SAJACC Working Group Status

Lee Badger
Jin Tong

April 8, 2011

Disclaimer: Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.
Standards Acceleration to Jumpstart Adoption of Cloud Computing (SAJACC)

Original Concept
March 2010

NIST will develop and populate the portal:
1. By direct insertion and validation of legacy specifications.
2. By validating contributed specifications.

NIST Cloud Standards Portal
- Use Cases
- Validated Specifications
- Reference Implementations

Community Outreach
- Existing Standards Working Group

Specifications
spec 1
spec 2
...
spec n

Use Cases
Case 1
Case 2
...

Validation Exercises
- Spec 1
- Spec 2
- ...
- Spec n
- Test 1
- Test 2
- ...
- Test n

- Shorter Standards Development Times.
- Number of downloads.
Acknowledgments

266 WG members
12 Weekly Meetings

Jan 11   Jan 18   Jan 25   Feb 1   Feb 8   Feb 15   Feb 22   Mar 1   Mar 8   Mar 15   Mar 22   Mar 29

special contributions:

Winston Bumpus (VMware) setup web-conf system, DMTF/OVF tie-in.

Alan Sill (TTU) GRID interoperability, identity management, OCCI standard.

Mark Carlson (Oracle, SNIA) CDMI help.

Bob Marcus (ET-Strategies) standards tie-in.

Babak Jahromi (Microsoft) use case implementations.

Kong Li (US Navy SPAWAR Knowcean) use case test drivers.

more….
Objectives and Charter

1. Develop and maintain **use cases** through an Open Process.

   **24 Use Cases**
   www.nist.gov/itl/cloud/use-cases.cfm

   (goal: highlight INTEROPERABILITY, PORTABILITY, SECURITY)

2. Collect and generate cloud **system interfaces** through an Open Process.

   collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/CloudInterfaceCatalog

3. Develop **tests** showing how interfaces can satisfy use cases.


Plan:

- Identify Interfaces
- Portal
- First test case run
- more runs
- Test Report
- Testing on Portal

Dec | Jan | Feb | Mar | Apr | May
Reminder: The NIST SAJACC Process

- **specifications, use cases**: provide insight on how clouds can work
- **reference implementations**: enable validation exercises
- **continuously growing portal**: new content added over time
- **publically available**: anyone can access
Prioritized Use Cases

**Management**
- 3.1 Open An Account
- 3.2 Close An Account
- 3.3 Terminate An Account
- 3.4 Copy Data Objects Into A Cloud
- 3.5 Copy Data Objects Out of a Cloud
- 3.6 Erase Data Objects In a Cloud
- 3.7 VM Control: Allocate VM Instance
- 3.8 VM Control: Manage Virtual Machine Instance State
- 3.9 Query Cloud-Provider and Capacities

**Interoperability**
- 4.1 Copy Data Objects between Cloud-Providers
- 4.2 Dynamic Operation Dispatch to IaaS Clouds
- 4.3 Cloud Burst From Data Center to Cloud
- 4.4 Migrate a Queuing-Based Application
- 4.5 Migrate (fully-stopped) VMs from one cloud-provider to another

**Security**
- 5.7 Sharing of access to data in a cloud

Future

...
## Getting Coverage - Testing

<table>
<thead>
<tr>
<th>SAJACC Use Cases</th>
<th>Pass 1</th>
<th>Pass 2</th>
<th>Pass 3</th>
<th>Pass 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case 1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Use Case 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use Case 3</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Use Case 4</td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Use Case n</td>
<td>✔️</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>
Test Artifacts

SAJACC Use Cases

1. Use Case 1
   - Interface A
   - Interface B
   - Interface M

2. Use Case 2
   - Interface A
   - Interface B
   - Interface M

Use Case n

Identified Interfaces

Develop Test Drivers

Execute Against Provider Endpoints

Acme Cloud
Cloud XYZ
NIST Lab

Produce Test Report

Results
## Catalogue of Cloud Interfaces

<table>
<thead>
<tr>
<th>Name</th>
<th>Functional Area</th>
<th>Operation Count</th>
<th>Web Access</th>
<th>Client Language Binding</th>
<th>Selected Implementations</th>
<th>OSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCCI</td>
<td>IaaS Mgmt (PaaS/SaaS)</td>
<td>14</td>
<td>RESTful</td>
<td>RESTful client</td>
<td>OpenNebula, SLA@SOI, SSF, among others</td>
<td></td>
</tr>
<tr>
<td>EC2</td>
<td>IaaS Mgmt</td>
<td>73</td>
<td>SOAP, RESTful</td>
<td>Java, .NET PHP Python Ruby</td>
<td>Eucalyptus, OpenNebula, AWS, OpenStack</td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>Storage</td>
<td>SOAP: 16 REST: 35</td>
<td>SOAP, RESTful</td>
<td>Java, .NET PHP Python Ruby</td>
<td>Eucalyptus, OpenNebula, AWS, OpenStack</td>
<td></td>
</tr>
<tr>
<td>Azure Storage Mgmt</td>
<td>Storage</td>
<td>42</td>
<td>RESTful</td>
<td>Java, .NET PHP Python</td>
<td>Windows Azure</td>
<td>Client</td>
</tr>
<tr>
<td>Azure Svc Mgmt</td>
<td>PaaS Mgmt</td>
<td>32</td>
<td>RESTful</td>
<td>Java, .NET Python</td>
<td>Windows Azure</td>
<td>Client</td>
</tr>
<tr>
<td>Cloud Servers</td>
<td>IaaS Mgmt</td>
<td>28</td>
<td>RESTful</td>
<td>Python</td>
<td>OpenStack, Rackspace</td>
<td></td>
</tr>
<tr>
<td>Cloud Files</td>
<td>Storage</td>
<td>17</td>
<td>RESTful</td>
<td>Java, .NET PHP Python Ruby</td>
<td>OpenStack, Rackspace</td>
<td></td>
</tr>
<tr>
<td>CDMI</td>
<td>Storage</td>
<td>28</td>
<td>RESTful</td>
<td>RESTful client</td>
<td>CDMI Ref Impl.</td>
<td></td>
</tr>
<tr>
<td>vCloud</td>
<td>IaaS Mgmt</td>
<td>85</td>
<td>RESTful</td>
<td>Java, .NET PHP</td>
<td>OpenNebula, Verizon, Biz Colt, Terremark, BlueLock, Virtacore</td>
<td></td>
</tr>
<tr>
<td>Deltacloud</td>
<td>IaaS Mgmt</td>
<td>7 groups</td>
<td>RESTful</td>
<td>Server: Ruby, Client: Java, Ruby, Python</td>
<td>(Client-side only)</td>
<td>Client Side</td>
</tr>
</tbody>
</table>
Choosing Initial Interfaces to Work With

NIST proposed, and WG fine tuned, **6 criteria for choosing** initial interfaces.

Characterized the 10 Interfaces using the criteria and chose:

1. **Data Management:**
   - Amazon Simple Storage Service (S3)
   - SNIA Cloud Data Management Interface (CDMI)

2. **Virtual Machine Management:**
   - Amazon Elastic Compute Cloud (EC2)

Intuitively map to use case.

Widely available as a network service.

Open Source implementation exists.

Relative simplicity.

Tester’s familiarity.

No significant intellectual property barriers.
**Current Status**

<table>
<thead>
<tr>
<th>Use Case Description</th>
<th>S3</th>
<th>CDMI</th>
<th>EC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4 Copy data objects into a cloud</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.5 Copy data objects out of a cloud</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.6 Erase data objects in a cloud</td>
<td></td>
<td>1/2</td>
<td></td>
</tr>
<tr>
<td>5.7 Sharing of access to data in a cloud</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1 Copy data objects between cloud providers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.7 Allocate VM Instance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.8 Manage VM Instance State</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

More on the way...

Test Drivers Developed

- S3
- CDMI
- EC2
public class UseCase3_8_XYZ {
    /**
     * The entry point of this test driver.
     * 
     */
    public static void main(String[] args) {
        String instanceId = null;
        ...
    }
}

SAJACC Use Cases

xyz_rsa_key.pem

xyz.properties

Test Driver
Java Code

3rd Party Libs
(Open Source)

Test Env Specific Configurations

Ant Build Tools

Documentations
(READMEs)

- Publicly available to download and run
- Standalone applications, cross platform
- Produce clear testing reporting
- Tests customizable and provider neutral
- 3rd party open source libraries included (batteries included)

“Batteries Included”

xy.properties

access-key = AbCdEfG...
secret-key = aBcDeFg...
service-endpoint = https://...
...
xyz_rsa_key.pem

-SAJACC Use Cases

-wildcard-
Code Walkthroughs

- **Jan 11**: UC 3.4 copy data objects into a cloud (S3)
- **Jan 18**: UC 3.4 copy data objects into a cloud (CDMI)
- **Jan 25**: UC 3.6 erase data objects in a cloud (S3)
- **Feb 1**: UC 5.7 sharing of access to data in a cloud (S3)
- **Feb 8**: UC 4.1 copy objects between cloud providers (S3, CDMI, EC2)
- **Feb 15**: UC 3.7 allocate VM instance (EC2)
- **Feb 22**: UC 3.8 manage VM instance state (EC2)
- **Mar 1**: UC 3.5 copy data objects out of a cloud (S3)
- **Mar 8**: UC 3.7 allocate VM instance (EC2)
- **Mar 15**: UC 3.8 manage VM instance state (EC2)
- **Mar 22**: UC 3.7 allocate VM instance (EC2)
- **Mar 29**: UC 3.8 manage VM instance state (EC2)
Driver Walkthrough of UC 4.1: Copy Data Objects between Cloud-Providers

1) create bucket
2) create object
3) ssh into the (running) VM
4) execute a command on the VM to download the object into the VM
5) execute command on the VM to create a CDMI container on provider 2
6) execute command on the VM to upload the object to CDMI in provider 2
7) execute a command at the user to verify that the object’s contents are correct
8) clean up provider 1
9) clean up provider 2
Selected Use Case Driver Illustrations

- UC 3.4 Copy Data Into a Cloud
- UC 5.7 Sharing Data Using ACL
- UC 4.1 Copy Between Clouds
- UC 3.5 (s3) Copy Data Out
- UC 3.7 Allocate VM
- UC 3.8 Control VM State
- UC 3.6 Erase Data Objects

- Driven from the subscriber side.
- Many alternative approaches possible.
A Few Observations

• No open support for direct cross-cloud copy (that we can find).
• Cloud-to-cloud implementations of common interfaces not widely deployed.
• Assured Cloud “Data Erase” appears to be problematic.
  – (because of replication, logs, expense, etc.)
• For data object and VM management, different clouds sometimes have quite similar core features.
• Federated identity management may facilitate cloud interoperability.
• Subtle differences (e.g., timing) in similar (or almost identical) interfaces limits portability.
  – (documented interface identity != compatibility)
Plan Going Forward

• Implement more use cases
  – Refine the Use Cases
  – Version 2 of the Use Case Document

• Cloud Interoperability focus
  – e.g., VM migration
  – e.g., identity management

• Encompass more of the available interfaces
  – re-implement the same use cases with more of the 10 known IaaS interfaces

• Expand the test driver infrastructure
  – e.g., mix/match multiple use cases + interfaces + providers, driven by configuration settings
Questions?
backup
A Use Case

**Use Case**: a description of how groups of users and their resources may interact with one or more systems to achieve specific goals.

<table>
<thead>
<tr>
<th><strong>Actors</strong>: the active entities</th>
<th><strong>Goals</strong>: what the use case tries to achieve</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Assumptions</strong>: conditions assumed true</td>
<td></td>
</tr>
</tbody>
</table>

**Success Scenario 1** (name, IaaS, PaaS, SaaS) A step-by-step narrative of what happens to achieve the use case goal

**Failure Conditions**: what might go wrong

**Failure Handling**: how to deal with known failures

**Success Scenario 2** (name, IaaS, PaaS, SaaS) Another narrative

**Failure Conditions**: what might go wrong

**Failure Handling**: how to deal with known failures

...  

**Credit**: any source that inspired us

We are using the approach of A. Cockburn, slightly customized

Actors

unidentified-user  An entity in the Internet (human or script) that interacts with a cloud over the network and that has not been authenticated.

cloud-subscriber  A person or organization that has been authenticated to a cloud and maintains a business relationship with a cloud.

payment-broker  A financial institution that can charge a cloud-subscriber for cloud services, either by checking or credit card.

cloud-provider  An organization providing network services and charging cloud-subscribers. A (public) cloud-provider provides services over the Internet.

Open An Account

unidentified-user

cloud-subscriber

payment-broker

cloud-provider

1

2

3

4

Currently, no taxonomy for actors

A use case
Actors (2)

**transport-agent**  A business organization that provides physical transport of storage media such as high-capacity hard drives.

**legal-representative**  A court, government investigator, or police.

**identity-provider**  An entity that is responsible for establishing and maintaining the digital identity associated with a person, organization, or (in some cases) a software program. [NSTIC]

**attribute-authority**  An entity that is responsible for creating and managing attributes (e.g., age, height) about digital identities, and for asserting facts about attribute values regarding an identity in response to requests. [NSTIC]

**cloud-subscriber-user**  A user of a **cloud-subscriber** organization who will be consuming the cloud service provided by the **cloud-provider** as an end user. For example, an organization's email user who is using a SaaS email service the organization subscribes to would be a **cloud-subscriber's user**.

**cloud-subscriber-administrator**  An administrator type of user of a **cloud-subscriber** organization that performs (cloud) system related administration tasks for the **cloud-subscriber** organization.

**cloud-user**  A person who is authenticated to a **cloud-provider** but does not have a financial relationship with the **cloud-provider**.
Current Use Cases

- **Management**
  - 3.1 Open An Account
  - 3.2 Close An Account
  - 3.3 Terminate An Account
  - 3.4 Copy Data Objects Into A Cloud
  - 3.5 Copy Data Objects Out of a Cloud
  - 3.6 Erase Data Objects In a Cloud
  - 3.7 VM Control: Allocate VM Instance
  - 3.8 VM Control: Manage Virtual Machine Instance State
  - 3.9 Query Cloud-Provider Capabilities and Capacities

- **Interoperability**
  - 4.1 Copy Data Objects between Cloud-Providers
  - 4.2 Dynamic Operation Dispatch to IaaS Clouds
  - 4.3 Cloud Burst From Data Center to Cloud
  - 4.4 Migrate a Queuing-Based Application
  - 4.5 Migrate (fully-stopped) VMs from one cloud-provider to another

- **Future**
- **Security**

*preliminary taxonomy*
Current Use Cases

Management

Interoperability

preliminary taxonomy

Security

5.1 Identity Management - User Account Provisioning
5.2 Identity Management - User Authentication in the Cloud
5.3 Identity Management - Data Access Authorization Policy Management in the Cloud
5.4 Identity Management - User Credential Synchronization Between Enterprises and the Cloud
5.5 eDiscovery
5.6 Security Monitoring
5.7 Sharing of access to data in a cloud

Future

6.1 Cloud Management Broker
6.2 Transfer of ownership of data within a cloud
6.3 Fault-Tolerant Cloud Group
A (simple) use case

Open An Account

Actors: unidentified-user, cloud-subscriber, payment-broker, cloud-provider.

Goals: Cloud-provider opens a new account for an unidentified-user who then becomes a cloud-subscriber.

Assumptions: A cloud-provider’s account creation web page describes the service offered and the payment mechanisms. An unidentified-user can access the cloud-provider’s account creation web page.

Success Scenario: (open, IaaS, PaaS, SaaS): An unidentified-user accesses a cloud-provider’s account creation web page. The unidentified-user provides: (1) a unique name for the new account; (2) information about the unidentified-user’s financial; and (3) when the unidentified-user wants the account opened. The cloud-provider verifies the unidentified-user’s financial information; if the information is deemed valid by cloud-provider, the unidentified-user becomes a cloud-subscriber and the cloud-provider returns authentication information that the cloud-subscriber can subsequently use to access the service.

Failure Conditions: (1) the unidentified-user does not provide a suitable name; (2) the financial information is not valid; (3) cloud-provider fails to notify the cloud-subscriber the account is open.

Failure Handling: For (1) and (2), new account is not created; For (3) See Use Case 3.2 below on failure handling related to notifications from cloud-provider to cloud-subscriber. Requirements File: None.

Credit: TBD
A (simple) use case

Open An Account

Actors: unidentified-user, cloud-subscriber, payment-broker, cloud-provider.

Goals: Cloud-provider opens a new account for an unidentified-user who then becomes a cloud-subscriber.

Assumptions: A cloud-provider’s account creation web page describes the service offered and the payment mechanisms. An unidentified-user can access the cloud-provider’s account creation web page.

Success Scenario: (open, IaaS, PaaS, SaaS): An unidentified-user accesses a cloud-provider’s account creation web page. The unidentified-user provides: (1) a unique name for the new account; (2) information about the unidentified-user’s financial; and (3) when the unidentified-user wants the account opened. The cloud-provider verifies the unidentified-user’s financial information; if the information is deemed valid by cloud-provider, the unidentified-user becomes a cloud-subscriber and the cloud-provider returns authentication information that the cloud-subscriber can subsequently use to access the service.

Failure Conditions: (1) the unidentified-user does not provide a suitable name; (2) the financial information is not valid; (3) cloud-provider fails to notify the cloud-subscriber the account is open.

Failure Handling: For (1) and (2), new account is not created; For (3) See Use Case 3.2 below on failure handling related to notifications from cloud-provider to cloud-subscriber. Requirements File: None.

Credit: TBD