ORCA 2.0 Release

- New Substrate API
  - Simplify integration of new substrates
- New drivers and handlers: NLR, Eucalyptus
- Simplified actor configuration file format
  - Define an actor with a few lines of XML
- Lease cancellation (vacate) support
  - Brokers can allocate resources more efficiently
- New controllers: inter-domain, xmlrpc (protogeni)

New Substrate API

- New simplified **IConcreteSet** implementation
  - **UnitSet** represents a collection of **Unit**-s
  - **Unit** designed to represent any resource unit
    - VM, VLAN, Testbed, etc.
- New **IShirakoPlugin** extension: **ISubstrate**
  - Implemented as **Substrate** and **AuthoritySubstrate**
  - Defines clearly the extension points for a new substrate
    - transferIn, modify, transferOut

New Substrate API (2)

- New authority Resource Control classes
  - **UnitsControl**
  - **SimpleVMControl**
  - **VMControl**
  - **VLANControl**
- First steps to simplify and generalize the web portal
  - New project: **webapp2**
  - Basic functionality in place
  - Some minor bugs still to be addressed

New Substrate API (3)

- COD is still supported but deprecated
  - Use **webapp** with COD
- COD and the new API cannot be mixed
  - Choose one or the other
  - Trivial to update COD-based code to the new API
- COD will not be included in the next release
  - Too much stale and dead code
  - Large source of confusion
  - Not worth supporting

New Drivers and Handlers

- Added two new drivers & handlers
  - Do not require a node agent for execution
- NLR
  - Provisions dynamically a path over NLR
- Eucalyptus
  - Provisions dynamically a VM from an Eucalyptus cluster
    - Can also provision from Amazon EC2 (same API)
  - Supports assigning VLANs to VMs
    - Requires a patch to Eucalyptus 1.5

Simplified Actor Configuration

- Extended the configuration format
  - Simplified descriptions
  - Sane defaults
  - Backwards compatible
- Define an actor with a few lines

```
<actor>
    <type>broker</type>
    <name>mybroker</name>
</actor>
```

Simplified Actor Configuration (2)

- Simplified resource pool configuration (authority)
- To configure BEN:

```
<pool>
   <type>ben.vlan</type>
   <label>BEN VLAN</label>
   <description>A VLAN over BEN</description>
   <units>100</units>
   <start>2010-01-30T00:00:00</start>
   <end>2011-01-30T00:00:00</end>
   <handler path="controllers/ben/gec7/ben.xml" />
   <properties>
      <property name="substrate.file" value="orca/network/ben-6509.rdf" />
   </properties>
</pool>
```

Vacating Leases

If ticket /lease closed before expiration

- Before 2.0:
  - broker unaware, cannot reuse resources
- In 2.0:
  - Orca informs the broker
  - Brokers can use freed resources immediately
    - No need to wait to lease expiration
  - Handled transparently by Orca

What's Next?

- No major new features in the next release
- Primary focus:
  - Harden and productize existing codebase
  - Simplify installation and setup
  - Documentation
- If time permits:
  - SOAP management API
    - Integrate Orca in command-line tools and other portals
  - Distributed Orca without Tomcat

Orca 2 .0 Availability

- Available immediately BUT:
  - Some critical bugs discovered after the release
  - Fixed in trunk not yet pushed to release branch
  - Documentation is being updated
- All issues will be addressed by end of month

How to Integrate a Substrate?

- Determine allocation & assignment policies:
  - Allocation: broker issuing a ticket
  - Assignment: substrate provisioning the resources
- Most likely policies you need already exist
  - If not, can easily extend and subclass
  - Extend unit test framework and verify policies work
- Decide how to provision a unit of resource
  - What API does your substrate expose?
    - CLI, SOAP, XMLRPC,…

How to Integrate a Substrate? (2)

- Write a handler/driver to provision a unit
  - Decide if you need a Node Agent–based driver
  - Write the driver/library and unit test
  - Write the handler
  - Test your handler in isolation
    - We provide tools to help
- Test end-to-end requests in emulation
- Test end-to-end requests in real mode