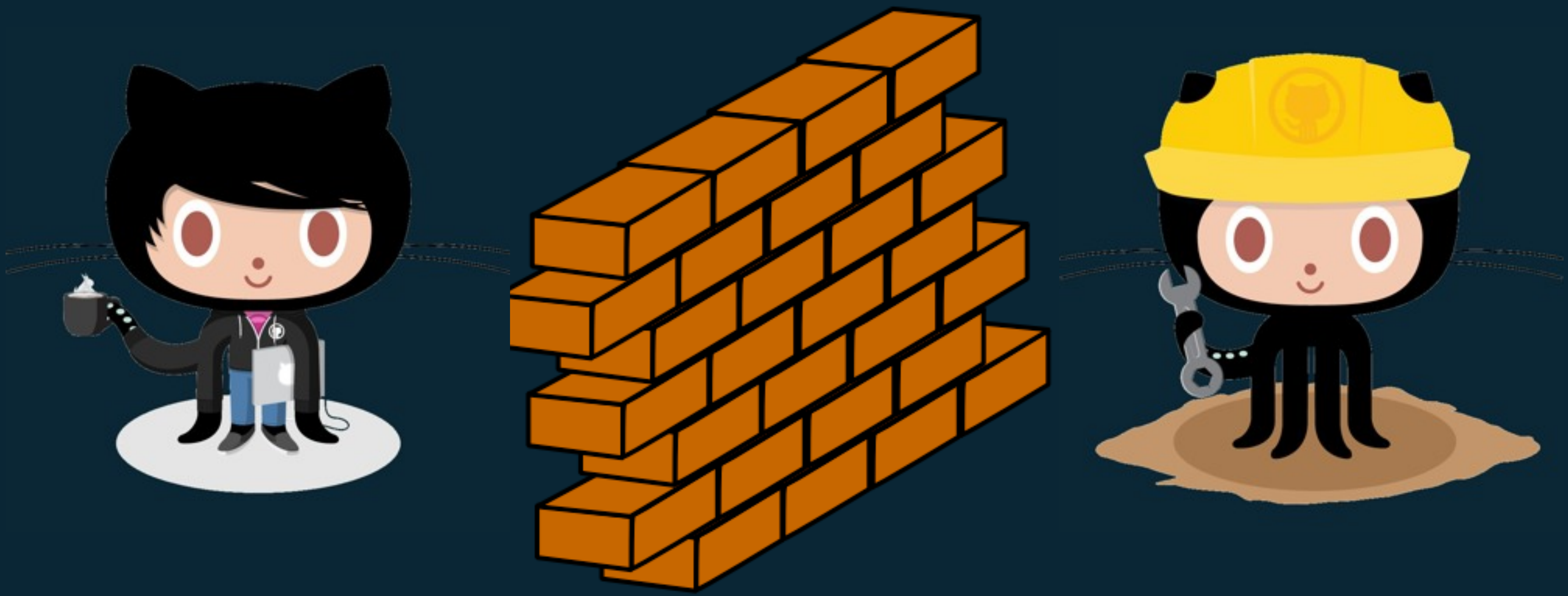
DevOps & Chef

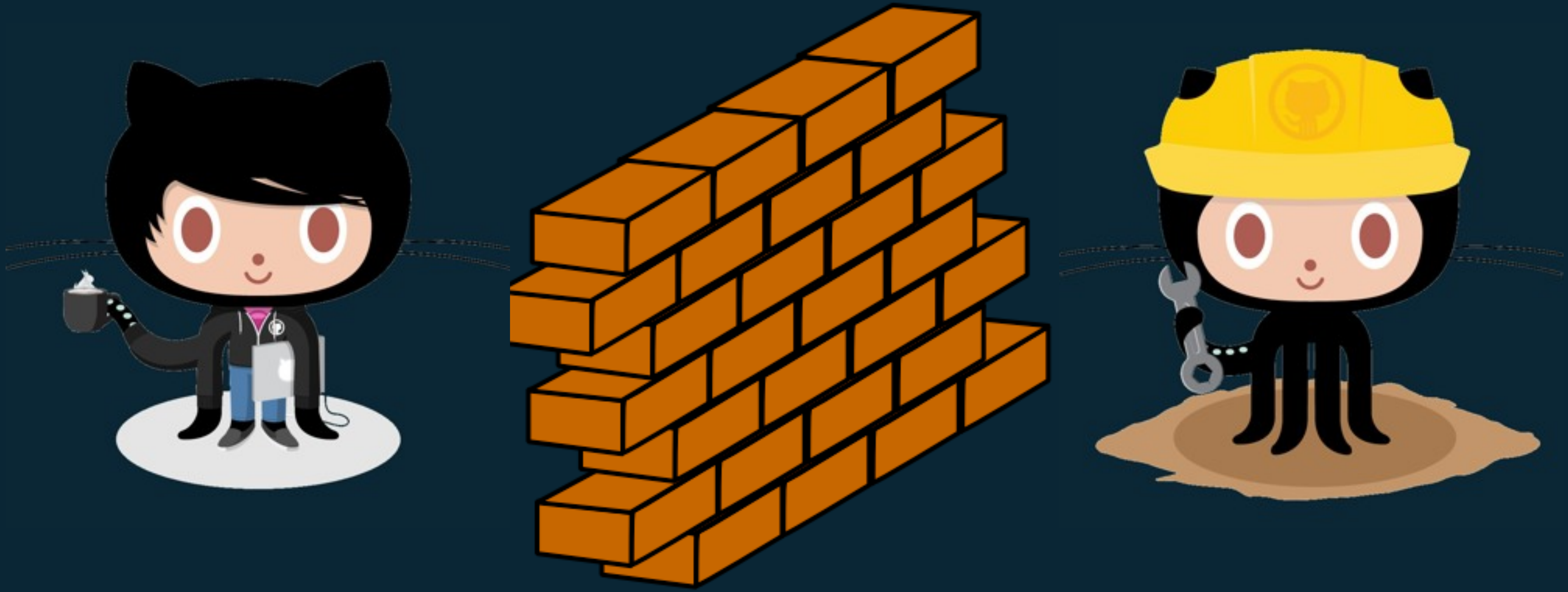
DevOps

Devs vs Ops

Devs vs Ops



Devs vs Ops



The Wall Of Confusion

Change

vs

Stability

A Meeting of Minds

Follow code into the wild

Accept change: become agile

Buzzword: DevOps

Buzzword: DevOps

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword: DevOps

Buzzword

Buzzword

Buzzword

Buzz Lightyear

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword: DevOps

Buzzword

Buzzword

Buzzword Buzzword

Buzzword

Buzzword

Buzzword

Buzz Lightyear

Buzzword Buzzword

Buzzword Buzzword

Buzzword Buzzword

Buzzword Buzzword

Buzzword

Buzzword

Buzzword

Buzzword

Buzzword Buzzword

Buzzword

Buzz Aldrin

Buzzword Buzzword

Buzzword

What is DevOps?

What is DevOps?

What is DevOps NOT?

What is DevOps?

They are making a play for our jobs

What is DevOps NOT?

Devs want root

What is DevOps?

They are making a play for our jobs

What is DevOps NOT?

DevOps is not a job title, it’s a philosophy

Devs want root

What is DevOps?

They are making a play for our jobs

What is DevOps NOT?

DevOps is not a job title, it’s a philosophy

Devs want root

What is DevOps?

They are making a play for our jobs

What is DevOps NOT?

So many words

DevOps, for me

DevOps, for me:

Testing

DevOps, for me:

Testing

Monitoring

DevOps, for me:

Testing

Monitoring

Automation

DevOps, for me:

Testing

Monitoring

Automation

DevOps, for me:

Testing	Don’t release broken things
Monitoring	Know when things break
Automation	Replace/fix broken things instantly

SCCS:
Source Code Control System

SCCS:
Source Code Control System

1972

40 Years Ago!

Revision Control

Revision Control

Test Driven Development

Revision Control

Test Driven Development

Feature Branches

Revision Control

Test Driven Development

Feature Branches

Continuous Integration

Revision Control

Test Driven Development

Feature Branches

Continuous Integration

Staging Environments

Revision Control

Test Driven Development

Feature Branches

Continuous Integration

Staging Environments

AGILE

AGILE

AGILE OPS

AGILE OPS

Continuous Deployment

Feature Flags

Dark Launches

Automated Recovery

IMVU - 50+

Etsy - 25

Flickr - 10+

Instant Feedback

Rollbacks

Confidence

No fear of change

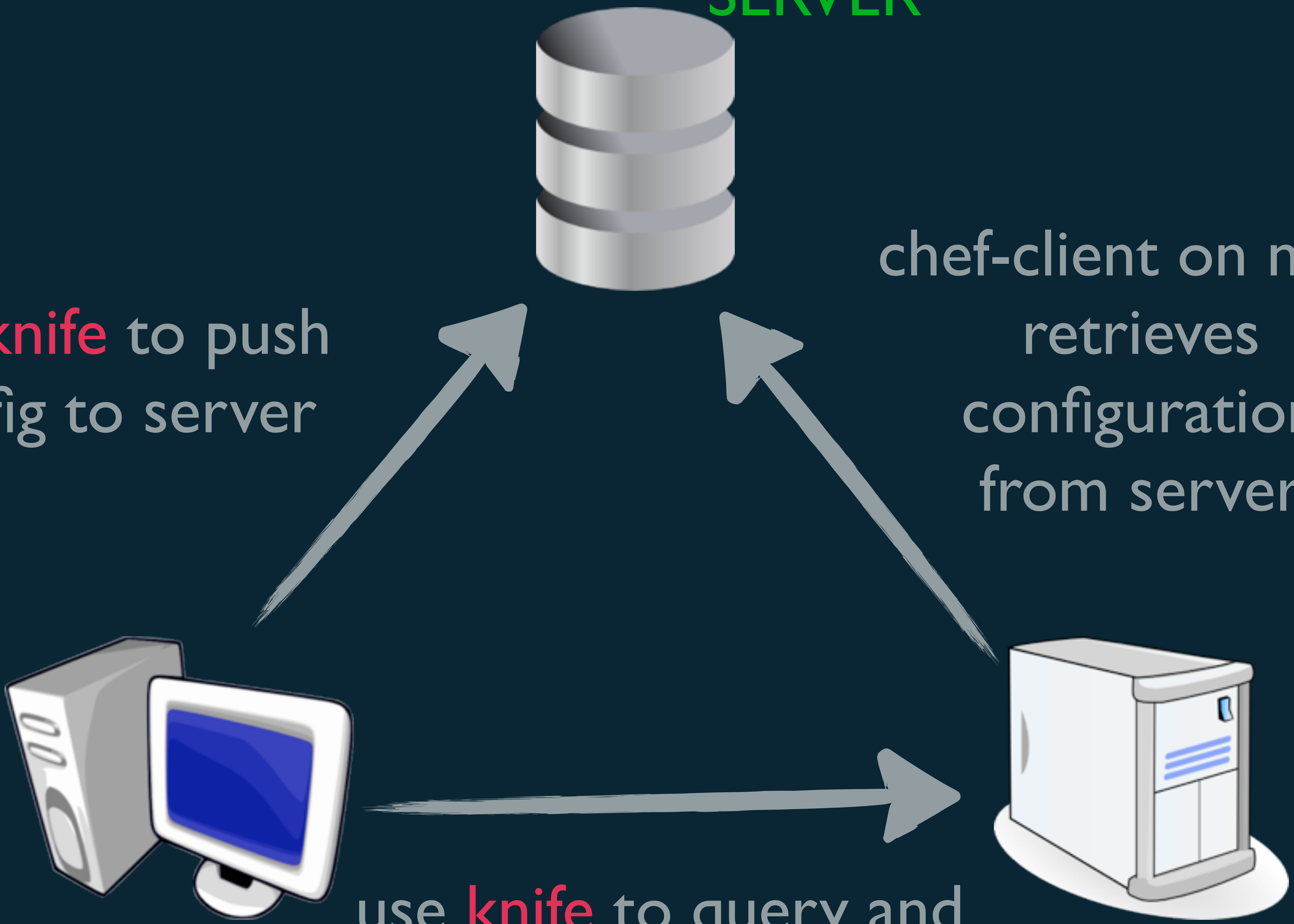
Code ↔ Infrastructure

Infrastructure ↔ Code

Infrastructure **As** Code

```
$ ssh root@web1
Last login: Tue Mar 13
# apt-get install apache2
...
# vim /etc/apache2/sites-enabled/default.conf
...
# /etc/init.d/apache2 restart
...
# scp web2:/root/.ssh/authorized_keys ~/.ssh
```

Chef



SERVER

use knife to push config to server

chef-client on node retrieves configuration from server

CLIENT (workstation)

use knife to query and ssh to clients

CLIENT (server/node)

Chef/nodes/web1.json

```
{
    "access" : [ "developers", "designers" ],
    "name": "web1",
    "run_list": [ "role[base]", "role[web]",
                  "recipe[splunk]" ]
}
```

```
{
    "access" : [ "developers", "designers" ],
    "name": "web1",
    "run_list": [ "role[base]", "role[web]",
                  "recipe[splunk]" ]
}
```

```
Chef/nodes/web1.json
Chef/roles/web.rb
```

```
run_list("recipe[apache]",
        "role[monitoring]")
```

```
run_list("recipe[apache]",
        "role[monitoring]")
```

```
Chef/nodes/web1.json

Chef/roles/web.rb

Chef/cookbooks/apache/
                recipes/default.rb
```

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

```
apache_modules = ['wsgi', 'ssl',
        'auth_tkt', 'proxy_http',
        'headers', 'rewrite', 'status']

apache_modules.each do |mod|
    apache_module mod do
        enable true
        notifies :restart,
              "service[apache2]"
    end
end
```

```
apache_modules = ['wsgi', 'ssl',
        'auth_tkt', 'proxy_http',
        'headers', 'rewrite', 'status']

apache_modules.each do |mod|
    apache_module mod do
        enable true
        notifies :restart,
              "service[apache2]"
    end
end
```

```
package "apache2" do
  case node[:platform]
  when "centos","redhat","fedora","suse"
     package_name "httpd"
  when "debian","ubuntu"
     package_name "apache2-mpm-prefork"
  end
  action :install
end
```

```
package "apache2" do
  case node[:platform]
  when "centos","redhat","fedora","suse"
     package_name "httpd"
  when "debian","ubuntu"
     package_name "apache2-mpm-prefork"
  end
  action :install
end
```

```
Chef/nodes/web1.json

Chef/roles/web.rb

Chef/cookbooks/apache/
           recipes/default.rb

           attributes/default.rb
```

```
default[:apache][:version] = "2.2"
```

```
package "apache2" do
  action :install
  version "#{node[:apache][:version]}"
end
```

```
Chef/nodes/web1.json

Chef/roles/web.rb

Chef/cookbooks/apache/
           recipes/default.rb

           attributes/default.rb

           templates/default/apache2.conf.erb
```

```ruby
template "#{node[:apache][:dir]}/apache2.conf" do
  source "apache2.conf.erb"
  owner "root"
  mode 0644
  notifies :restart, resources(:service =>
                             "apache2")
end
```

It’s all Ruby

It’s all Ruby

http://wiki.opscode.com/display/chef/Just+Enough+Ruby+for+Chef

“Just Enough Ruby for Chef”

It’s all Ruby

Let’s hack

```
Chef/nodes/web1.json

Chef/roles/web.rb

Chef/cookbooks/apache/
           recipes/default.rb

           attributes/default.rb

           templates/default/apache2.conf.erb

           libraries/default.rb
```

```ruby
class Chef
  class Recipe
    def apache_ports(&block)
      @node[:apache][:vhosts].each do |vh|
        vh[:ports].each do |p|
          block.call(p)
        end
      end
    end
  end
end
```

```
apache_ports do |p|
   firewall_port p do
      action :open
   end
end
```

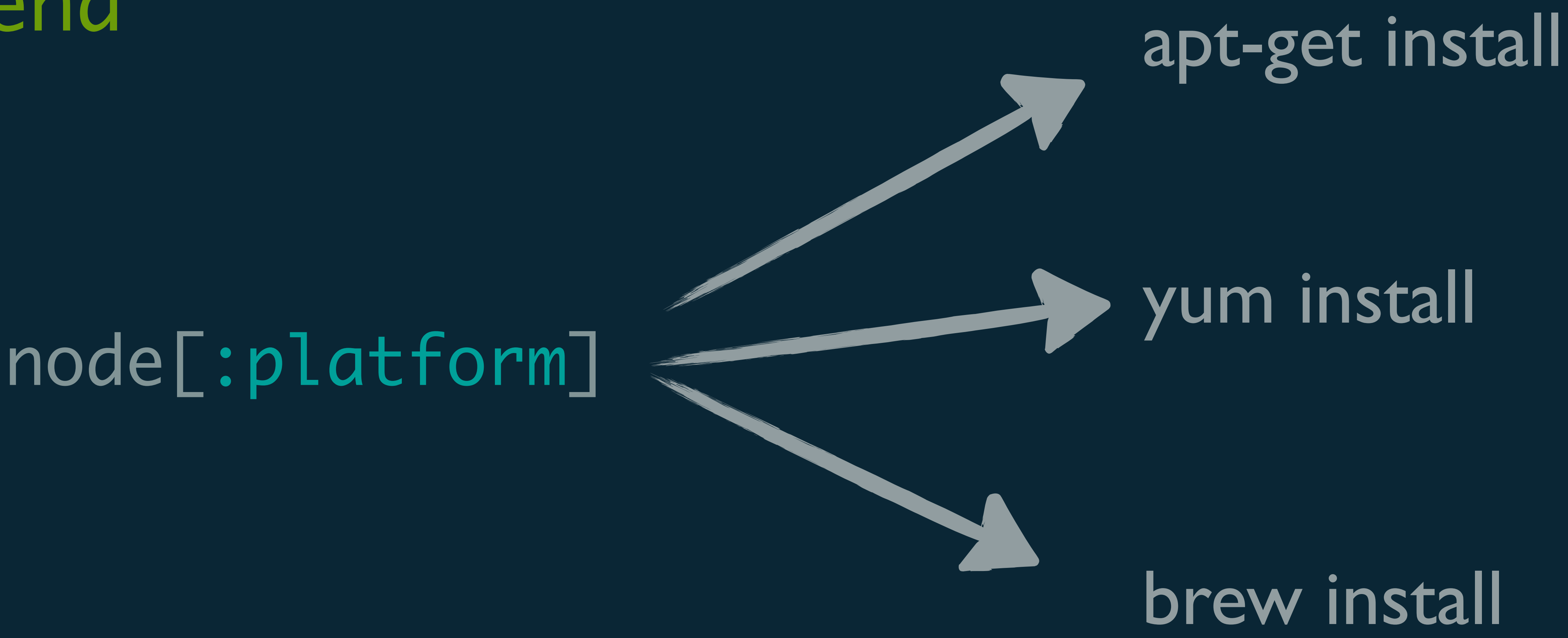
```
apache_ports do |p|
   firewall_port p do
      action :open
   end
end
```

```
define :firewall_port, :action => :open do
  if not params.has_key?(:port)
     params[:port] = params[:name]
  end

  if params[:action] == :open
    execute "open #{params[:port]}" do
      command "iptables -A INPUT --destination-port
                      #{params[:port]} -j ACCEPT"
    end
  else
    execute "close #{params[:port]}" do
      command "iptables -A INPUT --destination-port
                      #{params[:port]} -j DROP"
    end
  end
end
```

Providers

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



node[:platform]

apt-get install

yum install

brew install

```
"#{File.expand_path(
     Chef::Config[:file_cache_path]
         )}/handlers/yola.rb"
```

```
"#{File.expand_path(
      Chef::Config[:file_cache_path]
          )}/handlers/yola.rb"
```

```
class YolaHandler < Chef::Handler
   def report()
      run_status.updated_resources.each do |r|
         campfire_speak(r.to_s)
      end
   end
end
```

```
"#{File.expand_path(
     Chef::Config[:file_cache_path]
         )}/handlers/yola.rb"
```

```
class YolaHandler < Chef::Handler
   def report()
      run_status.updated_resources.each do |r|
         campfire_speak(r.to_s)
      end
   end
end
```

```
Chef/nodes/web1.json

Chef/roles/web.rb

Chef/cookbooks/apache/
           recipes/default.rb

           attributes/default.rb

           templates/default/apache2.conf.erb

           libraries/default.rb

Chef/environments/production.rb
```

```
Chef/nodes/web1.json

Chef/roles/web.rb

Chef/cookbooks/apache/
           recipes/default.rb

           attributes/default.rb

           templates/default/apache2.conf.erb

           libraries/default.rb

Chef/environments/production.rb

Chef/environments/testing.rb
```

```
default_attributes({
  "aws_region" => "usa-east"
})
```

```
default_attributes({
  "aws_region" => "usa-east",
  "logging" => {
    "debug" => true
  }
})
```

```
Chef/nodes/web1.json
Chef/roles/web.rb
Chef/cookbooks/apache/
           recipes/default.rb
           attributes/default.rb
           templates/default/apache2.conf.erb
           libraries/default.rb
Chef/environments/production.rb
Chef/environments/testing.rb
Chef/data_bags/employees/jonathan.json
```

```
{
  "name": "Jonathan Hitchcock",
  "id": "jonathan",
  "groups": ["developers", "operations"],
  "ssh_keys": [
    { "key":
"AAAB3NzC1yc2EAAAABIwAAAQEA3advEyDE0aEo=="
    }
  ]
}
```

Provisioning

Provisioning

```
$ knife ec2 server create -r 'role[webserver]'
                    -I ami-7000f019 -f m1.small
```

Provisioning

```
$ knife ec2 server create -r 'role[webserver]'
                    -I ami-7000f019 -f m1.small
```

Amazon EC2

Openstack

Rackspace Cloud

libvirt

Eucalyptus

KVM

Kickstart

XEN

Vagrant

Integration

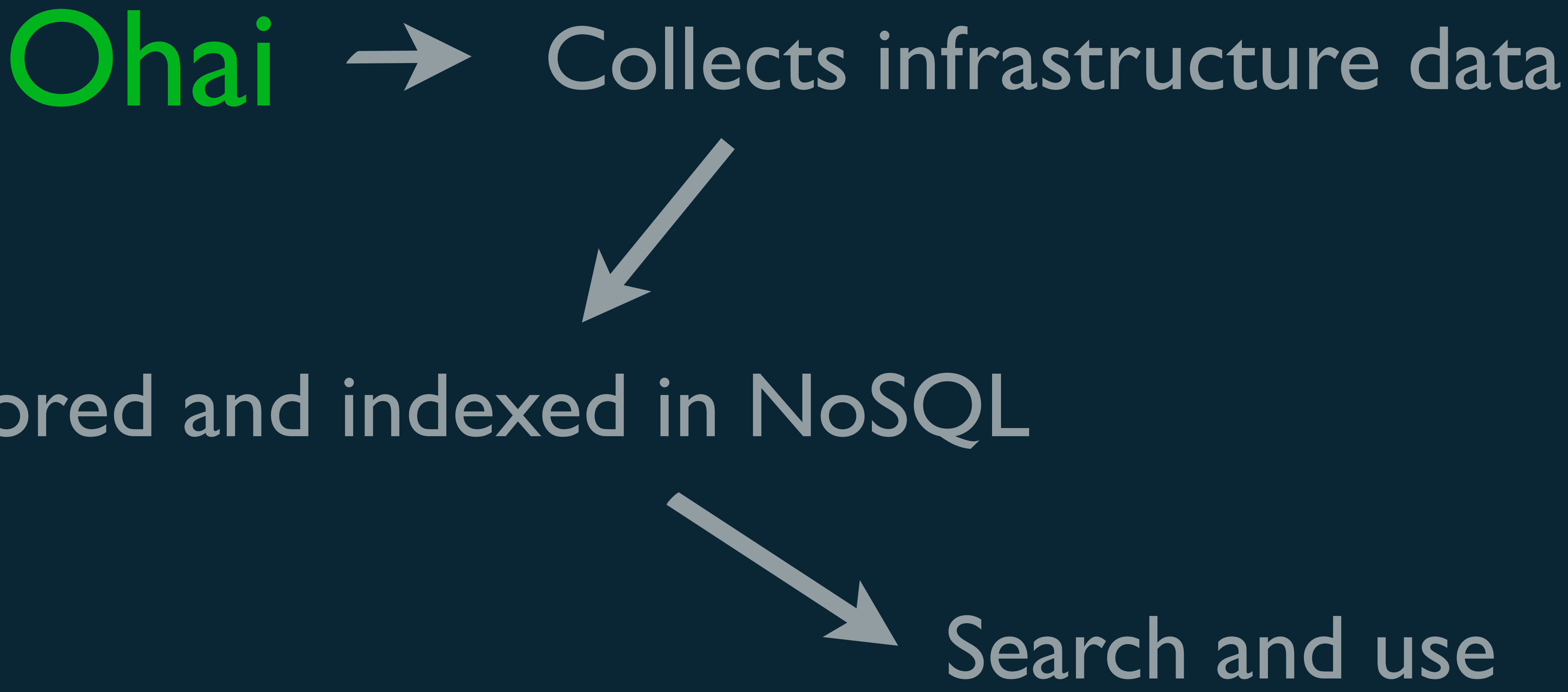
configuration data vs configuration logic

Integration

configuration data vs configuration logic

Integration

configuration data vs configuration logic



Ohai → Collects infrastructure data

Stored and indexed in NoSQL

Search and use

(A side note on) Attributes

default

normal

override

automatic

(A side note on) Attributes

default	cookbooks, data bags
normal	node definitions
override	for special cases
automatic	ohai

(Back to) Integration

```
search(:node, 'role:webbackend') do |wb|
  haproxy_backend "#{wb[:apache][:port]}" do
    action :create
  end
end
```

Systems auto-discover each other

Configuration is responsive to changes

Integration

```
search(:node, 'role:webbackend') do |wb|
  haproxy_backend "#{wb[:apache][:port]}" do
    action :create
  end
end
```

Systems auto-discover each other

Configuration is responsive to changes

So you don’t have to be!

Jonathan Hitchcock

@vhata

github.com/vhata