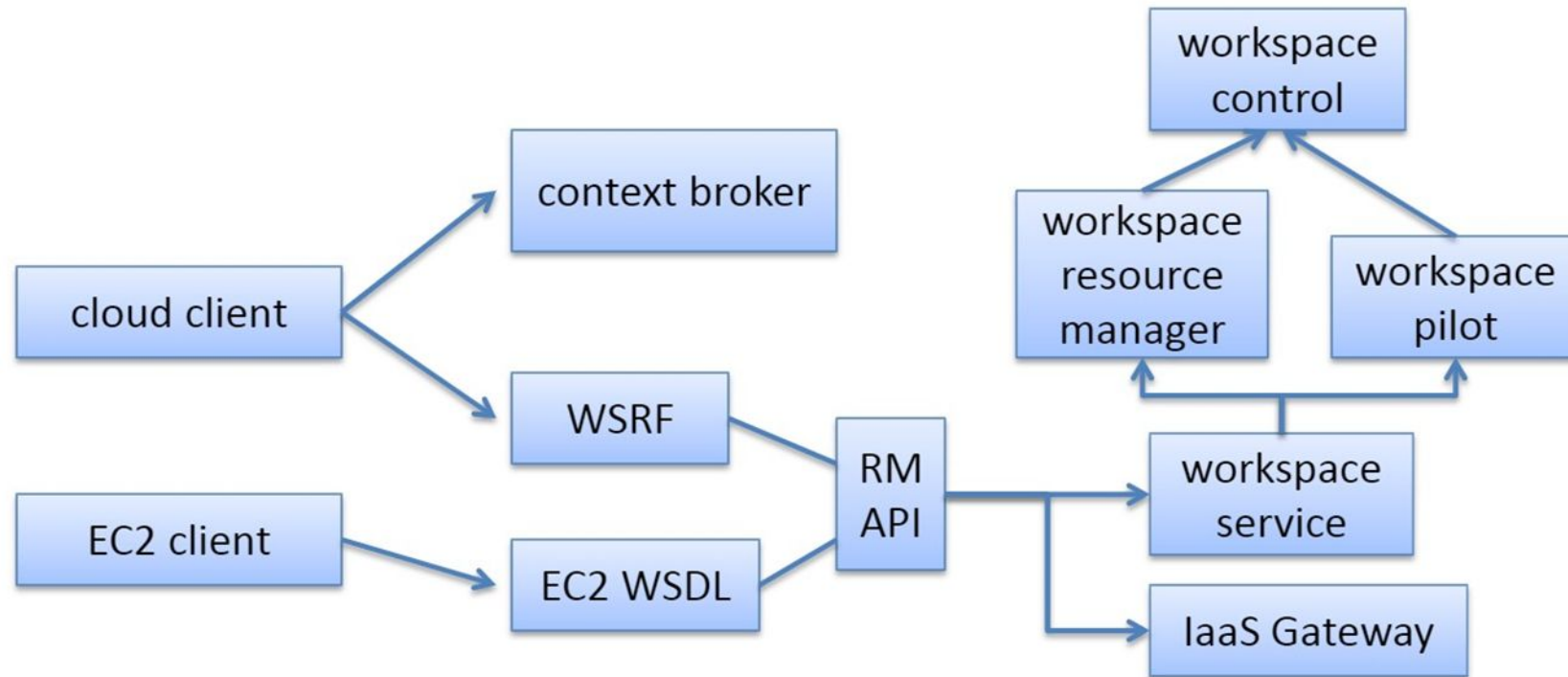


Nimbus & OpenNebula

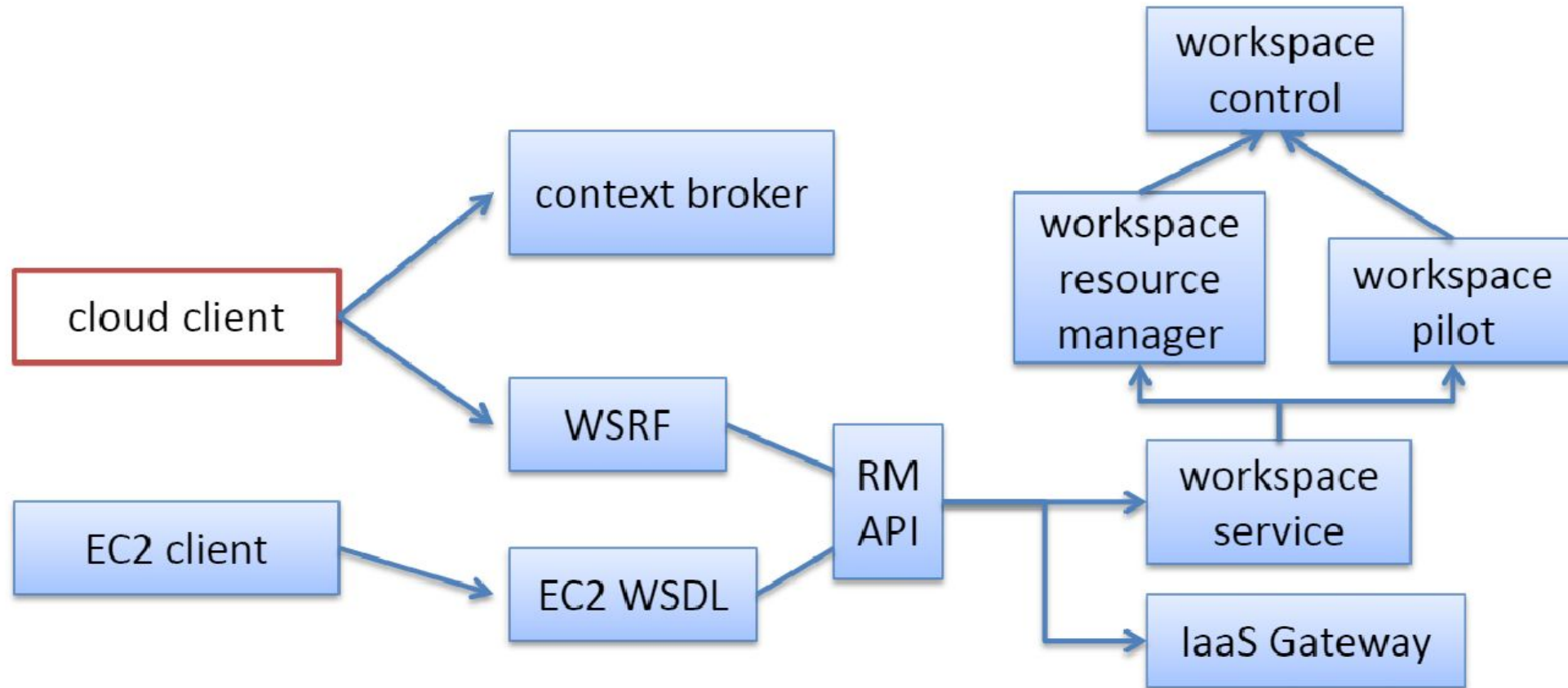
Nimbus - Intro

- Open source toolkit
- Provides virtual workspace service (Infrastructure as a Service)
- A client uses virtual machines (Xen) on remote resources.
- Virtual machines are configured as user demand
- Uses X.509 certificates

Nimbus - Architecture

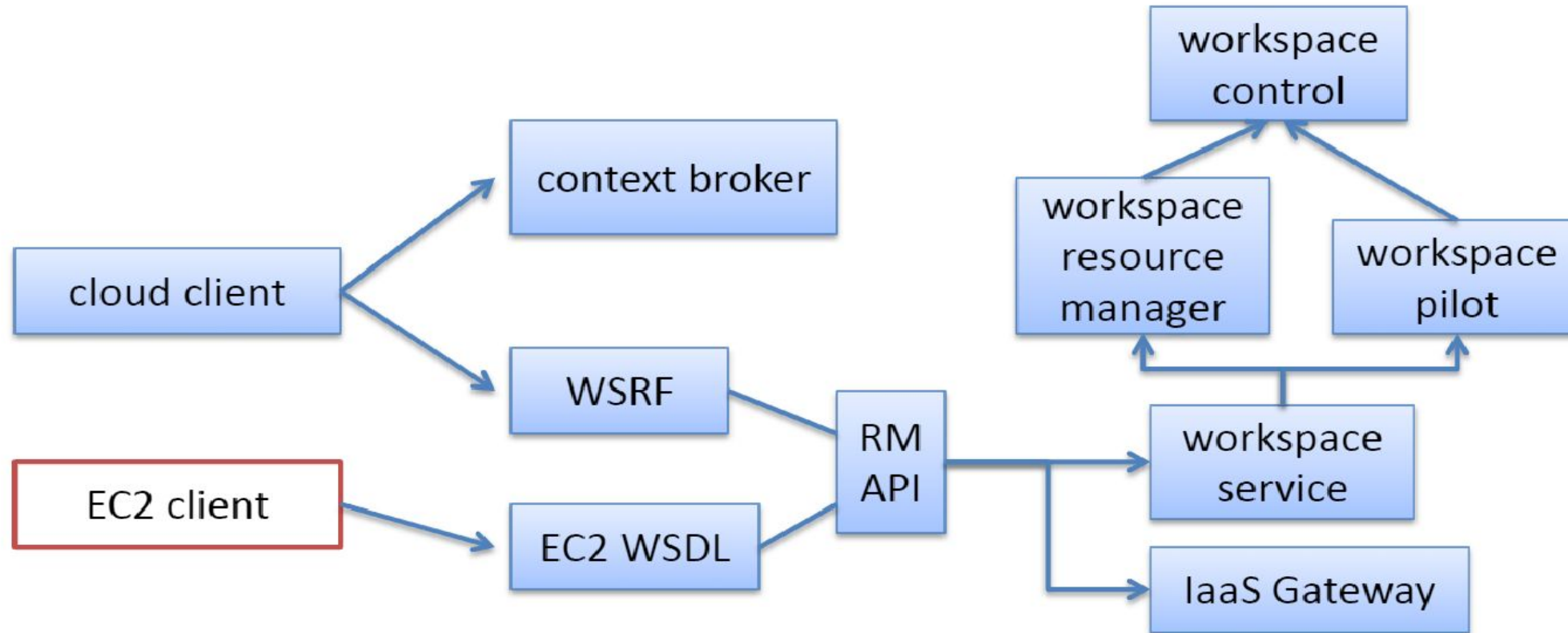


Nimbus - Architecture



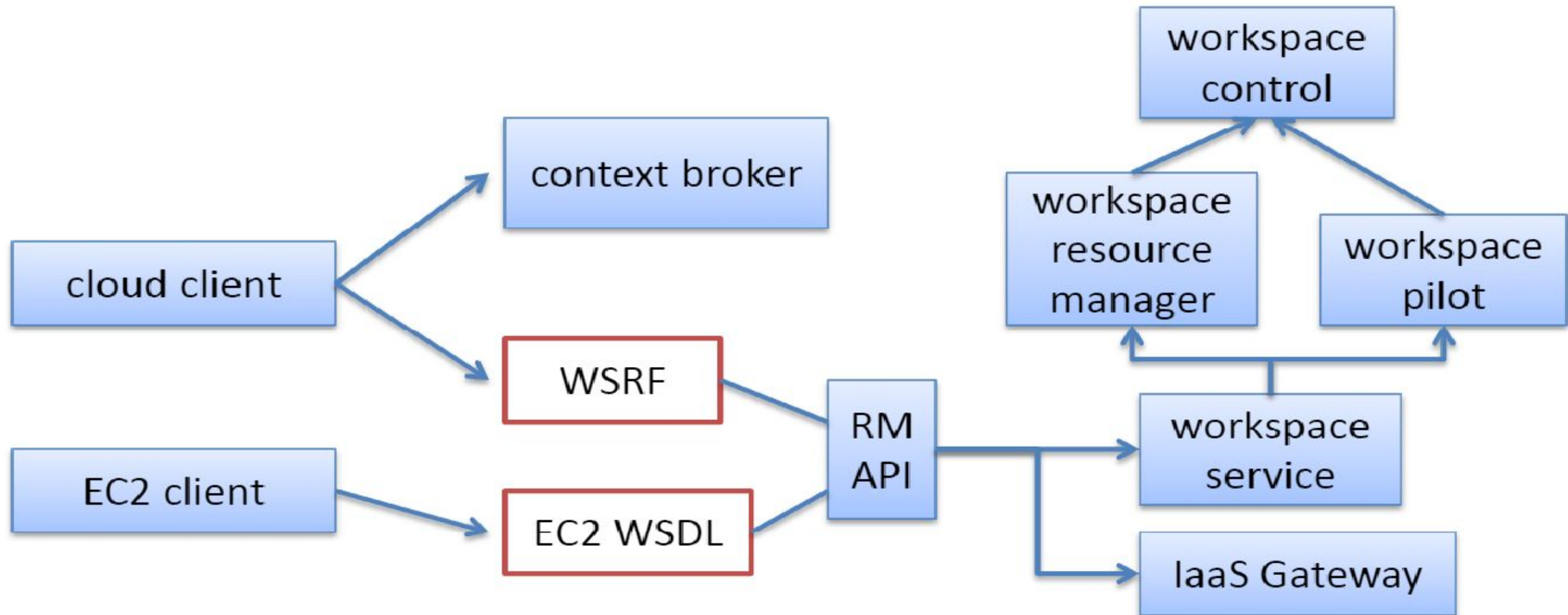
Cloud client: connects to Nimbus and use VMs

Nimbus - Architecture



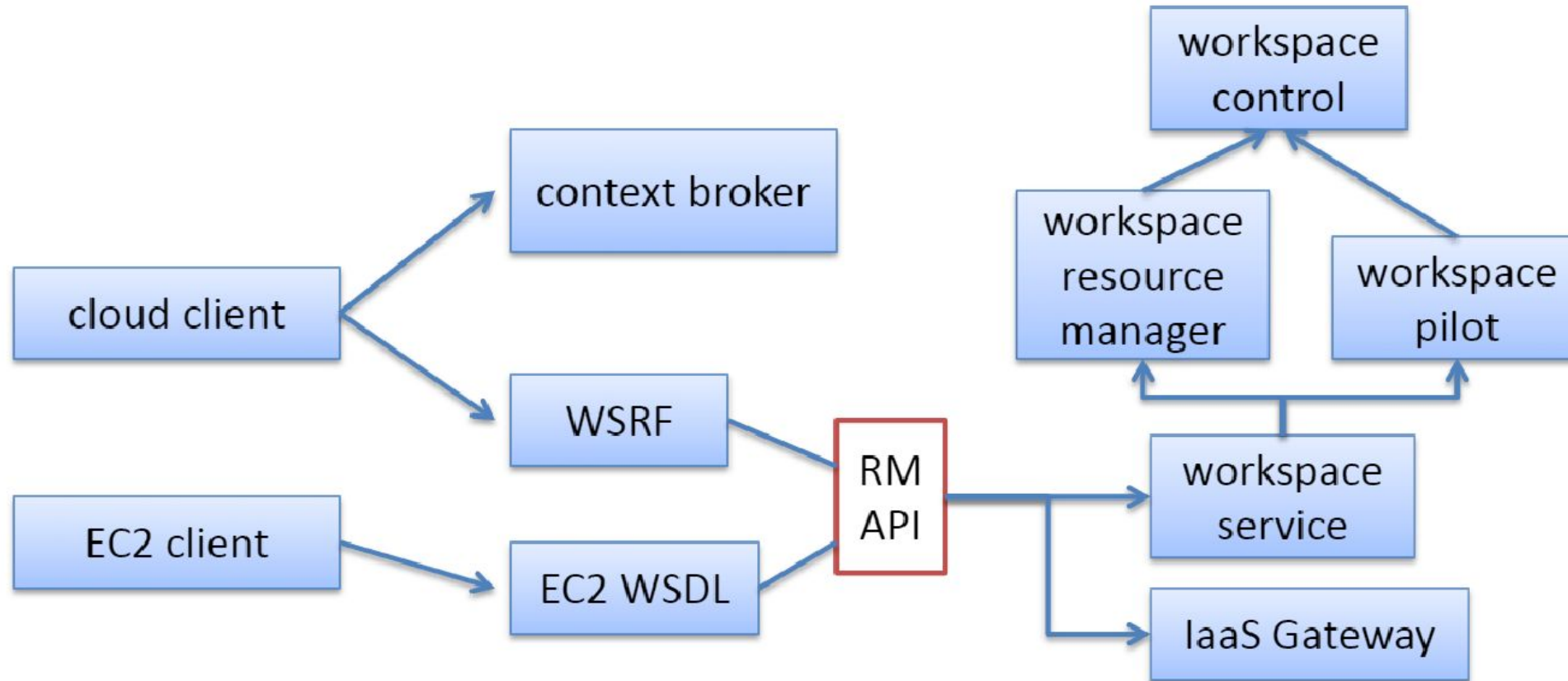
EC2 client: connects to the Amazon Elastic Compute Cloud (EC2)

Nimbus - Architecture



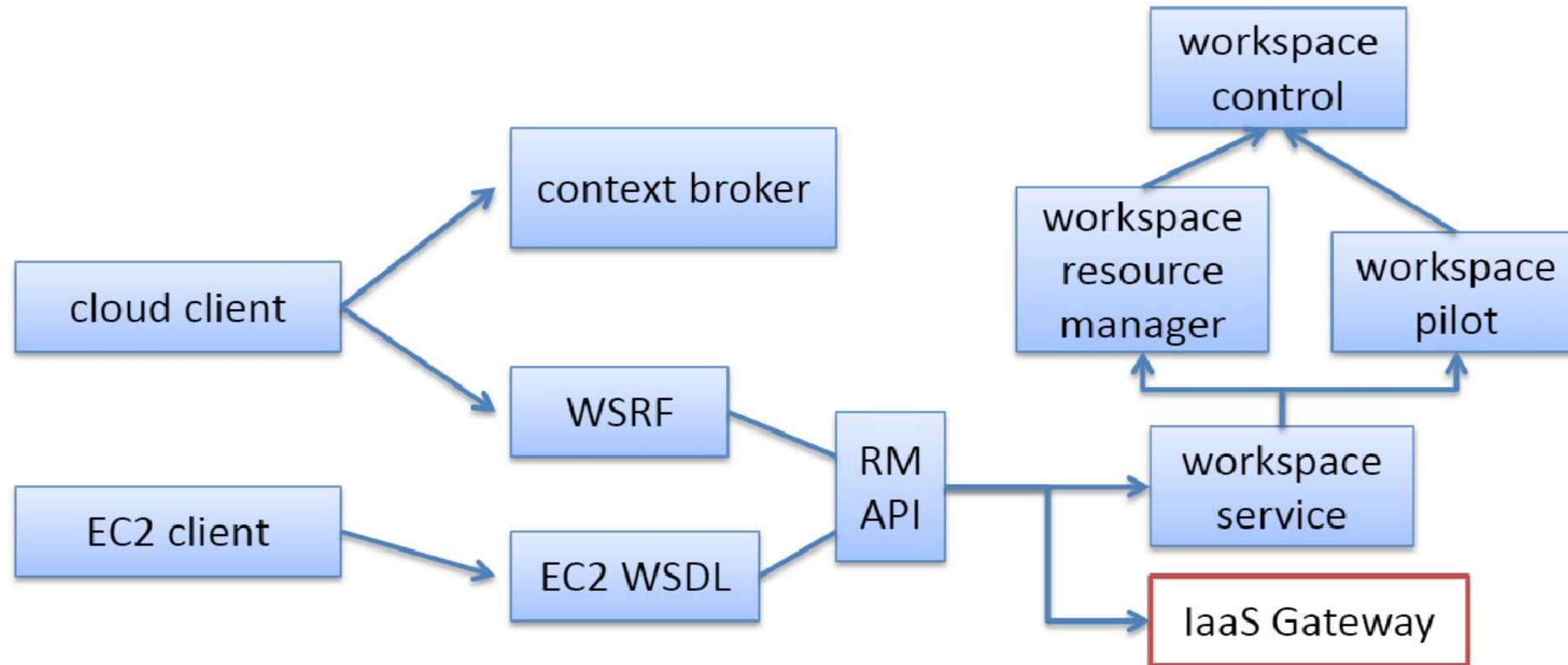
WSRF & EC2 WSDL: protocol implementations

Nimbus - Architecture



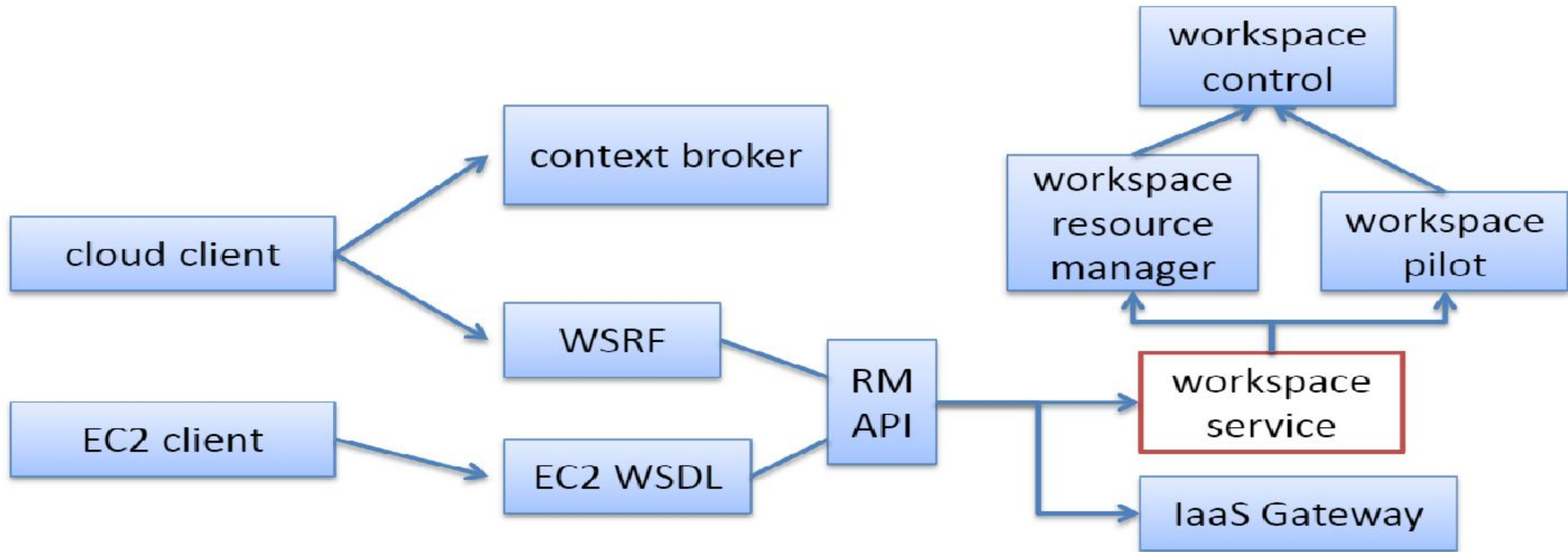
RM API: interface to Virtual Workspace Service

Nimbus - Architecture



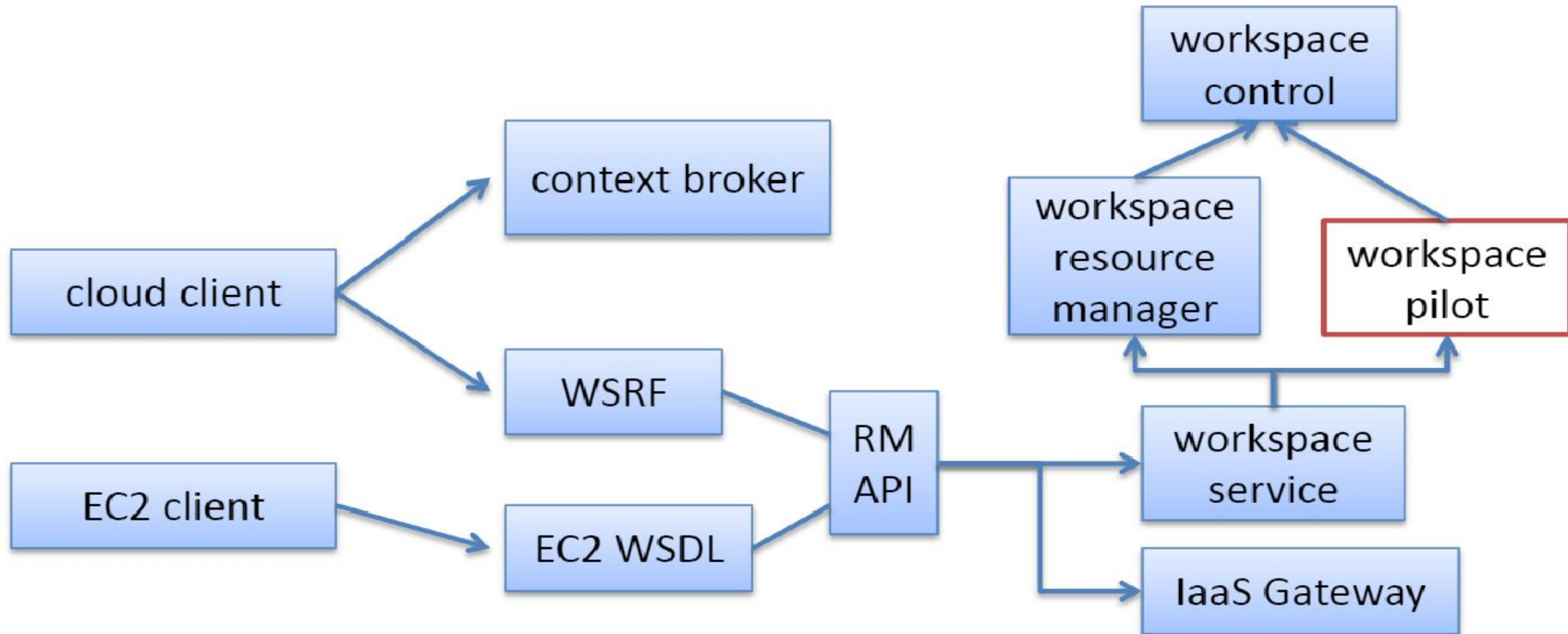
IaaS Gateway: connects to EC2 or other cloud services

Nimbus - Architecture



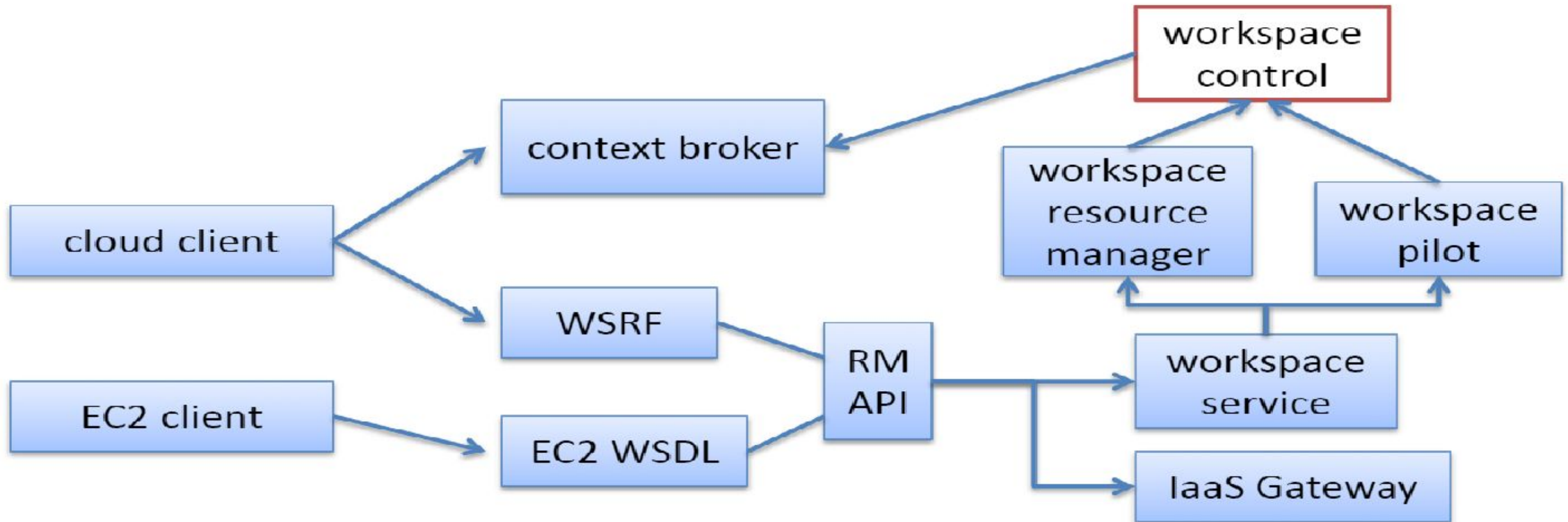
Workspace service: VM manager, Web Service base, run in GT Java container

Nimbus - Architecture



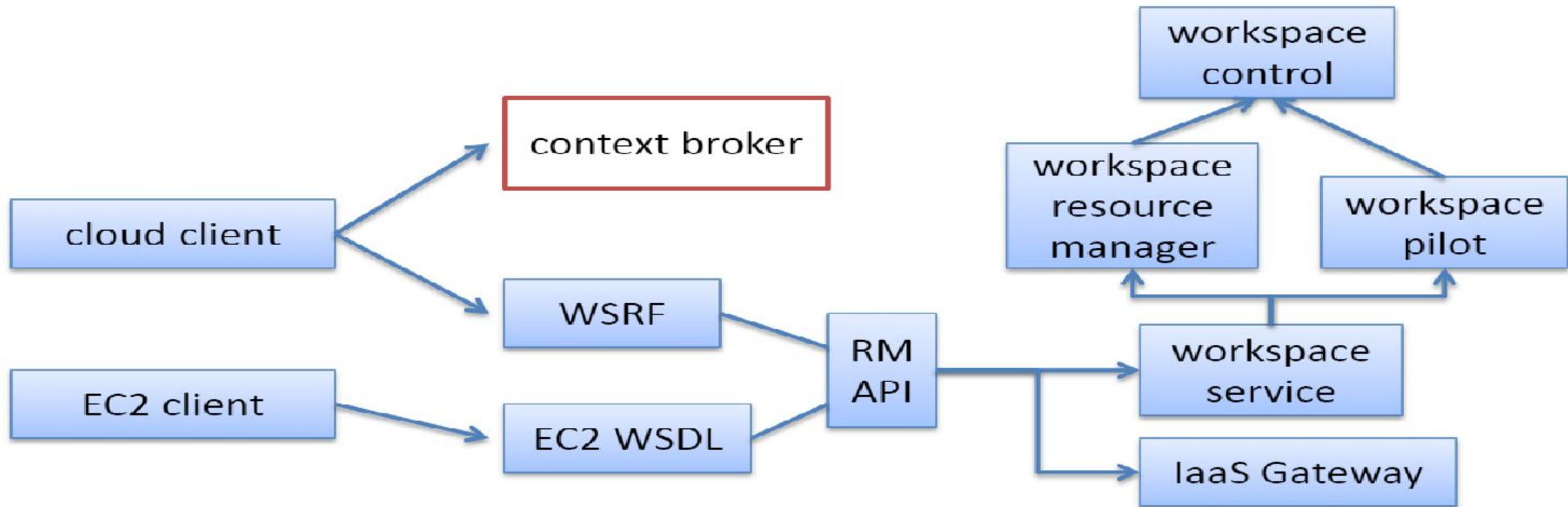
Workspace pilot: uses local site manager such as PBS

Nimbus - Architecture



Workspace control: run VMs, build VM image, connect VMs to network, send contextualization information to the context broker

Nimbus - Architecture



Context broker: coordinates virtual cluster at its deployment
Context agent: runs on each VM to communicate with the context broker

Nimbus – Administration - I

- Setting service container
- Creating new host / user certificates
- Relevant file configuration
- Installing Xen
- Setting DHCP (Dynamic Host Configuration Protocol)
- Each hypervisor needs DHCP server and ebtable

Nimbus Administration II

- Installing Nimbus service packages
 - RM API, workspace service (VM, VMM)
 - WSRF frontend
 - EC2 frontend
- Configuration
 - Service host name
 - VMM (Virtual Machine Manager) names
 - Networks (DNS, subnet, etc.)
 - Authorization

Nimbus Administration III

- Installing workspace-control program
- Configuration
 - sudoers file: set paths to dhcp-config.sh, mount-alter.sh, etc. for the users
 - DHCP (only for delivery): add subnet in dhcpd.conf
 - Kernels: copy kernel images and edit [image] section in worksp.conf file for clients to choose
 - Networks: edit [networking] section in worksp.conf file Example:
association_0: private; xenbr0; vif0.0

Nimbus Client Administration

- Download and install

- Create proxy credential: `$./bin/grid-proxy-init.sh`

- Select VM

- `$./bin/cloud-client.sh -list`

- Deploy VM

- `$./bin/cloud-client.sh --run --name hello-cloud --hours 1`

- Output:

SSH public keyfile contained tilde:

- `'~/ssh/id_rsa.pub' --> '/home/guest/.ssh/id_rsa.pub'`

Launching workspace.

Using workspace factory endpoint:

`https://cloudurl.edu:8443/wsrf/services/WorkspaceFactoryService` Creating workspace "vm-023"... done.

IP address: 123.123.123.123

Hostname: **ahostname.cloudurl.edu**

Start time: Fri Feb 29 09:36:39 CST 2008

Shutdown time: Fri Feb 29 10:36:39 CST 2008

Termination time: Fri Feb 29 10:46:39 CST 2008

Waiting for updates

.State changed: Running

Running: 'vm-023'

Nimbus Client Commands

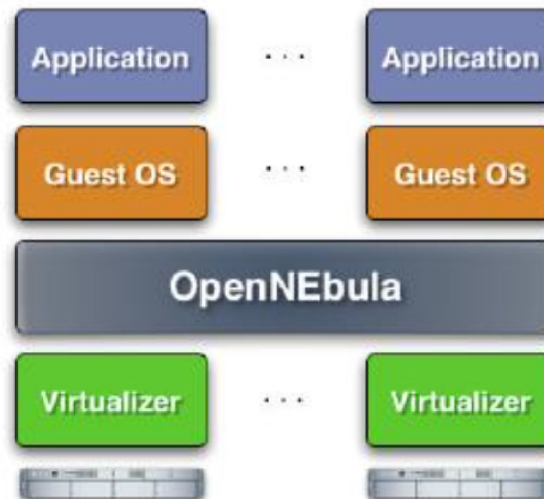
- Log on
 - `$ ssh root@ahostname.cloudurl.edu`
- Get info about VM
 - `$./bin/cloud-client.sh --status --handle vm-023`
- Save changes
 - `$./bin/cloud-client.sh --save --handle vm-023 --newname custom-1`
- Terminate VM
 - `$./bin/cloud-client.sh --terminate --handle vm-023`

Nimbus API Java

- Request authorization: creation process
 - `org.globus.workspace.service.binding.GlobalPolicies`
 - `org.globus.workspace.service.binding.authorization.CreationAuthorizationCal
lout`
- Initial request intake: handle creation requests
 - `org.globus.workspace.creation.Creation`
 - `org.globus.workspace.service.binding.Bind*`

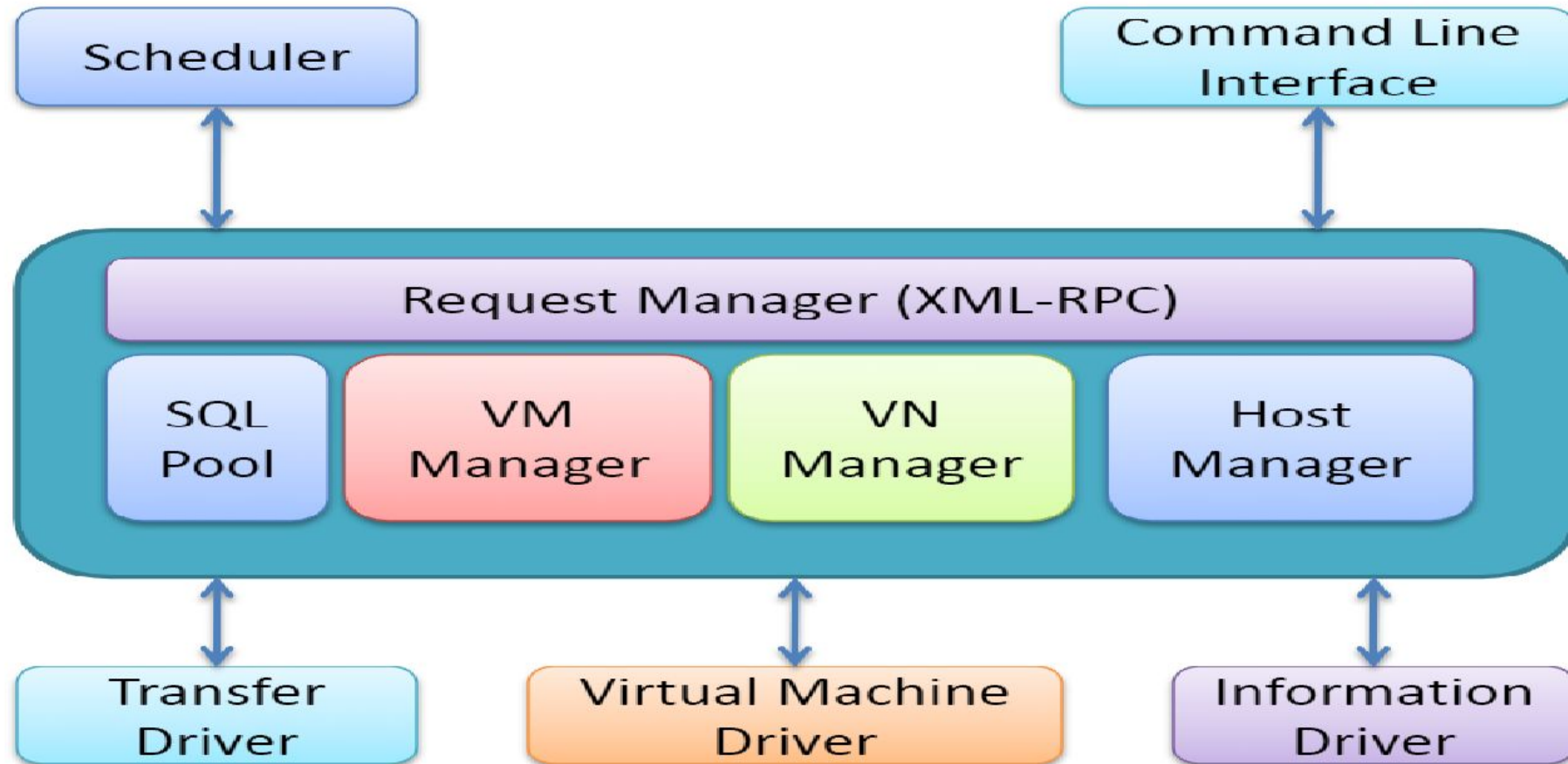
OpenNebula Intro

- Open source virtual infrastructure engine
- Deploy VMs
- A distributed virtualization layer



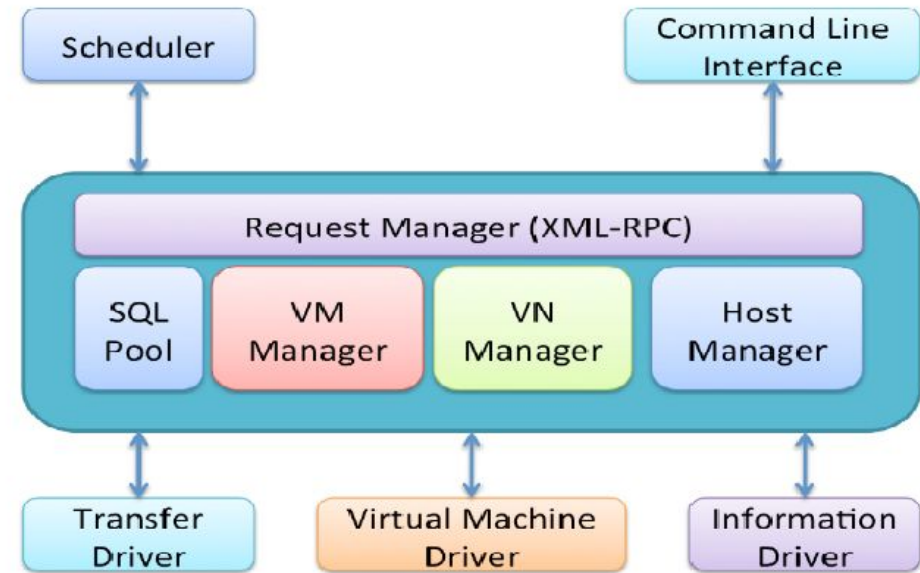
OpenNebula: The Open Source Virtual Machine Manager for Cluster Computing,
<http://www.opennebula.org/lib/exe/fetch.php?id=outreach&cache=cache&media=opennebula-oakland.pdf>
f

OpenNebula Architecture



OpenNebula - Architecture

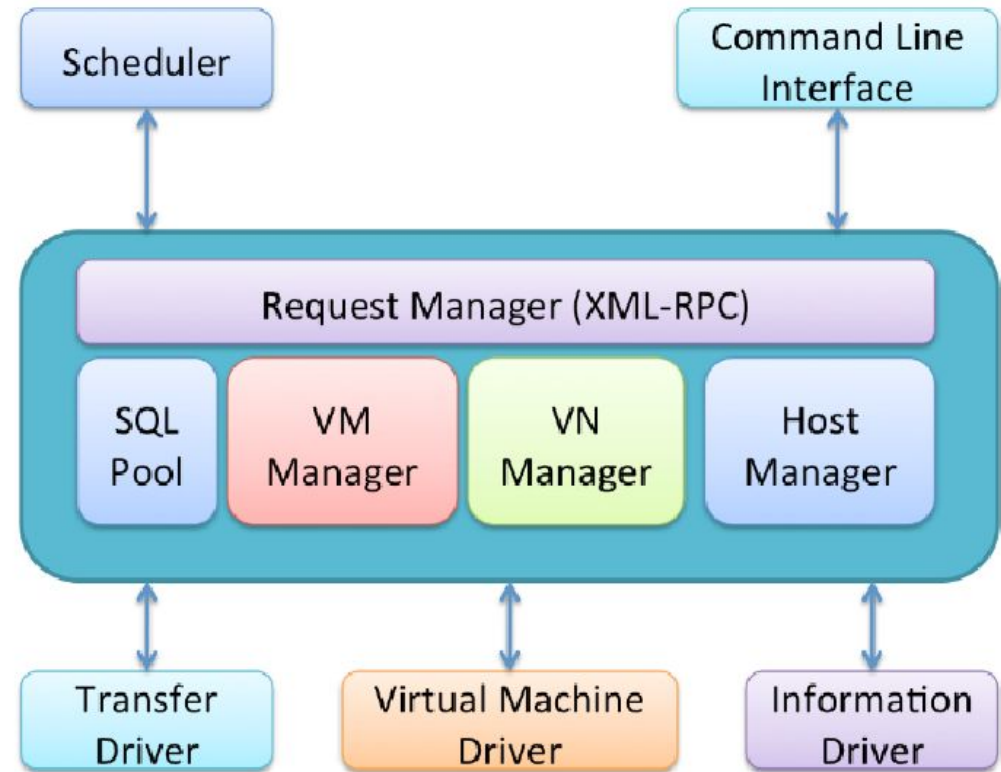
- Scheduler: deploy VMs in physical hosts
- Request Manager: implements a XML-RPC interface
- SQL Pool: maintains status of VMs, hosts, virtual networks.
- VM (Virtual Machine) Manager: manages VM life cycles
- VN (Virtual Network) Manager: generates MAC / IP addresses



- Host Manager: maintains information about hosts and communicate with them

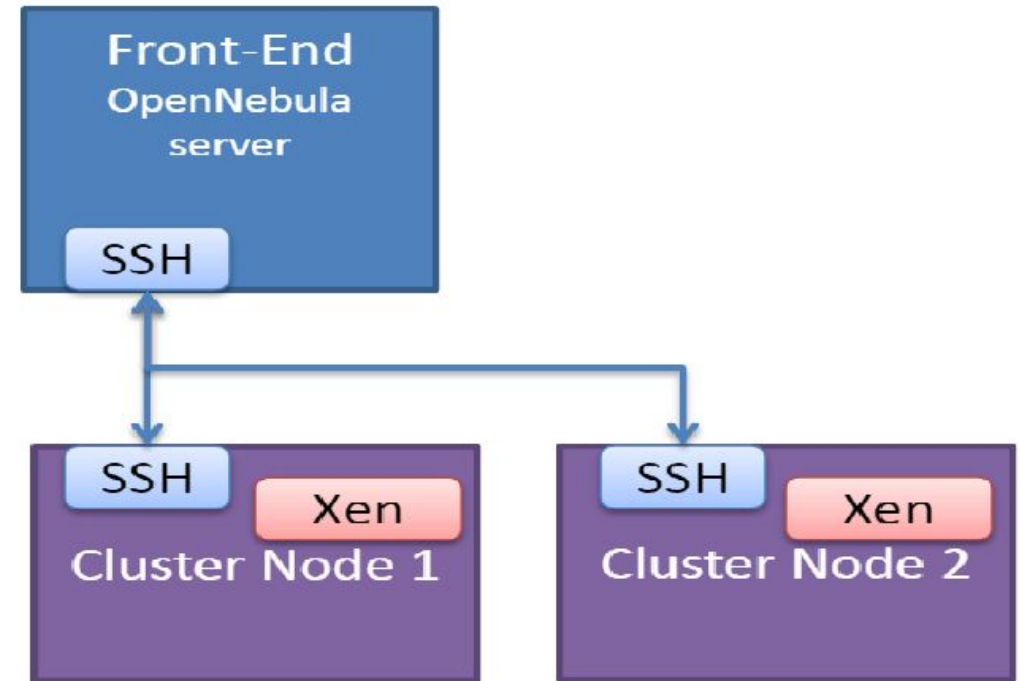
OpenNebula - Architecture

- Transfer Driver: interacts with the image operations (cloning, deleting, ...)
- Virtual Machine Driver: interacts with the VM life cycle operations (deploy, shutdown, ...)
- Information Driver: gets information about physical hosts (memory, cpu, ...)



OpenNebula - Architecture

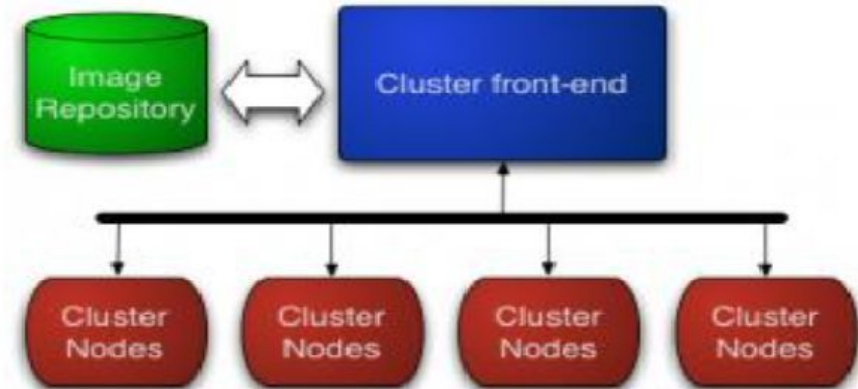
- Set up users & groups by NIS (Network Information Service)
 - `server$ groupadd xen`
 - `server$ useradd -G xen oneadmin`
 - `server$ cd /var/yp`
 - `server$ make`
- Create local groups
 - `node1$ echo "rootxen:x:<xen_gid>:root >> /etc/group`
 - `node2$ echo "rootxen:x:<xen_gid>:root >> /etc/group`
- Configure SSH
 - `server$ ssh-keygen`
 - `server$ scp id_rsa.pub node1:`
 - `node1$ cd ~/.ssh`
 - `node1$ cat id_rsa.pub >> authorized_keys`



OpenNebula Admin: Storage

- Image repository (separate or in the front-end)

- Contains VM images



<http://www.opennebula.org/doku.php?id=documentation:rel1.2:ignc>

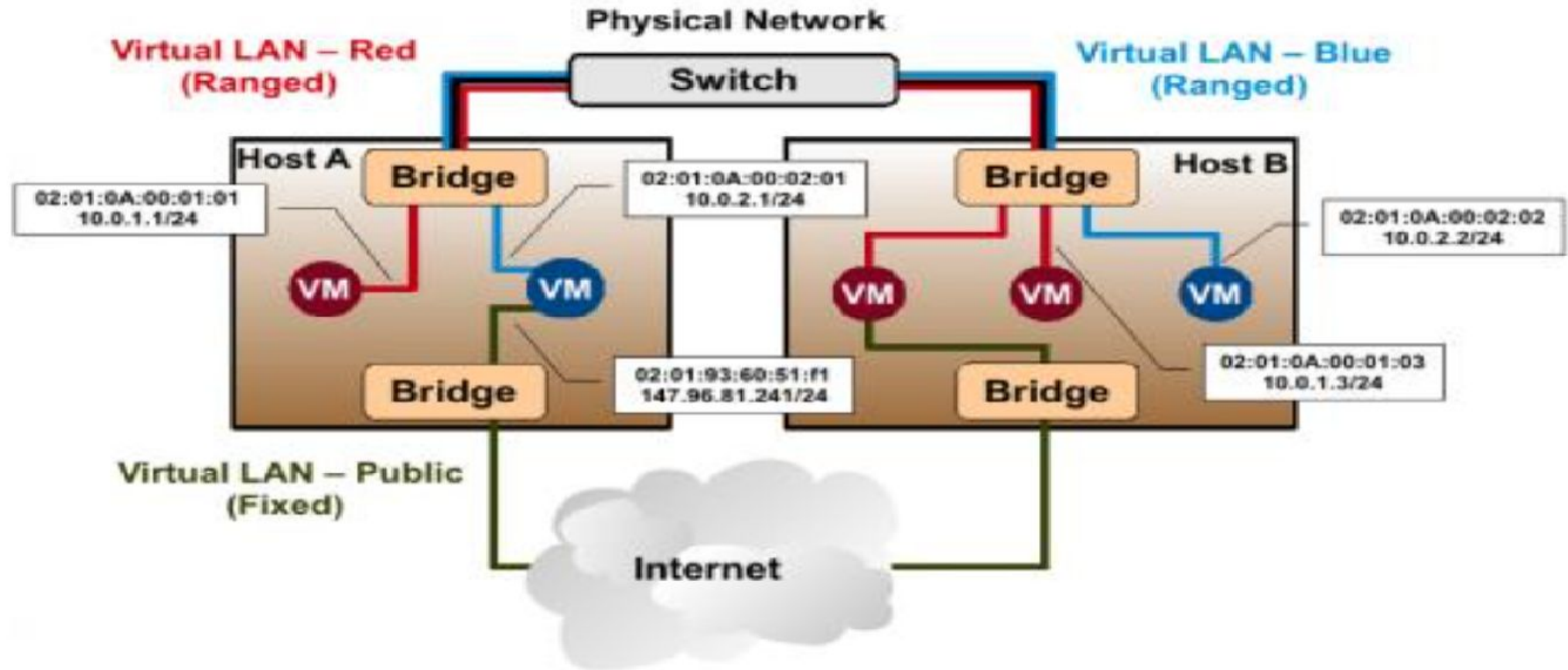
- Image life-cycle

- Preparation: images are in the repository
 - Cloning: copy an image from the repository to VM directory
 - Save / remove: save and dispose the image

- Physical cluster configuration

- `$ONE_LOCATION/etc/oned.conf`
 - `$ONE_LOCATION/etc/tm_nfs/tm_nfs.conf`

OpenNebula – Admin: Networks



2 VM clusters have virtual networks and access to the Internet

OpenNebula – Admin: Networks

- Define a virtual network
 - oned.conf
 - Fixed virtual networks: set specific addresses
 - Ranged virtual networks: set a base address
- Create a virtual network
 - `$ onevnet -v create private_red.net`

OpenNebula Administrator

Install OpenNebula

- `$./install.sh -d /opt/nebula/ONE`

Set up cluster

- `$ onehost create node1 im_xenvmm_xentm_ssh`
- `$ onehost create node2 im_xenvmm_xentm_ssh`
im_xen: reference to information driver
vmm_ssh: reference to virtual machine driver
tm_ssh: reference to transfer driver

Start OpenNebula

- `$ ONE_LOCATION/bin/one start`

OpenNebula – Command Line Interface

- Three commands
 - onevm [<options>] <command> [<parameters>]
 - onehost <command> [<parameters>]
 - onevnet <command> [<parameters>]
- Commands of onevm
 - create, deploy, shutdown, livemigrate, migrate, hold, release, stop, suspend, resume, delete, list, show, top, history
- Commands of onehost
 - create, show, delete, list, enable, disable, top
- Commands of onevnet
 - create, show, delete, list

OpenNebula – Client APIs

- XMLrpc methods
- VM allocation: `one.vmallocate`
- VM deployment: `one.vmdeploy`
- VM actions: `one.vmaction`
 - shutdown, hold, release, stop, suspend, resume
- VM migration: `one.vmmigrate`
- VM information: `one.vmget_info`

OpenNebula – Java Example

- Invoking one.vmallocate

```
import org.apache.xmlrpc.XmlRpcClient;
import java.util.Vector;

public class XmlRpcTest {
    public static void main( String args[] ) throws Exception {
        XmlRpcClient client = new XmlRpcClient( "http://localhost:2633/RPC2" );
        Vector params = new Vector();
        params.addElement("SESSION-GOLA&4H910");
        params.addElement("MEMORY=345 CPU=4
DISK=[FILE=\"img\",TYPE=cd]""DISK=[FILE=\"../f\"]");

        Object result = client.execute( "one.vmallocate", params );

        if ( result != null )
            System.out.println( result.toString() );
    }
}
```

http://www.opennebula.org/doku.php?id=documentation:rel1.2:api_examples

References

- <http://workspace.globus.org/>
- <http://www.opennebula.org/doku.php>
- Open Source Grid and Cluster Software Conference - May 12th-16th 2008 - San Francisco, USA, OpenNebula: The Open Source Virtual Machine Manager for Cluster Computing,
<http://www.opennebula.org/lib/exe/fetch.php?id=outreach&cache=cache&media=opennebula-oakland.pdf>
- Tutorial about Elastic Management of a Grid Computing Service with OpenNebula and Amazon EC2, March 2th, 2009,
http://www.opennebula.org/lib/exe/fetch.php?id=outreach&cache=cache&media=tutorial_ogf25.pdf
<http://grid.rit.edu>