed-on operational semantics [Brownsword 2009]. As mentioned earlier, even though the community desires standards for cloud interoperability, the reality is that existing standards efforts are so far focusing mainly on portability, which is the ability to migrate workloads and data from one provider to another.

Cloud interoperability, based on Brownsword's definition, refers to the ability of resources on one cloud provider to communicate with resources on another cloud provider. With this definition in mind, I examine whether each of the three types of service models would benefit from existing standards that promote interoperability, such as those that support service-oriented systems, or whether they would require new standards specific to the type of service model a cloud provider uses.

There are two basic use cases (UCs) for IaaS that exercise this service model's potential for interoperability:

UC1: Workload W_1 on Cloud C_1 can communicate with Workload W_2 on Cloud C_2.

UC2: Workload W_1 on Cloud C_1 can access Data Store *DS* in Cloud C_2.

To support UC1, the following conditions must be true:

1. Workload W_2 is accessible over the network and has a known address, uniform resource identifier (URI), or other unique identifier.
2. Workload W_1 is authorized to communicate with Workload W_2.
3. Workload W_2 exposes an interface that Workload W_1 can use.

This is a common interoperability scenario between two systems that does not require standards built specifically for the cloud. Standards such as SOAP and REST as well as existing user-authentication standards could support this scenario if the cloud meets the conditions listed above. Once workloads are running in a cloud instance, they behave like any other server.

Similarly to supporting *UC1*, to support *UC2* the following conditions must be true:

1. *DS* is accessible over the network and has a known address, URI, or other unique identifier.
2. Workload W_1 is authorized to access *DS*.
3. *DS* exposes an interface that Workload W_1 can use.

This use case does benefit from standards for cloud data access such as CDMI and the Amazon S3 API.

The basic use case that exercises the PaaS service model's potential for interoperability is similar to *UC1* for IaaS: Application A_1 deployed on Cloud C_1 can communicate with Application A_2 on Cloud C_2. Also similarly to supporting *UC1*, to support this use case the following must be true:

1. Application A_2 is accessible over the network and has a known address, URI, or other unique identifier.
2. Application A_1 is authorized to interact with Application A_2.
3. Application A_2 exposes an interface that Application A_1 can use.

This is also a common interoperability scenario that does not require standards built specifically for the cloud.

The basic use case that exercises the SaaS service model's potential for interoperability is the same as for PaaS, except that it refers to interoperability between SaaS products instead of between applications. Interoperability between PaaS-deployed applications and IaaS workloads/data stores and SaaS products could also be supported the same way, if the cloud meets the conditions listed above.

The bottom line is that existing standards such as those that support service-oriented systems can support real cloud interoperability. However, there are different levels of system interoperability, as shown in Figure 1. Technical interoperability is about exchanging data, semantic interoperability is about exchanging meaningful data, and organizational interoperability is about participating in multi-organizational business processes.

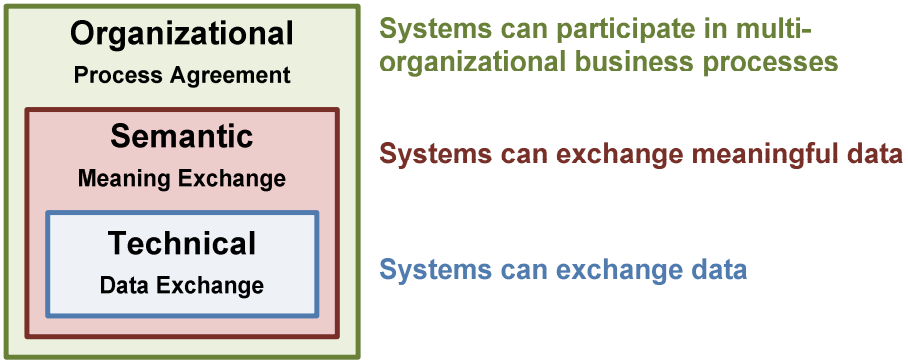

Figure 1: Interoperability Levels

Standards such as SOAP and REST enable technical (or syntactic) interoperability but do not guarantee semantic or organizational interoperability. Systems or data deployed inside cloud providers will have to rely on documentation or formal/informal agreements to provide meaning to the interaction, just as in any use case that required systems to interoperate.

6 Thoughts and Recommendations

6.1 Contingency Plans

A simple definition of a contingency plan is an "organized and coordinated set of steps to be taken if an emergency or disaster strikes" [BusinessDictionary 2012]. A contingency plan is a basic element of any IT strategy. It should be no different for cloud resources, especially because an organization does not have full control of these resources if they reside with an external provider. The contingency plan should determine what the organization will do if the cloud resources fail or become unavailable. This could happen temporarily because of a technical failure, but it could also happen permanently if the provider goes out of business, terminates the contract, or fails to meet SLA parameters and the organization must terminate the contract. Whatever the case, the organization must have an exit strategy that includes how to get back its assets (computation and data) and where to move those assets so that the business can keep operating [Harding 2010]. An organization's SLA with the provider should establish this exit strategy, especially to clearly indicate how the provider will return the assets [Badger 2011].

The organization must also test its contingency plan, and the complexity and sophistication of the plan will depend on the risks associated with losing the assets. An organization should answer questions such as Can the organization continue to operate without the assets? How fast will the organization need to migrate the assets? And can the organization temporarily migrate the assets to a local deployment while it negotiates a new contract?

Testing and implementing the contingency plan may require opening accounts with different cloud providers, creating migration scripts, uploading assets to the new providers, testing, and then either closing the account, keeping the account empty but in "standby mode," or setting the account up such that it is just a matter of starting the instance. The latter case will have an associated cost even if the resources are not active.

6.2 Sound Architecture Principles

System developers should leverage standards to support the architecture of a system, but standards should not drive the architecture of a system [Linthicum 2010b]. The rationale for this principle is that a system architecture should be fairly stable and withstand changes in standards and technology over time. As stated in Section 3, there are many ongoing standardization efforts. Given that typical standards definition, review, and approval processes can take up to three years, organizations will have to move forward with or without them [Hemsoth 2010].

In addition, the challenge for data migration between providers is not standardizing at the API level but ensuring that the system will maintain quality attributes after the migration [Ricknäs 2010]. Quality attributes are nonfunctional properties of software systems by which stakeholders judge their quality [Bass 2003]. Performance, security, scalability, and ease of monitoring of the system are examples of quality attributes that could vary among providers [Badger 2011, Fogarty 2011]. This means that developers should design systems that make use of cloud resources to account for system quality attributes that are important but over which they have no control. For example, developers may need to consider how quality attributes inform the system's architecture

requirements [Linthicum 2010b]. For example, if data privacy is important, then an architecture that enables data encryption before storing it in the cloud is important. If security is important, then a strategy that involves trusted, third-party authentication might be necessary. If portability is important, then system developers should implement abstraction layers that hide differences among providers (e.g., data access, resource management).

6.3 First-, Second-, and Third-Generation Cloud-Based Systems

In 2005, a group of researchers from the European Union defined three generations of service-oriented systems [Papazoglou 2008]. In the first generation of service-oriented systems, services are discovered at design time and integrated at compile time. In the second generation of service-oriented systems, services are composed into business processes that can be adapted and reconfigured at installation and to some extent at runtime. In the third generation, services are integrated at runtime and are context sensitive and reconfigurable in an autonomic, ad hoc manner.

After six years, and after many years of research, we have not reached the point where third-generation service-oriented systems are of production quality [Lewis 2009, 2010]. This is because dynamic service discovery and composition require agreements regarding data models and ontologies, SLA representation and negotiation, representation of quality attributes, and other aspects that go beyond simply agreeing on an interface that can execute the process at runtime with minimum (ideally no) human intervention.

The development of cloud-based systems over time is analogous to Papazoglou and colleagues' classification of the way that service-oriented systems have evolved. In the first generation of cloud-based systems, the location and negotiation of cloud resources occur at design time. Cloud resources are provisioned and instantiated following the negotiation process. In the second generation of cloud-based systems, the location and negotiation of cloud resources occur at design time. However, cloud resources are provisioned either at design time or runtime and instantiated at runtime, depending on business needs. This would support, for example, a cloud-bursting strategy in which developers design a system for an average load but the system can balance its load to a cloud provider when it reaches its full capacity. In the third generation of cloud-based systems, the location, negotiation, provisioning, and instantiation of cloud resources occur at runtime.

Today, we are in the first generation and on the verge of entering the second generation of cloud-based systems. The third generation will require much more dynamic and automated negotiation and provisioning of cloud environments than today's practice of a more manually negotiated process between consumer and provider [Considine 2011]. Reaching the third generation of cloud-based systems will require cloud consumer, cloud provider, and software-vendor groups to work together to define standardized, self-descriptive, machine-readable representations of

- basic resource characteristics such as size, platform, and API
- advanced resource characteristics such as pricing and quality attribute values
- negotiation protocols and processes
- billing protocols and processes

For now, standardization should focus on the basic use cases of user authentication, workload migration, data migration, and workload management that will serve as a starting point for the more

dynamic use cases in which location, negotiation, and provisioning of cloud resources occur at runtime.

7 Conclusion

Cloud computing is an economic model. It is a different way to acquire and manage IT resources. Organizations adopt cloud computing as a way to solve business problems, not technical problems. A decision to move resources to the cloud requires risk analysis and cost-benefit analysis as with any IT investment.

A valid concern for organizations interested in cloud computing is vendor lock-in. How do I move assets if a cloud provider disappears or if I find a better option? A potential solution to this problem is the creation of cloud interoperability standards to support basic use cases of user authentication, workload migration, data migration, and workload management that would ease the migration of workloads and data from one provider to another. However, these standards apply mostly to IaaS environments, where assets are indeed data and workloads. They do not apply to PaaS and SaaS environments, where assets are platforms and applications tightly coupled to an infrastructure and value-added features.

One of the problems with cloud standards is that there are too many standardization efforts. This is similar to what happened around 2006 with web service standards. At some point, there were approximately 250 standards, specifications, and recommendations to support different quality attributes. Now there are approximately 100 standards efforts related to web services. Over time, some standards have become de facto or are widely supported, such as WSDL, SOAP, BPEL, WS-Security, and WS-Addressing. Some have simply died because of lack of support, such as WS-Privacy and WS-Authorization. Others have fallen in and out of favor. For example, REST has become in many cases a preferred architecture for web-service implementation over SOAP-based implementations because it is easier to use [DuVander 2010].

Cloud computing is currently going through what web services went through in 2006. As with web service standards, it will take some time for a robust and widely supported set of standards to emerge. In the meantime, cloud-based systems should be implemented in a manner that separates standards-reliant components from the rest of the system in order to minimize the impact of standards evolution.

The bottom line is that any migration—whether between cloud providers or just between local servers, databases, or applications—has a cost. The cost will depend on how different the source and the target environments are and, in the case of cloud environments, how different the representations of workloads and data are between the two environments.

Cloud standardization efforts should focus on finding common representations of user identity, workload (virtual-machine images), cloud-storage APIs, and cloud management APIs. We cannot assume that each will have a single standard because vendors influence many standards committees. However, an agreement on a small number of each can also enable the creation of transformers, importers, exporters, or abstract APIs that can reduce migration efforts. These standards will potentially enable the dynamic third generation of cloud-based systems, but only business needs will motivate and determine this evolution.

References

URLs are valid as of the publication date of this document.

[Ahronovitz 2010]
Ahronovitz, Miha, et al. for the Cloud Computing Use Cases Discussion Group. *Cloud Computing Use Cases White Paper* (Version 4.0). http://opencloudmanifesto.org/Cloud_Computing_Use_Cases_Whitepaper-4_0.pdf (2010).

[Amazon 2012a]
Amazon. *Amazon Elastic Compute Cloud (Amazon EC2)*. http://aws.amazon.com/ec2 (2012).

[Amazon 2012b]
Amazon. *Amazon Machine Images (AMIs)*. http://aws.amazon.com/amis (2012).

[Amazon 2012c]
Amazon. *Amazon Simple Storage Service (Amazon S3)*. http://aws.amazon.com/s3 (2012).

[Amazon 2012d]
Amazon. *AWS Identity and Access Management (IAM)*. http://aws.amazon.com/iam (2012).

[Armbrust 2009]
Armbrust, Michael, et al. *Above the Clouds: A Berkeley View of Cloud Computing* (UCB/EECS-2009-28). University of California, 2009. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf

[Badger 2010]
Badger, Lee, et al. *Cloud Computing Use Cases*. National Institute of Standards and Technology, 2010. http://www.nist.gov/itl/cloud/use-cases.cfm

[Badger 2011]
Badger, Lee, Grance, Tim, Patt-Corner, Robert, & Voas, Jeff. *Cloud Computing Synopsis and Recommendations* [draft] (Special Publication 800-146). National Institute of Standards and Technology, 2011. http://csrc.nist.gov/publications/drafts/800-146/Draft-NIST-SP800-146.pdf

[Bass 2003]
Bass, L., Clements, P., & Kazman, R. *Software Architecture in Practice*, 2nd ed. Addison-Wesley SEI Series in Software Engineering, 2003.

[Brownsword 2009]
Brownsword, Lisa, et al. *Current Perspectives on Interoperability*. Software Engineering Institute, Carnegie Mellon University, 2004. http://www.sei.cmu.edu/library/abstracts/reports/04tr009.cfm

[BusinessDictionary 2012]
BusinessDictionary.com. “Contingency Plan.” http://www.businessdictionary.com/definition/contingency-plan.html (2012).

[CloudBees 2012]
Cloud Bees. *Cloud Platform as a Service for Java Web Apps.* http://www.cloudbees.com (2012).

[CloudStack 2012]
CloudStack. *CloudStack | Open Source Cloud Computing.* http://cloudstack.org (2012).

[Cloud Standards 2012]
Cloud Standards Organization. *Cloud Standards Coordination Wiki*. http://cloud-standards.org (2012).

[Considine 2011]
Considine, John. "Cloud Computing Standards—Not This Year." *CloudSwitch*. http://www.cloudswitch.com/page/cloud-computing-standards-not-this-year (2011).

[DMTF 2010]
Distributed Management Task Force. *Use Cases and Interactions for Managing Clouds* (DSP-IS0103). http://dmtf.org/sites/default/files/standards/documents/DSP-IS0103_1.0.0.pdf (2010).

[DMTF 2012]
Distributed Management Task Force. *Open Virtualization Format*. http://www.dmtf.org/standards/ovf (2012).

[DuVander 2010]
DuVander, Adam. "New Job Requirement: Experience Building RESTful APIs." *Programmable Web*. http://blog.programmableweb.com/2010/06/09/new-job-requirement-experience-building-restful-apis (2010).

[Engine Yard 2012]
Engine Yard. *Ruby on Rails and PHP Cloud Hosting PaaS*. http://www.engineyard.com (2012).

[Eucalyptus 2012]
Eucalyptus. http://open.eucalyptus.com (2012).

[Fogarty 2011]
Fogarty, Kevin. "Cloud Computing Standards: Too Many, Doing Too Little." *CIO Magazine*. http://www.cio.com/article/679067/Cloud_Computing_Standards_Too_Many_Doing_Too_Little (2011).

[GoGrid 2012]
GoGrid. http://www.gogrid.com (2012).

[Google 2012a]
Google. *Google App Engine*. http://code.google.com/appengine (2012).

[Google 2012b]
Google. *Google Apps*. http://www.google.com/apps (2012).

[Google 2012c]
Google. *Google Maps API Family*. http://code.google.com/apis/maps/index.html (2012).

[Harding 2010]
Harding, Chris. "Cloud Computing Needs Standards." *Baseline Magazine*. http://www.baselinemag.com/c/a/Utility-Computing/Cloud-Computing-Needs-Standards-778678 (2010).

[Hemsoth 2010]
Hemsoth, Nicole. "Inching Closer to a New Era for Interoperability Standards." *HPC in the Cloud*. http://www.hpcinthecloud.com/hpccloud/2010-09-07/inching_closer_to_a_new_era_for_interoperability_standards.html (2010).

[Heroku 2012]
Heroku. *Heroku | Cloud Application Platform.* http://www.heroku.com (2012).

[Hinchcliffe 2009]
Hinchcliffe, Don. "As Cloud Computing Grows, Where Are the Standards?" *ebizQ*. http://www.ebizq.net/blogs/enterprise/2009/10/as_cloud_computing_grows_where.php?mkt_tok=3RkMMJWWfF9wsRolsq3fLqzsmxzEJ8v56OosT%2Frn28M3109ad%2BrmPBy524s%3D (2009).

[HP 2012]
Hewlett-Packard. *CloudSystem Solutions*. http://www8.hp.com/us/en/business-solutions/solution.html?compURI=1079455 (2012).

[IBM 2008]
IBM Corporation. *RESTful Web Services.* https://www.ibm.com/developerworks/webservices/library/ws-restful (2008).

[ITU 2005]
International Telecommunications Union. *Definition of "Open Standards."* http://www.itu.int/en/ITU-T/ipr/Pages/open.aspx (2005).

[Krill 2010]
Krill, Paul. "Cerf Urges Standards for Cloud Computing. *InfoWorld*. http://www.infoworld.com/d/cloud-computing/cerf-urges-standards-cloud-computing-817?source=IFWNLE_nlt_cloud_2010-01-11 (2010).

[Kundra 2011]
Kundra, Vivek. *Federal Cloud Computing Strategy*. The White House, 2011. http://www.cio.gov/documents/federal-cloud-computing-strategy.pdf

[Lewis 2008]
Lewis, G. A., Morris, E., Simanta, S., & Wrage, L. "Why Standards Are Not Enough to Guarantee End-to-End Interoperability," 164–173. *Proceedings of the Seventh International Conference on Composition-Based Software Systems.* Madrid, Spain, Feb. 2008. IEEE Computer Society

Press, 2008.
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4464021&isnumber=4463987

[Lewis 2009]
Lewis, Grace. "Is SOA Being Pushed Beyond Its Limits?" *Microsoft Architecture Journal 21* (2009): Article 9. http://msdn.microsoft.com/en-us/architecture/aa699422

[Lewis 2010]
Lewis, Grace A., Smith, Dennis B., & Kontogiannis, Kostas. *A Research Agenda for Service-Oriented Architecture (SOA): Maintenance and Evolution of Service-Oriented Systems* (CMU/SEI-2010-TN-003). Software Engineering Institute, Carnegie Mellon University, 2010. http://www.sei.cmu.edu/library/abstracts/reports/10tn003.cfm

[Linthicum 2009]
Linthicum, David. "Top 5 Cloud Computing Predictions for 2010." *InfoWorld*. http://www.infoworld.com/d/cloud-computing/top-5-cloud-computing-predictions-2010-188?source=IFWNLE_nlt_daily_2009-12-08 (2009).

[Linthicum 2010a]
Linthicum, David. "The API Is Everything for Cloud Computing: While Many Cloud Providers Consider APIs as an Afterthought, They Should Be Front and Center." *InfoWorld*. http://www.infoworld.com/d/cloud-computing/the-api-everything-cloud-computing-481?source=IFWNLE_nlt_cloud_2010-06-07 (2010).

[Linthicum 2010b]
Linthicum, David. "Cloud Computing Standards Are a Double Edged Sword." *EbizQ*. http://www.ebizq.net/blogs/cloudsoa/2010/06/cloud-computing-standards-are-a-double-edged-sword.php?mkt_tok=3RkMMJWWfF9wsRonuq7MZKXonjHpfsX76u8rWLHr08Yy0EZ5VunJEUWy2YIFSQ%3D%3D (2010).

[Mell 2011]
Mell, Peter & Grance, Timothy. *The NIST Definition of Cloud Computing* (Special Publication 800-145). National Institute of Standards and Technology, 2011. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[Microsoft 2006]
Microsoft Corporation. *Virtual Hard Disk*. http://technet.microsoft.com/en-us/library/bb676673.aspx (2006).

[Microsoft 2012a]
Microsoft Corporation. *Windows Azure*. http://www.microsoft.com/windowsazure (2012).

[Microsoft 2012b]
Microsoft Corporation. *Office Online Services – Hosted in the Cloud – Microsoft Office 365*. http://www.microsoft.com/en-us/office365 (2012).

[Microsoft 2012c]
Microsoft Corporation. *Microsoft Private Cloud.* http://www.microsoft.com/en-us/server-cloud/private-cloud (2012).

[NetSuite 2012]
NetSuite. http://www.netsuite.com/portal/home.shtml (2012).

[Novakouski 2011]
Novakouski, Marc & Lewis, G. *Interoperability in the e-Government Context* (CMU/SEI-2011-TN-014). Software Engineering Institute, Carnegie Mellon University, 2011. http://www.sei.cmu.edu/library/abstracts/reports/11tn014.cfm

[OASIS 2006]
OASIS. *OASIS Web Services Security (WSS)*. http://www.oasis-open.org/committees/wss (2006).

[OAuth 2010]
OAuth. http://oauth.net (2010).

[Open Cloud 2009]
Open Cloud Manifesto Group. *Open Cloud Manifesto*. http://www.opencloudmanifesto.org/Open%20Cloud%20Manifesto.pdf (2009).

[OpenID 2012]
OpenID. http://openid.net (2012).

[OpenStack 2011]
OpenStack. http://www.openstack.org (2011).

[Papazoglou 2008]
Papazoglou, Mike P., Traverso, Paolo, Dustdar, Schahram, & Leymann, Frank. “Service-Oriented Computing: A Research Roadmap.” *International Journal of Cooperative Informative Systems 17*, 2 (June 2008): 223–255.

[Perera 2011]
Perera, David. “Military Won't Commit to Single Cloud Computing Architecture, Say Panelists.” *Fierce Government IT: The Government IT New Briefing*. http://www.fiercegovernmentit.com/story/military-wont-commit-single-cloud-computing-architecture-say-panelists/2011-05-17?utm_medium=nl&utm_source=internal#ixzz1RQqkS8Na (2011).

[Rackspace 2012]
Rackspace. *The Rackspace Cloud*. http://www.rackspace.com/cloud (2012).

[Ricknäs 2010]
Ricknäs, Mikael. “Red Hat's CEO: Clouds Can Become the Mother of All Lock-ins.” *InfoWorld*. http://www.infoworld.com/d/cloud-computing/red-hats-ceo-clouds-can-become-the-mother-all-lock-ins-812?source=IFWNLE_nlt_cloud_2010-06-07 (2010).

[Salesforce 2012a]
Salesforce. *Force.com*. http://www.force.com (2012).

[Salesforce 2012b]
Salesforce. *Salesforce.com*. http://www.salesforce.com (2012).

[SNIA 2011]
SNIA. *Cloud Data Management Interface (CDMI)*. http://www.snia.org/cdmi (2011).

[SurveyTool 2012]
SurveyTool.com. http://www.surveytool.com (2012).

[Ubuntu 2012]
Ubuntu.com. *Cloud | Ubuntu*. http://www.ubuntu.com/cloud (2012).

[VMWare 2012]
VMWare. *Private Cloud Computing*. http://www.vmware.com/cloud-computing/private-cloud (2012).

[W3C 2007]
W3C. *SOAP*. http://www.w3.org/TR/soap12-part1 (2007).

[Yahoo! 2012]
Yahoo! *Yahoo! Social API Guide*. http://developer.yahoo.com/social/rest_api_guide (2012).

[Zoho 2012]
Zoho. http://www.zoho.com (2012).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	October 2012	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
The Role of Standards in Cloud-Computing Interoperability	FA8721-05-C-0003

6. AUTHOR(S)

Grace A. Lewis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2012-TN-012

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFLCMC/PZE 20 Schilling Circle, Bldg 1305, 3rd floor Hanscom AFB, MA 01731-2125	n/a

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

In cloud computing, interoperability typically refers to the ability to easily move workloads and data from one cloud provider to another or between private and public clouds. A common tactic for enabling interoperability is the use of open standards, and many cloud standardization projects are developing standards for the cloud. This report explores the role of standards in cloud-computing interoperability. It covers cloud-computing basics and standard-related efforts, discusses several cloud-interoperability use cases, and provides some recommendations for moving forward with cloud-computing adoption regardless of the maturity of standards for the cloud.

14. SUBJECT TERMS	15. NUMBER OF PAGES
cloud computing, interoperability, standards, portability	38

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL