System Provisioning in a Cloud Scale Environment

Scott Jaffa

Launch machine

```
$ onevm deploy 24
…

$ onevm list
 ID USER     GROUP    NAME            STAT UCPU    UMEM HOST             TIME
  8 oneadmin oneadmin CentOS Server 6 runn    0    1.3G node1      219d 01h25
 17 oneadmin oneadmin ttylinux1       runn    2    256M node1      218d 21h15
 18 oneadmin oneadmin ttylinux1       runn    2    256M node1      218d 21h15
 19 oneadmin oneadmin ttylinux12      runn    2    256M node1      218d 12h58
 24 oneadmin oneadmin ttylinux12      runn    2    256M node1      218d 12h58
 28 oneadmin oneadmin ttylinux12      pend    0      0K            218d 12h58

$ onevm show 24
…
ID  NETWORK          VLAN   BRIDGE      IP           MAC
0   Net1             no     br0         10.0.0.122   02:00:0a:00:00:7a
```

```
scott$ ssh root@10.0.0.122
root@10.0.0.122's password:

Chop wood, carry water.

# hostname
ttylinux_host
```

Agenda

- Introduction and background
- Definitions
- Cloud architecture
- Machine Lifecycle
- Tools
- Provisioning workflow
- Questions?

About Me

- Scott Jaffa
- Linux System Engineer
- scott@jaffafamily.org

About This Presentation

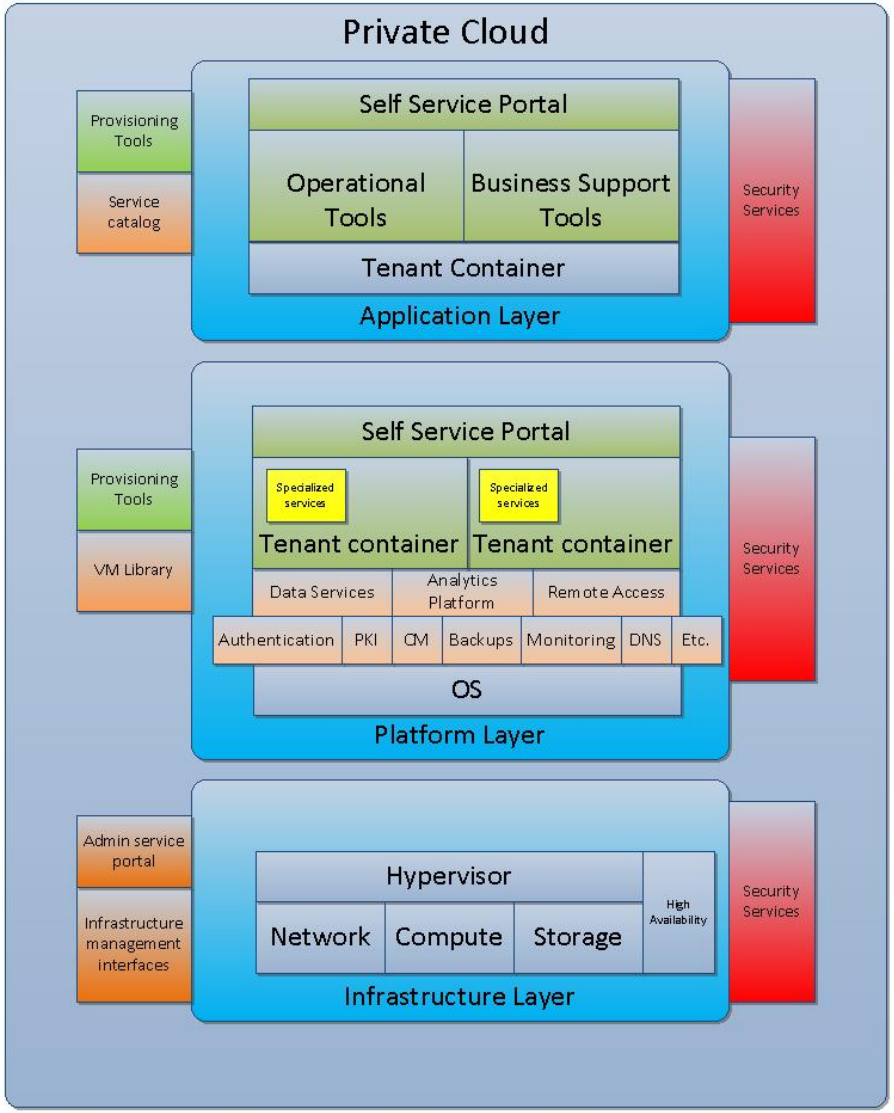
- Home Lab Project
- Problem – how do I provision machines such that they are efficiently managed?
- Objective – Create an environment where 100 potentially unique, fully configured, and operational machines can be launched within 15 minutes
- Be able to scale to many thousands of machines
- Assumes a functioning environment

Definitions

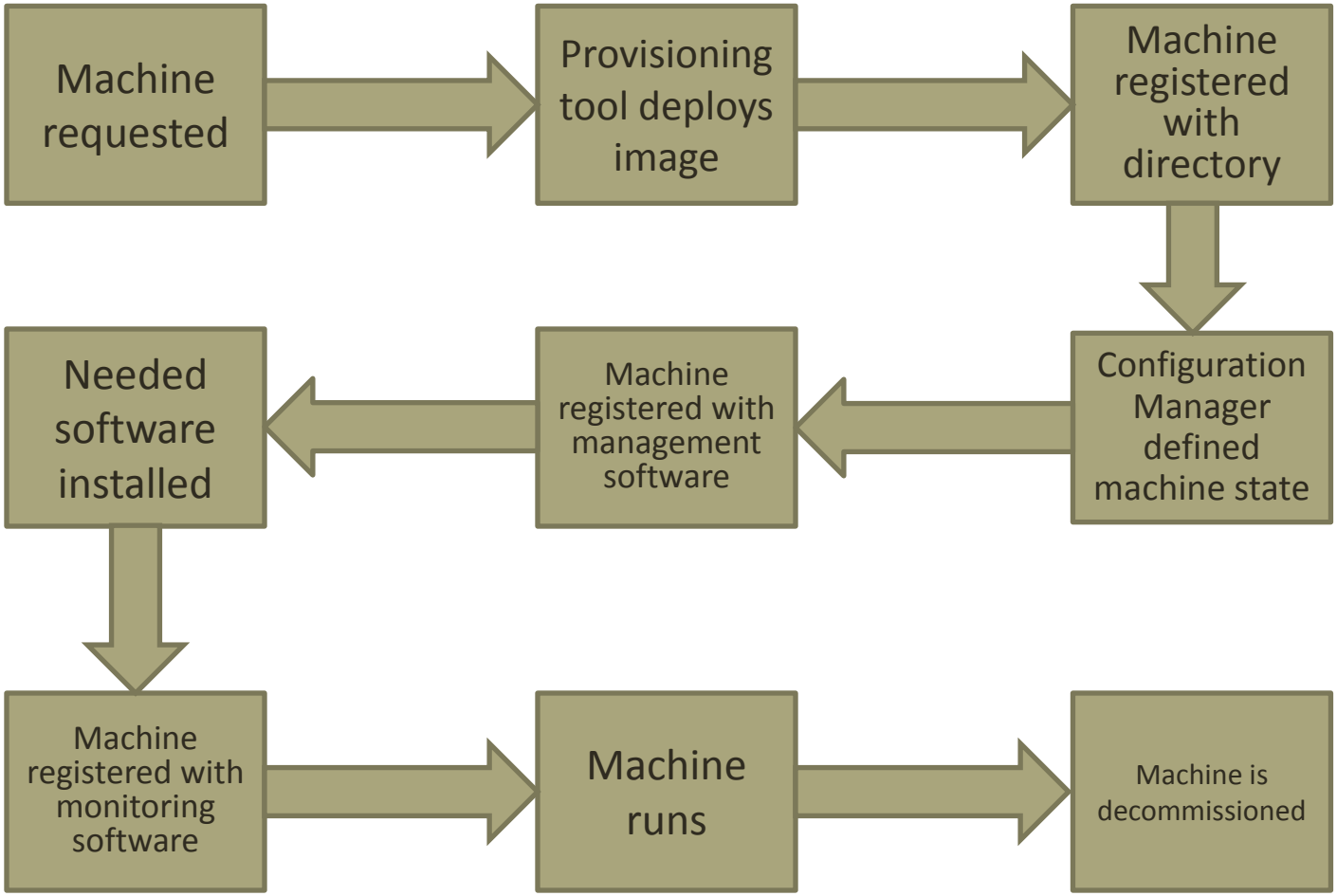
- Cloud Computing – On-demand self-service, broad network access, resource pooling, rapid elasticity, measured service - NIST
- Cloud Scale – Infrastructure too large to individually manage machines
- Virtualization – Virtualization is a platform which abstracts out the hardware

http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

Cloud

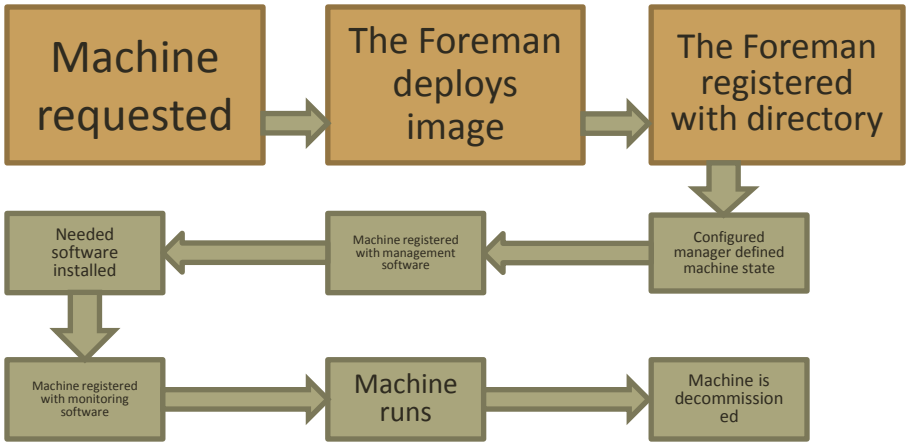

Private Cloud
Provisioning Tools
Service catalog
Self Service Portal
Operational Tools
Business Support Tools
Tenant Container
Application Layer
Security Services
Provisioning Tools
VM Library
Self Service Portal
Specialized services
Specialized services
Tenant container
Tenant container
Data Services
Analytics Platform
Remote Access
Authentication
PKI
CM
Backups
Monitoring
DNS
Etc.
OS
Platform Layer
Security Services
Admin service portal
Infrastructure management interfaces
Hypervisor
Network
Compute
Storage
High Availability
Infrastructure Layer
Security Services

Machine Lifecycle

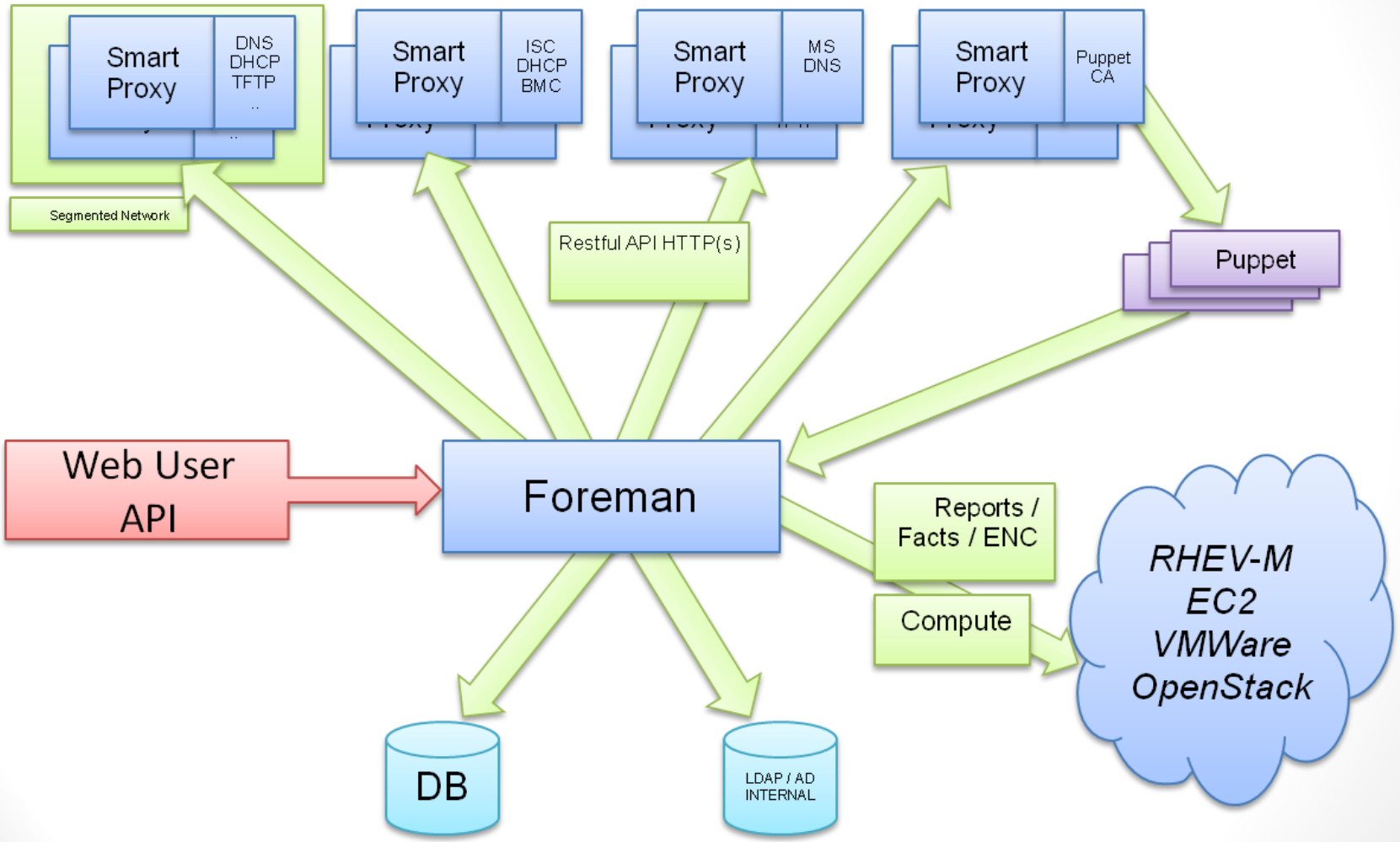


The Foreman

- Provisioning tool
- Provides front end CLI and GUI interface for user interaction
- Connects to the various management components



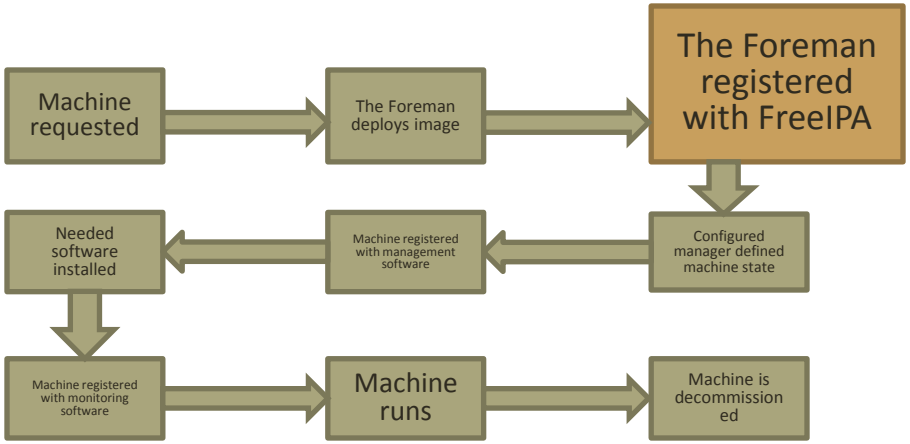
The Foreman



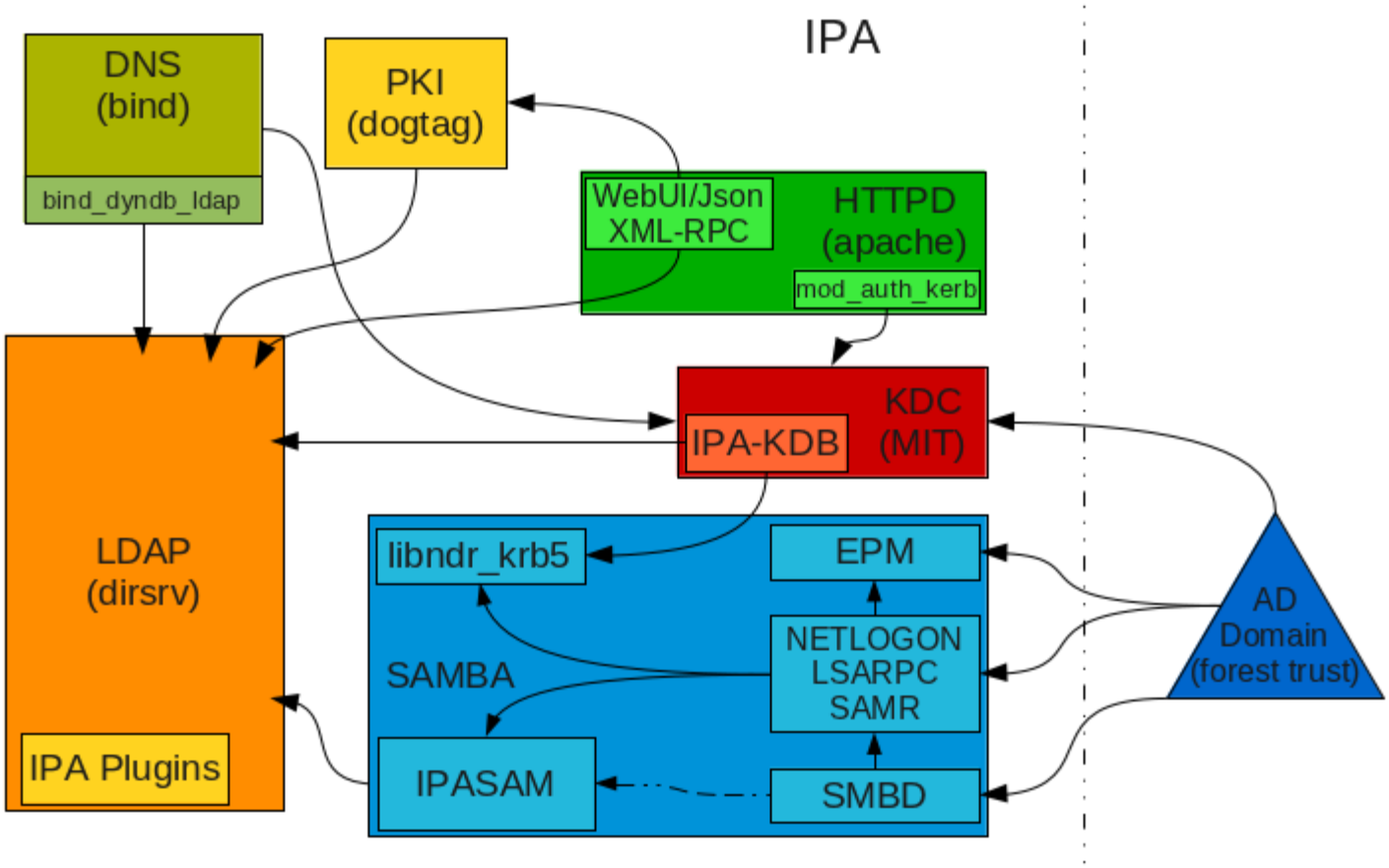
http://theforeman.org/manuals/1.1/index.html#ForemanArchitecture

FreeIPA

- Directory server (AD for Linux)
- Multi-master replication
- DNS server (through BIND LDAP connector)
- Kerberos
- Certificate Authority
- Web and command line interfaces



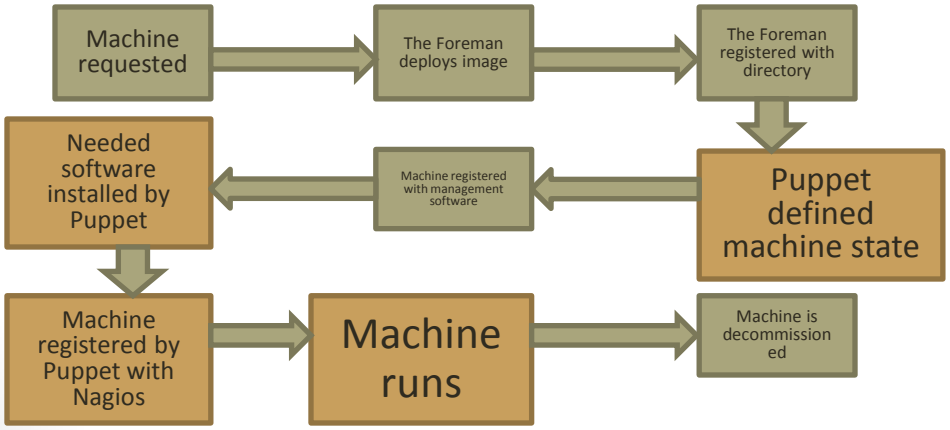
FreeIPA



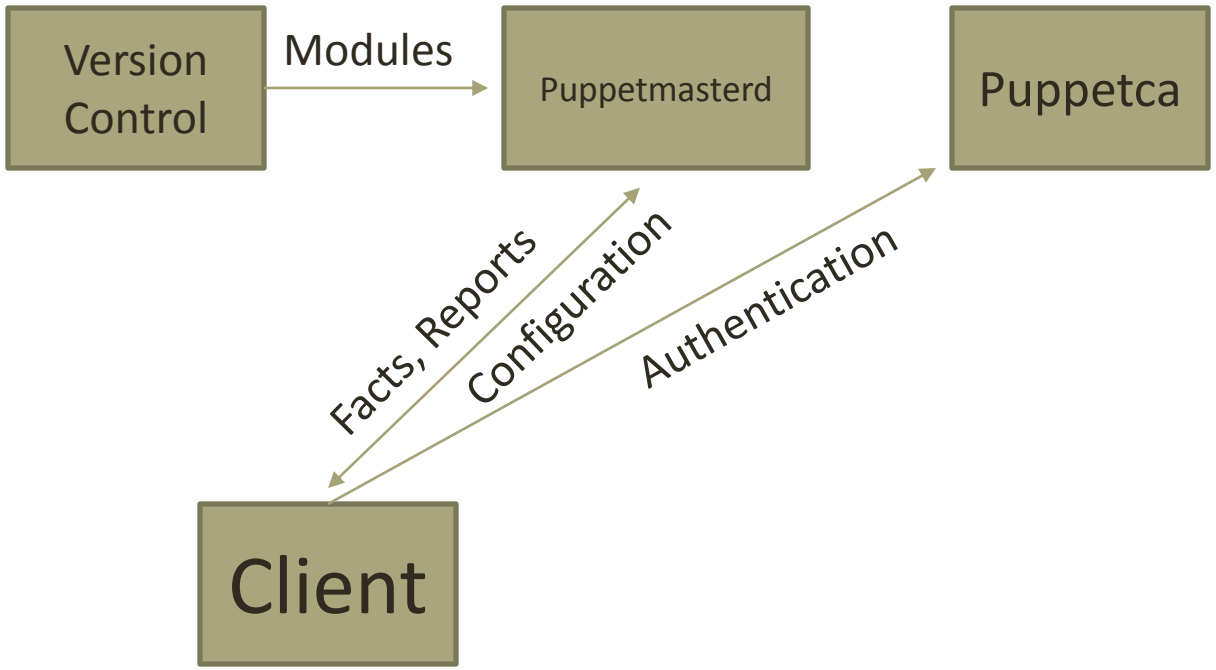
http://www.freeipa.org/page/IPAv3_Architecture

Puppet

- Lifecycle management engine
- Functionality
  - System configuration management
  - Change management
  - Disaster recovery
  - System configuration reporting

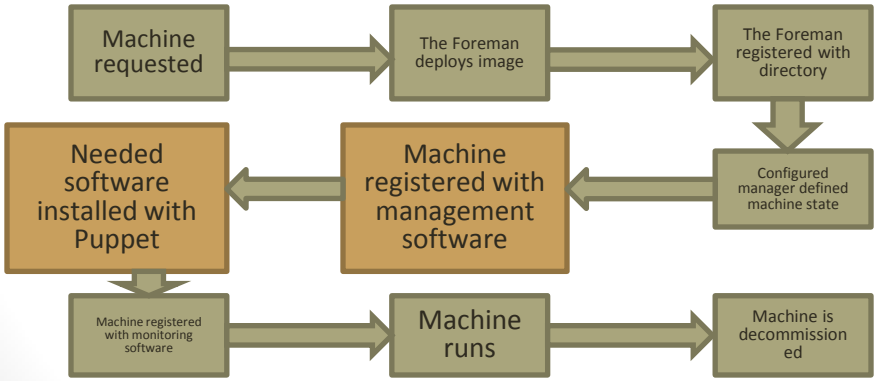


Puppet

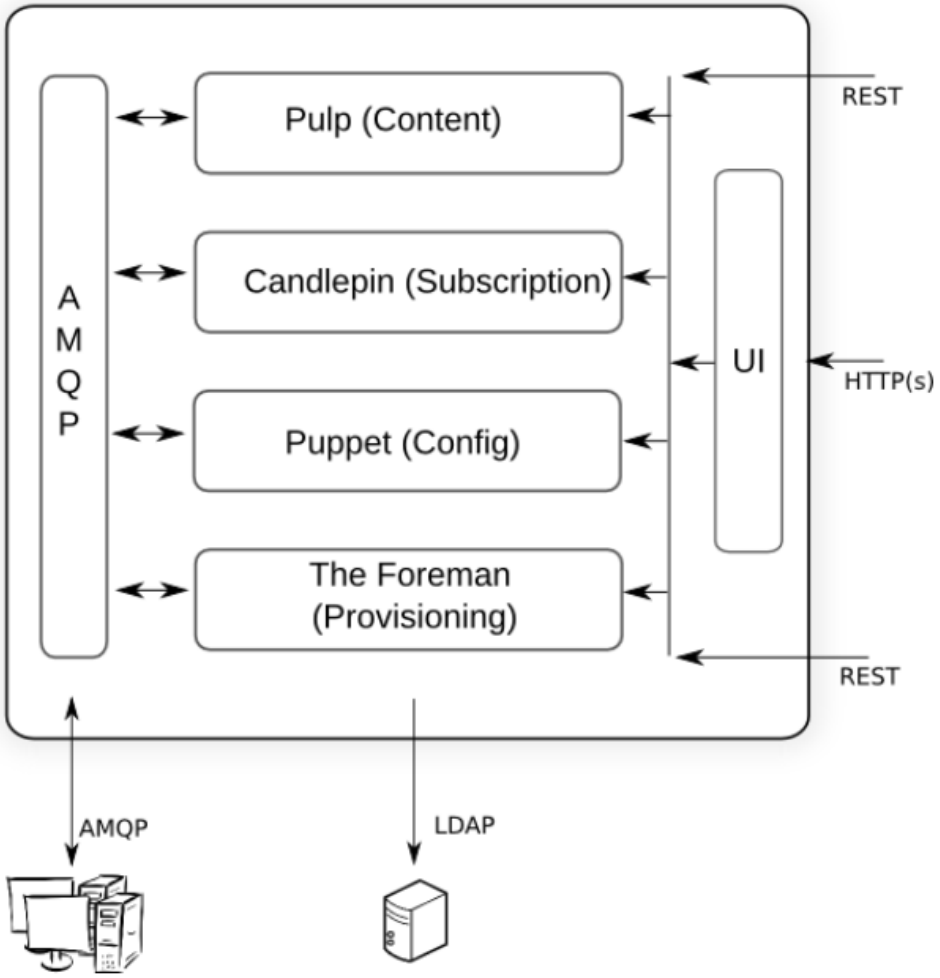

Version Control
Modules
Puppetmasterd
Puppetca
Facts, Reports
Configuration
Authentication
Client

Katello

- Wrapper software for system management
- Provides
  - Software repositories
  - Patching management
  - Provisioning
  - Configuration Management
- Environment aware

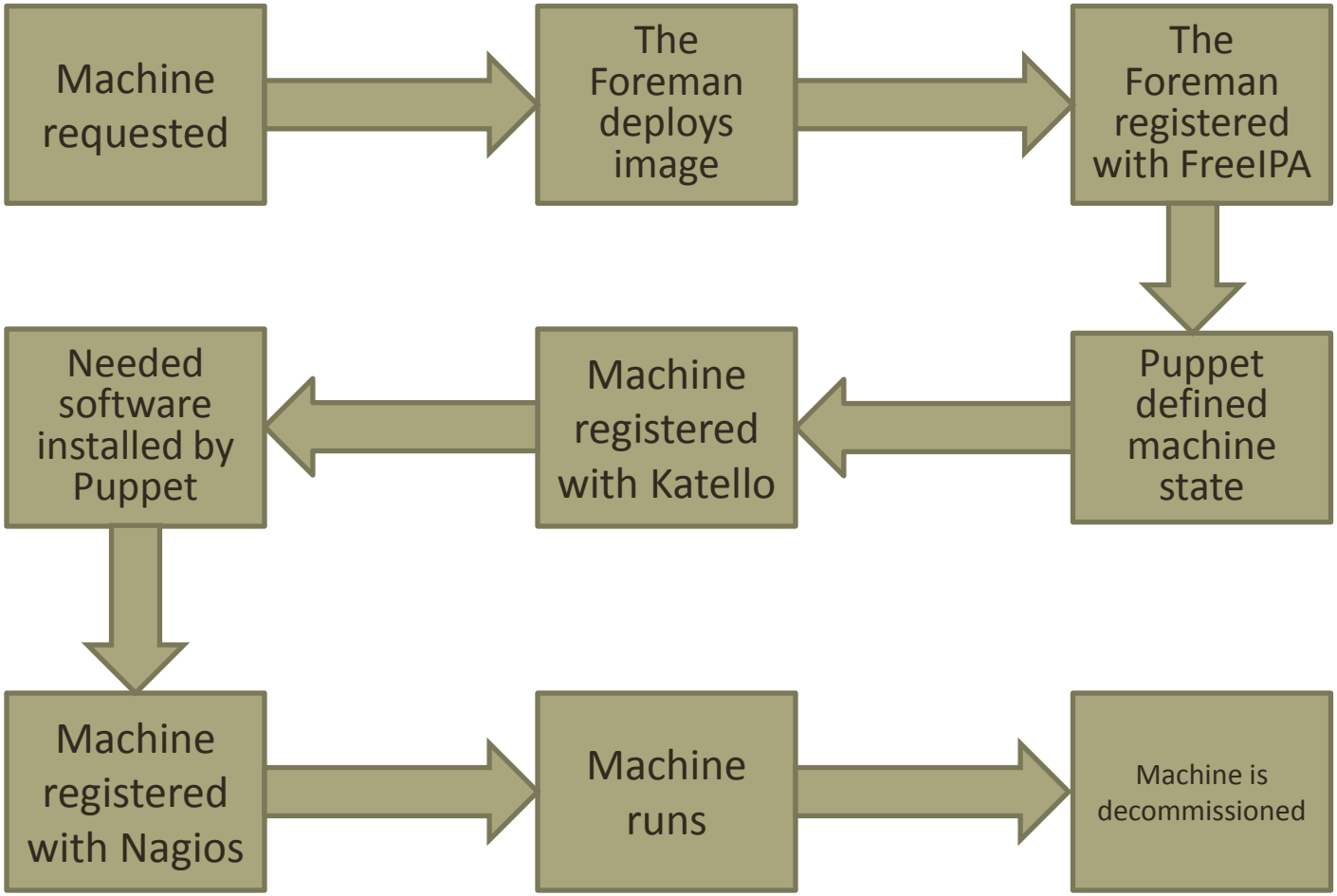


Katello



http://www.redhat.com/summit/2011/presentations/summit/whats_next/thursday/summit-2011.warner_sanders_t_1400_future_of_satellite-v7.pdf

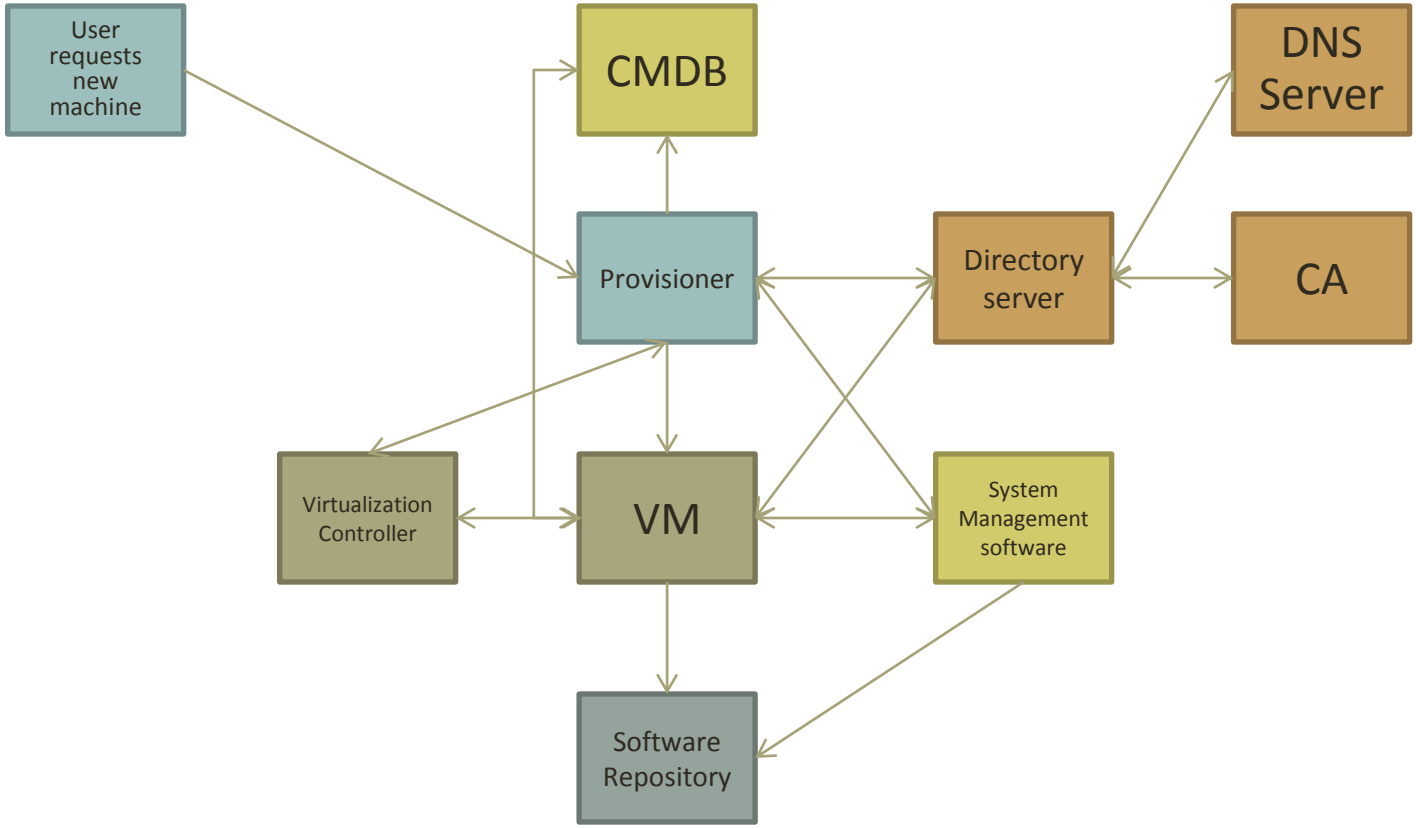
Machine Lifecycle



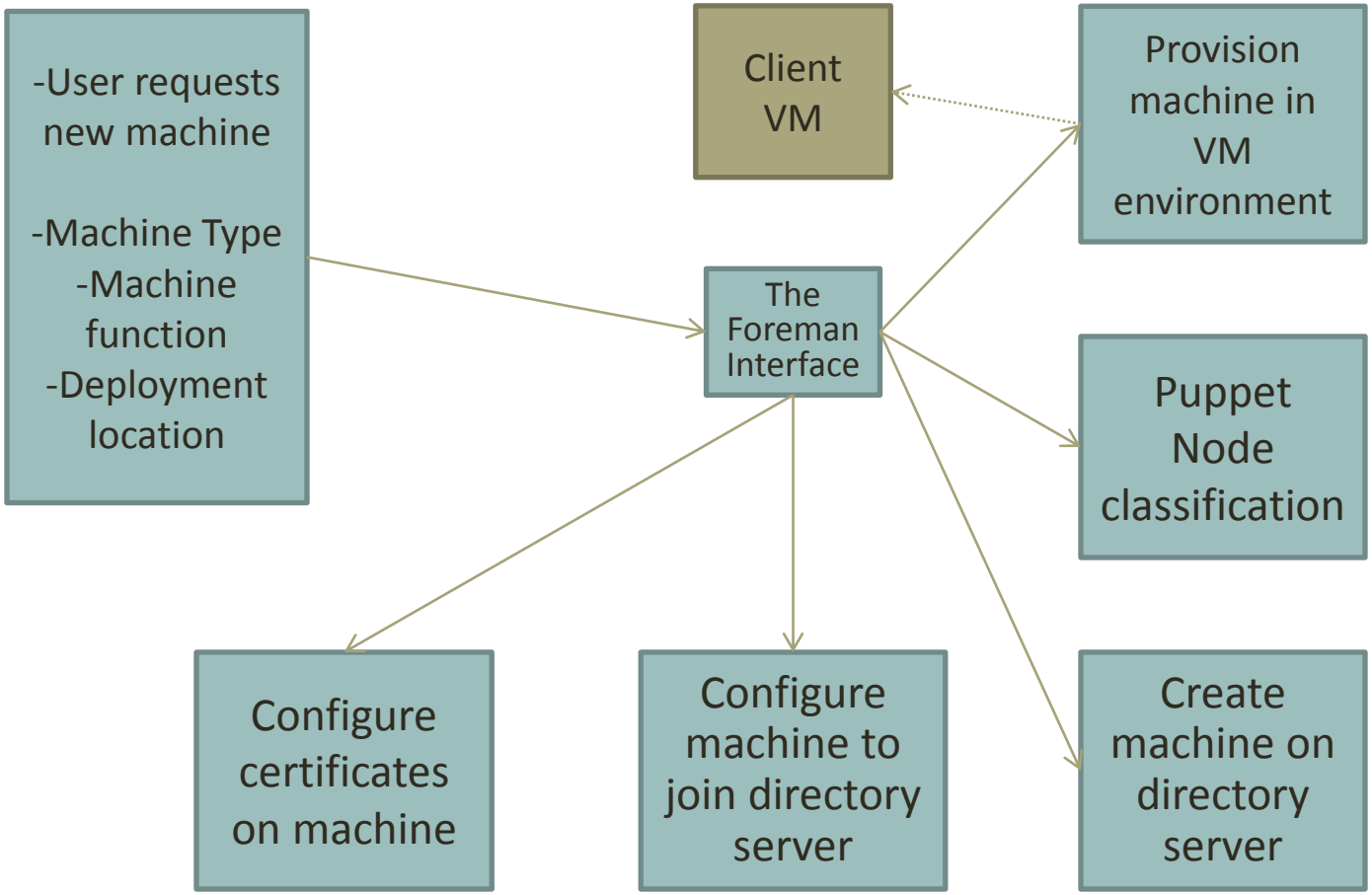
Provisioning Workflow Non Technical

- Process is transparent to requestor

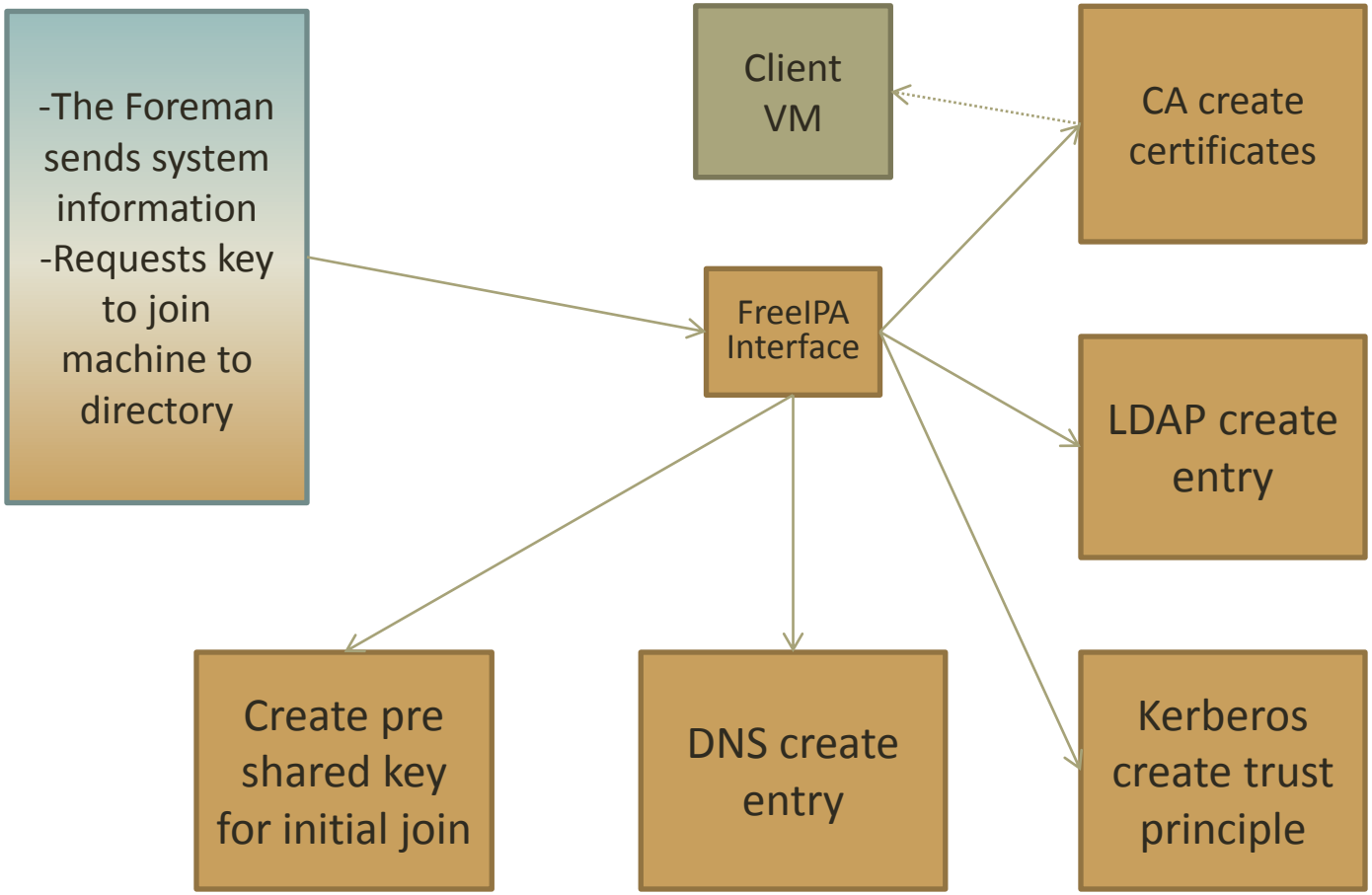
Provisioning Workflow Service Interactions



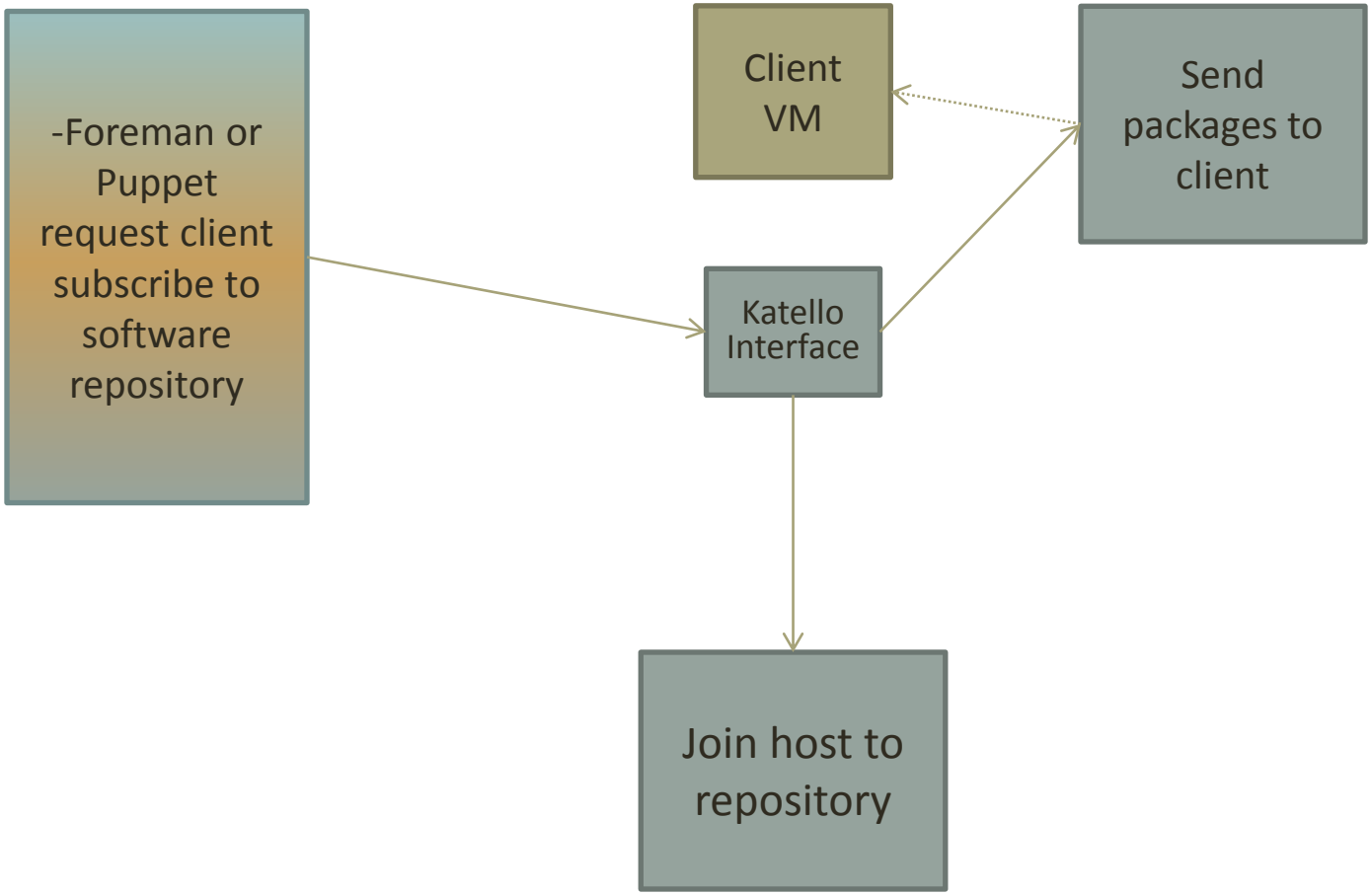
The Foreman


-User requests new machine
-Machine Type
-Machine function
-Deployment location
Client VM
Provision machine in VM environment
The Foreman Interface
Puppet Node classification
Configure certificates on machine
Configure machine to join directory server
Create machine on directory server

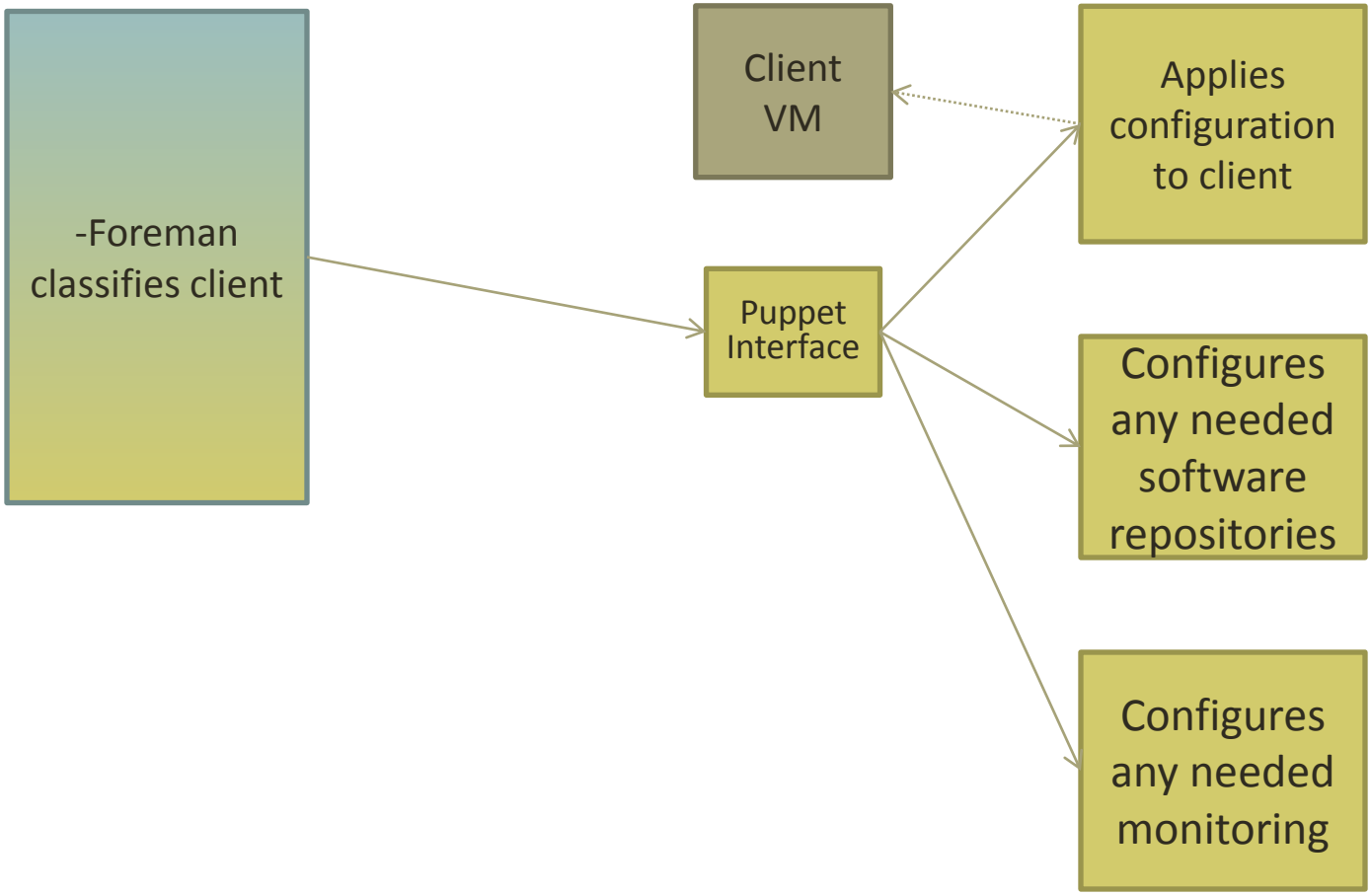
FreeIPA


-The Foreman sends system information
-Requests key to join machine to directory
Client VM
FreeIPA Interface
CA create certificates
LDAP create entry
Kerberos create trust principle
DNS create entry
Create pre shared key for initial join

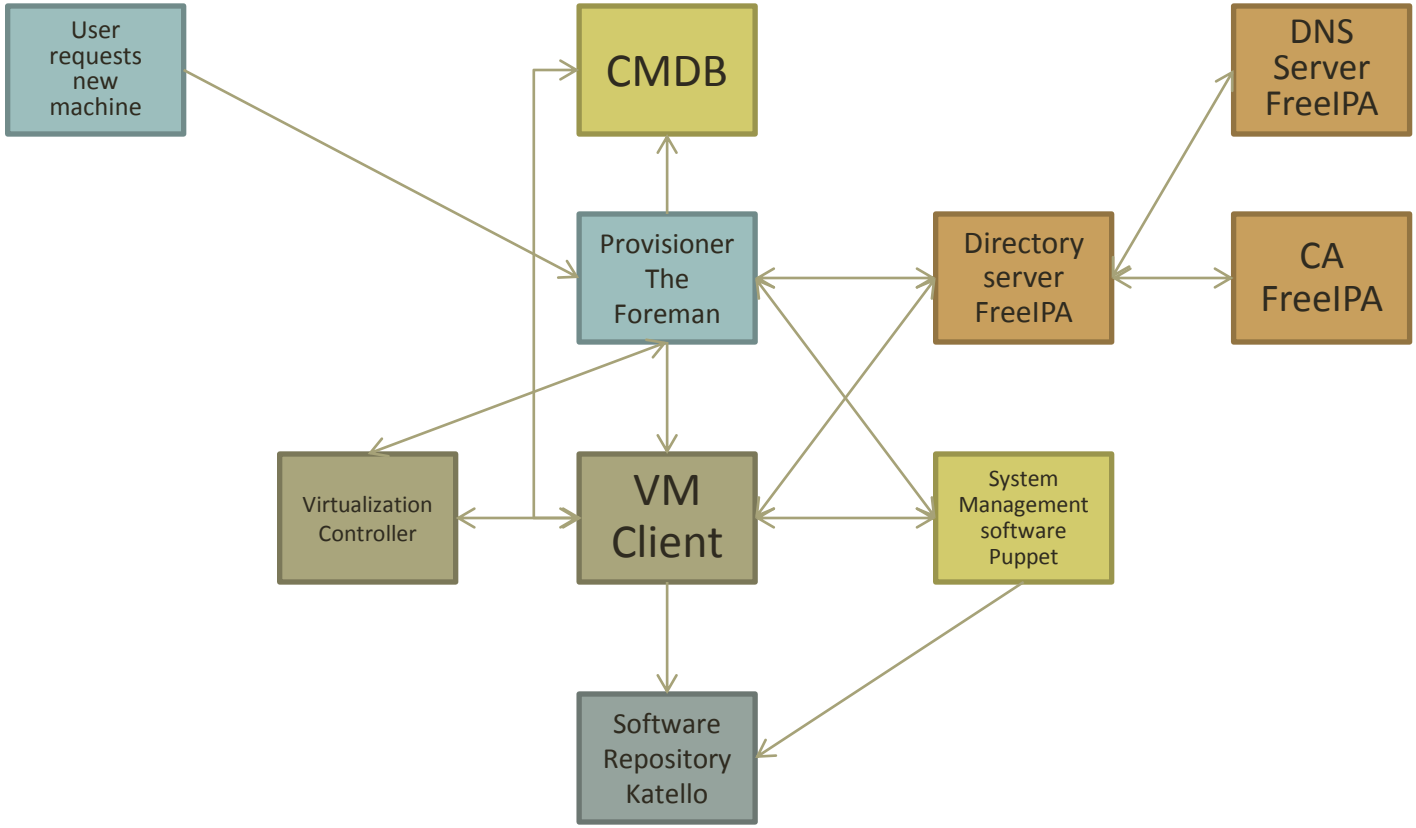
Katello


-Foreman or Puppet request client subscribe to software repository
Client VM
Send packages to client
Katello Interface
Join host to repository

Puppet


-Foreman classifies client
Client VM
Puppet Interface
Applies configuration to client
Configures any needed software repositories
Configures any needed monitoring

Provisioning Workflow Service Interactions



Advantages

- Security for free
- Known baseline
- Dev & test environments
- Disaster recovery
- Audit tracking
- Documentation
- Rapid updating

Questions?

- Scott Jaffa
- scott@jaffafamily.org

References

- NIST Cloud Computing definition http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf
- Foreman Architecture Diagram http://theforeman.org/manuals/1.1/index.html#ForemanArchitecture
- FreeIPA Architecture Diagram http://www.freeipa.org/page/IPAv3_Architecture
- Katello Architecture Diagram http://www.redhat.com/summit/2011/presentations/summit/whats_next/thursday/summit-2011.warner_sanders_t_1400_future_of_satellite-v7.pdf

Backup Slides

Virtualization Technologies

Related to Assumptions

- Compute – Basic hardware
- Storage – GlusterFS
- Network – VLANs
- Hypervisor – KVM
- Controller – OpenNebula
- High Availability – Gluster Replication and OpenNebula

Configuration Specifics – IPA PKI + Puppet

IPA Server:
# enrolling the puppet master service
$ ipa service-add puppetmaster/puppetmaster.example.com
# enrolling the puppet agent service
$ ipa service-add puppet/puppet.example.com

```
# install latest puppet-server
# (yum install puppet-server is a couple minor versions behind)
# version 3.2 fixes a CA bug that isn't in the yum repo
$ rpm -ivh http://yum.puppetlabs.com/fedora/f19/products/i386/puppetlabs-release-19-2.noarch.rpm
$ yum install -y http://yum.puppetlabs.com/fedora/f19/products/x86_64/puppet-server-3.2.4-1.fc19.noarch.rpm
# stop the puppetmaster service since we'll be using apache
$ service puppetmaster stop
# install additional requirements
$ yum install -y mod_nss mod_passenger
```

```
rm /etc/httpd/alias/*.db
certutil -d /etc/httpd/alias/ -N
chmod 644 /etc/httpd/alias/*.db

ipa-getcert request -r -K puppetmaster/puppet.lab.the-depths-of-hell.com -d /etc/httpd/alias -n
puppetmaster/puppet.lab.the-depths-of-hell.com

ipa-getcert request -K puppet/puppet.lab.the-depths-of-hell.com -D puppet.lab.the-depths-of-
hell.com -k /var/lib/puppet/ssl/private_keys/puppet.lab.the-depths-of-hell.com.pem -f
/var/lib/puppet/ssl/certs/puppet.lab.the-depths-of-hell.com.pem

certutil -K -d /etc/pki/nssdb -a
pk12util -o keys.p12 -n "IPA Machine Certificate - puppet.lab.the-depths-of-hell.com" -d /etc/pki/nssdb
openssl pkcs12 -in keys.p12 -out /var/lib/puppet/ssl/private_keys/puppet.lab.the-depths-of-
hell.com.pem -nodes

certutil -L -d /etc/pki/nssdb -a -n "IPA CA" > /var/lib/puppet/ssl/certs/ca.pem
certutil -d /alias -A -n "IPA CA" -t "CT,," -a -i /var/lib/puppet/ssl/certs/ca.pem
```

Puppet Agent setup

On the Puppet Agent:

1.Installation:

install latest puppet (agent) # (yum install puppet-server is a couple minor versions behind) # version 3.2 fixes a CA bug that isn't in the yum repo $ rpm -ivh http://yum.puppetlabs.com/fedora/f19/products/i386/puppetlabs-release-19-2.noarch.rpm $ yum install -y http://yum.puppetlabs.com/fedora/f19/products/x86_64/puppet-3.2.4-1.fc19.noarch.rpm

2.Setup certificates for the agent

$ ipa-getcert request -K puppet/puppet.example.com -D puppet.example.com -k /var/lib/puppet/ssl/private_keys/puppet.example.com.pem -f /var/lib/puppet/ssl/certs/puppet.example.com.pem

3.Setup the agent configuration in /etc/puppet/puppet.conf, by editing/adding the [agent] & [main] block:

[main] # <--snip--> server = 'puppetmaster.example.com' certname = 'puppetmaster.example.com' # <--snip--> [agent] # <--snip--> certificate_revocation = false certname = 'puppet.example.com' # <--snip-->

4.Test the entire setup in puppet agent:

open up port for Puppet $ firewall-cmd --add-port=8140/tcp # test to see if the setup works $ puppet agent --test # you'll probably get a catalog error if you have no catalogs # setup with your puppet master