Puppet Fundamentals :
Session 3

Cobbler
The Foreman
Architecture
Architecture Case Studies
Architecture Examples
Q & A
Wind-up

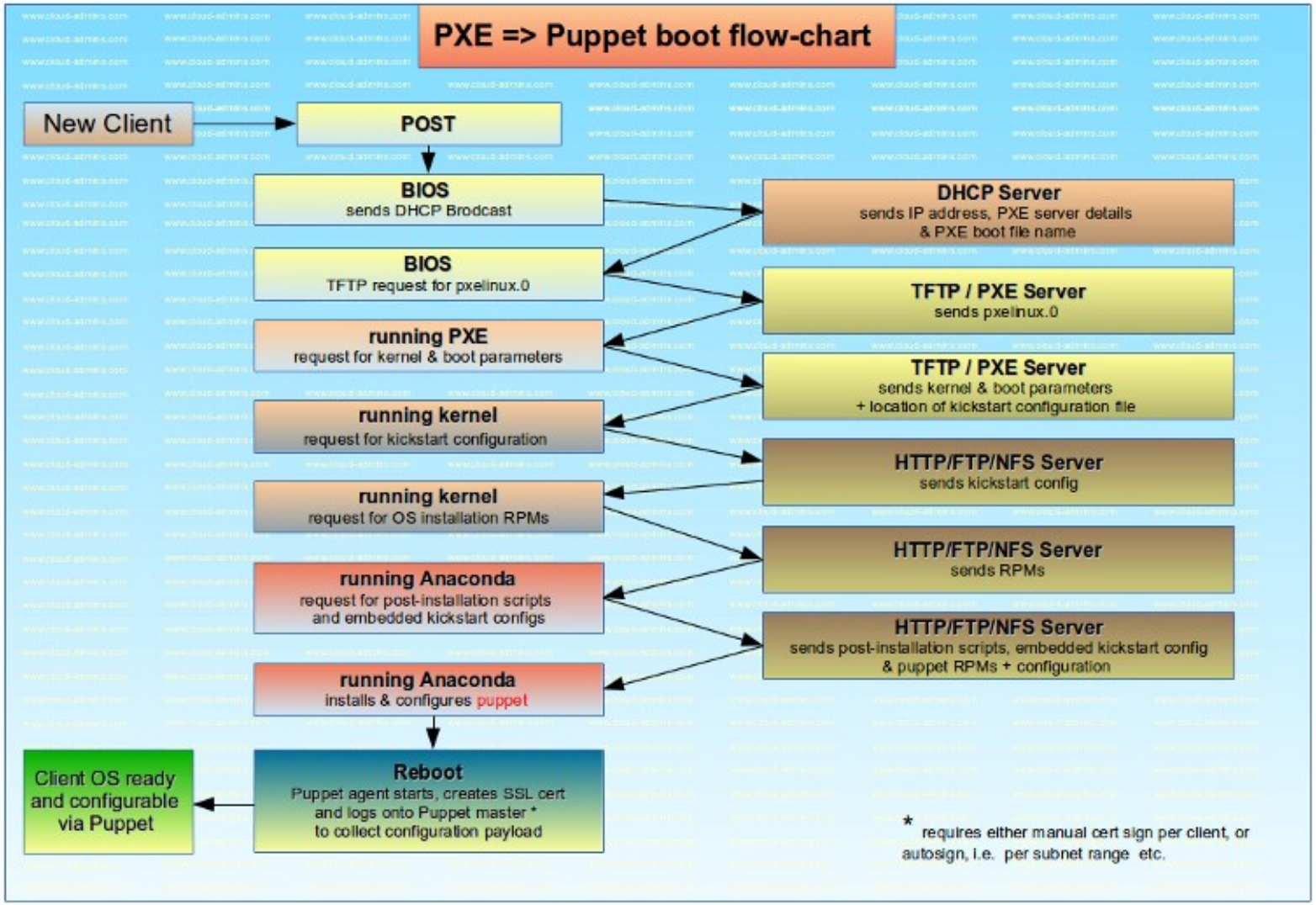
Cobbler : Introduction

http://www.cobblerd.org

- A lightweight installation management system
- Boots multiple machines via PXEboot, TFTP, and DHCP
- Supports Fedora, RedHat, openSuSE, Debian and Ubuntu

- Supports two forms of Puppet integration
  - Class management (as an ENC)
  - Simple hand-off

Cobbler : PXE booting



Cobbler : Integration

- Cobbler can generate classes for puppet in yaml.
- This is standard behaviour for an External Node Classifier

```
---
classes:
    - distro1
    - webserver
    - likes_llamas
    - orange
parameters:
    tree: 'http://.../x86_64/tree'
```

Cobbler : Integration

- Setting up puppet to use Cobbler as an ENC involves configuring puppet.conf

/etc/puppet/puppet.conf

[main]
node_terminus = exec
external_nodes = /usr/bin/cobbler-ext-nodes

Cobbler : Integration

- Build your node form cobbler with its own commands:

```
cobbler distro edit --name=distro1 --mgmt-classes="distro1"
cobbler profile add --name=webserver –distro=distro1 \
        --mgmt-classes="webserver likes_llamas" --kickstart=/etc/cobbler/my.ks
cobbler system edit --name=system --profile=webserver --mgmt-classes="orange" \
        --dns-name=system.example.org
```

The Foreman : Index

- Introduction
- Components
- How it should look
- Walk-through

The Foreman : Life-cycle Management

The Foreman: A way to manage your entire infrastructure

- Designed for Openstack management
- Supports Bare Metal, VMs and Openstack Clould
- Full OS install
- Network Configuration
- Puppet Agent configuration
- Monitoring

The Foreman : Components

Smart Proxy

A remote agent allow Foreman to integrate with other servers that are performing a specific function; such as TFTP, DHCP, Puppet, Puppet CA, and DNS

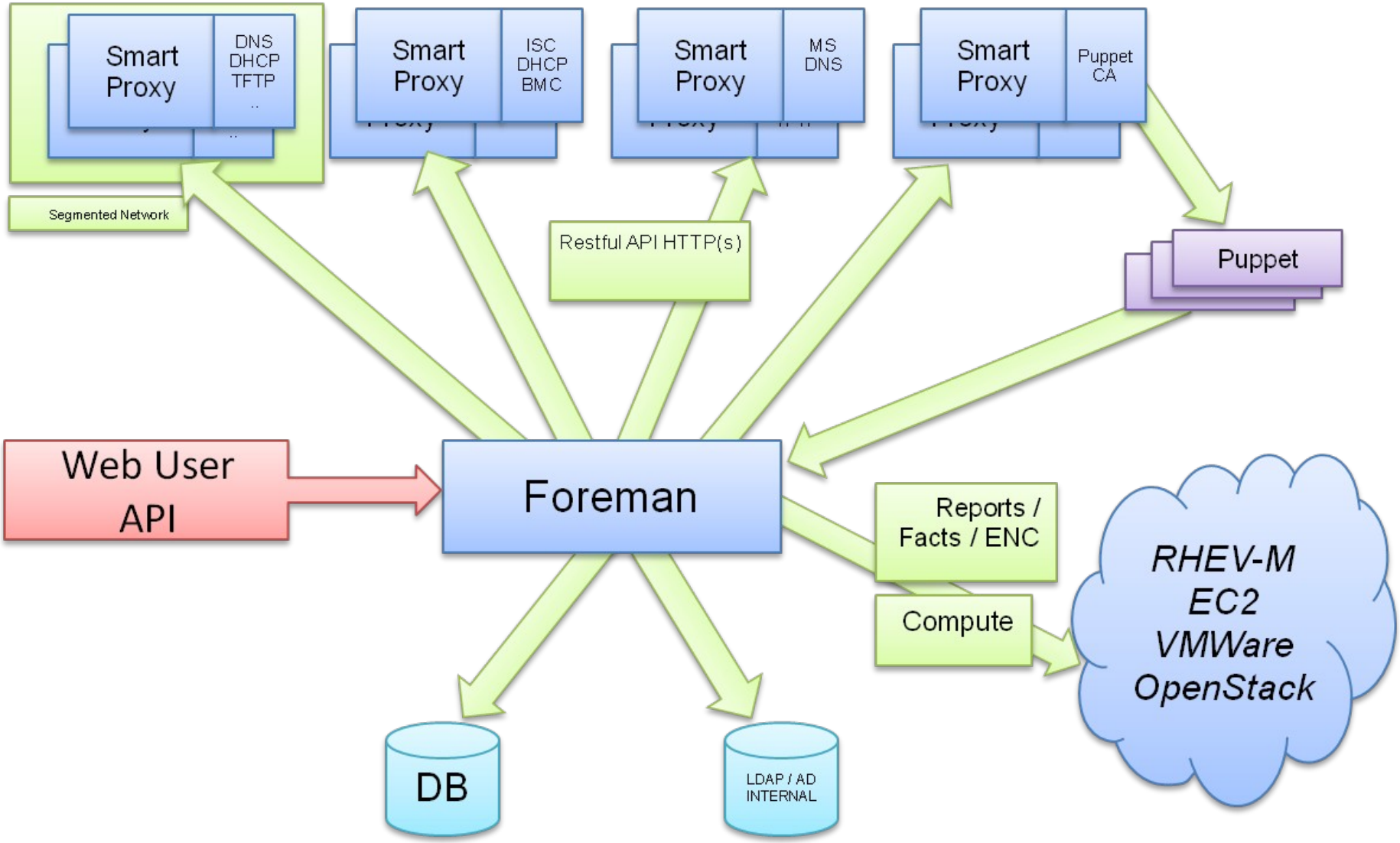
Web GUI

The main Foreman instance is responsible for the web interface, as-well as configuration files

Puppet

Foreman installs Puppet and/or Puppet CA as part of its installation

The Foreman : Components



The Foreman : Single Interface

- Configure DHCP and populate DNS
- Define VM resources and VLANs (Virtual Box)
- Select relevant puppet classes
- OS installation with preconfigured puppet agent
- Monitoring and Management of Puppet after build

The Foreman : New Node

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

New compute

Name: EC2-East-corp

Provider: EC2

Description:

Access Key: xxxxxxxxxxxxxxxxxxxxxx

Secret Key: ••

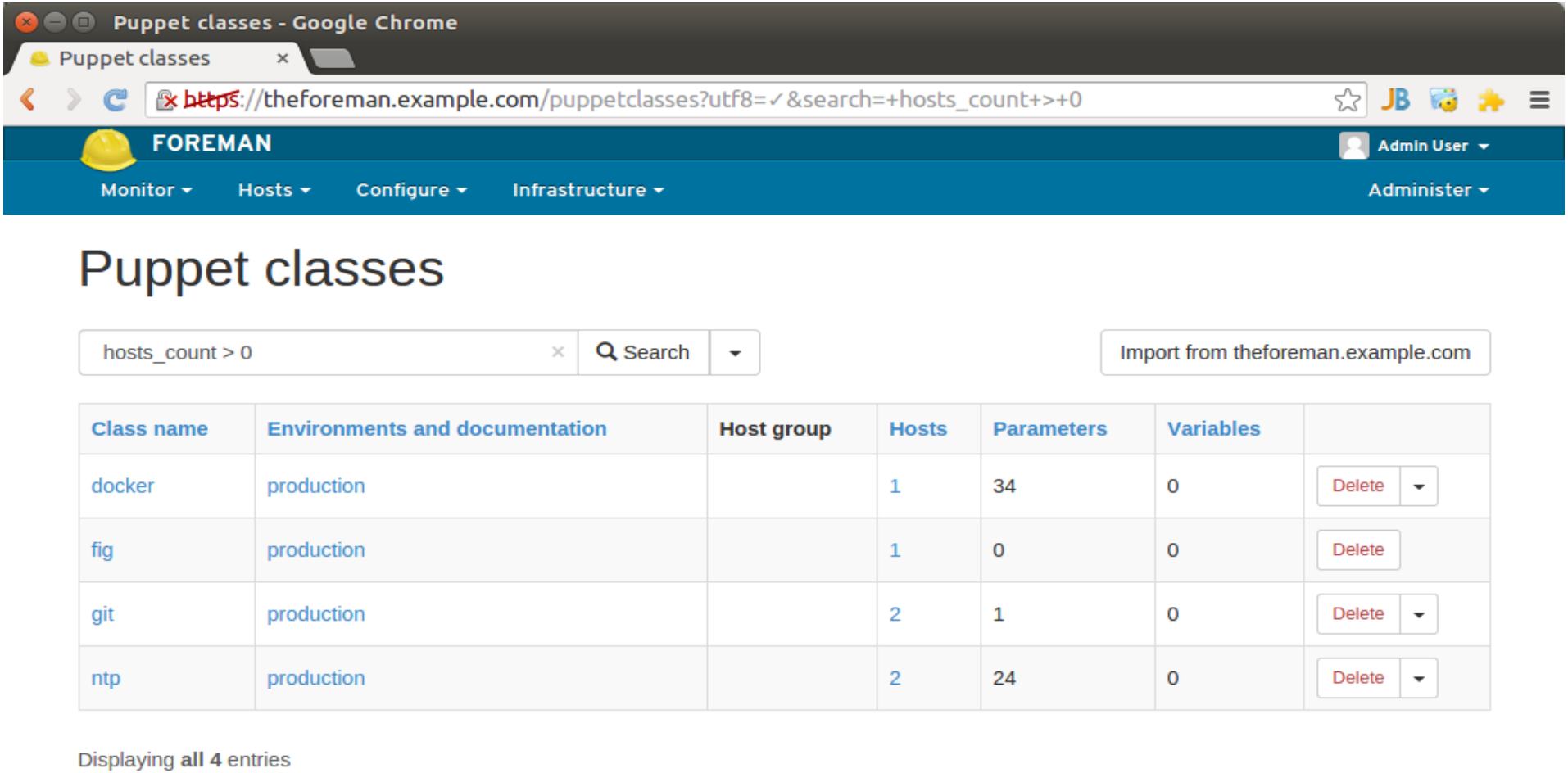
Region: us-east-1 [Test Connection]

[Cancel] [Submit]

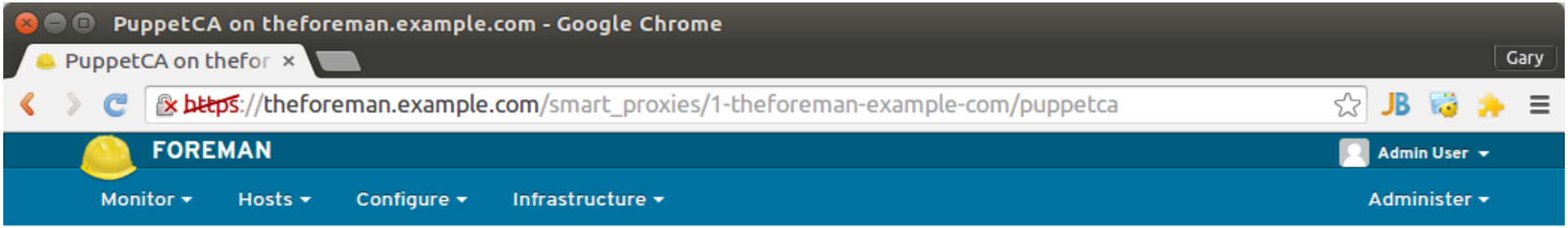
Version 0.5 © 2009-2012 Paul Kelly and Ohad Levy

Help | Wiki | Support

The Foreman : Puppet Classes



The Foreman : Certificate Management



PuppetCA on theforeman.example.com

Filter by state:

Autosign Entries

Certificate Name	State	Valid from	Expires	Fingerprint	
agent01.example.com	valid	3 days ago	in almost 5 years	SHA256	Delete
agent02.example.com	pending	N/A	N/A	SHA256	Sign
theforeman.example.com	valid	3 days ago	in almost 5 years	SHA256	Delete

Displaying **all 3** entries

The Foreman : Monitoring

Overview - Google Chrome

Overview

https://theforeman.example.com/dashboard

FOREMAN

Admin User

Monitor | Hosts | Configure | Infrastructure | Administer

Overview

Filter ... | Search

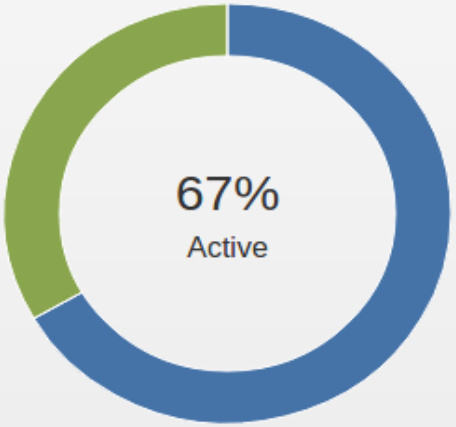
Generated at 18 Jan 03:33 | Manage dashboard

Host Configuration Status

Hosts that had performed modifications without error	2
Hosts in error state	0
Good host reports in the last 35 minutes	1
Hosts that had pending changes	0
Out of sync hosts	0
Hosts with no reports	0
Hosts with alerts disabled	0

Total Hosts: 3

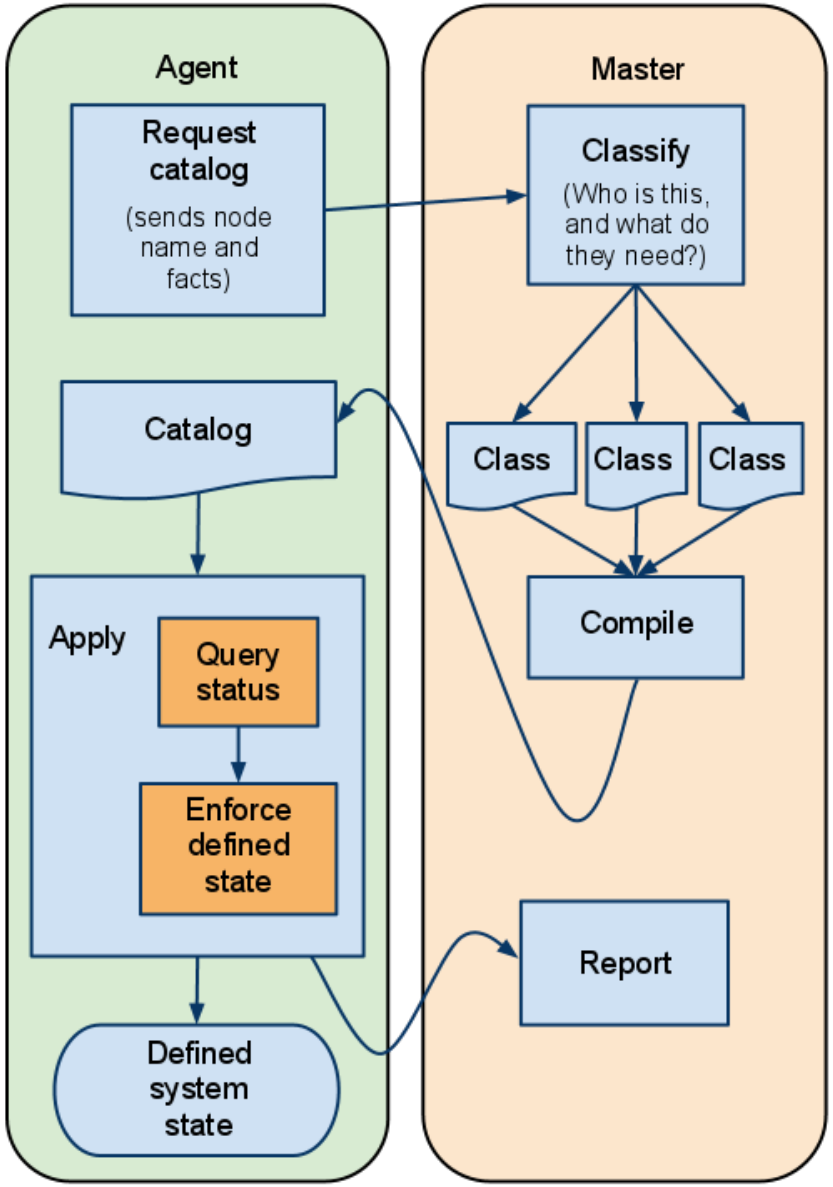
Host Configuration Chart



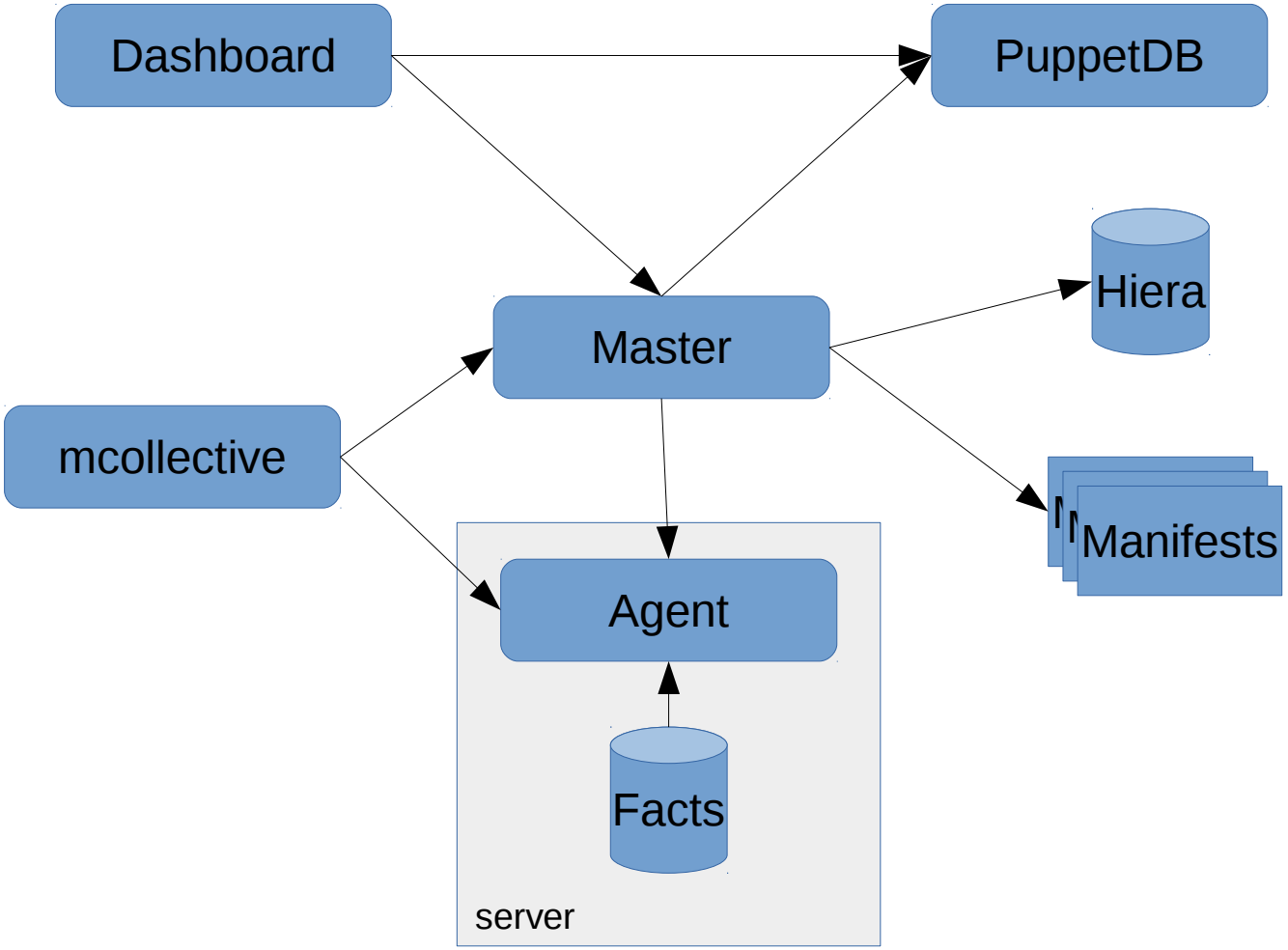
https://theforeman.example.com/dashboard#

Architecture : Recap

Architecture : Interaction



Architecture : Components



Architecture : Major Decisions

- Master or Master-less
- Redundant puppet-masters
- Puppet-master network location
- Orchestration Tool : *mcollective, Salt-stack, Fabric*
- Reporting front-end : *PuppetDB, PuppetBench, Puppet Dashboard, Foreman*
- Module deployment mechanism : *git, rsync, tar*
- Puppet environments : *production, testing, development*
- Testing approach : *git hooks, pair programming, test environments*
- Development approach : *github, CVS, Subversion*
- Puppetizing Nodes : *Kickstart, cloud image*

Architecture : Master vs Master-less

Puppet-master :

- Provides a central point of control (and failure)
- Suitable for long-lived machines where configuration changes over time

Master-less :

- “puppet apply” is executed on every node according to
  - A deployment job
  - Cron
  - On boot
- Good for massively distributed or disposable environments

Architecture : Orchestration Tools

Why Orchestration :

- Orchestration describes the automated arrangement, coordination, and management of complex, cross-functional, enterprise-wide systems, middleware, and services.
- Perform staged tasks, or complex application rollouts

  (because puppet cant configure your database server before your web-server)

mcollective :

- From the PuppetLabs team – Better integration, and with lots plugins
- Simple CLI (but complex to install the first time)

  https://puppetlabs.com/mcollective

Fabric :

- Python-based command-line orchestration tool
- Uses ssh and is highly scriptable

  http://www.fabfile.org/

Architecture : Puppetizing Nodes

Manually:

- Logging on to a server and installing puppet and its dependencies with your local package manager.
- Simple, controlled
- Not automated, diagnostic tool

'Baked in' :

- Adding the puppet call to your base image
- Simple, automated, excellent for mass rollout
- Awkward to change (depending on your build method)

'Kickstart' :

- Add the puppet call to your post-installation script
- Simple, automated
- You still need to get puppet on there first

```
puppet agent -t --server puppet --onetime
```

Architecture : Case Study 1

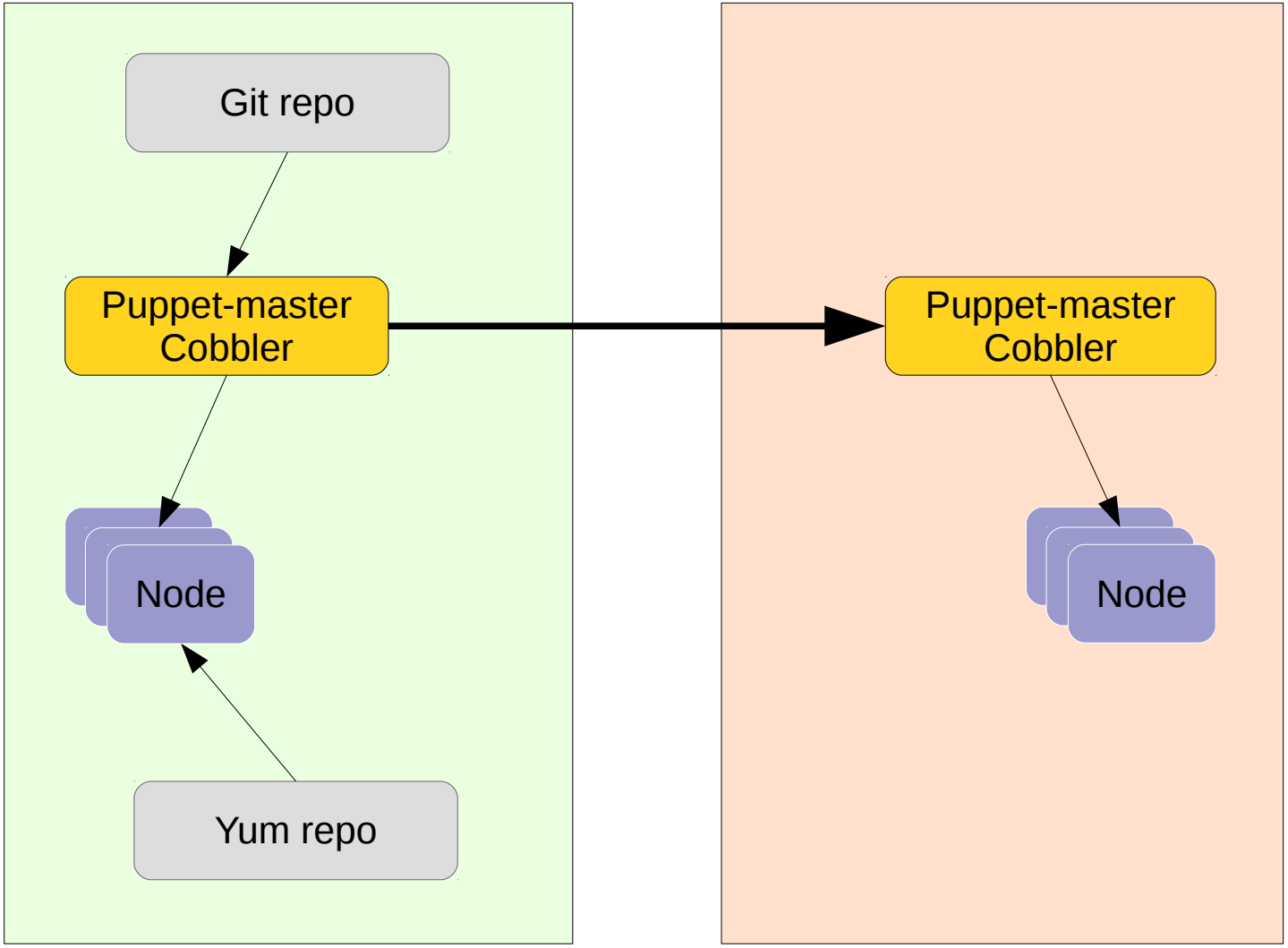
Site 1

- Combination Cobbler and Puppet server
- Private YUM repository
- Private GIT repository

Site 2

- Combination Cobbler and Puppet server, cloned from Site 1
- Bare-metal builds

Architecture : Case Study 1



Architecture : Case Study 2

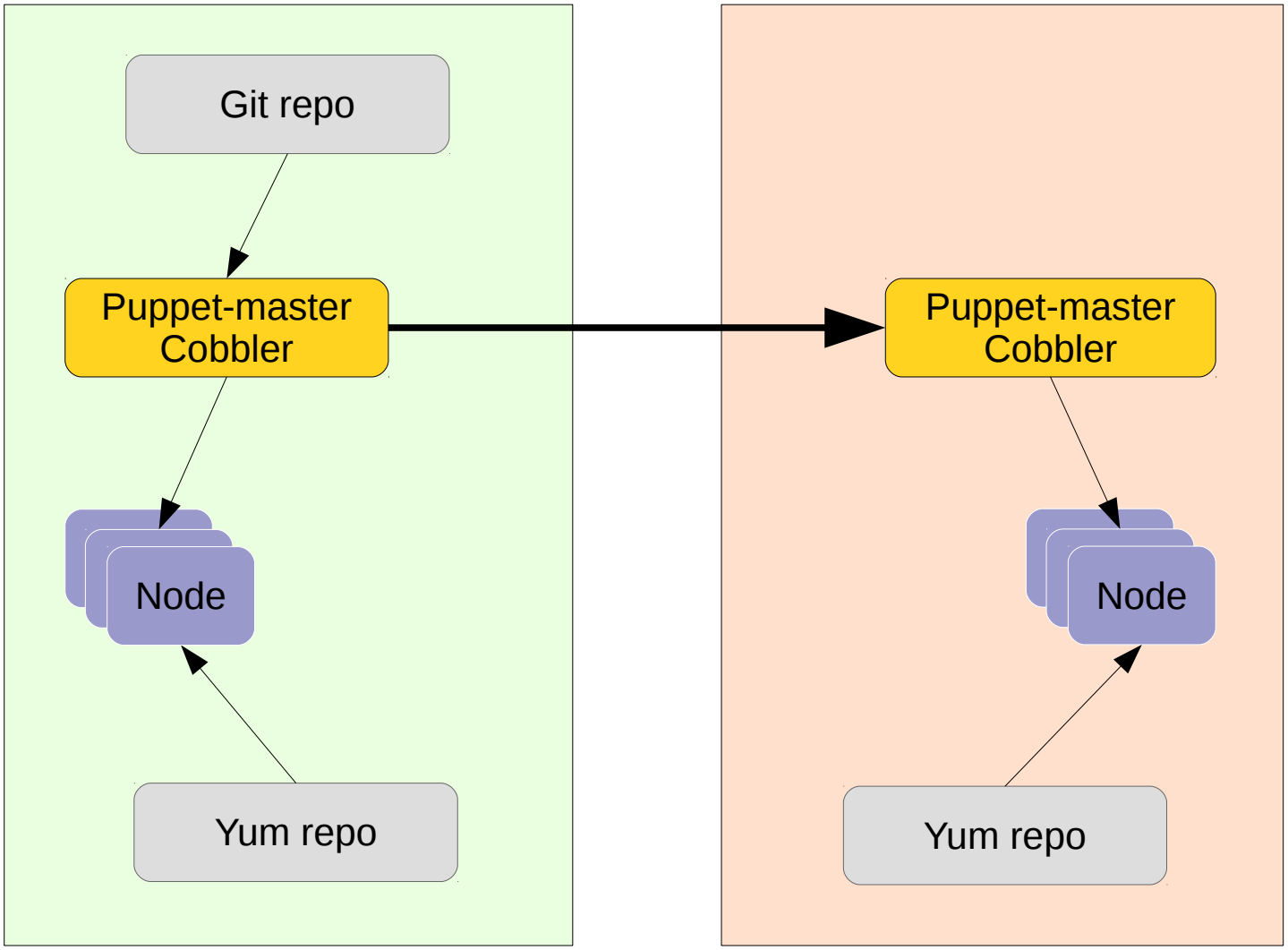
Site 1

- Combination Cobbler and Puppet server
- Private YUM repository
- Private GIT repository

Site 2

- Combination Cobbler and Puppet server built from Puppet-master in Site 1
- Bare-metal builds
- Private YUM repository

Architecture : Case Study 2



Architecture : Case Study

Adding a private repository :

The repo_manager module

/etc/puppet/manifests/site.pp

```
node default {
  Package {
    require => Class['repo_manager']
  }
  hiera_include('classes')
}
```

/etc/puppet/modules/repo_manager/manifests/init.pp

```
class repo_manager(
  $name     = 'dummy'
  $baseurl  = 'http://dummyvals'
  $desc     = 'dummy description'
  $enabled  = 0
  $gpgcheck = 0
 ) {
  yumrepo { "$name":
    baseurl => "$baseurl/$operatingsystem/$operatingsystemrelease/$architecture",,
    descr    => "$desc",
    enabled => $enabled,
    gpgcheck => $gpgcheck
  }
}
```

Architecture : Case Study

Adding a private repository : Hieradata

/etc/puppet/hieradata/

common.yaml
dev
prod-core
prod-remote
test

/etc/puppet/hieradata/prod-remote/common.yaml

```
repo_manager::name: 'remote_repo'
repo_manager::baseurl: 'http://yourrepoaddress/pub/centos/stable/'
repo_manager::desc: 'remote Private repository'
repo_manager:enabled: true
repo_manager::gpgcheck: false
```

/etc/puppet/hieradata/common.yaml

```
---
classes:
 - repo_manager
 - core_packages

repo_manager::name: 'private_repo'
repo_manager::baseurl: 'http://yourrepoaddress/pub/centos/stable/'
repo_manager::desc: 'Default Private repository'
repo_manager::enabled: true
repo_manager::gpgcheck: false
```

Architecture : Case Study

Strict Package ordering

```
class core_packages {
  # Local defaults
  Package { ensure => 'latest' }

  # Packages
  package { 'motd': }
  package { 'ntp': }
  package { 'rsync': }
  package { 'dnsmasq': }

 # Related services
  service { 'dnsmasq':
    ensure     => running,
    enable     => true,
    hasrestart => true,
    hasstatus  => true,
    require    => Package['dnsmasq']
  }
                                ...
  # Resource dependencies
  Package['motd'] -> Package['ntp']
  Package['rsync'] -> Package['ntp']
  Package['ntp'] -> Package['dnsmasq']
}
```

Architecture : Case Study

Bootstrapping puppet from Kickstart

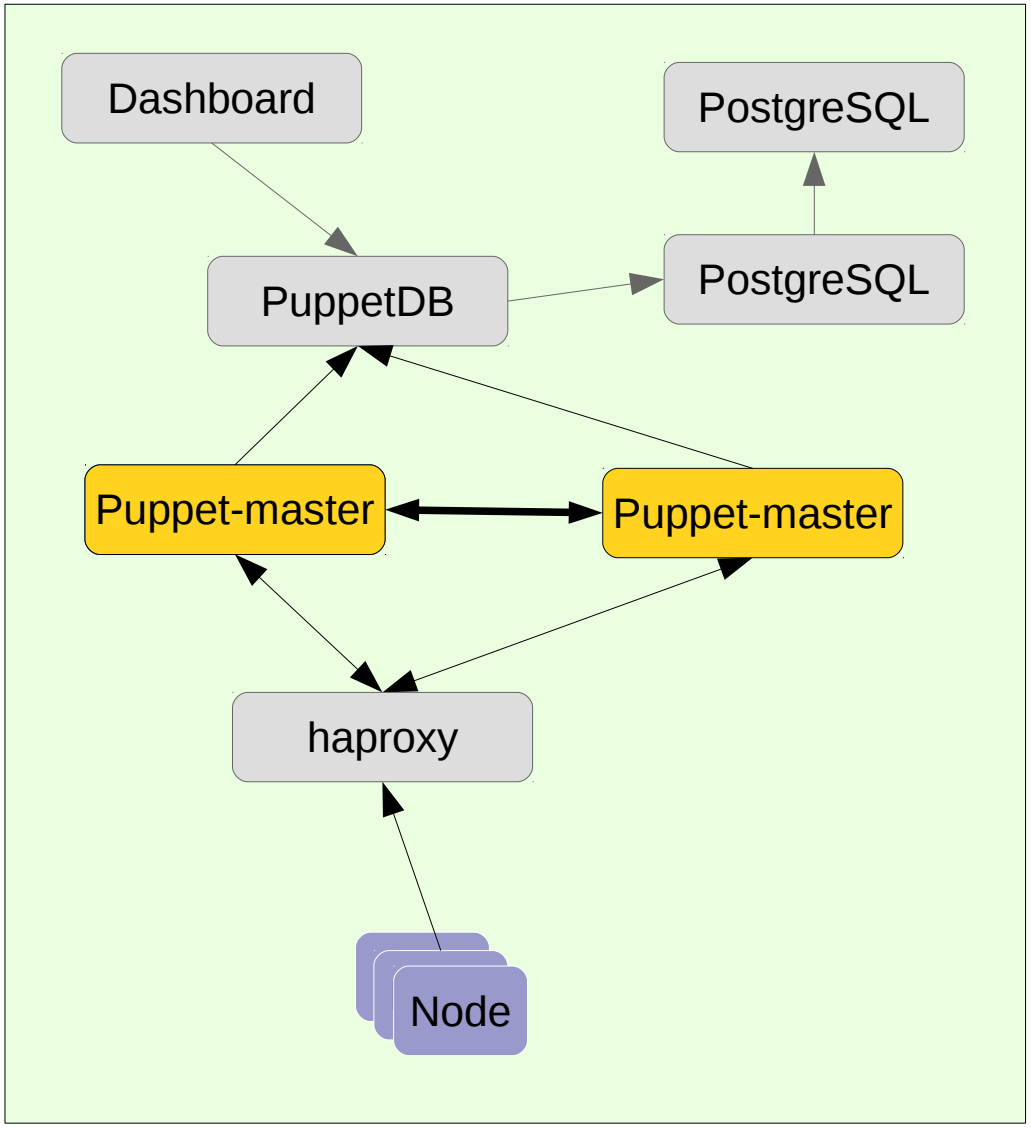
Snippet from end of kicksart script:

```
%post
#####################################################################
#
# Post Script - the following script runs on the newly
# installed machine, immediately after installation
#
#####################################################################

# Assumption: We have puppet installed as part of the server build

# Create the facts.d directory
mkdir -p /etc/facter/facts.d

# Add an environment fact
echo "environment: core-prod" >> /etc/facter/facts.d/base.yaml
echo "role: webserver" >> /etc/facter/facts.d/base.yaml

# Run against the cname of puppet - From here we will configure from the master
puppet agent -t --server puppet --onetime
```

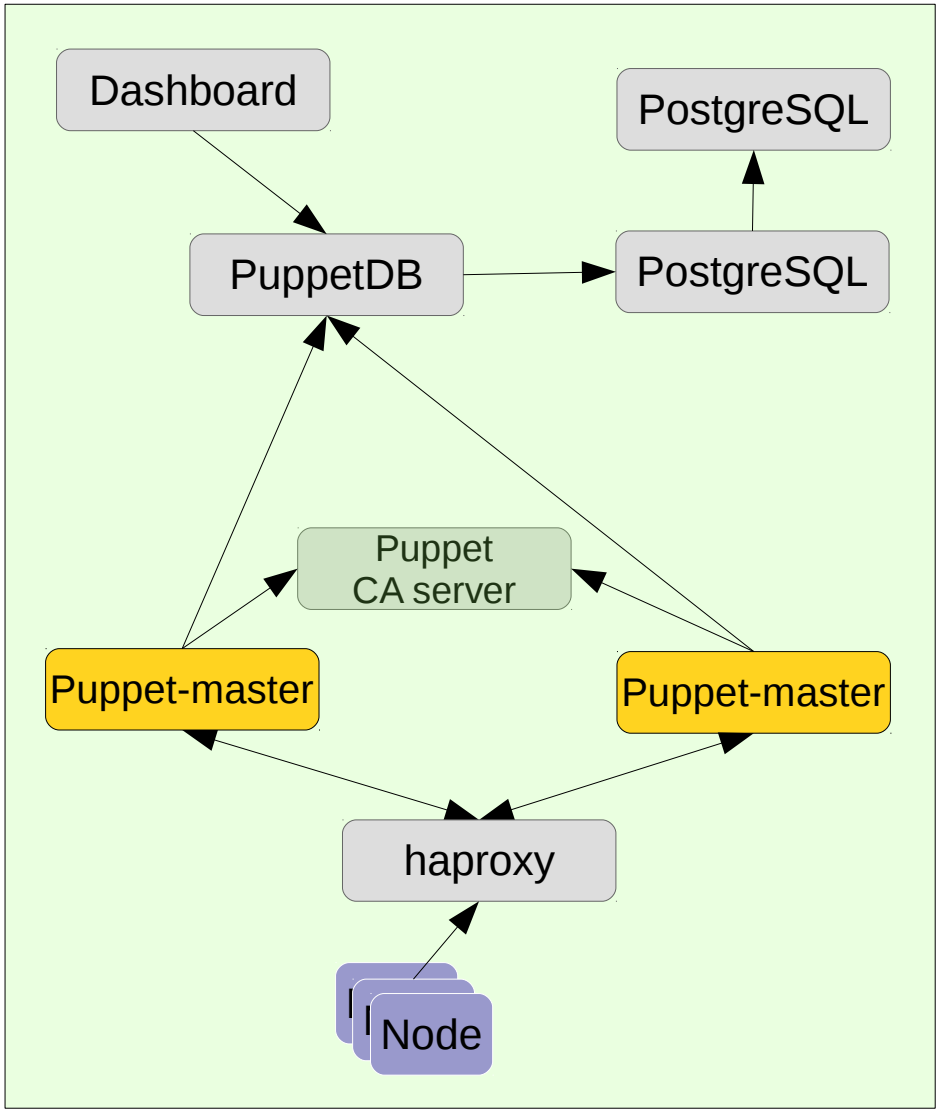
Architecture : Example Index

- Site Redundancy (with synced certificates)
- Site redundancy (with CA server)
- Masterless
- Multi-site replicated

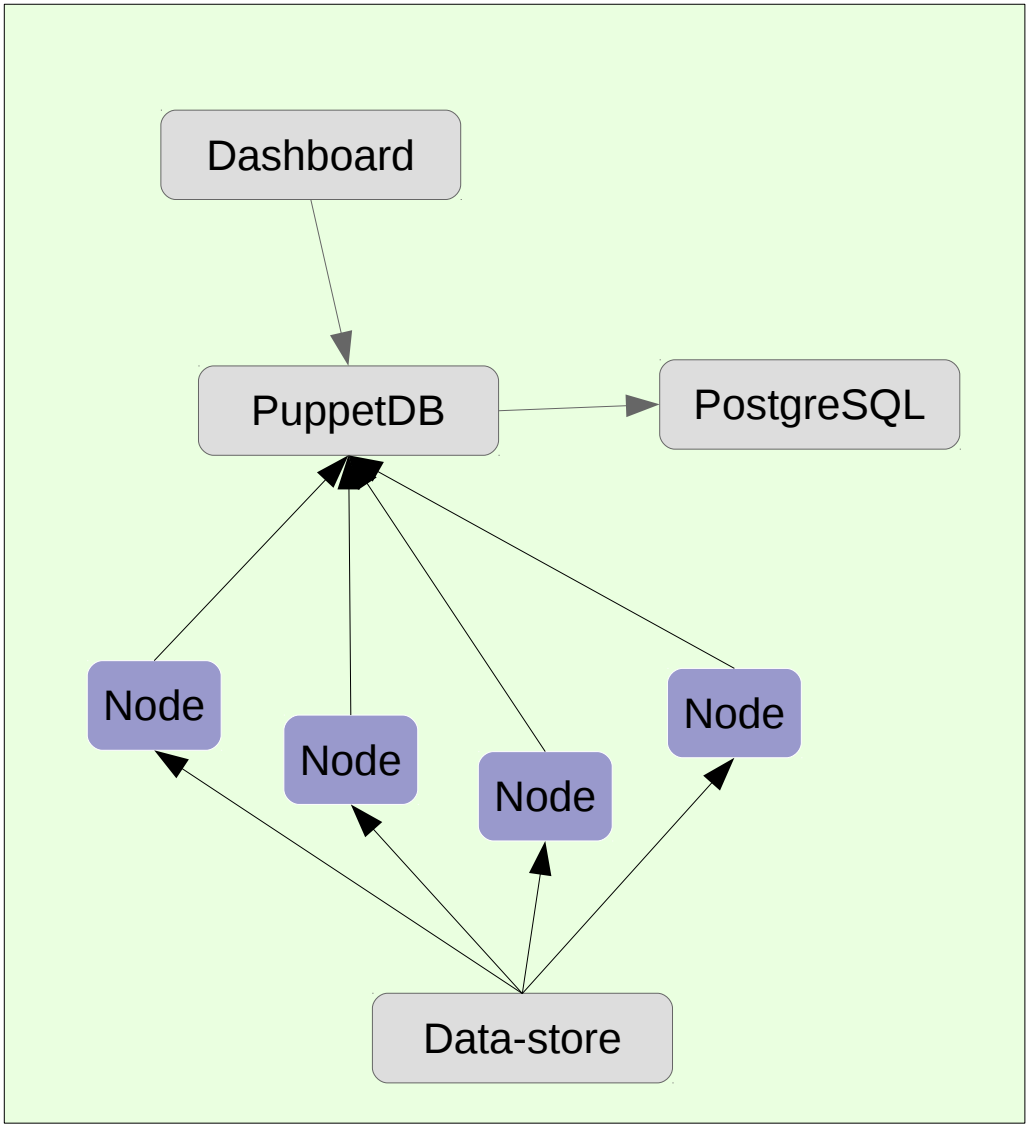
Architecture : Site Redundancy (sync)



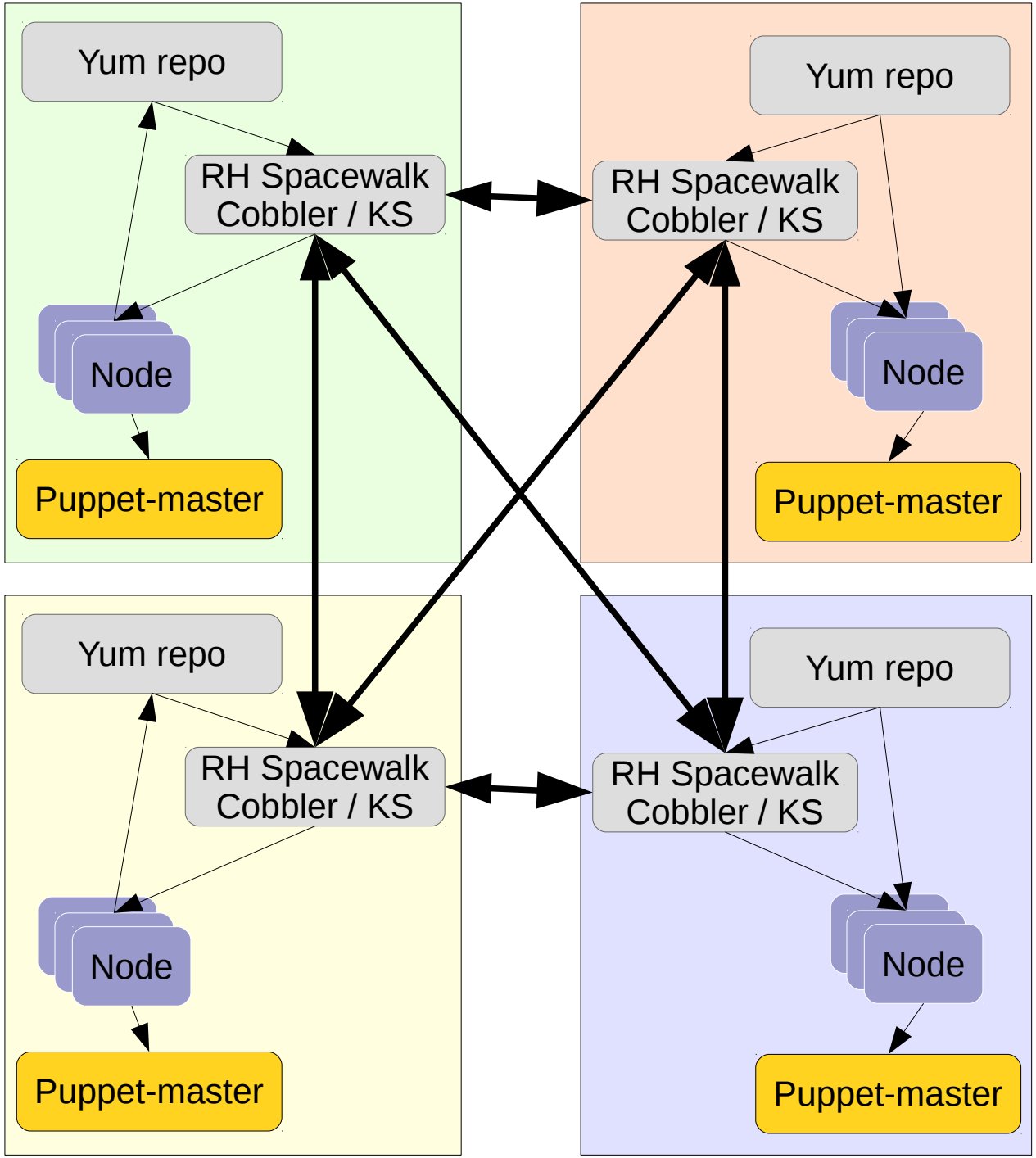
Architecture : Site Redundancy (CA)



Architecture : Masterless



Architecture : Multi-site replicated (Satellite)


Yum repo
RH Spacewalk
Cobbler / KS
Node
Puppet-master
Yum repo
RH Spacewalk
Cobbler / KS
Node
Puppet-master
Yum repo
RH Spacewalk
Cobbler / KS
Node
Puppet-master
Yum repo
RH Spacewalk
Cobbler / KS
Node
Puppet-master

Q & A

Thankyou

- Please fill in the feedback forms
- This course was brought to you by QA and NobleProg