Keeping Your Options Open, Even if the Cloud is Not

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

Session 7635

SHARE in Boston

Agenda

- A very few words about the cloud
- How standards work
- Principles of openness
- Levels of APIs
- Openness in action:
  The Simple Cloud API and Apache libcloud
- Getting involved

A very few words about the cloud

"Cloud is more of a delivery model than a new technology."

– IBM's Ric Telford

Before the Web

- If you wanted to sell things to the public, you needed a storefront.
- Massive cost in real estate, fixtures, maintenance, shrinkage.
- Prohibitive cost to entry

What if...

- You could have hundreds of millions of storefronts worldwide?
  - Without real estate
  - Without fixtures
  - Without maintenance
  - Without shrinkage
  - With [relatively] zero cost to entry.
- **The Web changed everything.**

Before the cloud

- If you wanted to start a enterprise, you needed an IT shop.
- Massive costs in hardware, software, power, cooling, administrative staff
- Prohibitive cost to entry

What if...

- You could have unlimited computing resources?
  - All the processing power you want
  - All the data storage you want
  - Data mining whenever you want
- **Cloud computing will be the biggest change to our industry since the rise of the Web.**

The cloud is here to stay

- Extremely stupid idea of assuming the entire planet just can not be bothered with their own data (nor the security thereof). As always there will be some who think they 'need' this. I hope this whole cloud [stuff] just goes away. Logistically speaking it will never be anything but a waste of money.
  - Posted by ____**@gmail.com**, who apparently just can not be bothered with their own email (nor the security thereof).

How standards work

How standards work

- For a standards effort to work, three things have to happen:
  - The standard has to solve a common problem in an elegant way.
  - The standard has to be implemented consistently by vendors.
  - Users have to insist that the products they use implement the standard.

How standards work

- **All three things have to happen.**
  - If the standard doesn't solve a common problem, or if it solves it in an awkward way, the standard fails.
  - If the standard isn't implemented by anyone, the standard fails.
  - If customers buy and use products even though they don't implement the standard, the standard fails.

Portability and Interoperability

- In writing code for the cloud, there are two key concepts:
  - **Portability** is the ability to run components or systems written for one cloud provider in another cloud provider's environment.
  - **Interoperability** is the ability to write one piece of code that works with multiple providers, regardless of the differences between them.

Portability

- The portability of your work depends on the platform you choose and the work you're doing.
  - A GAE application
  - An Azure application
  - An AMI hosting an application container
  - A SimpleDB database
  - An Amazon RDS database

Interoperability

- Discussions of openness often focus on leaving one cloud provider and moving to another.
- In reality, it's far more common that you'll have to write code that works with multiple providers at the same time.

Vendor lock-in

- If there's a new technology, any talented programmer will want to use it.
  - Maybe the shiny new thing is appropriate for what we're doing.
  - Maybe not.
  - We're probably going to use it anyway.
- The challenge is to walk the line between using the newest, coolest thing and avoiding vendor lock-in.

Principles of openness

1. Cloud providers must work together to ensure that the challenges to cloud adoption are addressed through **open collaboration and the appropriate use of standards.**
2. Cloud providers must not use their market position to lock customers into their particular platforms and limiting their choice of providers.
3. **Cloud providers must use and adopt existing standards wherever appropriate.** The IT industry has invested heavily in existing standards and standards organizations; there is no need to duplicate or reinvent them.

Principles of openness

4. When new standards (or adjustments to existing standards) are needed, we must be judicious and pragmatic to avoid creating too many standards. **We must ensure that standards promote innovation and do not inhibit it.**
5. **Any community effort around the open cloud should be driven by customer needs**, not merely the technical needs of cloud providers, and should be tested or verified against real customer requirements.
6. Cloud computing standards organizations, advocacy groups, and communities should work together and stay coordinated, making sure that efforts do not conflict or overlap.

An extemporaneous diatribe on the timing of standards

Utility, uniqueness and “Haven’t we seen this before?”

A few words about APIs

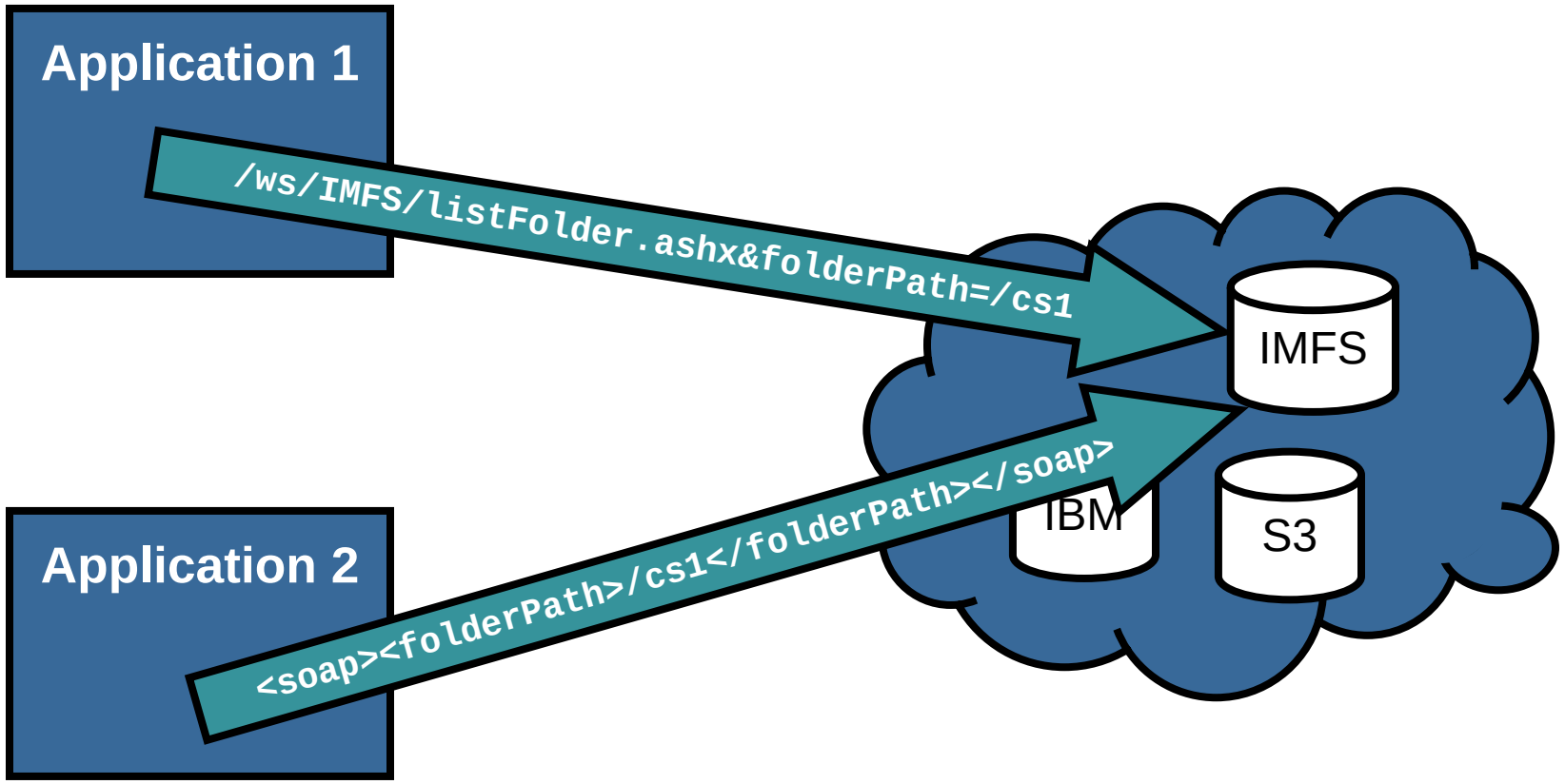
Levels of APIs

- How developers invoke a service:
  - Level 1 – Write directly to the REST or SOAP API.
  - Level 2 – Use a language-specific toolkit to invoke the REST or SOAP API.
  - Level 3 – Use a service-specific toolkit to invoke a higher-level API.
  - Level 4 – Use a service-neutral toolkit to invoke a high-level API for a *type* of service.

Level 1 – The Wire

- Developers write directly to the REST or SOAP API.
  - Developers must deal with all the details of the request, including URL formats, XML parsing, and HTTP headers and response codes.
  - Writing code at this level is rarely done today in the SOAP world. REST services (simple ones in particular) are still invoked this way.

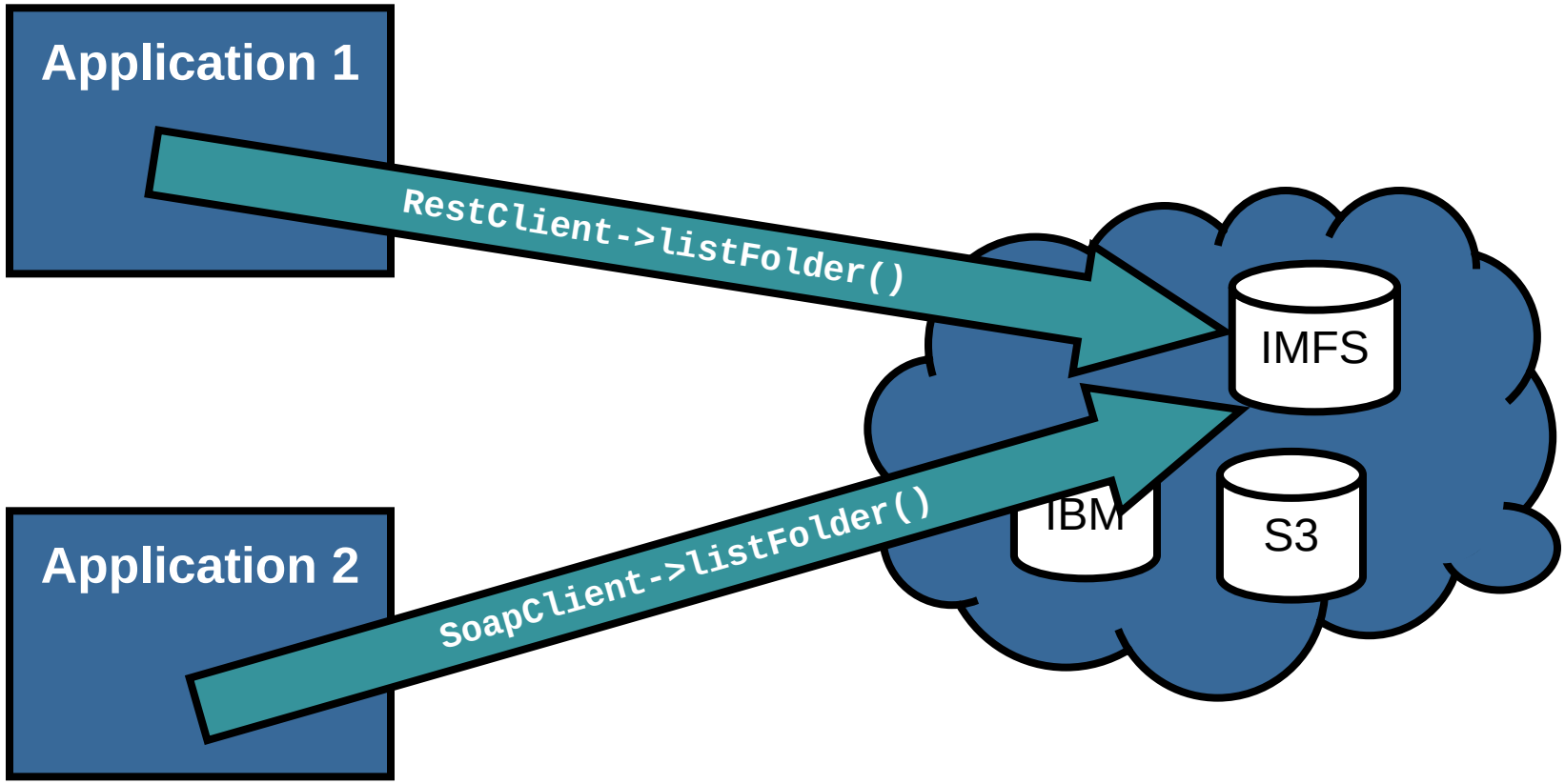
Level 1


Application 1
/ws/IMFS/listFolder.ashx&folderPath=/cs1
IMFS
IBM
S3
Application 2
<soap><folderPath>/cs1</folderPath></soap>

Level 2 – Language-specific

- Developers use a language-specific toolkit to build the REST or SOAP requests.
  - Slightly higher-level than writing directly to the REST or SOAP APIs.
  - Details such as managing HTTP error codes or XML parsing are handled by the toolkit.

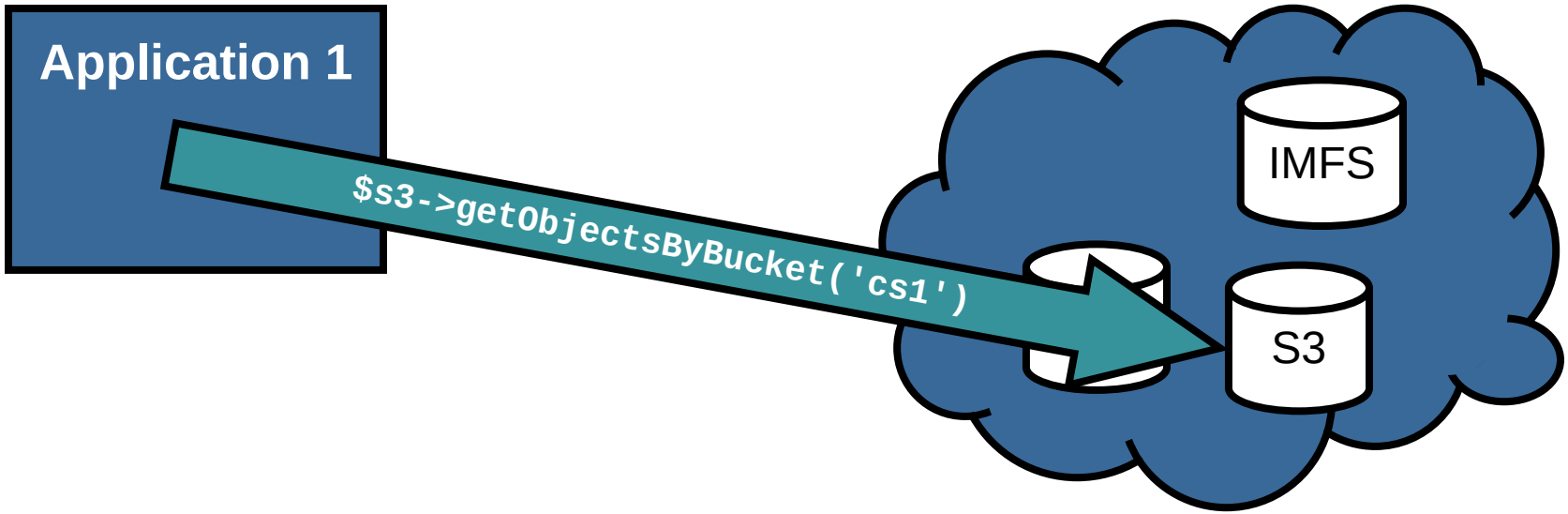
Level 2


Application 1
RestClient->listFolder()
IMFS
Application 2
SoapClient->listFolder()
IBM
S3

Level 3 – Service-specific

- Developers use objects that wrapper a particular service.
  - Developers don’t know if they’re using REST or SOAP.
  - Developers focus on using a particular service to get something done.

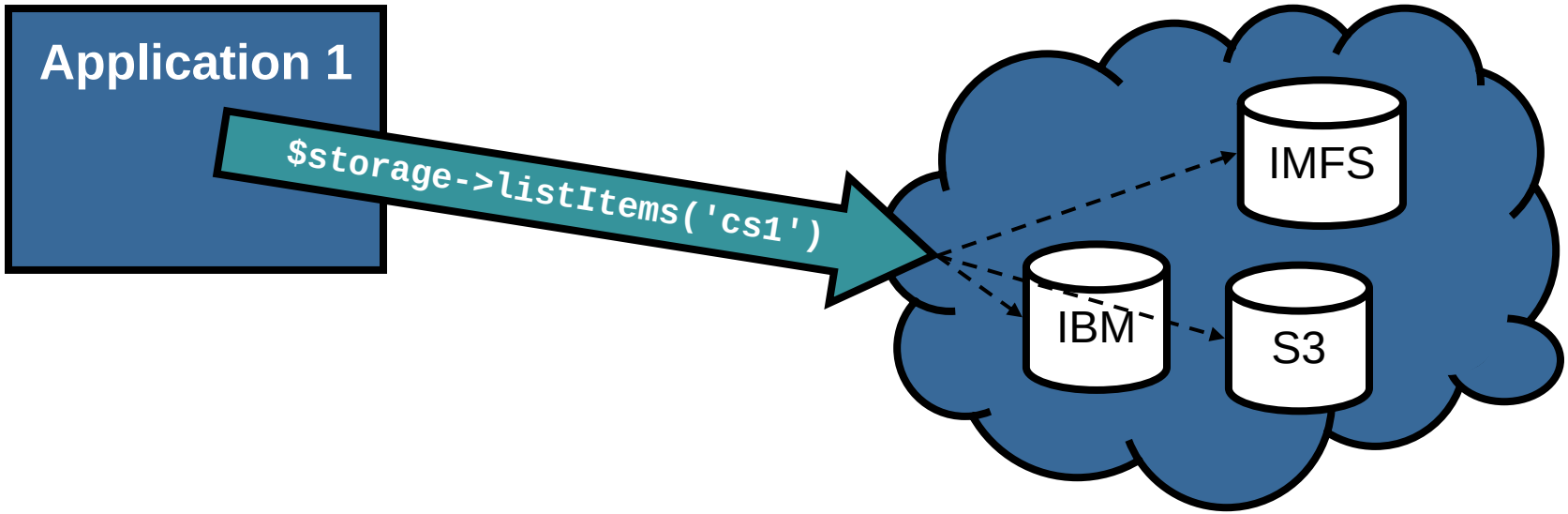
Level 3



Level 4 – Service-neutral

- Developers use objects that wrapper a particular *type* of service.
  - Developers have no idea which service they're using.
  - **Developers focus on getting something done.**

Level 4



The Simple Cloud API

simplecloud.org

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

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' => '/dougtidwell',
              '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);
$stuff = Zend_Cloud_Storage_Factory::getAdapter
          ($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_Storage_Adapter_Nirvanix"
auth_accesskey = "338ab839-ac72870a"
auth_username = "skippy"
auth_password = "/p@$$w0rd"
remote_directory = "/dougtidwell"
```

Exciting demonstration!

- **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.

Controlling VMs with Apache libcloud

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

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.

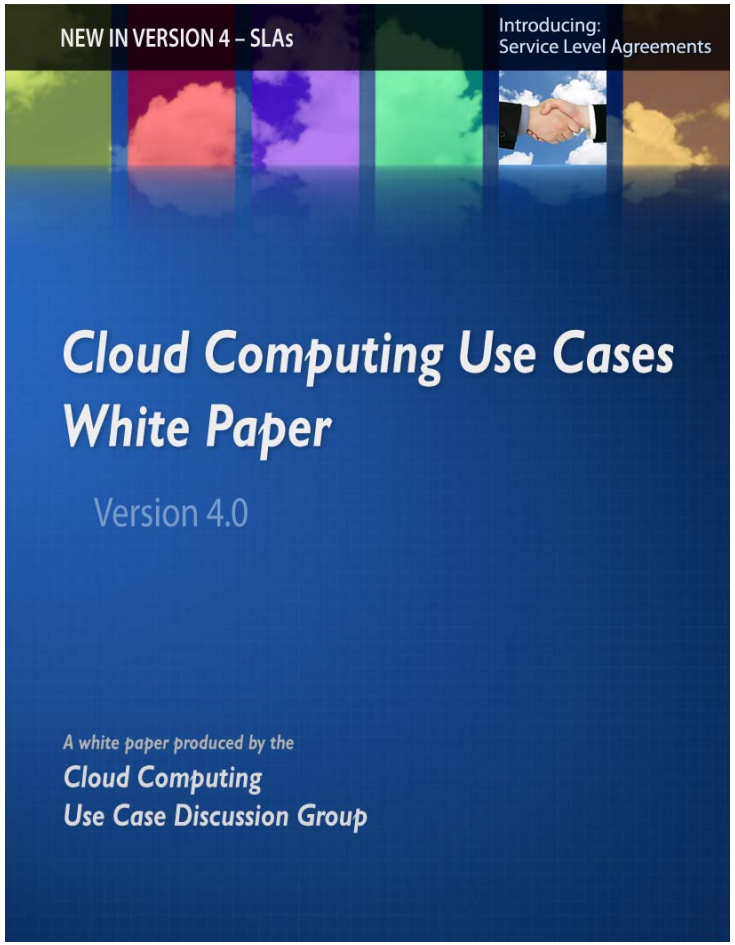
Openness in action

- 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...
- A Java implementation of Simple Cloud is underway.

Summary / Resources / Next Steps

cloudusecases.org – Get involved!

- 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

cloudusecases.org

- Chinese discussion group on LinkedIn:
  - 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・コミュニティー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ー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・コミュニティーの要件を反映する構成となっています。

developerWorks cloud zone – Get involved!

- Dozens of articles on cloud computing, including introductions, code samples, tutorials and podcasts.
- **ibm.com/developerworks/cloud**

Simple Cloud and libcloud – Get Involved!

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

Issues with the Web, 1994

- “It's not secure.”
- “I don't want to lose control of my infrastructure.”
- “I don't know how reliable it is.”
- “I don't know if my partners and customers are going to use it.”
- **All of these were important, legitimate issues.**
  - With VPNs and SSL and other technologies, the industry solved these problems.

Issues with the Cloud, 2010

- “It's not secure.”
- “I don't want to lose control of my infrastructure.”
- “I don't know how reliable it is.”
- “I don't know if my partners and customers are going to use it.”
- **All of these are important, legitimate issues.**
  - We've got some work to do, but the massive economic opportunities involved mean someone will find a way to solve these problems.

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.

More details on the code...

- ...will be covered in two sessions tomorrow:
  - **Writing Java Applications for the Cloud**
    9:30 in Room 200
  - **Open Cloud Computing with libcloud and the Simple Cloud API**
    1:30 in Room 310

Thanks!

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

This is session 7635.