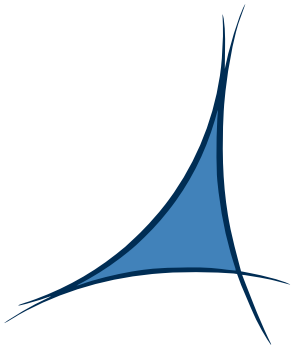


PIC
port d'informació
científica

Site integral management with Puppet

M. Caubet, A. Bria, X. Espinal

PIC (Port d'Informació Científica)

Barcelona (Spain)

Generalitat de Catalunya
Departament d'Innovació, Universitats i Empresa

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN
Ciemat
Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

Institut de Física d'Altes Energies IFAE

UAB
Universitat Autònoma de Barcelona

UAB CEI
CAMPUS D'EXCEL·LÈNCIA INTERNACIONAL

PIC
port d'informació
científica

Index

1. Introduction
2. Puppet Architecture
3. Puppet Internals
4. Puppet in production: examples
5. Conclusions

PIC
port d'informació
científica

Introduction

- PIC (Port d'Informació Científica) is a data center of excellence for scientific-data processing.
- Current capacities: 4PB on disk, 3.5PB on tape and 3k cores
  - >600 servers and >70 diferent profiles
- Services group is composed by 8 people
- Persons/services balance indicates:
  - Clear need for centralized management tools
  - Target on automation
- Different tools evaluated since 2003, some basic (scripts) and some complex (quattor)
- In 2010 puppet was adopted as our central management tool.

PIC
port d'informació
científica

Introduction - Puppet Highlights

- Offers a gradual integration
- Declarative Language
- Ensure an homogeneous environment (transversal configs)
    - And service specific tuning on demand
- Runs over several O.S. platforms
    - High flexibility for adapting new projects (new requirements)
- Deploy personalized modules
- Quick benefits:
    - decrease of the administration load
    - reduction of human administration errors
    - Rapid & reusable configuration
- Great community support

PIC
port d'informació
científica

Puppet Architecture - Services handled with puppet (100%)

GridFTPs

Core Servers

dCache Pools
- Solaris
- Linux

C.E./CreamC.E.

L.F.C.

W.N.

P.B.S. ... F.T.S.

...and NON-CORE SERVICES!

Tape Servers

Core Servers

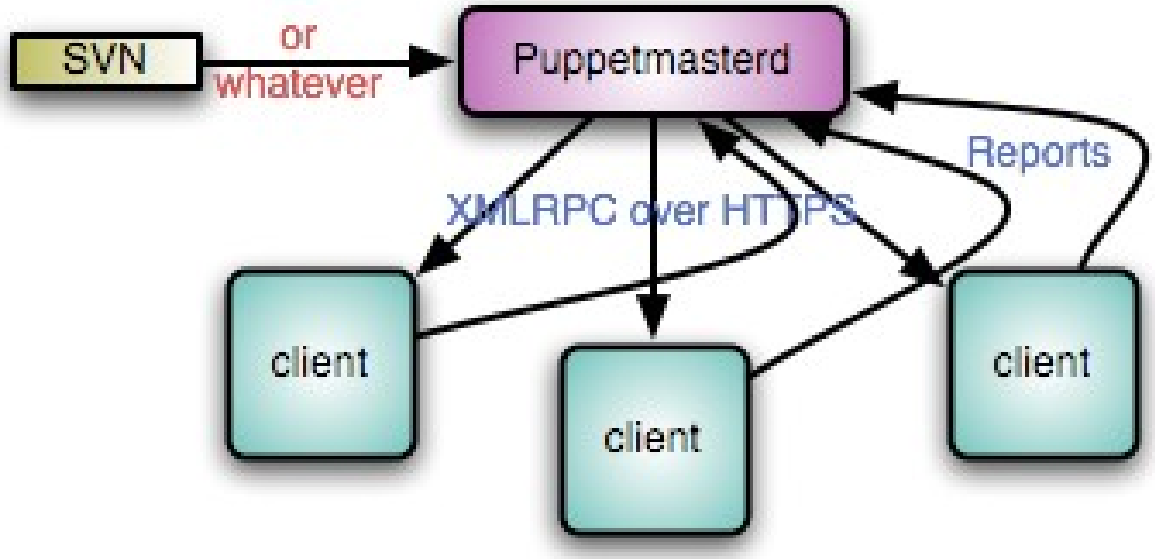
...

Pakiti

N.F.S.

Squid

Puppet Architecture

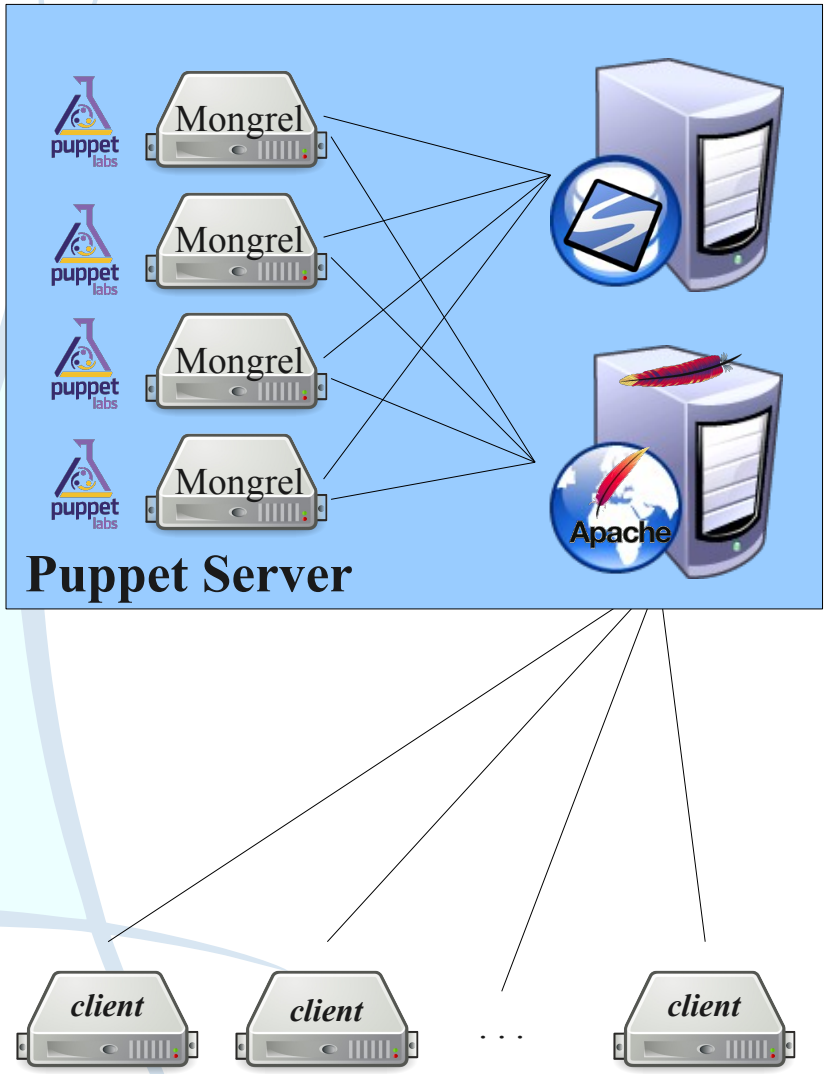


- Encrypted communication
- Agent receives a compiled catalog describing the desired configuration
- Puppet agent takes on the job to apply changes (configurations) if needed

PIC
port d'informació
científica

Puppet Architecture - Server Configuration

- Default HTTP Server: Webbrick
  - SSL
  - No Load Balancing
  - Does not scale
- Puppet + Mongrel + Apache
  - SSL Manager (Apache)
  - Load Balancing (Apache)
  - Mongrel allows to run several puppetmaster daemons
- SVN keeps code up2date
  - Change Control
  - Code update errors check



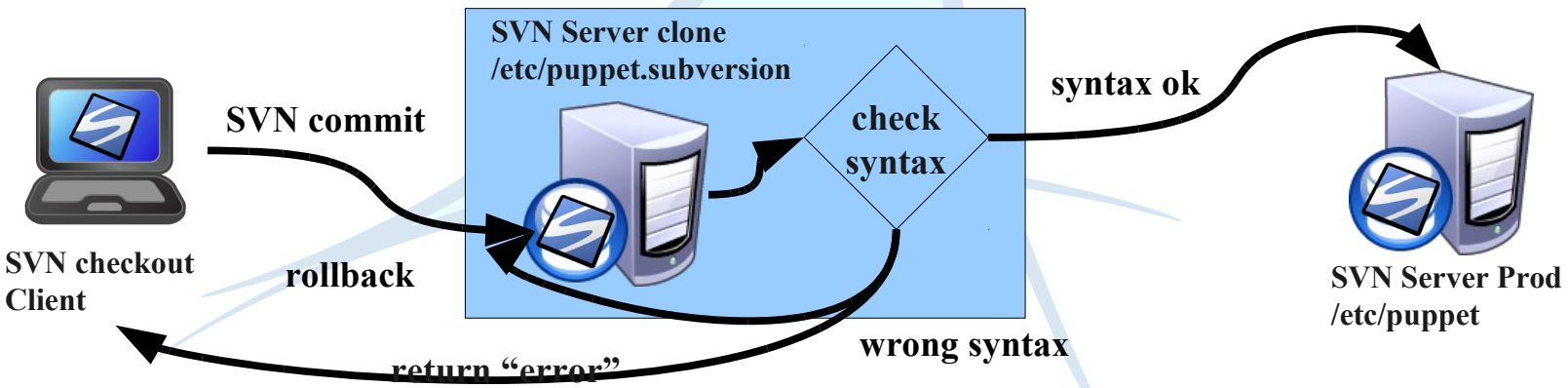
PIC
port d'informació
científica

Puppet Architecture - Change Control & Workflow

- Production SVN location: /etc/puppet
- Services are served under the directory:

  /etc/puppet/manifests/services/$module

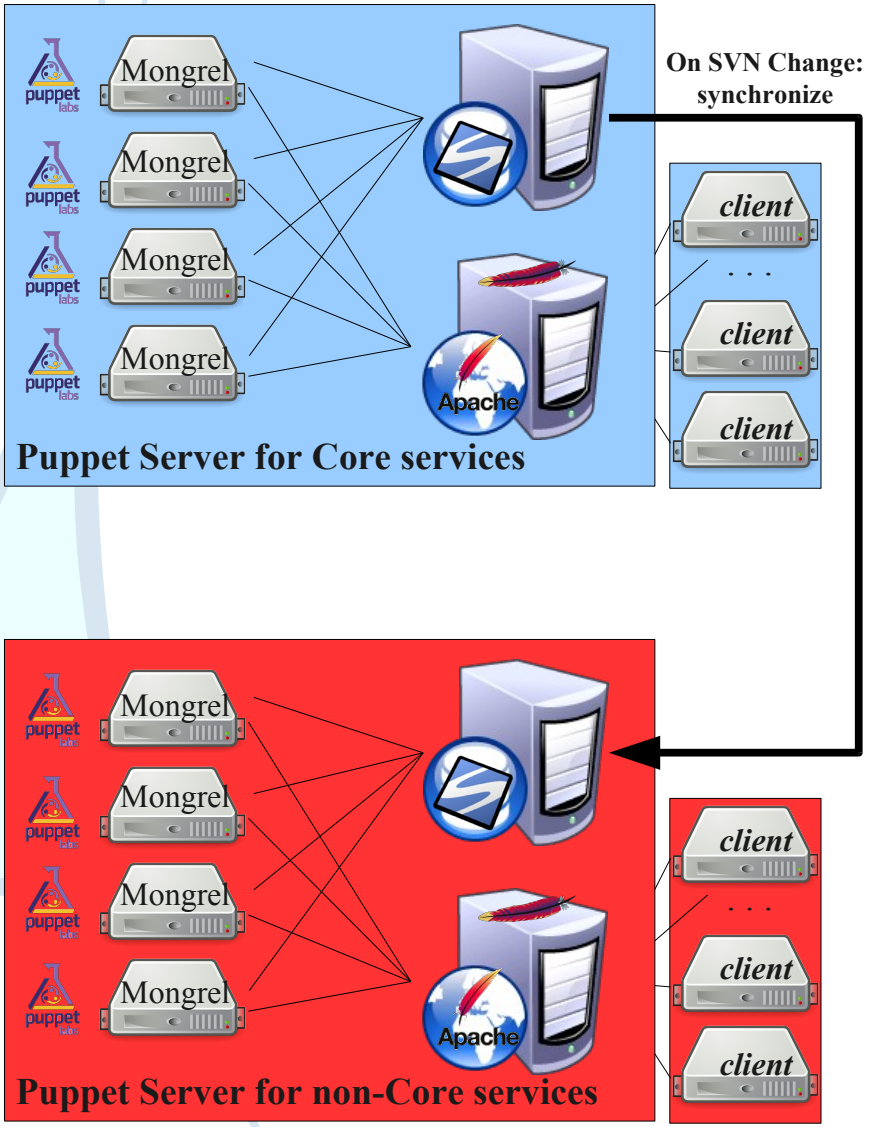
  – We configure which modules (services) we enable importing them at /etc/puppet/manifests/site.pp
- Syntax check on /etc/puppet.subversion before any SVN commit operation
  – Correct Syntax.: upload changes to /etc/puppet
  – Wrong Syntax: rollback on /etc/puppet.subversion



PIC
port d'informació
científica

Puppet Architecture - Core vs. non-Core Services

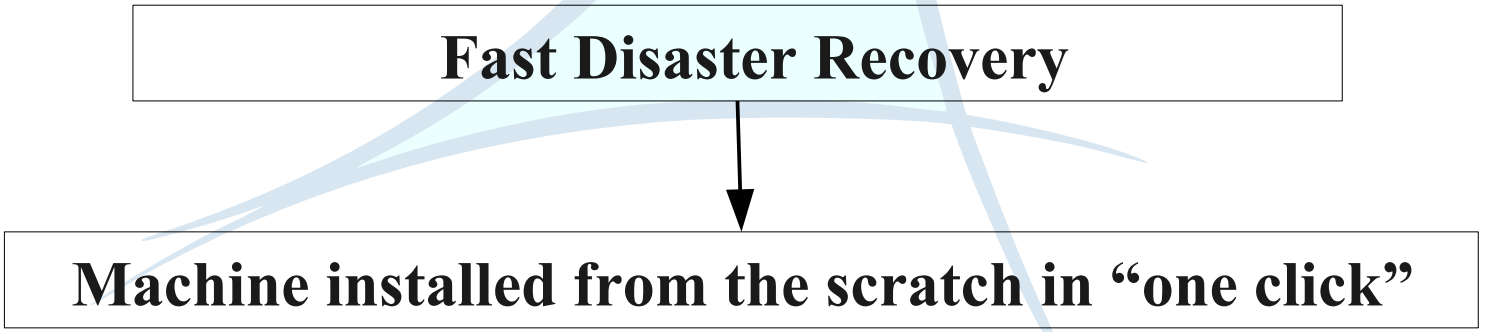
- Puppet Server dedicated for non-Core services
- SVN sync
- Common puppet basic profile for all nodes hosted at pic
- Service modules from Core Puppet Server can be reused
- Non-core services users can build their own modules



PIC
port d'informació
científica

Puppet Architecture - PIC streamlined machine installation system

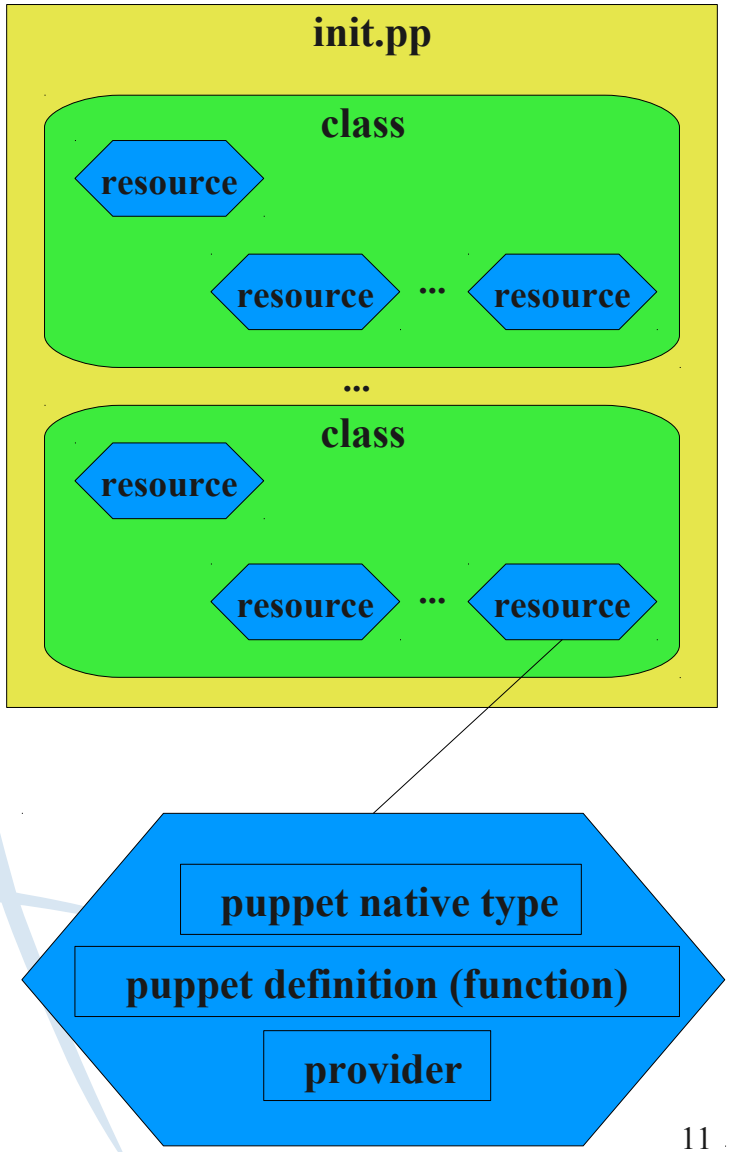
- Installation is done via PXE.
- Custom kickstart files are created by local script
- Custom postinstall is added
  - Adds local puppet repo
  - Installs desired puppet client version
  - Runs puppet against server
- The host wakes up configured and “linked” to puppet server
  - which is the case for every host at pic



Puppet Internals - Puppet Module (I)

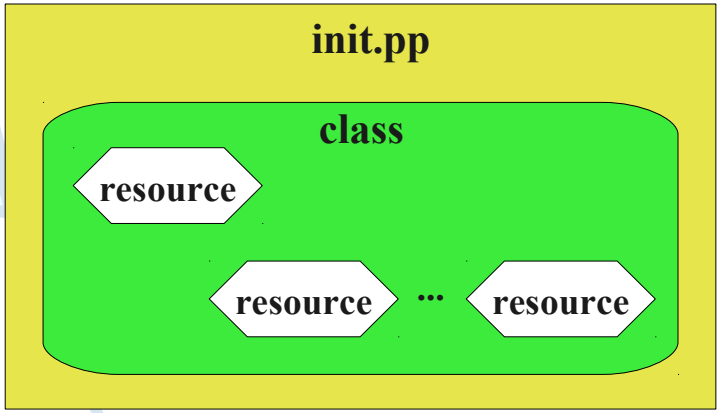
- A Puppet module is a collection of:
  - resources
  - classes
  - files
  - definitions
  - templates

```
MODULE_PATH/
    downcased_module_name/
        files/
        manifests/
            init.pp
        lib/
            puppet/
                parser/
                    functions
                provider/
                type/
            facter/
        templates/
        README
```



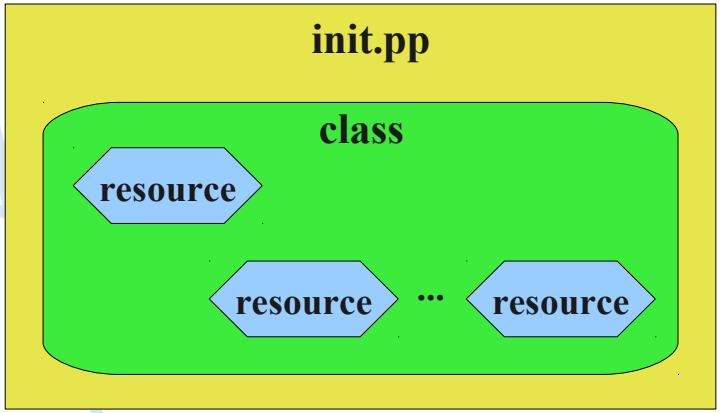
Puppet Internals - Puppet Module (II)

```
class bacula_client {

        package { 'bacula-client.$architecture':
            ensure   => latest,
            alias    => "bacula",
            provider => yum,
            require  => Repo["sl55${architecture}.repo"];
        }

        file { "bacula-fd.conf": # ... ; }

        service { "bacula-fd": # ... ; }
}
```



Puppet Internals - Puppet Module (III)

```
class bacula_client {

        package { 'bacula-client.$architecture':
            ensure   => latest,
            alias    => "bacula",
            provider => yum,
            require  => Repo["sl55${architecture}.repo"];
        }

        file { "bacula-fd.conf": # ... ; }

        service { "bacula-fd": # ... ; }
}
```

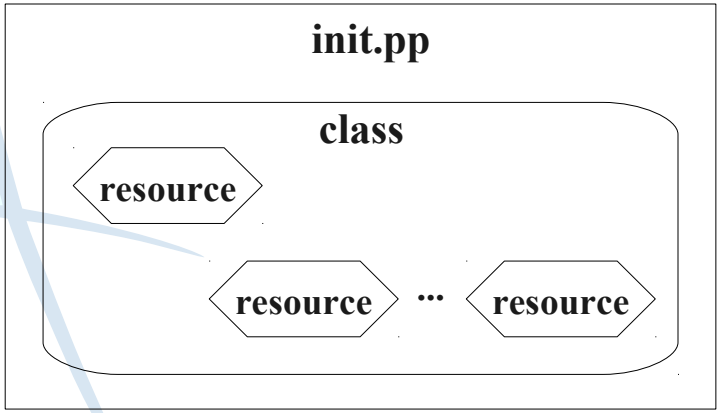


Puppet Internals - Puppet Module (IV)

Resource type

```
package { 'bacula-client.$architecture':
        ensure   => latest,
        alias    => "bacula",
        provider => yum,
        require  => Repo["sl55${architecture}.repo"];
}
```

Puppet Native Resource



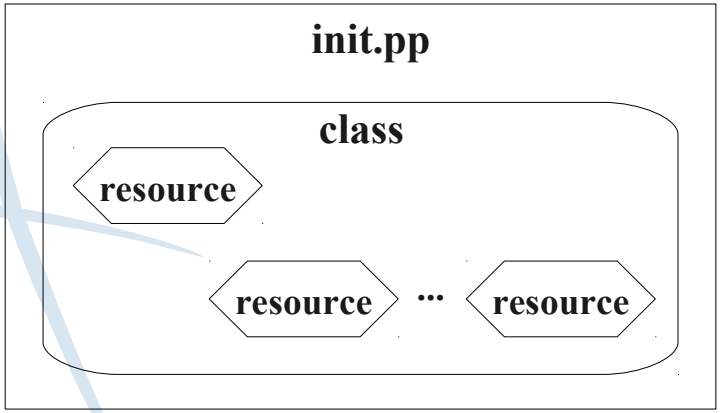
Puppet Internals - Puppet Module (V)

Resource type

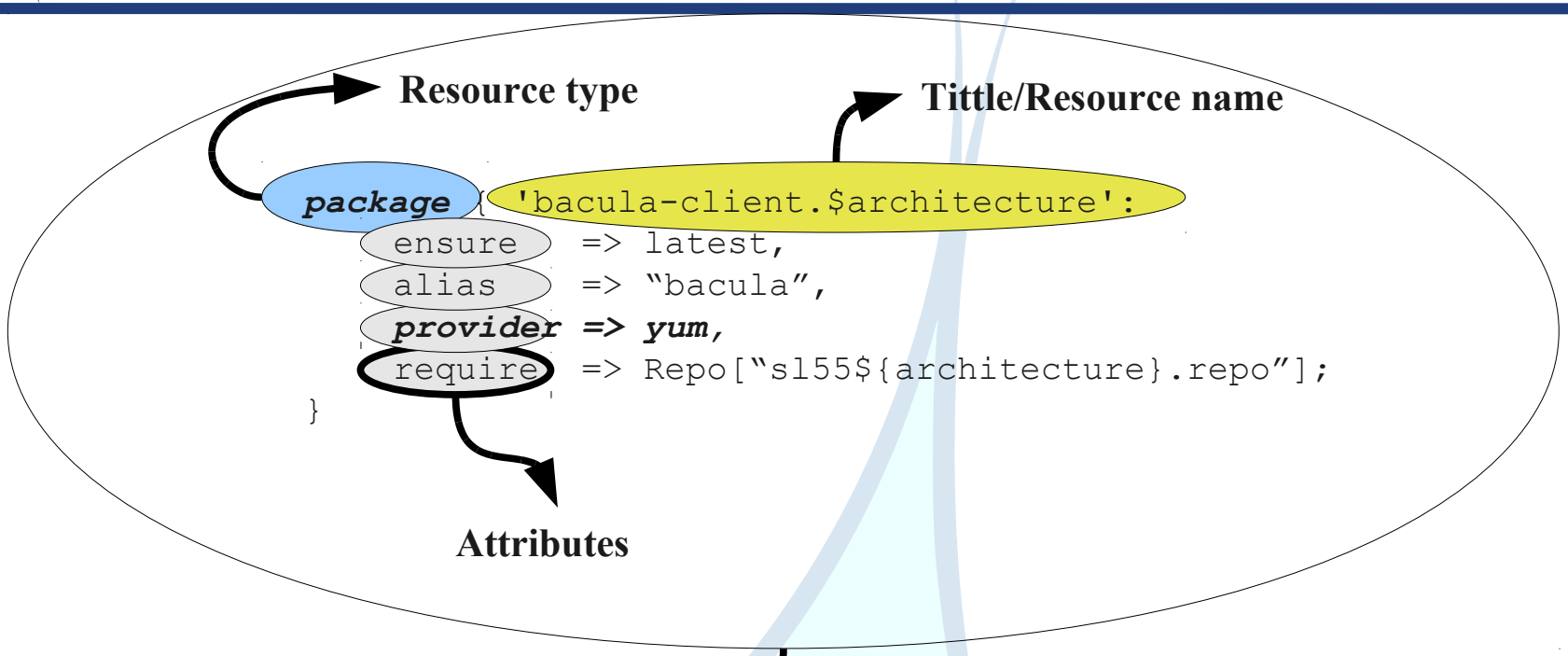
Title/Resource name

```
package { 'bacula-client.$architecture':
    ensure   => latest,
    alias    => "bacula",
    provider => yum,
    require  => Repo["sl55${architecture}.repo"];
}
```

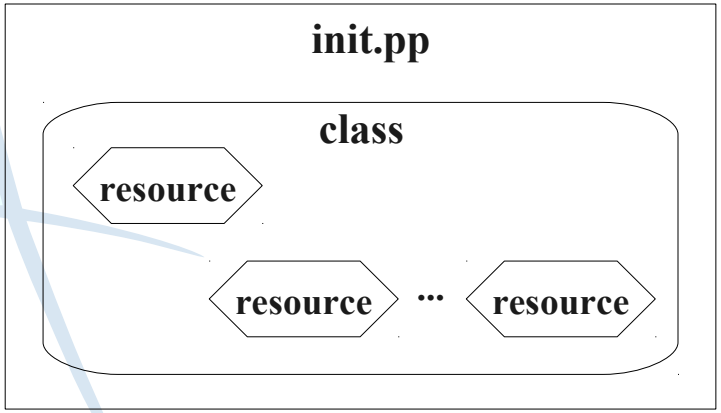
Puppet Native Resource



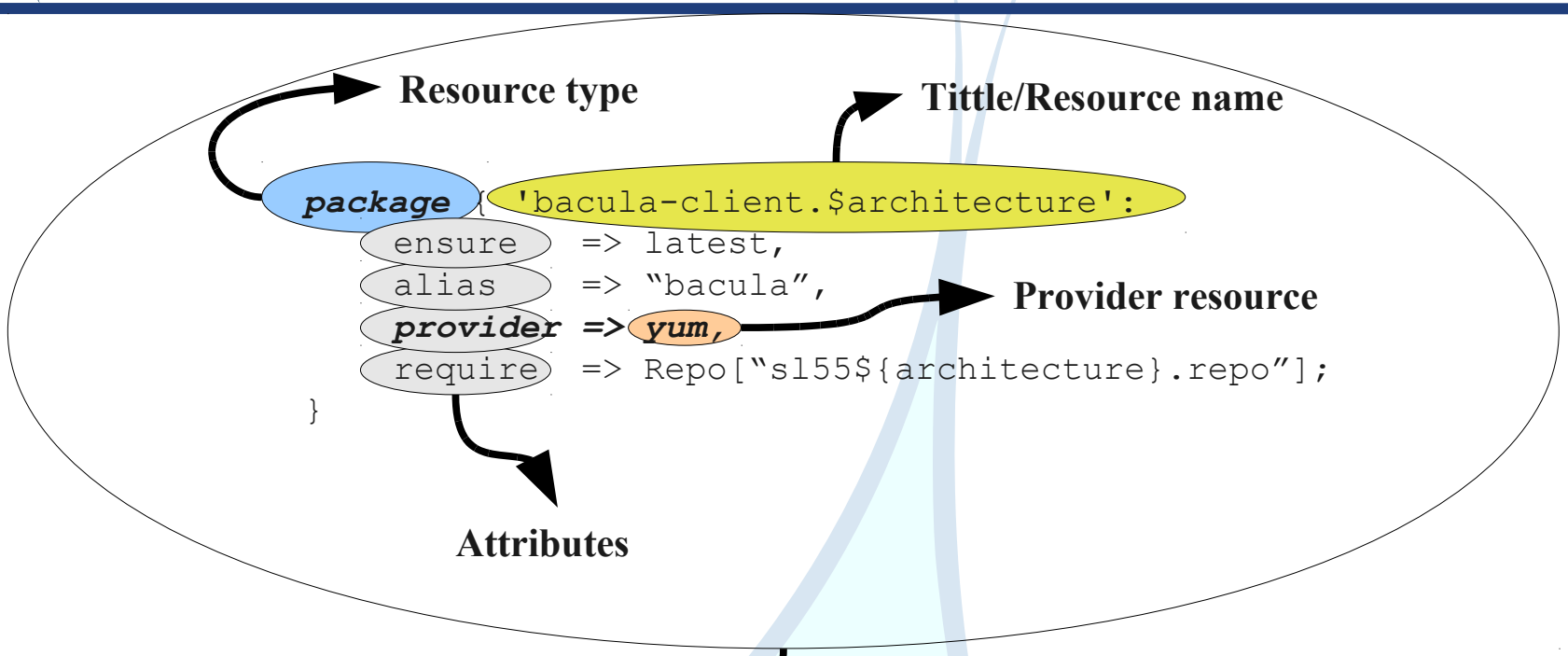
Puppet Internals - Puppet Module (VI)



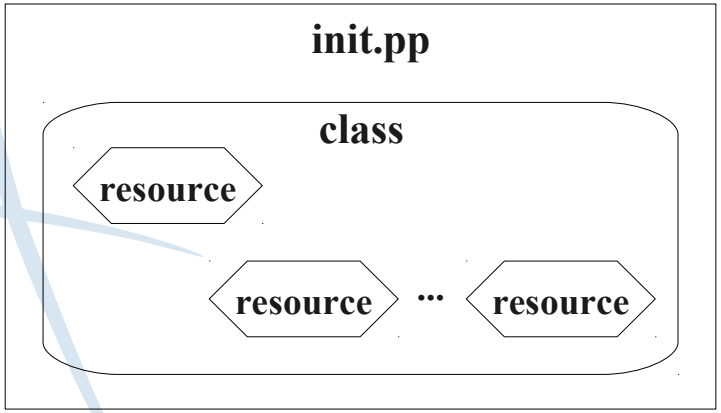
Puppet Native Resource



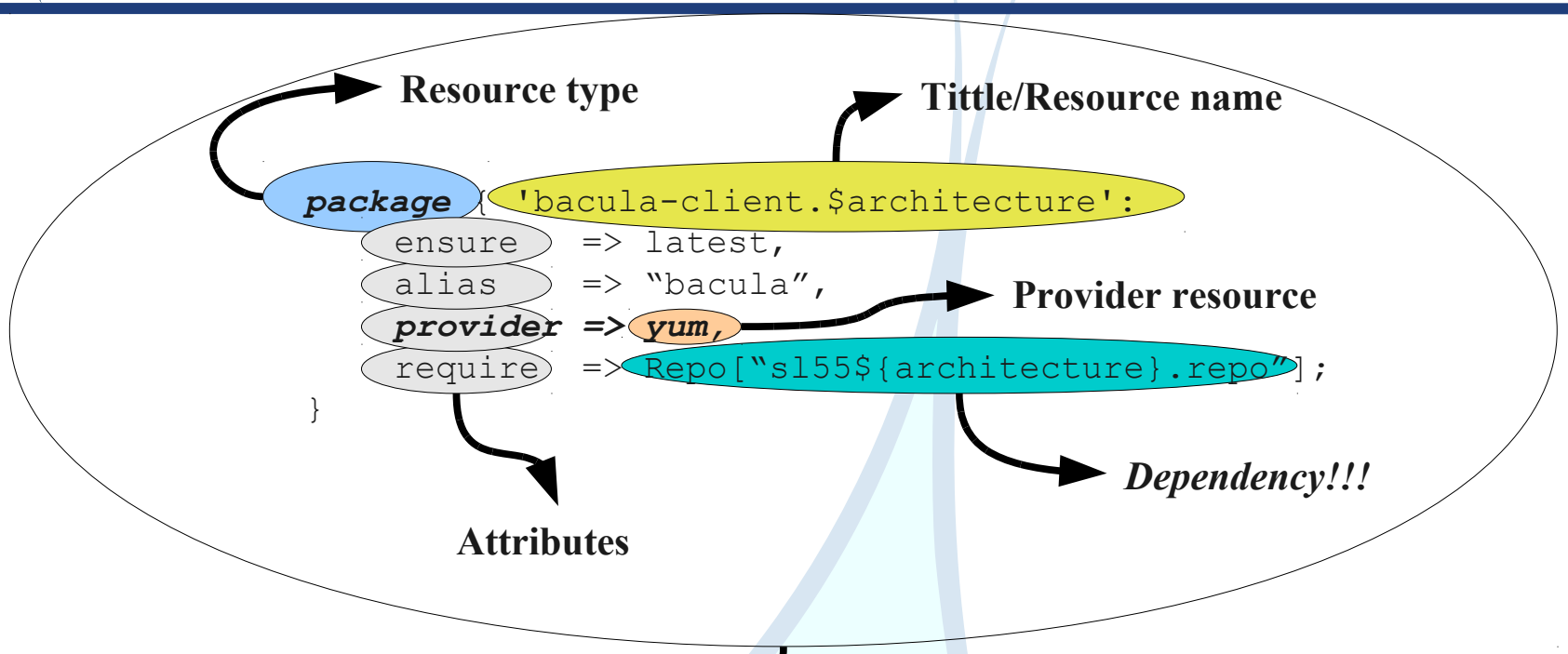
Puppet Internals - Puppet Module (VII)



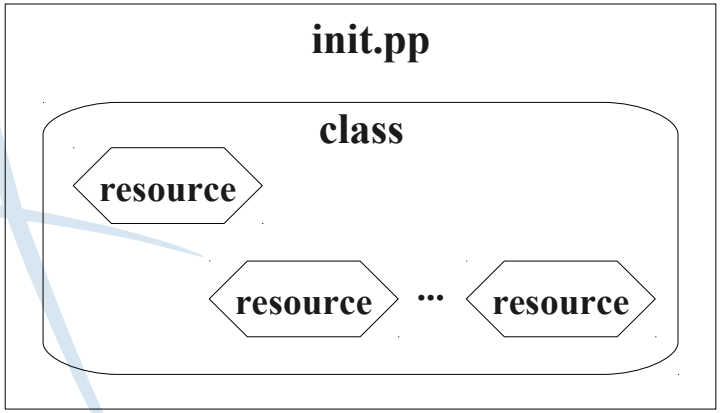
Puppet Native Resource



Puppet Internals - Puppet Module (VIII)

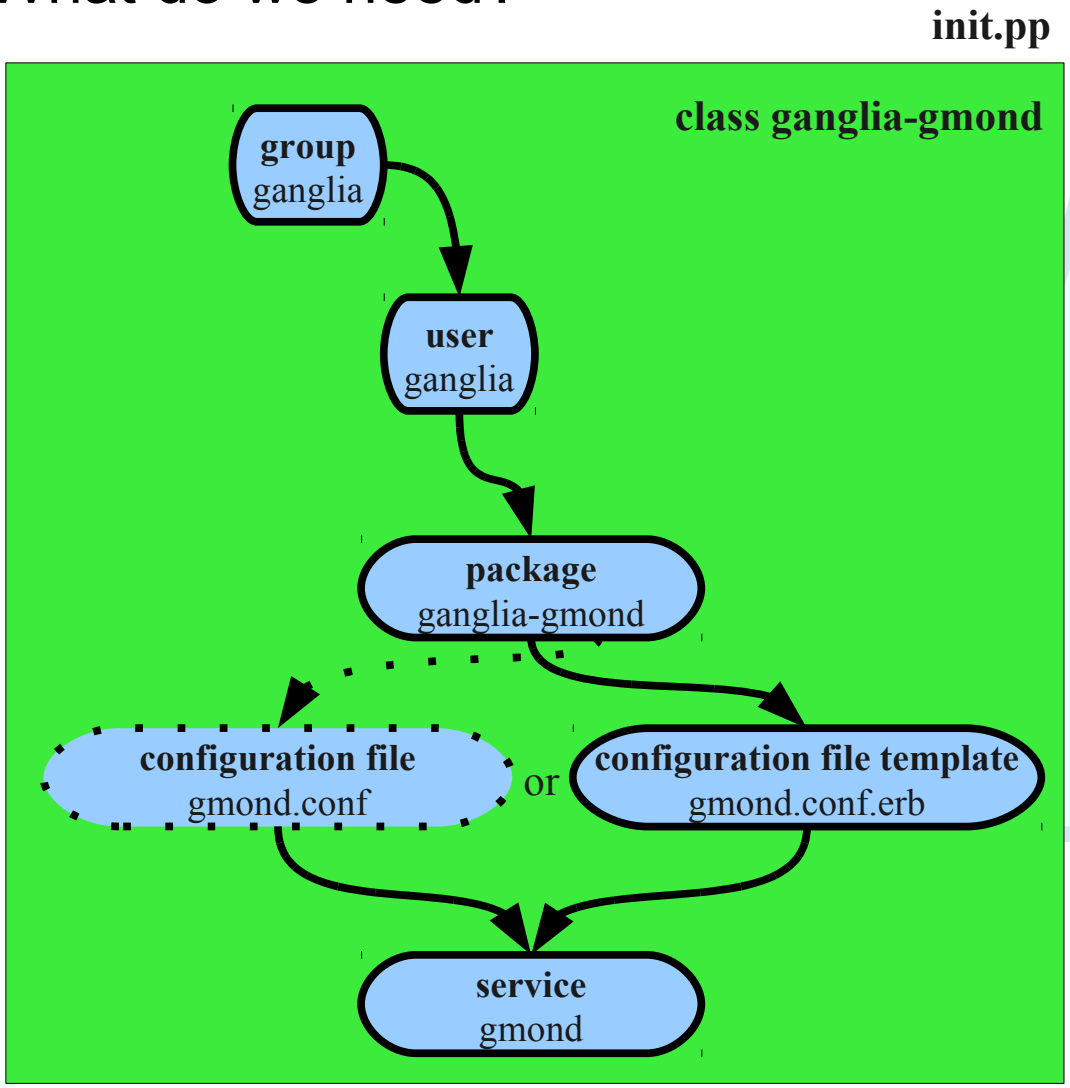


Puppet Native Resource



Puppet in production: Ganglia Client example

What do we need?

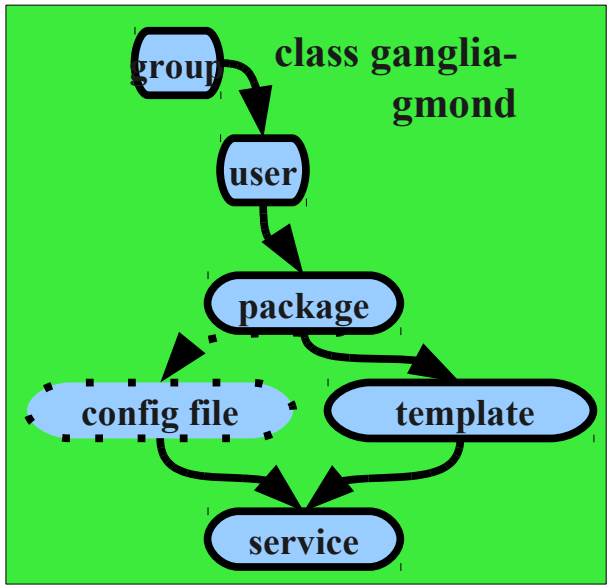



```
MODULE_PATH/
    gangliaclient/
        files/
          etc/
              gmond.conf
        manifests/
            init.pp
        lib/
            puppet/
                parser/
                    functions
                provider/
                type/
            facter/
        templates/
              gmond.conf.erb
        README
```

PIC
port d'informació
científica

Puppet in production: Ganglia Client example

```
class ganglia {
        group { 'ganglia':
                name    => 'ganglia',
                ensure  => 'present',
                gid     => 200;
        }
        user { 'ganglia':
                name    => 'ganglia',
                ensure  => 'present',
                uid     => 200,
                gid     => 200,
                home    => '/var/lib/ganglia',
                shell   => '/sbin/nologin',
                require => Group['ganglia'];
        }
        package { "ganglia-gmond.$architecture" : require => User["Ganglia"]; }
        file { '/etc/gmond.conf' :
                content => template("common_ganglia/gmond.conf.erb"),
                notify  => Service["gmond"],
        }
        service { 'gmond':
                name    => 'gmond',
                ensure  => running,
                require => Package["ganglia-gmond.$architecture"],
        }
}
```



Puppet in production: Ganglia Client example

templates/gmond.conf.erb

```
/* Beggining of the file */

...

globals {
  setuid = yes
  user = nobody
  cleanup_threshold = 300
}

cluster {
  name = "<%= cluster %>"
}

udp_send_channel {
  mcast_join = <%= mcast_ip %>
  port = 8649
  ttl = 5
}

...
```

```
...

udp_recv_channel {
  mcast_join = <%= mcast_ip %>
  port = 8649
  bind = <%= mcast_ip %>
}

tcp_accept_channel {
  port = 8649
}

...

/* End of the file */
```

Puppet in production: YAIM module at pic

- Active
  - administrator triggers the node configuration with YAIM
- What do we need?

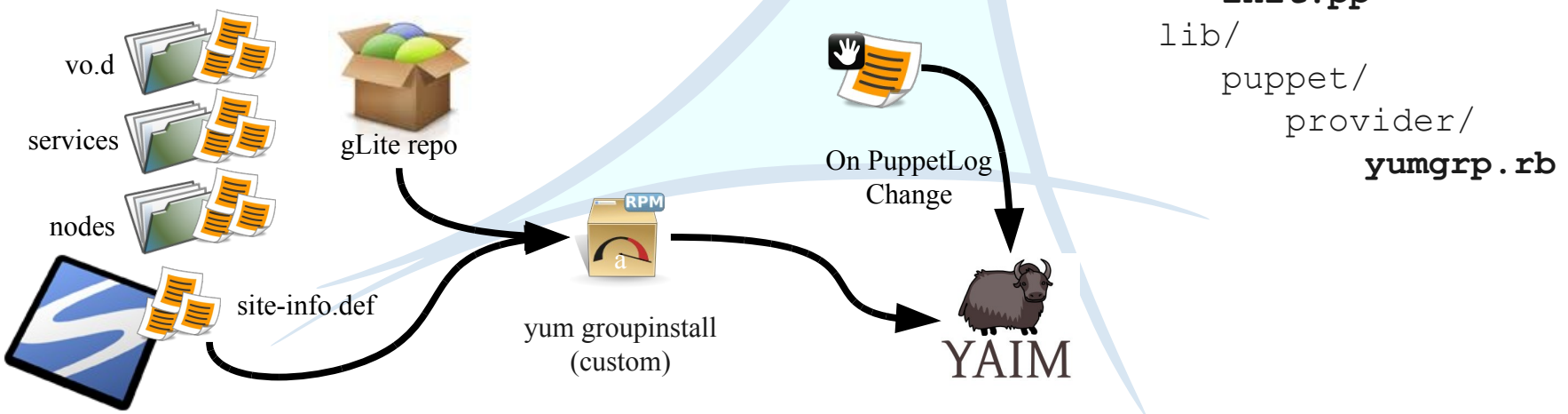
  gLite Repositories

  gLite Packages

  YAIM Configuration files

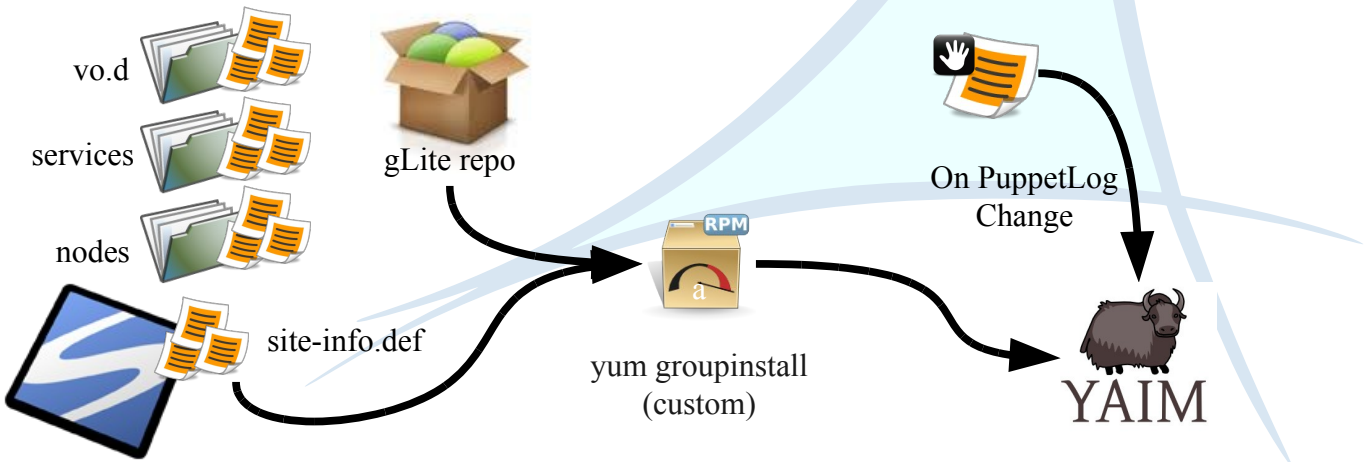
  YAIM Node Configuration

```
MODULE_PATH/
    yaim/
        manifests/
            init.pp
        lib/
            puppet/
                provider/
                    yumgrp.rb
```



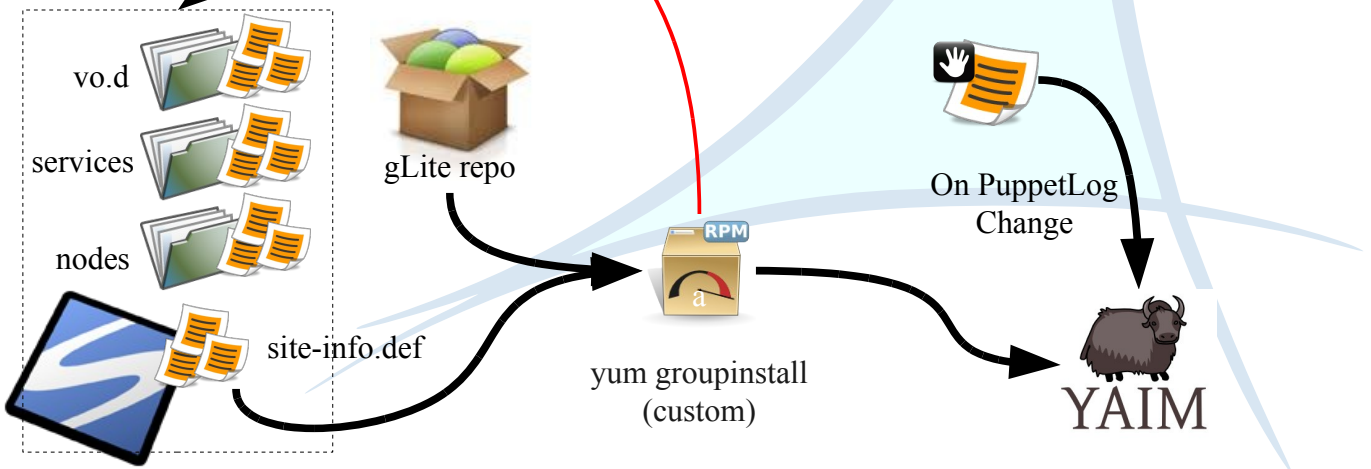
Puppet in production: YAIM module at pic

```
# Base repository (same for updates and extras repositories)

yumrepo { "glite$glite-UI.repo":
      baseurl  => "http://repo.pic.es/mrepo/glite-$glite-release-UI-$architecture/RPMS.base/",
      name     => "glite-UI",
      descr    => "gLite 3.2 UI service release repository",
      gpgkey   => "http://glite.web.cern.ch/glite/glite_key_gd.asc",
      exclude  => "maui maui-client",
      gpgcheck => 0,
      enabled  => 1,
}
```



Puppet in production: YAIM module at pic

```
package { "glite-UI":
    ensure  => installed,
    provider => yumgroupinstall,
    require     => [ Class["common_yaimfiles"], Yumrepo["glite-UI"], ... ];
}
```



PIC
port d'informació
científica

Puppet in production: YAIM module at pic

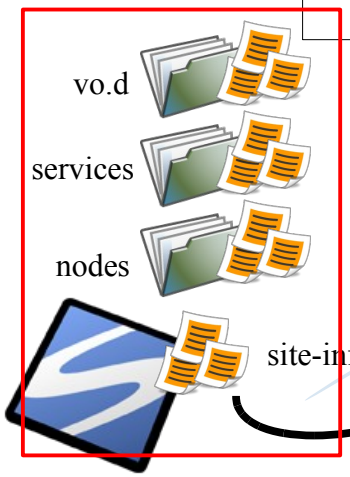
```
file { '/opt/localconf/' :
          ensure  => directory,
          mode    => 700 ,
          recurse => true;
       # ...
       '/root/.subversion/auth/svn.simple/038204f6e0a3451cbdf1440fa00a6e10' :
          require => File['/root/.subversion/auth/svn.simple/'],
          content => '$SVN_PASSWORD'; }

exec { 'svn_check_out' :
       cwd     => '/opt/localconf',
       command => 'svn co svn://ser01.pic.es/yaim_conf/gLite/',
       creates => '/opt/localconf/gLite/',
       require => File['localconf'];
     'svn_update' :
       cwd     => '/opt/localconf',
       command => 'svn up gLite',
       require => [ Exec['svn_check_out'],
                    File['/root/.subversion/auth/svn.simple/
                    038204f6e0a3451cbdf1440fa00a6e10']]; }
```

Secure gLite permisssions

Define SVN authentication

SVN checkout

SVN update




gLite repo


On PuppetLog Change

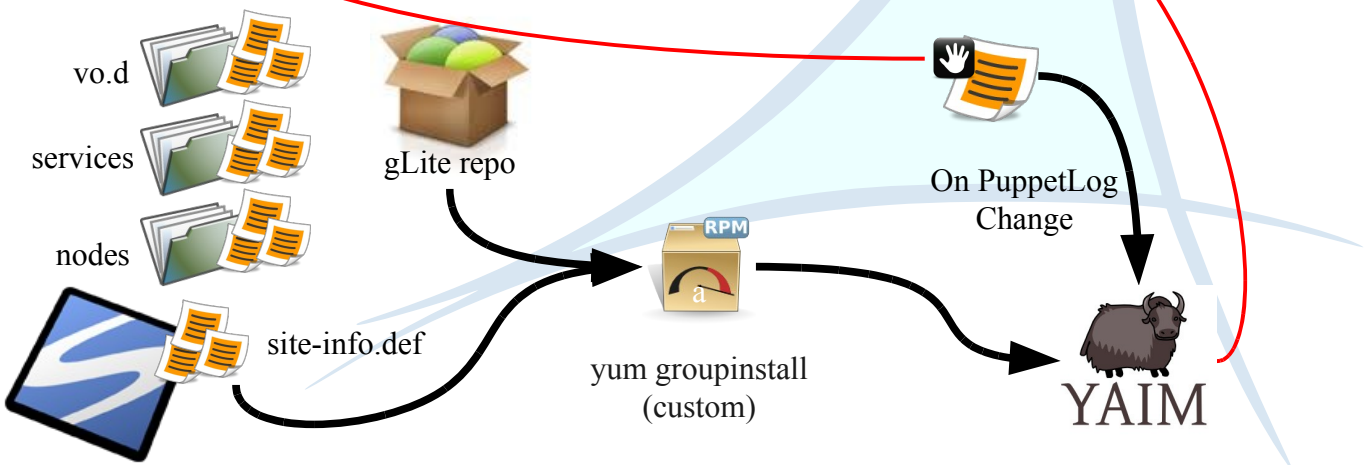
yum groupinstall (custom)


YAIM

PIC
port d'informació
científica

Puppet in production: YAIM module at pic

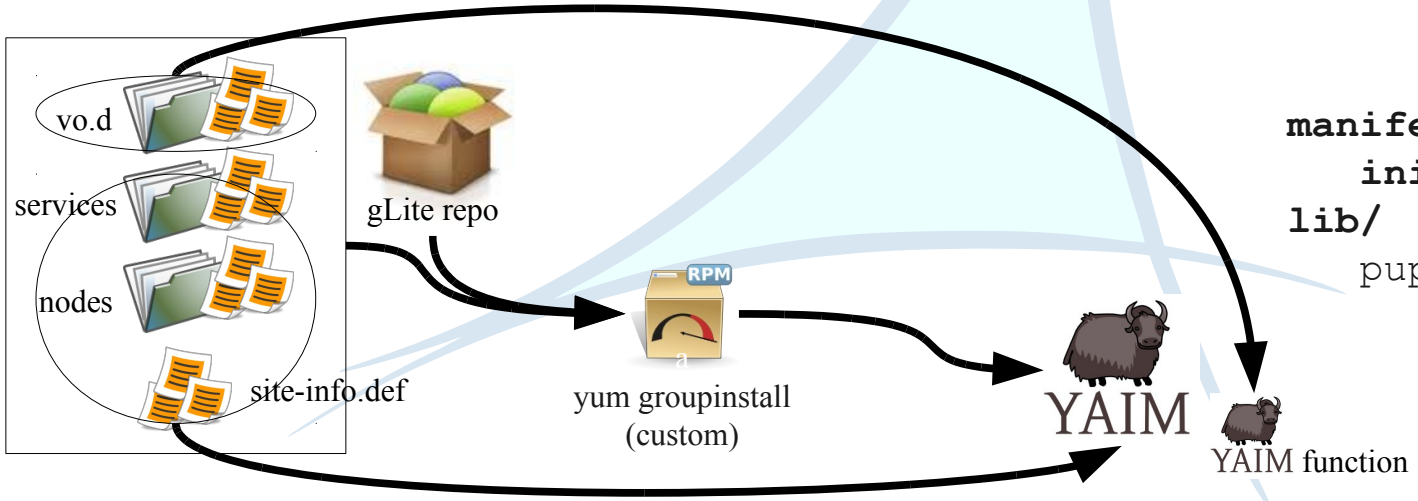
```
define common_exec_yaim($common_yaim_environemnt,$yaim_meta) {
        exec {
        'yaim_conf' :
                command => "/opt/glite/yaim/bin/yaim -c -s /opt/localconf/gLite/yaim/
$common_yaim_environemnt/site-info.def $yaim_meta",
                unless  => "tail -n1 /opt/glite/yaim/log/yaimlog|grep 'INFO: YAIM terminated
succesfully'",
                require => Package[“glite-UI”];
        }
}

common_exec_yaim {
        'yaim_UI_pic' :
                common_yaim_environemnt => prod,
                yaim_meta               => '-n glite-UI',
                notify                  => Class['pbsclient_conf'],
}
```



Puppet in production: YAIM module alternatives

- Passive
  - On configuration file update
  - Puppet inmediatly reconfigures the node with YAIM
- What do you need?
  - gLite Repositories
  - gLite Packages
  - YAIM Configuration files
  - YAIM Node Configuration

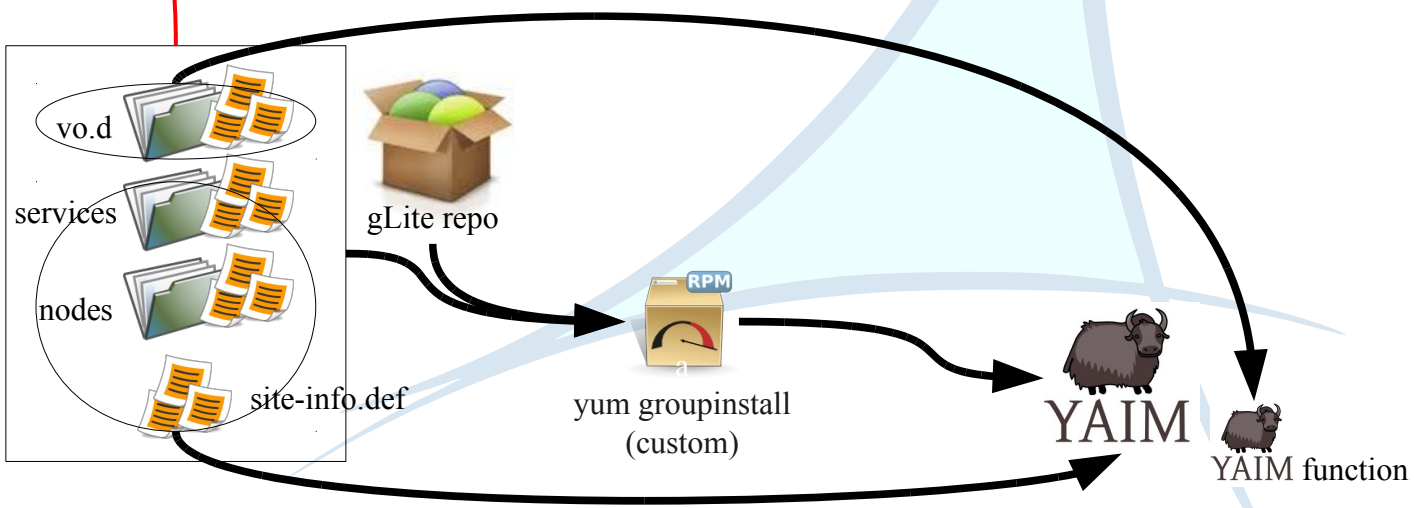
```
MODULE_PATH/
    yaim/
        files/
            opt/
                yaim_prod/
                    site-info.def
                    ...
                    vo.d/
                        atlas
                        ...
                    services/
                        ...
                    nodes/
                        ...
                yaim_test/
                    ...
        manifests/
            init.pp
        lib/
            puppet/
                provider/
                    yumgrp.rb
```



PIC
port d'informació
científica

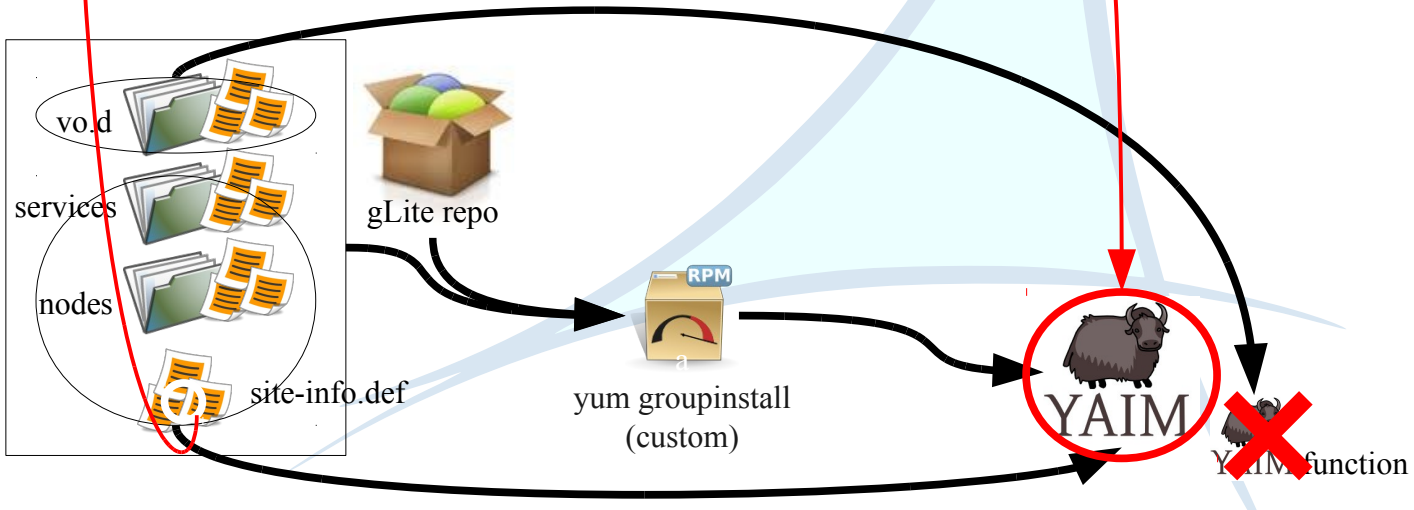
Puppet in production: YAIM module alternatives

```
$yaim_location = "/opt/localconf/gLite/yaim/$common_yaim_environment"

File {
   ensure  => directory,
   mode    => 700,
   owner   => root,
   group   => root,
}

file {
   # ...
   "${yaim_location}":            require => File["/opt/localconf/gLite/yaim"] ;
   "${yaim_location}/vo.d":       require => File["${yaim_location}"] ;
   "${yaim_location}/nodes":      require => File["${yaim_location}"] ;
   "${yaim_location}/services":   require => File["${yaim_location}"] ;
}
```



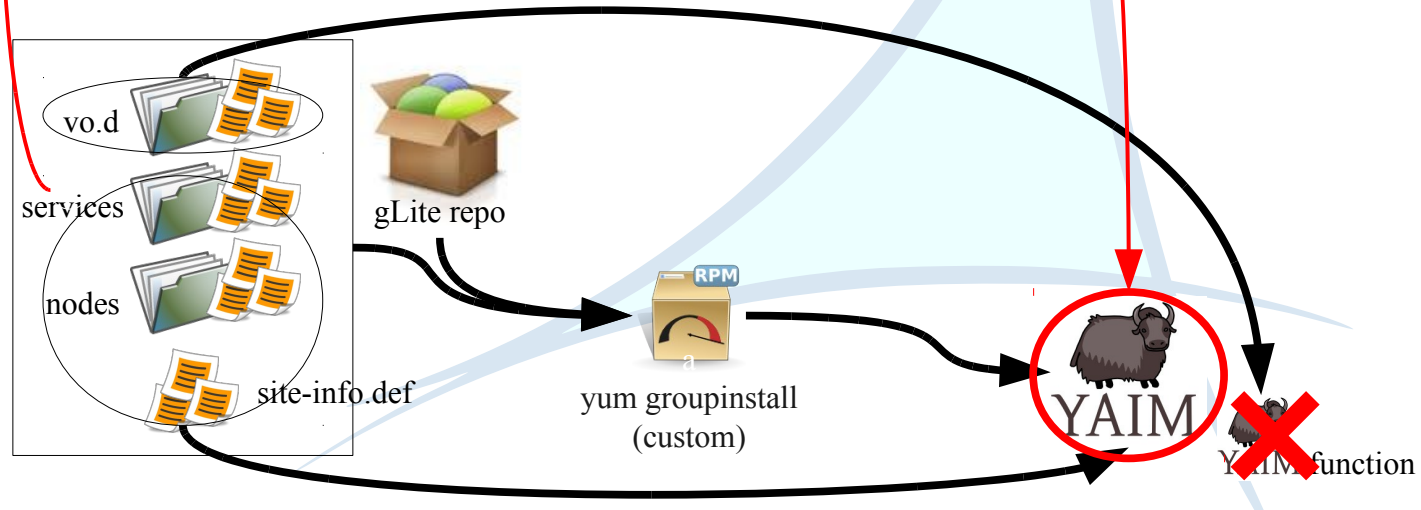
Puppet in production: YAIM module alternatives

```
#### $yaim_location
define yaim_base {
   file { "$name":
      path    => "${yaim_location}/${name}",
      source  => "puppet://$pserver/opt/yaim_${environment}/${name}",
      require => File["${yaim_location}"],
      notify  => Run yaim node[$yaim_nodetype];
   }
}
yaim_base { [ "site-info.def", "users.conf", "groups.conf", <...> , "edgusers.conf" ]: }
```



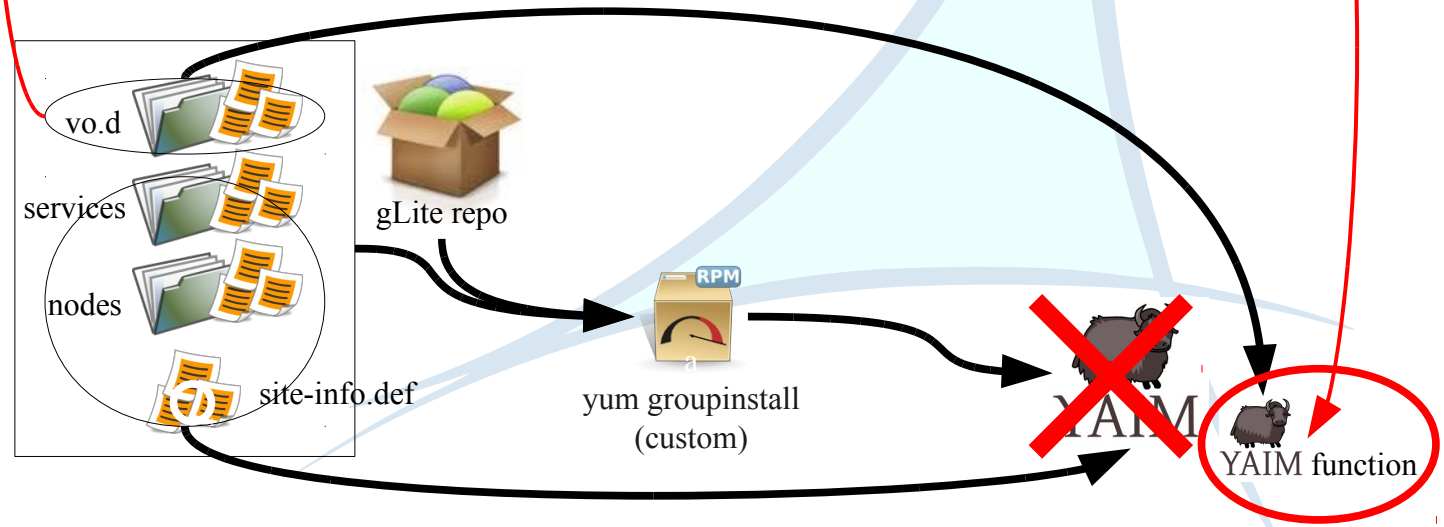
Puppet in production: YAIM module alternatives

```
#### $yaim_location/services ($yaim_location/nodes should be the same)
define yaim_services {
   file { "$name":
      path    => "${yaim_location}/services/${name}",
      source  => "puppet://$pserver/opt/yaim_${environment}/services/${name}",
      require => File["${yaim_location}/services"],
      notify  => Run yaim node[$yaim_nodetype];
   }
}
yaim_services { [ "glite-fta2" , "glite-fts2", <...>, "glite-creamce" ]: }
```



Puppet in production: YAIM module alternatives

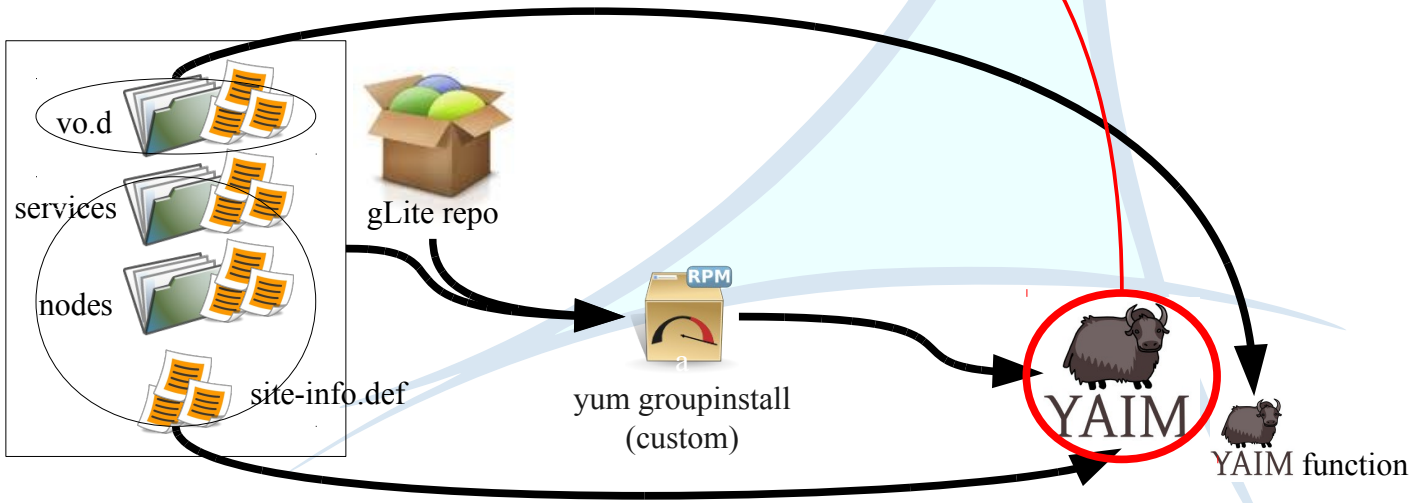
```
#### $yaim_location/vo.d
define yaim_vod {
   file { "$name":
      path    => "${yaim_location}/vo.d/${name}",
      source  => "puppet://$pserver/opt/yaim_${environment}/vo.d/${name}",
      require => File["${yaim_location}/vo.d"],
      notify  => Run yaim function vomsdir[$yaim_nodetype];
   }
}
yaim_vod { [ "ops", "cms", "lhcb", "atlas", "dteam", "magic", <...> , "t2k.org" ]: }
```



Puppet in production: YAIM module alternatives

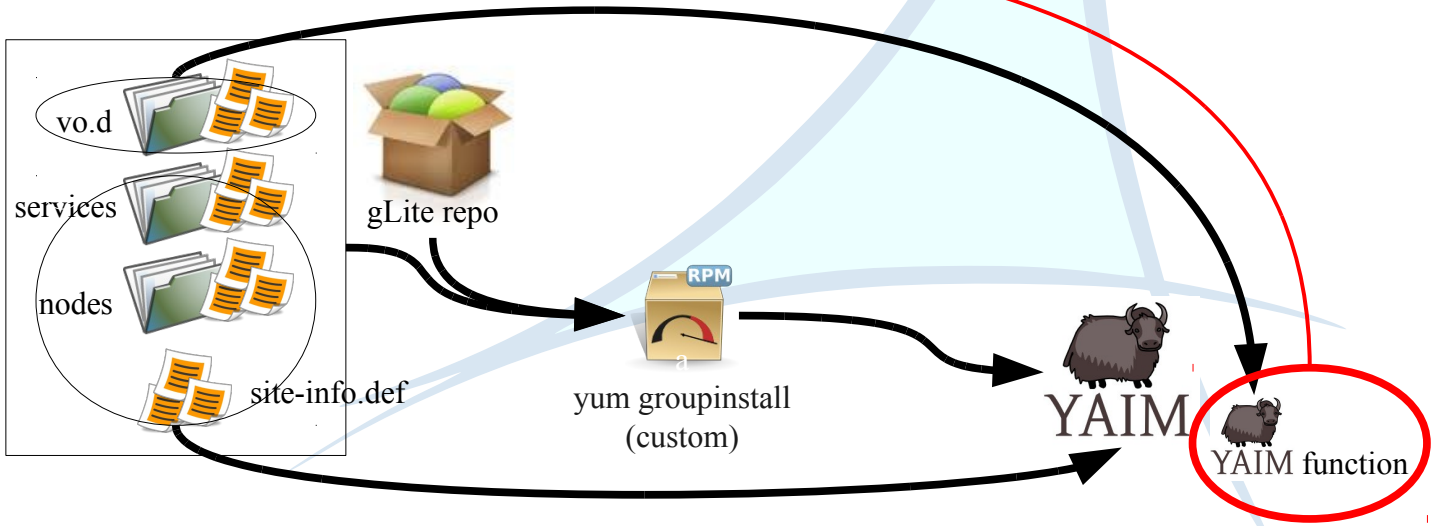
```
#### Run entire YAIM node configuration
define run_yaim_node() {
    exec { "run_yaim_node_$name" :
        command => "/opt/glite/yaim/bin/yaim -c -s $yaim_location/site-info.def -n $name",
        refreshonly => true,
    }
}
run_yaim_node { $yaim_nodetype: }
```

```
### case "$nodetype" {
###    "fta":{ $yaim_nodetype = "FTA2" }
###    "fts":{ $yaim_nodetype = "FTS2" }
###    # ...
###    "wn": { $yaim_nodetype = [ "glite-WN",
                "TORQUE_client","glite-GLEXEC_wn" ] }
### }
```



Puppet in production: YAIM module alternatives

```
#### Run a single YAIM function. Condition: service must have this function
    define run_yaim_function_vomsdir() {
       case "$config_vomsdir" {
          "yes": {
             exec { "run_yaim_function_vomsdir_$name" :
                command     => "/opt/glite/yaim/bin/yaim -r -s $yaim_location/site-info.def -f
config_vomsdir -n $name",
                refreshonly => true,
             }
          }
       }
    }
    run_yaim_function_vomsdir { $yaim_nodetype: }
```



PIC
port d'informació
científica

3. Conclusions

- Dramatic reduction in service administration loads
- Standardization of service profiles
- Possibility of full site homogeneization
- Fast disaster recovery capability when combined with streamlined installation system (ie. kickstart)
- Time invested in maintaining a puppet infrastructure is negligible when compared with the gain
- High flexibility, hence fast integration of new projects/requirements
- Abstraction level used allows sysadmins to deal with all services

PIC
port d'informació
científica

Contact us

- PIC Puppet Team

  puppet@pic.es

- Services Department

  services@pic.es

- PIC Web Page

  www.pic.cat

PIC
port d'informació
científica

Backup Slides: Automation tools evaluation

	Quattor	cfEngine	Puppet
Flexibility	-	+	+
Config. Control	+++	+++	+++
Complexity	+++	+	+
Gradual Integration	+	++	++
Documentation + Support	+	++	+++
Supported O.S.	-	++	+
Execution Speed	+	++	+