RedIRIS Foro de Seguridad, March 9-10 2011

Valencia, España

# Security Management in OpenNebula Cloud Architectures

**Javier Fontán**
**jfontan@opennebula.org**

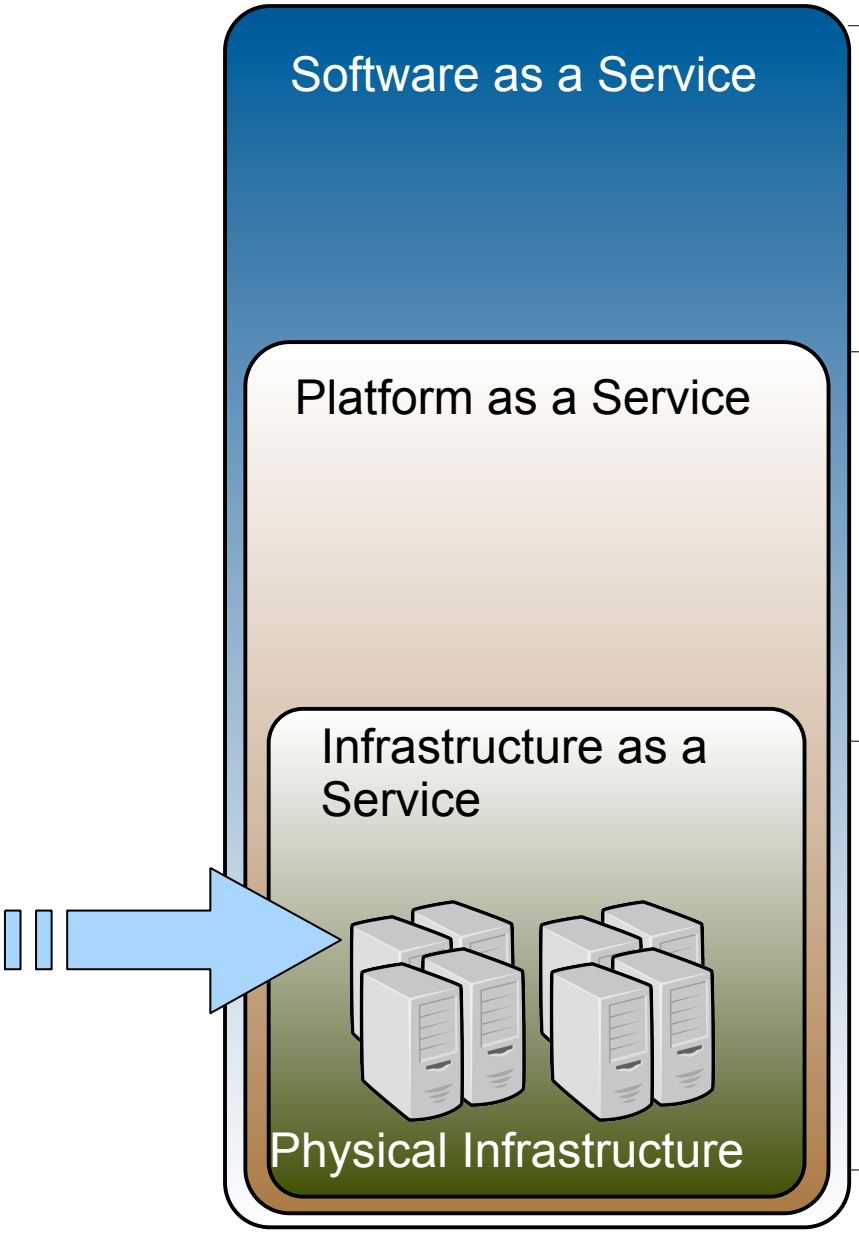
OpenNebula.org

StratusLab

SEVENTH FRAMEWORK PROGRAMME

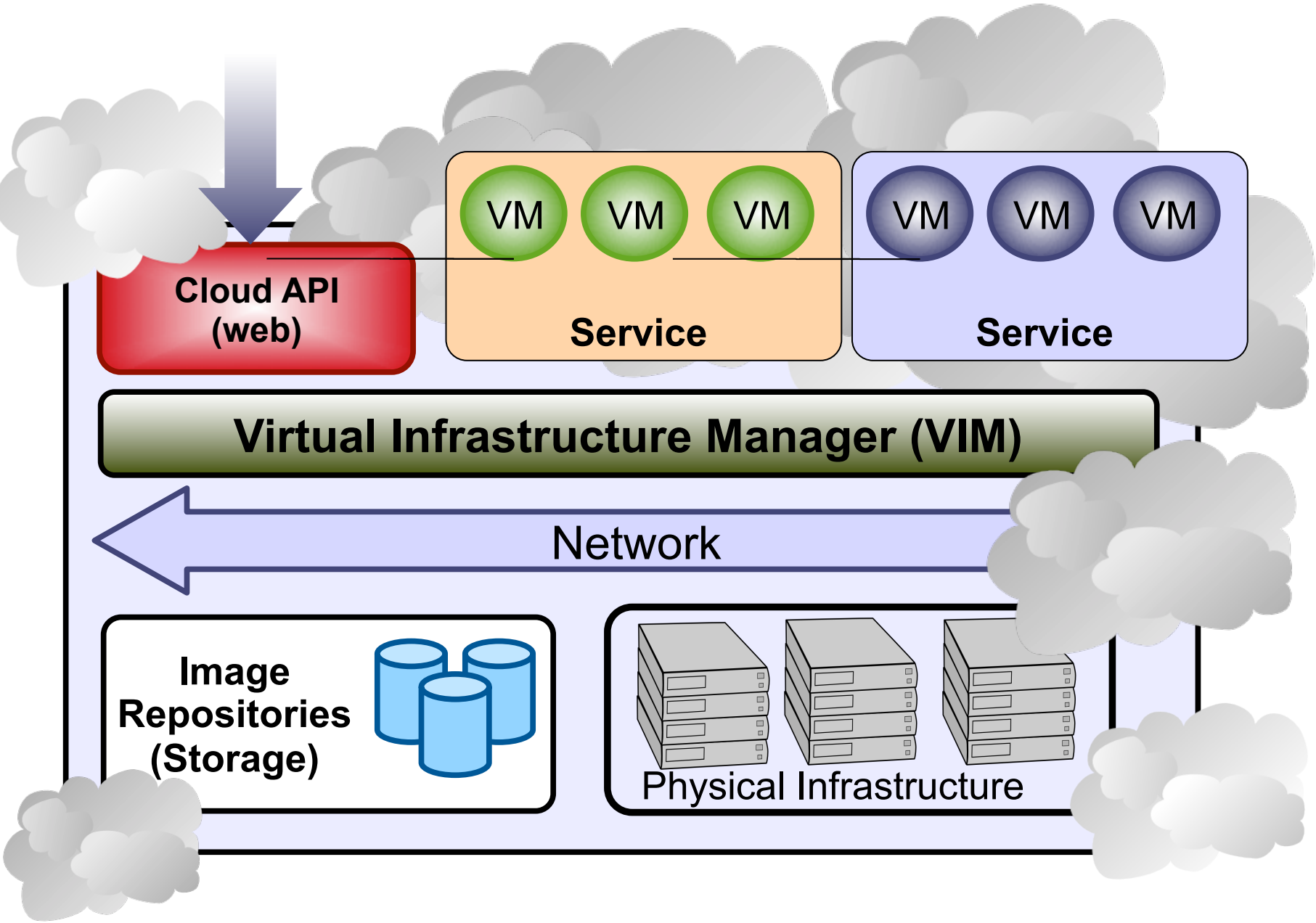
European Commission Information Society and Media

# Cloud Computing in a Nutshell



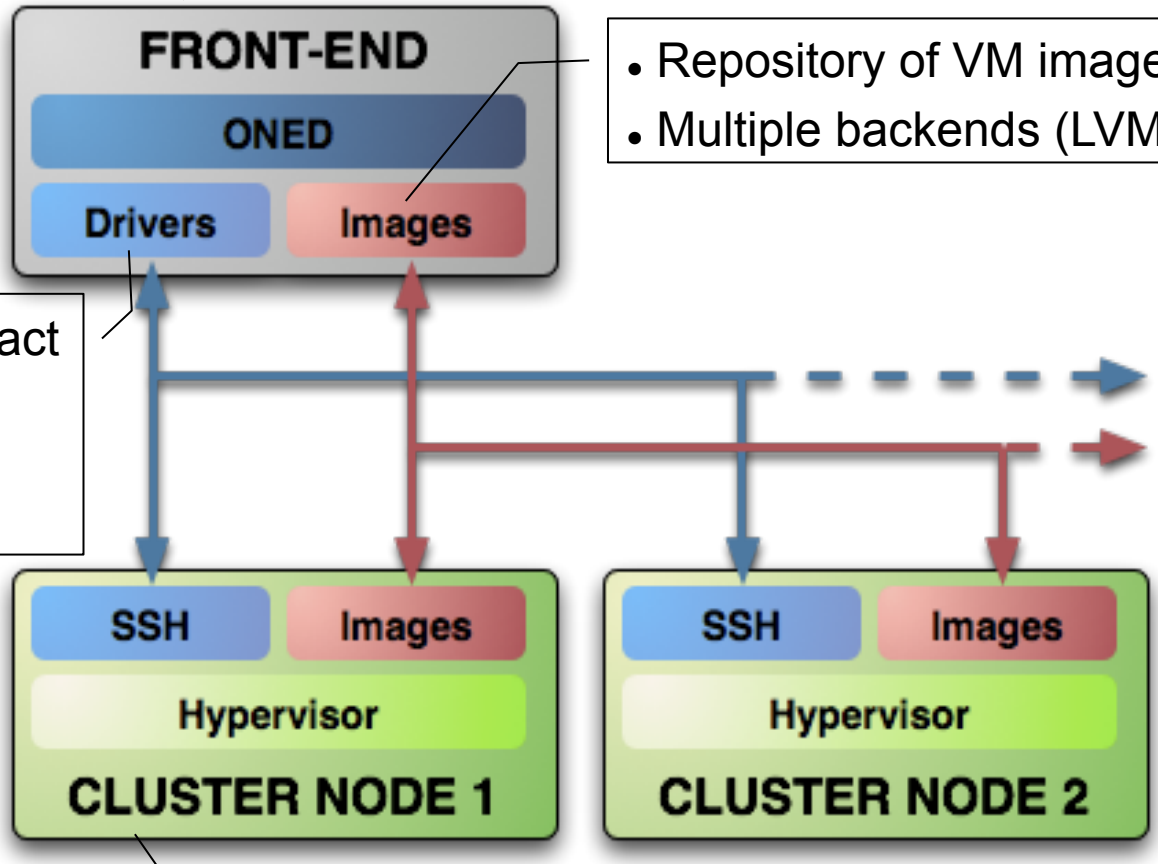

| What | Who |
|---|---|
| On-demand access to any application | **End-user** (does not care about hw or sw) |
| Platform for building and delivering web applications | **Developer** (no managing of the underlying hw & swlayers) |
| Delivery of a *raw* computer infrastructure | **System Administrator** (complete management of the computer infrastructure) |

# The Anatomy of an IaaS Cloud



# OpenNebula: System Overview

- Executes the OpenNebula Services
- *Usually* acts as a classical cluster front-end



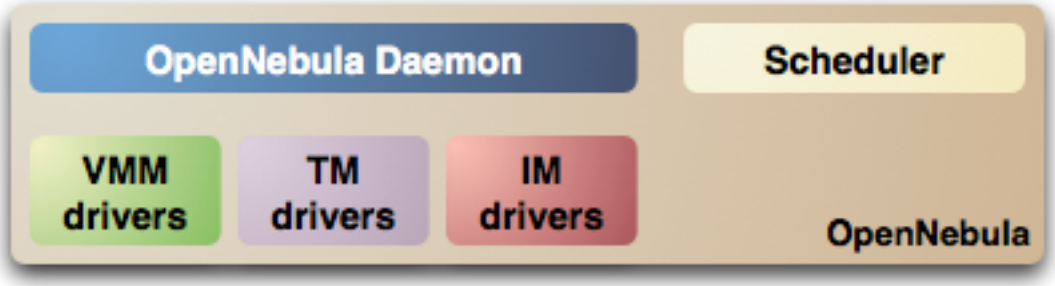

- Repository of VM images
- Multiple backends (LVM, iSCSI..)

- Modular components to interact with the cluster services
- *Types:* storage, monitoring, virtualization and network

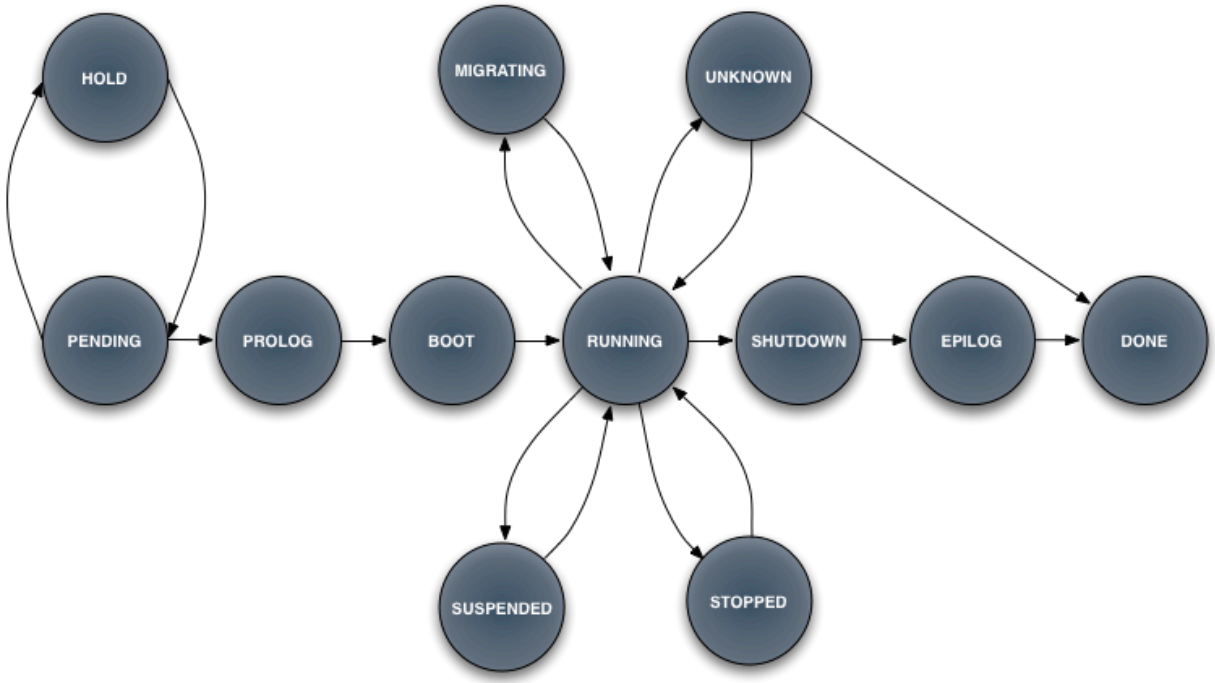
- Provides physical resources to VMs
- *Must have* a hypervisor installed

# OpenNebula: Pluggable Architecture

- Decoupled processes



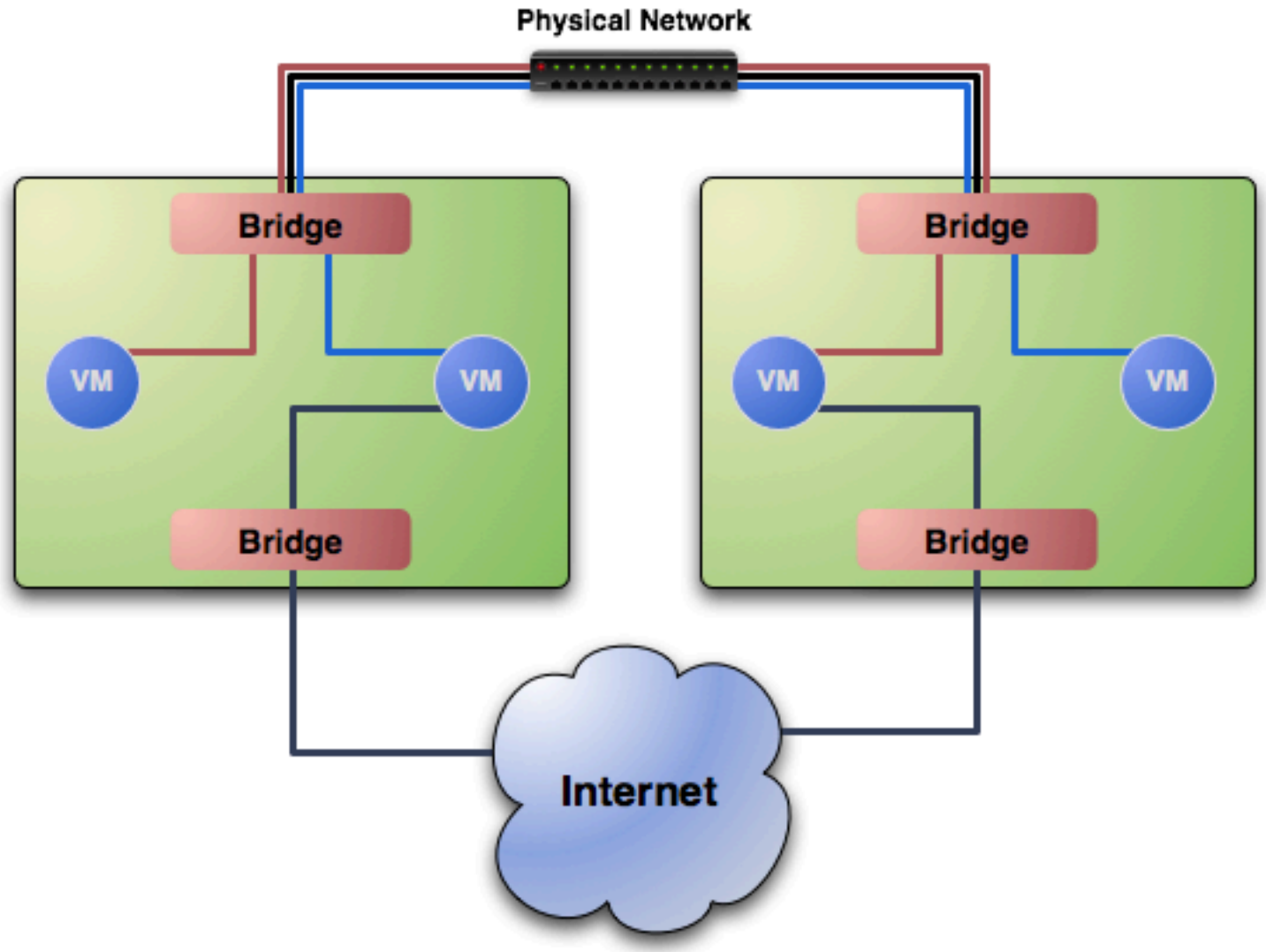

- State changes can trigger actions (Hooks)



# Application Programming Interfaces / Auth

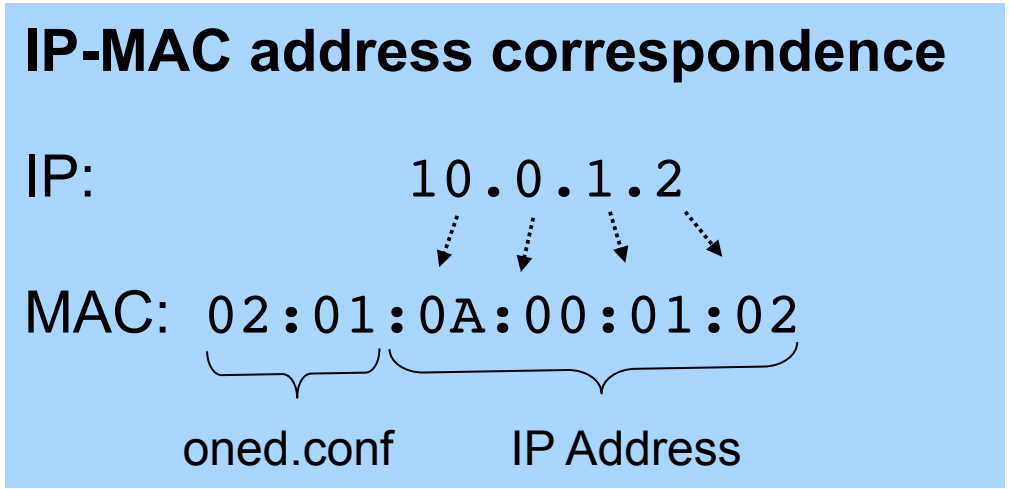
- Native API is XML-RPC (http, can be hardened with proxies)
- Conversion layers to interact with other API
  - EC2
  - OCCI
- Native Authentication is User/Password
- Authentication methods can be added using AUTH plugins
  - SSH (RSA key)
  - LDAP
  - X509
- AUTH plugins can also implement authorization
  - User Quotas
  - Tweaking permissions

# Network Anatomy



# Virtual Network in OpenNebula

- Manages Virtual Network
  - IP and MAC addresses are generated in a given range
  - A network is connected to an specific bridge
  - Networks have an owner and can also be public
  - MAC generation is directly related to the IP



# Virtual Network Isolation

- A Hook can be started on execution host on VM start
- A script is provided that creates a set of ebtables rules
  - Can only access machines from the same Virtual Network

```
ebtables -A FORWARD -s !00:02:0A:00:01:00/ff:ff:ff:ff:ff:00
    -o vif1.1 -j DROP
```

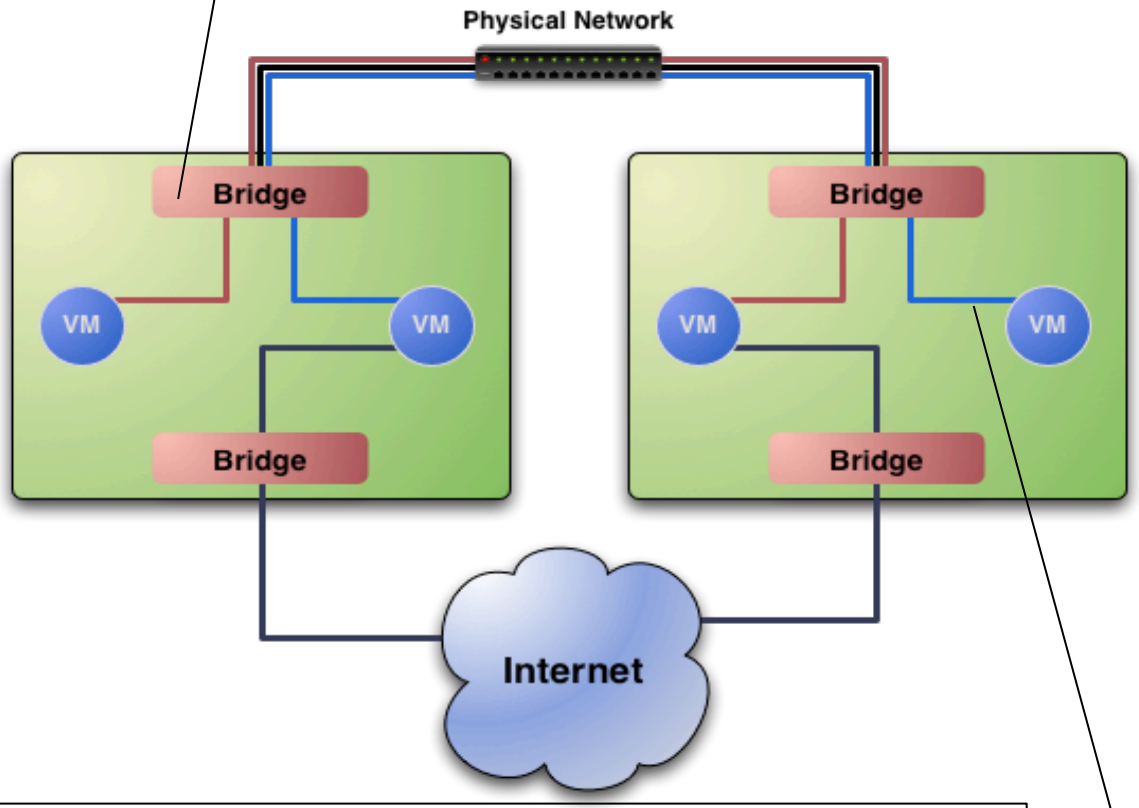
  - Cannot send with a MAC address other than the one given to the interface (no mac spoofing)

```
ebtables -A FORWARD -s !00:02:0A:00:01:08 -o vif1.1 -j
    DROP
```

- The same technique can be used to set `iptables` rules (e.g. open only a set of ports like EC2)

# Virtual Network Isolation

- IN: Only Ethernet frames from a MAC in Red LAN
- OUT: Only Ethernet frames from the MAC assigned by OpenNebula




- Networks are isolated at layer 2
- You can put any TCP/IP service as part of the VMs (e.g. DHCP, nagios...)

# Fair Share / QoS

- VMs can be resource greedy
- We set up XEN scheduler credits so a VM does not consume full CPUs (can be done with KVM also using hooks and `nice`)
- VMs can also use huge amounts of bandwidth. Traffic shaping can be applied using hooks as in the `ebtables` example
- Disk IO can also be controled using hooks and an tools like ionice.