Developing a Cloud Client

David Moolenaar  
Lead Software Manager, VP West Coast

Michael Behrens  
Chief Technology Officer

R2AD, LLC
High Level View

- Public Internet
- Private Network

JavaFX GUI

2 or 3 client machines

Cloud Layer Software
- OCCI – Open Nebula
- CDMI RI
- Glue code between

Compute & Hosting Infrastructure
- Raw x86 based machines (Sun Solaris)
- Virtual Machine instances

Storage
- Storage Array or NAS
- Public Clouds (being proxied)
Development Environment

- NetBeans 6.9.1 IDE with Glassfish and JavaFX modules
Development Experience

- Remote Testing, Access
  - Used Teleclient® Solution which incorporated Sun Secure Global Desktop for Full Screen Server access and control.
  - SSH and WinSCP also used
  - VMware’s vSphere™ client came in very handy
Reference implementations:

- CDMI: Not initially secured, HTTP was simple
- OCCI: Secure from start, HTTPS headers needed
  - Initial libraries/code for HTTPS did not work
  - Used manual creation of authentication headers

Also used a RackSpace account to test security logic – worked well.

- We like the RackSpace security model...after initial login, a dynamic endpoint is provided and authorization token must be in each header request
The Client

A Standards Based Cloud Client
Used JavaFX
Developed by R2AD, LLC
Sponsored by DISA CTO
cloud.r2ad.net
DISA Sponsorship

- DISA is interested in Cloud Technology
  - Storage and Compute, basically IaaS
  - Application, basically SaaS and PaaS
- They have an existing cloud, which they call RACE
  - Rapid Access Computing Environment
    - http://www.disa.mil/race
- Tracking vendor neutral standards from SDO’s:
  - SNIA, OGF, DMTF, etc.
- Very interested in Use Cases and Commercial Adoption levels
Client Requirements

- Simple/Small
  - Implement use-case (first with simple get/put)
  - Keep UI footprint small – potential phone app
  - UI became more sophisticated with experience

- Flat list to Tree
- Added Tabs
- Icons
- Edit Resource Location
- Learning JavaFX
- Authentication Models
- Added Log

April 2010 — June 2010
Client Interactions

- Demo communicates using OCCI + CDMI
  - Clear specifications and examples very helpful
  - Used local test server to code up initial ReSTful Interface: GET/PUT
    - Used various debugging tools such as RESTTest FireFox plugin
  - Use case driven (keeping goals on target)
Client Lesson Learned

- Code for network not being reliable
  - Let user cancel requests. Timeouts.
  - Use local cache or persistence?

- Use of remote admin tools helpful
  - ssh terminal, vSphere admin console
  - Virtual Desktop Interface (VDI)
    - Teleclient Solution w/Sun Secure Global Desktop or VNC

- Virtual Machines for servers helpful
  - Snapshots, transportability, etc.

- Way Ahead:
  - Make code more efficient (reduce server calls, parsing)
  - More integration with OCCI/CDMI – more features, security
  - Looking for feedback and chance to develop more
Client Side Tools Helpful

- TCP Monitor to capture outbound and inbound traffic. We used tcpmon for CDMI testing:
  - https://tcpmon.dev.java.net
- FireFox RESTTest also helpful
- Specify Headers IAW spec
NetBeans IDE – version 6.9.1
Demo available on-line

http://cloud.r2ad.net

OCCI/CDMI Client Demo Access Page

cloud.r2ad.net

Join the Google Cloud Demo group for the demo.
Read the OCCI Implemented Guide
Examine the OCCI/CDMI Demo Home Page.
Read about OpenNebula as they are implementing the OCCI spec.
Read about SNIA’s CDMI and the Open Grid Forum (OGF) OCCI
Read about OCCI code.

To Access the clients...stay tuned! (working on mock-ups)

Mockup

Run the Client (work in progress):

Run the Client.

Source Code...
R2AD is releasing the source code for this OCCI/CDMI demonstration client under the BSD license.
Please visit the R2AD cloud client website for more information. Source for the community edition is now on github.

Future Test Links
Retrieve Action Test: OCCI Simulator Server
Debug Headers: OCCI Sim (debug headers)

Screenshot from our main hosting page
Android Cloud Client...next?

Developer Announcements

Google I/O

Thanks to everyone who visited us at Google I/O in San Francisco! Stay tuned for videos and slides from the Android sessions, which will be posted at the Google I/O website.

Learn more »

Get Android 2.2!

The Android 2.2 platform is now available for the Android SDK, along with new tools, documentation, and a new NDK. For information about new features and APIs, read the version notes.

If you have an existing SDK, add Android 2.2 as an SDK component. If you're new to Android, install the SDK starter package.

Learn more »

Download

The Android SDK has the tools, sample code, and documentation you need to create great Android apps.

Learn more »

Publish

Android Market is an online service that lets you distribute your apps to hundreds of millions of Android devices worldwide.

Learn more »

Contribute

Android Open Source Project gives you access to the entire platform source.

Learn more »

Target Devices

Done