

# Writing Java Applications for the Cloud

Doug Tidwell  
Cloud Computing Evangelist, IBM  
dtidwell@us.ibm.com

**Session 7663**



**SHARE** in Boston

# Agenda

- A brief overview of cloud computing and REST
- Cloud use cases and developer requirements
- Apache Wink
- The Simple Cloud API
- Controlling VMs with Apache libcloud
- Resources / Next steps

# A very few words about the cloud

And REST and WS-\*



**SHARE** in Boston

# The cloud

- Cloud computing . . . is a style of computing where IT-related capabilities are provided ‘as a service,’ allowing users to access technology-enabled services ‘in the cloud’ **without knowledge of, expertise with or control over the technology** infrastructure that supports them.
  - From Wikipedia
- Everybody has a slightly different idea of what cloud computing really is.
  - *This definition has been edited 350+ times in 2009.*

# REpresentational State Transfer

- Again, from Wikipedia:
  - Application state and functionality are resources
  - Every resource can be accessed via a URI
  - Simple interface: POST, GET, PUT, DELETE (think create, read, update, delete)
  - Cacheable, stateless protocol
- Clearly defined in Roy Fielding's seminal Ph.D. dissertation
- Everybody has a slightly different idea about what REST really is.
  - *The definition above has been edited 5,000+ times since I started reading this slide.*

## WS-\*

- We'll refer to WS-\* as SOAP, WSDL and standards such as WS-Security, WS-ReliableMessaging, WS-Conversation, etc.
  - Proponents of REST often call this “Big Web Services.”
- This means packaging data in a SOAP envelope, using WSDL to determine the service's interface and endpoint, and using the WS-\* standards to add headers to the SOAP envelope as needed.

## A plea for sanity

- It's easy to find avid proponents of REST or WS-\* out there.
  - You can find borderline psychotic proponents, too!
- As always, **the sane answer is to examine the two technologies and figure out which meets your needs.**
  - Maybe it's both.
  - David Chappell's blog post ["REST vs. WS-\\*: War is Over \(If You Want It\)"](#) is a good read. As are the responses.

# Cloud use cases and developer requirements



**SHARE** in Boston

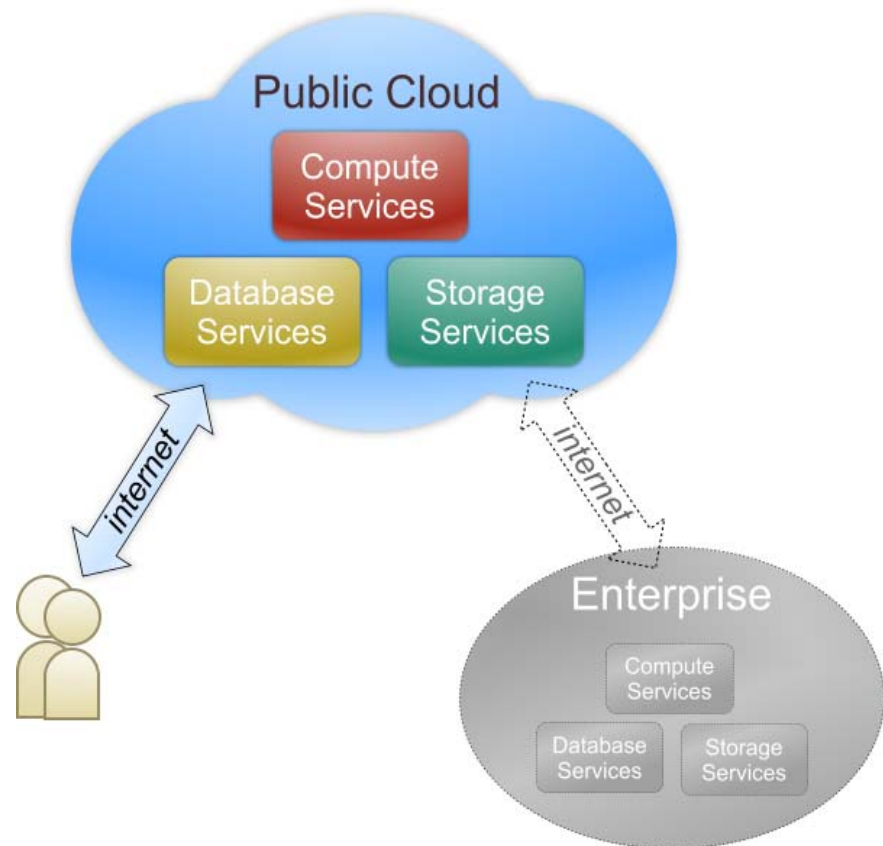


# Customer scenarios

- There are seven ways customers are using the cloud:
  1. End User to Cloud
  2. Enterprise to Cloud to End User
  3. Enterprise to Cloud
  4. Enterprise to Cloud to Enterprise
  5. Private Cloud
  6. Changing Cloud Vendors
  7. Hybrid Cloud

## End User to Cloud

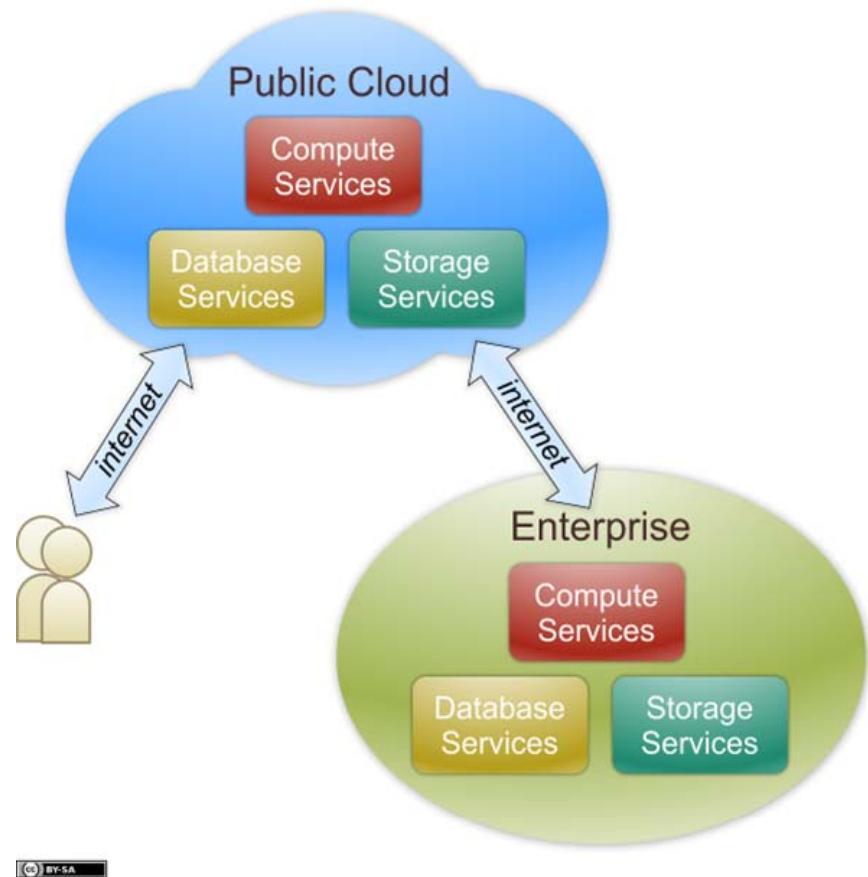
- End users access applications and data running in the cloud
  - Users have no idea they are using the cloud



CC BY-SA

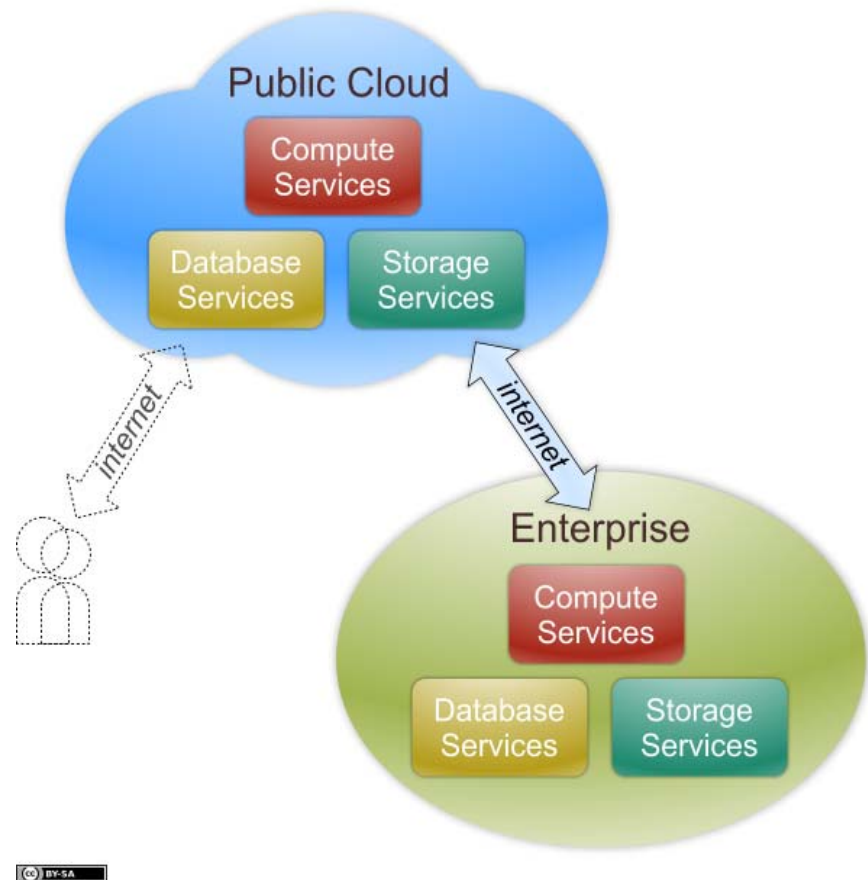
# Enterprise to Cloud to End User

- Applications are hosted in the public cloud and are accessed by users and customers



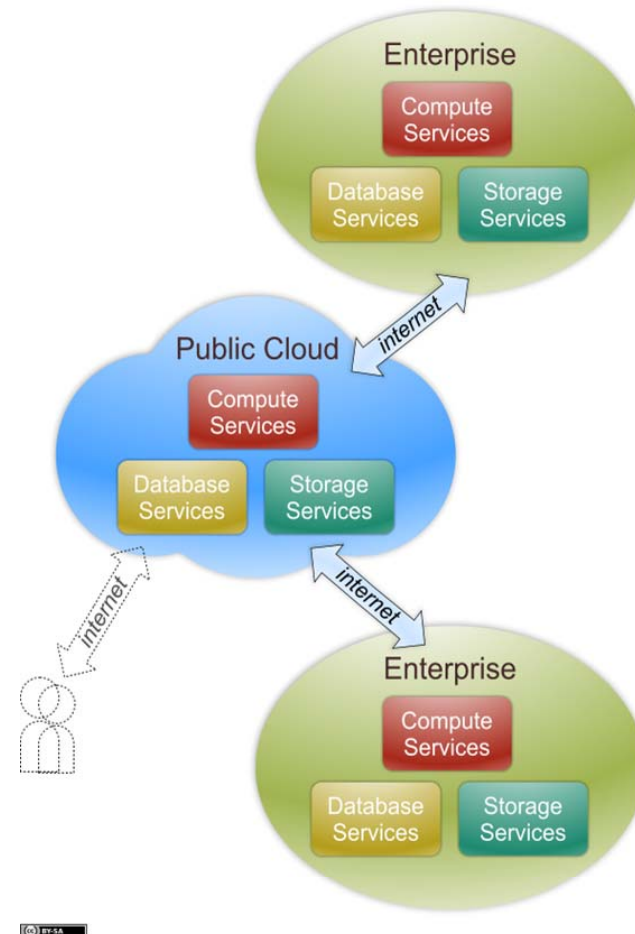
# Enterprise to Cloud

- Cloud applications are integrated with internal IT systems



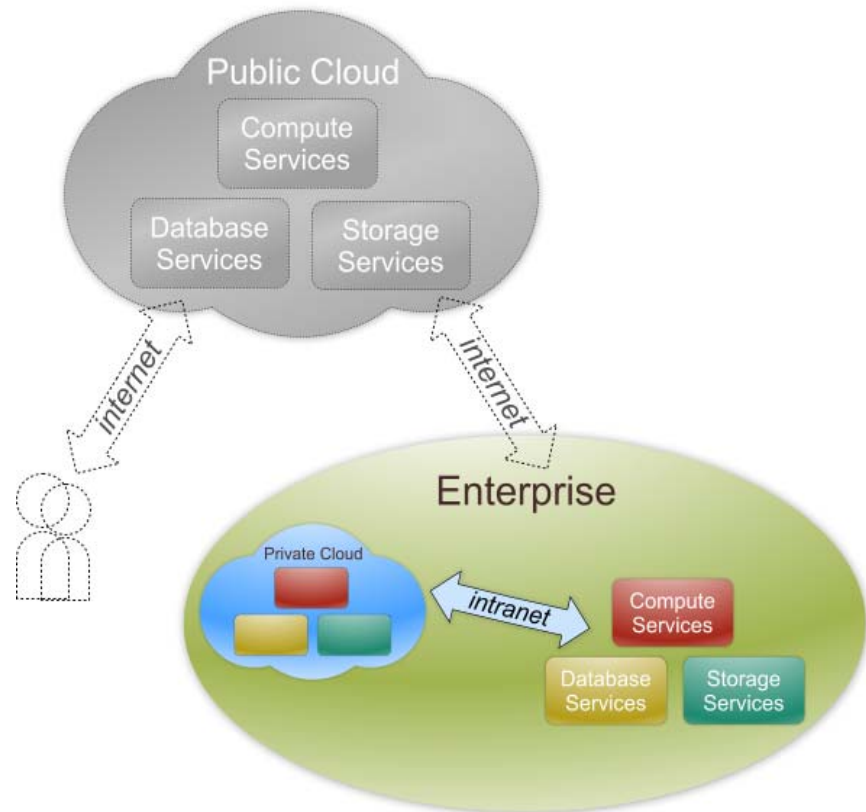
## Enterprise to Cloud to Enterprise

- Applications run in the public cloud and interoperate with IT systems in multiple enterprises



# Private Cloud

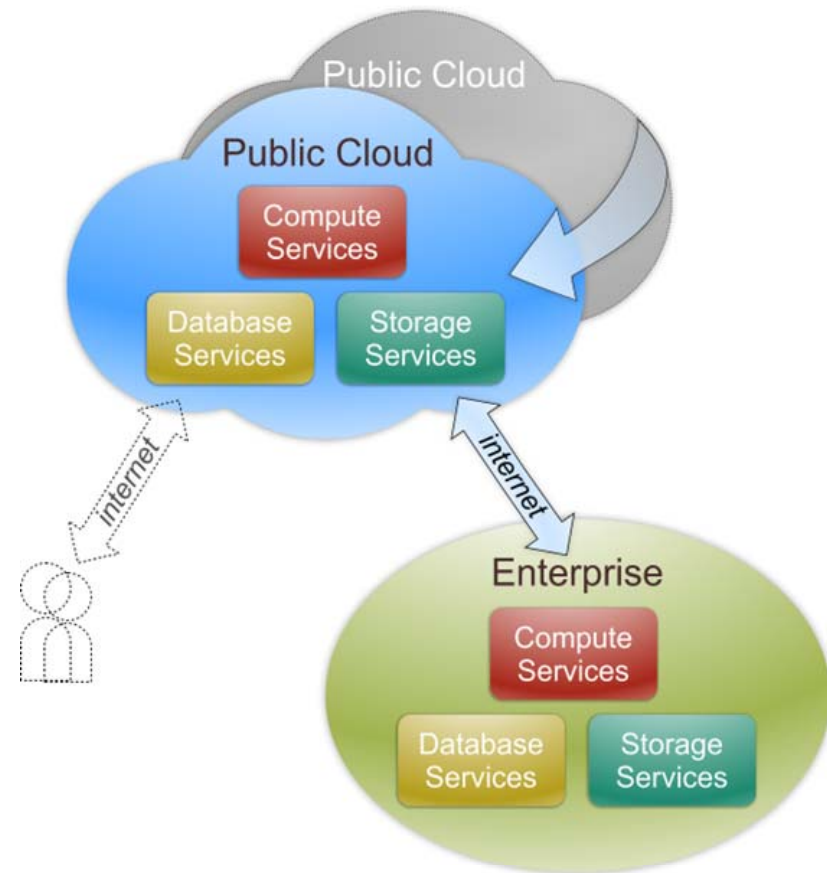
- The entire cloud is inside an enterprise's firewall.
  - The cloud infrastructure can be hosted by the enterprise or by a third party.



CC BY-SA

# Changing Cloud Vendors

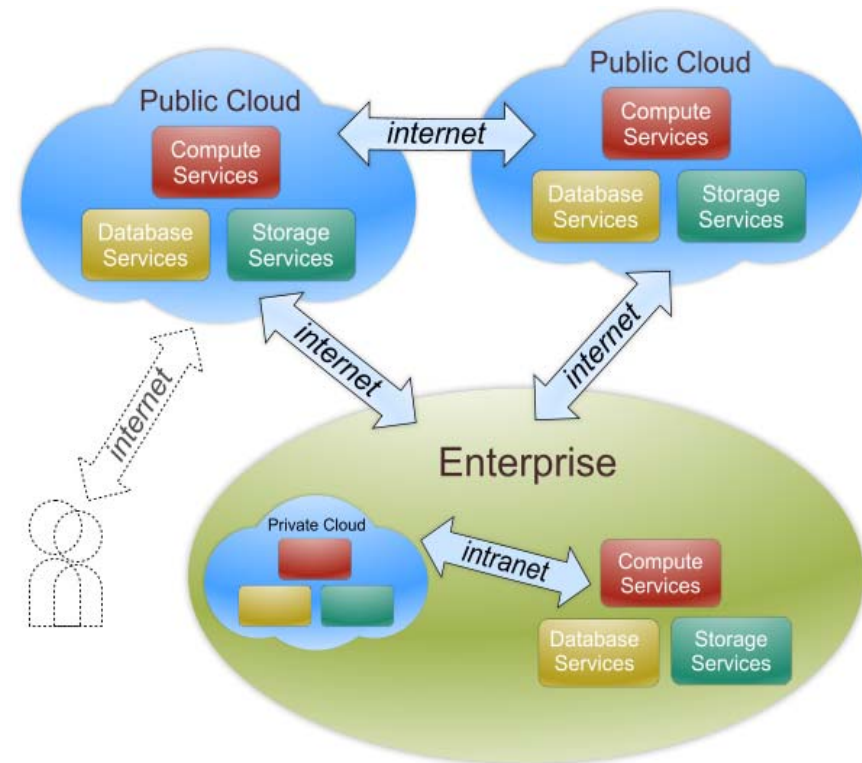
- An enterprise using cloud services decides to switch cloud providers or work with additional cloud providers



BY-SA

# Hybrid Cloud

- Multiple clouds work together, coordinated by a cloud broker that federates data, applications, user identity, security and other details.





## Categories of APIs

1. **Ordinary programming** – Nothing cloud-specific here
2. **Deployment** – APIs to deploy applications to the cloud.
  - This includes traditional packaging mechanisms in addition to cloud-specific techniques.
3. **Cloud services** – APIs to invoke storage, queues, databases, etc.
4. **Image and infrastructure management** – APIs to find images, start images, reboot instances, etc.
5. **Internal interfaces** – APIs for the internal interfaces between parts of a cloud infrastructure.

## Developer roles

- **Client application developers** – Writes cloud-based applications for end users (category 3, cloud service APIs)
- **Application developers** – Writes traditional applications that use the cloud (categories 1 and 3, ordinary programming and cloud services APIs)
- **Deployers** – Package, deploy and maintain applications that use the cloud (categories 2, 3 and 4: deployment, cloud services and image / infrastructure APIs)
- **Administrators** – Work with applications at multiple levels (categories 2, 3 and 4)
- **Cloud providers** – Work with internals (category 5)

## Developer requirements

- Caching
- Centralized logging
- Database
- Identity management
- Messaging – Both point-to-point and pub-sub
- Raw compute / Job processing
- Session management
- Service discovery
- Storage
- SLAs (machine readable)

# Apache wink

[incubator.apache.org/wink](http://incubator.apache.org/wink)



**SHARE** in Boston

# Apache wink

- Apache Wink is an implementation of JSR-311 / JAX-RS.
- It makes it easy to create REST services as well as clients that invoke them.

## JSR-311/JAX-RS

- As the cloud matures, chances are you'll need to share data and services with partners.
- Chances are you'll need to deploy something with a REST interface.
- JSR-311, also known as JAX-RS, is a way of deploying Java code as a REST service.
- The spec makes it easy to deploy an annotated Java class as a resource accessible via HTTP.
- Annotations in your code define how different methods are addressed from a URL.

## A sample REST service

- Defining the path for the service:

```
@Path("/calculator/")
```

```
public class RestCalculatorServiceImpl  
    implements CalculatorService {
```

- The URL for the service is [baseUrl]/calculator/.

## A sample REST service

- Defining a method and parameters:

```
@GET
```

```
@Path( "/add/{n1}/{n2}/" )
```

```
public double add(
```

```
    @PathParam( "n1" ) double n1,
```

```
    @PathParam( "n2" ) double n2 ) {
```

```
    return n1 + n2;
```

```
}
```

- [baseUrl]/calculator/add/37.5/87.2/ returns.



## Other annotations

- `@Produces("text/html")`
- `@Consumes("application/atom+xml")`

## Running the service

```
JAXRSServerFactoryBean sf = new
    JAXRSServerFactoryBean();
sf.setResourceClasses
    (RestCalculatorServiceImpl.class);
sf.setResourceProvider
    (RestCalculatorServiceImpl.class,
     new SingletonResourceProvider(new
         RestCalculatorServiceImpl()));
sf.setAddress("http://ec2...aws.com/");
sf.create();
```

## Demo

- A quick look at our calculator service
  - Written using JSR-311
  - Deployed to an Amazon Machine Image running... *in the cloud!*
- We should get an integer when we load `http://ec2...com/calculator/add/37.5/87.2/`.

# The Simple Cloud API

[simplecloud.org](http://simplecloud.org)



**SHARE** in Boston



# The Simple Cloud API

- A joint effort of Zend, GoGrid, IBM, Microsoft, Nirvanix and Rackspace
  - But you can add your own libraries to support other cloud providers.
- The goal: Make it possible to write portable, interoperable code that works with multiple cloud vendors.
- There's an article on the Simple Cloud API in the developerWorks Open Source zone: [bit.ly/1bSkTx](http://bit.ly/1bSkTx)

# The Simple Cloud API

- Covers three areas:
  - File storage (S3, Nirvanix, Azure Blob Storage, Rackspace Cloud Files)
  - Document storage (SimpleDB, Azure Table Storage)
  - Simple queues (SQS, Azure Table Storage)
- Uses the Factory and Adapter design patterns
  - A configuration file tells the Factory object which adapter to create.

## Vendor-specific APIs

- Listing all the items in a Nirvanix directory:

```
$auth = array('username' => 'your-username',  
             'password' => 'your-password',  
             'appKey'   => 'your-appkey');  
$nirvanix = new Zend_Service_Nirvanix($auth);  
$imfs = $nirvanix->getService('IMFS');  
$args = array('folderPath' => '/dougtdwell',  
             'pageNumber' => 1,  
             'pageSize'   => 5);  
$stuff = $imfs->ListFolder($args);
```

- All of these lines of code are specific to Nirvanix.

## Vendor-specific APIs

- Listing all the items in an S3 bucket:

```
$s3 = new Zend_Service_Amazon_S3  
    ($accessKey, $secretKey);  
$stuff = $s3->getObjectsByBucket($bucketName);
```

- All of these lines of code are specific to S3.



# The Simple Cloud API

- Listing all the items in a Nirvanix directory or S3 bucket:

```
$credentials =  
    new Zend_Config_Ini($configFile);  
$stuff = Zend_Cloud_StorageService_Factory  
    ::getAdapter($credentials)->listItems();
```

- These lines of code work with Nirvanix and S3.
  - Which adapter is created and which storage is used is defined in the configuration file.

# Dependency injection

- The Simple Cloud API uses dependency injection to do its magic.
- A sample configuration file:

```
storage_adapter =  
    "Zend_Cloud_StorageService_Adapter_Nirvanix"  
auth_accesskey = "338ab839-ac72870a"  
auth_username = "skippy"  
auth_password = "/p@$w0rd"  
remote_directory = "/dougtdwell"
```

# Exciting demonstrations!

- **Prepare to be astounded** by the Simple Cloud API in action!
  - Due to cost constraints, we are unable to provide tissues for those moved to tears by the demonstration.
  - Persons prone to hyperventilation or motion sickness are advised to look away.
    - Be advised the management cannot be held responsible for your medical expenses.

## Issues

- Not all storage services support renaming files.
  - You can hack this, but....
- Not all storage services support listing containers.

# Issues

- How many messages are in a queue?
  - SQS lets you ask, Azure doesn't.
- Can I peek a message?
  - Azure lets you peek, SQS doesn't.
- What's the best way to handle this?
  - Introspection?
  - `instanceof`?
  - XSLT style? `system-property('xsl:is-schema-aware')`

# Controlling VMs with Apache libcloud



**SHARE** in Boston

# Apache libcloud

- A common library for controlling VMs in the cloud
  - Create, destroy, reboot and list instances, list and start images
- [incubator.apache.org/libcloud](http://incubator.apache.org/libcloud)

# Apache libcloud

- Find all the VMs I have running in the IBM, Slicehost and Rackspace clouds:

```
IBM = get_driver(Provider.IBM)
Slicehost = get_driver(Provider.SLICEHOST)
Rackspace = get_driver(Provider.RACKSPACE)
drivers =
    [ IBM('access key id', 'secret key'),
      Slicehost('api key'),
      Rackspace('username', 'api key') ]
# Now do what you like with your running VMs
```



## The libcloud interface

- `list_images()`
- `list_sizes()`
- `list_locations()`
- `create_node()`
- `list_nodes()`
- `reboot_node()`
- Other calls for querying UUIDs, locations, setting passwords, etc.

## Openness in action

- IBM has contributed a Java implementation of libcloud:
  - <https://svn.apache.org/repos/asf/incubator/libcloud/sandbox/java/trunk/>
- The Java implementation includes the basic framework plus an adapter for the IBM Smart Business Cloud.
  - Other adapters are underway...

# Summary / Resources / Next steps



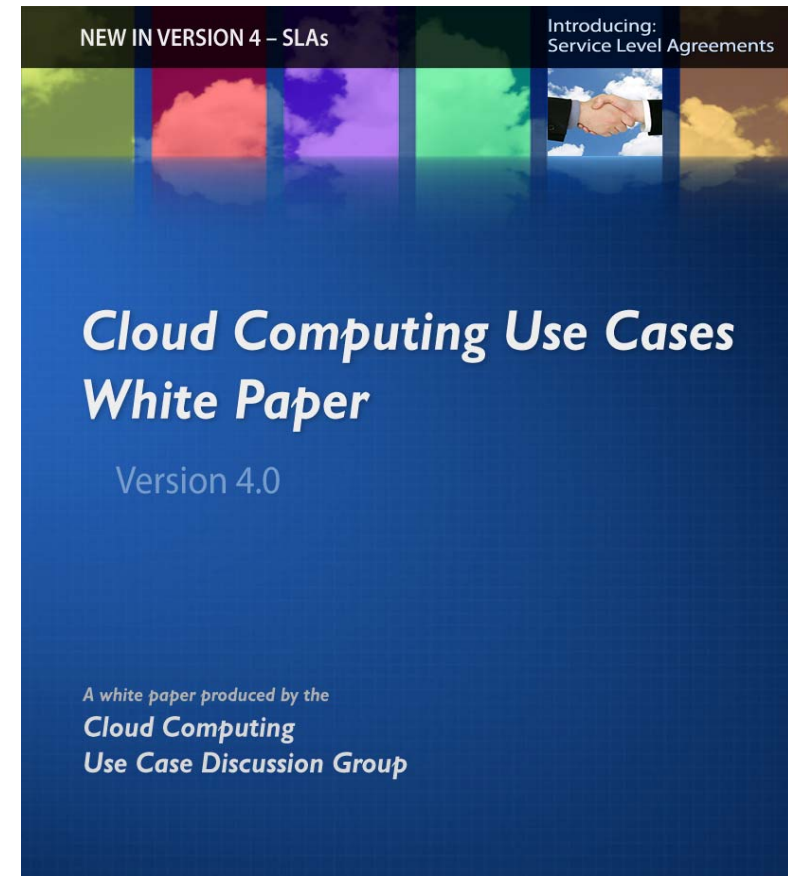
**SHARE** in Boston

# Get Involved!

- Simple Cloud API
  - Download the code, build a prototype, submit requirements / new adapters / bug reports
  - [simplecloud.org](http://simplecloud.org)
- libcloud
  - [incubator.apache.org/libcloud](http://incubator.apache.org/libcloud)

## cloudusecases.org

- The Cloud Computing Use Cases group is focused on documenting customer requirements.
- Covers Security, SLAs, developer requirements and cloud basics.
- **Join us!**



## Also available in Chinese



## Also available in Japanese

- Chinese discussion group on LinkedIn:
  - [linkedin.com/groups?gid=2919533&trk=myg\\_ugrp\\_ovr](https://www.linkedin.com/groups?gid=2919533&trk=myg_ugrp_ovr)
- Japanese discussion group and translated paper coming soon!

### クラウド・コンピューティング ユース・ケース

#### Cloud Computing Use Case Discussion Group

第 3.0 版

2010 年 2 月 2 日

コントリビューター: Dustin Amrhein, Patrick Anderson, Andrew de Andrade, Joe Armstrong, Ezhil Arasan B, James Bartlett, Richard Bruklis, Ken Cameron, Reuven Cohen, Tim M. Crawford, Vikas Deolaliker, Andrew Easton, Rodrigo Flores, Gaston Fourcade, Thomas Freund, Valery Herrington, Babak Hosseinzadeh, Steve Hughes, William Jay Huie, Nguyen Quang Hung, Pam Isom, Sam Johnston, Ravi Kulkarni, Anil Kunjunny, Thomas Lukasik, Bob Marcus, Gary Mazzaferro, Craig McClanahan, Meredith Medley, Walt Melo, Andres Monroy-Hernandez, Dirk Nicol, Lisa Noon, Santosh Padhy, Greg Pfister, Thomas Plunkett, Ling Qian, Balu Ramachandran, Jason Reed, German Retana, Bhaskar Prasad Rimal, Dave Russell, Matt F. Rutkowski, Clark Sanford, Krishna Sankar, Alfonso Olias Sanz, Mark B. Sigler, Wil Sinclair, Erik Sliman, Patrick Stingley, Robert Syputa, Doug Tidwell, Kris Walker, Kurt Williams, John M Willis, Yutaka Sasaki, Michael Vesace, Eric Windisch, Pavan Yara, Fred Zappert

この文書に関するコメントは <http://groups.google.com/group/cloud-computing-use-cases> にお寄せください。皆様のコメントをお待ちしています。

『クラウド・コンピューティング・ユース・ケース ホワイト・ペーパー』は、オープン・コミュニティのアプローチを用いて作成され、1,300 を超える世界中のメンバーからなるコミュニティからのインプットがベースとなっています。メンバーには小規模および大規模企業、政府機関、コンサルタント、サプライヤーも含まれており、クラウド・コンピューティング・コミュニティの要件を反映する構成となっています。



この著作物は、[Creative Commons Attribution Share Alike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/) の下で認可されています。

## developerWorks cloud zone



- Dozens of articles on cloud computing, including introductions, code samples, tutorials and podcasts.
- **[ibm.com/developerworks/cloud](http://ibm.com/developerworks/cloud)**



## One more time...

- **<hype>**  
Cloud computing will be the biggest change to IT since the rise of the Web.  
**</hype>**
- But to make the most of it, we have to keep things open.
- And everybody has to get involved to make that happen.

# Thanks!

Doug Tidwell  
Cloud Computing Evangelist  
dtidwell@us.ibm.com

**This is session 7663.**



**SHARE** in Boston