



# KermIT

An IT management solution with a WebUI frontend over  
MCollective

Louis Coilliot - Marco Mornati

`http://www.kermit.fr`

`http://www.think.fr`



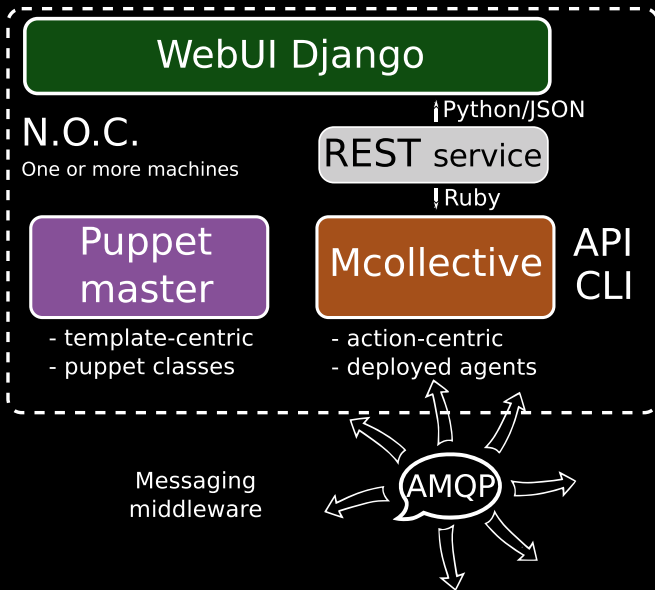
KermIT integrates best of breed opensource components with a WebUI

- ▶ Puppet
  - ▶ template-centric
  - ▶ deployment of the solution
  - ▶ configuration templates
  - ▶ categorization of managed nodes
- ▶ MCollective
  - ▶ action-centric
  - ▶ communication between the nodes and the NOC(s)
  - ▶ orchestration tool
- ▶ a WebUI developed with the Django web python framework and some AJAX
- ▶ a REST server for communication b/t frontend & backend  
loosely coupled, language agnostic and resilient with Phusion Passenger



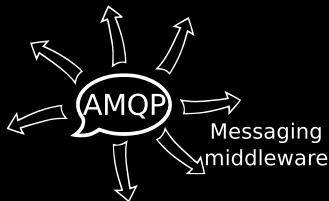
## Why MCollective?

- ▶ encryption of communications with the SSL+AES plugin
- ▶ authentication with standard plugins
  - i.e. asymmetric keys with SSL
- ▶ very scalable with the AMQP messaging middleware
  - ▶ works fine with RabbitMQ (Erlang) or ActiveMQ (Java)
  - ▶ clustering of the brokers
    - ▶ for HA
    - ▶ for replication of messages across datacenters and bastions
      - We aggregate data directly at the messaging level, not in a database
- ▶ very flexible and reusable framework
  - i.e. custom queues using the mc transport wrappers and security mechanisms



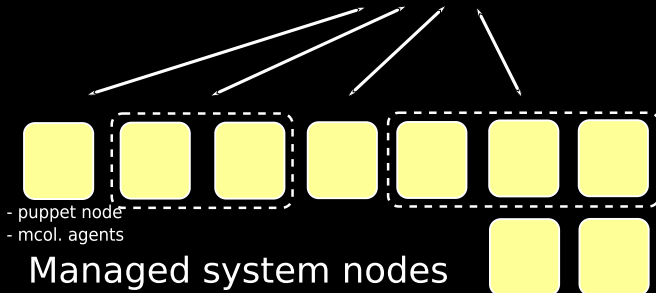


NOC



**filters :**

- puppet classes
- facts
- identity
- compound





Packages provided and framework tested on :

- ▶ RHEL 4 i386/x86\_64 (managed node)
- ▶ RHEL 5 i386/x86\_64 (managed node or NOC)
- ▶ RHEL 6 i386/x86\_64 (managed node or NOC)
- ▶ AIX 6.1 (managed node)

Should work fine also on Fedora 15

Should be packaged for other Linux distributions without much problems

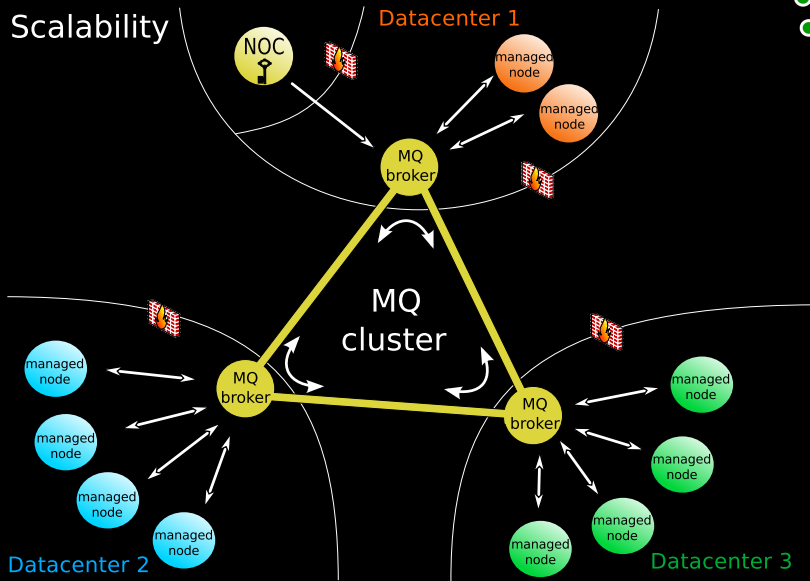


## Scalability

- ▶ AMQP messaging middleware
  - "sub-millisecond latency in transient mode under a load of 10k messages/s"
- ▶ AMQP cluster for multiple datacenters
- ▶ add consumers ('workers') to the specific queues
- ▶ REST server with Sinatra through mod\_passenger in Apache
  - Passenger designed for performance, stability and security
- ▶ install the components on separate systems
  - ▶ Puppet master(s) synchronized with git (for example)
  - ▶ REST server
  - ▶ MCollective management node(s)
  - ▶ AMQP broker



# Scalability







## Security

- ▶ WebUI
  - ▶ served with Apache and mod\_ssl
  - ▶ various authentication plugins
    - in-built, saml2 ; more to come
- ▶ MCollective SSL plugin
  - public/private key based message signing and identification of mgt clients
- ▶ MCollective AES plugin
  - payload encryption, identification of servers and optional key distribution
- ▶ compliant with SELinux

## Extensibility



- Few lines of code are required to add a new agent or action.

# Example :

```
action "remove" do remove end
private
def remove
  file = get_filename
  if ! File.exists?(file)
    logger.debug("#{file}' doesn't exist")
    reply.statusmsg = "OK"
  end

  begin
    FileUtils.rm(file)
    logger.debug("Removed file '#{file}'")
    reply.statusmsg = "OK"
  rescue
    logger.warn("Could not remove file '#{file}'")
    reply.fail! "Could not remove file '#{file}'"
  end
end
```



## Extensibility (Cont'd)

### Agent definition (DDL) for

- ▶ help
- ▶ validation of inputs and outputs
- ▶ configuration settings
- ▶ agent inventory
- ▶ auto generation of UI



# Example :

```
action "remove", :description => "Removes a file" do
  input :file ,
        :prompt      => "File ",
        :description => "File to remove",
        :type         => :string ,
        :validation  => '^.+$' ,
        :optional     => false ,
        :maxlength    => 256
end
```



## Autogeneration of the UI form :

The screenshot displays the KermIT web interface. On the left, a tree view shows the directory structure: **Home** > **retail** > **proddotcom** > **development** > **zen** > **oas**. A context menu is open over the **oas** folder, listing **Agents**, **Operations**, **Deployment**, and **Deploy SQL**. Below this, the **Agents Available** section lists various agents: **rpcutil**, **curb**, **service**, **system**, **a7xinventory**, **package**, **nodeinfo**, **puppetagent**, and **filemgr**. The main panel shows a table of agents with columns **Online**, **Hostname**, and **Operating System**. The table contains one entry: **el6** (Online). A context menu is open over the **filemgr** agent, listing **status**, **touch**, and **remove**. A modal dialog titled **filemgr-remove execution...** is displayed, showing the parameters required to execute the **filemgr-remove** operation. The dialog includes a **File:** input field and an **Execute** button.

**KermIT**  
Infrastructure as Code (IaC) Management Tool

Home

- retail
  - proddotcom
    - development
      - zen
        - oas
          - Agents
          - Operations
          - Deployment
          - Deploy SQL

Showing 1 t

Agents Available

- rpcutil
- curb
- service
- system
- a7xinventory
- package
- nodeinfo
- puppetagent
- filemgr

Results

Online	Hostname	Operating System
Online	el6	

Showing 1 t

Results

- status
- touch
- remove

filemgr-remove execution...

Parameters required to execute filemgr-remove

File:

Execute