From Bare Metal to Cloud

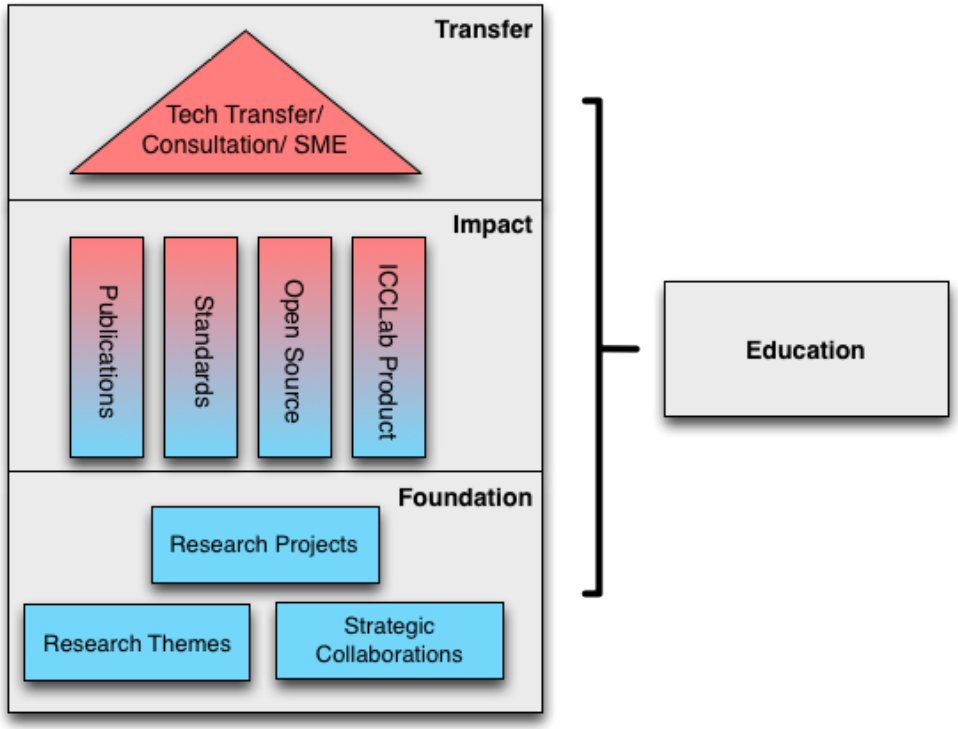
Andy Edmonds, Fabrice Manhart, Thomas Michael Bohnert, Christof Marti
ICCLab, ZHAW

www.cloudcomp.ch

Hello!

ICCLab

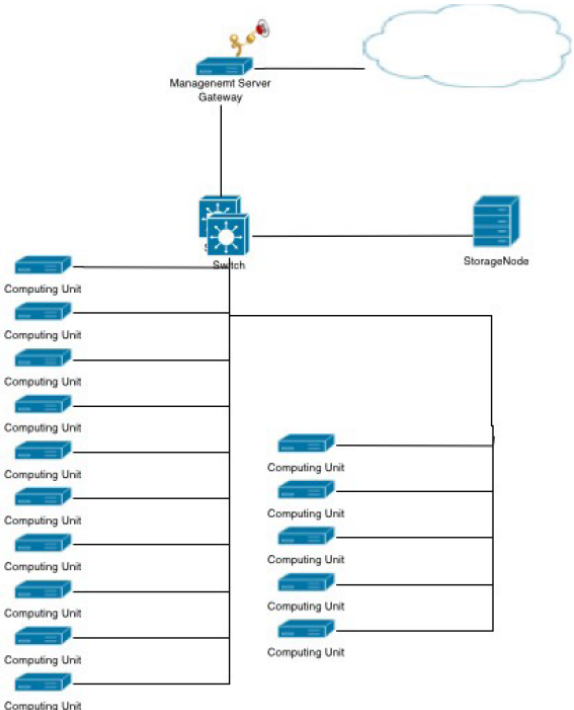
- Zurich University for Applied Sciences
- Cloud Computing Research



"From **Research Themes** to **Projects** creating **Impact** and **Transfer** for Educational Excellence"

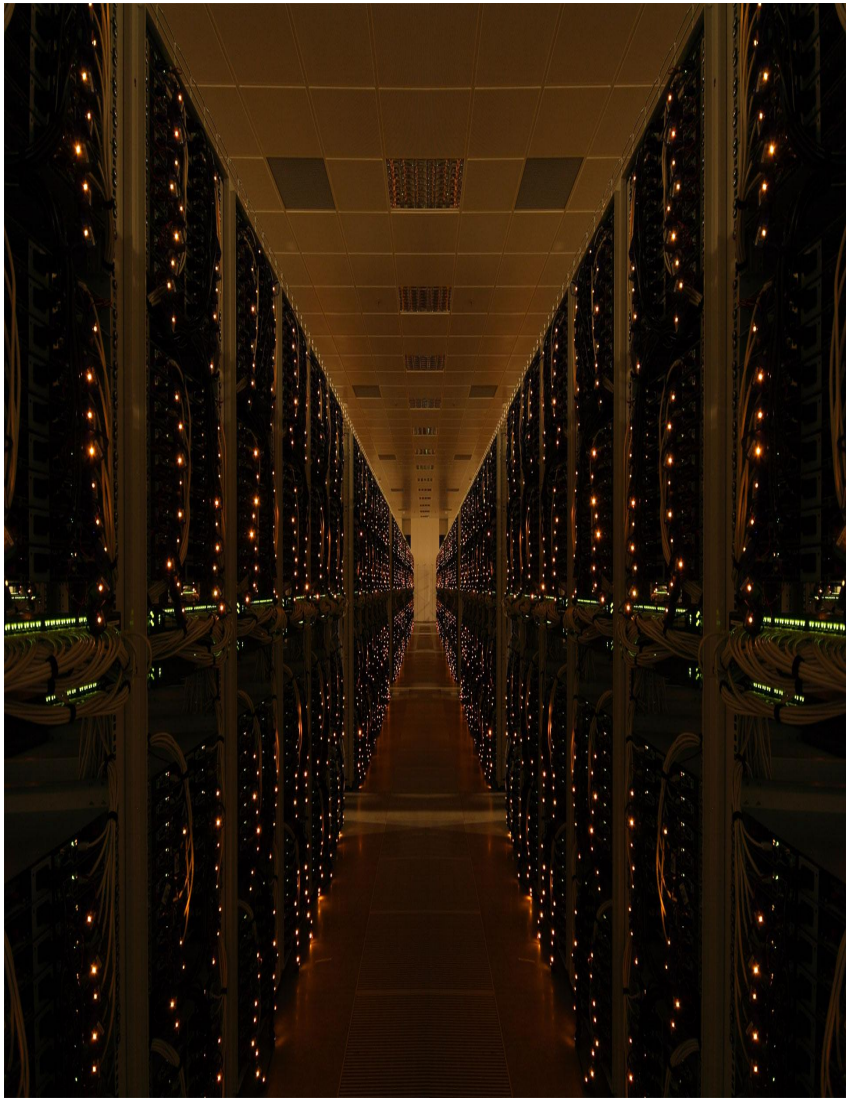
We've Hardware for Cloud!

Nodes	15
Core	240
Memory	960 GB
Storage	64 TB



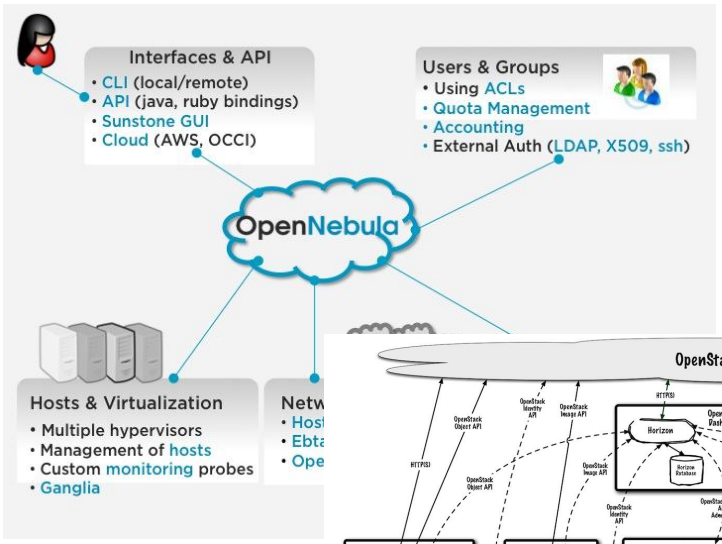
Challenges or Problems?

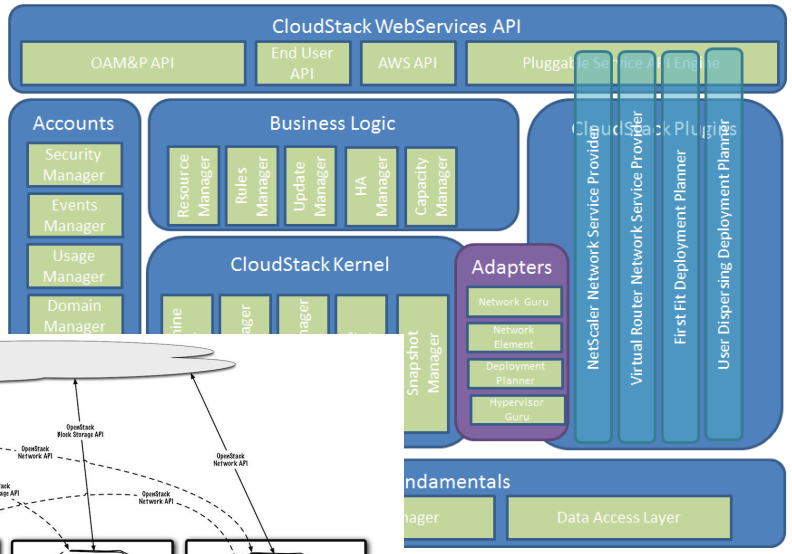
- Clouds are in essence big data centres
    - Means **lots** of servers:
        - Manual configuration **not an option**
        - Automation is **required**

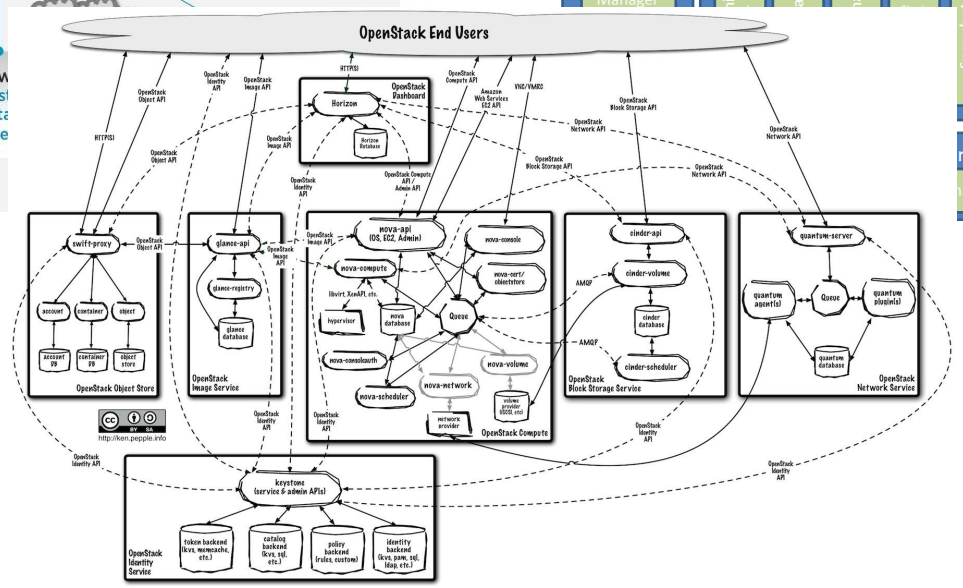


Challenges or Problems?

Cloud frameworks **can/are be complicated**!







Challenges or Problems?

- Clouds are "cool" - Aayyy!

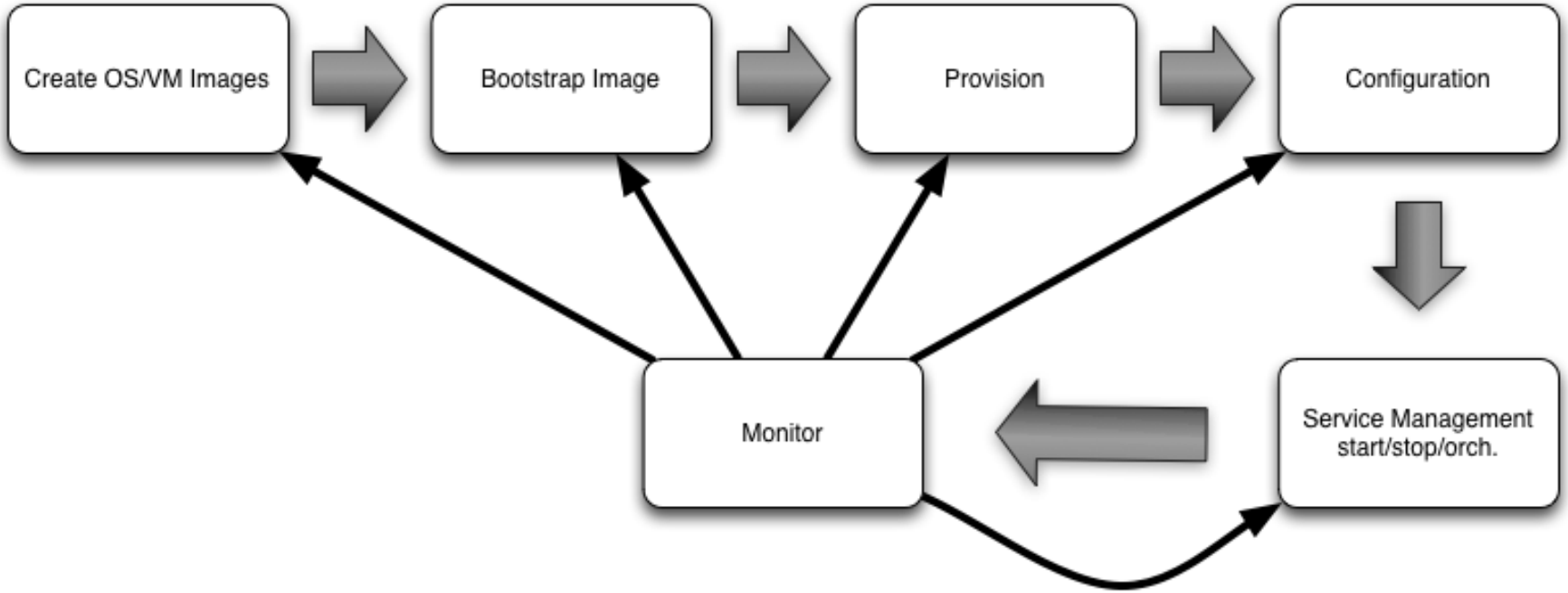
BUT

- How to deploy a "cloud"
  - with **minimal user interaction**?
  - **least number** of "hands"?
  - across **many servers**?

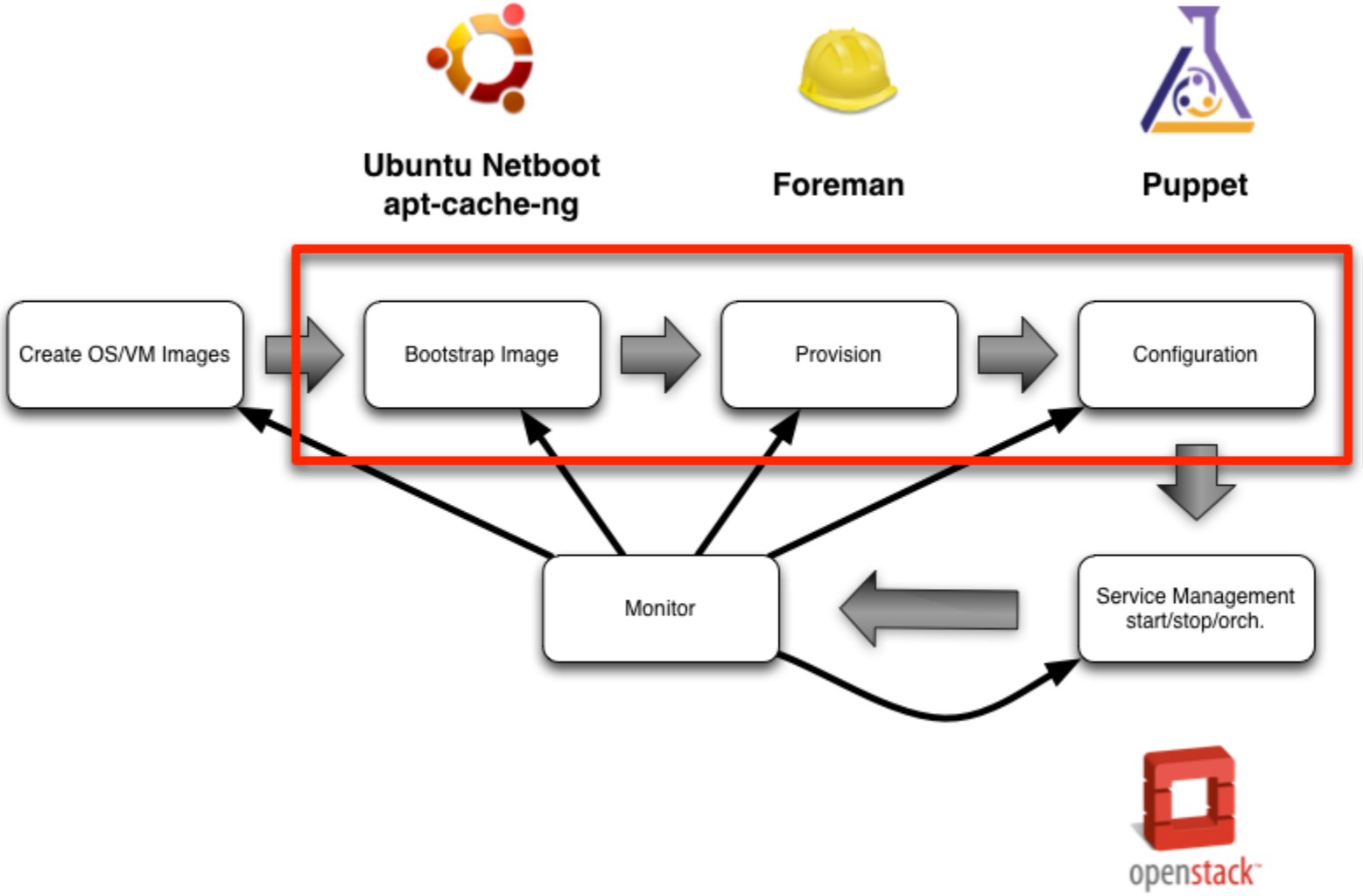
Challenges or Problems?

- How to **share/standardise** these processes?
  - Configuration - drift prevention
  - Testing - configuration, system functionality
  - Compliance - auditing, ITIL (eeek!)
  - Agility
  - Independence
    - Of physical/virtual deployment
    - Of infrastructure

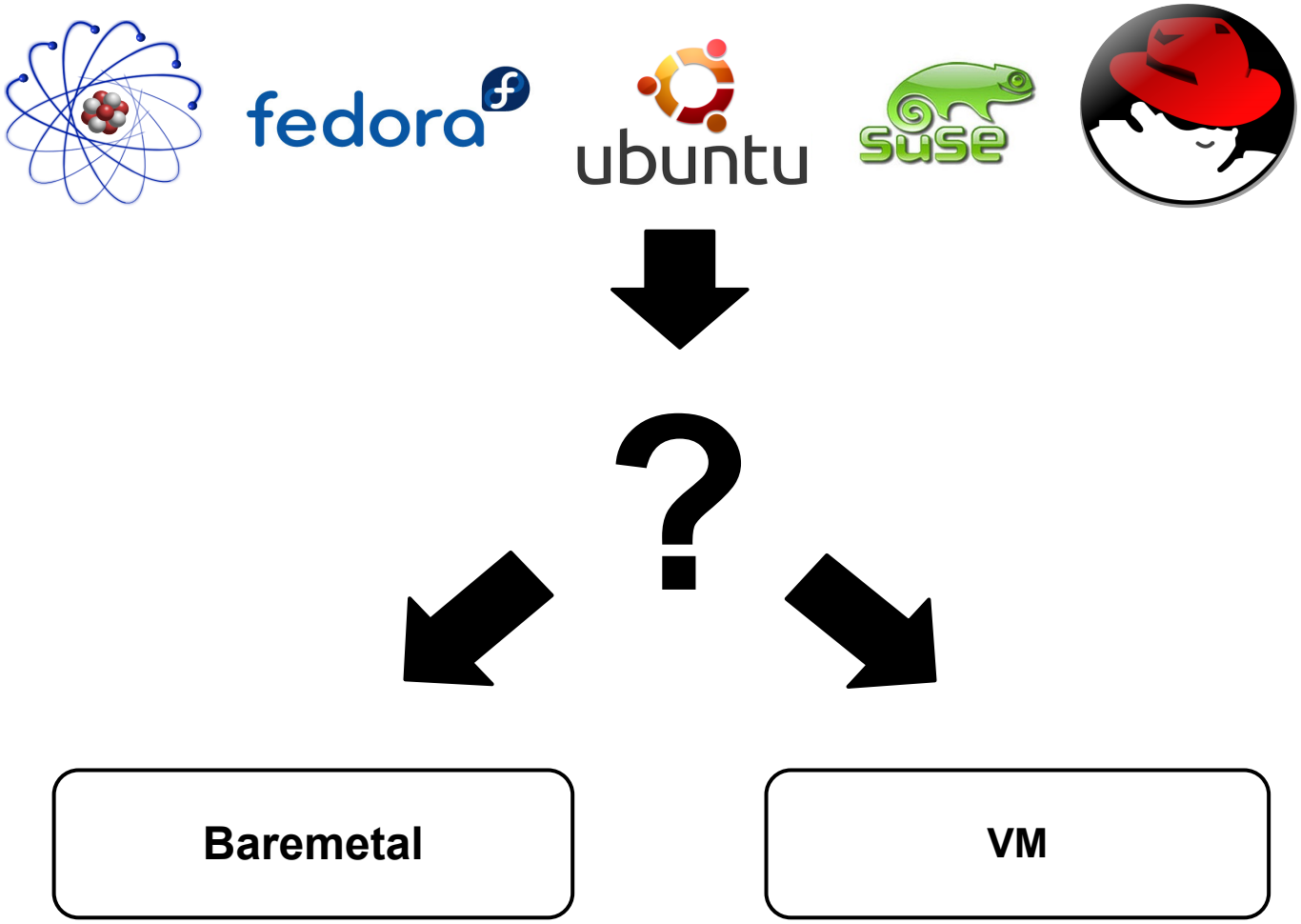
Automation Toolchain



Automation Toolchain



Provision - OS rollout



Provision - Foreman

- "Single Address For All Machines Lifecycle Management".
- Manages or proxies to DNS, DHCP, TFTP, Virtual Machines, PuppetCA, CMDB
- Integrates with Puppet (and acts as web front end to it).
- Provisions:
  - most flavours of *NIX, Windows
  - Virtual machines - libvirt, oVirt
  - Cloud Resources - Amazon EC2, VMware vCenter, OpenStack
- Has an API! Automate your Automation :-)

Provision - Foreman

What does it look like?

Foreman | Dashboard | Hosts | Reports | Facts | Audits | Statistics | More

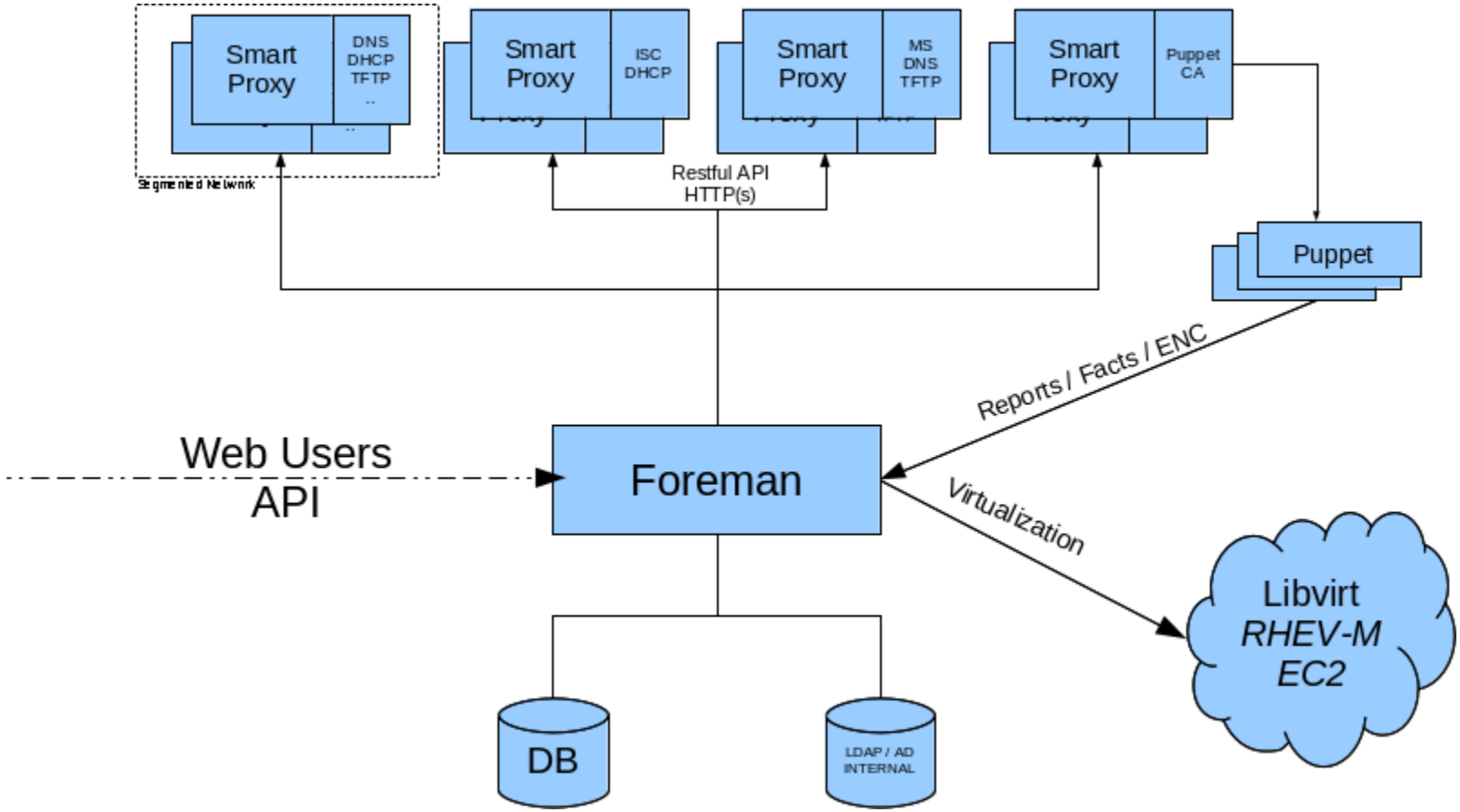
Hosts

Filter ... | Search | New Host

Name	Operating System	Environment	Model	Host Group	Last report	
compute1.cloudcomplab.ch	Ubuntu 12.04	iaas	X8DTU	production/os-worker	25 minutes ago	Edit Host
compute2.cloudcomplab.ch	Ubuntu 12.04	iaas	X8DTU	production/os-worker	25 minutes ago	Edit Host
compute3.cloudcomplab.ch	Ubuntu 12.04	iaas	X8DTU	production/os-worker	25 minutes ago	Edit Host
compute4.cloudcomplab.ch	Ubuntu 12.04	iaas	X8DTU	production/os-worker	25 minutes ago	Edit Host
compute5.cloudcomplab.ch	Ubuntu 12.04	iaas	X8DTU	production/os-worker	25 minutes ago	Edit Host
controller-prod.cloudcompla...	Ubuntu 12.04	iaas	VirtualBox	production/os-controller	11 minutes ago	Edit Host

Displaying **all 6** Host - **0** Selected

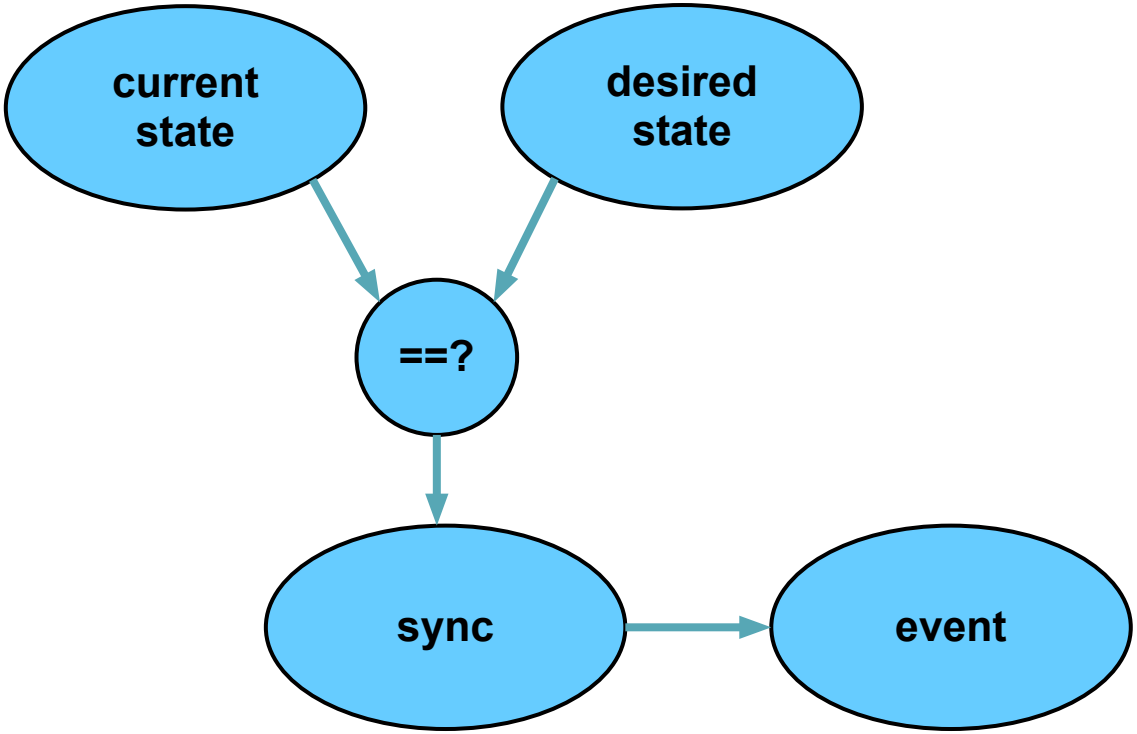
Provision - Foreman Arch



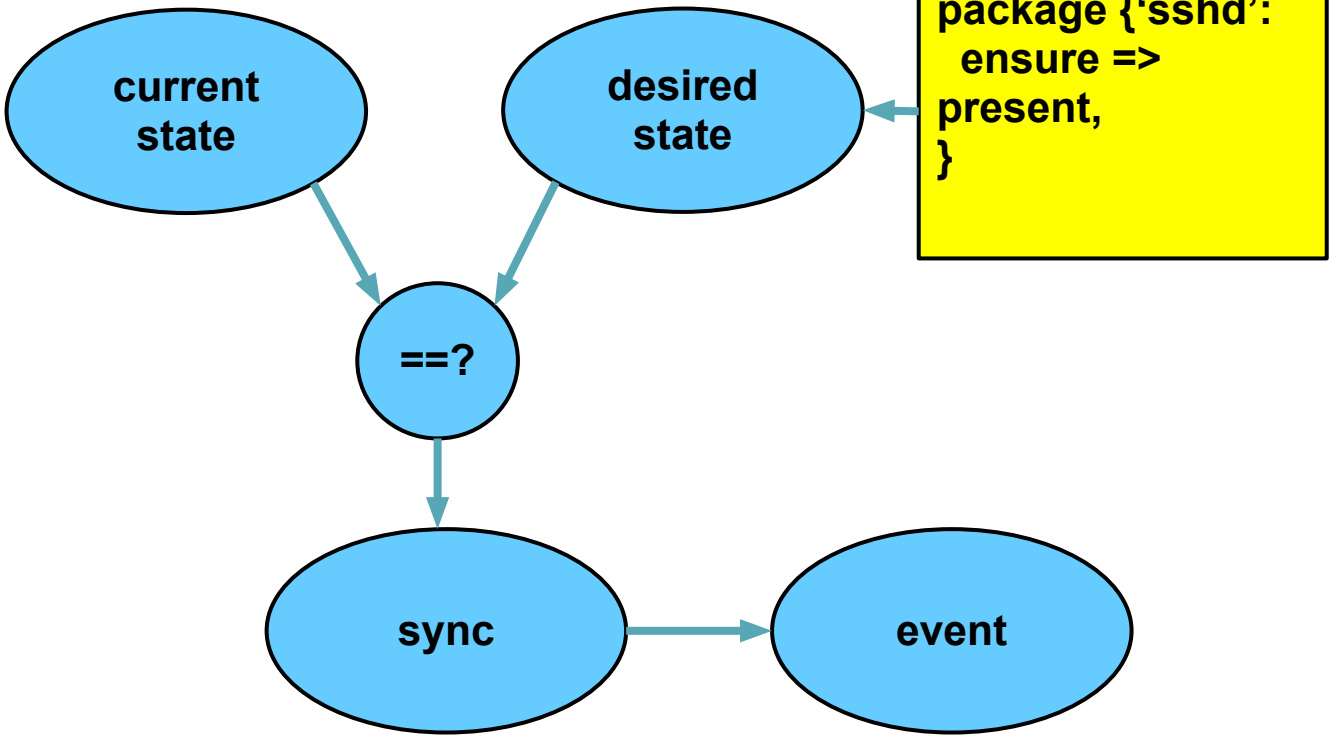
Configuration - Puppet

- Declarative configuration language
  - Describe desired state of a system, not how to achieve it
  - Idempotence
- Different types of resources: software package, service, user, configuration file, mysql database, ...
- Dependencies can be formulated
- Grouping of resources by "class" concept:
  - Way of structuring your descriptions
- Abstraction layer for resources:
  - Independence from system type (different variants of linux, *bsd, mac os, windows, ...)

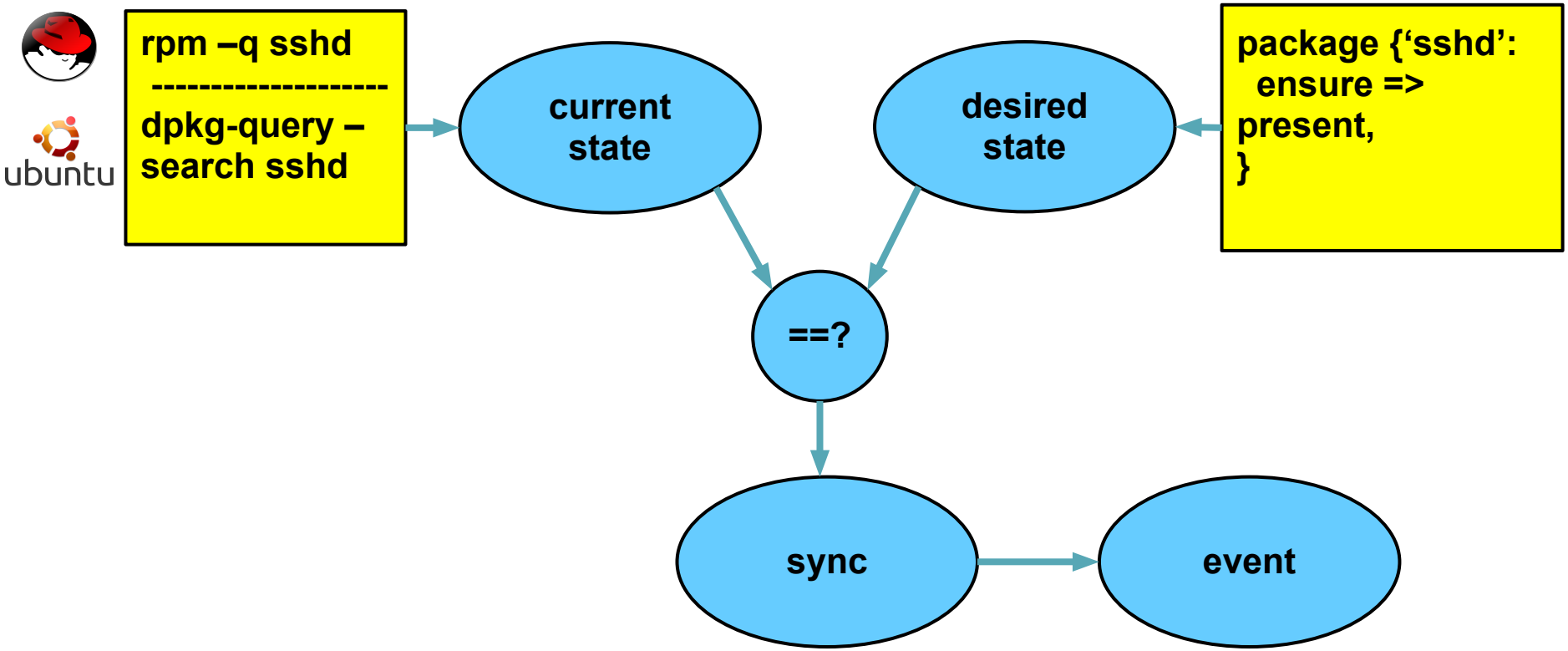
Configuration - Puppet's Model



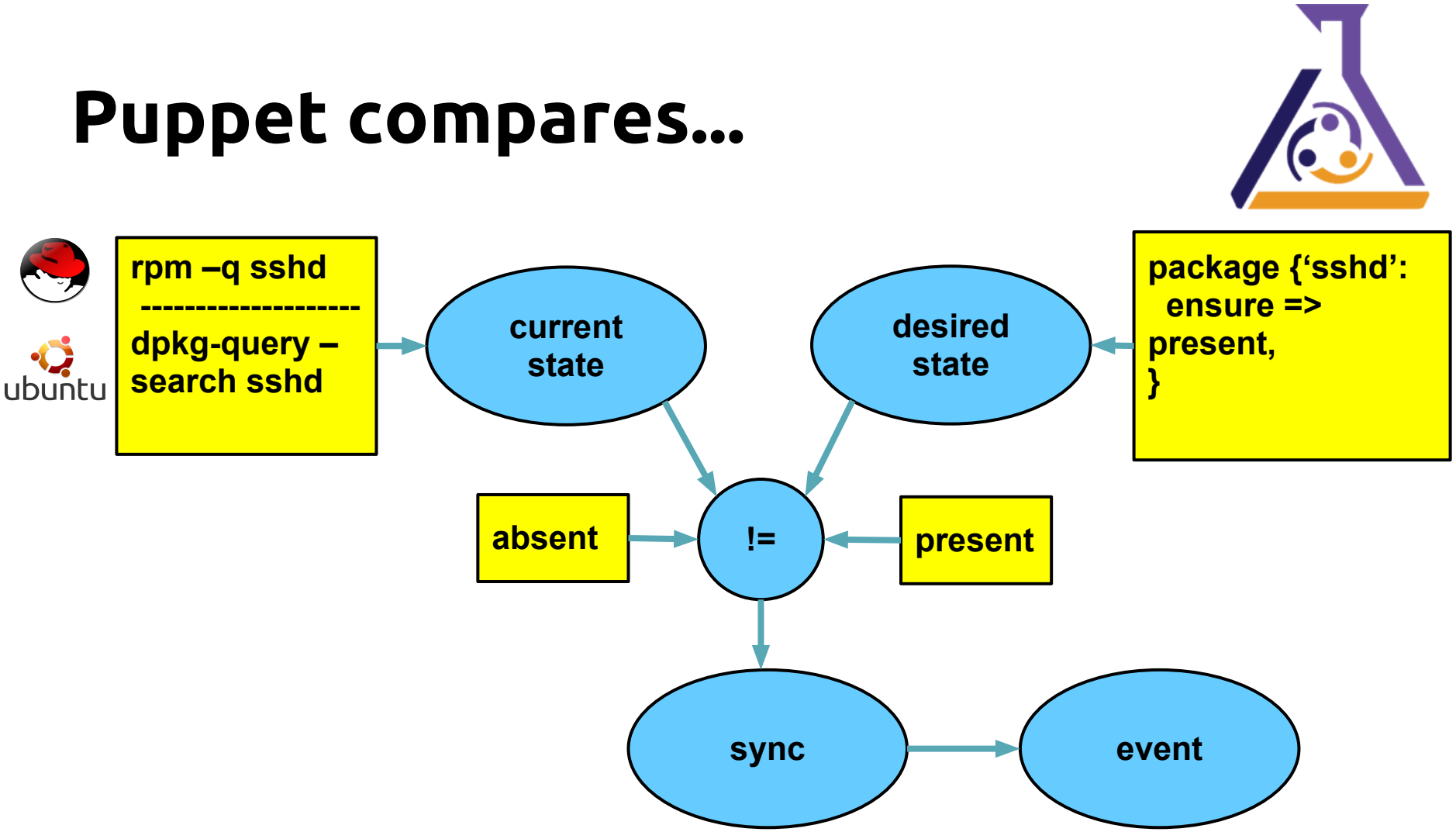
You describe system state...



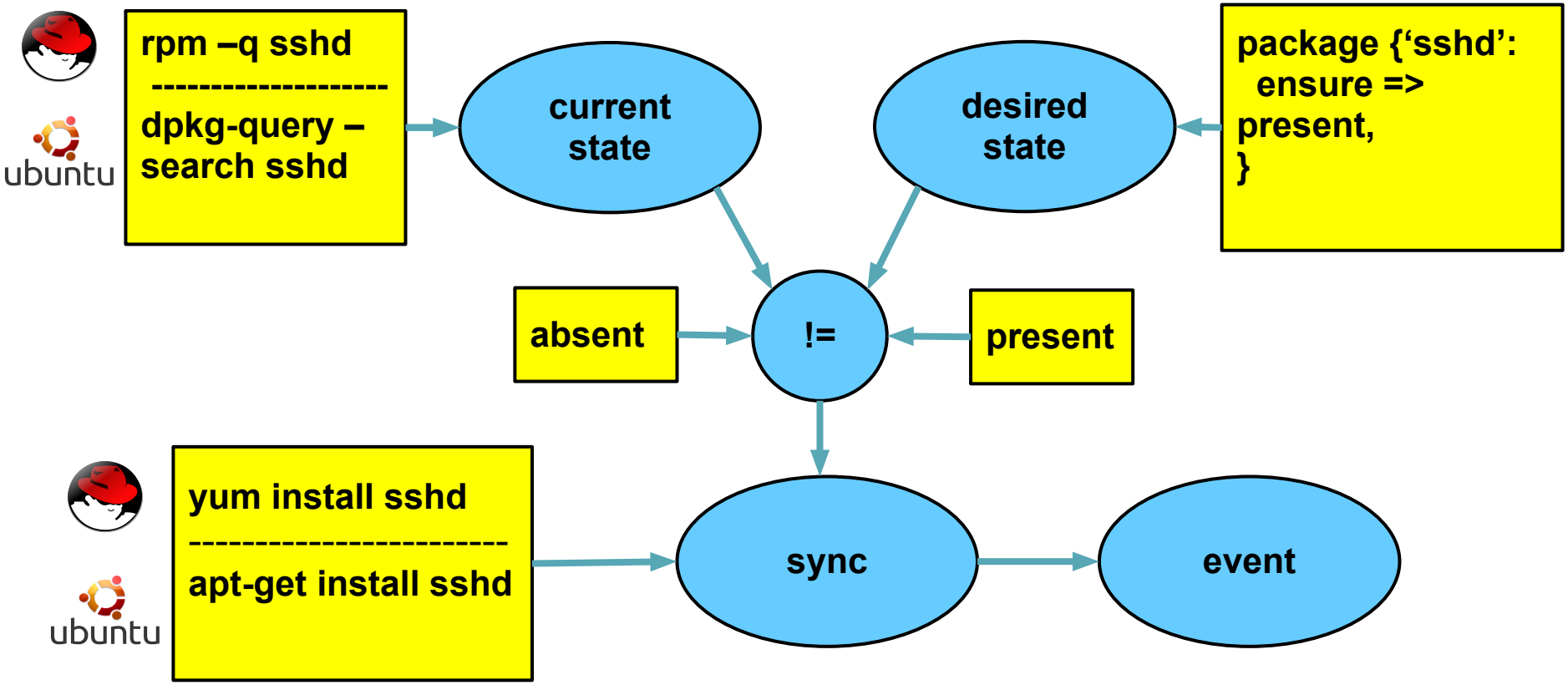
Puppet collects current state...



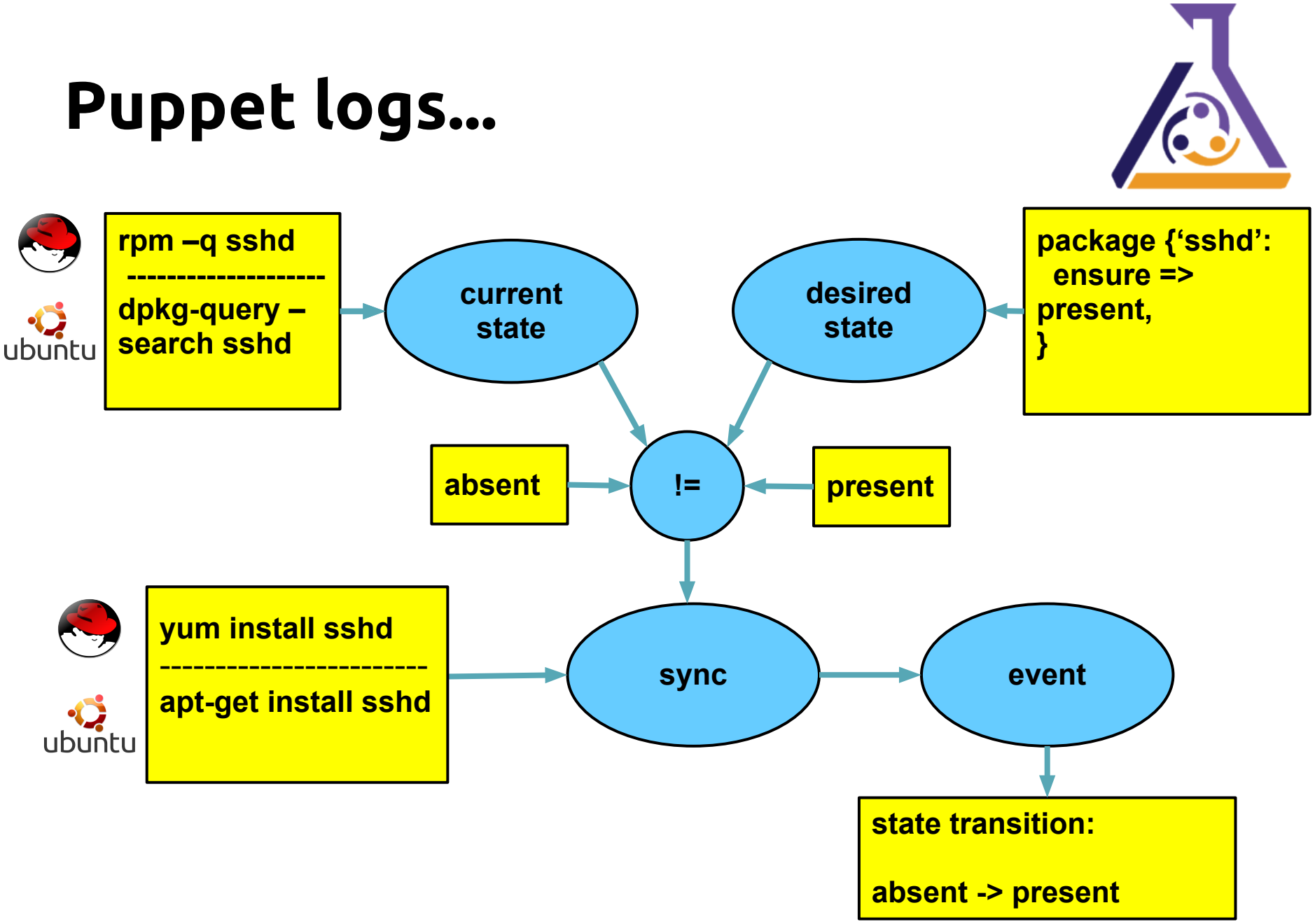
Puppet compares...



Puppet synchronizes...



Puppet logs...


rpm –q sshd
dpkg-query –search sshd
current state
desired state
package {'sshd': ensure => present, }
absent
!=
present
yum install sshd
apt-get install sshd
sync
event
state transition:
absent -> present
ubuntu
ubuntu

A more complete puppet manifest

```
class ssh::install {
        package { "openssh":
                ensure => present, }
}
class ssh::config {
        file { "/etc/ssh/sshd_config":
                ensure      => present,
                owner       => 'root',
                group       => 'root',
                mode        => 0600,
                source      => "puppet:///modules/ssh/sshd_config",
                require     => Class["ssh::install"],
                notify      => Class["ssh::service"], }
}
class ssh::service {
        service { "sshd":
                ensure          => running,
                hasstatus       => true,
                hasrestart      => true,
                enable          => true,
                require         => Class["ssh::config"], }
}
class ssh {
        include ssh::install, ssh::config, ssh::service
}
```

dependency

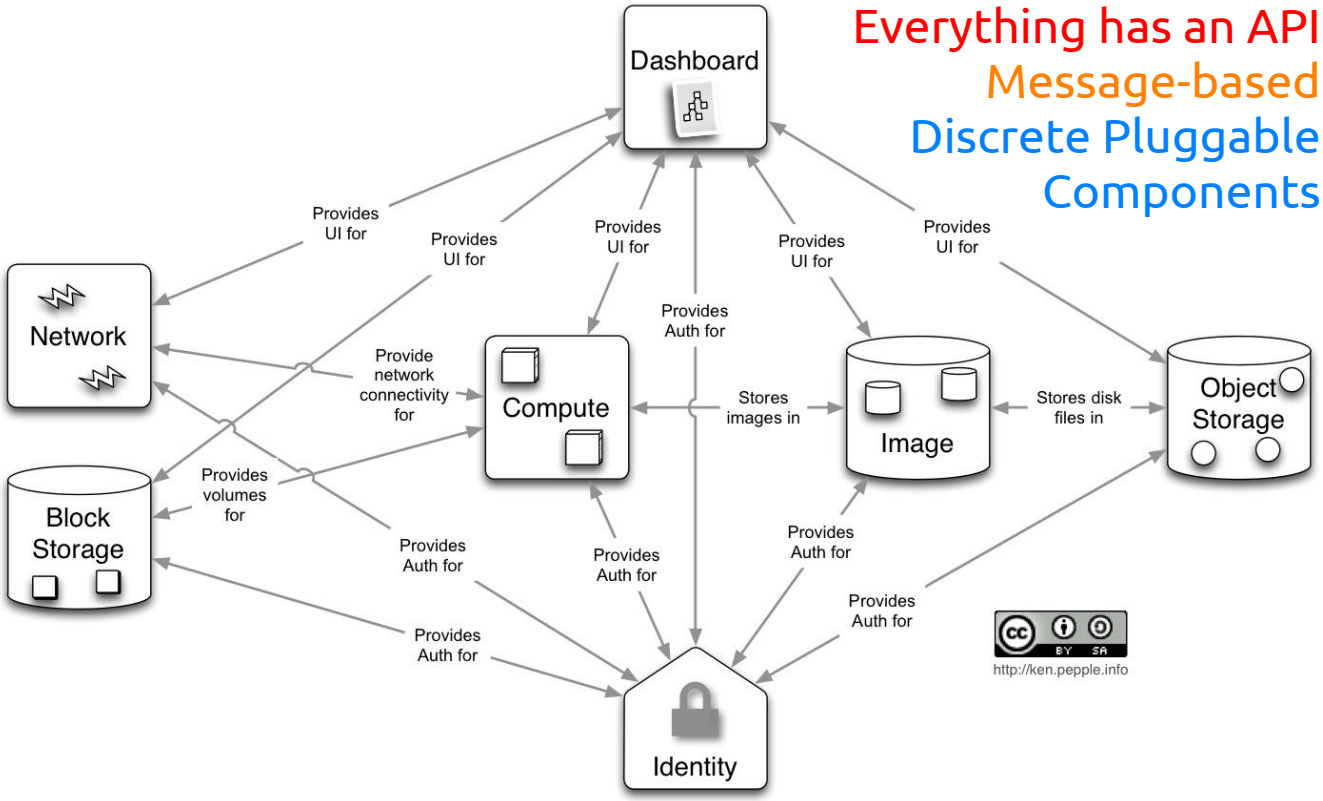
"if I change..."

OpenStack @ 10,000m, Looks Easy!

Software as a Service

Platform as a Service

Infrastructure as a Service



Everything has an API

Message-based

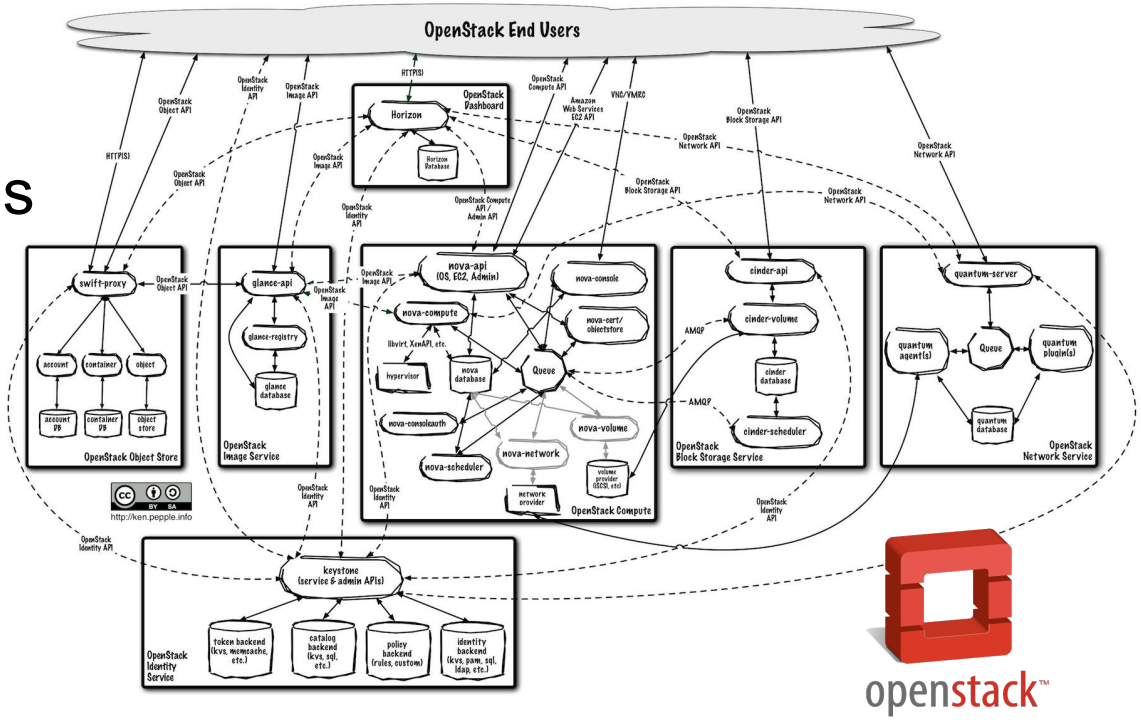
Discrete Pluggable Components

Essex was simpler!

OpenStack - The Ugly Close-up

Complicated

- Many Services
- Many Dependencies



Challenge to deploy

- 100's, 1000's of nodes?

You **need** an automated toolchain!

Apple Moment!

We have one!
Want to see?

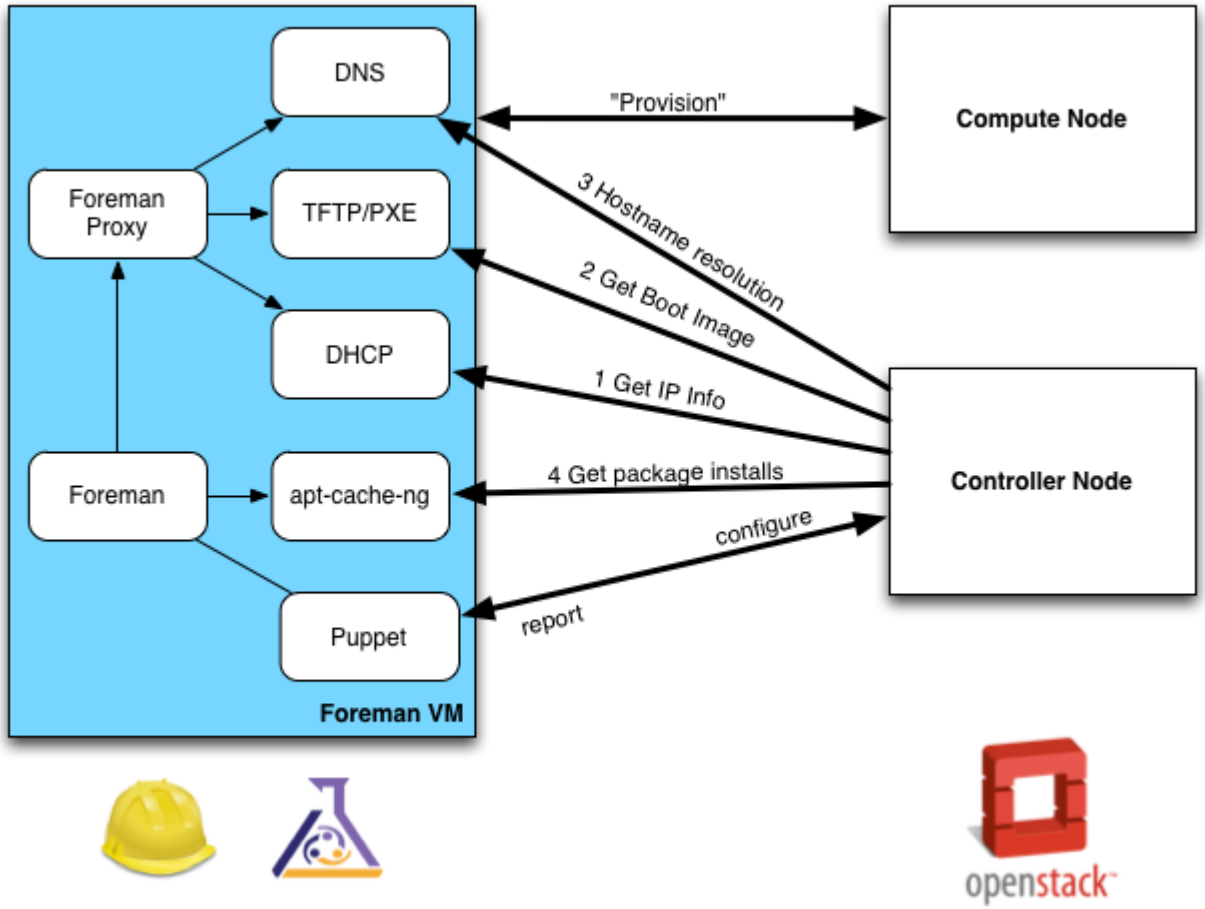
Demo - What Could Go Wrong?!

Multi-node OpenStack Installation

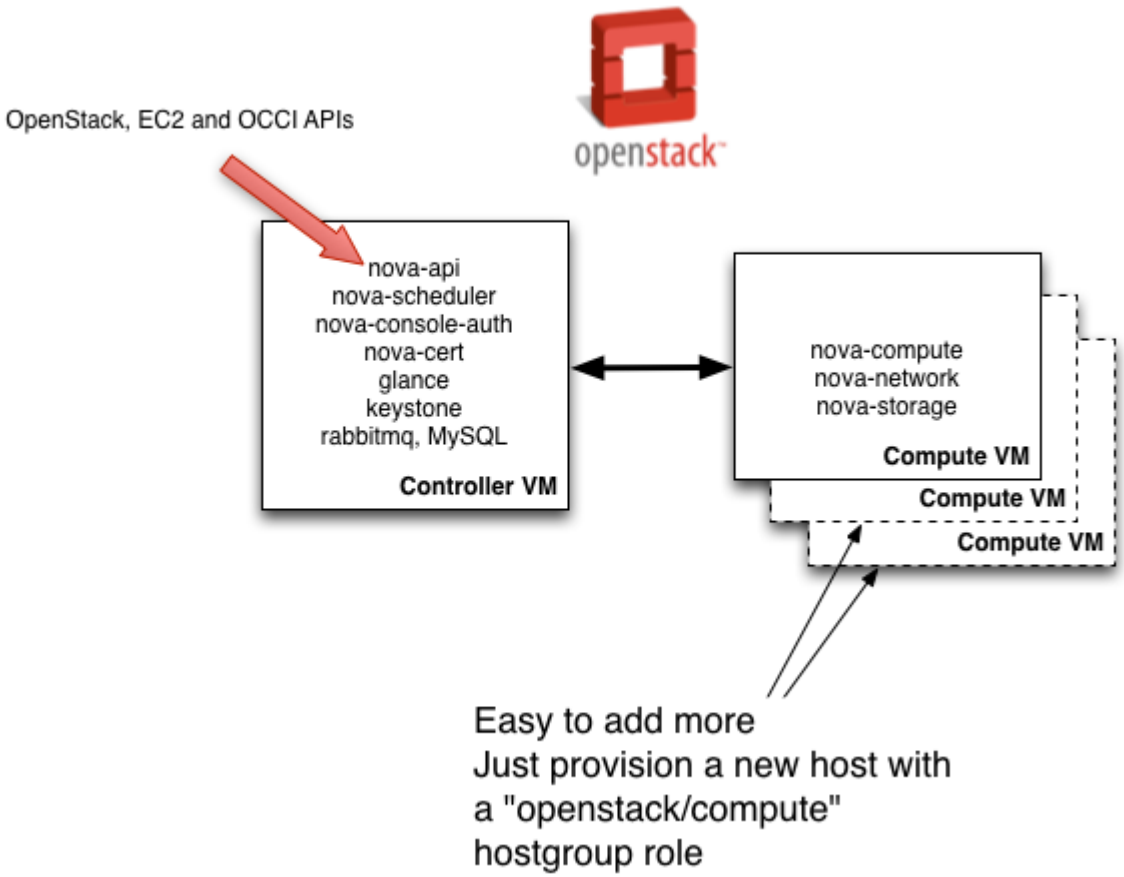
- 1 controller node
  - `"boss"`
- 1 compute node
  - `"worker1"`

- More time? Easy to add more.

Demo: Deployment Architecture



Demo: OpenStack Component Deployment



Demo: Code/Config Details

- There are 2 roles (*hostgroups*)
  - **openstack/controller** - `controller.pp`
  - **openstack/compute** - `compute.pp`

- Both have different puppet manifests
  - Same '`icclab`' module

```
root@foreman:/etc/puppet/modules/iaas# tree icclab
icclab
└── manifests
    ├── all_in_one.pp
    ├── controller.pp
    ├── compute.pp
    └── params.pp

1 directory, 4 files
```

What's in a controller node?

(excluding common params)

```
class icclab::controller{

  include icclab::params

  $admin_password           = 'admin_pass'
  $keystone_admin_token     = 'keystone_pass'

  class { 'openstack::controller':

    public_address          => $icclab::params::controller_node_public,
    public_interface        => $icclab::params::public_interface,
    private_interface       => $icclab::params::private_interface,
    internal_address        => $icclab::params::controller_node_internal,
    floating_range          => '192.168.56.128/25',
    fixed_range             => $icclab::params::fixed_range,
    multi_host              => true,
    network_manager         => $icclab::params::network_manager,
    verbose                 => true,
    auto_assign_floating_ip => false,
    mysql_root_password     => 'mysql_root_password',
    admin_email             => 'admin@iownz.you',
    admin_password          => $admin_password,
    keystone_db_password    => 'keystone_db_password',
    keystone_admin_token    => $keystone_admin_token,
    glance_db_password      => 'glance_pass',
    glance_user_password    => 'glance_pass',
    nova_user_password      => 'nova_pass',
    nova_user_password      => $icclab::params::nova_user_password,
    rabbit_password         => $icclab::params::rabbit_password,
    rabbit_user             => $icclab::params::rabbit_user,
    export_resources        => false,

  }

  # Optional: include if you want authorisation information
  #           stored in a local file, located in /root/
  class { 'openstack::auth_file':

    admin_password       => $admin_password,
    keystone_admin_token => $keystone_admin_token,
    controller_node      => $icclab::params::controller_node_internal,

  }

}
```

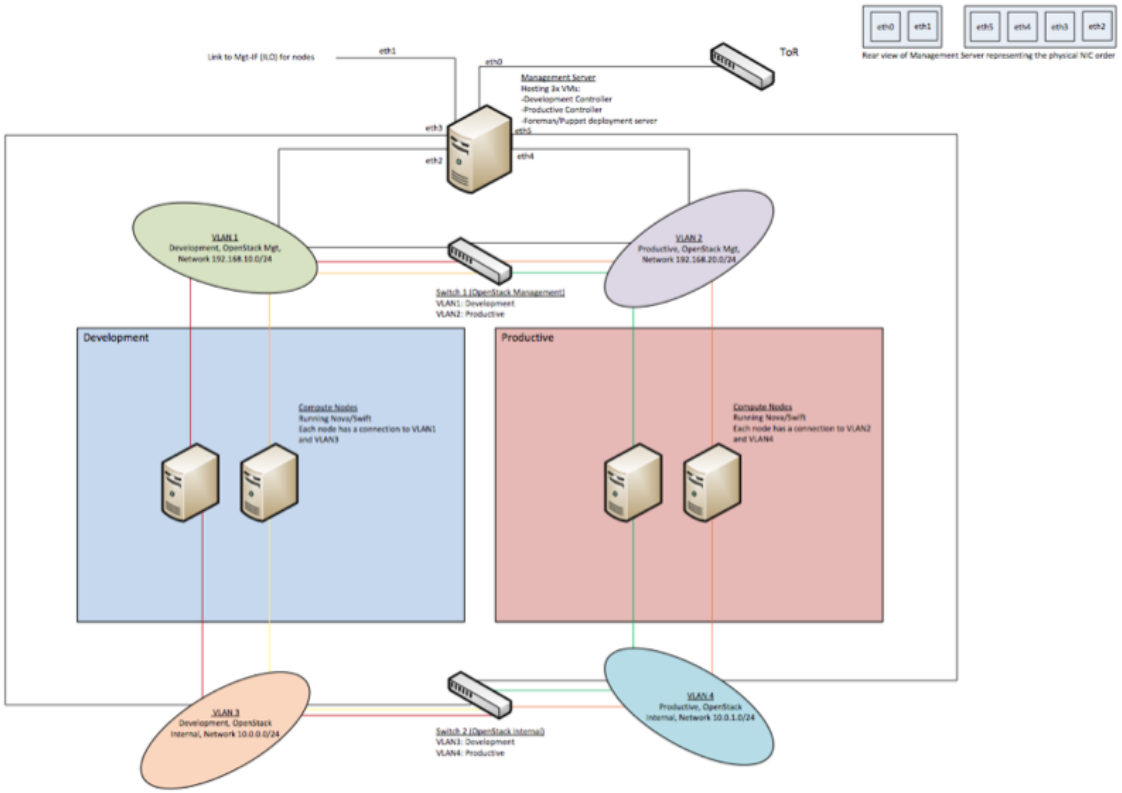
What's in a compute node?

(excluding common params)

```
class icclab::compute{

  include icclab::params

  class { 'openstack::compute':

    public_interface   => $icclab::params::public_interface,
    private_interface  => $icclab::params::private_interface,
    internal_address   => $ipaddress_eth0,
    libvirt_type       => 'qemu',
    fixed_range        => $icclab::params::fixed_range,
    network_manager    => $icclab::params::network_manager,
    multi_host         => true,
    sql_connection     => $icclab::params::sql_connection,
    nova_user_password => $icclab::params::nova_user_password,
    rabbit_host        => $icclab::params::controller_node_internal,
    rabbit_password    => $icclab::params::rabbit_password,
    rabbit_user        => $icclab::params::rabbit_user,
    glance_api_servers => "${icclab::params::controller_node_internal}:9292",
    vncproxy_host      => $icclab::params::controller_node_public,
    vnc_enabled        => true,
    verbose            => true,
    manage_volumes     => true,
    nova_volume        => 'nova-volumes'

  }

}
```

In the ICCLab

- 15 nodes in ~15 minutes.
- Supports 2 networks
  - research (essex, soon to be folsom)
  - stable (essex)



Conclusions/Learnings

- Automation is **essential**
- Puppet **codifies** learnings, makes **sharing** easy
- Foreman a **central** management point, full lifecycle, adaptable to other services
- Dependence on infrastructure service management frameworks is lessened
  - Fast and efficient to install new ones with a tool chain
- Other than SLA guarantees, the only guarantee to maintain is the API between provider and customer and this is where standard APIs are need such as OCCI/CDMI/OVF.

Going Forward...

- Folsom to be Deployed
  - **Only** on research network
  - Essex remains on stable
- High Availability to be implemented
- Deploy and use Ceilometer
- Develop further Hadoop as a Service
- Consider/deploy other host OSes (SmartOS)
- Integrate OpenFlow switches
- ICCLab to run/support:
  - EU FP7 IP MobileCloud Network
  - EU FI-PPP KIARA (FIware)
  - KTI projects

Thanks!
Questions?

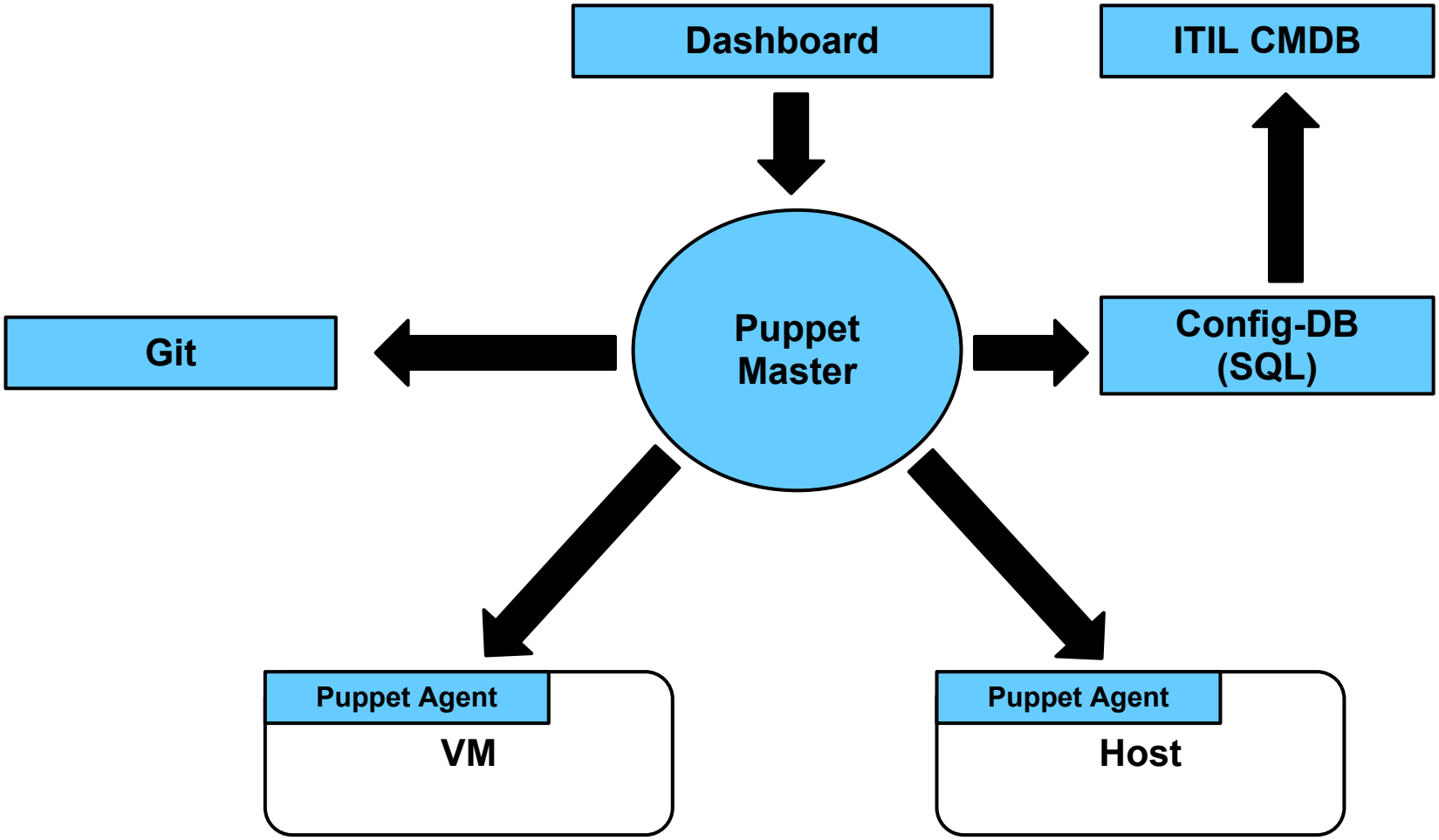
(ping me: @dizz, edmo@zhaw.ch)

Everything Presented is Documented at:
http://www.cloudcomp.ch

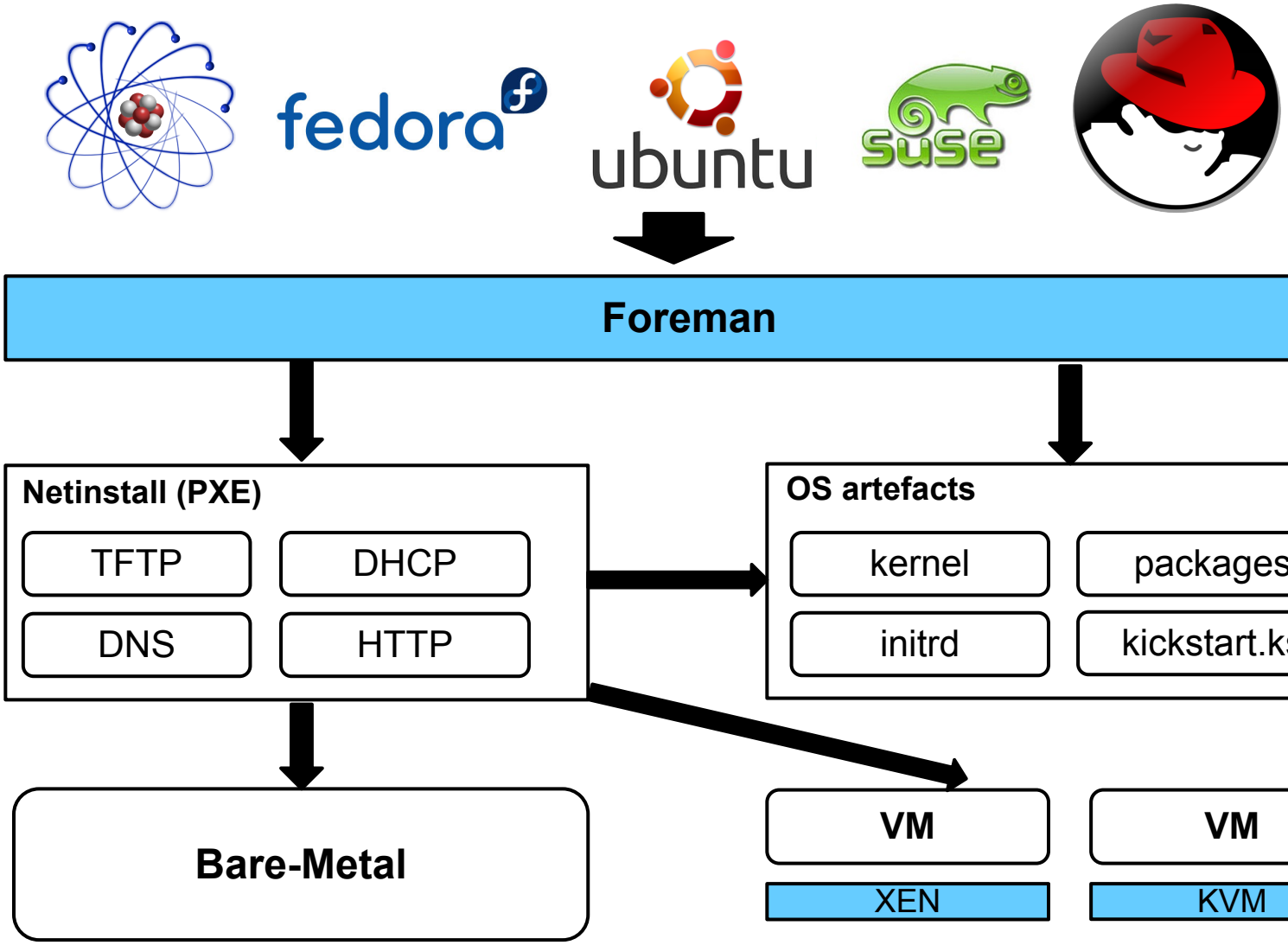
Including:
- HOWTOs
- Foreman, Puppet, OpenStack installs
- Virtual Machine images

Backup Slides

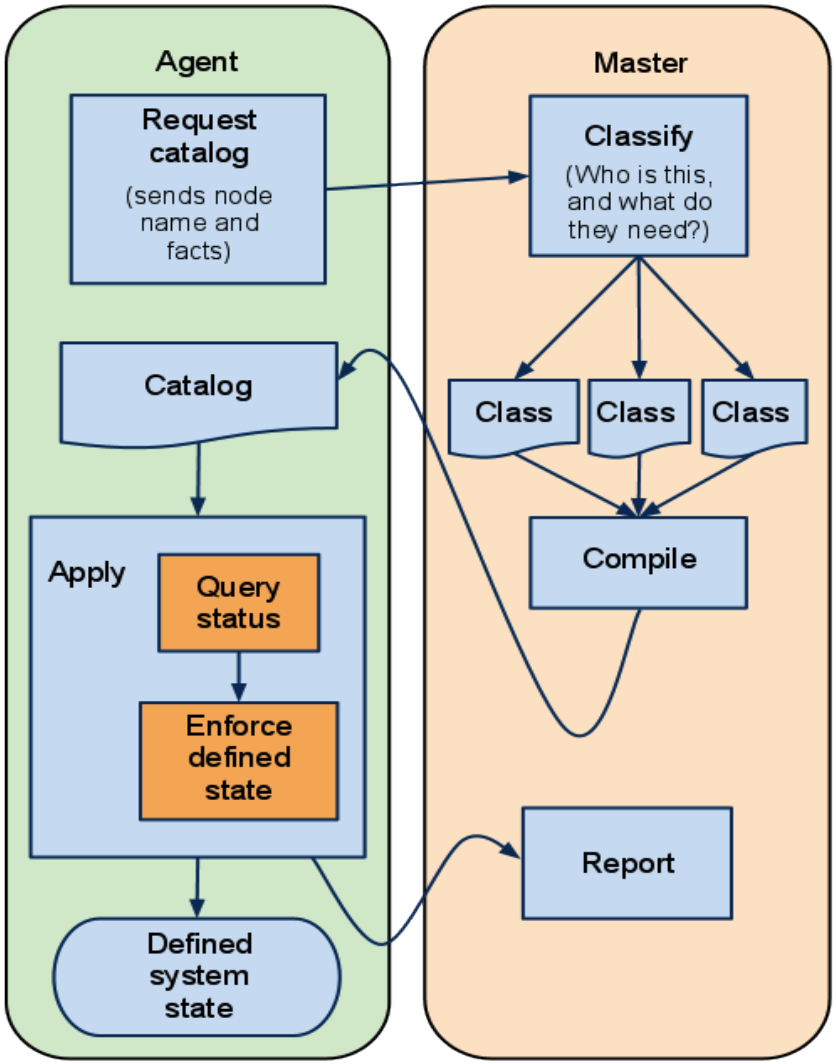
Toolchain map



Foreman Arch



Puppetmaster <-> agent interaction



What are the common config params?

```
class icclab::params{


  /* -----------Shared Connection Settings-------------*/

  ############ Important to set! ############
  $controller_node_address  = '192.168.56.3'

  $controller_node_public   = controller_node_address
  $controller_node_internal = controller_node_address
  $sql_connection           = "mysql://nova:${icclab::params::nova_db_password}@${controller_node_internal}/nova"

  /* ------------------------------------------------*/


  /* -------------Shared Auth Settings----------------*/
  $nova_user_password       = 'nova_pass'
  $rabbit_password          = 'rabbit_pass'
  $rabbit_user              = 'rabbit_user'
  /* ------------------------------------------------*/


  /* ----------Shared Networking Settings-------------*/
  $network_manager          = 'nova.network.manager.FlatDHCPManager'
  $fixed_range              = '10.0.0.0/24'
  $public_interface         = 'eth0'
  $private_interface        = 'eth1'
  /* ------------------------------------------------*/

}
```