



Puppet

Luke Kanies
Founder, Reductive Labs
Nashville, Tennessee
USA

Let's Talk About Resources

- Puppet's primitives are all 'resources', which map to the entities we really care about
 - users
 - groups
 - packages
 - services
 - cron jobs
 - hosts
 - filesystem mounts
 - etc.

A Portable Resource Abstraction Layer

- Puppet's fundamental advancement is a library that provides a consistent API to portable resources
- This library allows you to ignore low-level details when building configurations
- Not that the detail is gone, you just aren't **required** to deal with it

A Tool Stack

Configuration Language

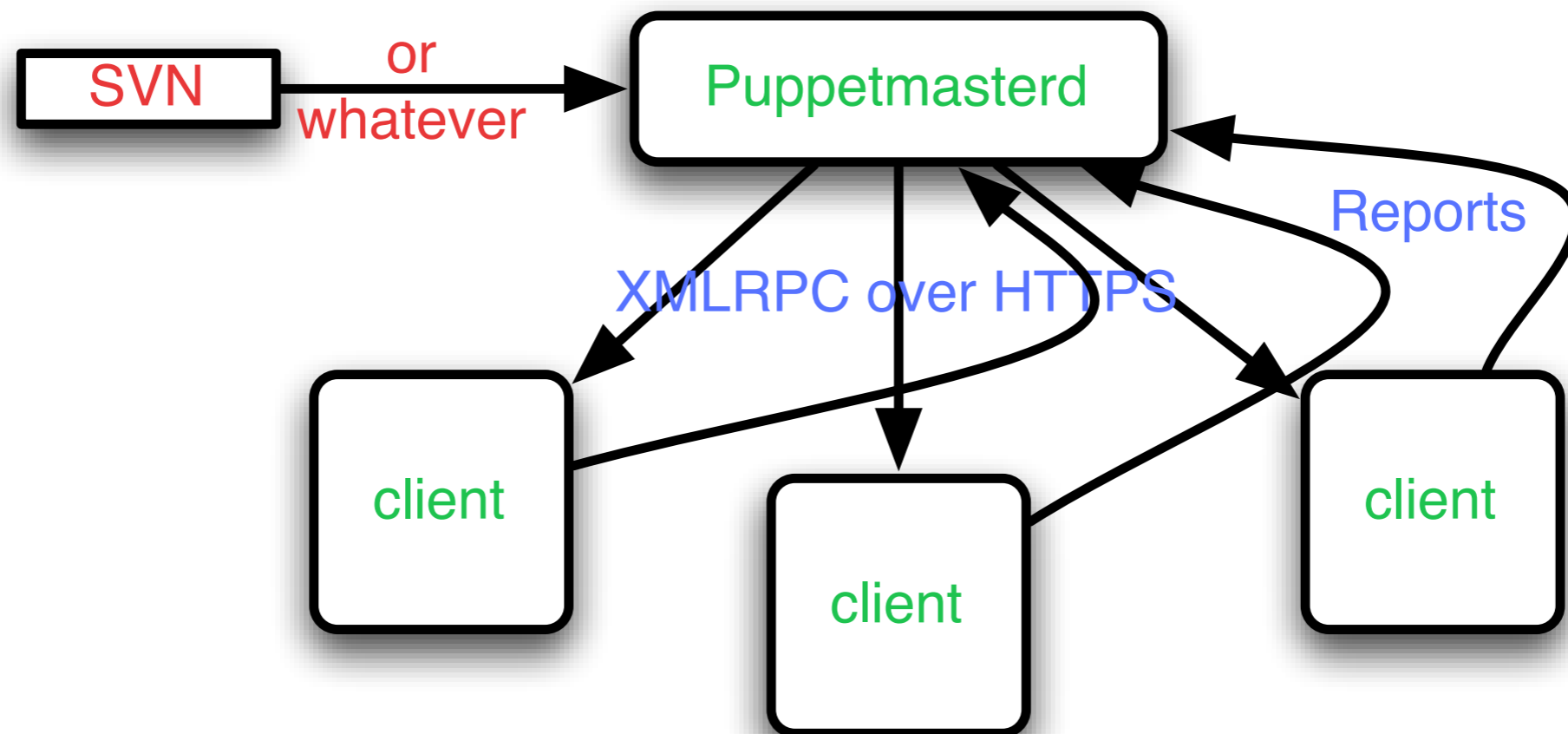
Client and Server Tools

Portable Resource Abstraction
Layer

Clients and Servers

- Configuration Retrieval
- Certificate Authority
- Triggering configuration runs
- Reporting
 - etc.

Centralized Management



Specifying Resources

```
package { openssh: ensure => latest }
```

```
user { luke:  
  uid => 100,  
  gid => 100,  
  home => "/home/luke",  
  shell => "/usr/bin/bash",  
  ensure => present  
}
```

```
service { openssh: ensure => running }
```

Resource Relationships

```
class ssh {  
  package { ssh: ensure => latest,  
    name => $operatingsystem ? { solaris => openssh, debian => ssh-server,  
      default => ssh  
    }  
  }  
  
  file { "/etc/ssh": source => "/nfs/files/ssh",  
    recurse => true, notify => Service[ssh]  
  }  
  
  service { ssh:  
    name => $operatingsystem ? { solaris => openssh, default => ssh },  
    ensure => running, subscribe => Package[ssh]  
  }  
}  
  
node mynode { include ssh }
```


Resource Relationships

```
class ssh {  
  package { ssh: ensure => latest,  
    name => $operatingsystem ? { solaris => openssh, debian => ssh-server,  
      default => ssh  
    }  
  }  
  
  file { "/etc/ssh": source => "/nfs/files/ssh",  
    recurse => true, notify => Service[ssh]  
  }  
  
  service { ssh:  
    name => $operatingsystem ? { solaris => openssh, default => ssh },  
    ensure => running, subscribe => Package[ssh]  
  }  
}  
  
node mynode { include ssh }
```

Resource Relationships

```
class ssh {  
  package { ssh: ensure => latest,  
    name => $operatingsystem ? { solaris => openssh, debian => ssh-server,  
      default => ssh  
    }  
  }  
  
  file { "/etc/ssh": source => "/nfs/files/ssh",  
    recurse => true, notify => Service[ssh]  
  }  
  
  service { ssh:  
    name => $operatingsystem ? { solaris => openssh, default => ssh },  
    ensure => running, subscribe => Package[ssh]  
  }  
}  
  
node mynode { include ssh }
```

Organization Portability

```
define virtual_host($docroot, $ip, $order = 500, $ensure = "enabled") {  
  $file = "/etc/sites-available/$name.conf"  
  file { $file:  
    content => template("virtual_host.erb"),  
    notify => Service[apache]  
  }  
  
  file { "/etc/sites-enabled/$order-$name.conf":  
    ensure => $ensure ? {  
      enabled => $file,  
      disabled => absent  
    }  
  }  
}  
  
virtual_host { "reductivelabs.com":  
  order => 100,  
  ip => "192.168.0.100",  
  docroot => "/var/www/reductivelabs.com/htdocs"  
}
```

Organization Portability

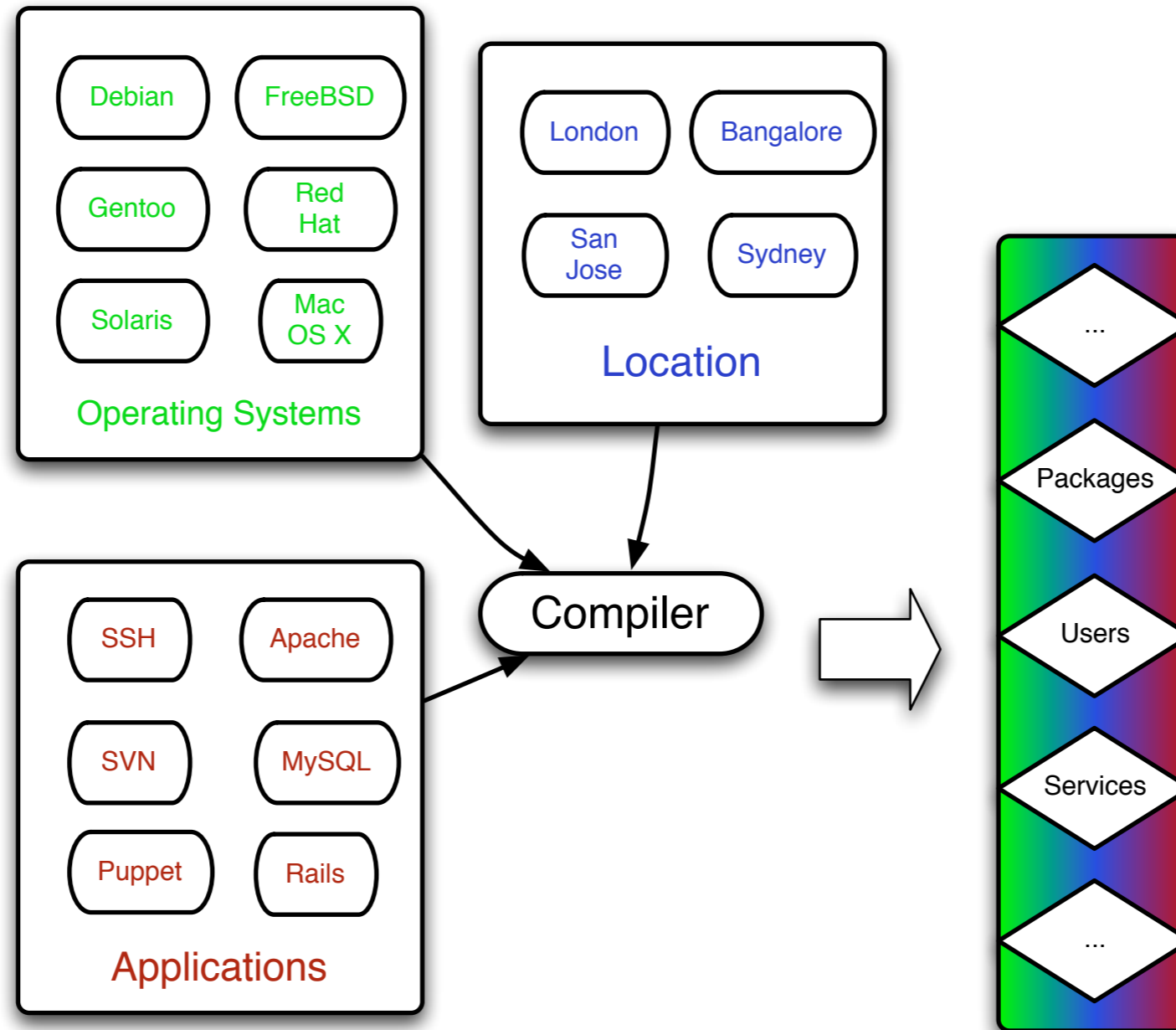
```
define virtual_host($docroot, $ip, $order = 500, $ensure = "enabled") {  
    $file = "/etc/sites-available/$name.conf"  
    file { $file:  
        content => template("virtual_host.erb"),  
        notify => Service[apache]  
    }  
  
    file { "/etc/sites-enabled/$order-$name.conf":  
        ensure => $ensure ? {  
            enabled => $file,  
            disabled => absent  
        }  
    }  
}  
  
virtual_host { "reductivelabs.com":  
    order => 100,  
    ip => "192.168.0.100",  
    docroot => "/var/www/reductivelabs.com/htdocs"  
}
```

Organization Portability

```
define virtual_host($docroot, $ip, $order = 500, $ensure = "enabled") {  
    $file = "/etc/sites-available/$name.conf"  
    file { $file:  
        content => template("virtual_host.erb"),  
        notify => Service[apache]  
    }  
  
    file { "/etc/sites-enabled/$order-$name.conf":  
        ensure => $ensure ? {  
            enabled => $file,  
            disabled => absent  
        }  
    }  
}
```

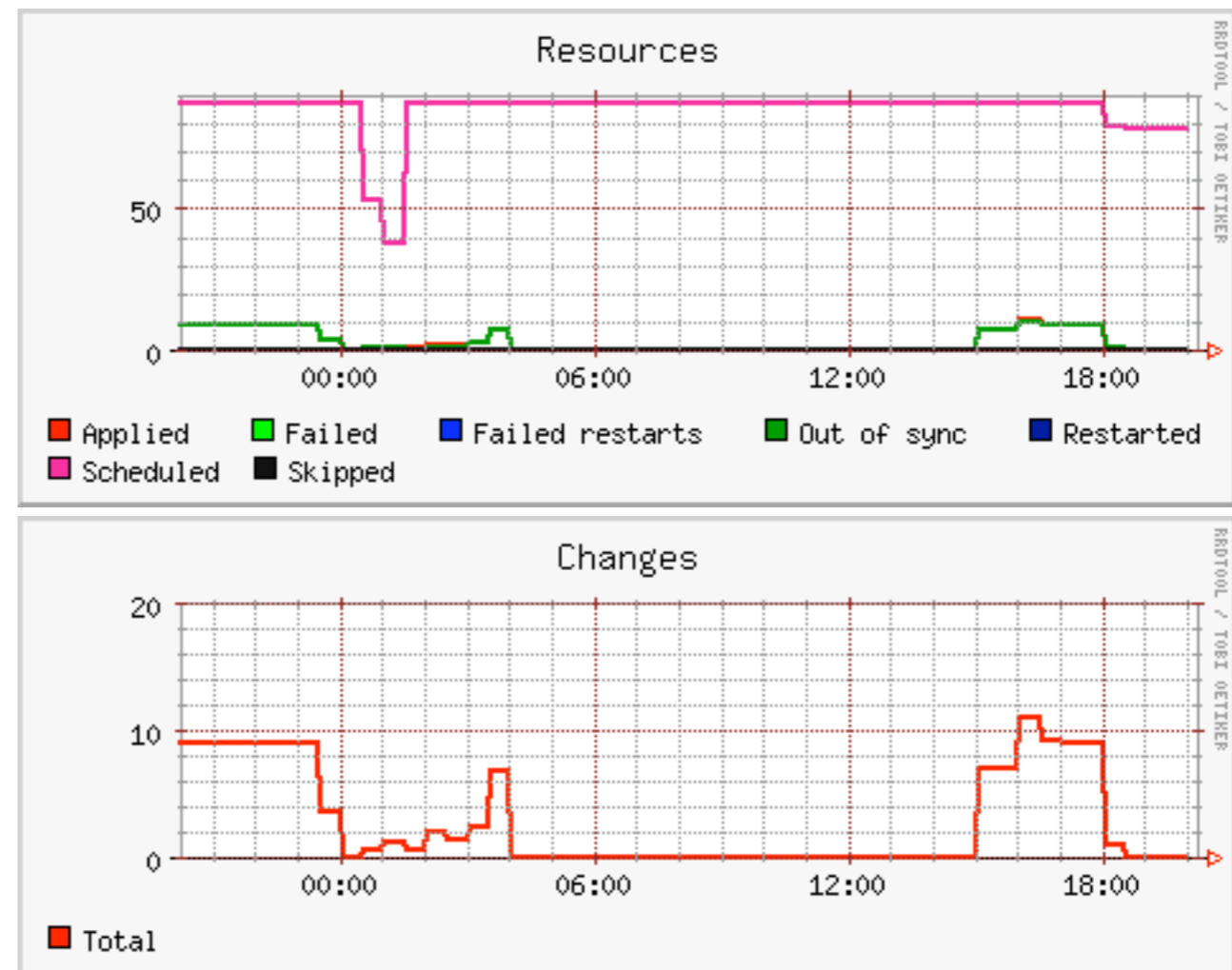
```
virtual_host { "reductivelabs.com":  
    order => 100,  
    ip => "192.168.0.100",  
    docroot => "/var/www/reductivelabs.com/htdocs"  
}
```

Aspect-based Language



Reporting

- Multiple simple reports are available
- They are easy to add



A tool, not **the** tool



- Puppet is one tool, but we need a whole ecosystem
- Puppet's library could be the basis for other portable tools
- Hopefully a step towards many small management tools working together

Easy to contribute

- *up2date* support was added by a first-time user and non-Ruby developer in around 30 minutes
- Gentoo support was provided by José González Gómez in Spain
- Red Hat has provided volume management support for Linux

About the Implementation



- Written in Ruby
- > 30% of the code is test code
- Types are usually short, and are generally 90% documentation and validation
- Everything autoloads: Drop a new file in and start using it

Current Status

- In production use around the world (at least Germany, Australia, US)
- I just finished an implementation at Stanford University
- What it lacks in maturity it makes up for in consistency and openness
- A great and growing community

Reductive Labs

- Open Source Software Startup
- Other projects planned, e.g., Runnels
- My full-time job
- Currently: consulting, custom development, and support

