

# Marionette Collective

Resistance is Futile...and stuff.



Gary Larizza

Puppet Labs  
@glarizza



# What is MCollective

- Framework for System Management
- Orchestration



# What is MCollective

- Framework for System Management
- Orchestration





*“MCollective is like a chainsaw, taped to a chainsaw, swinging in the air.”*



R.I Pienaar - <http://github.com/ripienaar>

# What Problems Does This Solve?

- System Discovery
- Inventory Collection
- Task Execution
- Configuration/Package Management Steroids

# WHY MCollective?

- No **centralized** inventory
- Thousands of Nodes
- Framework for Creation
- Asynchronous



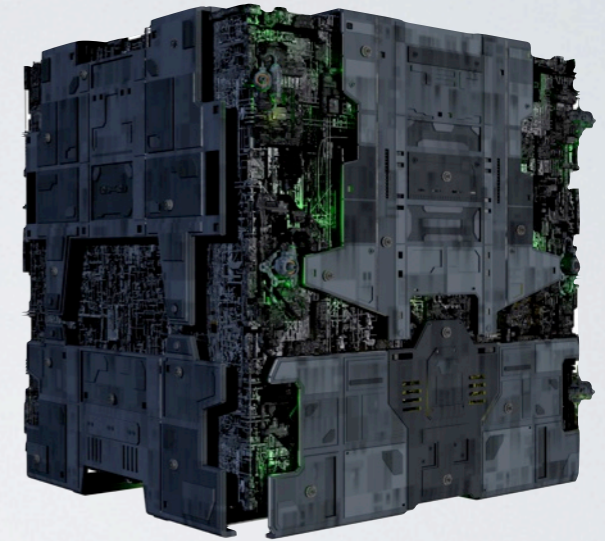
# Many Servers - One Client

- Servers: PROVIDE data
- Client: RECEIVES information/data





# The Big Black Box?



Message Queue

MCollective

Client

RPC

Topic

Middleware

Broker

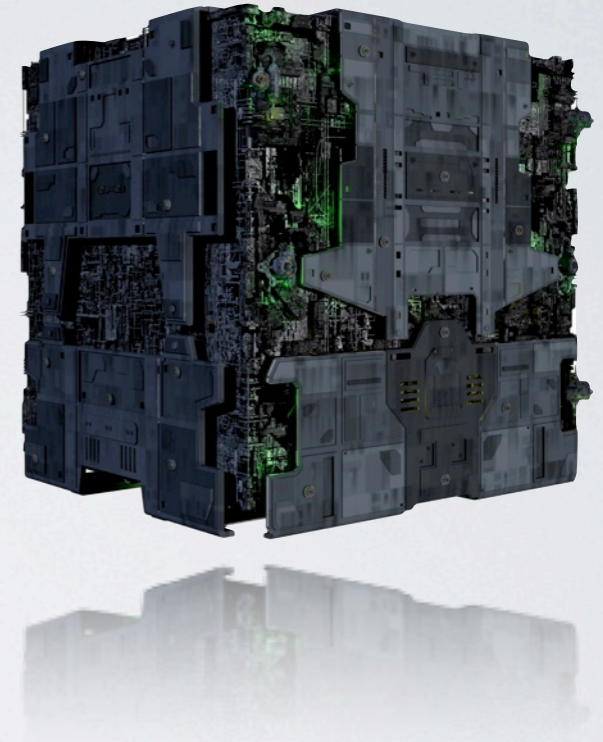
Agent

Collective

Server

# The Big Black Box?

- Sender
- Receiver
- Request (The Message)
- Message Transfer
- Message Storage
- Response (Also A Message)

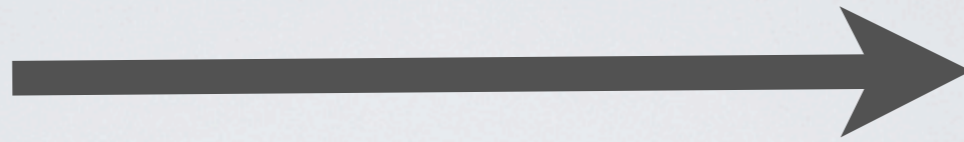


# How Does it Work?

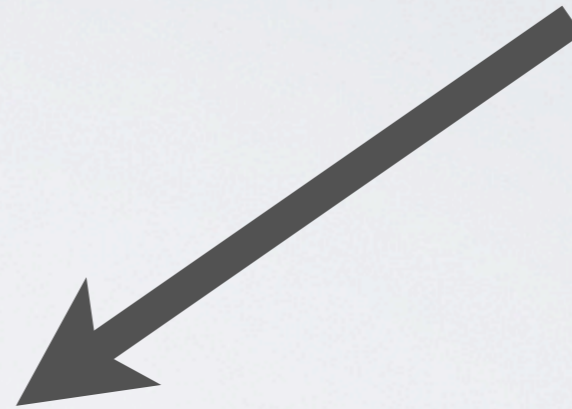
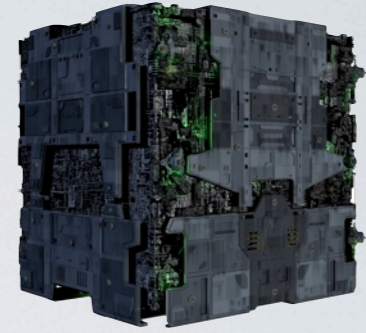
(in the simplest fashion ever, please)



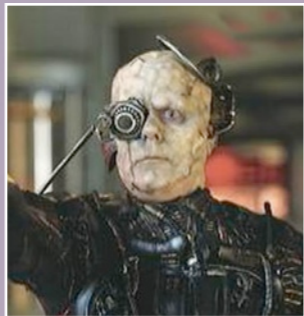
**‘client’**



**Message Queue**



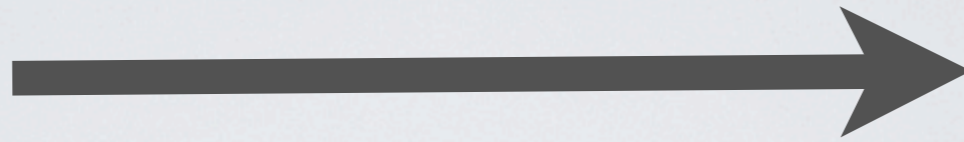
**‘servers’**



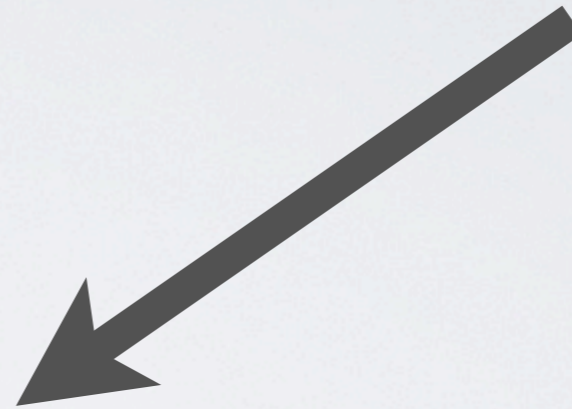
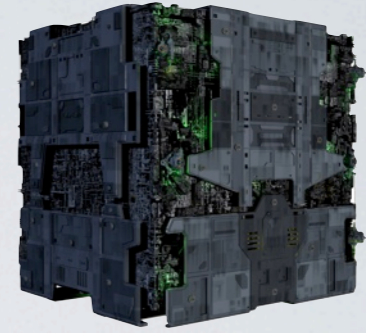
**Collective**



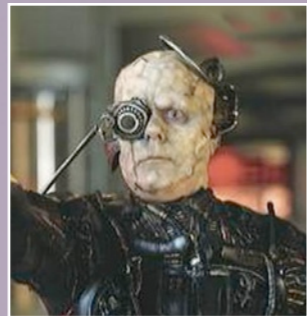
**‘client’**



**Message Queue**



**‘servers’**

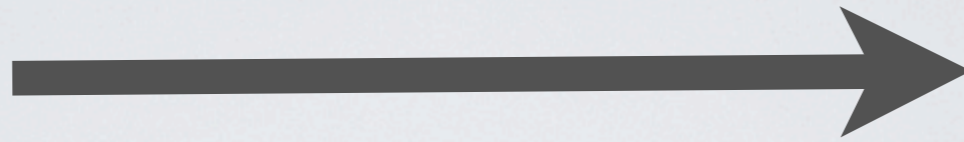


**Collective**

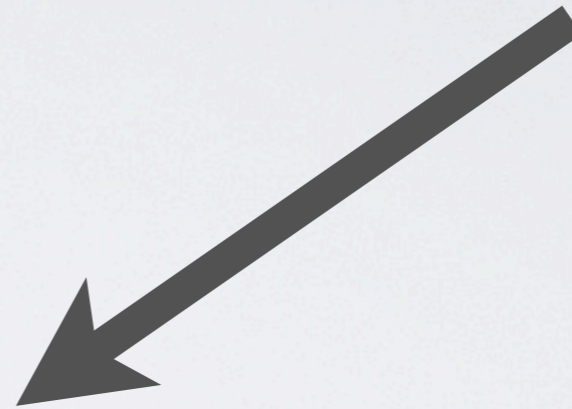
`mco ping --with-fact sevenofnine=true`



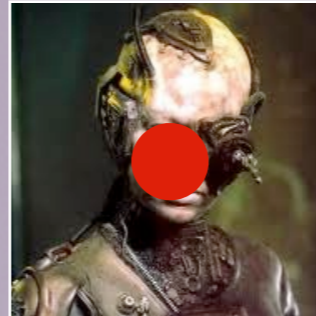
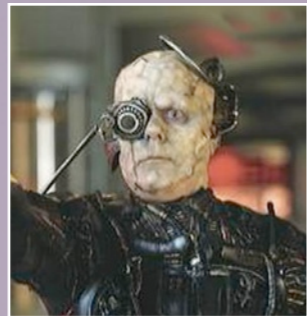
**‘client’**



**Message Queue**



**‘servers’**

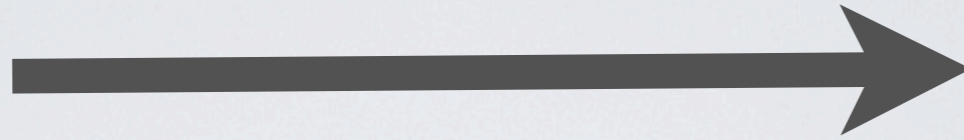


**Collective**

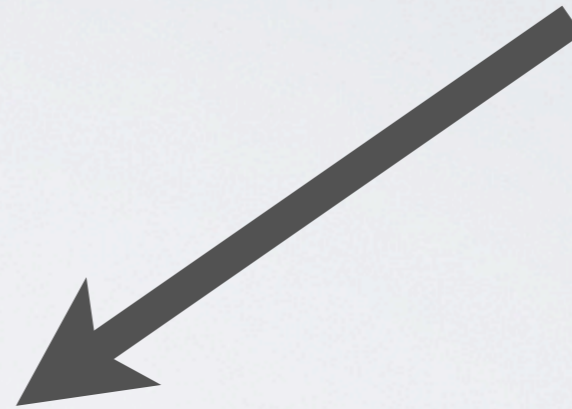
`mco ping --with-fact sevenofnine=true`



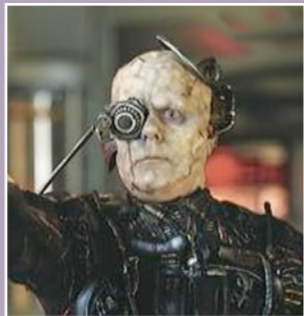
**‘client’**



**Message Queue**



**‘servers’**



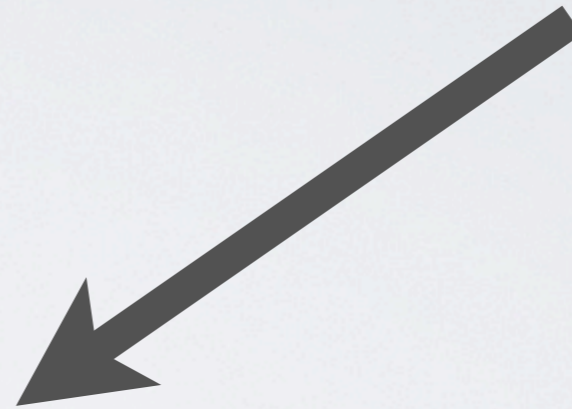
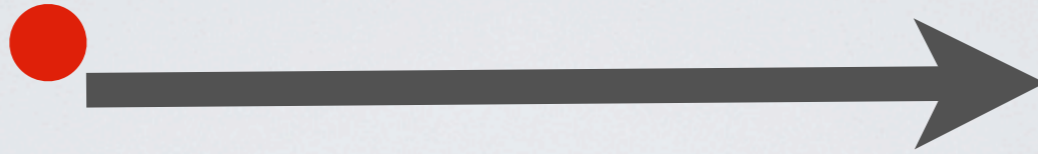
**Collective**

`mco ping --with-fact sevenofnine=true`

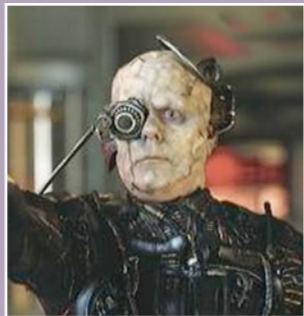


**‘client’**

**Message Queue**



**‘servers’**



**Collective**

`mco ping --with-fact sevenofnine=true`



# What is Middleware?

- The Software that handles communication between services.
- The SERVICE that transfers the 'request' and the 'response'.
- No logic ITSELF - like TCP/IP
- SMTP Analogy: Think of it as a mail server with routing rules.
- IRC Analogy: The IRC server



# What's a Message Queue?

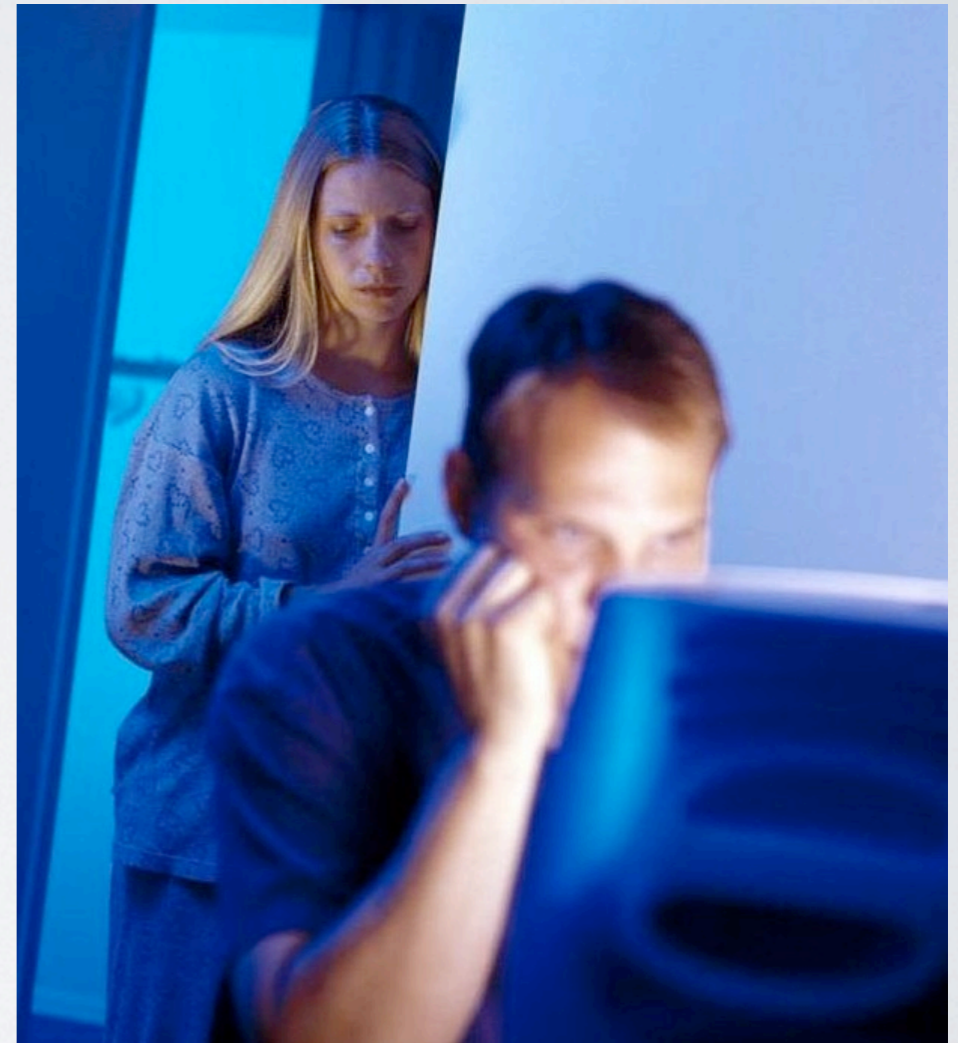
- Data store that applications can write-to or read-from.
- SMTP Analogy: Like the mail queue on your smtp server.
- The individual chat room in the IRC analogy.



We're British;  
we queue!

# So what is MCollective?

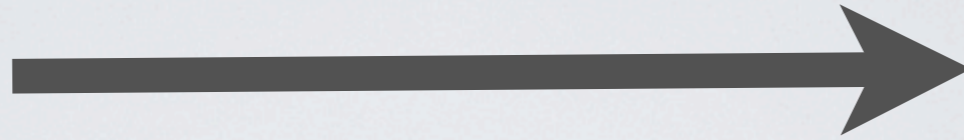
- The software that sends/receives messages over the queue
- ‘Agents’ perform actions on the system based on message content
- IRC Analogy: People in the Chatroom
- SMTP Analogy: Your Email Client and You



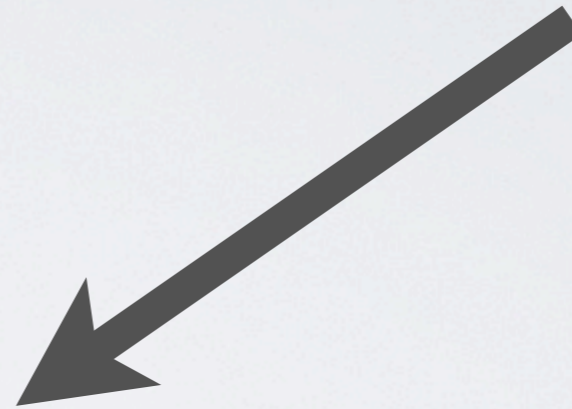
# Agents?

- Agents execute commands on all your 'Server' nodes
- The response from the agent is returned to the 'Client'
- MCollective provides a built-in filtering mechanism

```
home$ rm -Rf /
```

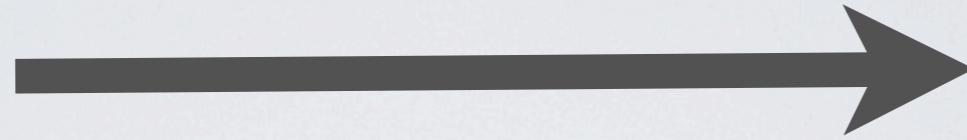


**Message Queue**

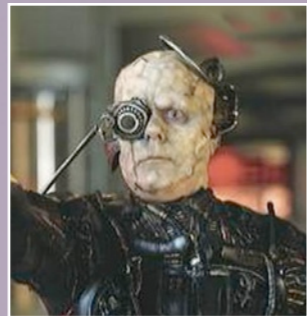
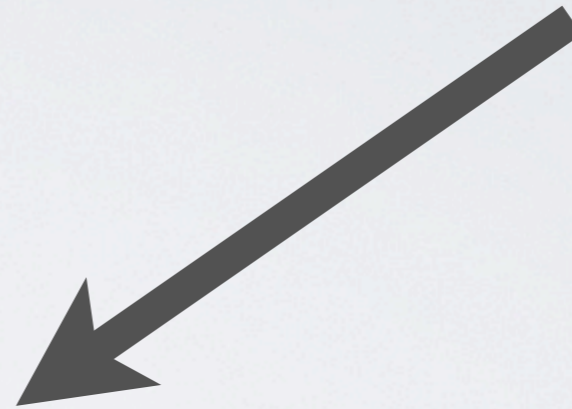




**Collective**

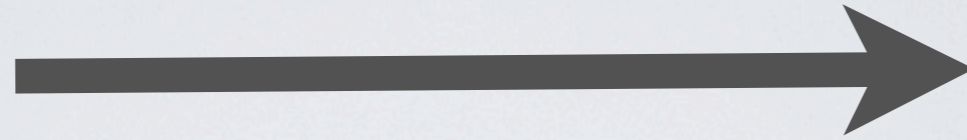


**Message Queue**

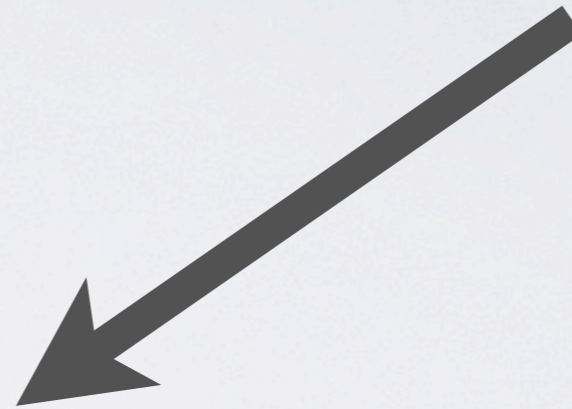
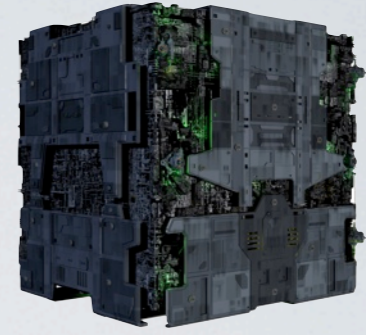



**Collective**

mco rpc sayhello -F sevenofnine=true



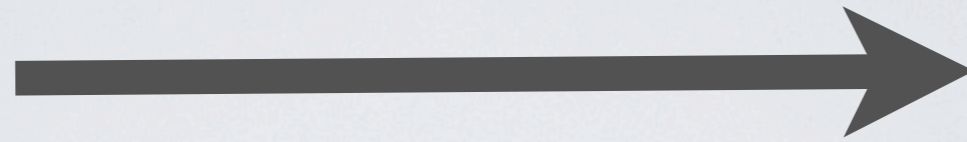
**Message Queue**



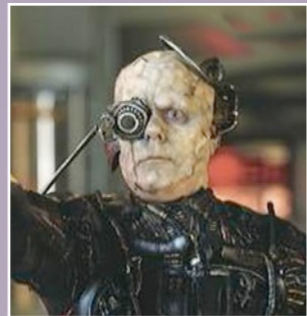
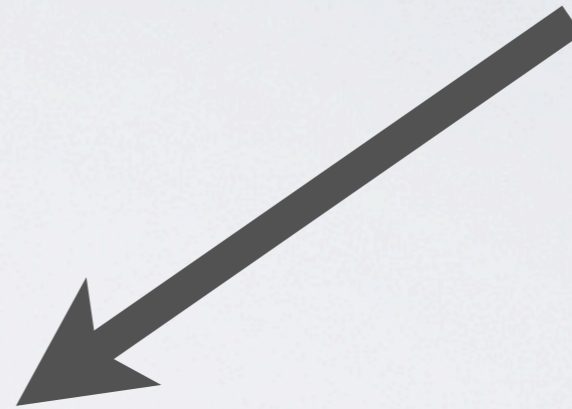
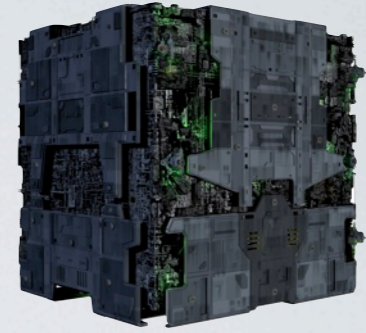


**Collective**

mco rpc sayhello -F sevenofnine=true



Message Queue

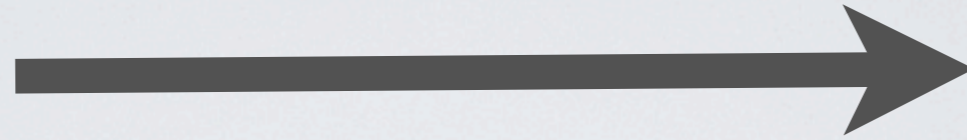


Collective

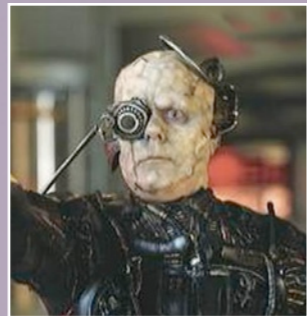
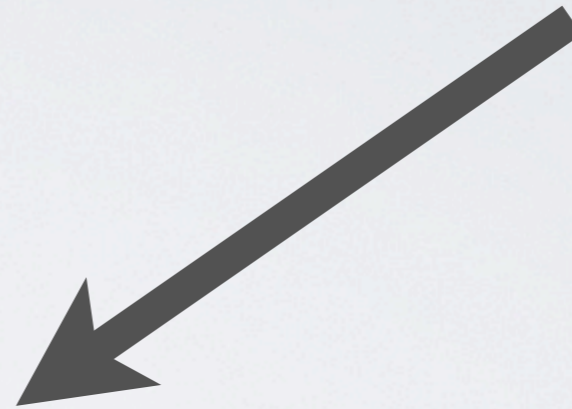
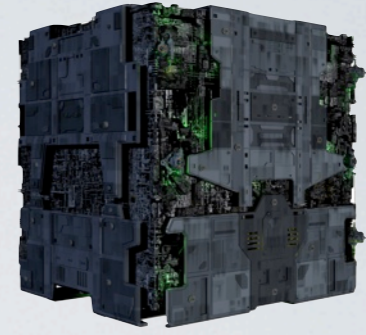
`/usr/bin/say Resistance is Futile!`

`mco rpc sayhello -F sevenofnine=true`





Message Queue



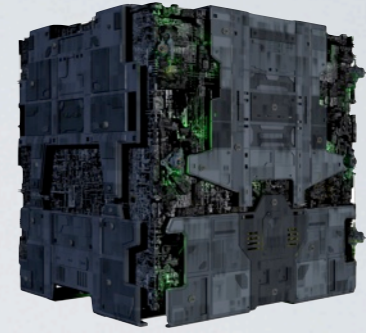
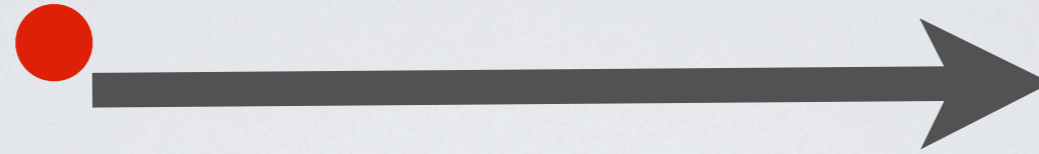
Collective

`/usr/bin/say Resistance is Futile!`

`mco rpc sayhello -F sevenofnine=true`



# Message Queue



`/usr/bin/say Resistance is Futile!`

`mco rpc sayhello -F sevenofnine=true`

# Writing Agents

## Simple (as) RPC

- ‘SimpleRPC’ is the framework we use for creating agents
- Abstracts common/difficult tasks (discovery, filtering)
- Agents written in Ruby
- Feel free to Shell-out as need be...or not

# Simple Agent

```
module MCollective
  module Agent
    class Printer < RPC::Agent
      metadata :name           => 'printer',
               :description    => 'Printers and printing',
               :author         => 'Gary Larizza',
               :license        => 'BSD',
               :version        => '0.1',
               :url            => 'http://puppetlabs.com',
               :timeout        => 100

      action 'list' do
        reply[:output] = `lpstat -a | cut -d ' ' -f 1`.split("\n")
      end
    end
  end
end
```

# Simple Agent

```
module MCollective
  module Agent
    class Printer < RPC::Agent
      metadata :name           => 'printer',
              :description    => 'Printers and printing',
              :author         => 'Gary Larizza',
              :license        => 'BSD',
              :version        => '0.1',
              :url            => 'http://puppetlabs.com',
              :timeout        => 100


      action 'list' do
        reply[:output] = `lpstat -a | cut -d ' ' -f 1`.split("\n")
      end
    end
  end
end
```

# Simple Agent

```
module MCollective
  module Agent
    class Printer < RHC::Agent
      metadata :name           => 'printer',
              :description    => 'Printers and printing',
              :author         => 'Gary Larizza',
              :license        => 'BSD',
              :version        => '0.1',
              :url            => 'http://puppetlabs.com',
              :timeout        => 100

      action 'list' do
        reply[:output] = `lpstat -a | cut -d ' ' -f 1`.split("\n")
      end
    end
  end
end
```

**Agent Name**



# Simple Agent

```
module MCollective
  module Agent
    class Printer < RHC::Agent
      metadata :name           => 'printer',
               :description    => 'Printers and printing',
               :author         => 'Gary Larizza',
               :license        => 'BSD',
               :version        => '0.1',
               :uri            => 'http://puppetlabs.com',
               :timeout        => 100

      action 'list' do
        reply[:output] = `lpstat -a | cut -d ' ' -f 1`.split("\n")
      end
    end
  end
end
```

**Agent Name**

**Action Name**

# Simple Agent

```
module MCollective
  module Agent
    class Printer < RHC::Agent
      metadata :name           => 'printer',
               :description    => 'Printers and printing',
               :author         => 'Gary Larizza',
               :license        => 'BSD',
               :version        => '0.1',
               :uri            => 'http://puppetlabs.com',
               :timeout        => 100

      action 'list' do
        reply[:output] = `lpstat -a | cut -d ' ' -f 1`.split("\n")
      end
    end
  end
end
```

**Agent Name**

**Action Name**

**Command to be Run**



# Simple Agent

```
module MCollective
  module Agent
    class Printer < RPC::Agent
      metadata :name           => 'printer',
               :description    => 'Printers and printing',
               :author         => 'Gary Larizza',
               :license        => 'BSD',
               :version        => '0.1',
               :url            => 'http://puppetlabs.com',
               :timeout        => 100

      action 'list' do
        reply[:output] = `lpstat -a | cut -d ' ' -f 1`.split("\n")
      end
    end
  end
end
```

# Running an Agent

```
mco rpc printer list --with-fact location=HHS
```

**Agent Name**

**Filter**

**Action Name**

**Fact Filter**

# Output

Determining the amount of hosts matching filter for 2 seconds .... 89

```
* [ =====> ] 89 / 89
```

lab | 5-hsimaclab-hhs

```
result: ["psm_HHS_201 Lab",  
        "psm_HHS_Bookroom",  
        "psm_HHS_Media_Center",  
        "psm_HHS_Office_9040",  
        "psm_HHS_Office_9050"]
```

lab | 2-hsimaclab-hhs

```
result: ["psm_HHS_201 Lab",  
        "psm_HHS_Bookroom",  
        "psm_HHS_Media_Center",  
        "psm_HHS_Office_9040",  
        "psm_HHS_Office_9050"]
```

# MCollective Agents vs ARD?

- Node Discovery
- Node Selection/Filtering
- Command Execution
- Input Parsing
- Timeouts and Blocking

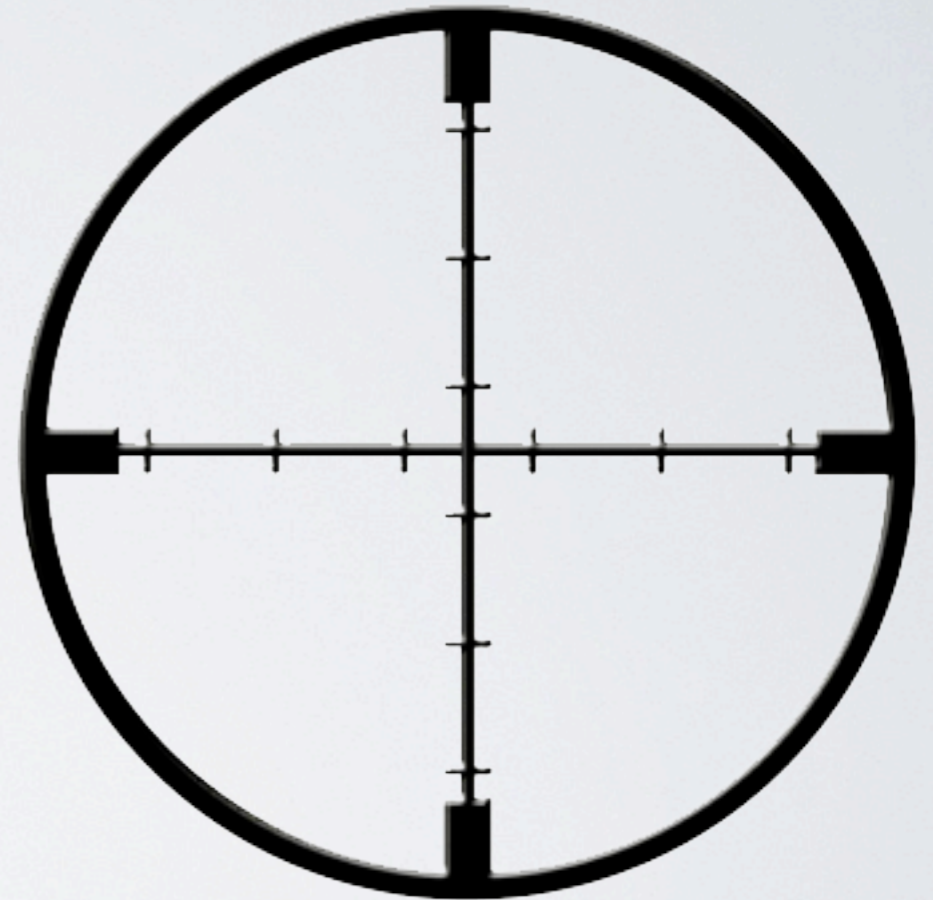


# Leverage the Power of Puppet

- Puppet NOT necessary for MCollective...but RAL is awesome!
- Use Resource Abstraction Layer:
  - Start/Stop Services
  - Query system resources (users, groups, packages, etc...)
  - Puppet NEED NOT be actively running!
  - Filter Nodes based on Puppet Classes

# Don't Forget Facter!

- Facter facts available as Metadata
- Filter nodes based on Facts
- Query fact values across collectives



# mco facts location

Report for fact: location

HHS	found 91 times
MJHS	found 8 times
SHEL	found 20 times
WIS	found 11 times

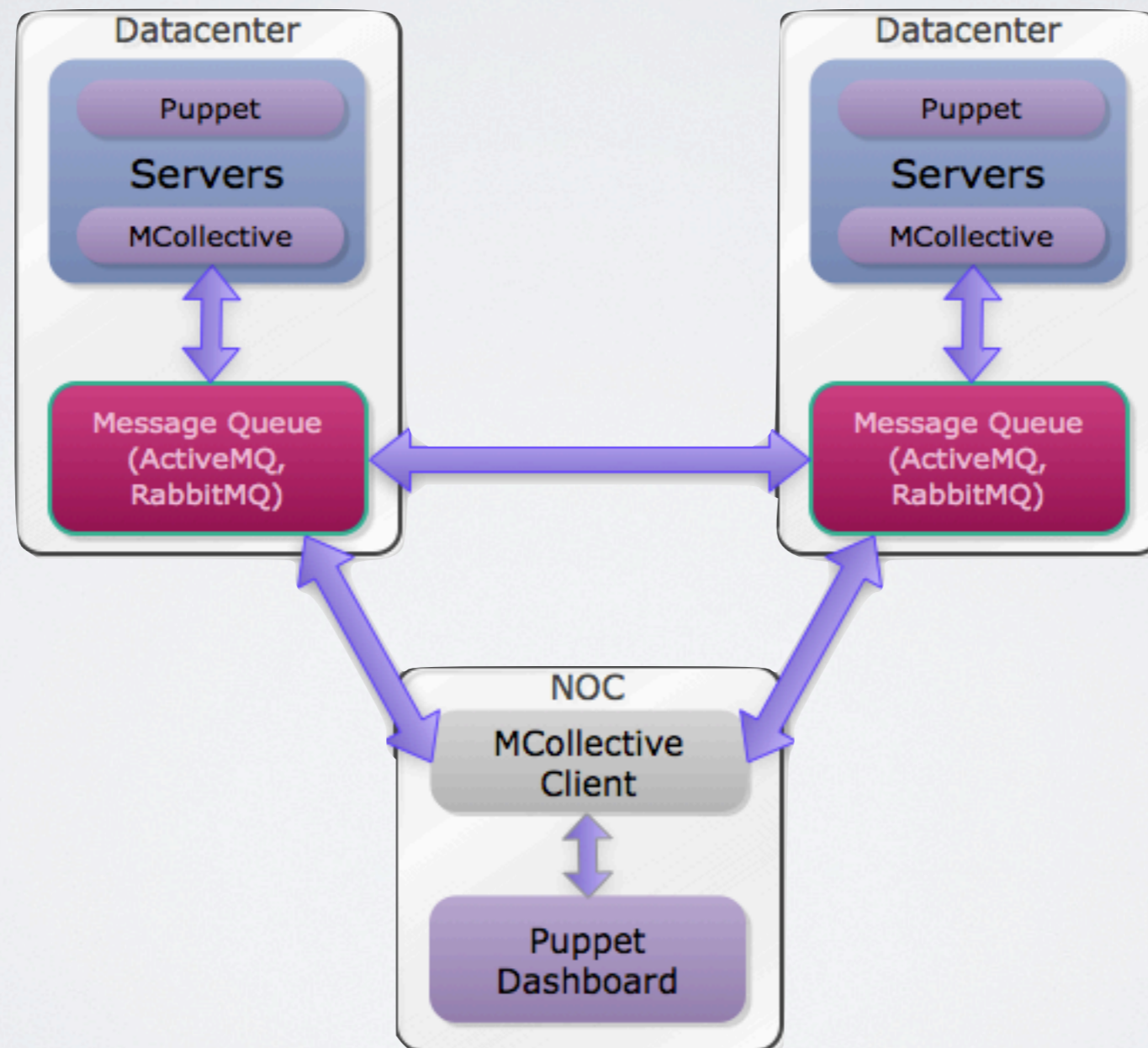
Finished processing 131 / 131 hosts in 3185.64 ms

# Centralized Inventory with Decentralization

- No CENTRAL data store for Inventory
- Use Facts/Metadata/Classes to compile list of nodes
- Query Warranty, Location, Purchase Date
- Export live data for archival (website, reports, wiki)



# Messaging across Locations



# Plug-able and Customizable!

- Plugins:
  - Custom Agents
  - Security
  - Facts
- MCollective is a platform on which we've shipped a set of rudimentary programs.



Let's Kick the Tires!

# Getting MCollective for Mac?

- bldmacpkg script
  - in the ext/osx directory in MCollective Source
- Luggage Recipe
  - <https://github.com/unixorn/luggage-examples/tree/master/mcollective>
- <http://bit.ly/mc-mac>
  - [Linked to my Dropbox](#)

# Getting Puppet for Mac?

- createpackage.sh script in conf/osx directory of Puppet Source
- Luggage Recipe
  - <https://github.com/unixorn/luggage-examples/tree/master/puppet>
- This goofy 'Cart' thing
  - <https://github.com/glarizza/cart-examples/tree/master/Puppet>

# Custom Agents

<https://github.com/glarizza/Custom-Agents>

Not the Official Analogy, but...

MCollective

Dashboard

Puppet

Facter

Hiera



Nigel says:  
Respect Copyright Laws!

MCollective

Dashboard

Puppet

Facter

Hiera





- @glarizza
- [gary@puppetlabs.com](mailto:gary@puppetlabs.com)
- <http://puppetlabs.com>
- 90% Meme Free



**Fig. 1: Portland Hipster (Latin: Hipsterius Portlandia)**

# Picture Citations

- [http://www.omgot.org/forums/cgi-bin/album/album.pl?photo=RicsGhost/matt\\_Avatar.JPG;photo\\_height=250](http://www.omgot.org/forums/cgi-bin/album/album.pl?photo=RicsGhost/matt_Avatar.JPG;photo_height=250)
- <http://www.imdb.com/media/rm1419216896/nm0422710>
- <http://www.yourcaketopper.co.uk/pages/meerkats-cake-topper.htm>
- <http://www.flickr.com/photos/43103276@N07/5063150491/in/photostream>
- <http://withfriendship.com/user/mithunss/voltron.php>
- <http://screenrant.com/voltron-movie-new-writer-pauly-30032/>

# Picture Citations

- <http://en.wikipedia.org/wiki/File:Norad-control-center.jpg>
- <http://www.fotothing.com/reemadreaming/photo/1a1fccd052d4d0207b1976b0ac49c451/>
- <http://knowyourmeme.com/memes/internet-husband#.TquCd5w7c44>
- [http://www.easyexport.us/cars-for-sale/SALVAGE\\_CERTIFICATE\\_BIOHAZARDOUS\\_CHEMICAL\\_2001\\_VOLVO\\_S60\\_21570521/image/](http://www.easyexport.us/cars-for-sale/SALVAGE_CERTIFICATE_BIOHAZARDOUS_CHEMICAL_2001_VOLVO_S60_21570521/image/)
- [http://apple.wikia.com/wiki/Apple\\_Remote\\_Desktop](http://apple.wikia.com/wiki/Apple_Remote_Desktop)