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, so there is currently a large amount of active work in standards development for the Cloud. This paper explores the role of standards in cloud-computing interoperability. It covers 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 active work 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.

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” [9].

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. PaaS is based on application development platforms that enable the use of external resources to create and host applications. SaaS is a model of software deployment in which a third-party provides an application to customers for use as a service on demand.

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. NIST defines two additional types of cloud deployment models: community cloud and hybrid cloud [9]. However, both community and hybrid cloud are classes of public and private clouds.

Interoperability refers to the ability of a collection of systems to exchange information and operate on it according to agreed-on semantics [12]. However, 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, so this is the term that is used in this paper.

In the e-Government context, cloud computing presents an opportunity for governments to develop and deploy nation-wide citizen services independent of local capabilities and infrastructure, host large amounts of public domain data, reduce integration costs, and reduce dependence on large vendors [21][23]. Therefore, it is not surprising that interoperability is a key topic of discussion for government policy makers. However, many governments see standards as the “cure-all” solution for the interoperability problem, often ignoring that interoperability is more than a simple technical problem that a standard might address [12][22].

This paper explores the role of standards in cloud-computing interoperability. Section 2 presents a list of standards-related efforts for cloud computing. Section 3 defines four basic use cases for cloud computing interoperability and discusses the standards that support these use cases. Section 4 discusses the role of standards in cloud computing in the context of the three main cloud computing models: IaaS, PaaS and SaaS. Section 5 provides
some thoughts and recommendations stemming from the study. Section 6 presents related work and Section 7 concludes the paper.

2. Standards-related efforts for cloud computing

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.” 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 [1][2][3][4]. 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. There are many cloud standardization projects — maybe too many. 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 at http://cloud-standards.org. Table 1 presents a 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.

3. Cloud-computing interoperability use cases

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. Use cases in this context refer to typical ways in which cloud consumers and providers interact.

NIST defines 21 use cases classified into three groups [10]: cloud management, cloud interoperability and cloud security. In particular, cloud interoperability use cases include copying data objects between cloud providers, loading workloads, cloud bursting, and migrating applications and virtual machines (VMs).

OMG presents a more abstract set of use cases as part of the Open Cloud Manifesto [2]. 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. The first one especially motivates organizations that do not want to be in a vendor lock-in situation. The additional use cases are End User to Cloud, Enterprise to Cloud to End User, Enterprise to Cloud, Enterprise to Cloud to Enterprise, and Private Cloud.

DMTF produced a list of 14 use cases specifically related to cloud management [11] that includes establishing and managing relationships; establishing, updating and terminating service contracts, contract reporting and billing; provisioning and monitoring resources, creating and deploying service templates; changing resource capacity; creating service offerings; and notifying service conditions or events.

If we look at the complete set of use cases proposed by NIST, OMG and DMTF, the use cases related to interoperability (or would benefit from the existence of standards) can be mapped to the following four basic cloud interoperability use cases:

• User Authentication: A user who has established an identity with a cloud provider can use the same identity with another cloud provider.

• Workload Migration: A workload that executes in one cloud provider can be uploaded to another cloud provider.

• Data Migration: Data that resides in one cloud provider can be moved to another cloud provider.

• 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.

3.1. User authentication

This use case 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: These 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: These 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.
Table 1. Cloud Computing Standardization Efforts

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudAudit, also known as Automated Audit, Assertion, Assessment, and Assurance API (A6)</td>
<td>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. As of October 2010, CloudAudit is part of the Cloud Security Alliance.</td>
</tr>
<tr>
<td>Cloud Computing Interoperability Forum</td>
<td>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.</td>
</tr>
<tr>
<td>Cloud Standards Customer Council</td>
<td>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.</td>
</tr>
<tr>
<td>Cloud Storage Initiative</td>
<td>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.</td>
</tr>
<tr>
<td>DeltaCloud</td>
<td>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.</td>
</tr>
<tr>
<td>IEEE P2302, Draft Standard for Intercloud Interoperability and Federation</td>
<td>Protocols for exchanging data, programmatic queries, functions, and governance for clouds sharing data or functions or for federating one cloud to another.</td>
</tr>
<tr>
<td>OASIS Identity in the Cloud (IDCloud)</td>
<td>Profiles of open standards for identity deployment, provisioning, and management in cloud computing. It performs risk and threat analyses on collected use cases and produces guidelines for mitigating vulnerabilities.</td>
</tr>
<tr>
<td>Open Cloud Computing Interface</td>
<td>REST-based interfaces for management of cloud resources including computing, storage, and bandwidth. Working group of the Open Grid Forum.</td>
</tr>
<tr>
<td>Open Cloud Consortium</td>
<td>Frameworks for interoperating between clouds and operation of the Open Cloud Testbed.</td>
</tr>
<tr>
<td>Open Data Center Alliance</td>
<td>Unified customer vision for long-term data-center requirements. Developing usage models for cloud vendors. Independent IT consortium.</td>
</tr>
<tr>
<td>OpenStack</td>
<td>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.</td>
</tr>
<tr>
<td>Standards Acceleration to Jumpstart Adoption of Cloud Computing</td>
<td>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.</td>
</tr>
<tr>
<td>The Open Group Cloud Work Group</td>
<td>Works with other cloud standards organizations to show enterprises how to best incorporate cloud computing into their organizations.</td>
</tr>
<tr>
<td>TM Forum Cloud Services Initiative</td>
<td>Common approaches to increase cloud-computing adoption such as common terminology, transparent movement among cloud providers, security issues, and benchmarking.</td>
</tr>
</tbody>
</table>

The 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. 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). 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. 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.


### 3.2. Workload migration

This use case corresponds to the migration of a workload, typically represented as a VM 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 VM that can be deployed within Amazon EC2 and is also becoming a de facto standard. Eucalyptus and OpenStack support AMI as well.

- Open Virtualization Framework (OVF): OVF is a VM packaging standard developed and supported by DMTF. Cloud platforms that support OVF include Amazon EC2, Eucalyptus, and OpenStack.

- Virtual Hard Disk (VHD): VHD is a VM file format supported by Microsoft. Cloud platforms that support VHD include Amazon EC2 and Microsoft Azure.

### 3.3. Data migration

This use case 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 VM 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). 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. 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. 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 Force.com, Microsoft Windows Azure (Windows Azure Storage), OpenStack (Object Storage), and Rackspace (Cloud Files). The API defined by CDMI is a RESTful interface.

3.4. Workload management

This use case 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 RESTful APIs for workload management include Amazon EC2, Eucalyptus, GoGrid Cloud Servers, Google App Engine, Microsoft Windows Azure, and OpenStack (Image Service).

4. Role of standards in cloud-computing environments

From a use-case perspective, the main use cases that benefit from standardization are workload migration and data migration to mitigate existing vendor lock-in concerns. This requires standardization of VM image file formats and APIs for cloud storage respectively [2]. 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 to user authentication, 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.

4.1. Infrastructure as a service (IaaS)

IaaS is the service model that would most benefit from this type of standardization because the main building blocks of IaaS are workloads represented as VM images and storage units that vary from typed data to raw data [3].

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 [3]. 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 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 advantages are ease of development and tool availability. However, these standards are more useful for raw data that is not typed (e.g., VM 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. 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 (i.e., two SOAP-based interfaces can have completely different operations).

4.2. 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
4.3. 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. 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. In the latter case, standardized APIs are useful because they facilitate the development process. 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. In this case, the only area where SaaS would benefit from standardization is data storage because the most important concern for any SaaS consumer, especially for enterprise software SaaS such as CRM or human resources, is how to extract its data. 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. Thoughts and recommendations

5.1. Do standards make sense beyond IaaS?

The answer to this question is probably 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. Dillon et al [18] support this position by saying that standardization for PaaS would additionally require uniformity in the way that applications are deployed and developed in the cloud.

Expecting PaaS and SaaS providers to standardize feature sets is equivalent to asking ERP software vendors to standardize feature sets. This is not going to happen because it is not in their best interest.

5.2. What areas of cloud computing should be standardized first?

In 2005 a group of researchers from the European Union defined three generations of service-oriented systems [13]:

• 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 many years of research, we have not reached the point where third-generation service-oriented systems are of production quality [14] [15]. This is because dynamic service discovery and composition require agreements regarding data models and ontologies, service-level agreement (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 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 [8] [17]. 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.

5.3. Can existing standards support cloud interoperability in addition to portability, or do clouds require new standards?

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.

With interoperability defined as the ability of a collection of systems to exchange information and operate on it according to agreed-on semantics, there are two basic use cases for IaaS that exercise this service model’s potential for interoperability:

UC1: Workload \( W_i \) on Cloud \( C_1 \) can communicate with Workload \( W_j \) on Cloud \( C_2 \).

UC2: Workload \( W_i \) 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 [12]. Technical interoperability is about exchanging data, semantic interoperability is about exchanging meaningful data, and organizational interoperability is about participating in multi-organizational business processes. 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. Related work

There are many authors of blogs and electronic magazine articles that cite the importance and need for standards mainly to avoid vendor lock-in, some of which are cited in this paper. Other authors state that using standards is just “one piece of the cloud interoperability puzzle” because achieving interoperability will also require sound architecture principles and dynamic negotiation between cloud providers and users [5] [6] [7] [8]. This latter position is consistent with the views presented in this paper.

The research community has also been actively investigating the role of standards in cloud computing interoperability. The focus ranges from simply stating that standards are necessary for achieving interoperability to presenting approaches to support the third-generation cloud-based systems mentioned earlier. Work that is representative of the research related to the role of standards includes that of Bernstein et al [16] that defines “Intercloud Protocols” as a candidate base set of standards and formats to enable cloud interoperability. The identified standards fully or partially support the basic use cases identified in Section 3, in addition to lower-level functions such as network addressing and resource discovery that would be necessary to realize third-generation systems. Loutas et al [19] performed a comprehensive survey of cloud computing standards specifically looking at semantics. Consistent with our findings, they report the existence of a large number of standards and the need for semantics to support higher levels of cloud interoperability. They also conclude that trying to agree on a single standard (or set of standards, APIs and models) is impossible and therefore propose the creation of cloud brokers. However, these brokers would still have to support a common standard that would serve as a mediator between cloud systems. Teckelmann et al [20] performed a study on the role of standards specifically for IaaS. They map a subset of existing standards to a taxonomy that is divided into access mechanism, virtual appliance, storage, network, security, SLA, and others, and provide information on support for each of these areas.
et al [17] also state that cloud interoperability problems will not be solved by standards alone and advocate the use of semantic models for describing cloud application functional and non-functional details, data modeling. Dillon et al [18] discuss standardization for IaaS, PaaS and SaaS and define cloud interoperability as information exchange not only between clouds but also between clouds and internal systems.

The main contribution of this paper is not the identification but rather the analysis of areas of cloud computing where standards would be useful for interoperability and areas where standards would not help or would need to mature to provide value, especially in the context of e-Government.

7. Conclusions

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 a business problem, not a technical problem. 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. 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 services. 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. But some have simply died because of lack of support, such as WS-Privacy and WS-Authorization. The practice of standardization also evolves. 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. Cloud computing is currently going through what web services went through in 2006. It will take some time for standards to emerge. This point emphasizes the importance of software architecture for cloud-based systems in which standards-reliant components should be implemented as separate 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.

System developers should leverage standards to support the architecture of a system, but standards should not drive the architecture of a system [7]. 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 2, 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.

From an e-Government perspective, interoperability has been a challenging issue for governments in the traditional client-server world and will continue to be in the cloud world. While most practitioners agree that interoperability is a multifaceted challenge, governments have often struggled to find the right role and emphasis for standards, sometimes viewing standards as a “cure-all” solution. The cloud brings with it many opportunities for governments, but like any technology, it also will bring interoperability challenges. To successfully navigate to the cloud, and reap its benefits, government policy makers will need to understand the role of standards in cloud computing. This paper hopes to shed some light on this understanding.
Acknowledgements

Copyright 2012 Carnegie Mellon University and IEEE. This material is based upon work funded and supported by the 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 material has been approved for public release and unlimited distribution (DM-0000041).

References