

Infrastructure Automation with



email: bill.hathaway@versatile.com
twitter: @billhathaway

whoami

- Consultant with Versatile since 1995
- ~ 100 clients
- Long-term interest in automation
- Puppet user for 8 months



What is Puppet?

Model-driven configuration management

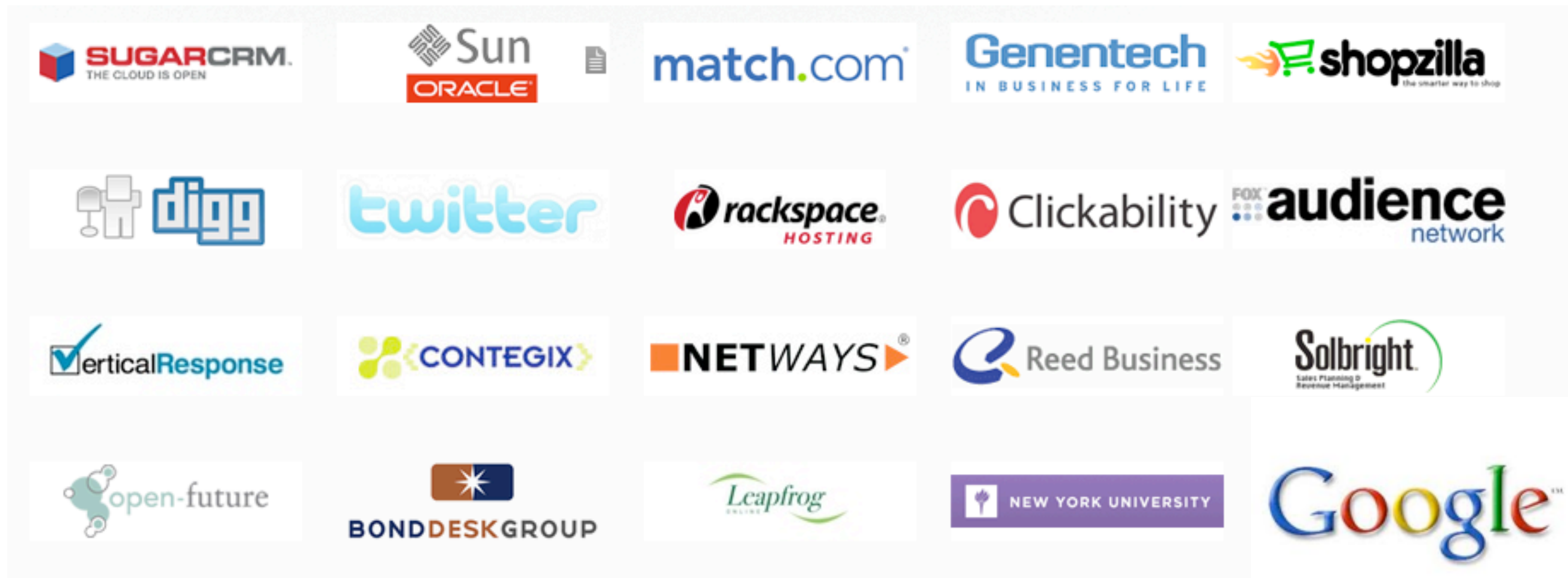
- Automated
- Repeatable
- Scalable

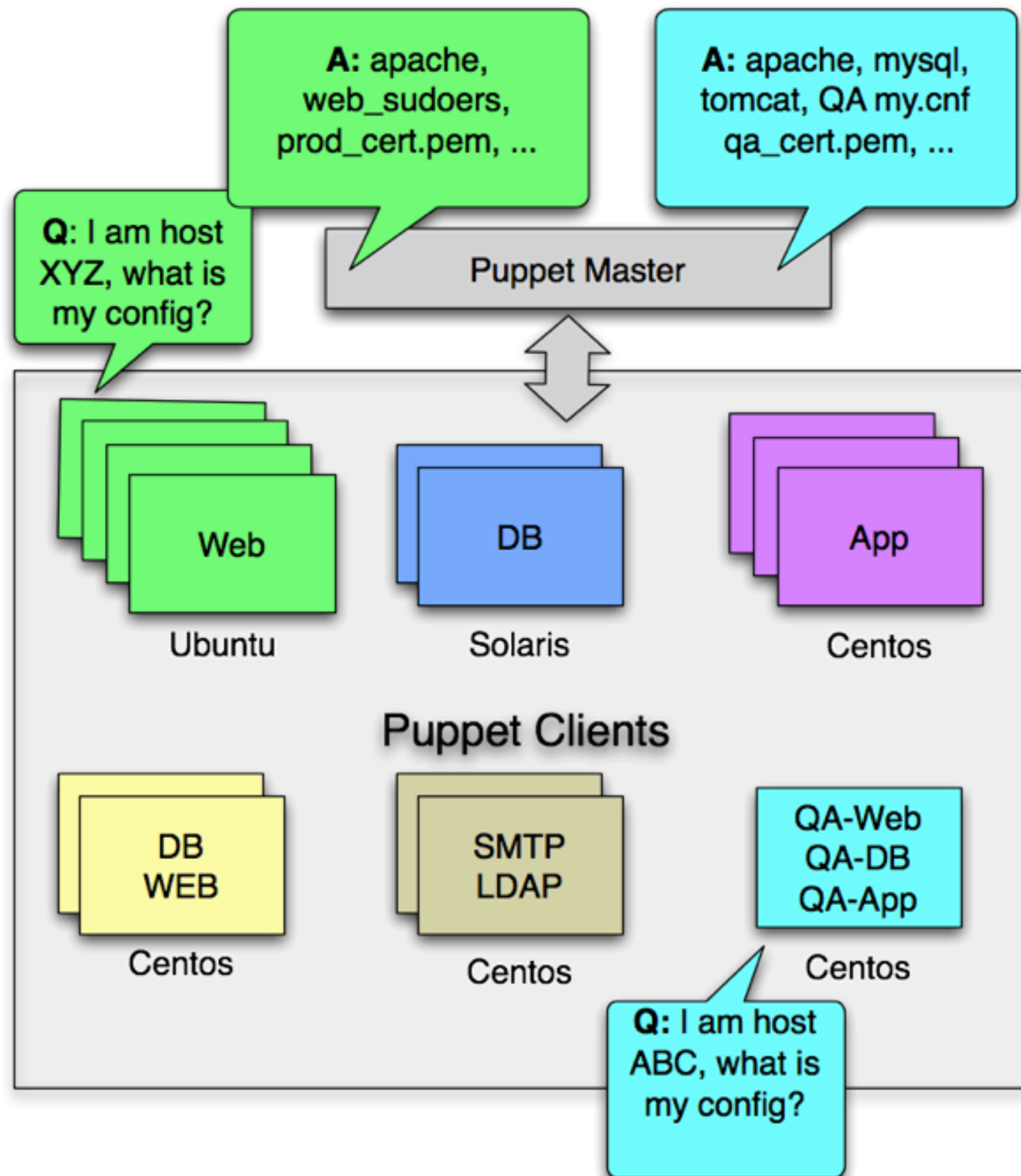


What is Puppet?

- Client-server model
- SSL communication
- UNIX centric
- Extensible

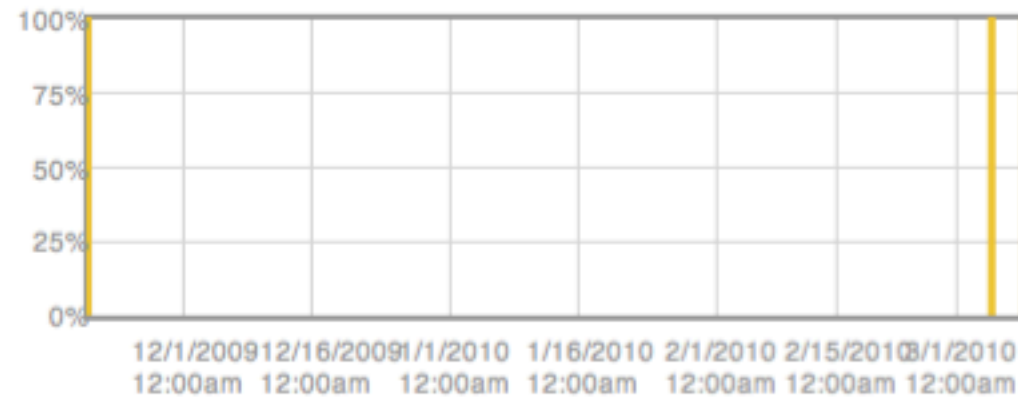
Argument from Authority



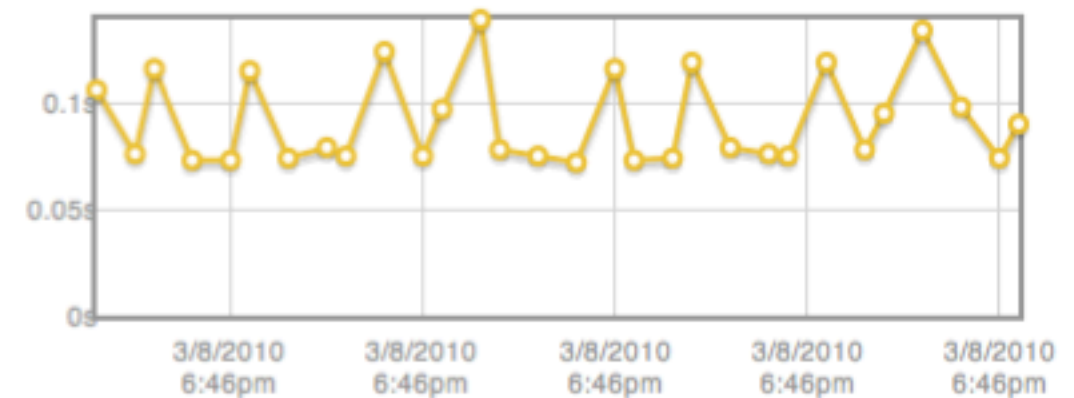


Dashboard

Run Success



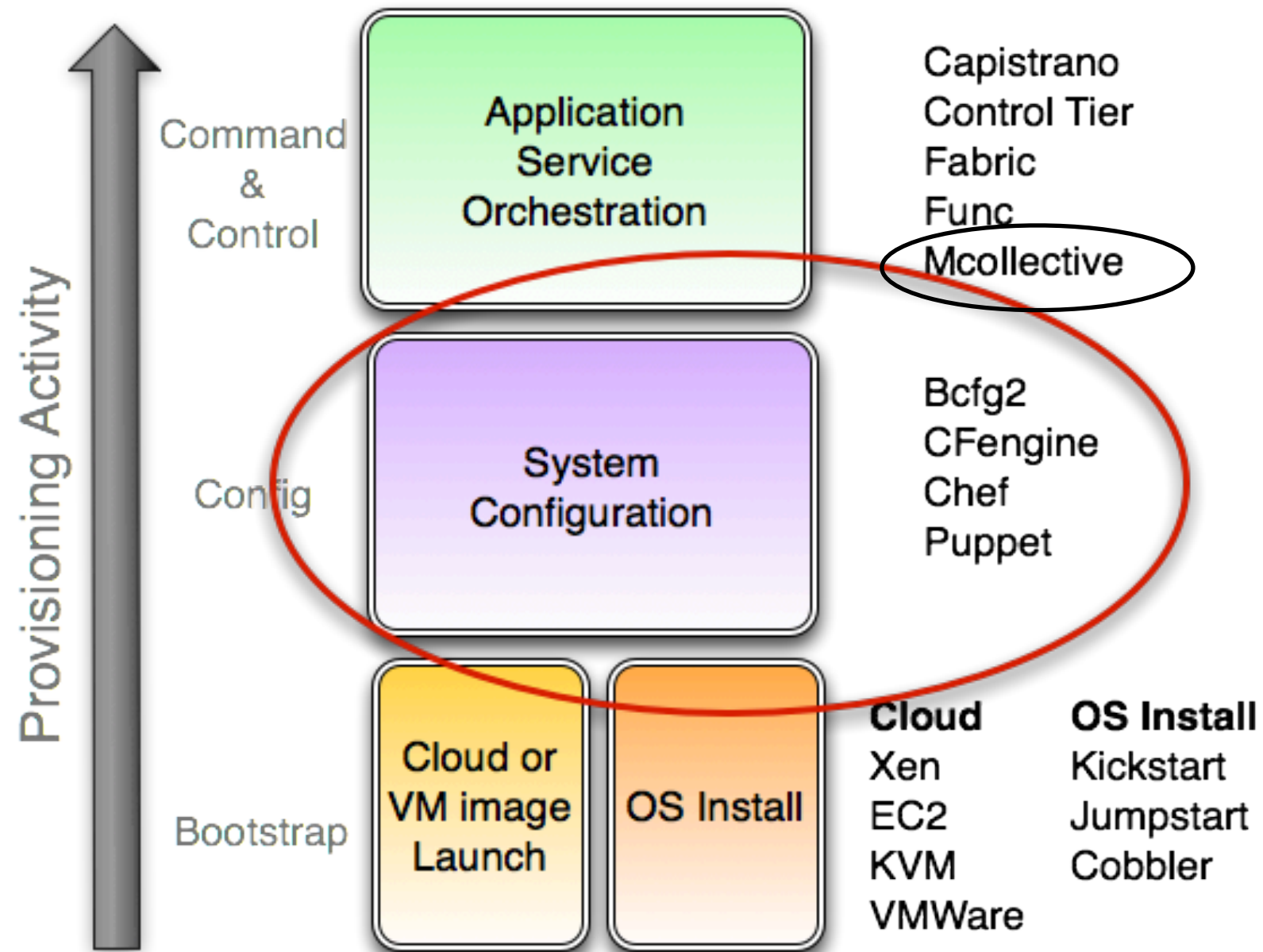
Run Time



Recent Node Reports

<input type="checkbox"/>	Hostname	↓ Latest Report
<input type="checkbox"/>	● boy_willy.puppetlabs.com	March 08, 2010 18:46
<input type="checkbox"/>	● mad_hamish.puppetlabs.com	March 08, 2010 18:46
<input type="checkbox"/>	● gaspode.puppetlabs.com	March 08, 2010 18:45
<input type="checkbox"/>	● errol.puppetlabs.com	March 08, 2010 18:45
<input type="checkbox"/>	● reg_shoe.puppetlabs.com	March 08, 2010 18:45
<input type="checkbox"/>	● nobby.puppetlabs.com	March 08, 2010 18:45
<input type="checkbox"/>	● sergeant_colon.puppetlabs.com	March 08, 2010 18:45
<input type="checkbox"/>	● detritus.puppetlabs.com	March 08, 2010 18:45
<input type="checkbox"/>	● littlebottom.puppetlabs.com	March 08, 2010 18:45
<input type="checkbox"/>	● angua.puppetlabs.com	March 08, 2010 18:45

Positioning





Versions



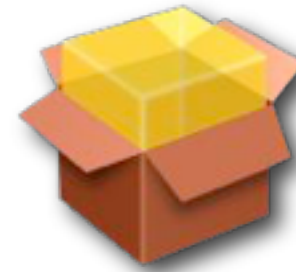
0.23 → 0.24 → 0.25 → 2.6
2007 2008 2009 2010

Terminology

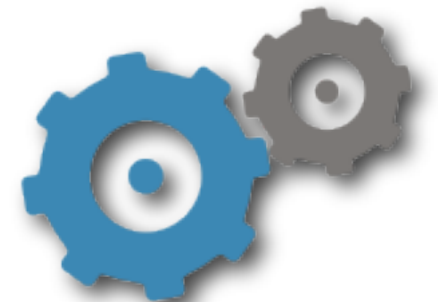
Resources - high level representations of system objects

Attributes - properties specific to resource types

Providers - implement logic for specific backend systems



package



service



files/
directories



users/
groups



cron



Sample resource

```
file {  
  “/etc/motd”:  
    mode => 644,  
    owner => root,  
    group => root,  
}
```

Sample resources

```
package { "ntp":  
  ensure => latest,  
}
```

```
service { "httpd":  
  ensure => running,  
  enable => true,  
}
```

Sample resources

```
package { ssh:  
  name => $operating systems ? {  
    debian => "ssh",  
    default => "sshd"  
  },  
  ensure => installed  
}
```

Managing file content

multiple ways

- Static files
- ERB templates
- Augeas

Terminology

Class - logical collection of resources

```
class mysql-server {  
  package { "mysql-server":  
    ensure => latest,  
  }  
  
  file { "/etc/my.cnf":  
    require => Package["mysql-server"],  
    source  => puppet:///files/mysql/my.cnf  
    mode    => 644,  
    owner   => mysql  
  }  
  
  service { "mysql":  
    ensure  => running,  
    require => File["/etc/my.cnf"],  
  }  
  
}
```

```
class mysql-server {  
  package { "mysql-server":  
    ensure => latest,  
  }  
  if $sp_number_processors > 2 {  
    $cnf_file = "my.cnf.big"  
  } else {  
    $cnf_file = "my.cnf"  
  }  
}  
  
file { "/etc/my.cnf":  
  require => Package["mysql-server"],  
  source  => /some/repo/$cnf_file,  
  mode    => 644,  
  owner   => mysql  
}  
}
```

Terminology

Node - a client with one or more classes

Sample nodes

```
node "frodo.foo.com" {  
  include "mysql-server"  
  include "ldap-nameservice"  
  include "site-config"  
}
```

```
node /web/ {  
  include "apache"  
  include "site-config"  
}
```

```
node default {  
  include "site-config"  
}
```

Main Components

Client

facter - system “facts”

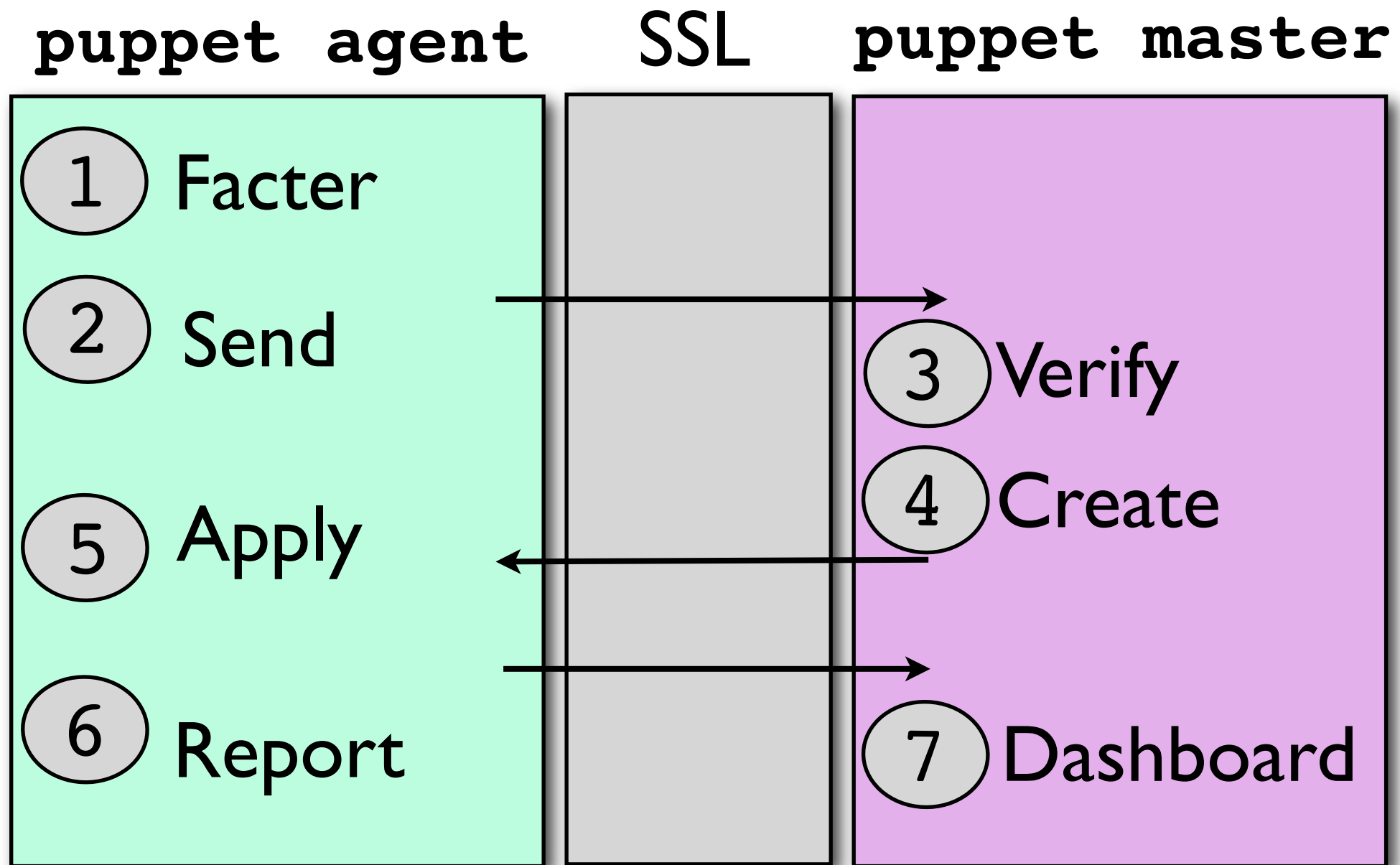
puppet agent - networked persistent client

Server

puppet master - builds/serves configs to clients

puppet cert - client certificate issue/revoke

Basic Puppet Flow



bill@smash: # facter

```
architecture => x86_64
domain => vm.testlab.wdh
fqdn => smash.vm.testlab.wdh
hardwaremodel => x86_64
hostname => smash
interfaces => eth0
ipaddress_eth0 => 172.16.60.131
is_virtual => true
operatingsystem => Ubuntu
kernelmajversion => 2.6
memorysize => 783.98 MB
selinux => false
```


#puppet agent

daemon running on clients

- Bootstraps from `/etc/puppet/puppet.conf`
 - Retrieves config from server
 - Applies config
 - Reports results to server
 - Sleep 1800 && repeat

#puppet master

central management daemon

- Authenticates clients
- Generates client-specific configs
- Serves files
- Receives reports from clients

#puppet cert

certificate administration interface

- Lists outstanding requests
- Approves or rejects requests
- Revokes de-commissioned clients

```
#puppet cert --sign $newclient
```

Quick Start

1. Install Puppet master on a server
2. Define a base configuration
3. Install Puppet on clients
4. Test

Base config on server

- Puppet files traditionally end in **.pp**
- Main file - `/etc/puppet/manifests/site.pp`
- Files can load other files

```
# cat /etc/puppet/manifests/site.pp
class base {
  file {
    “/tmp/test”:
      ensure => present,
      mode   => 644,
      owner  => root;
  }
}
node default {
  include test
}

# puppet master
```

Using a client

client# rpm -i puppet

client# vi /etc/puppet/puppet.conf

client# puppet agent --no-daemonize

server# puppet cert --list

server# puppet cert --sign \$client

Implementing Puppet

- Start simple
- Use version control
- Start with harmless resources
- Use 'noop' to see what would happen

Resources

- “Pulling Strings with Puppet”
- Puppet Users mailing list
- #puppet on irc.freenode.net
- Planet Puppet blog aggregator

Q & A

Misc Trivia

- Communication is over port 8140
- “puppet apply” is a standalone client