

# Usenix CM Summit 2012

[someara@opscode.com](mailto:someara@opscode.com)  
[www.opscode.com](http://www.opscode.com)  
[@someara](#)



# Promises, Lies, and Dry Run Mode



<http://www.flickr.com/photos/egoleonard/5634297866/>



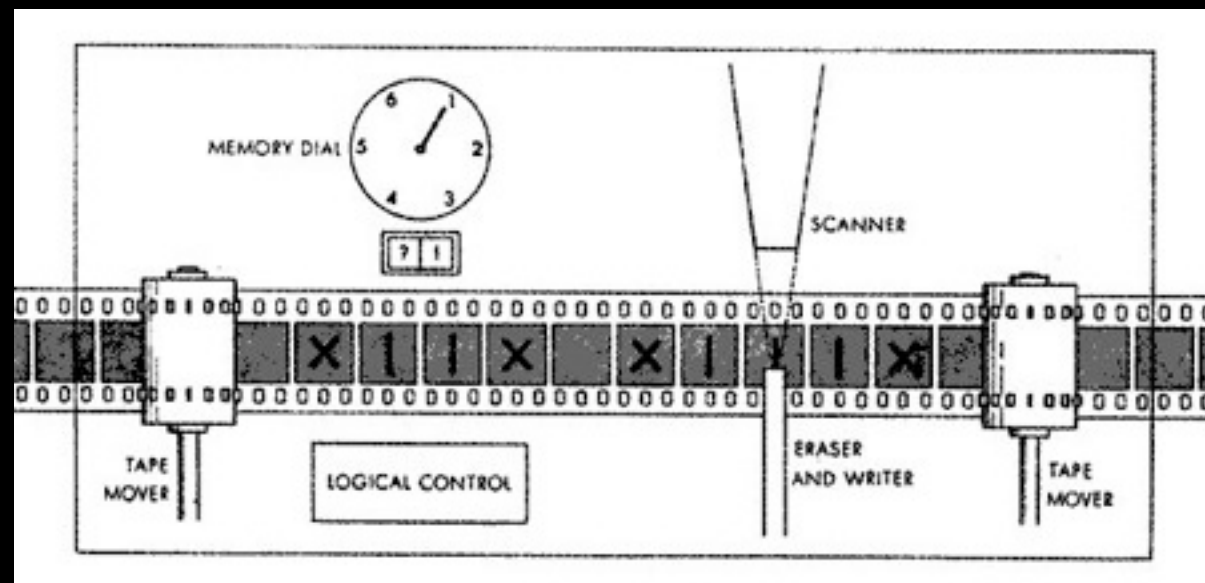
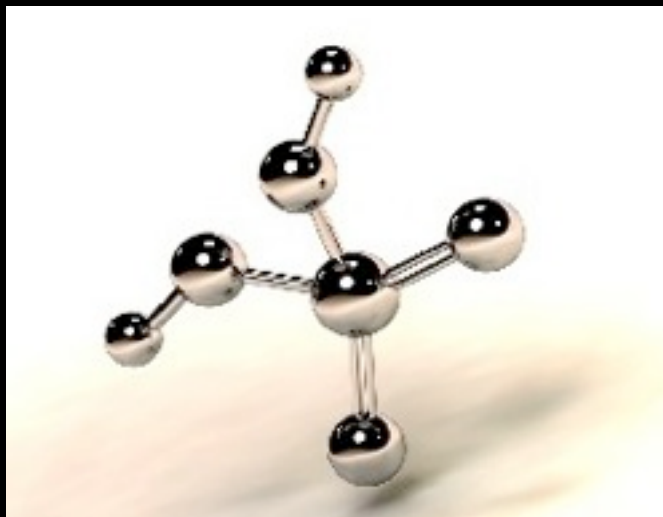
OPSCCODE



Chef



# ISConf



# Cfengine



# Puppet

# Chef

**chef-13**



**can plz has**

**dry run mode?**



**no**



**but wai?**



**can pretty plz has?**



because science



**but**

**i want it**





**no.**







**plz?**





ICANHASCHEEZBURGER.COM 🍪 🍪 🍪



# Convergence

<http://www.flickr.com/photos/dalbera/3009070876>



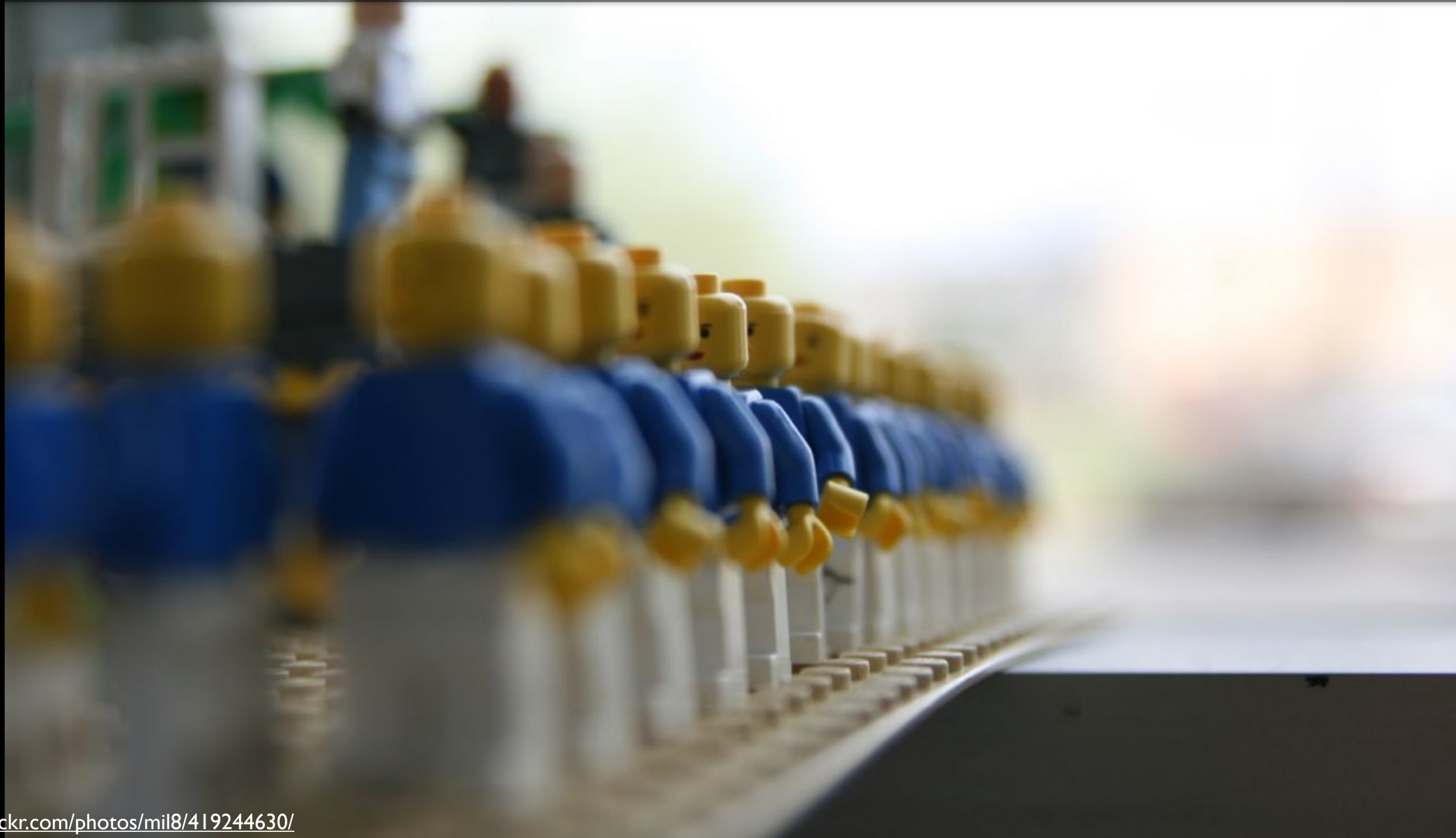
*A set of operators  $\mathcal{O}$*

Desired State -> Actual State



**Test for a condition.**  
**Take corrective action.**

# Sets of Sequences



<http://www.flickr.com/photos/mil8/419244630/>



```
<Base>  
  <Group name='fedora'>  
    <Package name='ntp' />  
    <Service name='ntpd' />  
    <ConfigFile name='/etc/ntp.conf' />  
  </Group>  
</Base>
```

```
bundle agent puppet_server
{
  vars:
    "debian_packages" slist => { "puppetmaster", "puppet" };

  packages:
    debian|ubuntu::
      "$(debian_packages)"
      comment => "install debian based distro packages",
      package_policy => "add",
      package_method => apt;

  processes:
    "/usr/bin/ruby /usr/sbin/puppetmasterd"
      comment => "the puppetmaster service",
      restart_class => canonify("restart_puppetmaster");
}
```



```
class chef-server {  
  
  package { 'chef-server':  
    ensure => installed  
  }  
  
  file { '/etc/chef/chef-server.rb':  
    ensure => present,  
    source => "chef-server.rb",  
    owner => "root",  
    mode => "644",  
    require => Package['chef-server']  
  }  
  
  service { 'chef-server':  
    ensure => running,  
    enable => true,  
    require => File['/etc/chef/chef-server.rb']  
  }  
  
}
```

```
cfengine_services = %w{ cf-execd cf-serverd }

package "cfengine" do
  action :install
end

directory "/var/cfengine/masterfiles" do
  action :create
end

template "/var/cfengine/inputs/promises.cf" do
  source "inputs/promises-server.cf.erb"
  notifies :restart, "service[cf-serverd]"
  notifies :restart, "service[cf-execd]"
end

cfengine_services.each { |s|
  service s do
    action [:enable, :start]
  end
}
```

# A Set of Promises

$\{O_1, O_2, O_3\}$

package

file

service



# Executed in a Sequence

$\{O_1, \dots, O_n\}$   
mount route execute user custom ...

# The Chef Resource Collection

```
run_list = [ 'role[a]', 'role[b]', 'recipe[a]' ]  
role = [ 'role[a]', 'recipe[a]', 'recipe[b]' ]
```

$\{O_1, \dots, O_n\}$   $\{O_1, \dots, O_n\}$   $\{O_1, \dots, O_n\}$   
recipe[a] recipe[b] recipe[c]

# A Set of Tests

test

test

test

$\{O_1, O_2, O_3\}$

action

action

action





# Why Run Mode

<http://www.flickr.com/photos/seatbelt67/502255276/>

**make -n**

# The Best You Can Do

test

test

test

$\{O_1, O_2, O_3\}$

noop

noop

noop



```
package "apache2" do
  action :install
end
```

```
template "/etc/apache2/conf.d/mahwebsite.conf" do
  owner "root"
  mode "0644"
  notifies :restart, "service[apache2]"
end
```

```
service "apache2" do
  action [:enable, :start]
end
```

```
template "aptconf" do
  path "/etc/apt/sources.opscode.conf"
  owner "root"
  mode "0644"
end

execute "update apt cache" do
  command "apt-get update"
  action :nothing
  subscribe :run, resources(:template => "aptconf")
end

package "ohai" do
  action :update
end
```

```
package "apache2"
```

```
template "/etc/apache2/apache2.conf"
```

```
mount "/srv/mahwebsite/"
```

```
directory "/srv/mahwebsite/conf.d/apache2.conf"
```

```
execute "clean_under_mount" do
```

```
  not_if ::File.exists?("/srv/mahwebsite/lock")
```

```
  command "rm -rf /srv/mahwebsite/*"
```

```
end
```



```
user "thepauge"
```

```
mount "/srv/gibson/"
```

```
file "/srv/gibson/garbage.txt" do  
  content lookup(evil_plans)  
  notifies :run, 'execute[hack the planet]'  
end
```

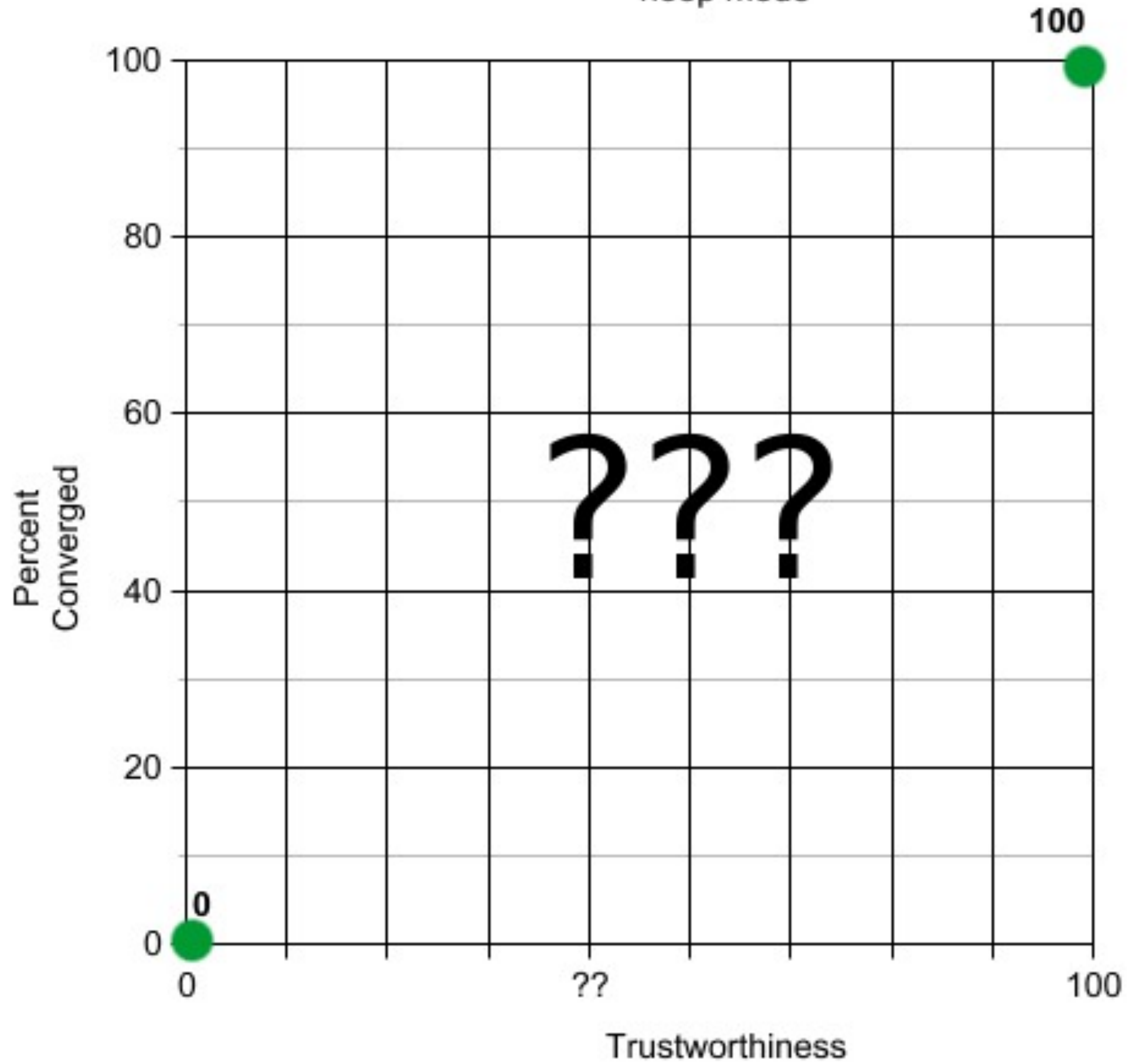
```
execute "hack the planet" do  
  command "cat garbage.txt | mail joey@aol.com"  
  cwd "/srv/gibson"  
  action :nothing  
end
```



# THE SYSTEM CHANGES BETWEEN TESTS

<http://www.flickr.com/photos/lentzstudios/4101127101/>

noop mode



# Further considerations

- \* Tests about dynamic aspects (CPU, Mem, Disk, etc)
- \* Tests about Infrastructure state (search for nodes, render config files)



A photograph featuring a cardboard Amazon box with a simple face drawn on it, a pink gift bow, and a green wrapped object, all set against a black background. The text "Publishing Assumptions" is overlaid in white.

# Publishing Assumptions

```
package 'apache2' do
  action :install
end
```

## **package[apache2]**

desired state is installed, current state is not installed  
would execute install action

```
service 'apache2' do
  action :start
end
```

## **service[apache2]**

```
WARNING: expected init script in /etc/init.d/apache2 does not exist
WARNING: assuming init script would have been previously installed.
WARNING: assuming current state of service is down
desired state is running, current state is down
would execute start action
```



```
template '/etc/apache2/fake_config.conf' do
  owner "root"
  group "root"
  mode  "0644"
  action :create
  notifies :restart, "service[apache2]"
end
```

### **template[/etc/apache2/fake\_config.conf]**

```
WARNING: enclosing directory /etc/apache2 does not exist
WARNING: assuming directory /etc/apache2 would have been previously created
desired state is content-sha(830ab1...) current state is nonexistent
would execute create action
would notify service[apache2] with restart action
```



## **service[apache2] (notified)**

```
WARNING: expected init script in /etc/init.d/apache2 does not exist
WARNING: assuming init script would have been previously installed.
WARNING: assuming current state of service is down
desired state is restarted, current state is down
would execute restart action
```

# Updated Template

---

When the file exists, but the content changes:

## template[/etc/sudoers]

```
desired state is content-sha(f0d965...) current state is content-sha(ede78b)
content changes:
```

```
--- /etc/sudoers      2011-11-03 12:51:53.000000000 -0700
+++ /tmp/tempfile-sudoers  2012-03-15 16:09:30.000000000 -0700
@@ -32,7 +32,7 @@
```

```
# User privilege specification
```

```
root    ALL=(ALL) ALL
```

```
-%admin ALL=(ALL) ALL
```

```
+%admin ALL=(ALL) NOPASSWD: ALL
```

```
# Uncomment to allow people in group wheel to run all commands
```

```
# %wheel    ALL=(ALL) ALL
```

```
would execute create action
```

```
template '/etc/sudoers' do
  owner "root"
  group "root"
  mode  "0440"
  action :create
end
```



# Use Cases





# Verification





# Boolean AND

test

test

test

$\{O_1, O_2, O_3\}$

noop

noop

noop

**YES/NO**



# Testing



**“I want to know what  
this would do to my  
production system  
before running it.”**





417

Expectation Failed



**406**  
**Not Acceptable**

**TEST PROPERLY**



# Development Workflows







Be careful out there!

# Questions?

[someara@opscode.com](mailto:someara@opscode.com)  
[www.opscode.com](http://www.opscode.com)  
[@someara](#)

