DevOps with Liberty and Chef

Jeremy Hughes

WebSphere DevOps Architect @ IBM

with thanks to Chef

Please Note

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

- What IS Chef?

- What IS Chef?
- What IS IBM WebSphere ® Liberty Profile?

- What IS Chef?
- What IS IBM WebSphere ® Liberty Profile?
- Why they're good together?

Overview of Chef

v2.0.0

Complexity

http://www.flickr.com/photos/michaelheiss/3090102907/

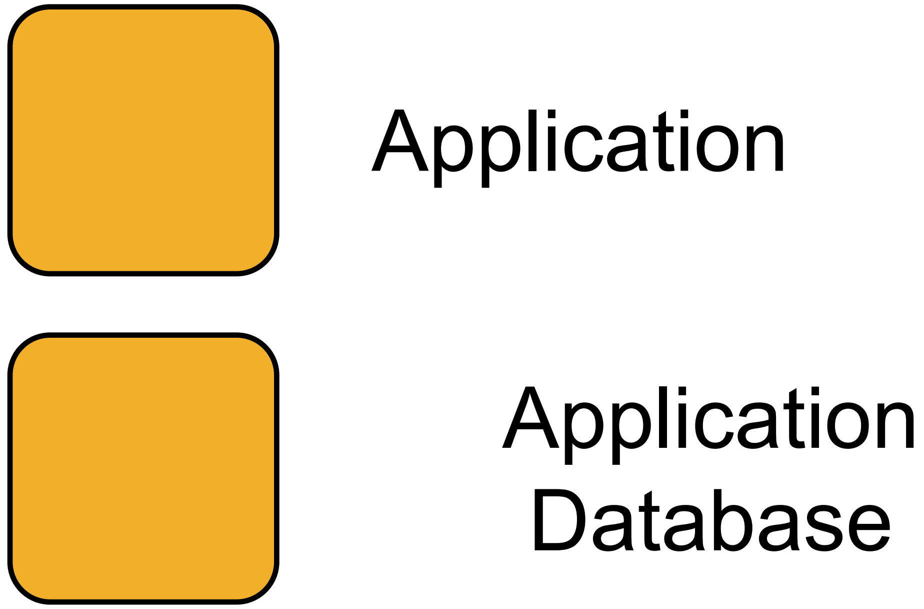
Items of Manipulation (Resources)

- Networking
- Files
- Directories
- Symlinks
- Mounts
- Registry Keys
- Powershell Scripts
- Users
- Groups
- Packages
- Services
- Filesystems

A tale of growth...

Application

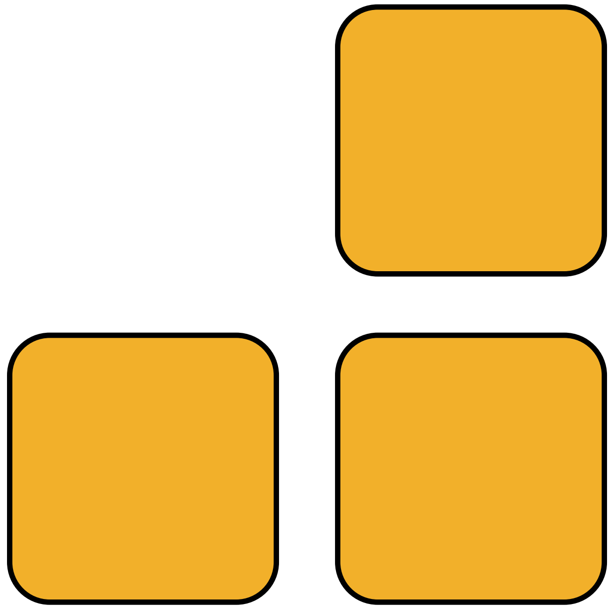
Add a database



Application

Application Database

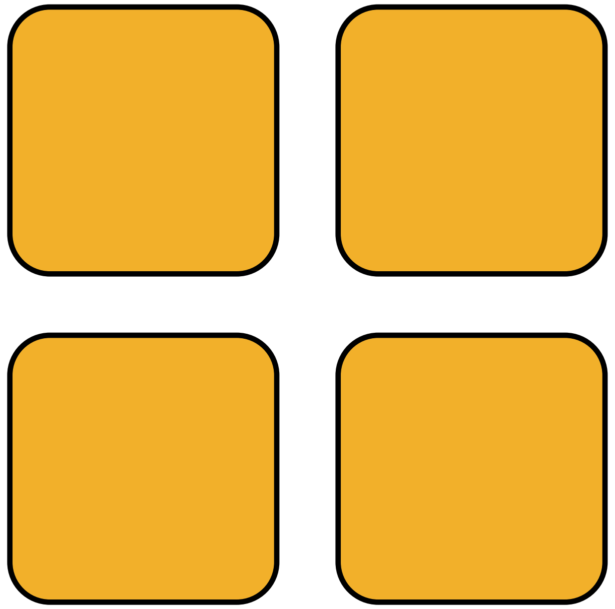
Make database redundant



Application

App Databases

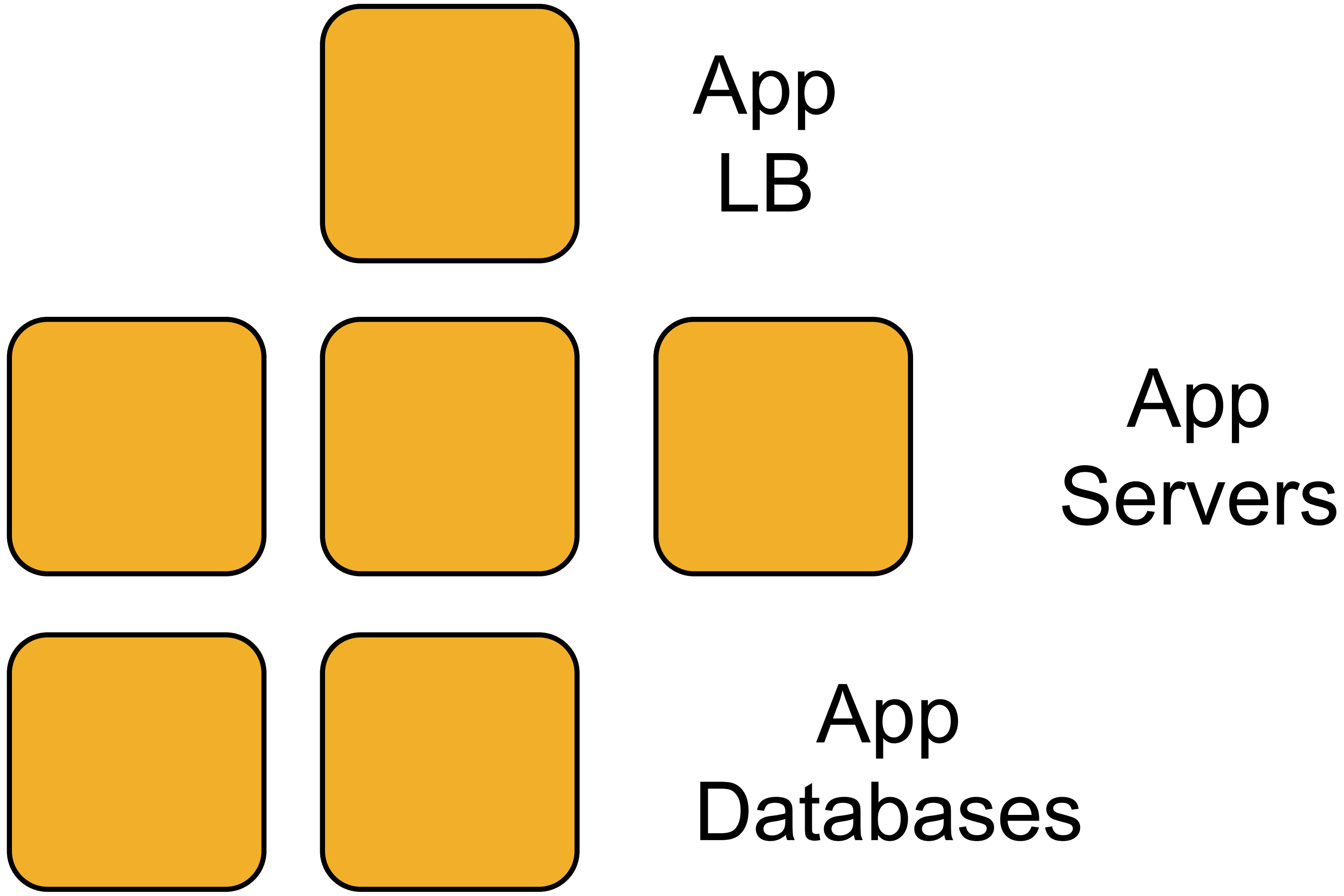
Application server redundancy



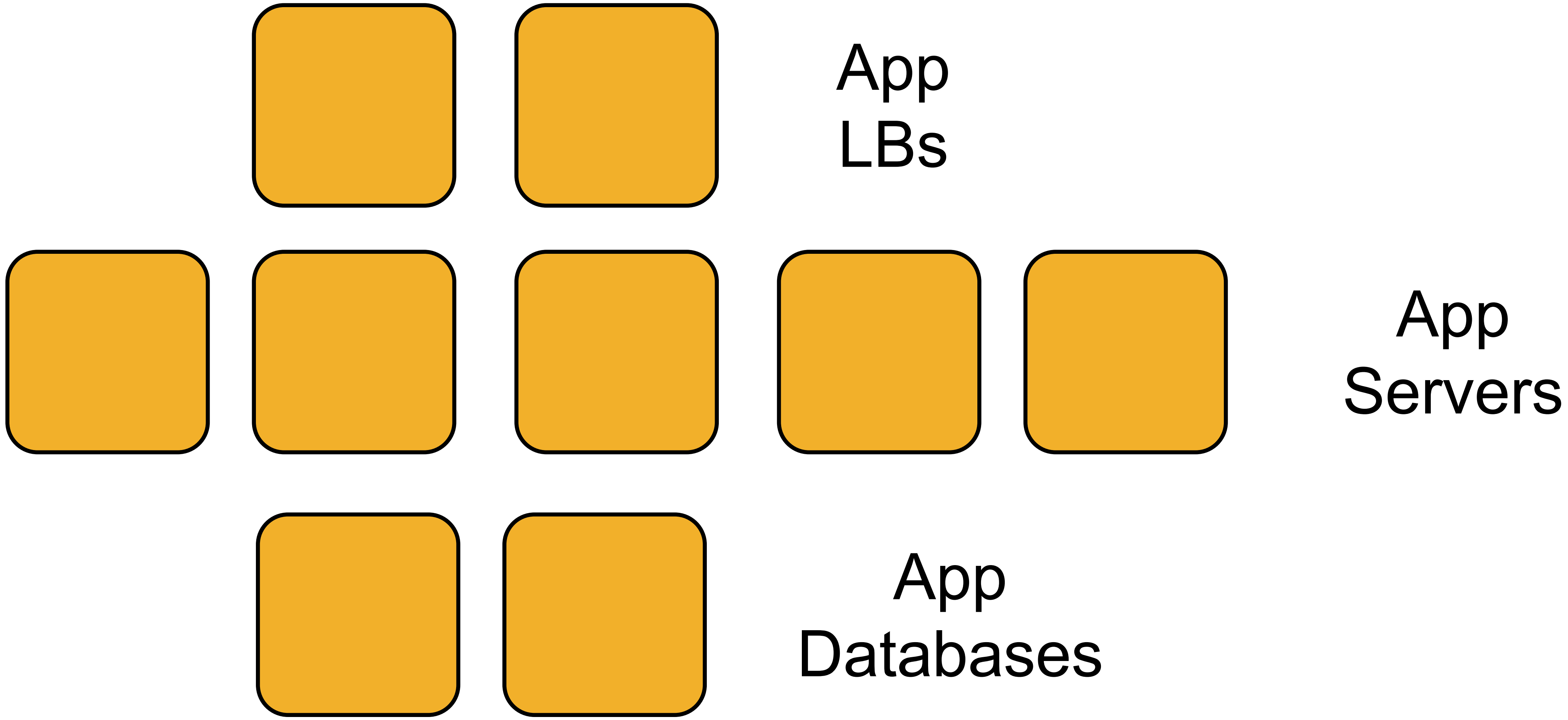
App Servers

App Databases

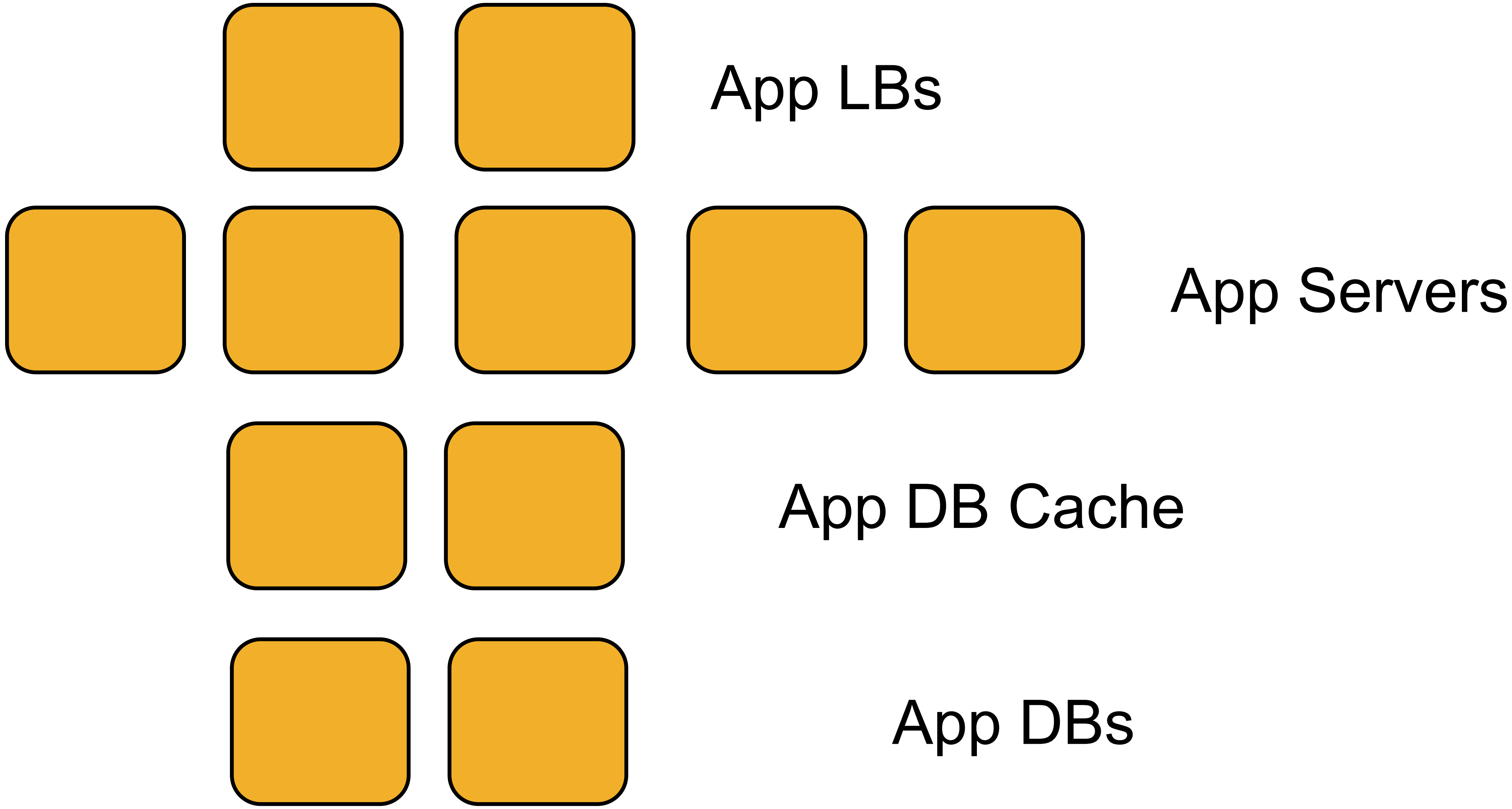
Add a load balancer



Webscale!



Now we need a caching layer



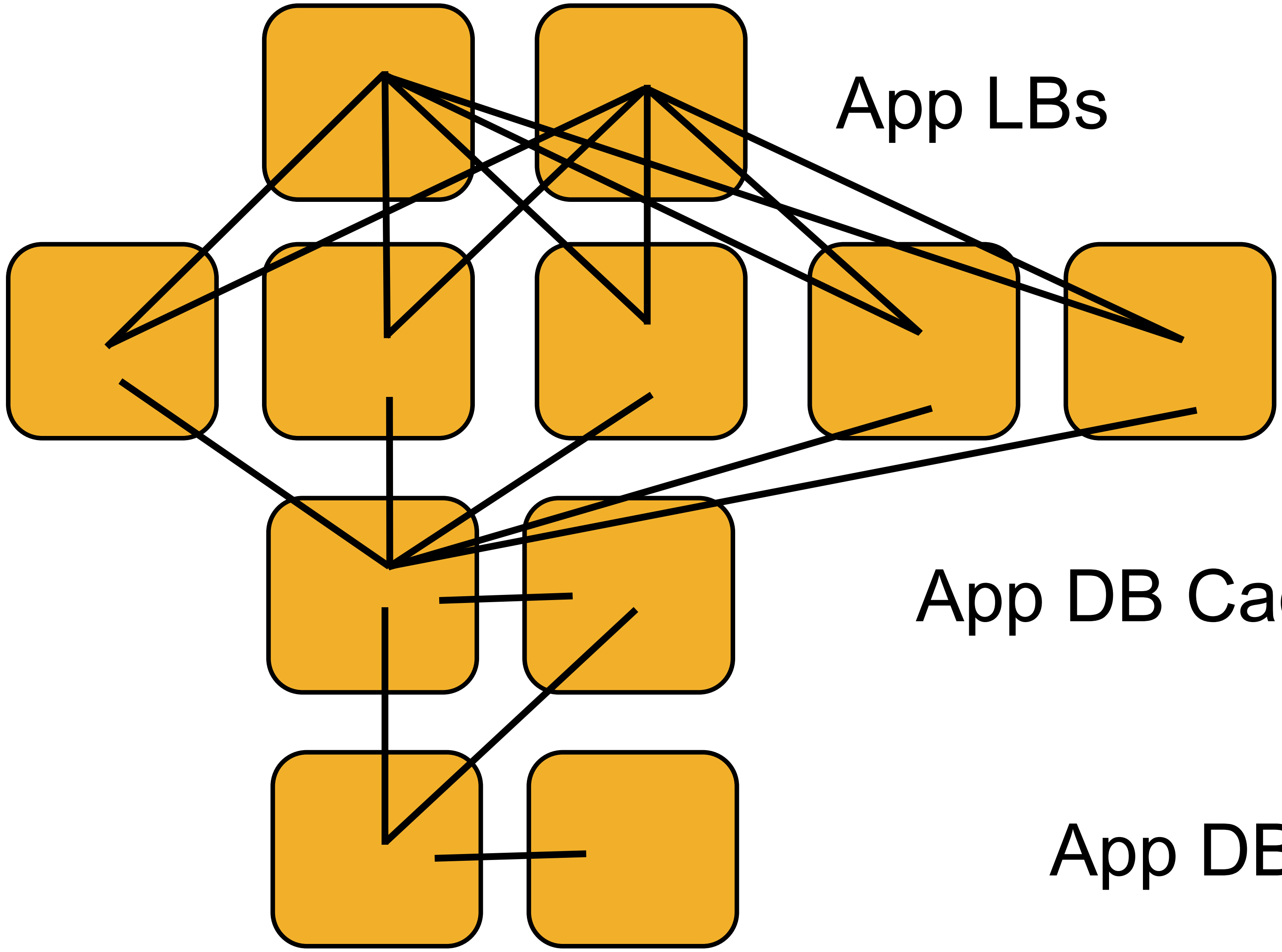
App LBs

App Servers

App DB Cache

App DBs

Infrastructure has a Topology



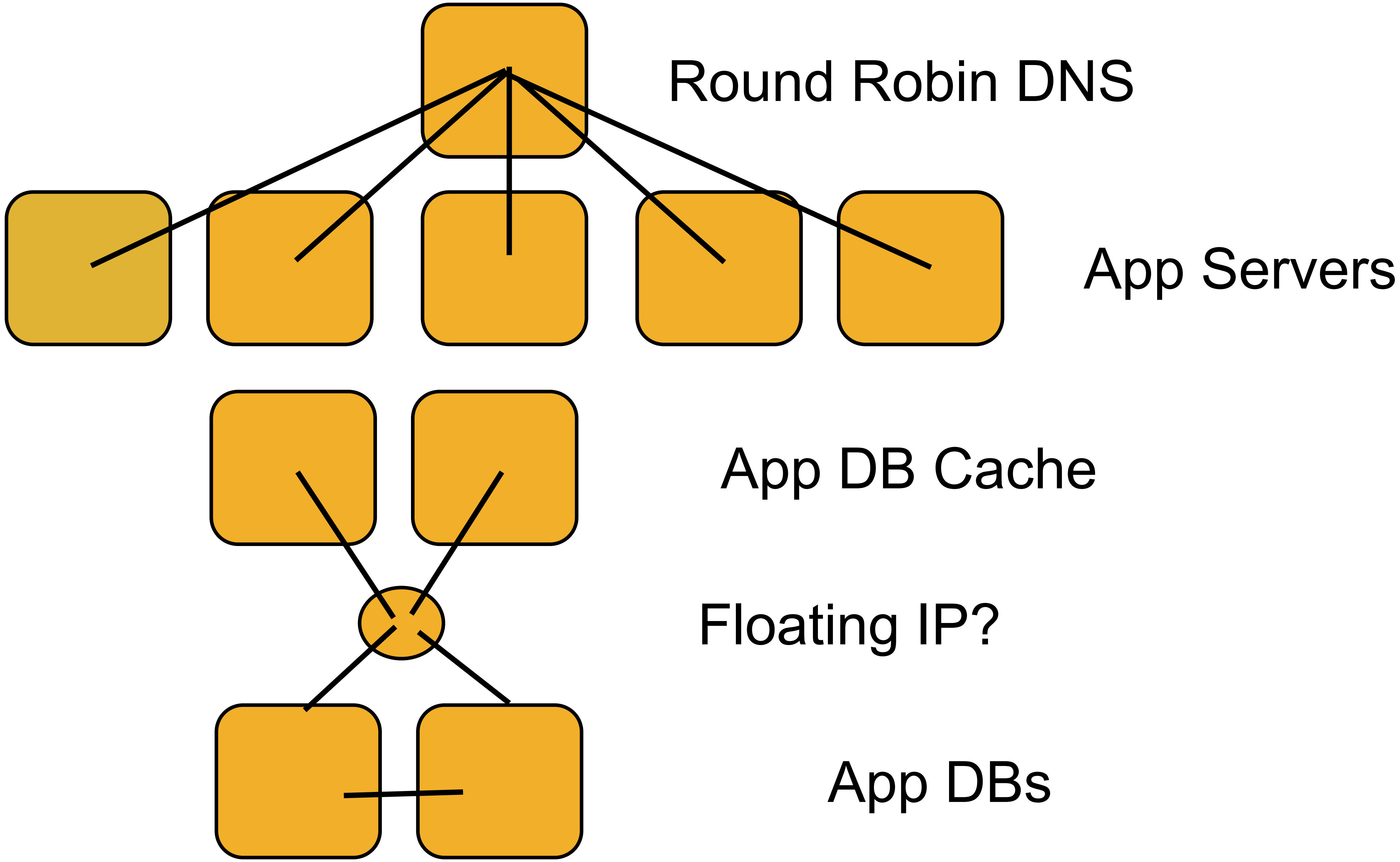
App LBs

App Servers

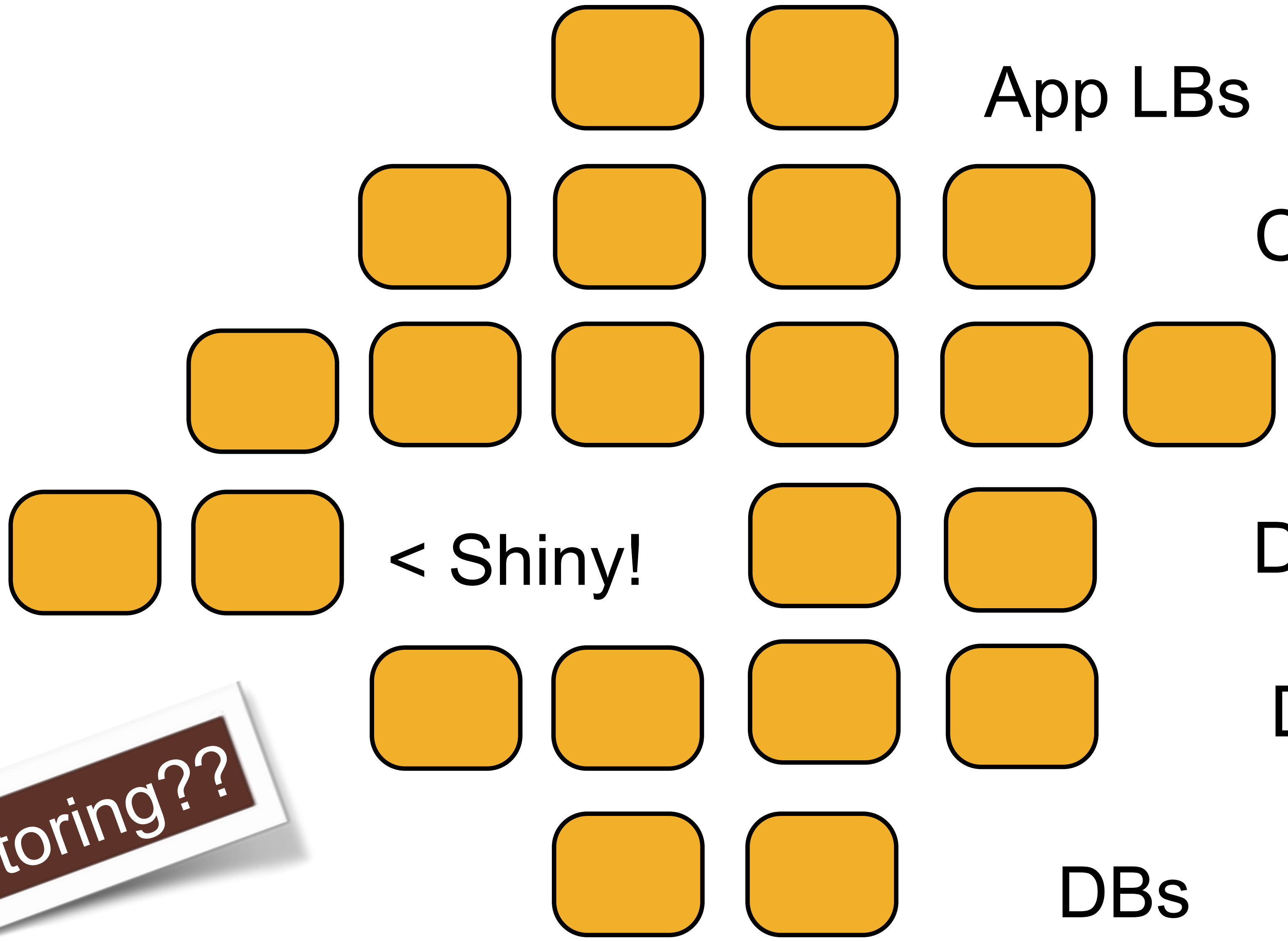
App DB Cache

App DBs

Your Infrastructure is a Snowflake



Complexity Increases Quickly



App LBs

Cache

App Servers

< Shiny!

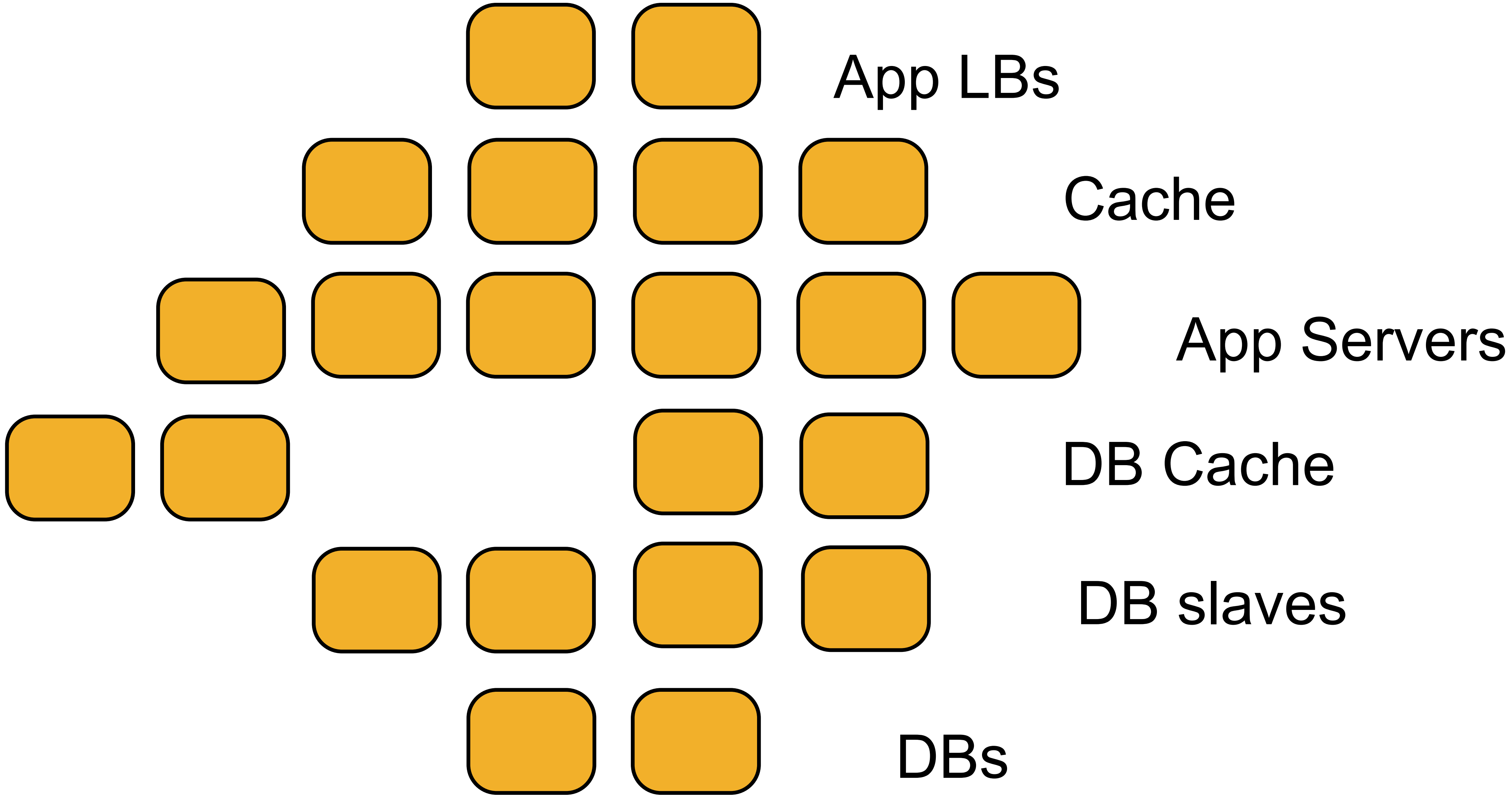
DB Cache

DB slaves

DBs

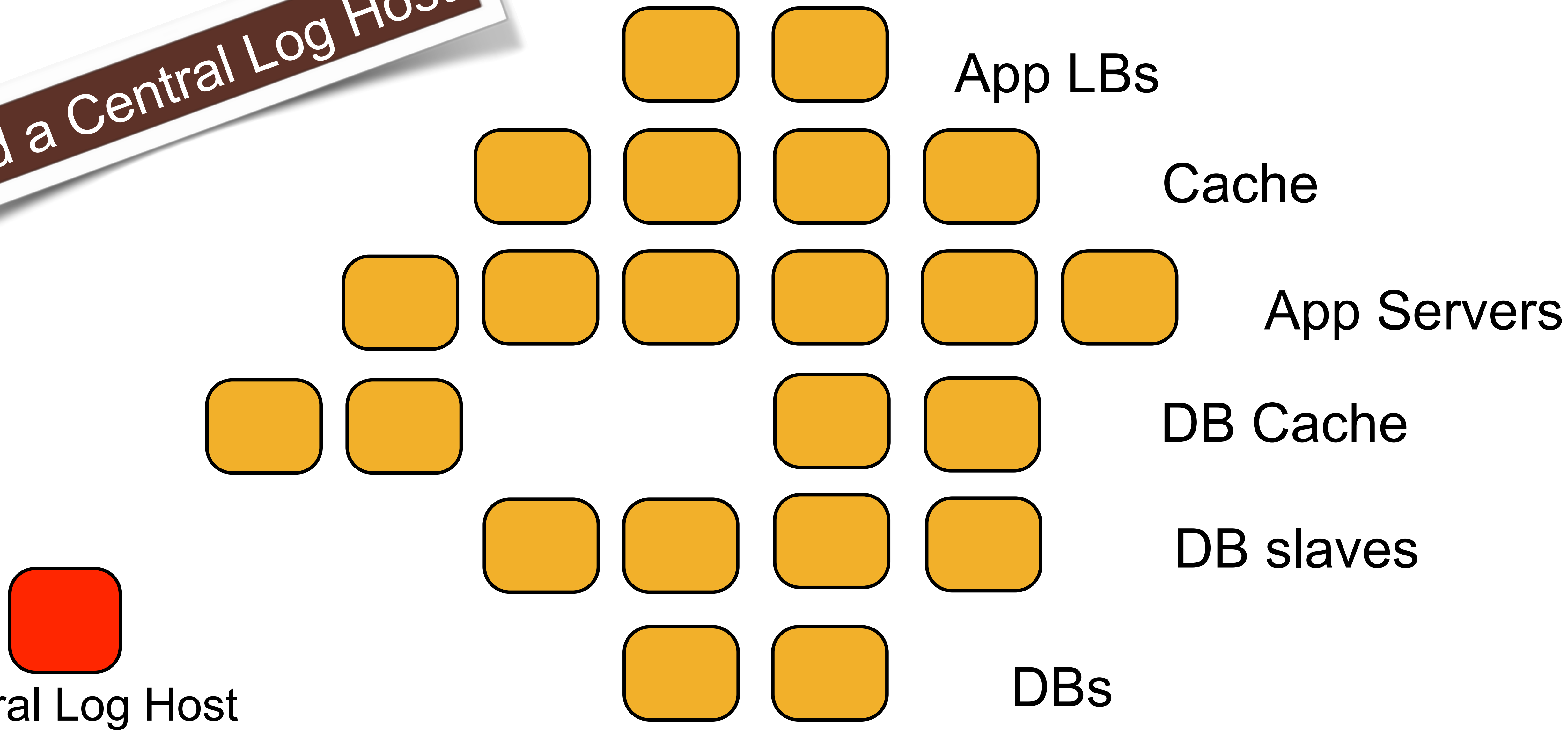
Are we monitoring??

...and change happens!



...and change happens!

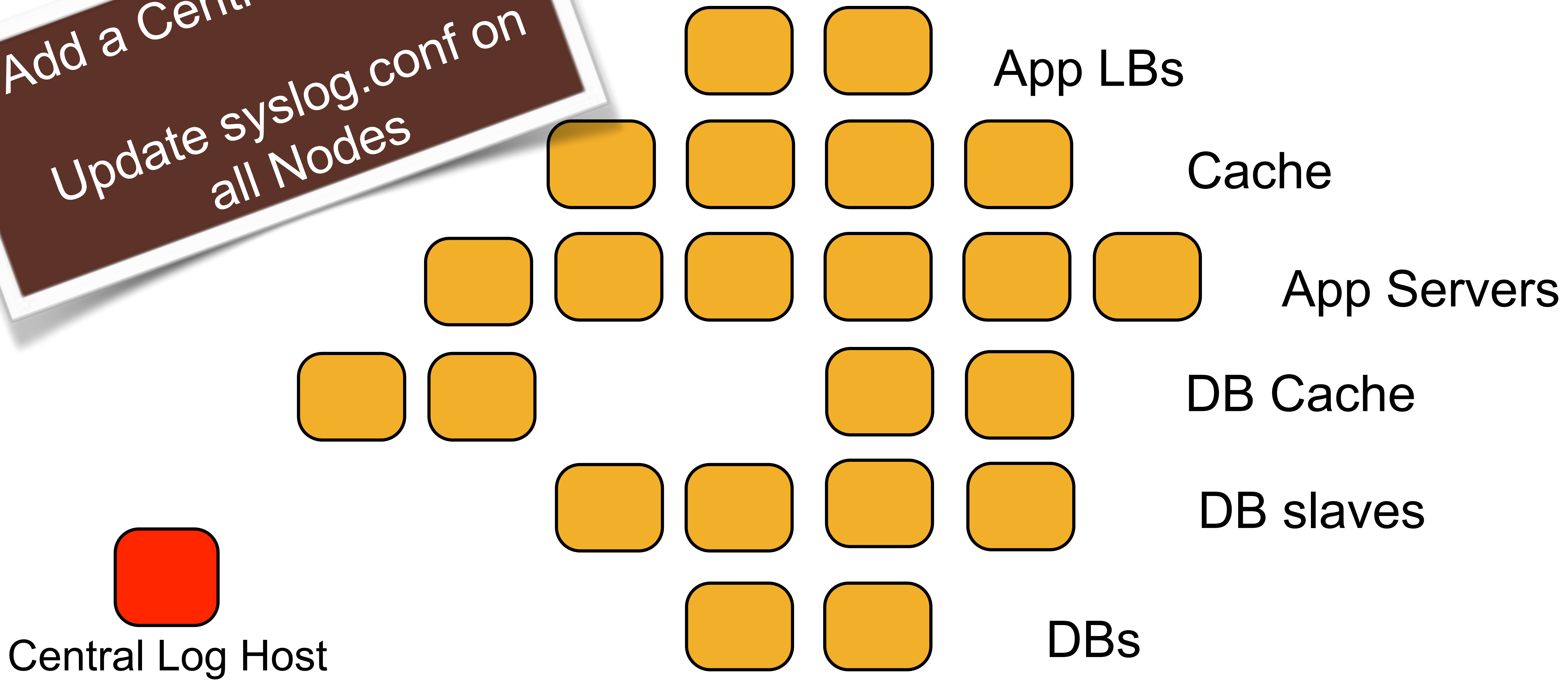
Add a Central Log Host



...and change happens!

Add a Central Log Host

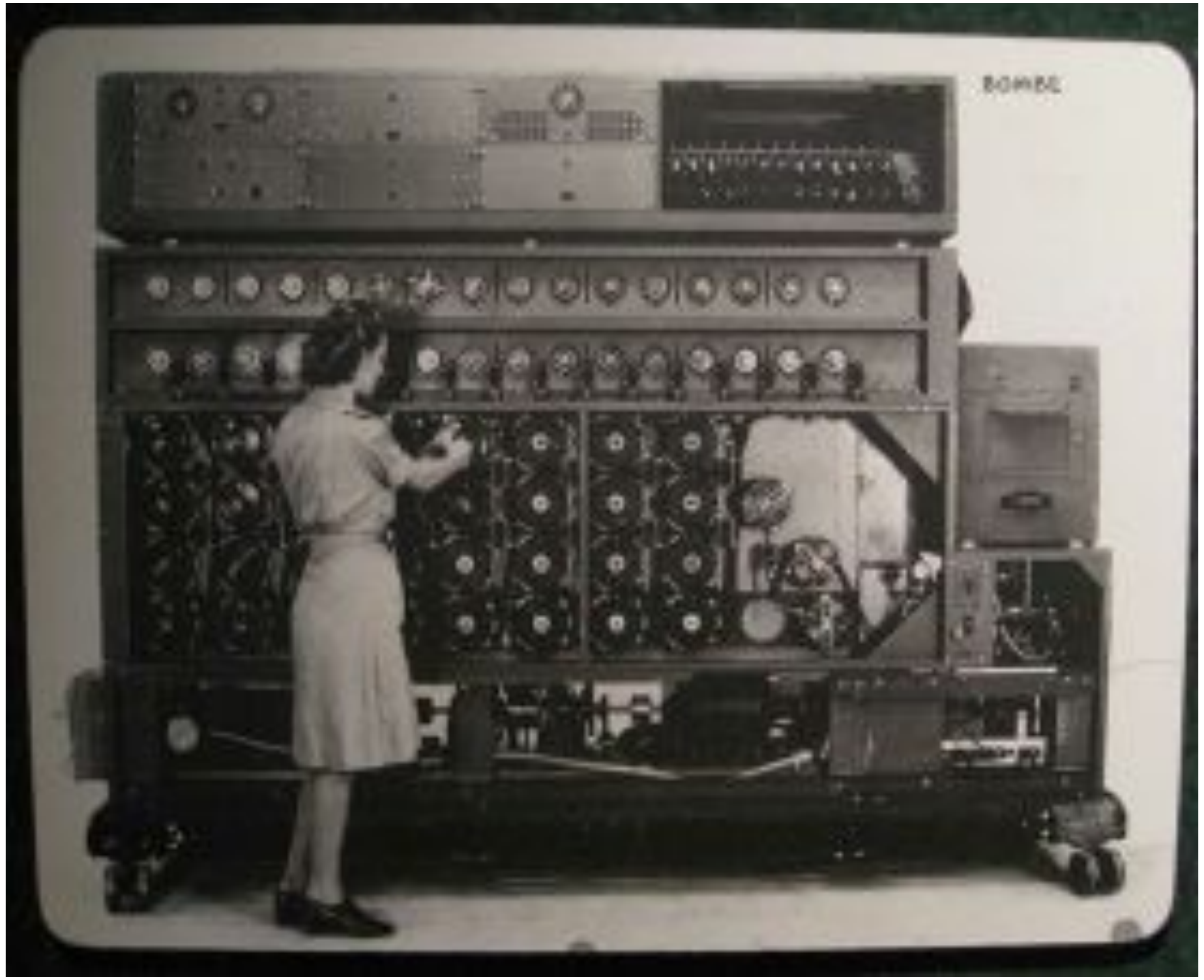
Update syslog.conf on all Nodes



Chef Solves This Problem

- But you already guessed that, didn’t you?

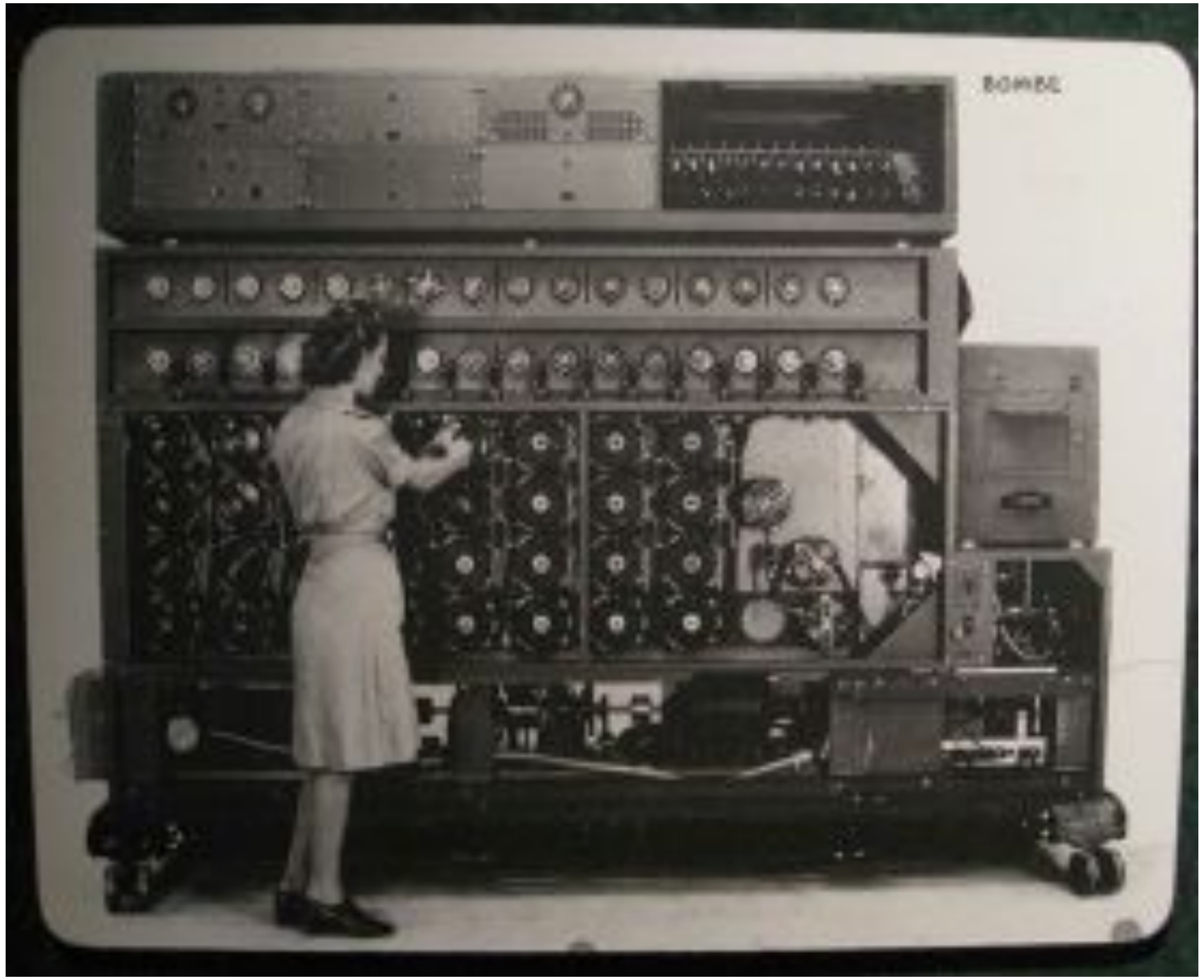
Chef is Infrastructure as Code



http://www.flickr.com/photos/louisb/4555295187/

- Programmatically provision and configure components

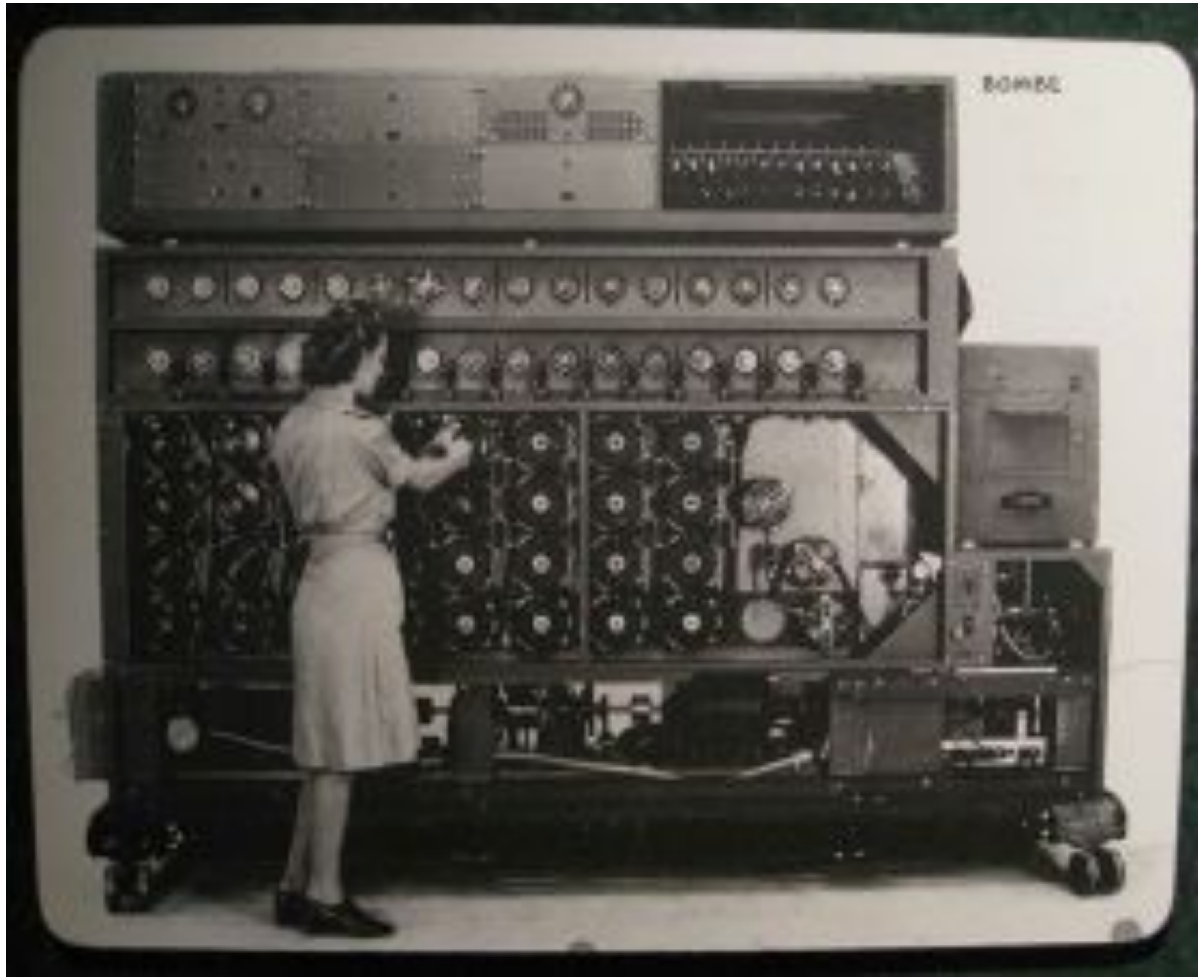
Chef is Infrastructure as Code



http://www.flickr.com/photos/louisb/4555295187/

- Treat like any other code base

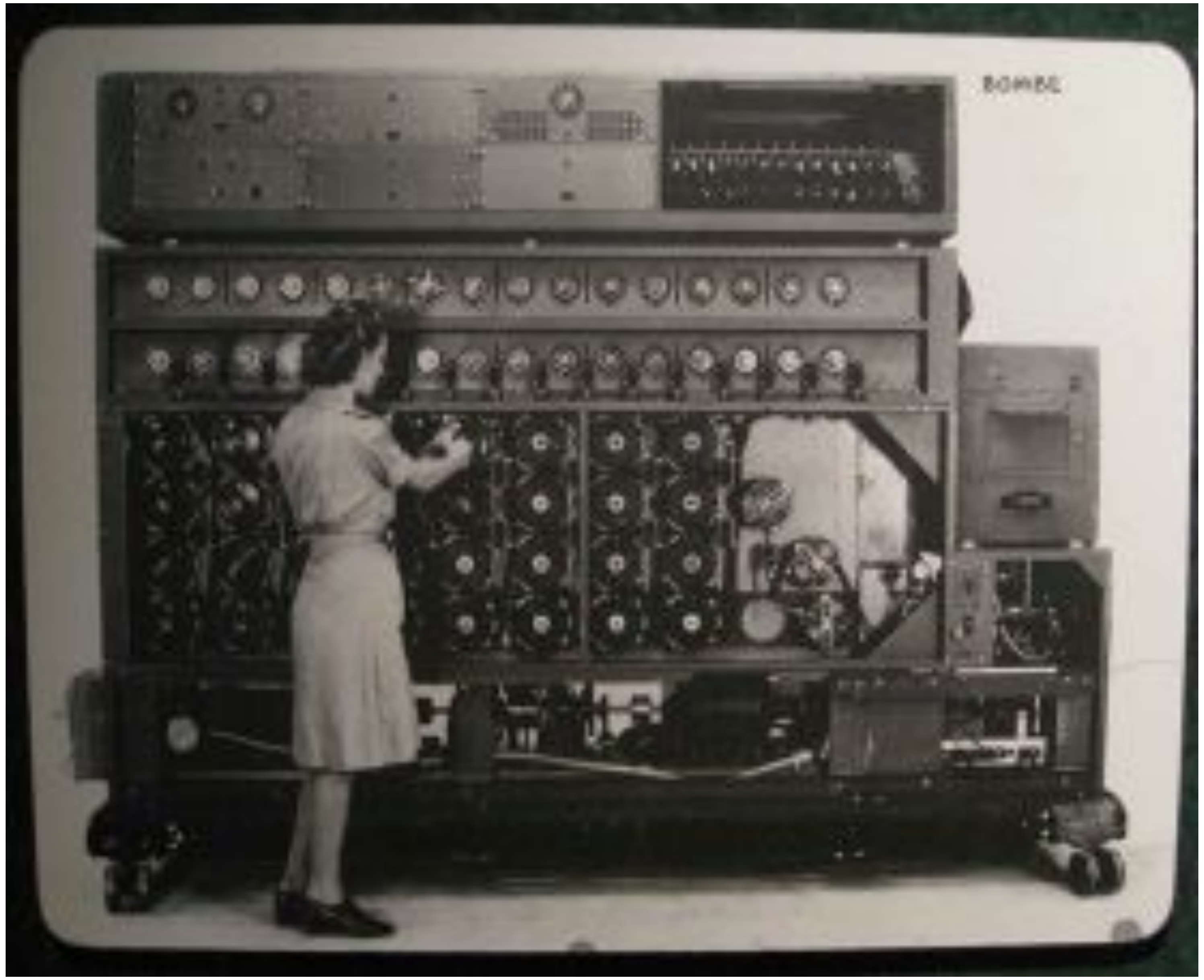
Chef is Infrastructure as Code



http://www.flickr.com/photos/louisb/4555295187/

- Reconstruct business from **code repository**, **data backup**, and **compute resources**

Chef is Infrastructure as Code



http://www.flickr.com/photos/louisb/4555295187/

- Programmatically provision and configure components
- Treat like any other code base
- Reconstruct business from **code repository**, **data backup**, and **compute resources**

Configuration Code

- Chef ensures each Node complies with the policy
- Policy is determined by the configurations in each Node’s run list
- Reduce management complexity through abstraction
- Store the configuration of your infrastructure in version control

Declarative Interface to Resources

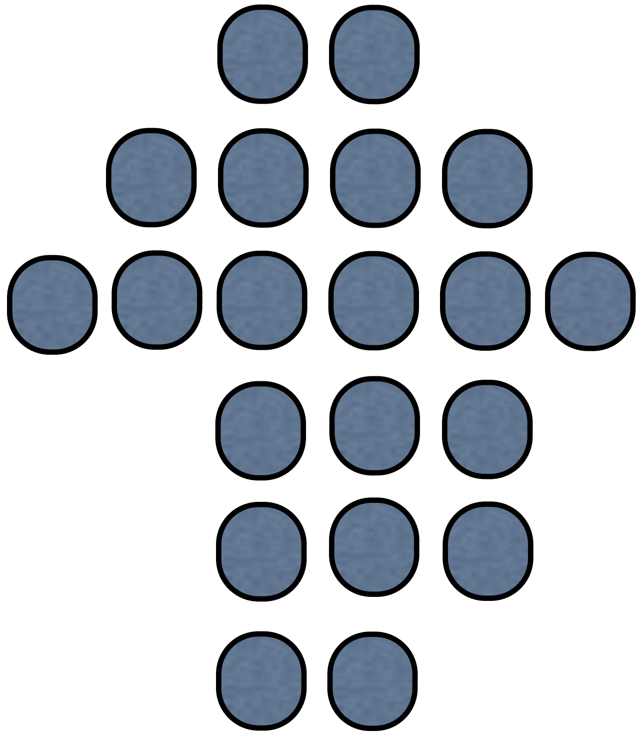
- You define the policy in your Chef configuration
- Your policy states what state each resource should be in, but not how to get there
- Chef-client will pull the policy from the Chef Server and enforce the policy on the Node

Managing Complexity

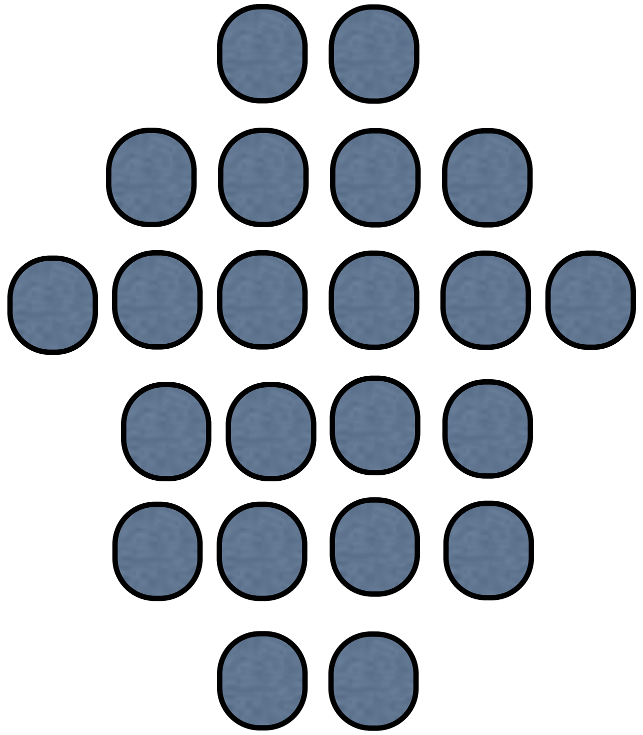
- Organizations
- Environments
- Roles
- Nodes
- Recipes
- Cookbooks
- Search

Organizations

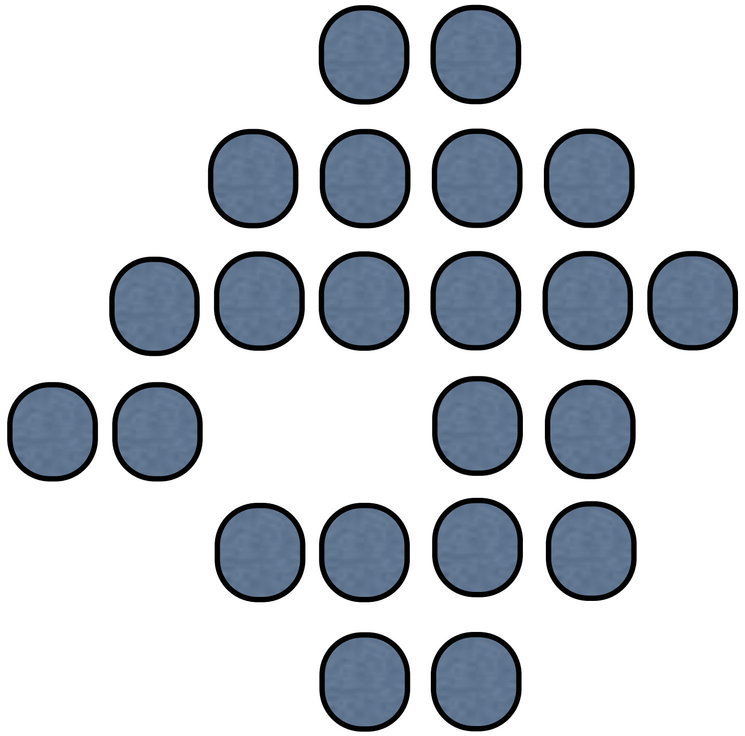
My Infrastructure



Your Infrastructure



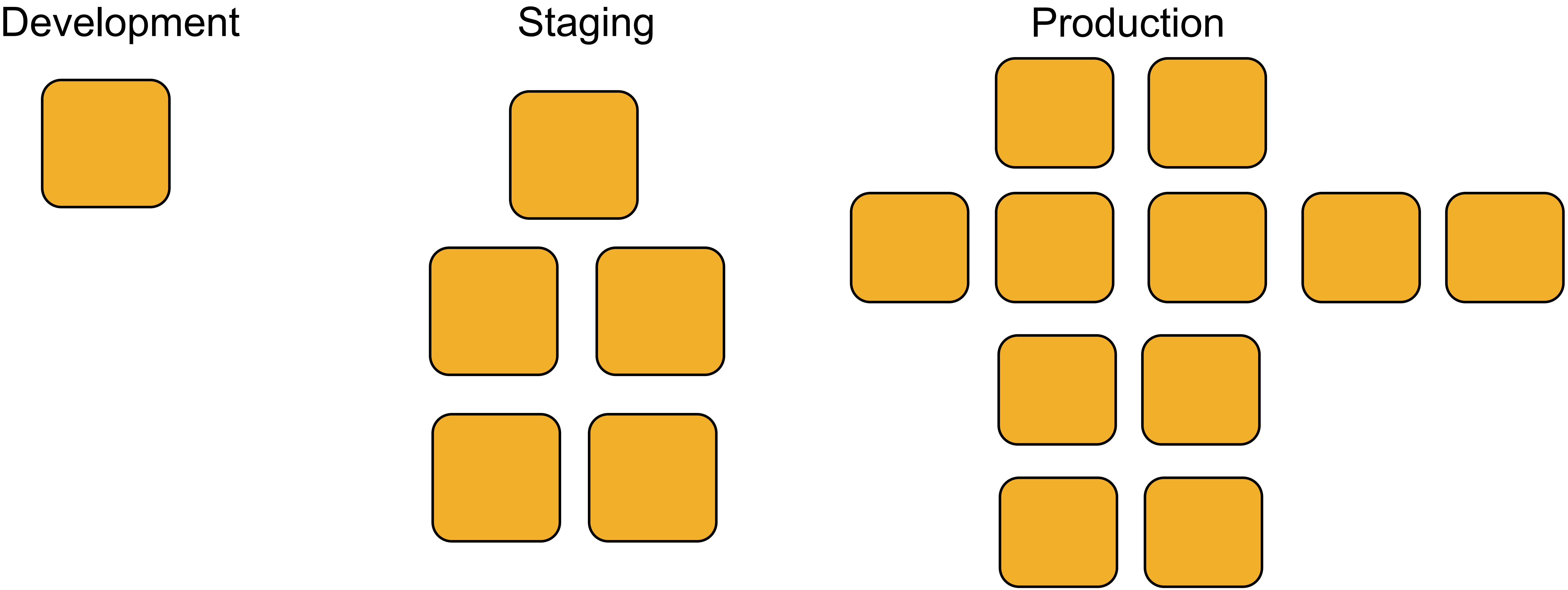
Their Infrastructure



Organizations

- Completely independent tenants of Enterprise Chef
- Share nothing with other organizations
- May represent different
  - Companies
  - Business Units
  - Departments

Environments



Development

Staging

Production

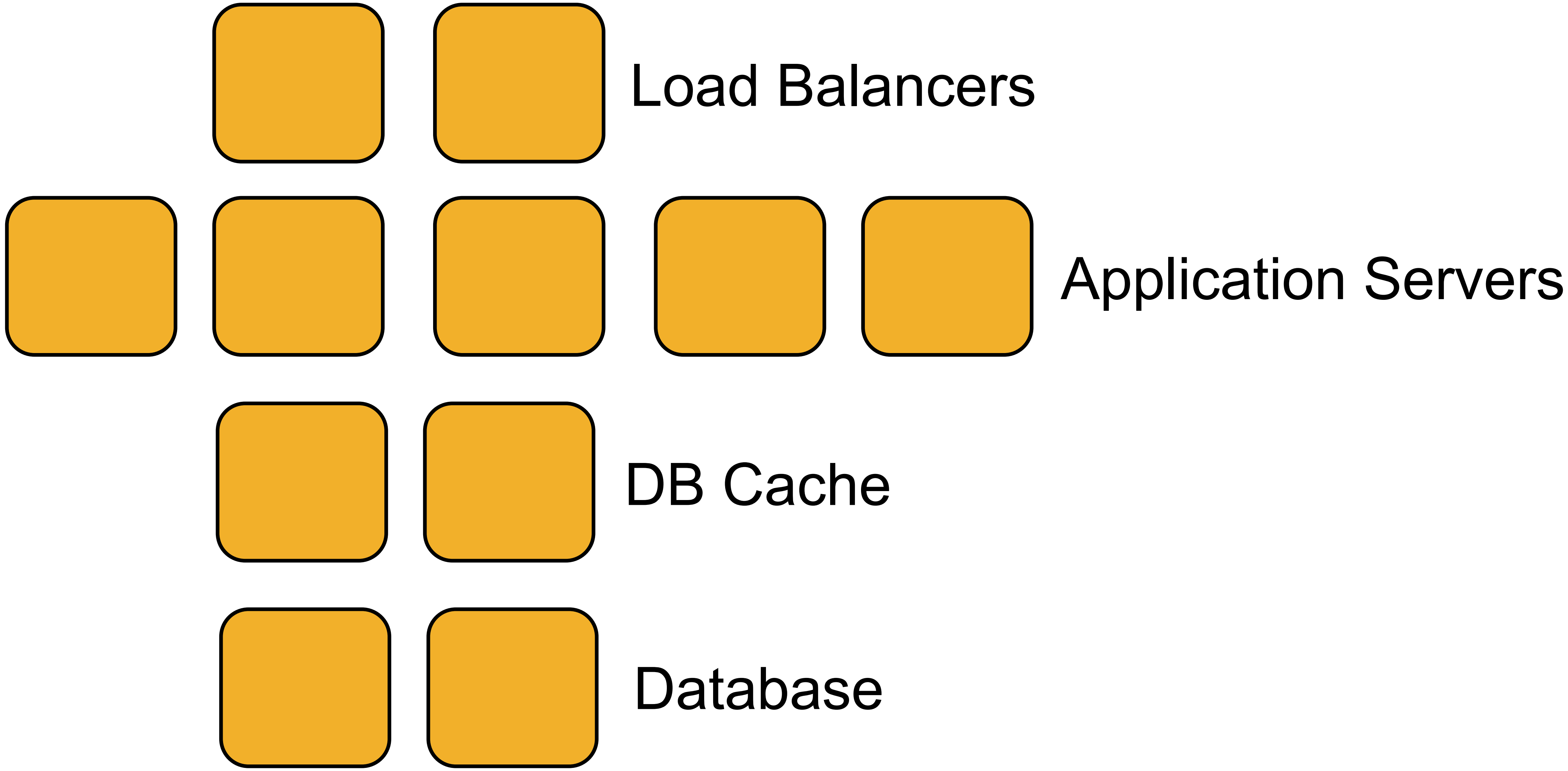
Environments

- Environments reflect your patterns and workflow, and can be used to model the life-stages of your applications
  - Development
  - Test
  - Staging
  - Production
  - etc.
- Every Organization starts with a single environment!

Environments Define Policy

- Environments may include data attributes necessary for configuring your infrastructure, e.g.
  - The URL of your payment service’s API
  - The location of your package repository
  - The version of the Chef configuration files that should be used

Roles



Load Balancers

Application Servers

DB Cache

Database

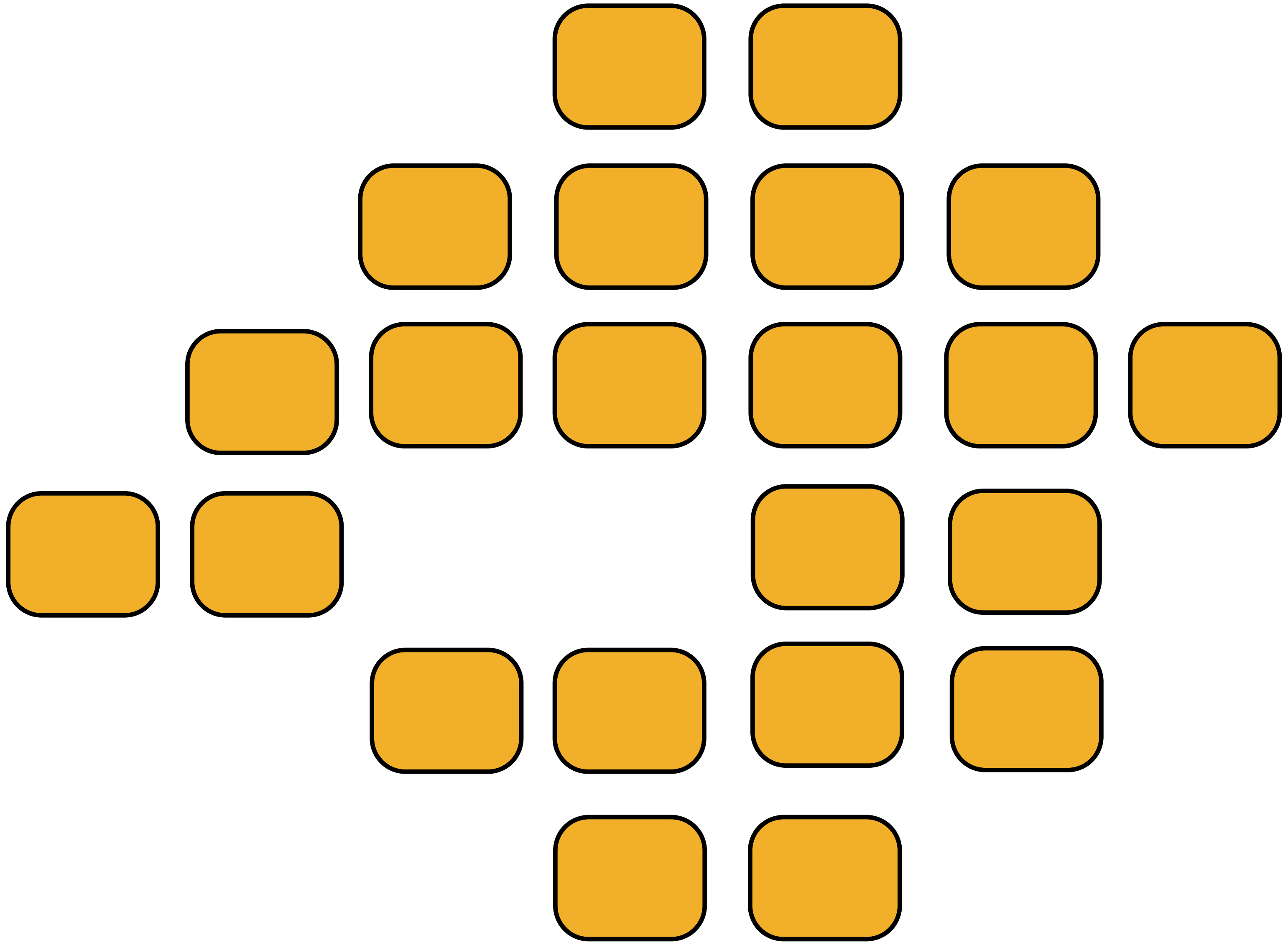
Roles

- Roles represent the types of servers in your infrastructure
  - Load Balancer
  - Application Server
  - Database Cache
  - Database
  - Monitoring

Roles Define Policy

- Roles may include an ordered list of Chef configuration files that should be applied
  - This list is called a Run List
  - Order is always important in the Run List
- Roles may include data attributes necessary for configuring your infrastructure, for example:
  - The port that the application server listens on
  - A list of applications that should be deployed

Nodes



Nodes

- Nodes represent the servers in your infrastructure
  - Could be physical servers or virtual servers
  - May represent hardware that you own or compute instances in a public or private cloud
- Could also be network hardware - switches, routers, etc

Node

- Each Node will
  - Belong to one Organization
  - Belong to one Environment
  - Have zero or more Roles

Nodes Adhere to Policy

- The chef-client application runs on each node, which
  - Gathers the current system configuration of the node
  - Downloads the desired system configuration policies from the Chef server for that node
  - Configures the node such that it adheres to those policies

Resources

- A Resource represents a piece of the system and its desired state
  - A package that should be installed
  - A service that should be running
  - A file that should be generated
  - A cron job that should be configured
  - A user that should be managed
  - and more

Resources in Recipes

- Resources are the fundamental building blocks of Chef configuration
- Resources are gathered into Recipes
- Recipes ensure the system is in the desired state

Recipes

- Configuration files that describe resources and their desired state
- Recipes can:
  - Install and configure software components
  - Manage files
  - Deploy applications
  - Execute other recipes
  - and more

Example Recipe

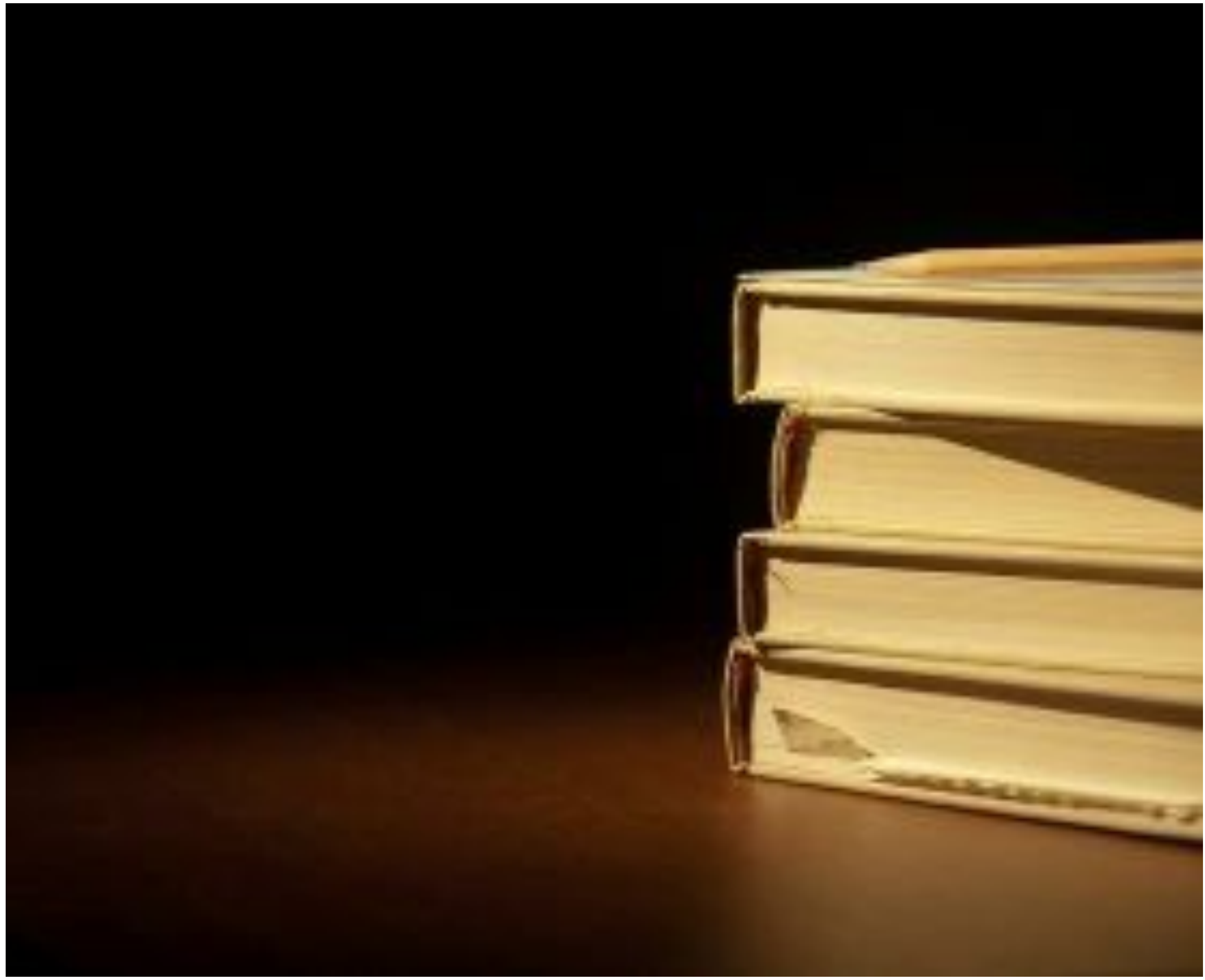
```
package "apache2"
```

```
template "/etc/apache2/apache2.conf" do
  source "apache2.conf.erb"
  owner "root"
  group "root"
  mode "0644"
  variables(:allow_override => "All")
  notifies :reload, "service[apache2]"
end
```

```
service "apache2" do
  action [:enable,:start]
  supports :reload => true
end
```

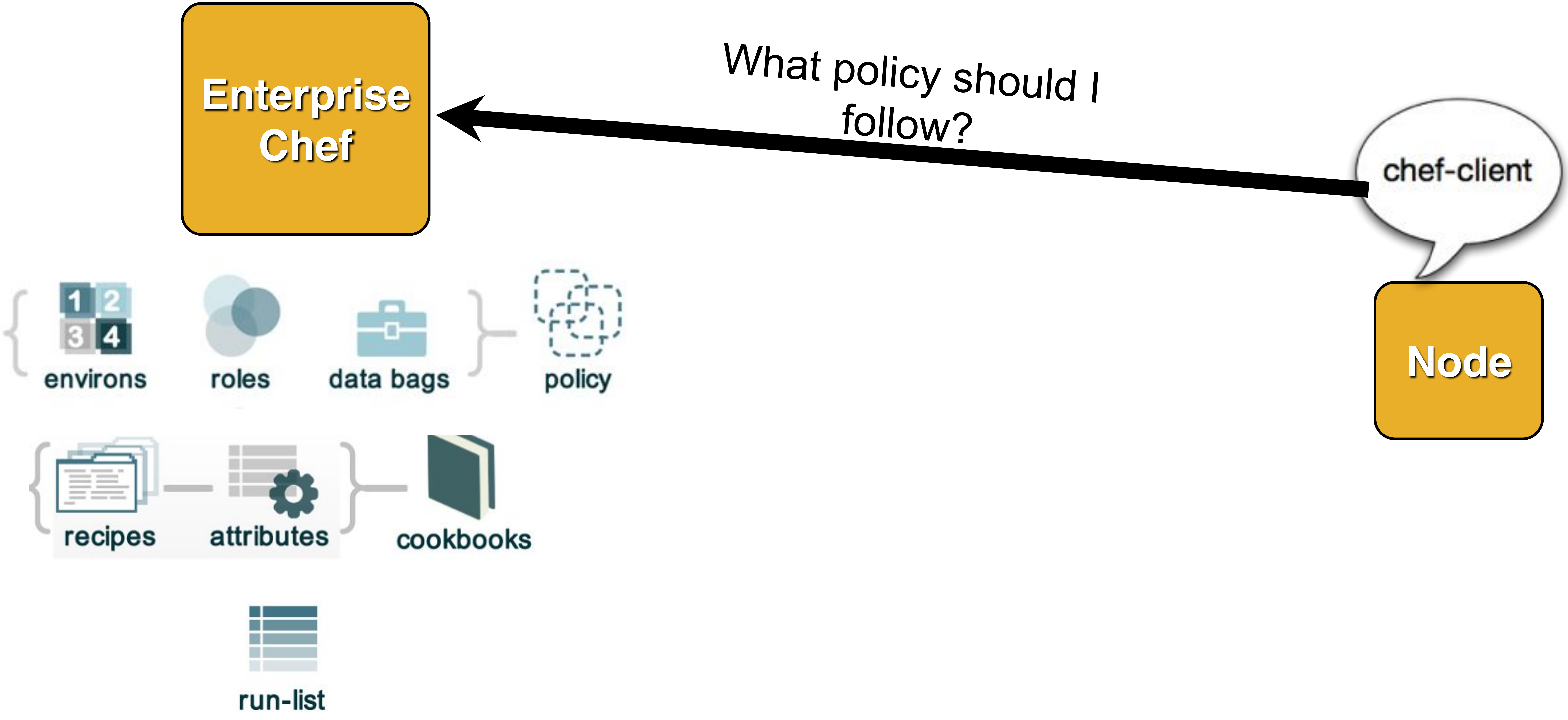
Cookbooks

- Recipes are stored in Cookbooks
- Cookbooks contain recipes, templates, files, custom resources, etc
- Code re-use and modularity



http://www.flickr.com/photos/shutterhacks/4474421855/

Run List



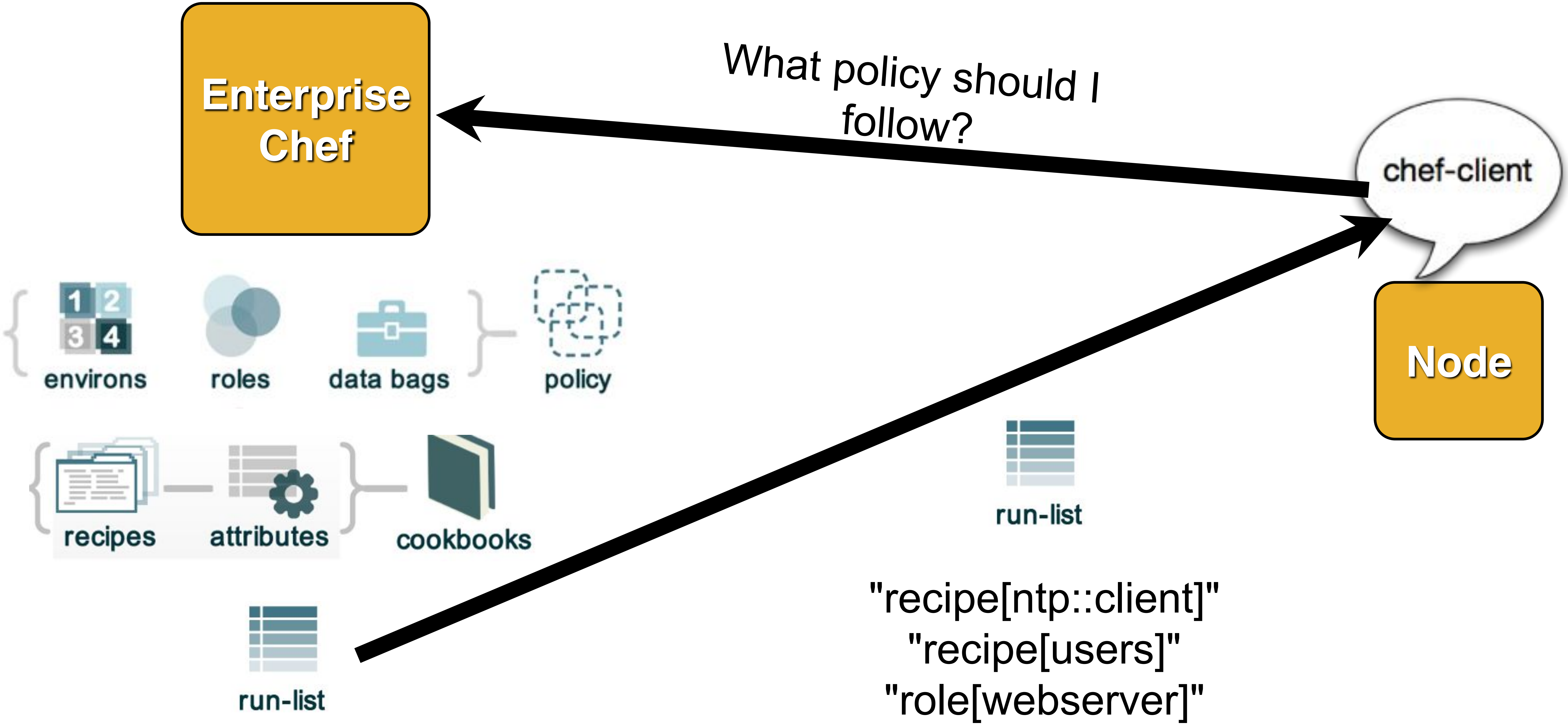
Enterprise Chef

What policy should I follow?

chef-client

Node

Run List



Enterprise Chef

What policy should I follow?

chef-client

Node

environs

roles

data bags

policy

recipes

attributes

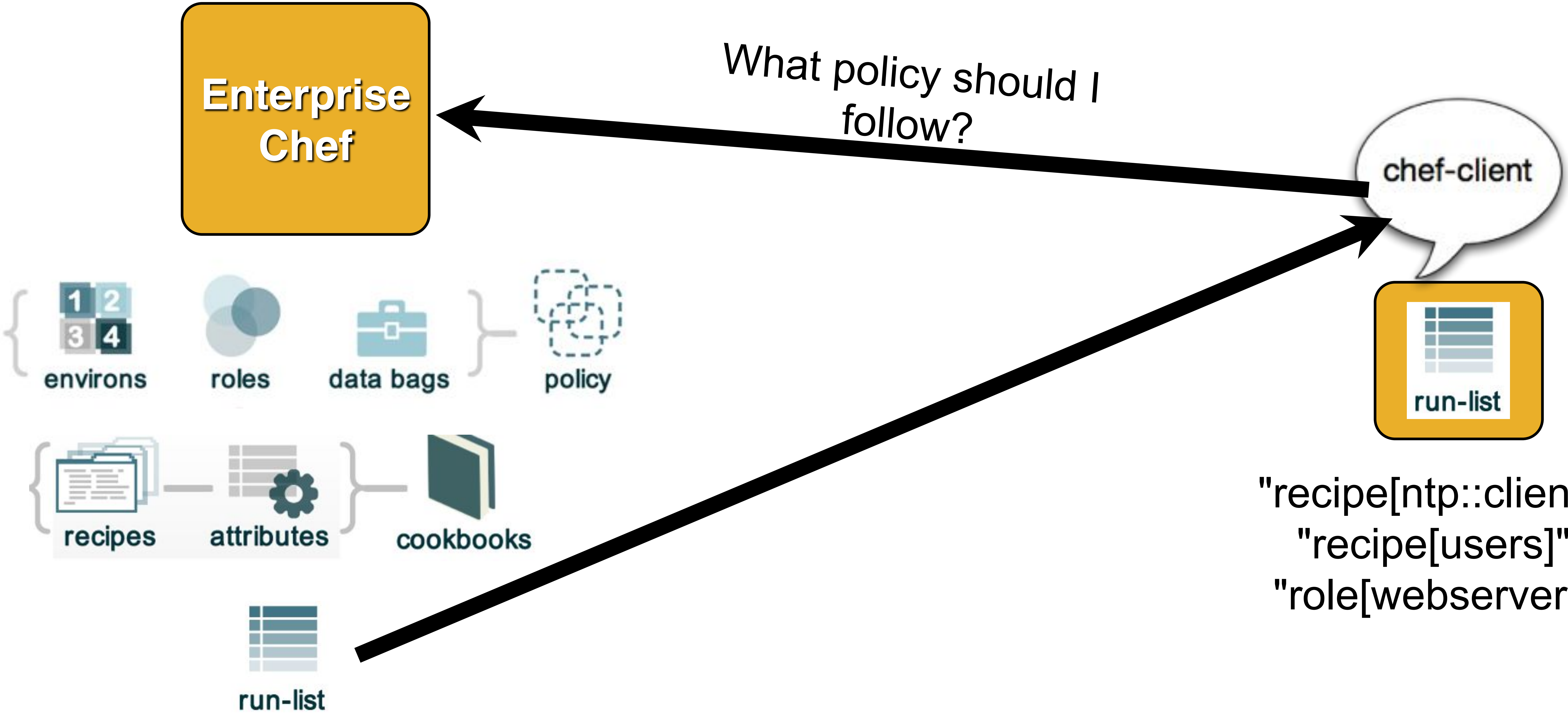
cookbooks

run-list

run-list

"recipe[ntp::client]"
"recipe[users]"
"role[webserver]"

Run List



"recipe[ntp::client]"
"recipe[users]"
"role[webserver]"

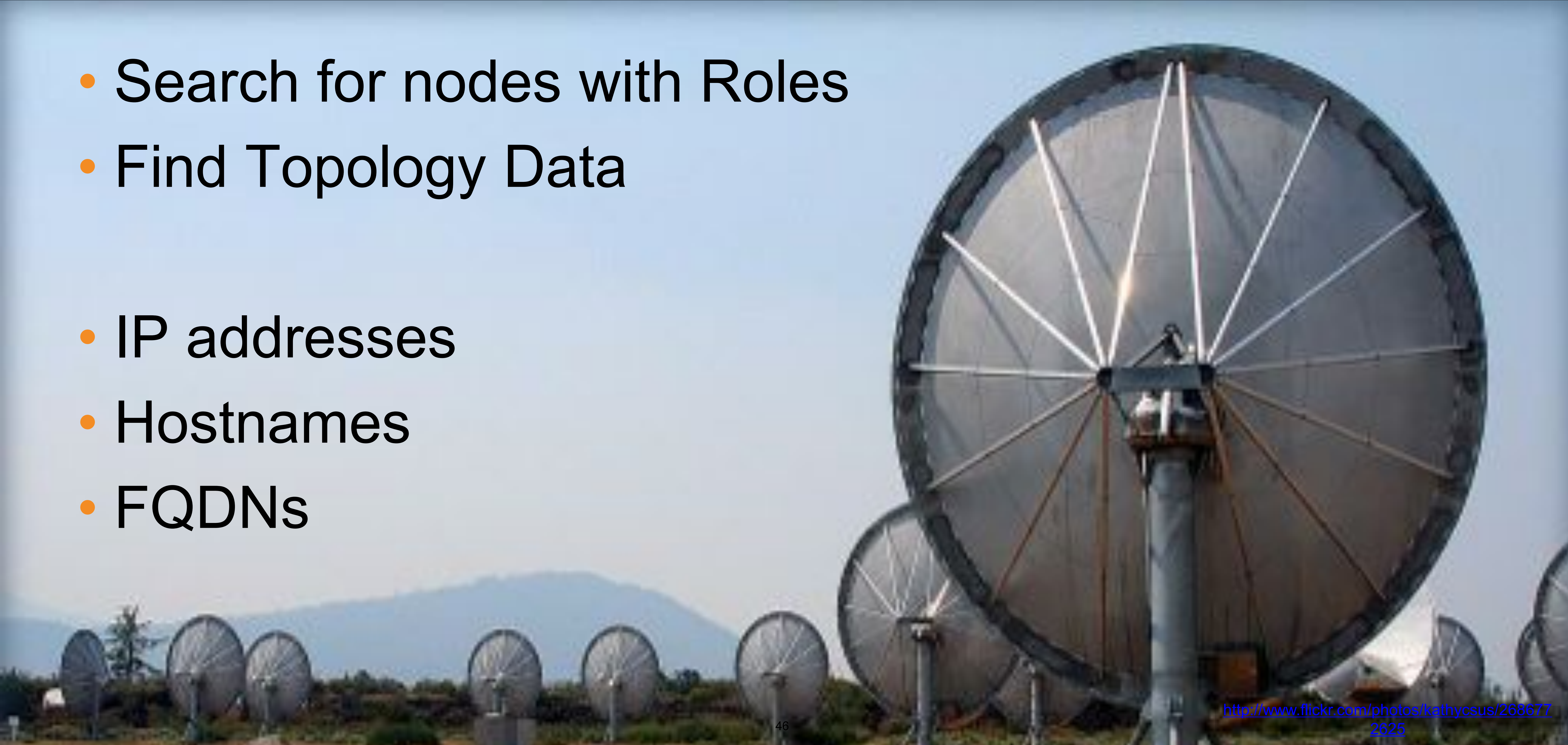
Run List Specifies Policy

- The Run List is an ordered collection of policies that the Node should follow
- Chef-client obtains the Run List from the Chef Server
- Chef-client ensures the Node complies with the policy in the Run List

Search

- Search for nodes with Roles
- Find Topology Data

- IP addresses
- Hostnames
- FQDNs



http://www.flickr.com/photos/kathycsus/2686772625

Search for Nodes

```
pool_members = search("node","role:webserver")

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy-app_lb.cfg.erb"
  owner "root"
  group "root"
  mode 0644
  variables :pool_members => pool_members.uniq
  notifies :restart, "service[haproxy]"
end
```

Search for Nodes

```
pool_members = search("node","role:webserver")

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy-app_lb.cfg.erb"
  owner "root"
  group "root"
  mode 0644
  variables :pool_members => pool_members.uniq
  notifies :restart, "service[haproxy]"
end
```

Pass results into Templates

```
# Set up application listeners here.
listen application 0.0.0.0:80
  balance roundrobin
  <% @pool_members.each do |member| -%>
  server <%= member[:hostname] %> <%= member[:ipaddress] %>:>
weight 1 maxconn 1 check
  <% end -%>
<% if node["haproxy"]["enable_admin"] -%>
listen admin 0.0.0.0:22002
  mode http
  stats uri /
<% end -%>
```

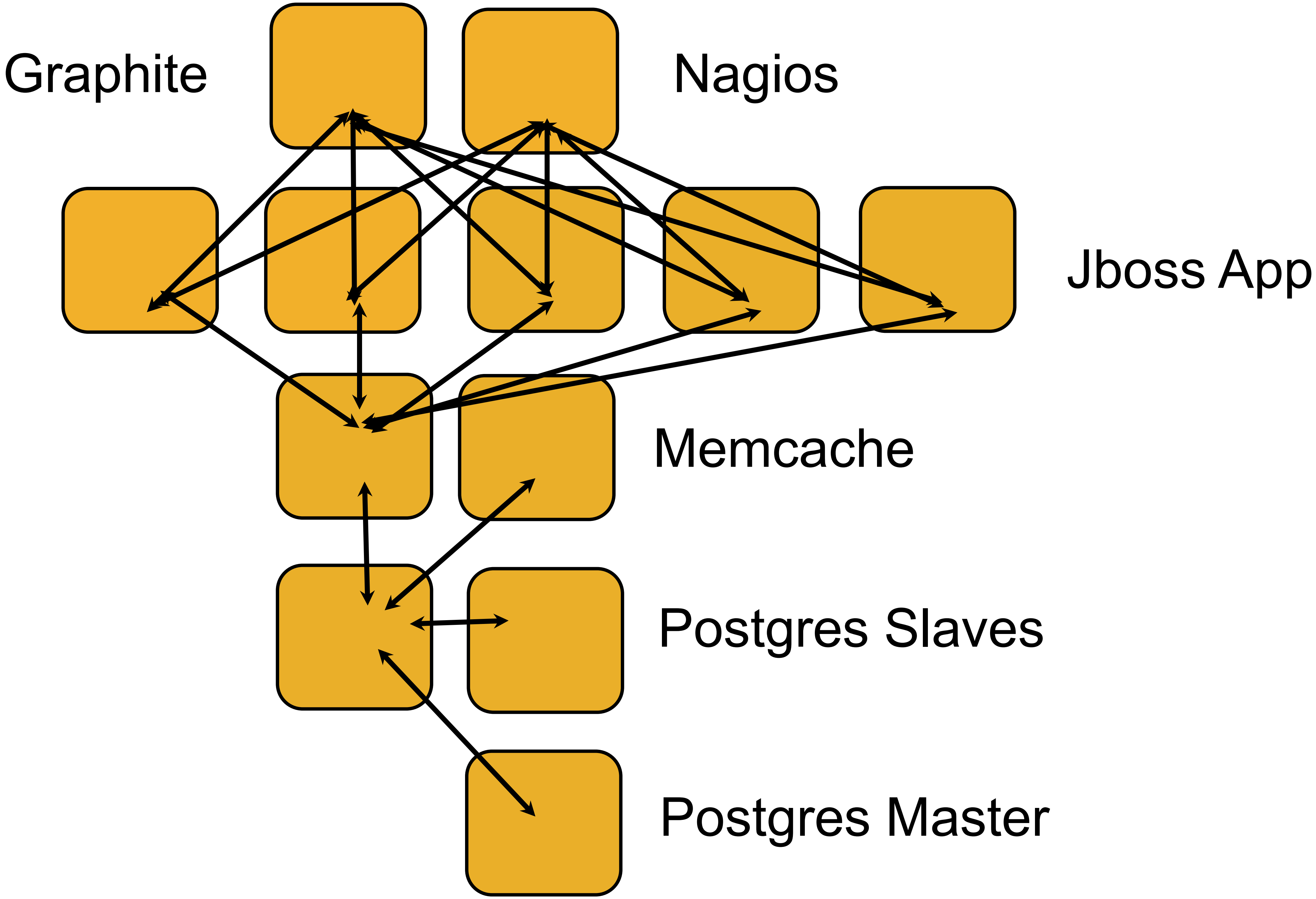
Pass results into Templates

```
# Set up application listeners here.
listen application 0.0.0.0:80
  balance roundrobin
  <% @pool_members.each do |member| -%>
  server <%= member[:hostname] %> <%= member[:ipaddress] %>:>
weight 1 maxconn 1 check
  <% end -%>
<% if node["haproxy"]["enable_admin"] -%>
listen admin 0.0.0.0:22002
  mode http
  stats uri /
<% end -%>
```

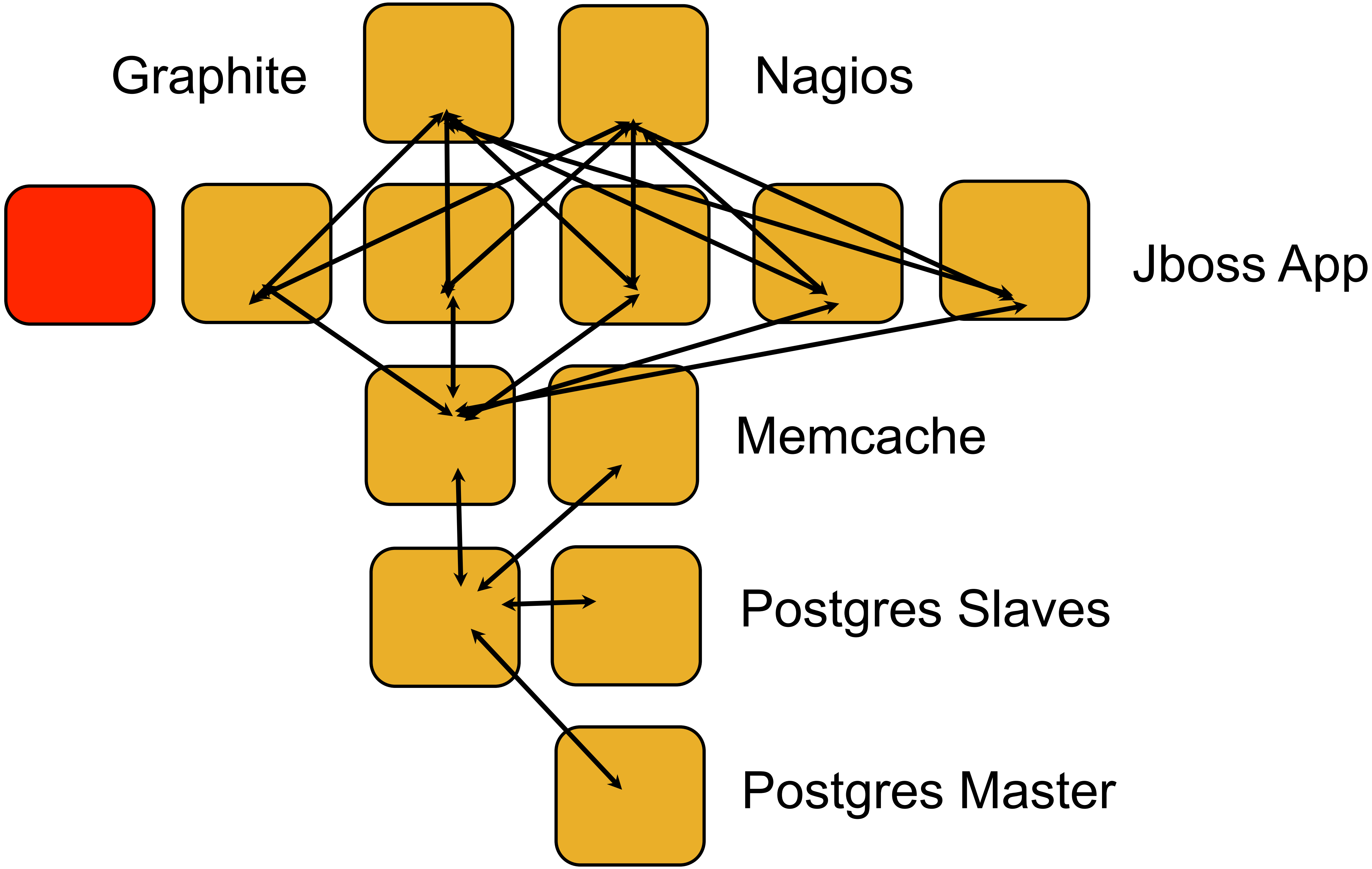
Pass results into Templates

```
# Set up application listeners here.
listen application 0.0.0.0:80
  balance roundrobin
  <% @pool_members.each do |member| -%>
  server <%= member[:hostname] %> <%= member[:ipaddress] %>:>
weight 1 maxconn 1 check
  <% end -%>
<% if node["haproxy"]["enable_admin"] -%>
listen admin 0.0.0.0:22002
  mode http
  stats uri /
<% end -%>
```

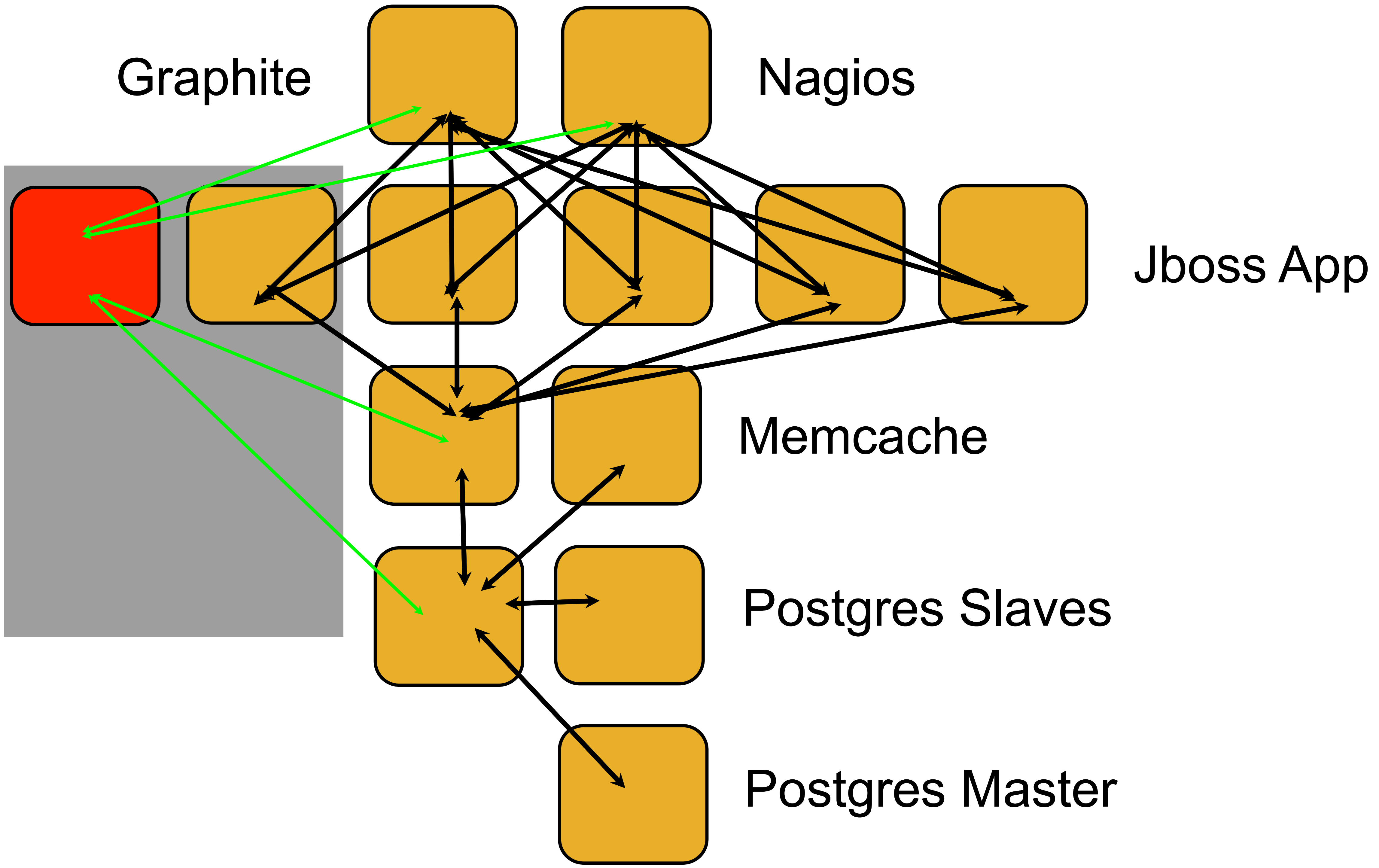
So when this...


Graphite
Nagios
Jboss App
Memcache
Postgres Slaves
Postgres Master

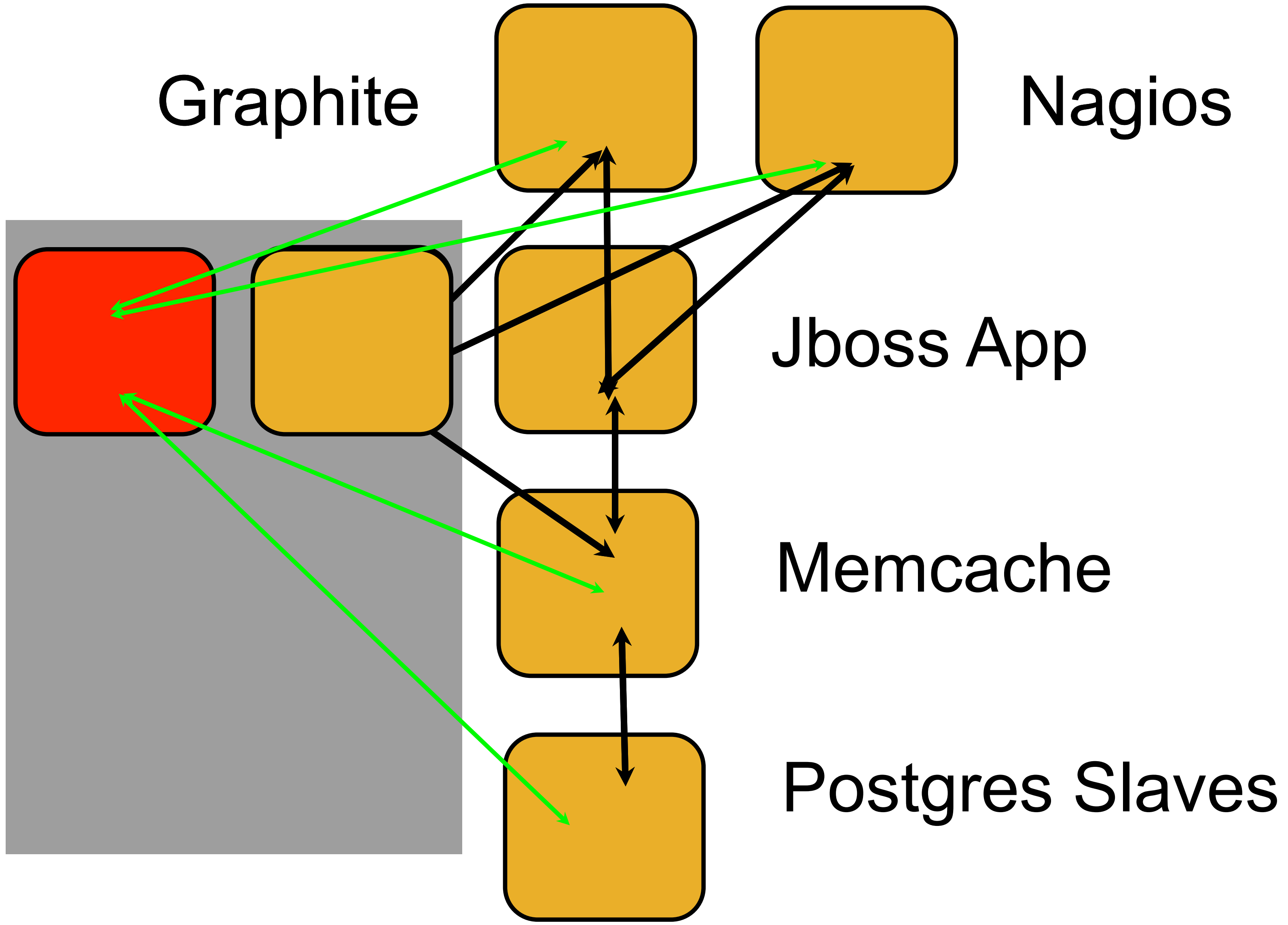
...becomes this



...this can happen automatically



Count the Resources



- 12+ resource changes for 1 node addition

- Load balancer config
- Nagios host ping
- Nagios host ssh
- Nagios host HTTP
- Nagios host app health
- Graphite CPU
- Graphite Memory
- Graphite Disk
- Graphite SNMP
- Memcache firewall
- Postgres firewall
- Postgres authZ config

Manage Complexity

- Determine the desired state of your infrastructure
- Identify the Resources required to meet that state
- Gather the Resources into Recipes
- Compose a Run List from Recipes and Roles
- Apply a Run List to each Node in your Environment
- Your infrastructure adheres to the policy modeled in Chef

Configuration Drift

- Configuration Drift happens when:
  - Your infrastructure requirements change
  - The configuration of a server falls out of policy

- Chef makes it easy to manage
  - Model the new requirements in your Chef configuration files
  - Run the chef-client to enforce your policies

Design Tenets of Chef

- Whipuptitude - aptitude for whipping things up
- Reasonability
- Sane defaults
- Flexibility
- Manipulexity - manipulation of complex things

WAS is the Java™ Foundation for IBM Software

Over 300 IBM offerings embed
or build upon WAS

Make WAS easy for developers!

- Ops use WAS for stability, scalability, performance …

Make WAS easy for developers!

- Ops use WAS for stability, scalability, performance …
- Dev less so. They have other requirements

Make WAS easy for developers!

- Ops use WAS for stability, scalability, performance …
- Dev less so. They have other requirements
  - Simple to configure

Make WAS easy for developers!

- Ops use WAS for stability, scalability, performance …
- Dev less so. They have other requirements
    - Simple to configure
    - Small and fast

Make WAS easy for developers!

- Ops use WAS for stability, scalability, performance …
- Dev less so. They have other requirements
  - Simple to configure
  - Small and fast
  - Quick to cycle through code, compile, test, debug in an IDE

Make WAS easy for developers!

- Ops use WAS for stability, scalability, performance …
- Dev less so. They have other requirements
  - Simple to configure
  - Small and fast
  - Quick to cycle through code, compile, test, debug in an IDE
  - Mac

Make WAS easy for developers!

- Ops use WAS for stability, scalability, performance …
- Dev less so. They have other requirements
  - Simple to configure
  - Small and fast
  - Quick to cycle through code, compile, test, debug in an IDE
  - Mac

⇒ Variety of appservers used in Dev

WAS v8.5.5 Liberty Profile &
WAS Developer Tools for Eclipse (WDT)

“Developer First” Focus

Simplified, shareable XML server config. New integrated messaging server, DynaCache support, new prog. models, such as Web Services, JMS & EJB-Lite.

WAS v8.5.5 Liberty Profile &
WAS Developer Tools for Eclipse (WDT)

"Developer First" Focus
Simplified, shareable XML server config. New integrated messaging server, DynaCache support, new prog. models, such as Web Services, JMS & EJB-Lite.

Small Download
50MB for Web Profile features

WAS v8.5.5 Liberty Profile & WAS Developer Tools for Eclipse (WDT)

"Developer First" Focus
Simplified, shareable XML server config. New integrated messaging server, DynaCache support, new prog. models, such as Web Services, JMS & EJB-Lite.

Small Download
50MB for Web Profile features

Dynamically Extensible
Install new features from repository (local or remote) with no svr restart

WAS v8.5.5 Liberty Profile & WAS Developer Tools for Eclipse (WDT)

"Developer First" Focus
Simplified, shareable XML server config. New integrated messaging server, DynaCache support, new prog. models, such as Web Services, JMS & EJB-Lite.

Small Download
50MB for Web Profile features

WAS v8.5.5 Liberty Profile & WAS Developer Tools for Eclipse (WDT)

Dynamically Extensible
Install new features from repository (local or remote) with no svr restart

Lightweight cluster Mgmt
Liberty servers can join a lightweight cluster for workload balancing and high availability

WAS v8.5.5 Liberty Profile & WAS Developer Tools for Eclipse (WDT)

"Developer First" Focus
Simplified, shareable XML server config. New integrated messaging server, DynaCache support, new prog. models, such as Web Services, JMS & EJB-Lite.

Small Download
50MB for Web Profile features

Dynamically Extensible
Install new features from repository (local or remote) with no svr restart

Lightweight cluster Mgmt
Liberty servers can join a lightweight cluster for workload balancing and high availability

Fidelity to full profile WAS
Same reliable containers & QOS. Develop on Liberty profile and deploy to Liberty or full-profile WAS

WAS v8.5.5 Liberty Profile & WAS Developer Tools for Eclipse (WDT)

"Developer First" Focus
Simplified, shareable XML server config. New integrated messaging server, DynaCache support, new prog. models, such as Web Services, JMS & EJB-Lite.

Small Download
50MB for Web Profile features

Dynamically Extensible
Install new features from repository (local or remote) with no svr restart

Lightweight cluster Mgmt
Liberty servers can join a lightweight cluster for workload balancing and high availability

Liberty Extensions
Add custom features and integrate 3rd party components via Liberty extensions interface

Fidelity to full profile WAS
Same reliable containers & QOS. Develop on Liberty profile and deploy to Liberty or full-profile WAS

WAS v8.5.5 Liberty Profile & WAS Developer Tools for Eclipse (WDT)

"Developer First" Focus
Simplified, shareable XML server config. New integrated messaging server, DynaCache support, new prog. models, such as Web Services, JMS & EJB-Lite.

Small Download
50MB for Web Profile features

Dynamically Extensible
Install new features from repository (local or remote) with no svr restart

Lightweight cluster Mgmt
Liberty servers can join a lightweight cluster for workload balancing and high availability

Unzip install and deploy
IM or unzip to install. New option to deploy "server package" of app + config + required subset of server runtime for highest density deploy

Liberty Extensions
Add custom features and integrate 3rd party components via Liberty extensions interface

Fidelity to full profile WAS
Same reliable containers & QOS. Develop on Liberty profile and deploy to Liberty or full-profile WAS

WAS v8.5.5 Liberty Profile & WAS Developer Tools for Eclipse (WDT)

"Developer First" Focus
Simplified, shareable XML server config. New integrated messaging server, DynaCache support, new prog. models, such as Web Services, JMS & EJB-Lite.

Small Download
50MB for Web Profile features

Dynamically Extensible
Install new features from repository (local or remote) with no svr restart

Lightweight cluster Mgmt
Liberty servers can join a lightweight cluster for workload balancing and high availability

Web Profile Certified
Create web apps for the Java EE Web Profile standard.

Unzip install and deploy
IM or unzip to install. New option to deploy "server package" of app + config + required subset of server runtime for highest density deploy

Liberty Extensions
Add custom features and integrate 3rd party components via Liberty extensions interface

Fidelity to full profile WAS
Same reliable containers & QOS. Develop on Liberty profile and deploy to Liberty or full-profile WAS

WAS v8.5.5 Liberty Profile & WAS Developer Tools for Eclipse (WDT)

"Developer First" Focus
Simplified, shareable XML server config. New integrated messaging server, DynaCache support, new prog. models, such as Web Services, JMS & EJB-Lite.

Small Download
50MB for Web Profile features

Dynamically Extensible
Install new features from repository (local or remote) with no svr restart

Lightweight cluster Mgmt
Liberty servers can join a lightweight cluster for workload balancing and high availability

Fidelity to full profile WAS
Same reliable containers & QOS. Develop on Liberty profile and deploy to Liberty or full-profile WAS

Liberty Extensions
Add custom features and integrate 3rd party components via Liberty extensions interface

Unzip install and deploy
IM or unzip to install. New option to deploy "server package" of app + config + required subset of server runtime for highest density deploy

Web Profile Certified
Create web apps for the Java EE Web Profile standard.

Integrated tools
Powerful tools in WDT Eclipse feature. Enhanced for v8.5.5 prog models, Maven integration, ++

WAS v8.5.5 Liberty Profile & WAS Developer Tools for Eclipse (WDT)

"Developer First" Focus
Simplified, shareable XML server config. New integrated messaging server, DynaCache support, new prog. models, such as Web Services, JMS & EJB-Lite.

Start fast, run efficiently
Starts in <3s; Mem footprint <50MB; (TradeLite benchmark)

Small Download
50MB for Web Profile features

Integrated tools
Powerful tools in WDT Eclipse feature. Enhanced for v8.5.5 prog models, Maven integration, ++

Dynamically Extensible
Install new features from repository (local or remote) with no svr restart

Web Profile Certified
Create web apps for the Java EE Web Profile standard.

Lightweight cluster Mgmt
Liberty servers can join a lightweight cluster for workload balancing and high availability

Unzip install and deploy
IM or unzip to install. New option to deploy "server package" of app + config + required subset of server runtime for highest density deploy

Liberty Extensions
Add custom features and integrate 3rd party components via Liberty extensions interface

Fidelity to full profile WAS
Same reliable containers & QOS. Develop on Liberty profile and deploy to Liberty or full-profile WAS

WAS v8.5.5 Liberty Profile & WAS Developer Tools for Eclipse (WDT)

Dynamic Server Profile
Not static like Web Profile; configured by app at a fine-grained level

"Developer First" Focus
Simplified, shareable XML server config. New integrated messaging server, DynaCache support, new prog. models, such as Web Services, JMS & EJB-Lite.

Start fast, run efficiently
Starts in <3s; Mem footprint <50MB; (TradeLite benchmark)

Small Download
50MB for Web Profile features

Integrated tools
Powerful tools in WDT Eclipse feature. Enhanced for v8.5.5 prog models, Maven integration, ++

Dynamically Extensible
Install new features from repository (local or remote) with no svr restart

Web Profile Certified
Create web apps for the Java EE Web Profile standard.

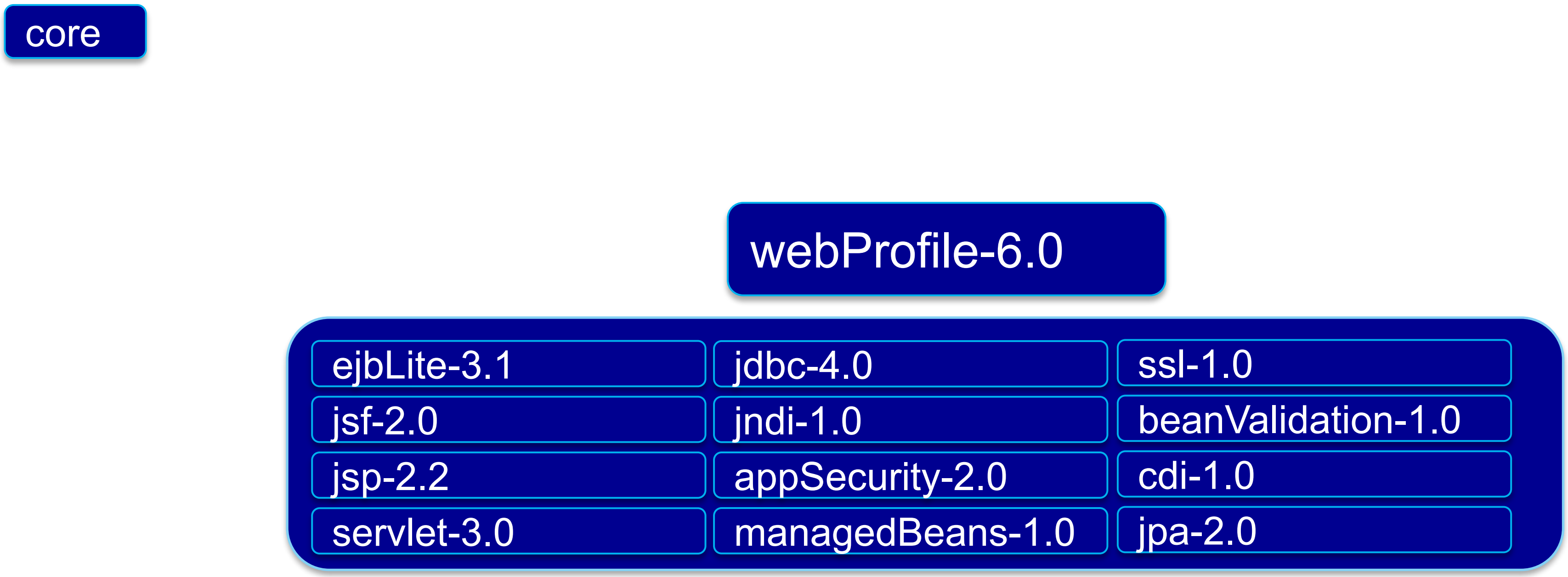
Lightweight cluster Mgmt
Liberty servers can join a lightweight cluster for workload balancing and high availability

Unzip install and deploy
IM or unzip to install. New option to deploy "server package" of app + config + required subset of server runtime for highest density deploy

Liberty Extensions
Add custom features and integrate 3rd party components via Liberty extensions interface

Fidelity to full profile WAS
Same reliable containers & QOS. Develop on Liberty profile and deploy to Liberty or full-profile WAS

Composable: features in 8.5.5

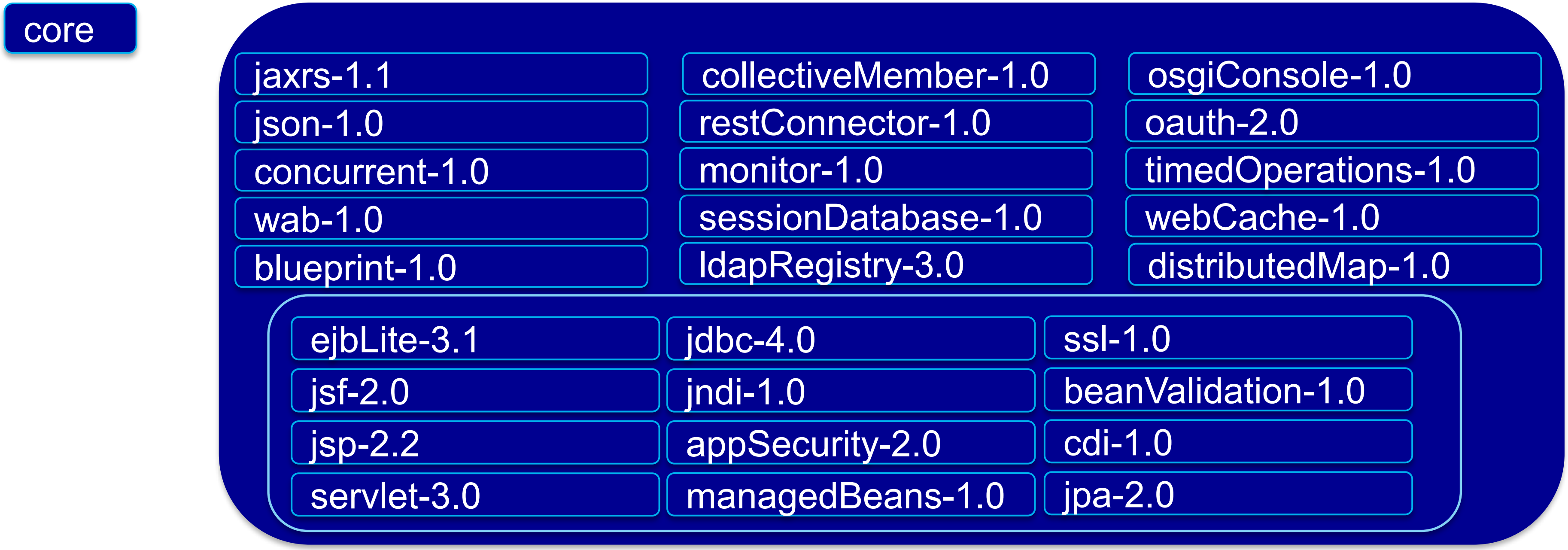


core

webProfile-6.0

ejbLite-3.1	jdbc-4.0	ssl-1.0
jsf-2.0	jndi-1.0	beanValidation-1.0
jsp-2.2	appSecurity-2.0	cdi-1.0
servlet-3.0	managedBeans-1.0	jpa-2.0

Composable: features in 8.5.5



core

jaxrs-1.1
json-1.0
concurrent-1.0
wab-1.0
blueprint-1.0

collectiveMember-1.0
restConnector-1.0
monitor-1.0
sessionDatabase-1.0
ldapRegistry-3.0

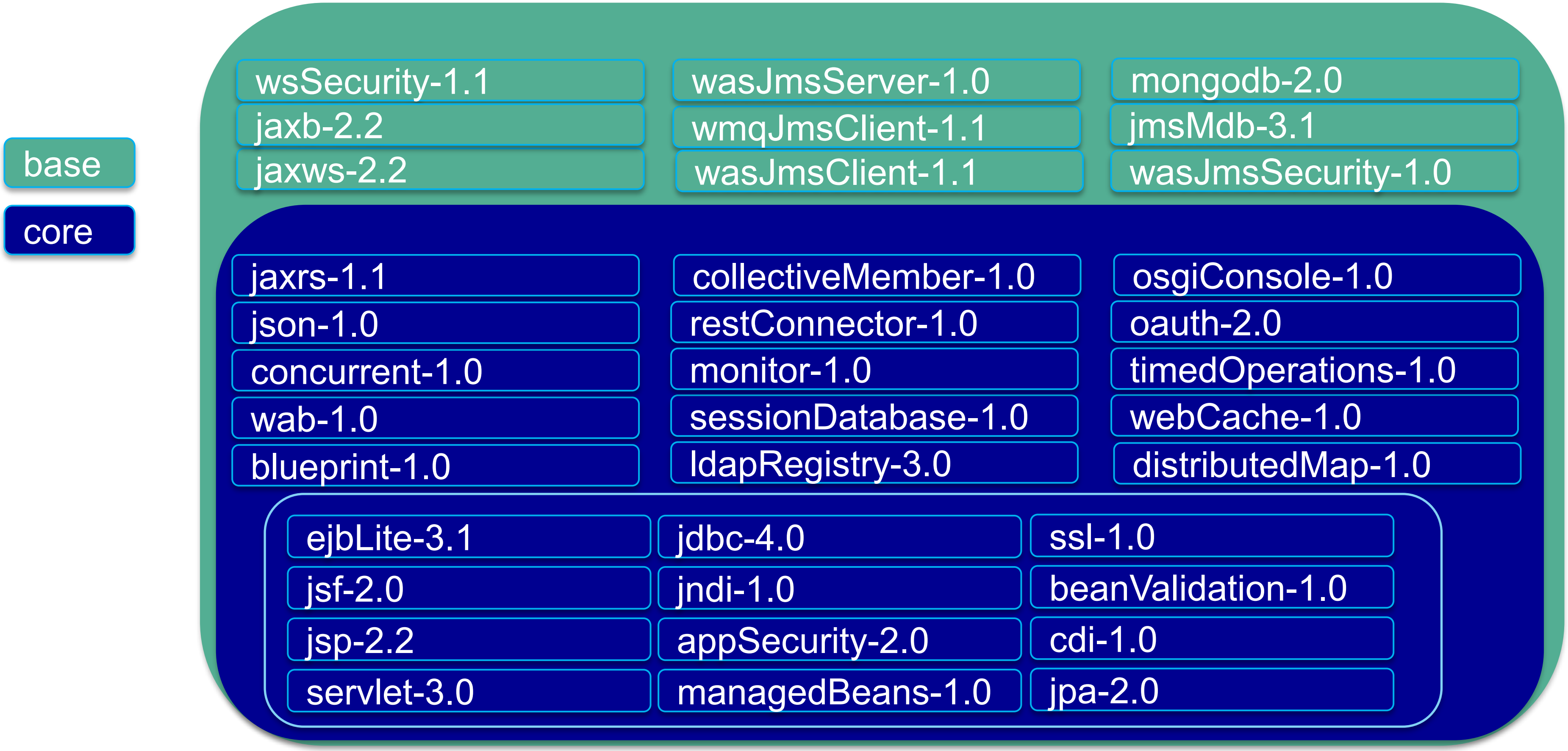
osgiConsole-1.0
oauth-2.0
timedOperations-1.0
webCache-1.0
distributedMap-1.0

ejbLite-3.1
jsf-2.0
jsp-2.2
servlet-3.0

jdbc-4.0
jndi-1.0
appSecurity-2.0
managedBeans-1.0

ssl-1.0
beanValidation-1.0
cdi-1.0
jpa-2.0

Composable: features in 8.5.5



base

core

wsSecurity-1.1
jaxb-2.2
jaxws-2.2

wasJmsServer-1.0
wmqJmsClient-1.1
wasJmsClient-1.1

mongodb-2.0
jmsMdb-3.1
wasJmsSecurity-1.0

jaxrs-1.1
json-1.0
concurrent-1.0
wab-1.0
blueprint-1.0

collectiveMember-1.0
restConnector-1.0
monitor-1.0
sessionDatabase-1.0
ldapRegistry-3.0

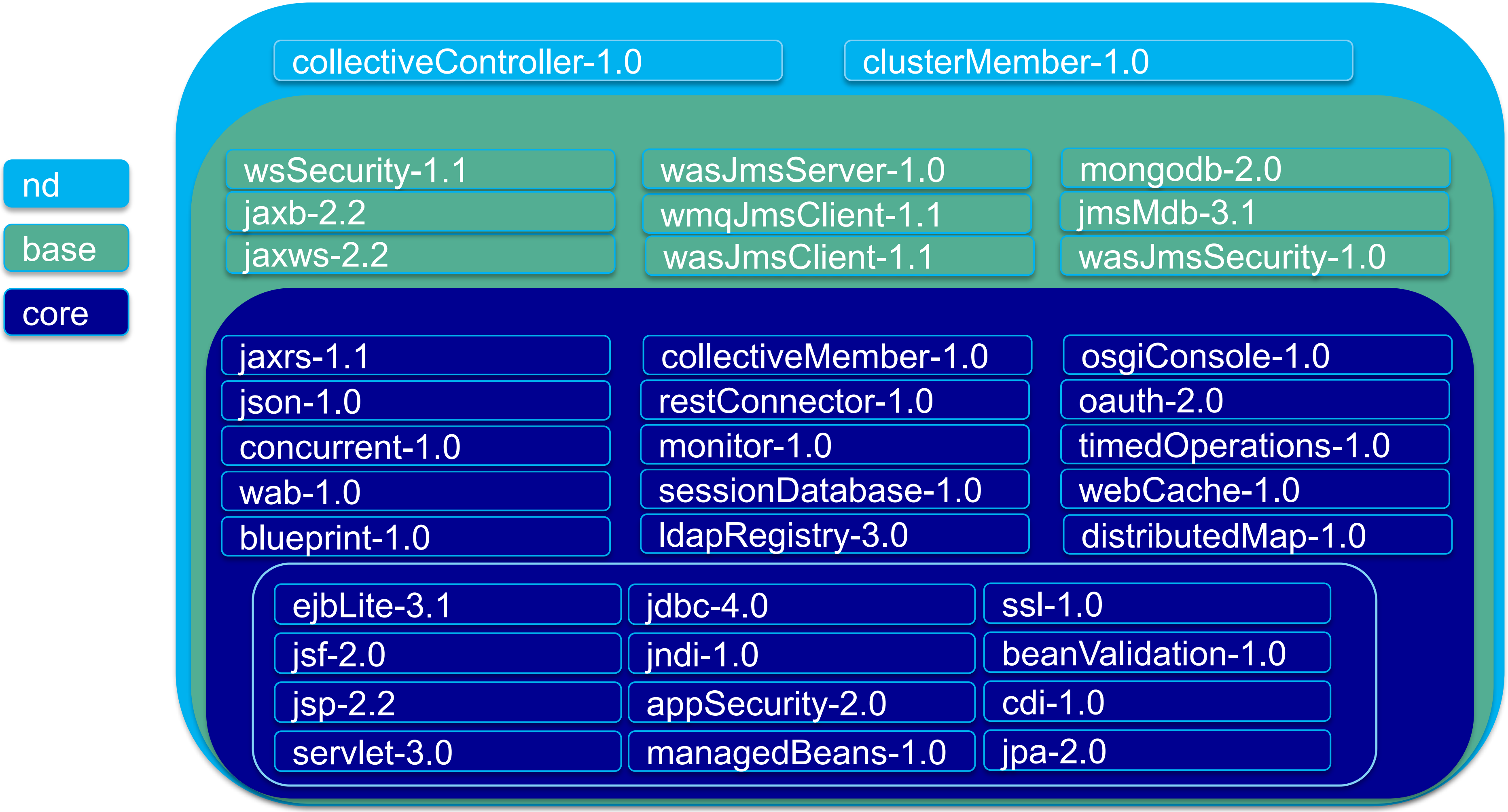
osgiConsole-1.0
oauth-2.0
timedOperations-1.0
webCache-1.0
distributedMap-1.0

ejbLite-3.1
jsf-2.0
jsp-2.2
servlet-3.0

jdbc-4.0
jndi-1.0
appSecurity-2.0
managedBeans-1.0

ssl-1.0
beanValidation-1.0
cdi-1.0
jpa-2.0

Composable: features in 8.5.5



nd

base

core

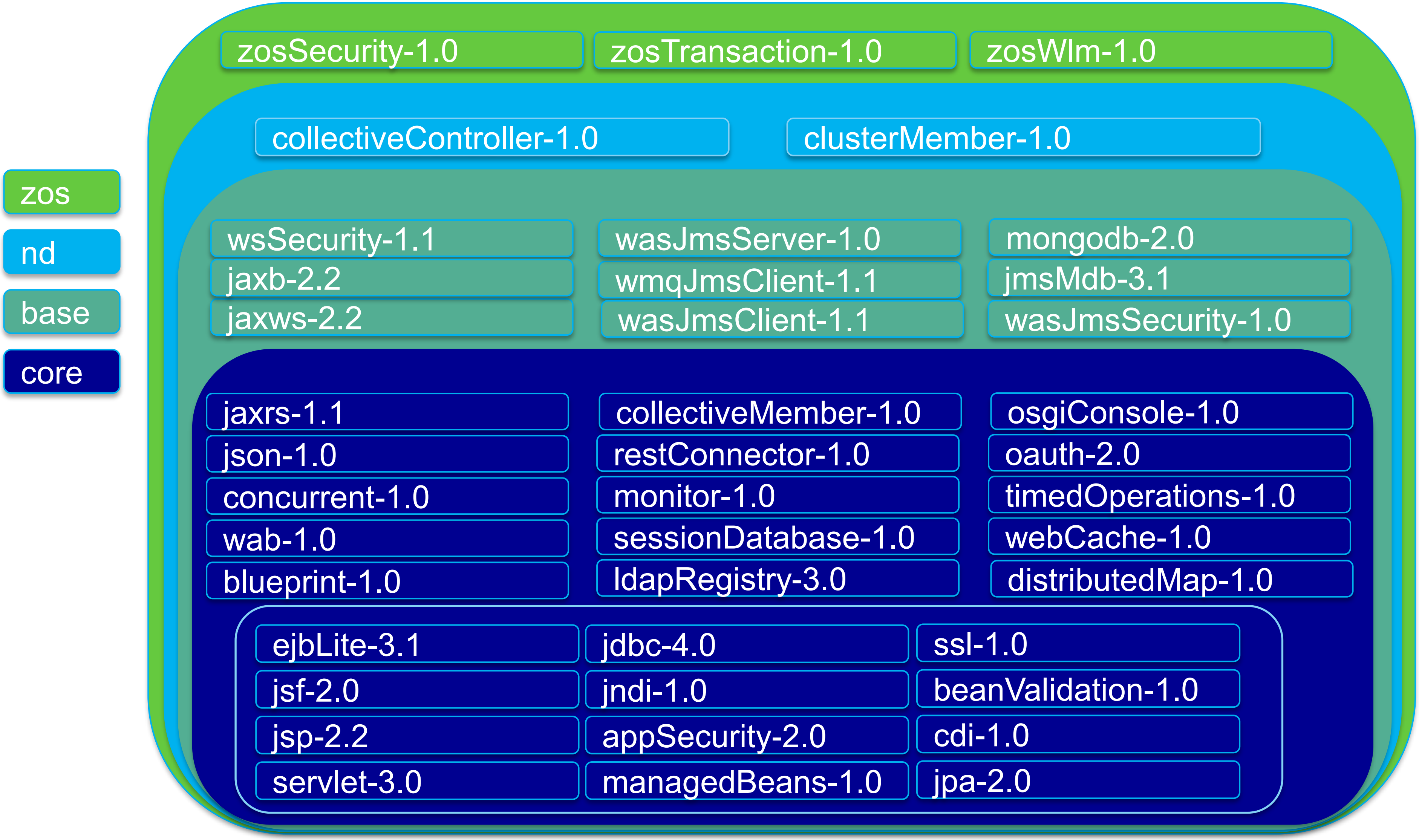
collectiveController-1.0	clusterMember-1.0

wsSecurity-1.1	wasJmsServer-1.0	mongodb-2.0
jaxb-2.2	wmqJmsClient-1.1	jmsMdb-3.1
jaxws-2.2	wasJmsClient-1.1	wasJmsSecurity-1.0

jaxrs-1.1	collectiveMember-1.0	osgiConsole-1.0
json-1.0	restConnector-1.0	oauth-2.0
concurrent-1.0	monitor-1.0	timedOperations-1.0
wab-1.0	sessionDatabase-1.0	webCache-1.0
blueprint-1.0	ldapRegistry-3.0	distributedMap-1.0

ejbLite-3.1	jdbc-4.0	ssl-1.0
jsf-2.0	jndi-1.0	beanValidation-1.0
jsp-2.2	appSecurity-2.0	cdi-1.0
servlet-3.0	managedBeans-1.0	jpa-2.0

Composable: features in 8.5.5



zos

nd

base

core

zosSecurity-1.0
zosTransaction-1.0
zosWlm-1.0

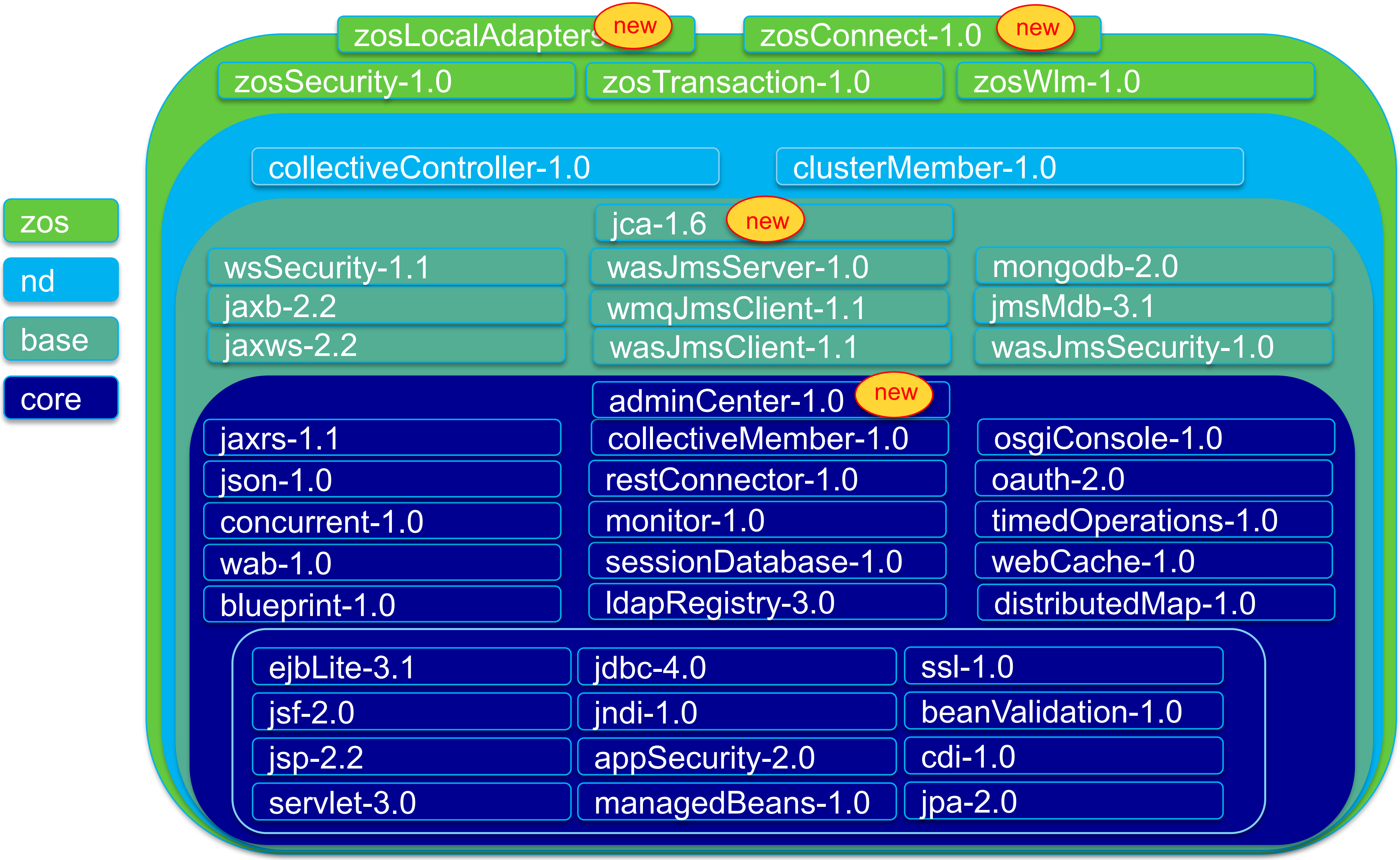
collectiveController-1.0
clusterMember-1.0

wsSecurity-1.1
jaxb-2.2
jaxws-2.2
wasJmsServer-1.0
wmqJmsClient-1.1
wasJmsClient-1.1
mongodb-2.0
jmsMdb-3.1
wasJmsSecurity-1.0

jaxrs-1.1
json-1.0
concurrent-1.0
wab-1.0
blueprint-1.0
collectiveMember-1.0
restConnector-1.0
monitor-1.0
sessionDatabase-1.0
ldapRegistry-3.0
osgiConsole-1.0
oauth-2.0
timedOperations-1.0
webCache-1.0
distributedMap-1.0

ejbLite-3.1
jsf-2.0
jsp-2.2
servlet-3.0
jdbc-4.0
jndi-1.0
appSecurity-2.0
managedBeans-1.0
ssl-1.0
beanValidation-1.0
cdi-1.0
jpa-2.0

Composable: features in 8.5.5



zos

nd

base

core

zosLocalAdapters new

zosConnect-1.0 new

zosSecurity-1.0

zosTransaction-1.0

zosWlm-1.0

collectiveController-1.0

clusterMember-1.0

jca-1.6 new

wsSecurity-1.1

jaxb-2.2

jaxws-2.2

wasJmsServer-1.0

wmqJmsClient-1.1

wasJmsClient-1.1

mongodb-2.0

jmsMdb-3.1

wasJmsSecurity-1.0

adminCenter-1.0 new

jaxrs-1.1

json-1.0

concurrent-1.0

wab-1.0

blueprint-1.0

collectiveMember-1.0

restConnector-1.0

monitor-1.0

sessionDatabase-1.0

ldapRegistry-3.0

osgiConsole-1.0

oauth-2.0

timedOperations-1.0

webCache-1.0

distributedMap-1.0

ejbLite-3.1

jsf-2.0

jsp-2.2

servlet-3.0

jdbc-4.0

jndi-1.0

appSecurity-2.0

managedBeans-1.0

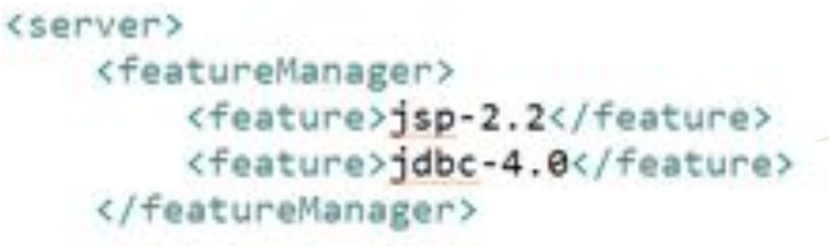
ssl-1.0

beanValidation-1.0

cdi-1.0

jpa-2.0

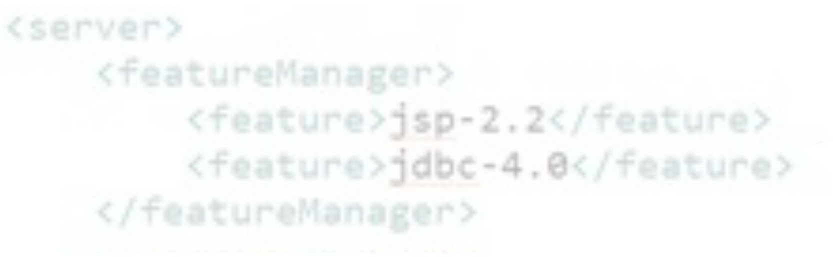
Simplified Server Configuration



```
<server>
    <featureManager>
        <feature>jsp-2.2</feature>
        <feature>jdbc-4.0</feature>
    </featureManager>
```

Features control which capabilities (bundles) are installed in the server

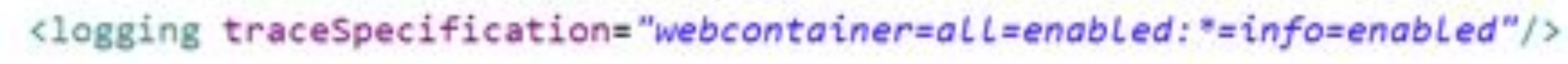
Simplified Server Configuration



```
<server>
    <featureManager>
        <feature>jsp-2.2</feature>
        <feature>jdbc-4.0</feature>
    </featureManager>
```

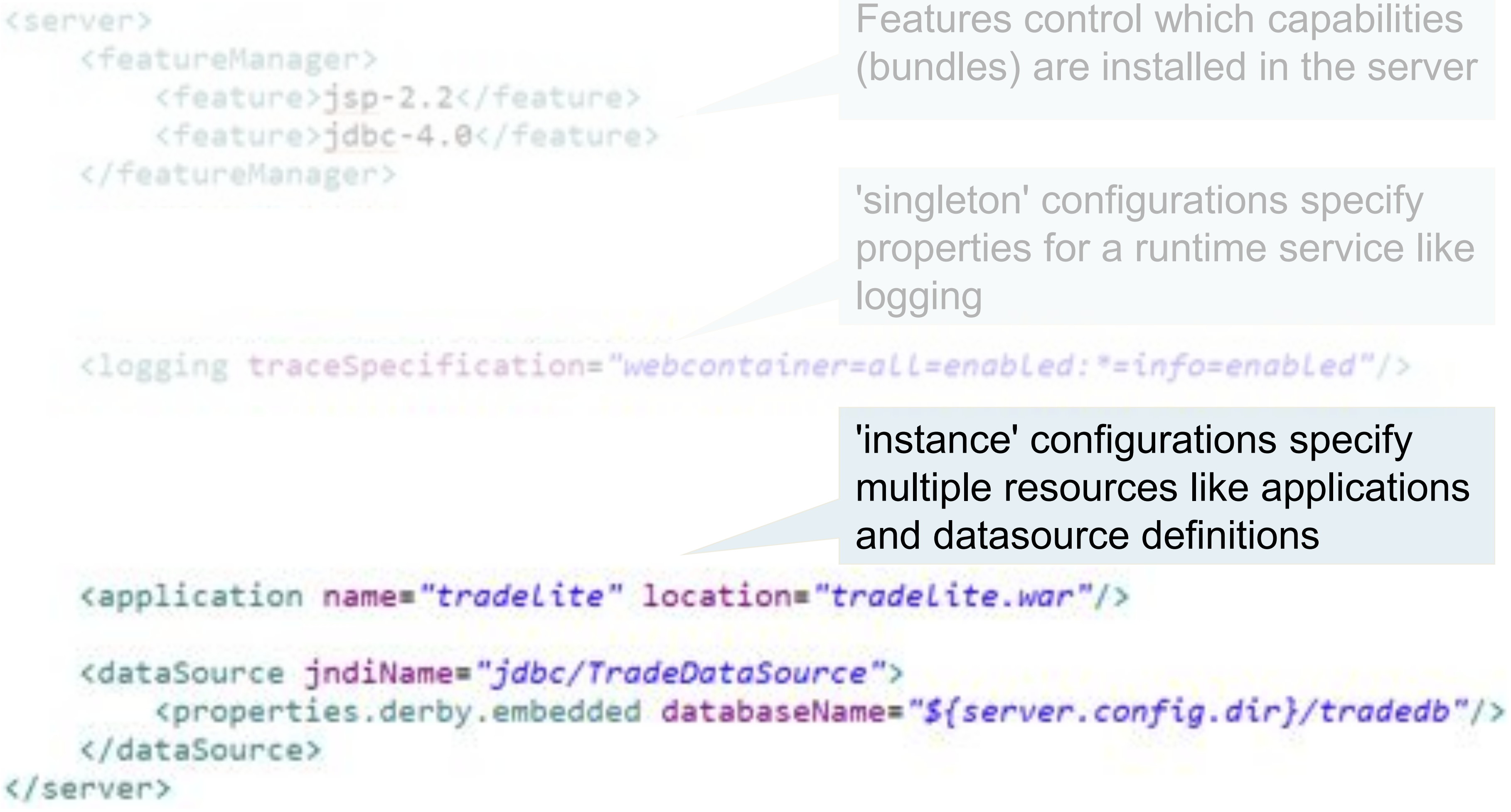
Features control which capabilities (bundles) are installed in the server

'singleton' configurations specify properties for a runtime service like logging



```
<logging traceSpecification="webcontainer=all=enabled:*=info=enabled"/>
```

Simplified Server Configuration



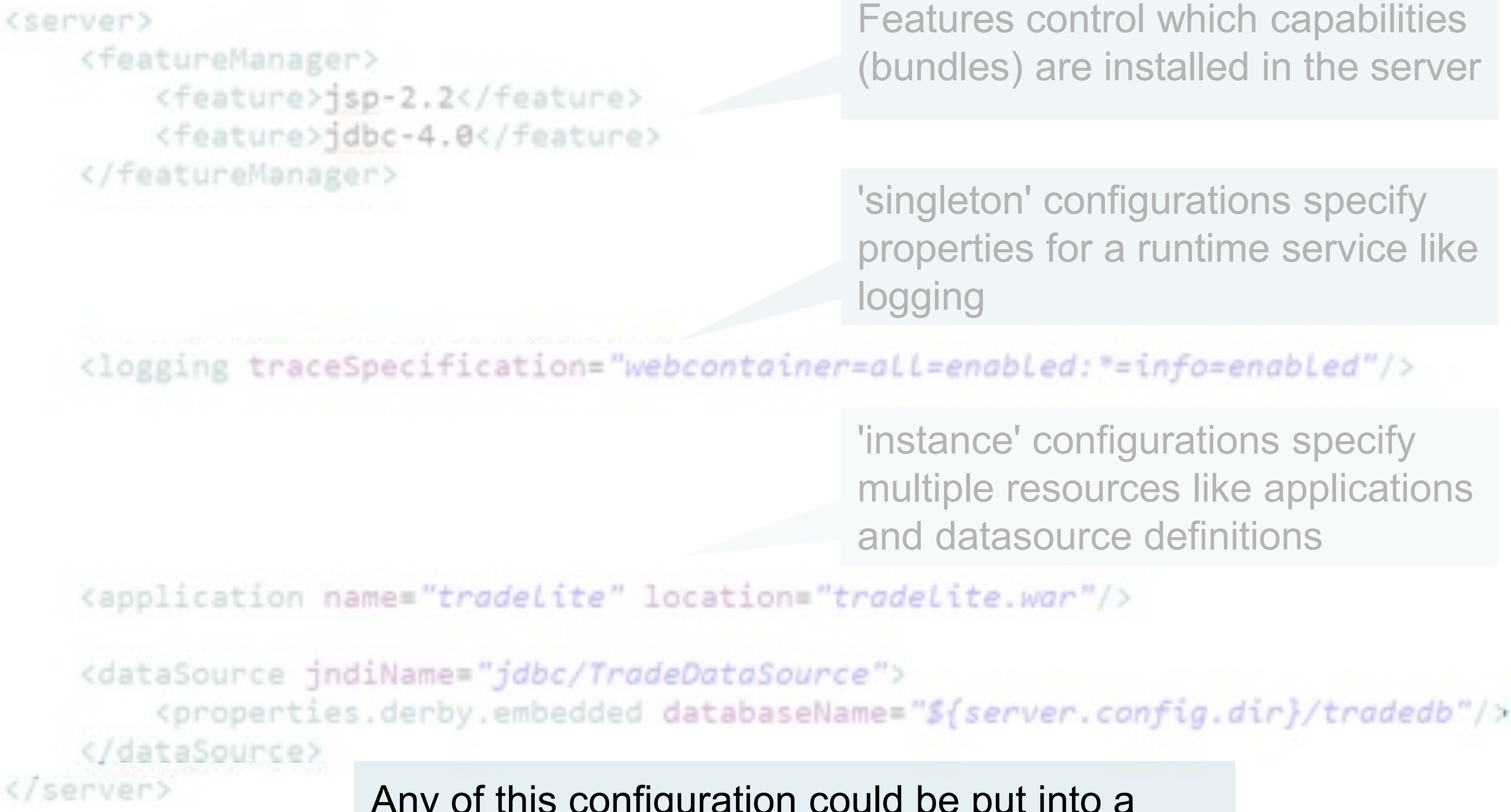
```
<server>
    <featureManager>
        <feature>jsp-2.2</feature>
        <feature>jdbc-4.0</feature>
    </featureManager>

    <logging traceSpecification="webcontainer=all=enabled:*=info=enabled"/>

    <application name="tradelite" location="tradelite.war"/>

    <dataSource jndiName="jdbc/TradeDataSource">
        <properties.derby.embedded databaseName="${server.config.dir}/tradedb"/>
    </dataSource>
</server>
```

Features control which capabilities (bundles) are installed in the server

'singleton' configurations specify properties for a runtime service like logging

'instance' configurations specify multiple resources like applications and datasource definitions

Simplified Server Configuration



```
<server>
    <featureManager>
        <feature>jsp-2.2</feature>
        <feature>jdbc-4.0</feature>
    </featureManager>

    <logging traceSpecification="webcontainer=all=enabled:*=info=enabled"/>

    <application name="tradelite" location="tradelite.war"/>

    <dataSource jndiName="jdbc/TradeDataSource">
        <properties.derby.embedded databaseName="${server.config.dir}/tradedb"/>
    </dataSource>
</server>
```

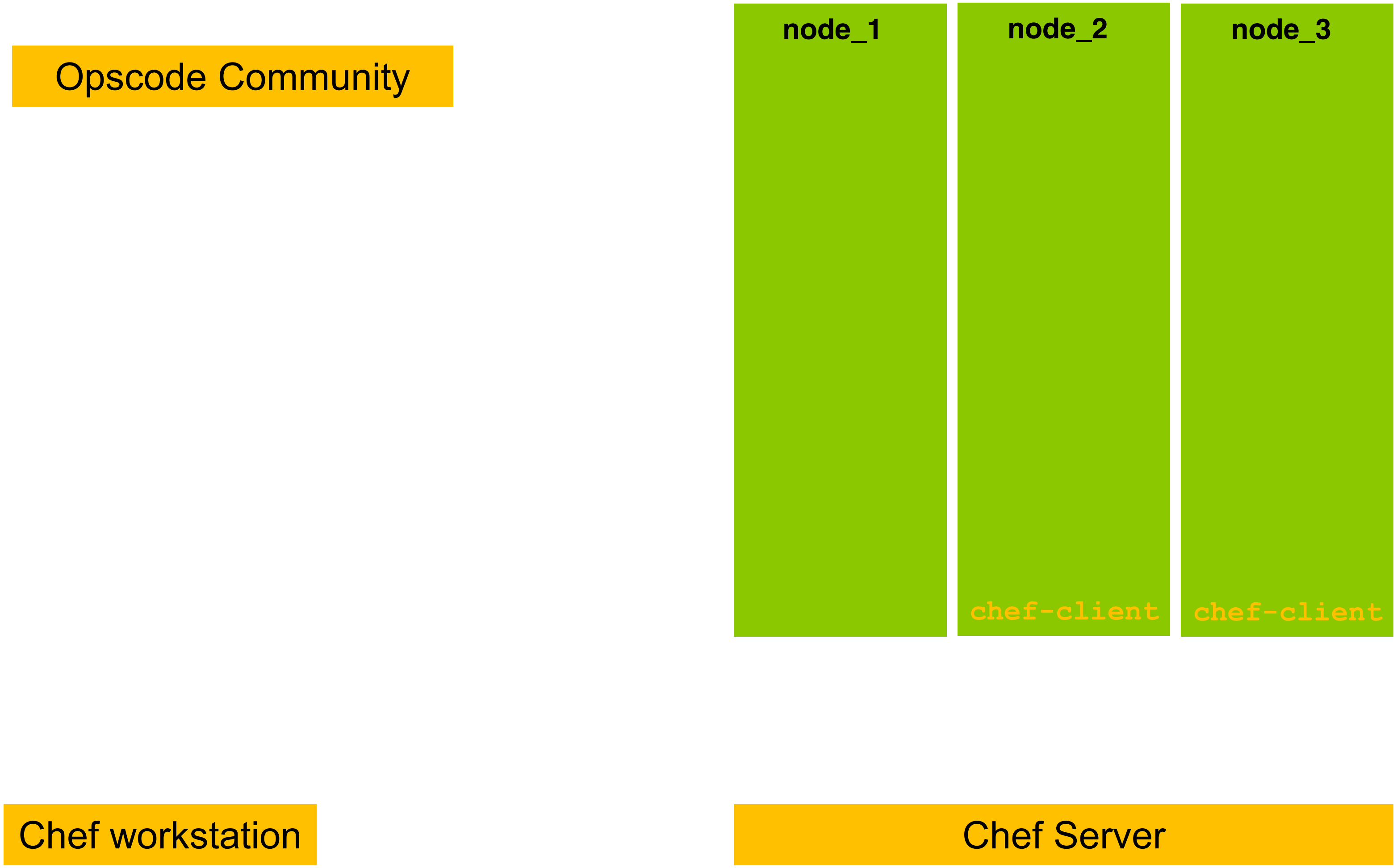
Features control which capabilities (bundles) are installed in the server

'singleton' configurations specify properties for a runtime service like logging

'instance' configurations specify multiple resources like applications and datasource definitions

Any of this configuration could be put into a separate xml file and 'included' in this 'master' configuration file

Chef node setup



Opscode Community

node_1

node_2

chef-client

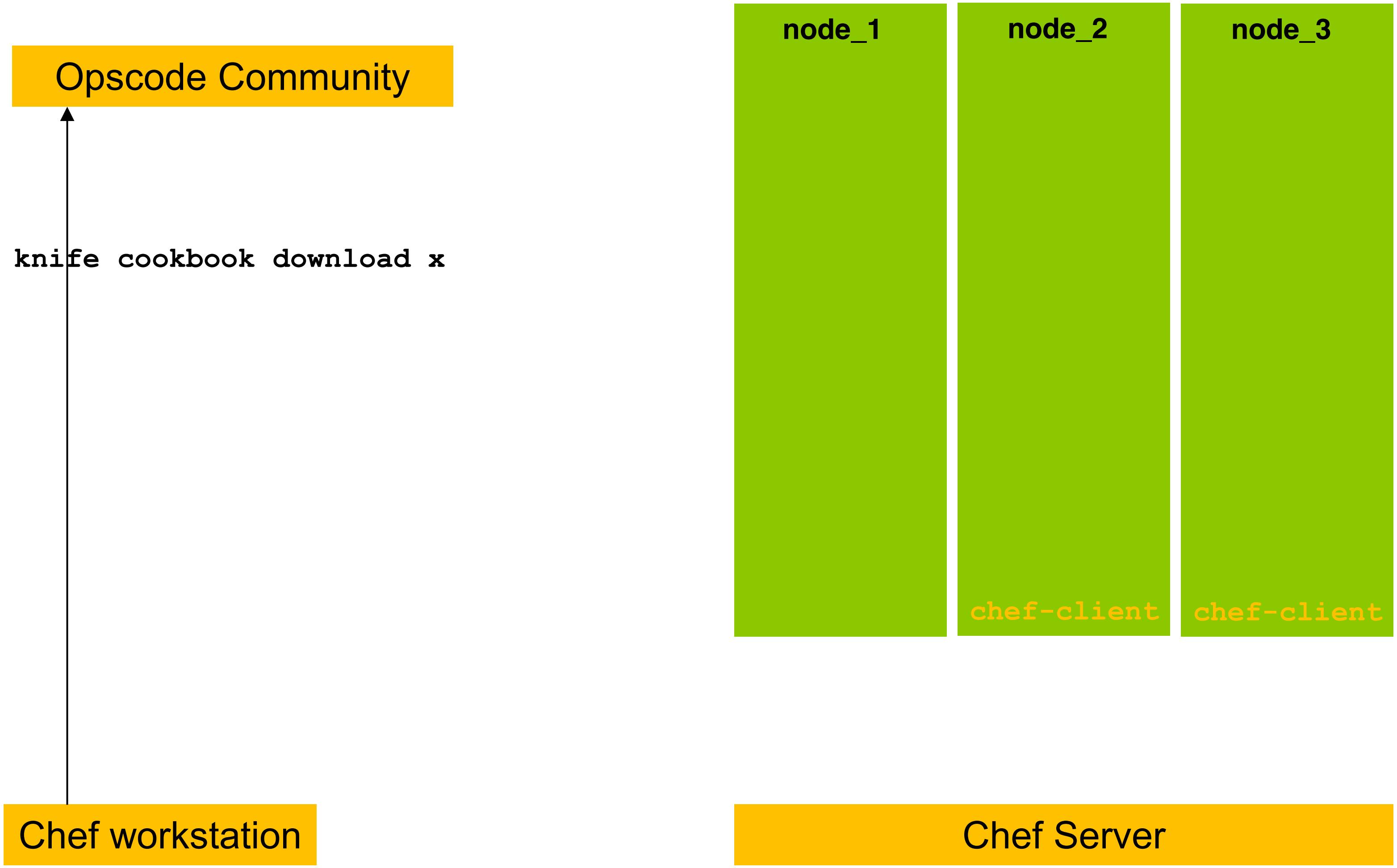
node_3

chef-client

Chef workstation

Chef Server

Chef node setup



Opscode Community

knife cookbook download x

Chef workstation

node_1

node_2

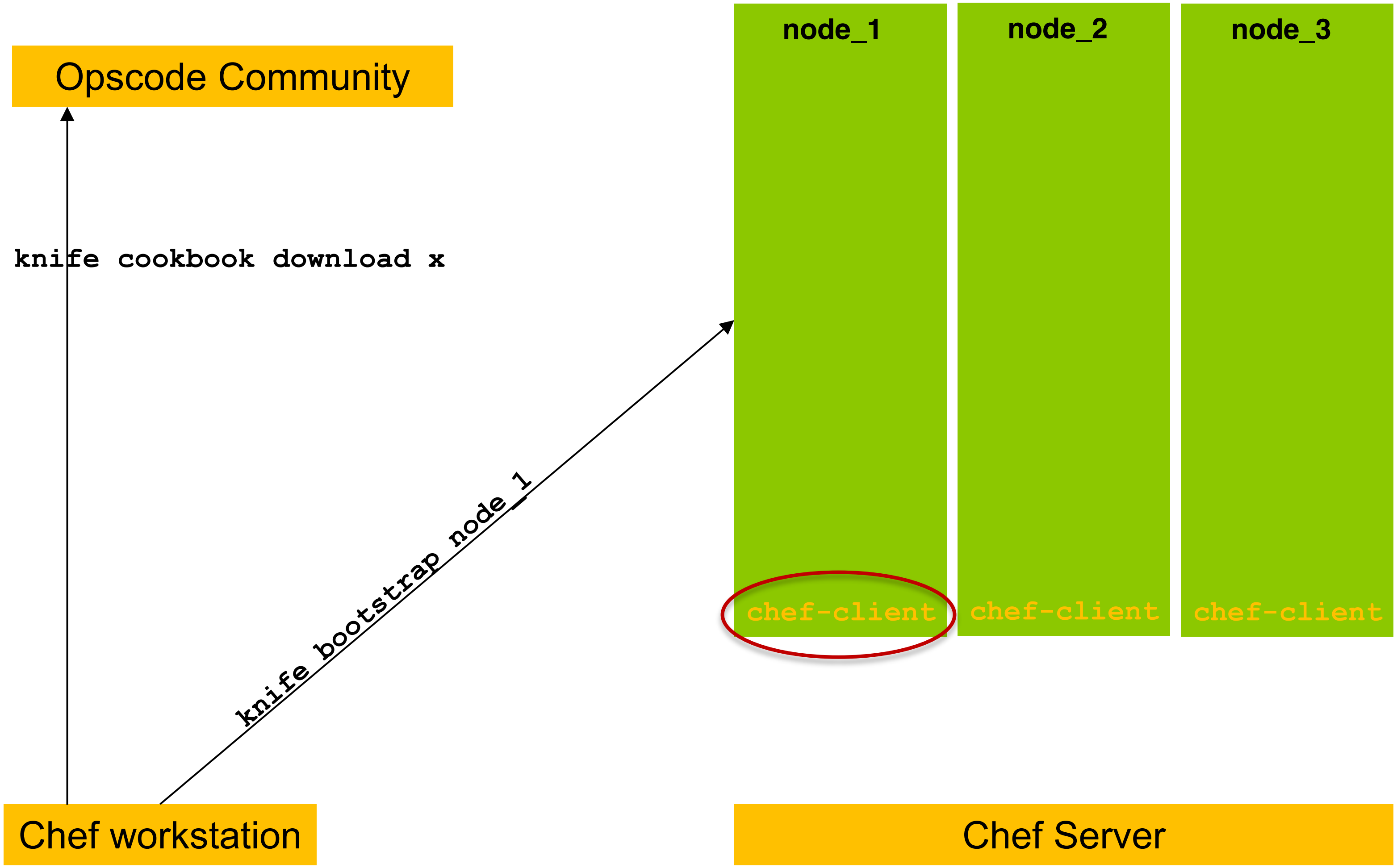
node_3

chef-client

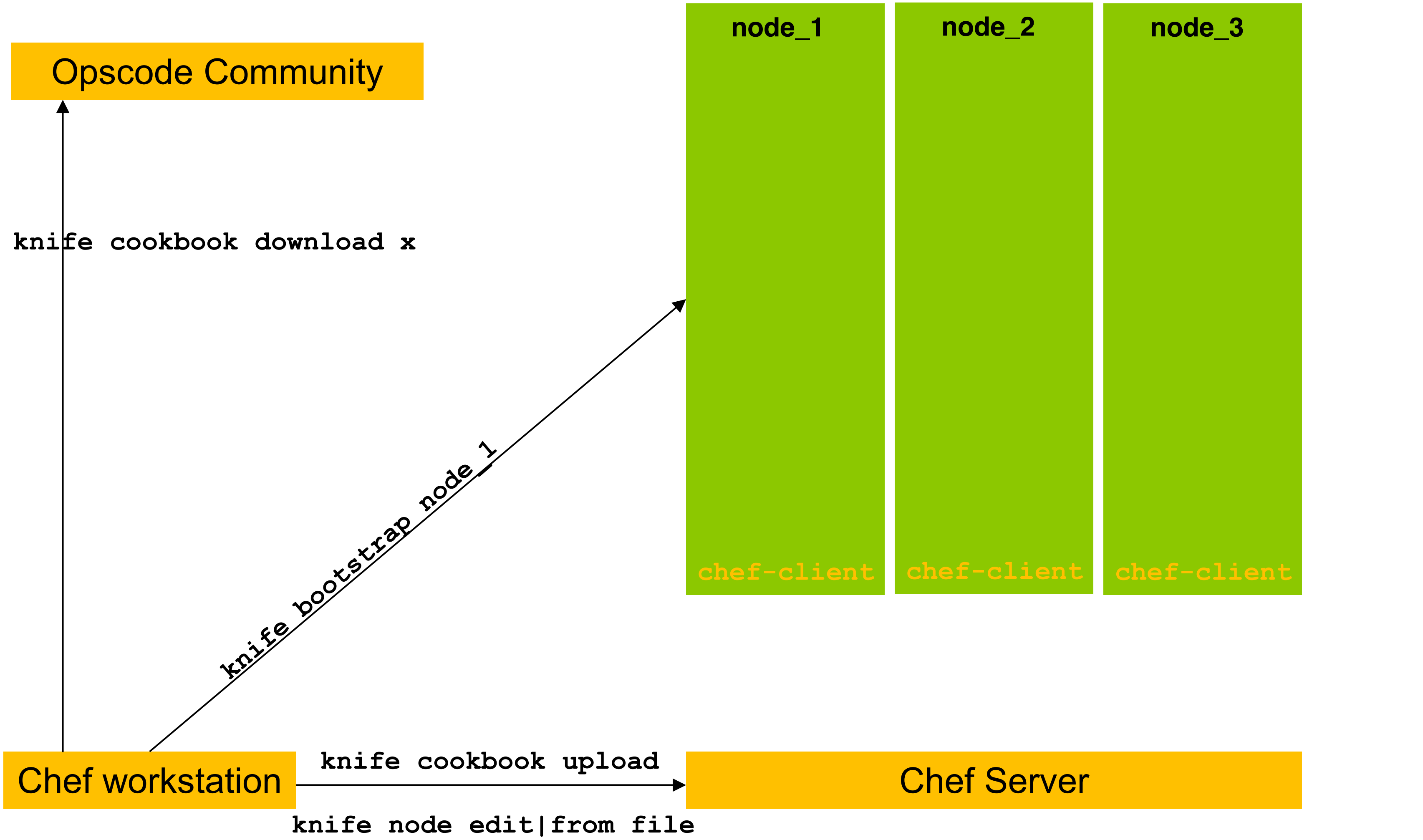
chef-client

Chef Server

Chef node setup



Chef node setup



Opscode Community

knife cookbook download x

node_1

node_2

node_3

chef-client

chef-client

chef-client

knife bootstrap node_1

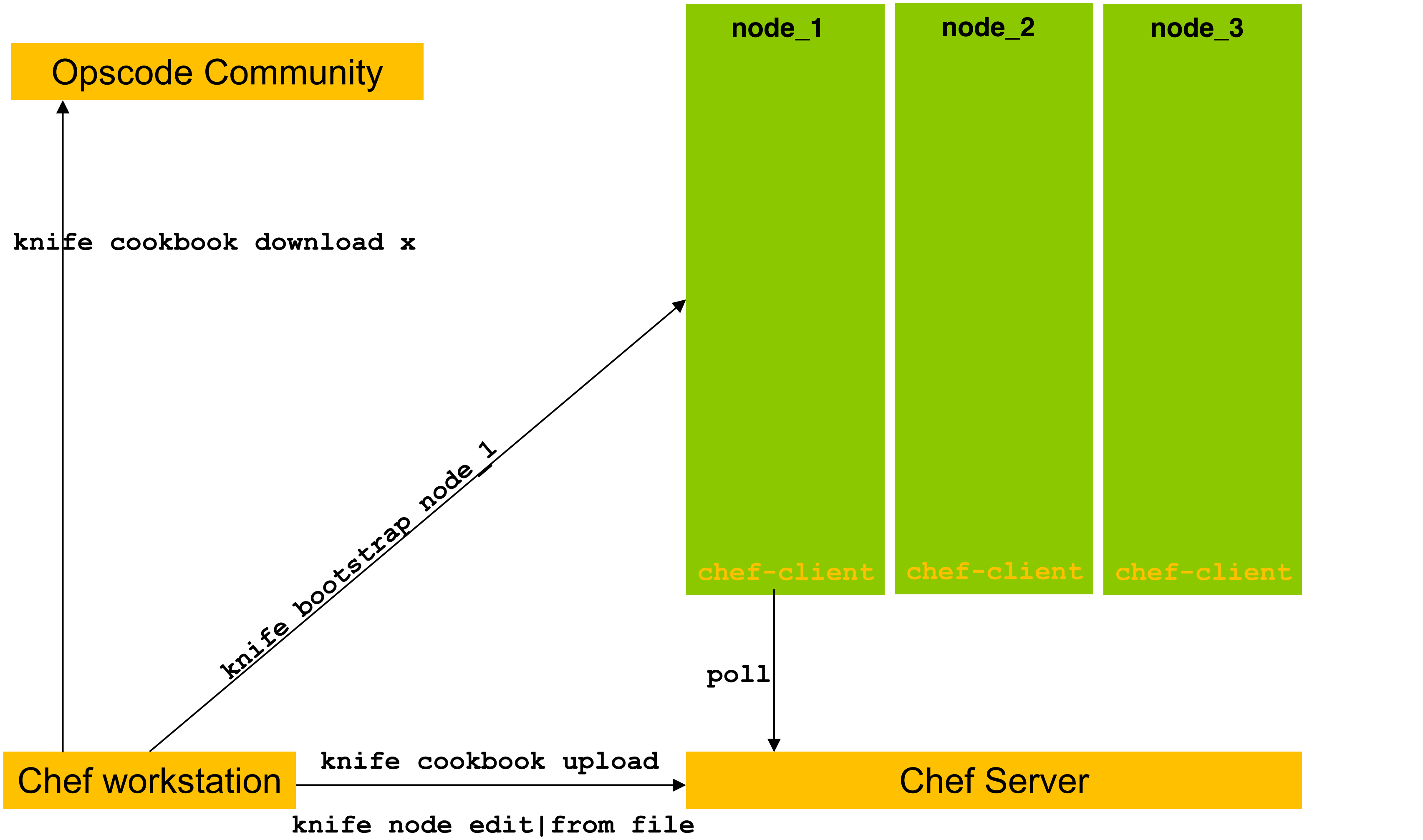
Chef workstation

knife cookbook upload

knife node edit|from file

Chef Server

Chef node setup



Opscode Community

node_1

node_2

node_3

chef-client

chef-client

chef-client

knife cookbook download x

knife bootstrap node_1

poll

Chef workstation

knife cookbook upload

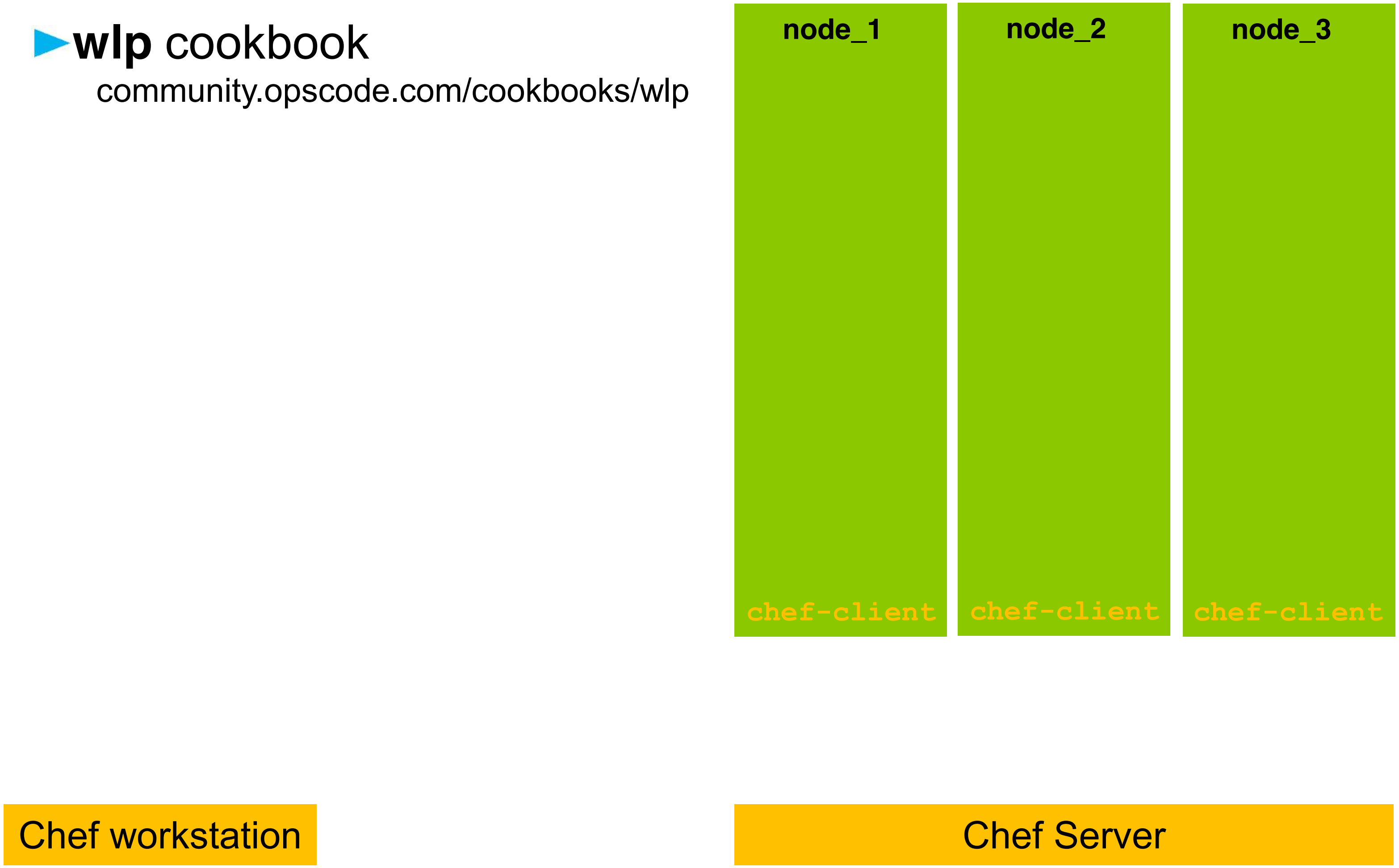
knife node edit|from file

Chef Server

Install & Create … cookbook “wlp”

Apache 2.0 license

- **wlp** cookbook
  community.opscode.com/cookbooks/wlp



Install & Create … cookbook “wlp”

Apache 2.0 license

- **wlp** cookbook
  community.opscode.com/cookbooks/wlp
- Install Liberty: archive|zip
  - Note: IBM Licensing

node_1

chef-client

node_2

chef-client

node_3

chef-client

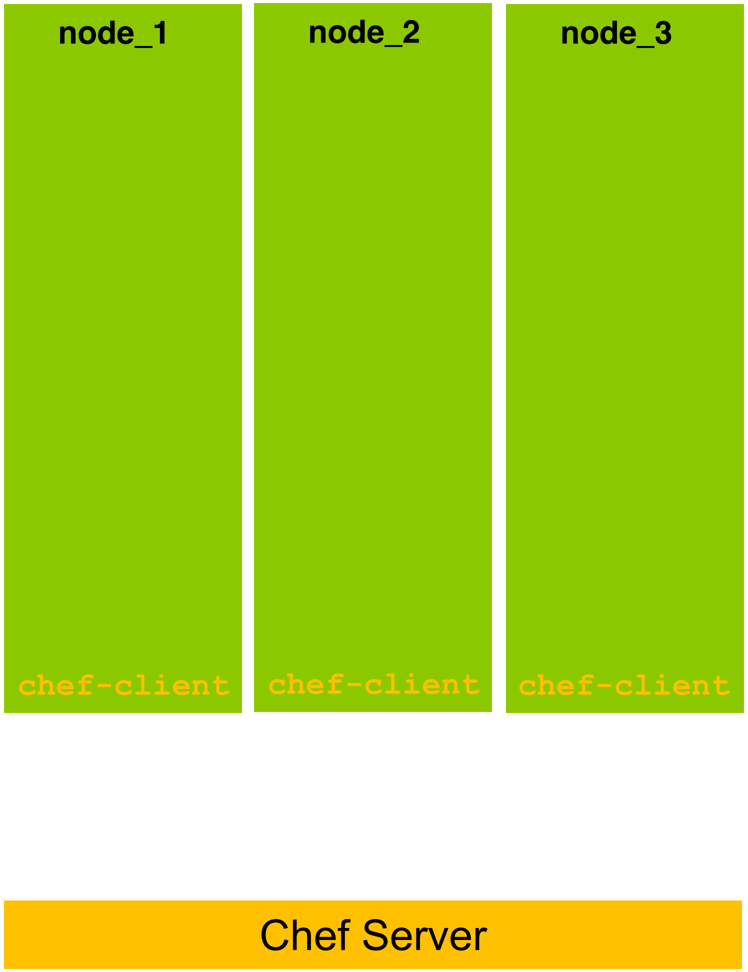
Chef workstation

Chef Server

Install & Create … cookbook "wlp"

Apache 2.0 license

- **wlp** cookbook
  community.opscode.com/cookbooks/wlp
- Install Liberty: archive|zip
  - Note: IBM Licensing
- Create server/s
  - Separate wlp/usr dir



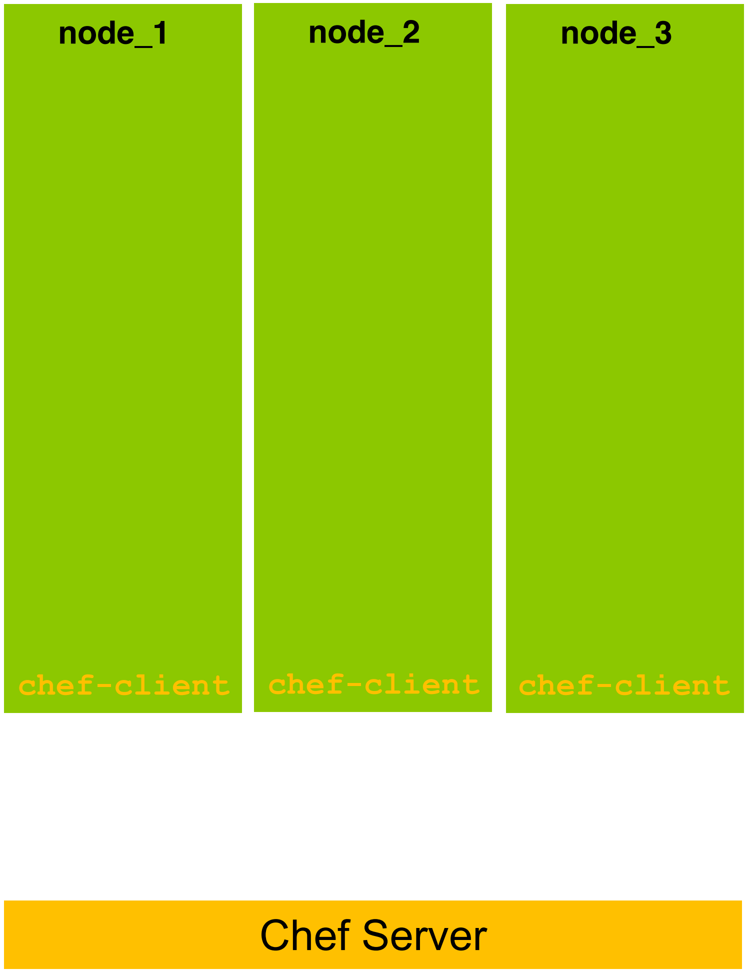
Chef workstation

Install & Create … cookbook “wlp”

Apache 2.0 license

- **wlp** cookbook

  community.opscode.com/cookbooks/wlp
- Install Liberty: archive|zip
  - Note: IBM Licensing
- Create server/s
  - Separate wlp/usr dir
- Set JVM options

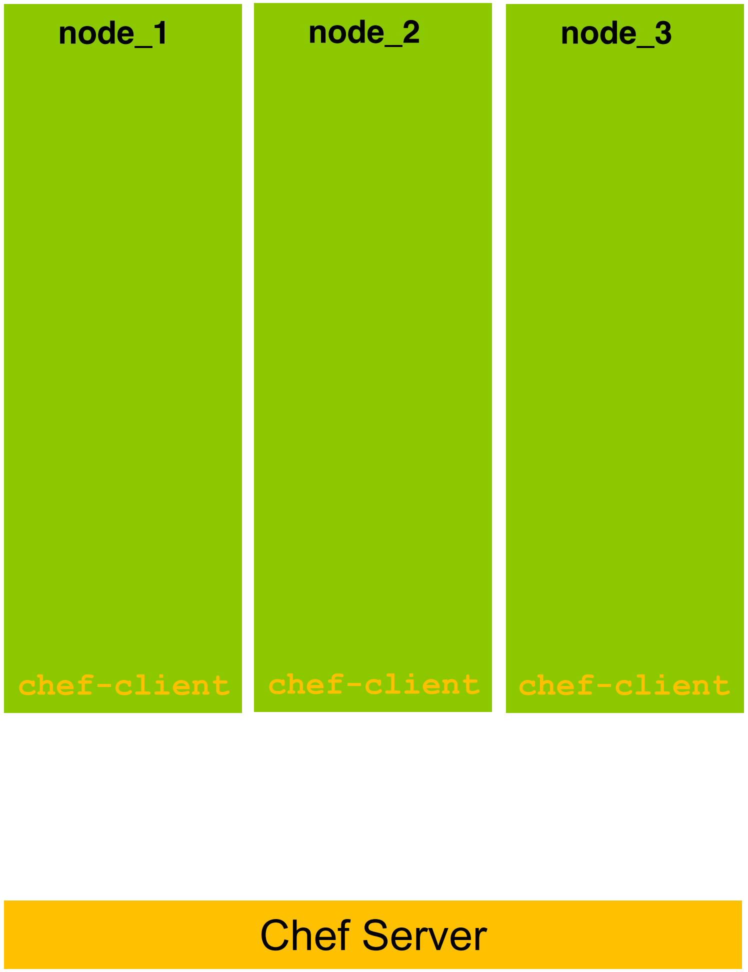


Chef workstation

Install & Create … cookbook "wlp"

Apache 2.0 license

- **wlp** cookbook
  community.opscode.com/cookbooks/wlp
- Install Liberty: archive|zip
  - Note: IBM Licensing
- Create server/s
  - Separate wlp/usr dir
- Set JVM options
- Manipulate server's live server.xml as Ruby hash



node_1

node_2

node_3

chef-client

chef-client

chef-client

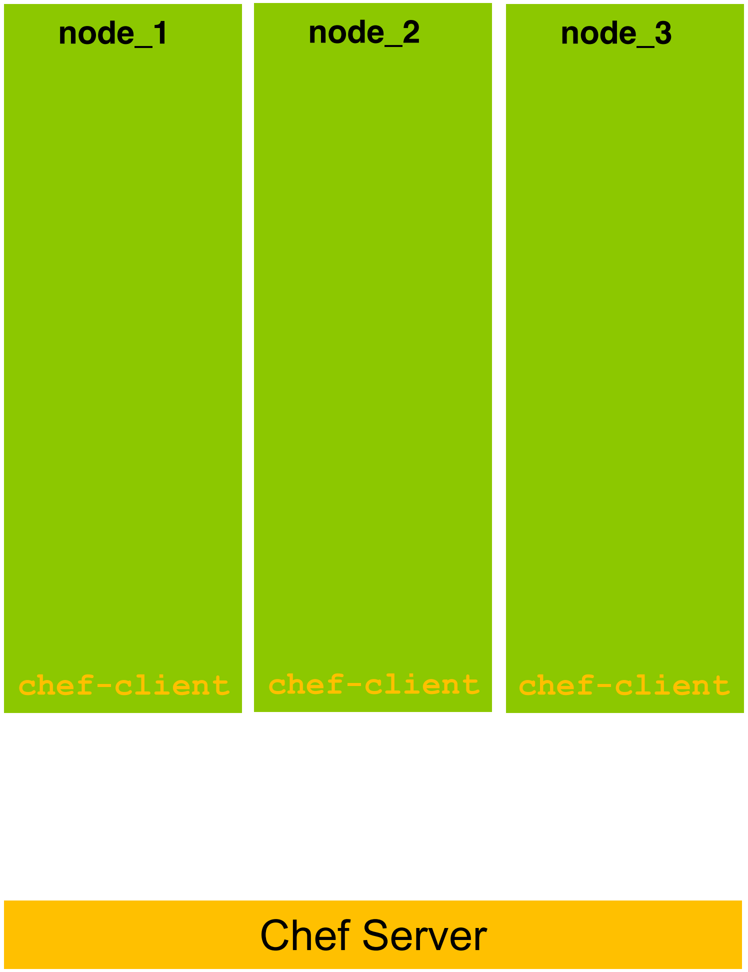
Chef Server

Chef workstation

Install & Create … cookbook “wlp”

Apache 2.0 license

- **wlp** cookbook
  community.opscode.com/cookbooks/wlp
- Install Liberty: archive|zip
  - Note: IBM Licensing
- Create server/s
  - Separate wlp/usr dir
- Set JVM options
- Manipulate server’s live server.xml as Ruby hash
- Create init.d service wlp-<server_name>
- …

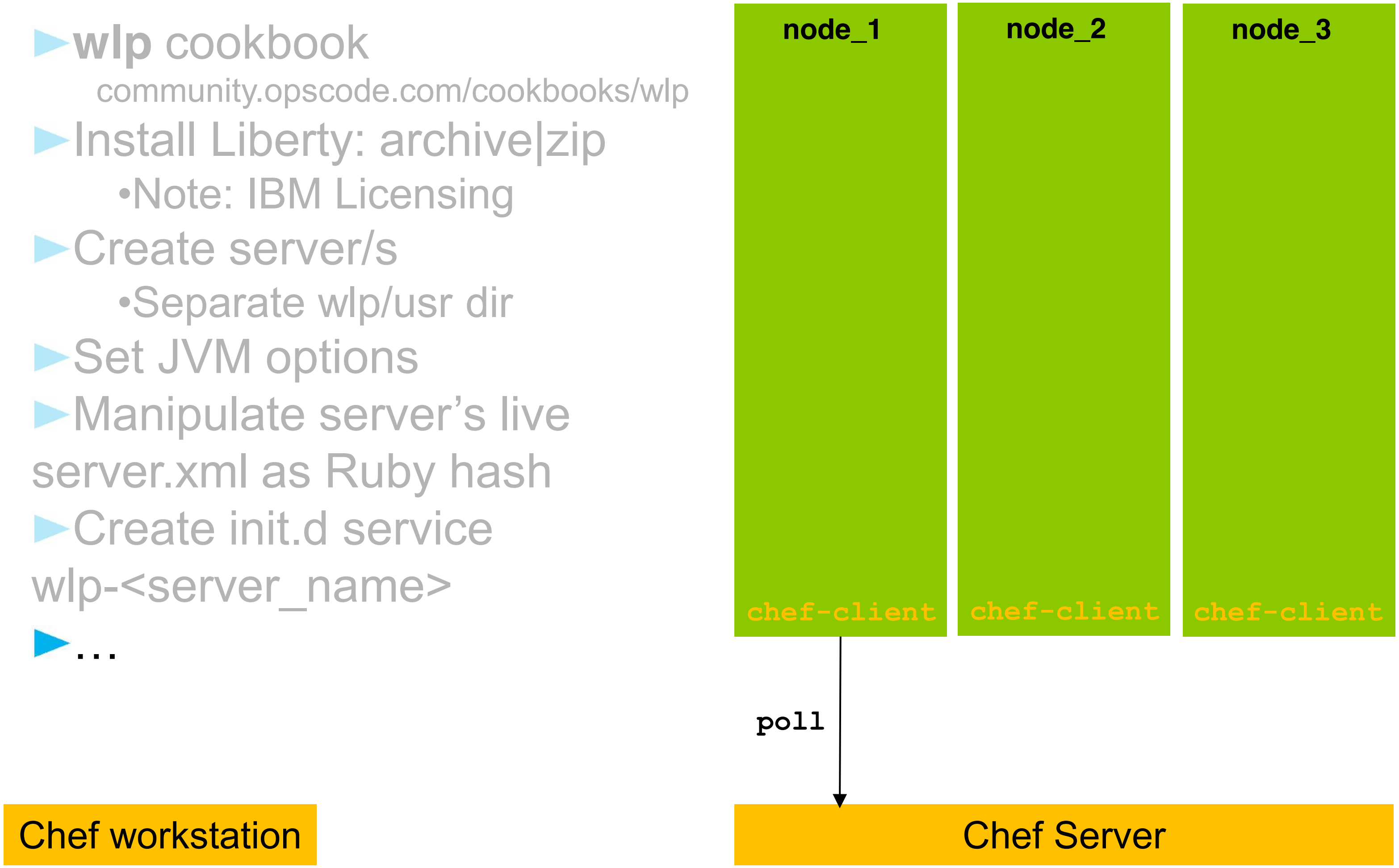


Chef workstation

Install & Create … cookbook “wlp”

Apache 2.0 license

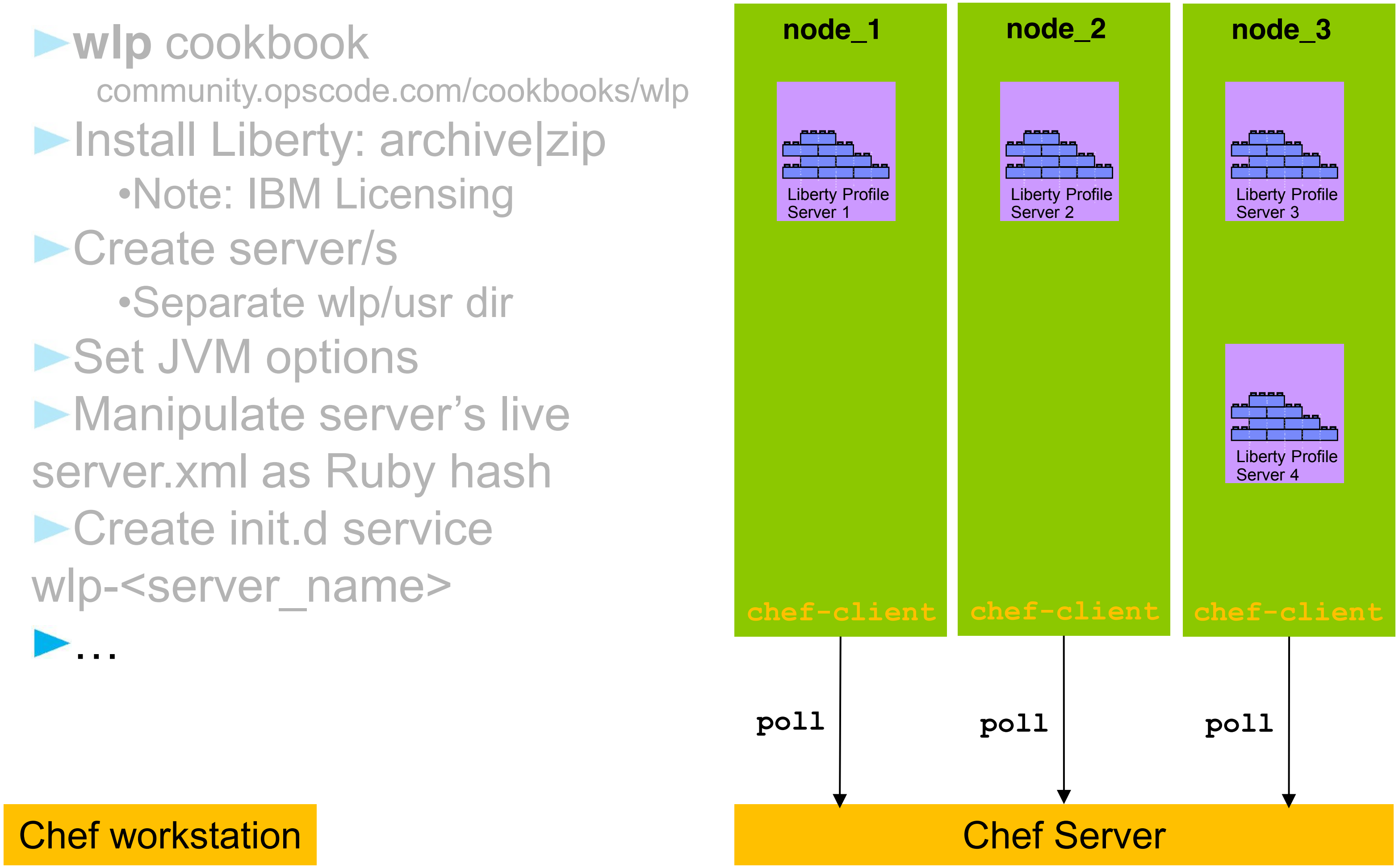
- **wlp** cookbook
  community.opscode.com/cookbooks/wlp
- Install Liberty: archive|zip
  - Note: IBM Licensing
- Create server/s
  - Separate wlp/usr dir
- Set JVM options
- Manipulate server’s live server.xml as Ruby hash
- Create init.d service wlp-<server_name>
- …



Install & Create … cookbook “wlp”

Apache 2.0 license

- **wlp** cookbook
  community.opscode.com/cookbooks/wlp
- Install Liberty: archive|zip
  - •Note: IBM Licensing
- Create server/s
  - •Separate wlp/usr dir
- Set JVM options
- Manipulate server’s live server.xml as Ruby hash
- Create init.d service wlp-<server_name>
- …

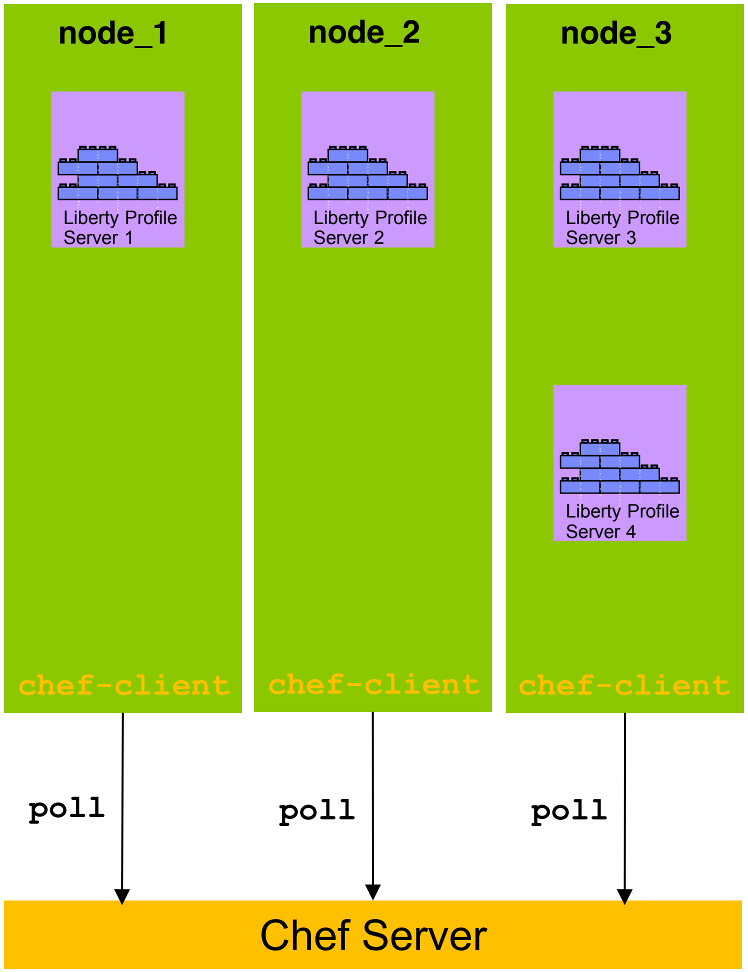


Install & Create … cookbook “wlp”

Apache 2.0 license

- **wlp** cookbook
  community.opscode.com/cookbooks/wlp
- Install Liberty: archive|zip
  - •Note: IBM Licensing
- Create server/s
  - •Separate wlp/usr dir
- Set JVM options
- Manipulate server’s live server.xml as Ruby hash
- Create init.d service wlp-<server_name>
- …

Chef workstation



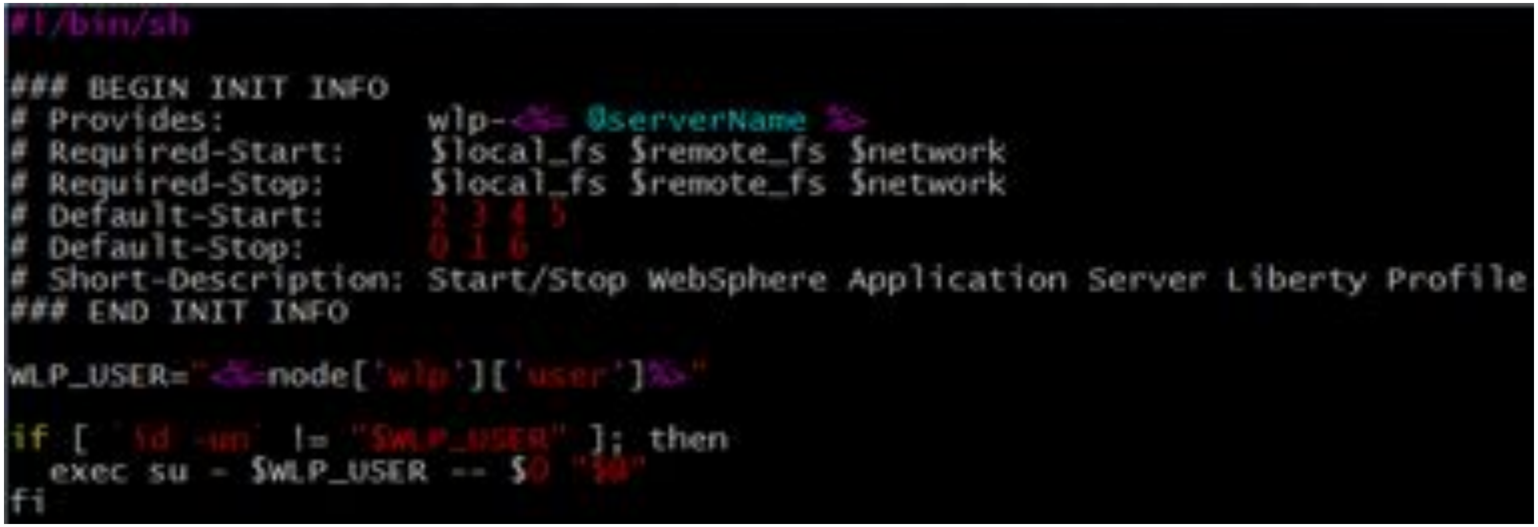
Templating config files

- Erebus Ruby Gem for sophisticated ‘variable evaluation’
- Embedded Ruby code

Templating config files

- Erebus Ruby Gem for sophisticated ‘variable evaluation’
- Embedded Ruby code

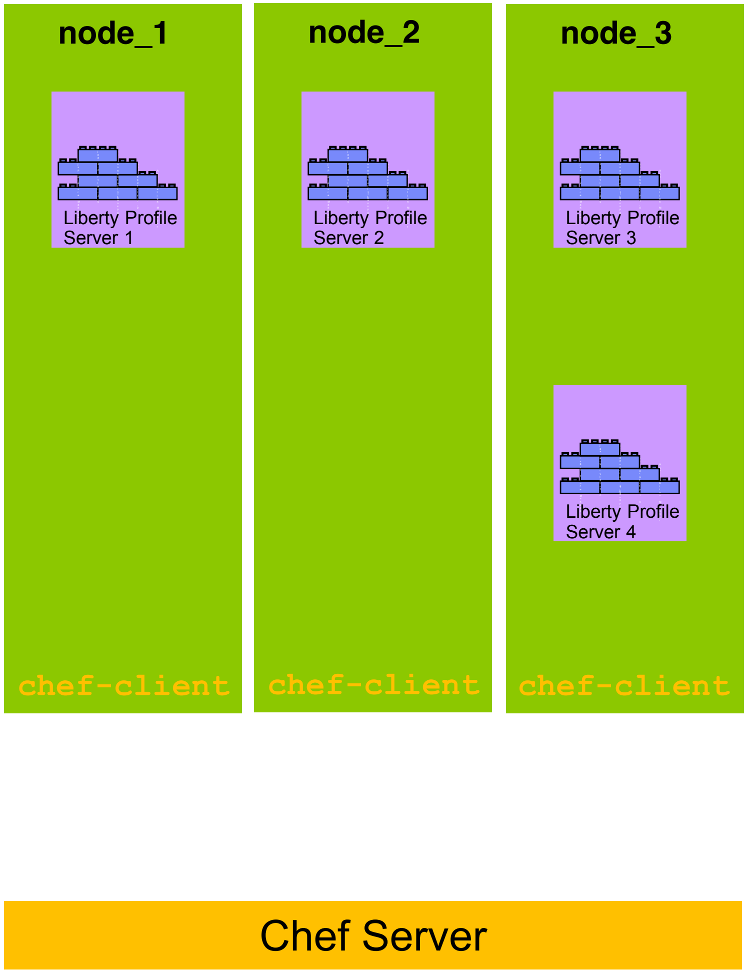
- Init.d service …



```
#!/bin/sh

### BEGIN INIT INFO
# Provides:          wlp-<%= @serverName %>
# Required-Start:    $local_fs $remote_fs $network
# Required-Stop:     $local_fs $remote_fs $network
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start/Stop WebSphere Application Server Liberty Profile
### END INIT INFO

WLP_USER="<%=node['wlp']['user']%>"

if [ `id -un` != "$WLP_USER" ]; then
  exec su - $WLP_USER -- $0 "$@"
fi
```

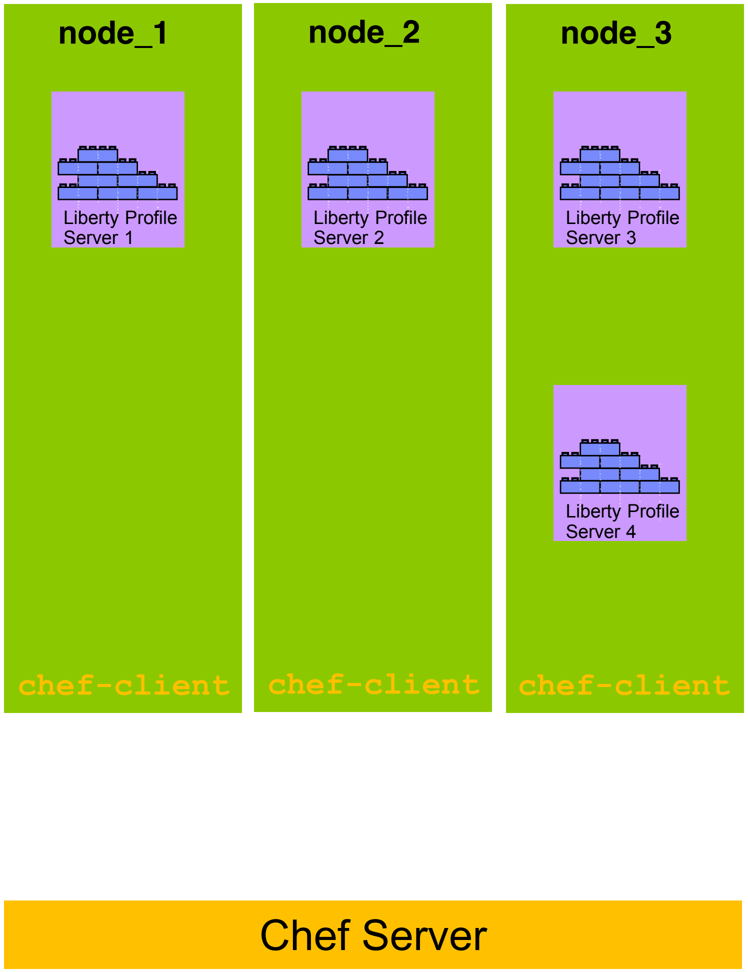
Deploy application … cookbook “application_wlp”

- **application_wlp** cookbook
  community.opscode.com/cookbooks/application_wlp



Deploy application … cookbook “application_wlp”

- **application_wlp** cookbook
  community.opscode.com/cookbooks/application_wlp
- Extends ‘application’
  - •Template Method pattern



node_1

Liberty Profile Server 1

chef-client

node_2

Liberty Profile Server 2

chef-client

node_3

Liberty Profile Server 3

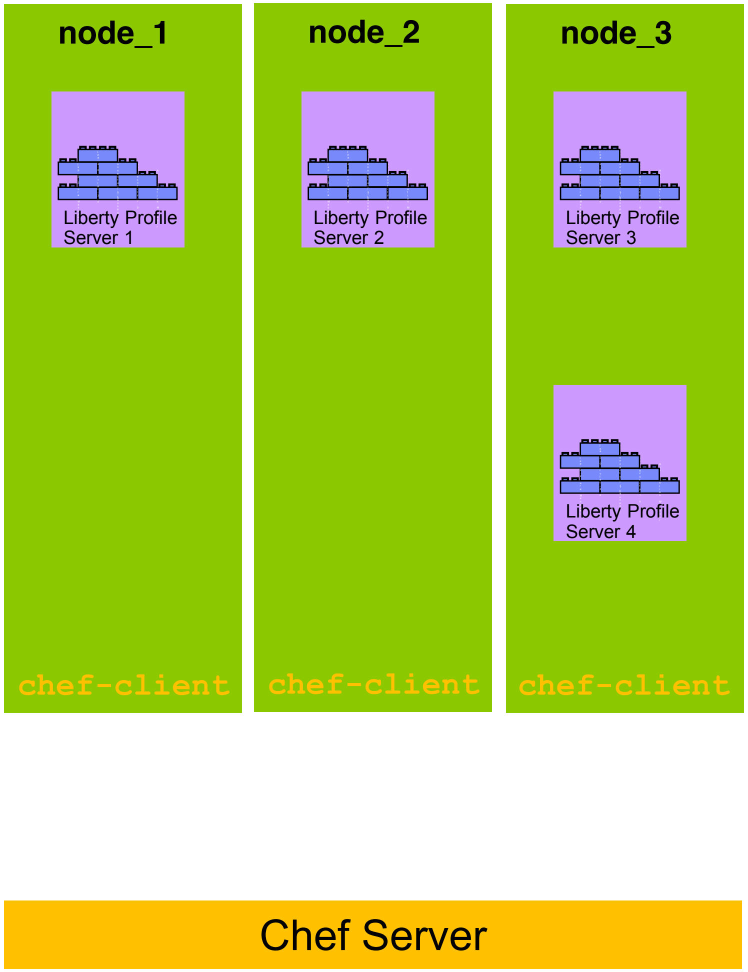
Liberty Profile Server 4

chef-client

Chef Server

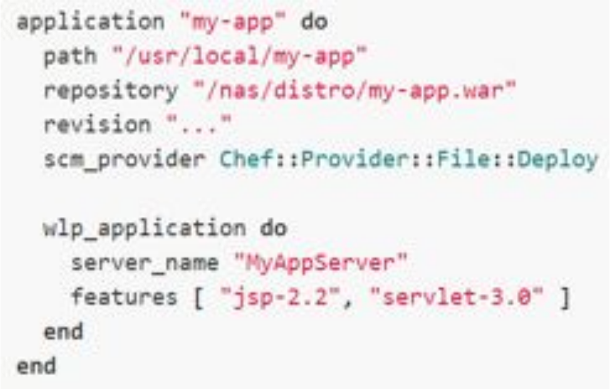
Deploy application … cookbook “application_wlp”

- **application_wlp** cookbook
  community.opscode.com/cookbooks/application_wlp
- Extends ‘application’
  - Template Method pattern
- Deploy application of any type
  .ear | .war | .eba



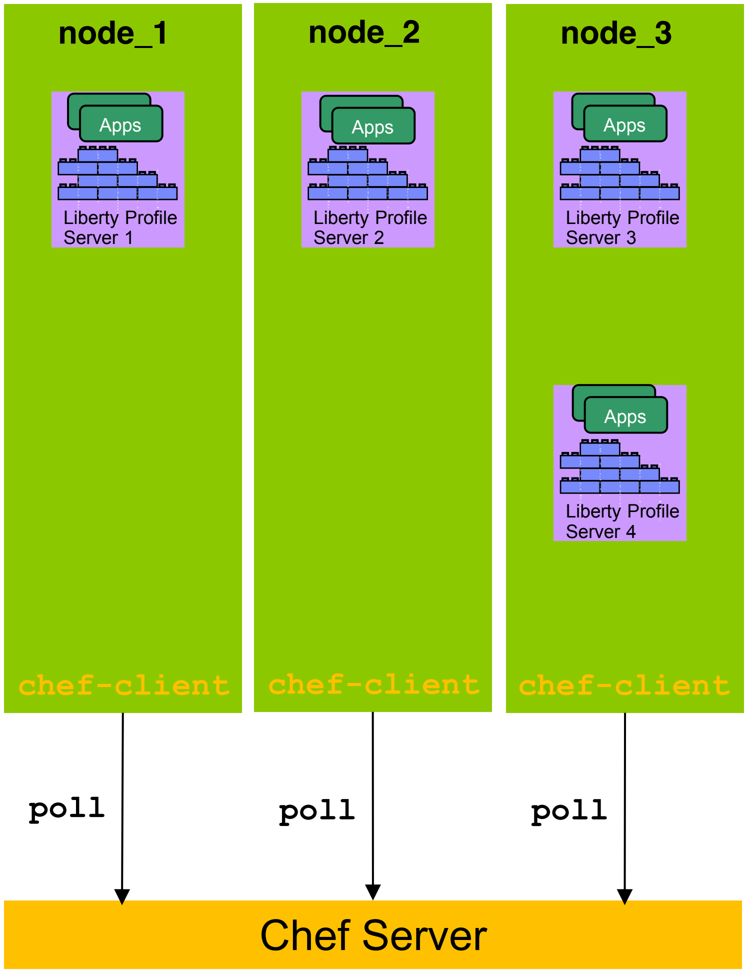
Deploy application … cookbook “application_wlp”

- **application_wlp** cookbook
  community.opscode.com/cookbooks/application_wlp
- Extends ‘application’
  - Template Method pattern
- Deploy application of any type
  .ear | .war | .eba



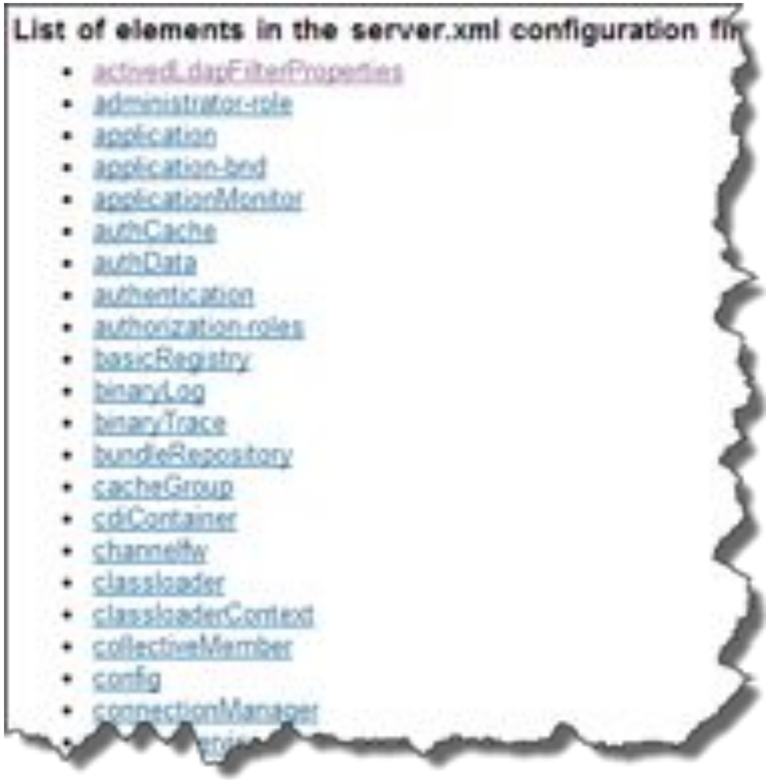
```
application "my-app" do
  path "/usr/local/my-app"
  repository "/nas/distro/my-app.war"
  revision "..."
  scm_provider Chef::Provider::File::Deploy

  wlp_application do
    server_name "MyAppServer"
    features [ "jsp-2.2", "servlet-3.0" ]
  end
end
```

Chef workstation



Modelling server.xml … and `<include>`s

- Simple, minimal, but rich (!!)
  - 94 different element types
- Represent server config in code



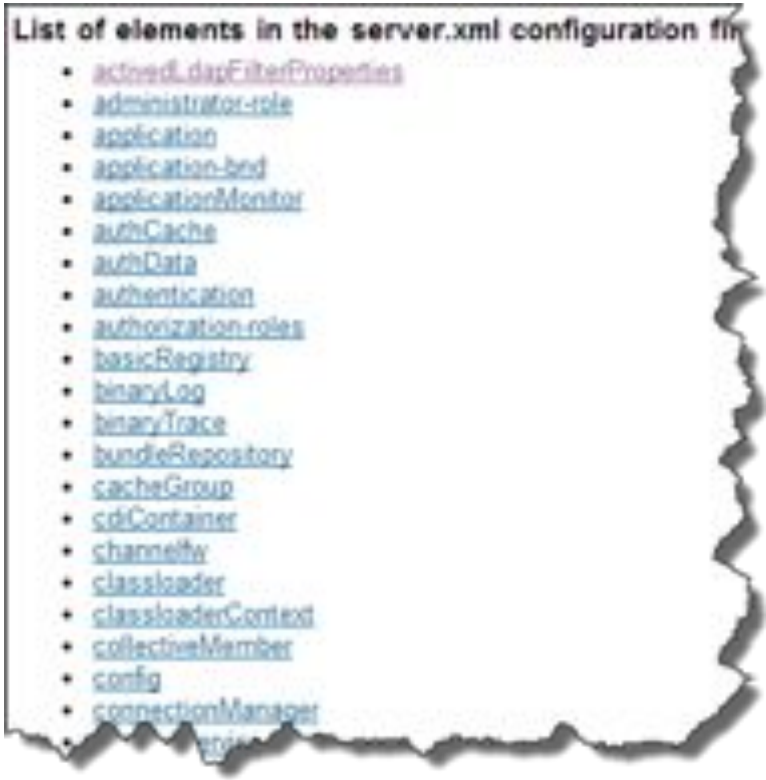
List of elements in the server.xml configuration fi

- activedLdapFilterProperties
- administrator-role
- application
- application-bnd
- applicationMonitor
- authCache
- authData
- authentication
- authorization-roles
- basicRegistry
- binaryLog
- binaryTrace
- bundleRepository
- cacheGroup
- cdiContainer
- channelfw
- classloader
- classloaderContext
- collectiveMember
- config
- connectionManager

Modelling server.xml … and `<include>`s

- Simple, minimal, but rich (!!)
  - 94 different element types
- Represent server config in code

1. Manage raw server.xml
   - Low level control



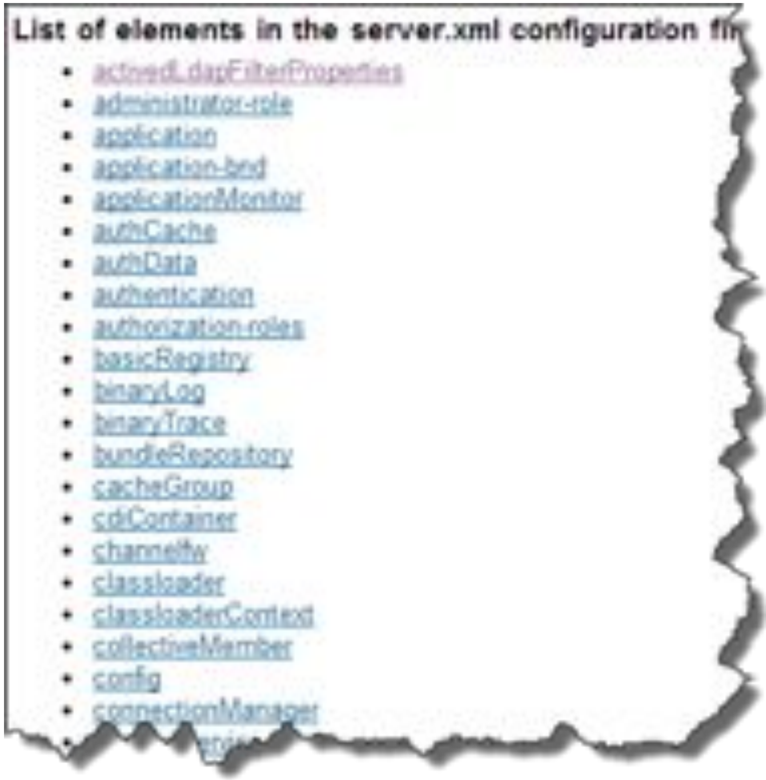
List of elements in the server.xml configuration fi

- activedLdapFilterProperties
- administrator-role
- application
- application-bnd
- applicationMonitor
- authCache
- authData
- authentication
- authorization-roles
- basicRegistry
- binaryLog
- binaryTrace
- bundleRepository
- cacheGroup
- cdiContainer
- channelfw
- classloader
- classloaderContext
- collectiveMember
- config
- connectionManager

Modelling server.xml … and `<include>`s

- Simple, minimal, but rich (!!)
    - 94 different element types
- Represent server config in code

1. Manage raw server.xml
    - Low level control
2. server.xml.erb template?
    - Model the xml as a template into which other configuration / properties is inserted
    - Inserts can be variables or Ruby code
    - Just enough element types that you need



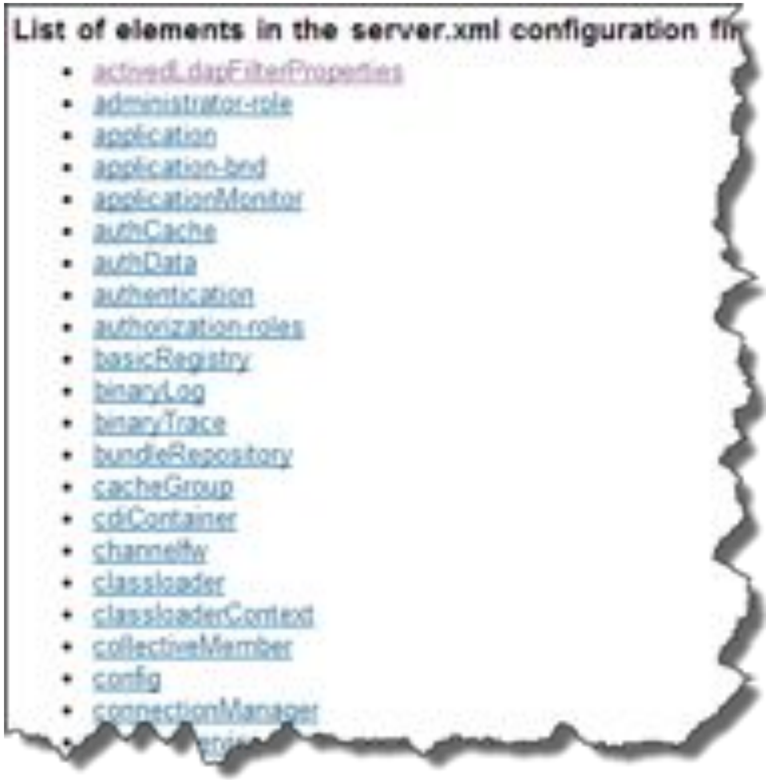
List of elements in the server.xml configuration fi

- activedLdapFilterProperties
- administrator-role
- application
- application-bnd
- applicationMonitor
- authCache
- authData
- authentication
- authorization-roles
- basicRegistry
- binaryLog
- binaryTrace
- bundleRepository
- cacheGroup
- cdiContainer
- channelfw
- classloader
- classloaderContext
- collectiveMember
- config
- connectionManager

Modelling server.xml … and <include>s

- Simple, minimal, but rich (!!)
  - 94 different element types
- Represent server config in code

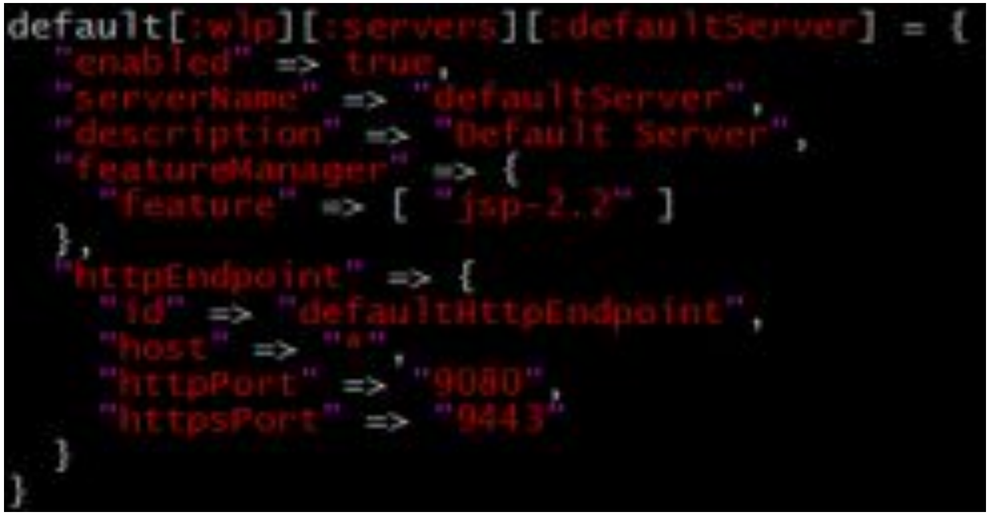
1. Manage raw server.xml
   - Low level control
2. server.xml.erb template?
   - Model the xml as a template into which other configuration / properties is inserted
   - Inserts can be variables or Ruby code
   - Just enough element types that you need
3. Simple, good enough:
   - Manipulate config in Ruby
   - Server config hash -> server.xml



List of elements in the server.xml configuration fi

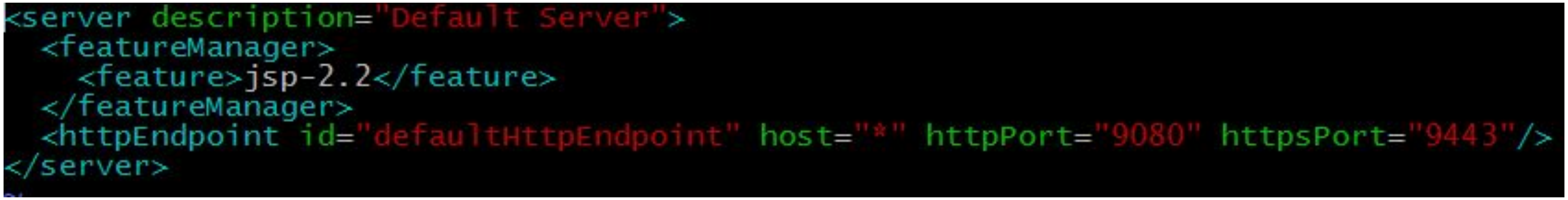
- activedLdapFilterProperties
- administrator-role
- application
- application-bnd
- applicationMonitor
- authCache
- authData
- authentication
- authorization-roles
- basicRegistry
- binaryLog
- binaryTrace
- bundleRepository
- cacheGroup
- cdiContainer
- channelfw
- classloader
- classloaderContext
- collectiveMember
- config
- connectionManager

Hash to XML



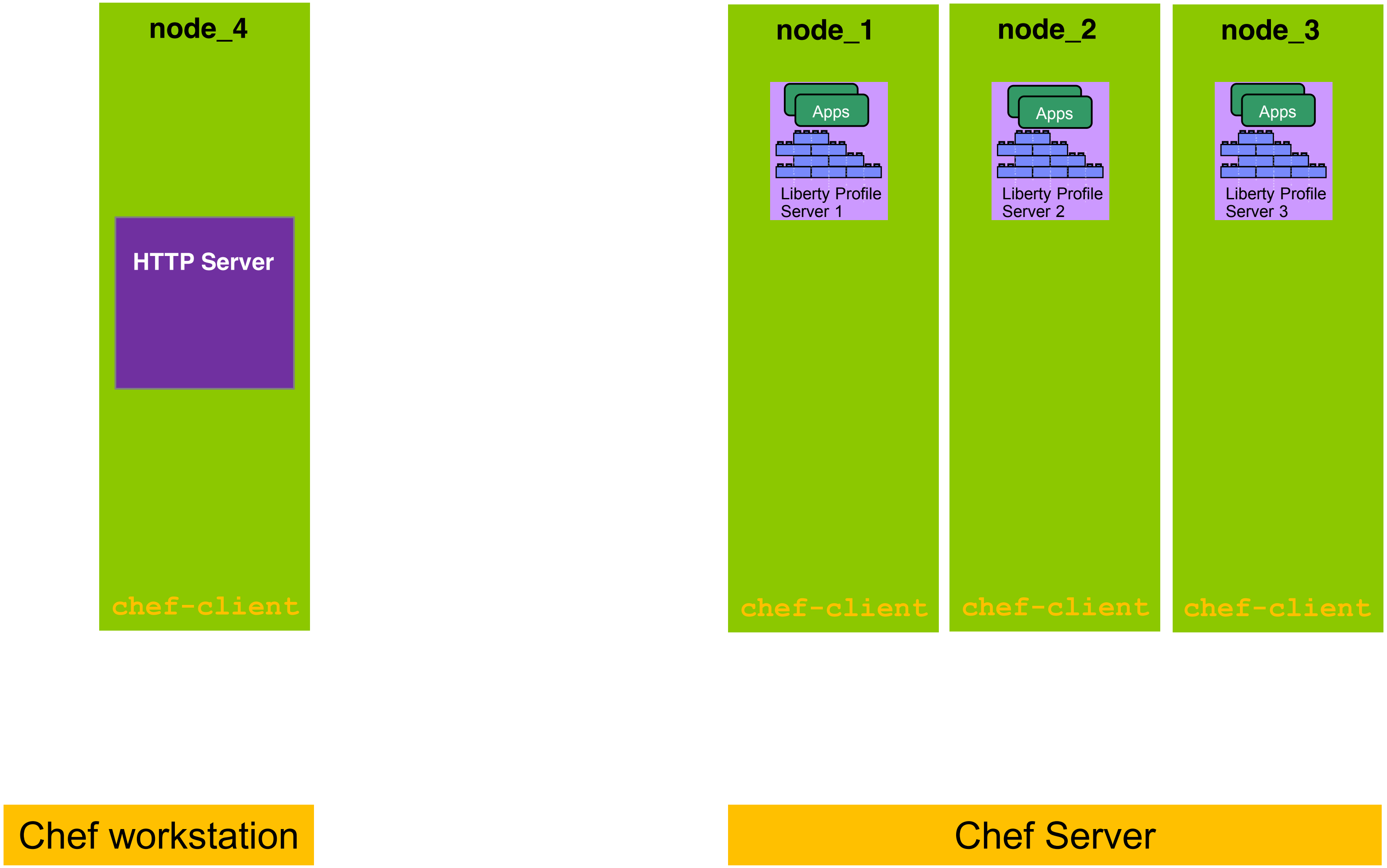
```
default[:wlp][:servers][:defaultServer] = {
  "enabled" => true,
  "serverName" => "defaultServer",
  "description" => "Default Server",
  "featureManager" => {
    "feature" => [ "jsp-2.2" ]
  },
  "httpEndpoint" => {
    "id" => "defaultHttpEndpoint",
    "host" => "*",
    "httpPort" => "9080",
    "httpsPort" => "9443"
  }
}
```

Converted by wlp cookbook



```
<server description="Default Server">
  <featureManager>
    <feature>jsp-2.2</feature>
  </featureManager>
  <httpEndpoint id="defaultHttpEndpoint" host="*" httpPort="9080" httpsPort="9443"/>
</server>
```

Load balancing … add in “apache2” cookbook



node_4

HTTP Server

chef-client

node_1

Apps

Liberty Profile Server 1

chef-client

node_2

Apps

Liberty Profile Server 2

chef-client

node_3

Apps

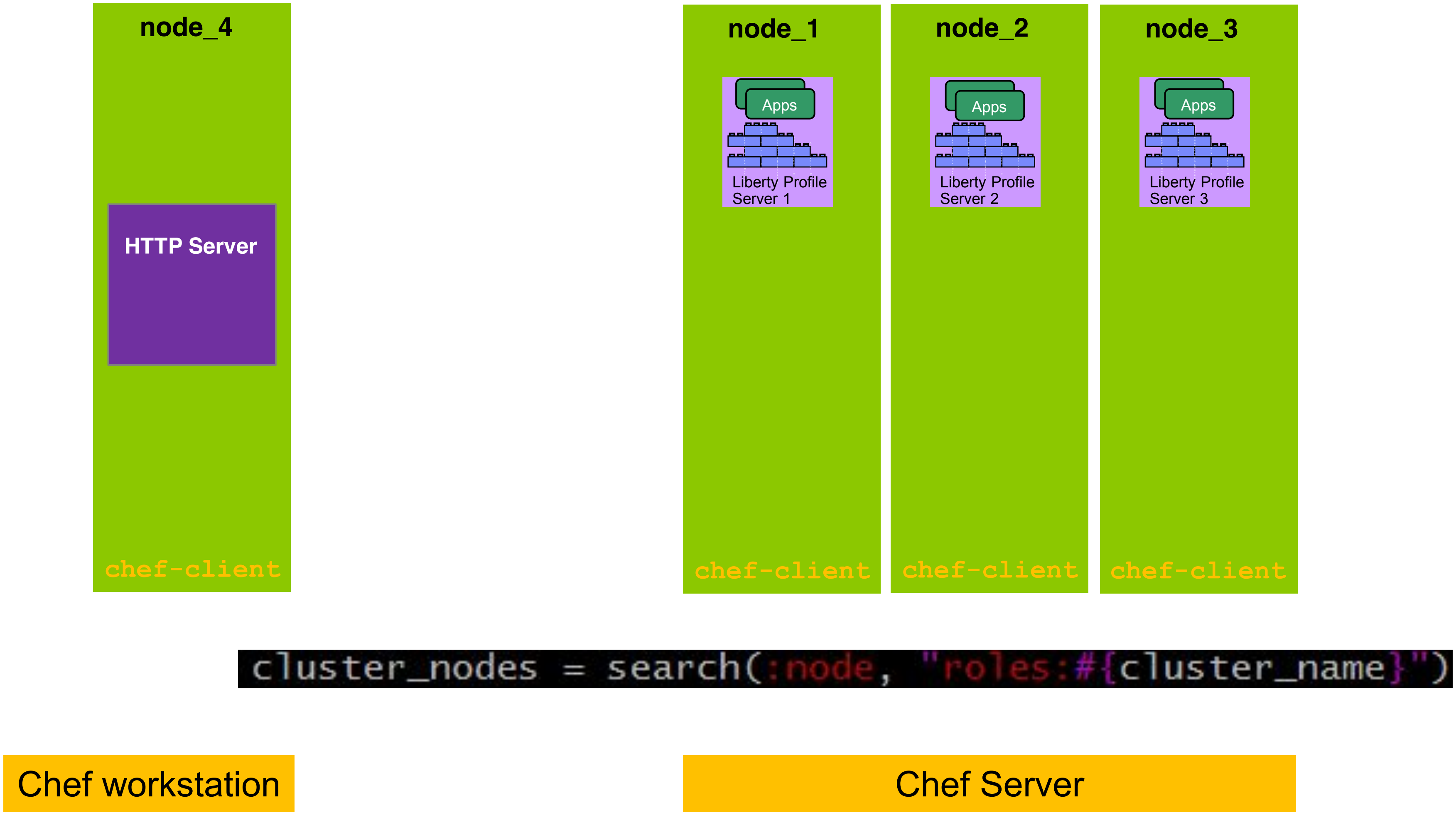
Liberty Profile Server 3

chef-client

Chef workstation

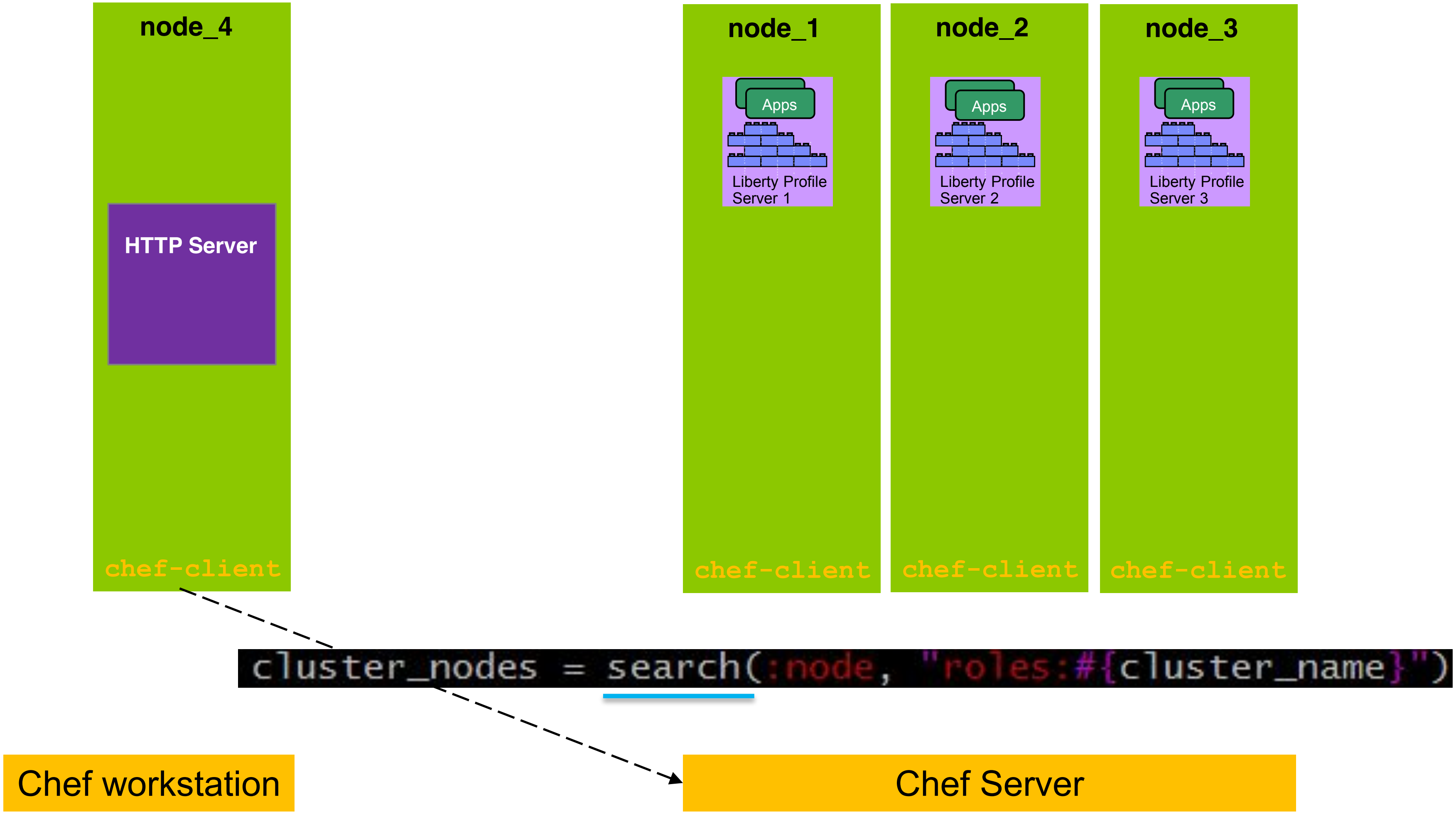
Chef Server

Load balancing … search for nodes in cluster role



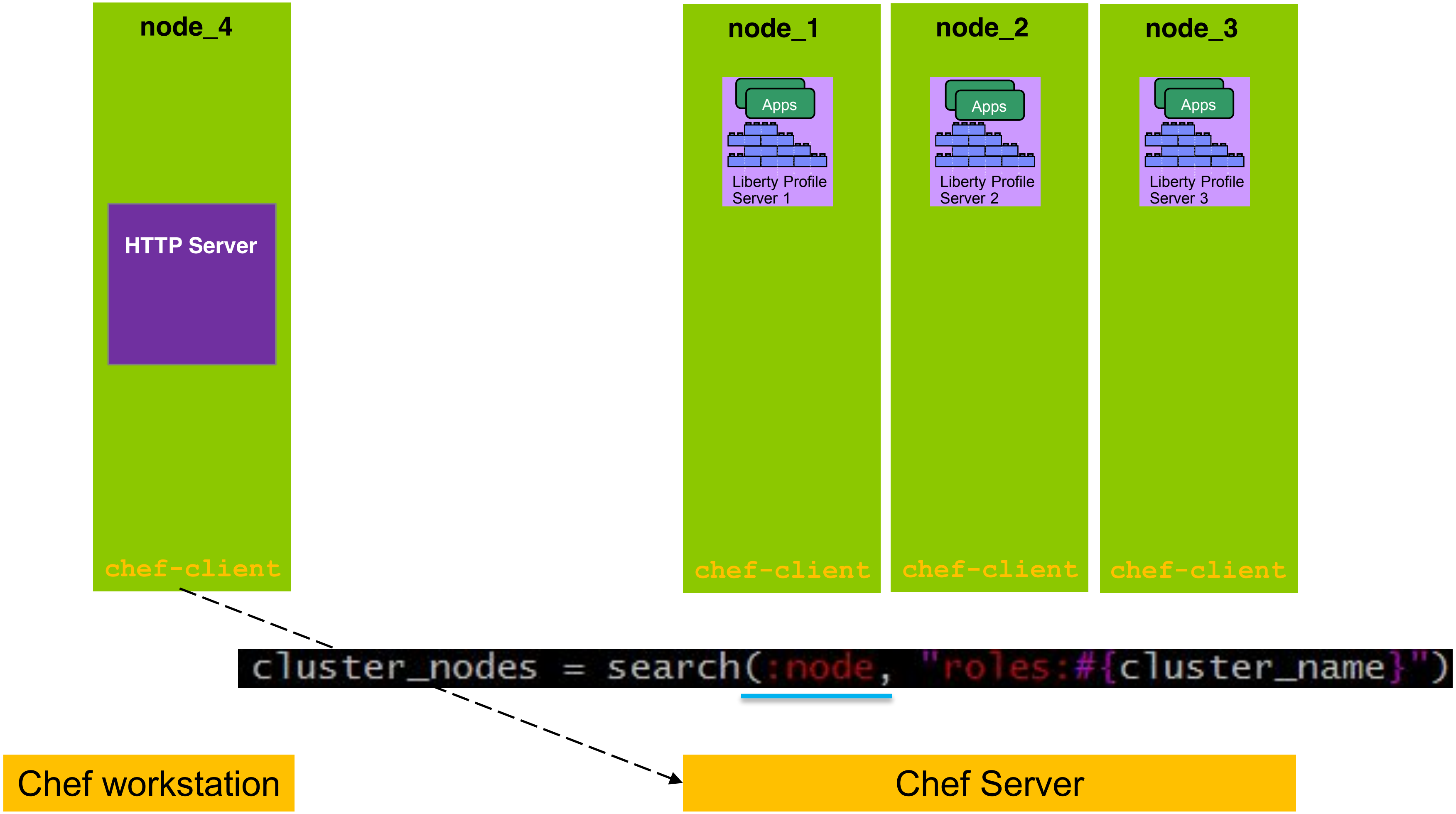
```
cluster_nodes = search(:node, "roles:#{cluster_name}")
```

Load balancing … search for nodes in cluster role



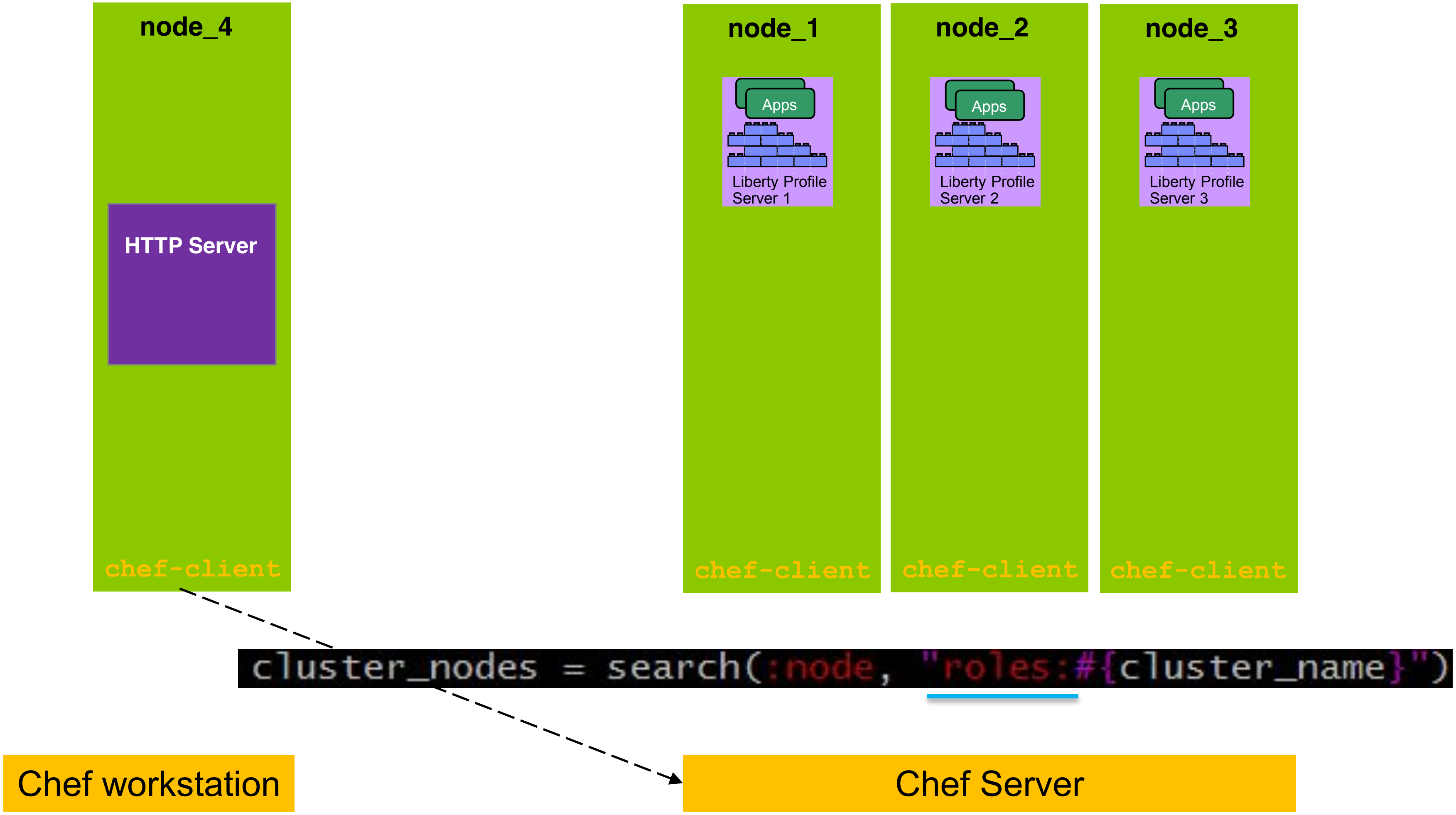
```
cluster_nodes = search(:node, "roles:#{cluster_name}")
```

Load balancing … search for nodes in cluster role



```
cluster_nodes = search(:node, "roles:#{cluster_name}")
```

Load balancing … search for nodes in cluster role



node_4

HTTP Server

chef-client

node_1

Apps

Liberty Profile Server 1

chef-client

node_2

Apps

Liberty Profile Server 2

chef-client

node_3

Apps

Liberty Profile Server 3

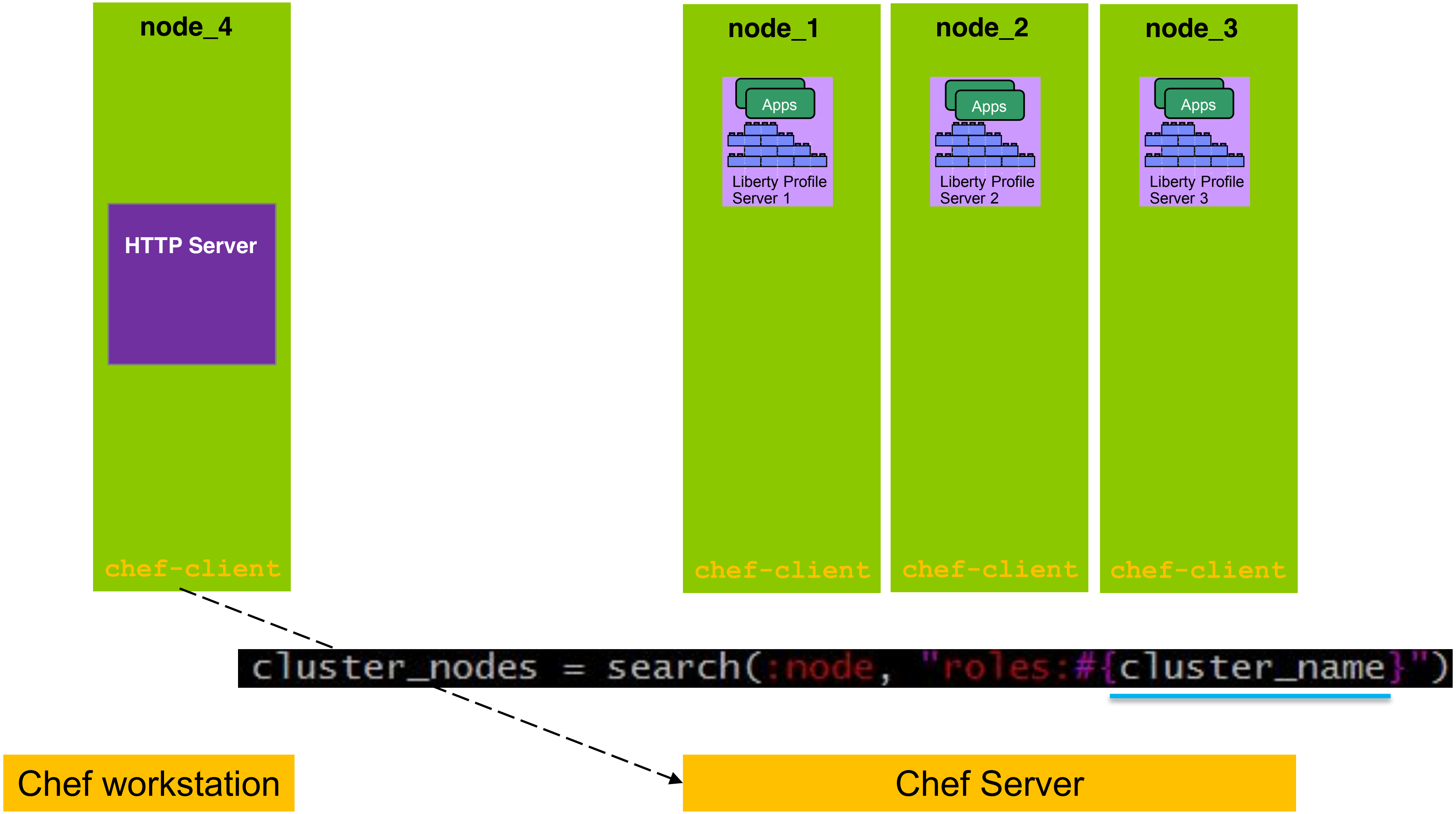
chef-client

```
cluster_nodes = search(:node, "roles:#{cluster_name}")
```

Chef workstation

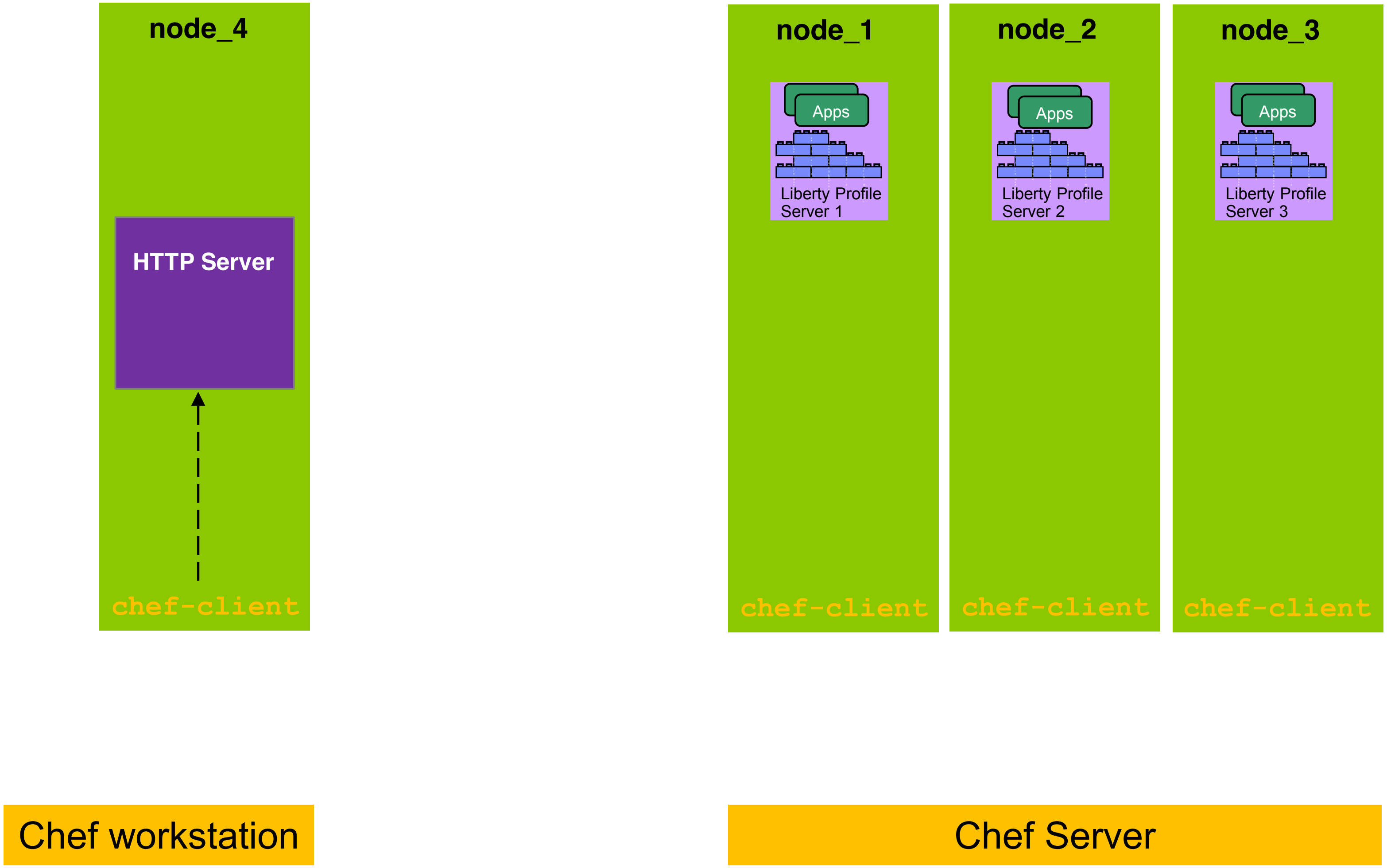
Chef Server

Load balancing … search for nodes in cluster role



```
cluster_nodes = search(:node, "roles:#{cluster_name}")
```

Load balancing … configure web server



node_4

HTTP Server

chef-client

node_1

Apps

Liberty Profile Server 1

chef-client

node_2

Apps

Liberty Profile Server 2

chef-client

node_3

Apps

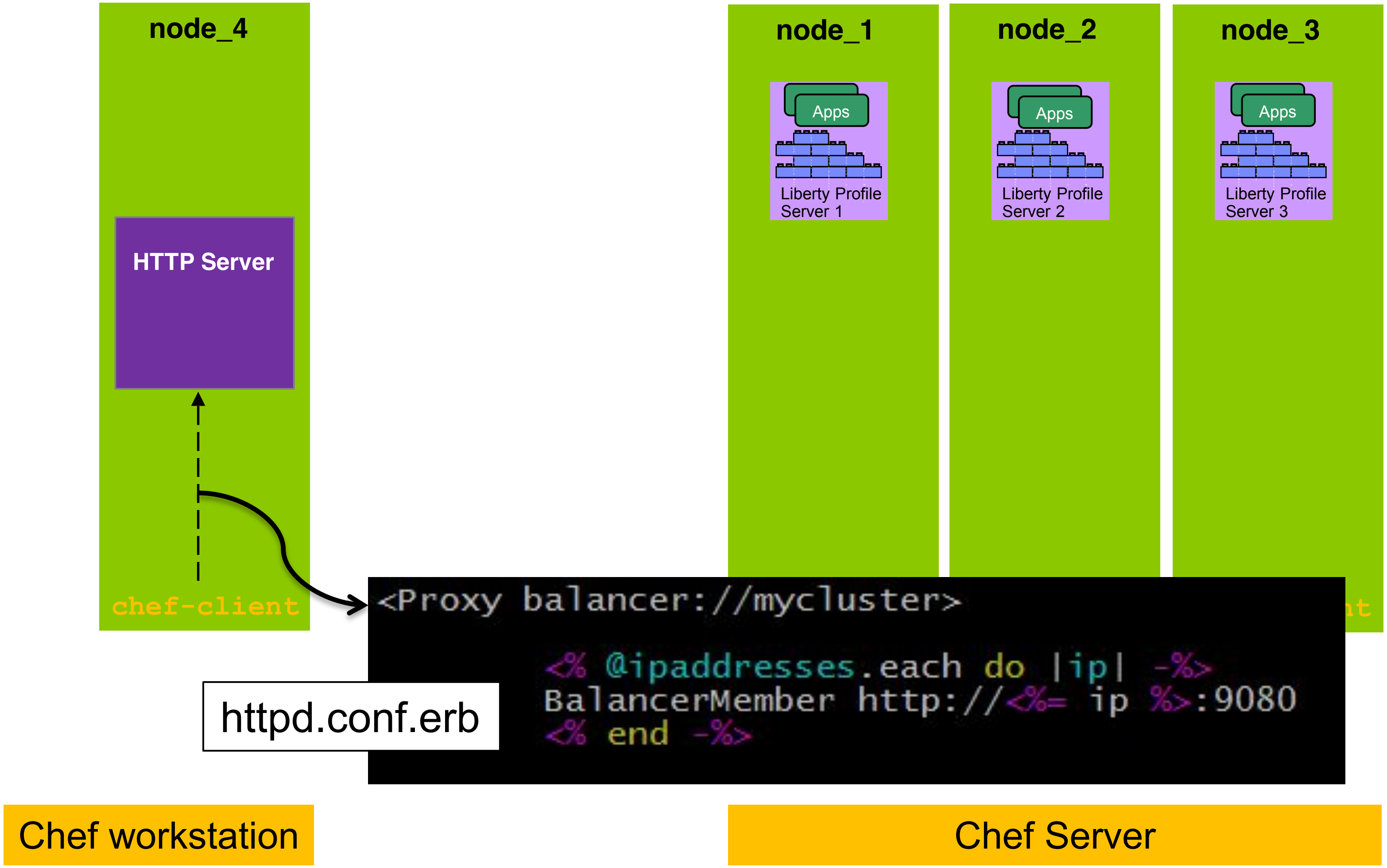
Liberty Profile Server 3

chef-client

Chef workstation

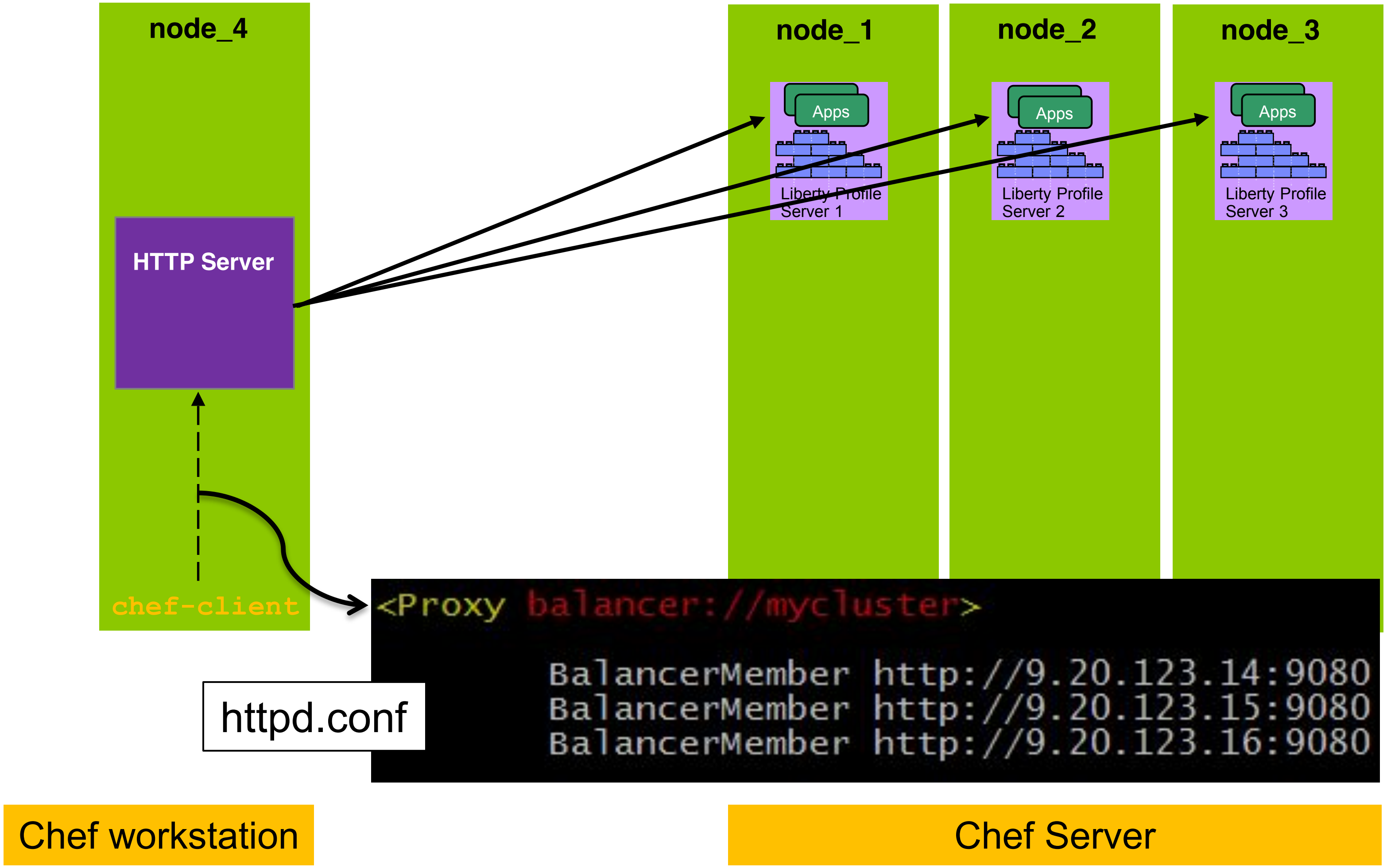
Chef Server

Load balancing … configure web server



```
<Proxy balancer://mycluster>

      <% @ipaddresses.each do |ip| -%>
      BalancerMember http://<%= ip %>:9080
      <% end -%>
```

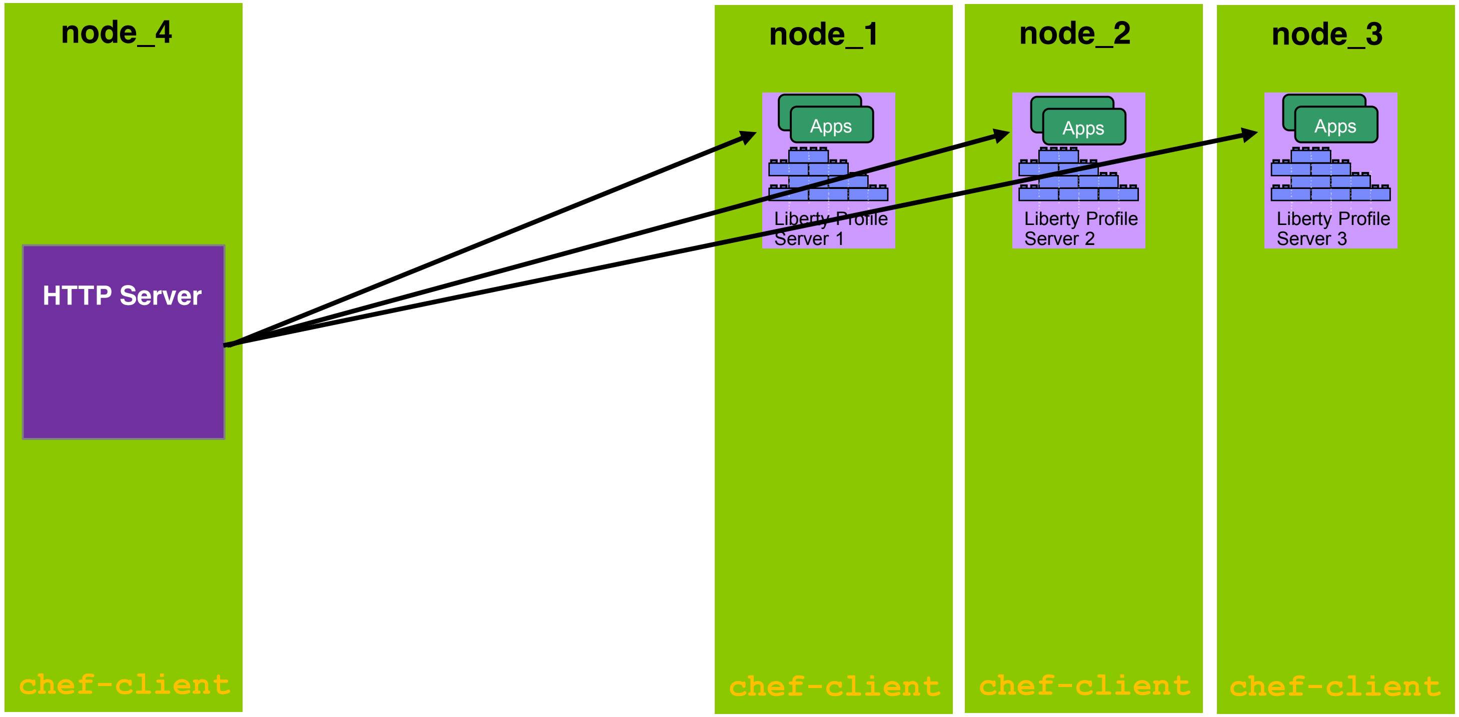
Load balancing … configure web server



```
<Proxy balancer://mycluster>

        BalancerMember http://9.20.123.14:9080
        BalancerMember http://9.20.123.15:9080
        BalancerMember http://9.20.123.16:9080
```

Load balancing … sample

Apache 2.0 license

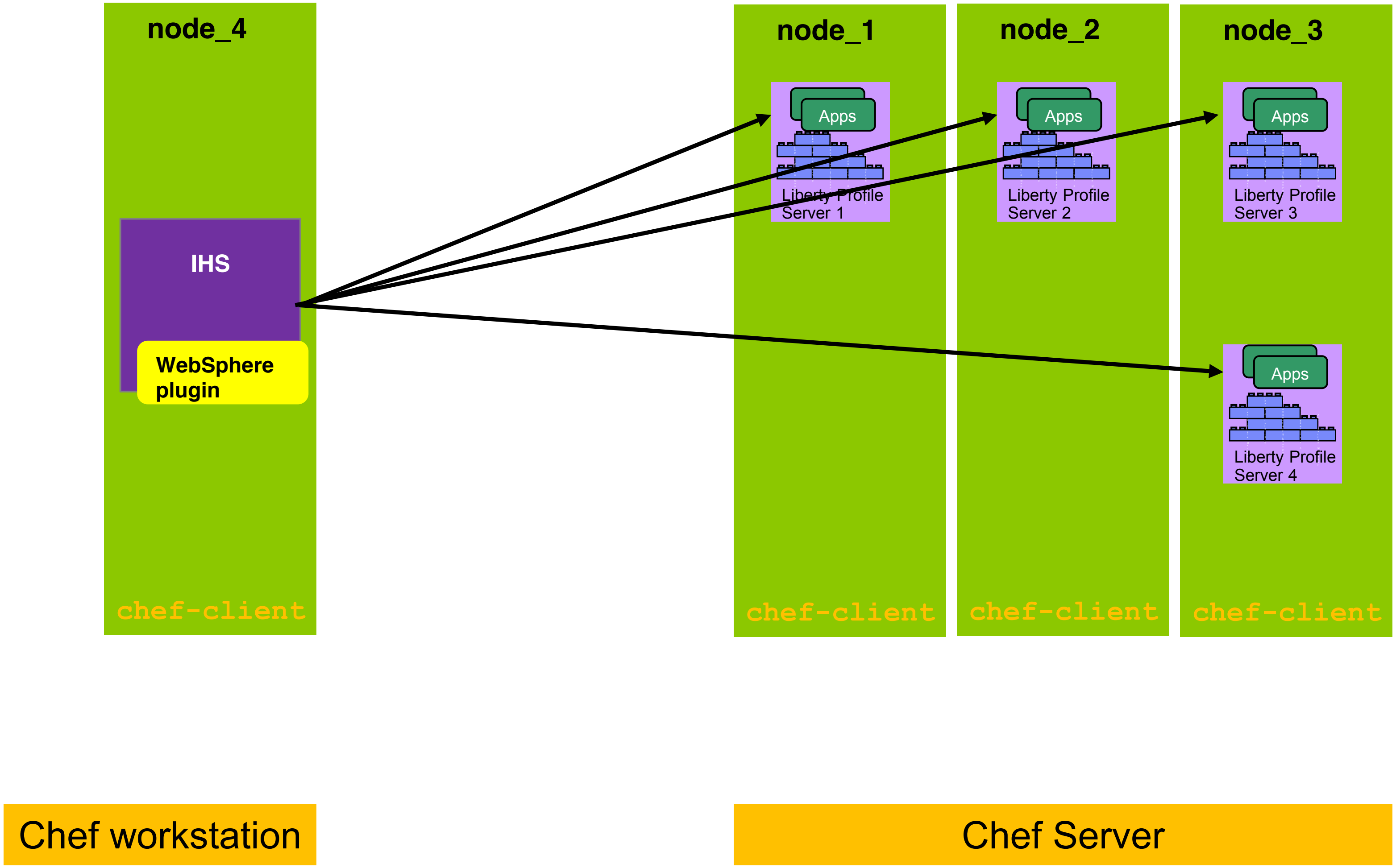


Chef recipe:
https://github.com/WASdev/**ci.chef.wlp.samples**/blob/master/recipes/**proxy.rb**

Chef workstation

Chef Server

What’s next: IBM HTTP Server



What’s next?

In development

- IBM Installation Manager
  https://github.com/WASdev/ci.chef.iim

What’s next?

In development

- IBM Installation Manager
  https://github.com/WASdev/ci.chef.iim

- IBM HTTP Server
  https://github.com/WASdev/ci.chef.ihs

What’s next?

In development

- IBM Installation Manager
  https://github.com/WASdev/ci.chef.iim

- IBM HTTP Server
  https://github.com/WASdev/ci.chef.ihs

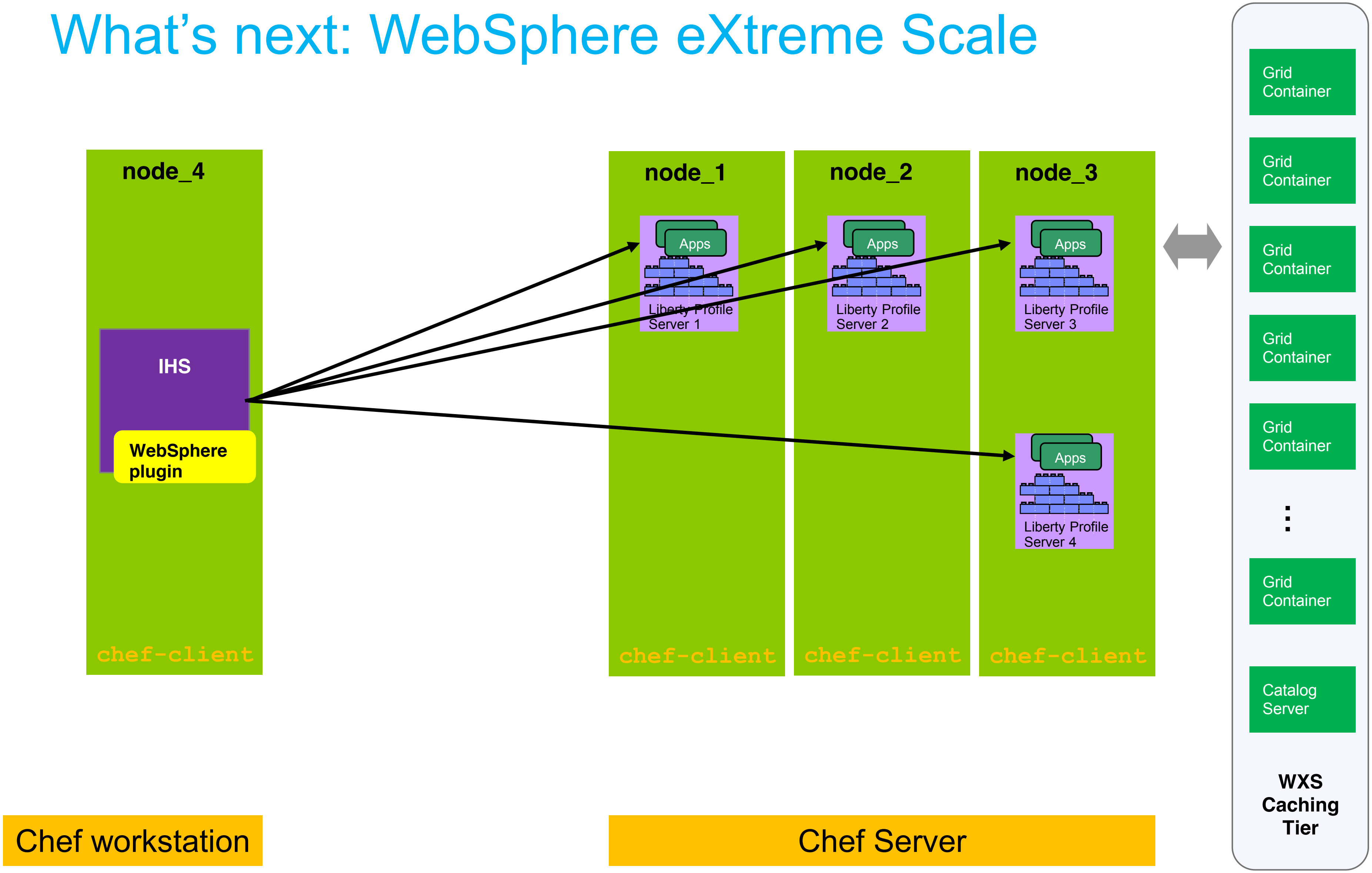
- WebSphere eXtreme Scale
  https://github.com/WASdev/ci.chef.wxs

What’s next?

In development

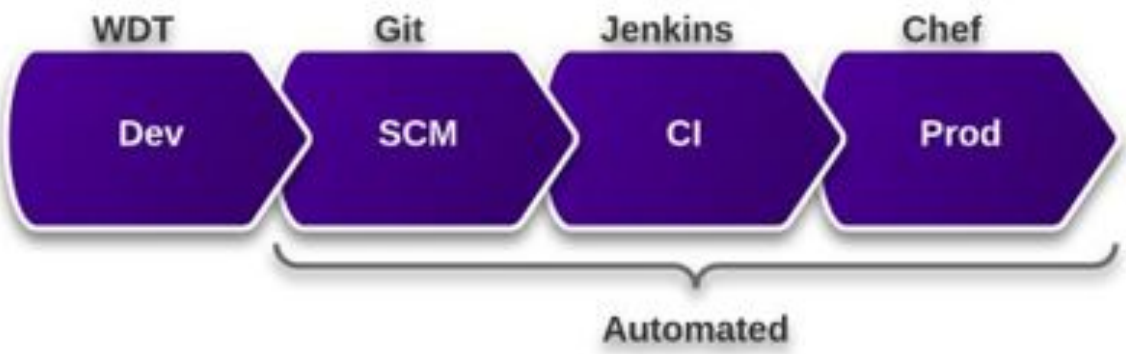
- IBM Installation Manager
  https://github.com/WASdev/ci.chef.iim
- IBM HTTP Server
  https://github.com/WASdev/ci.chef.ihs
- WebSphere eXtreme Scale
  https://github.com/WASdev/ci.chef.wxs
- Full WebSphere Application Server
  https://github.com/WASdev/ci.chef.was

What's next: WebSphere eXtreme Scale



Walk through 'lab'

- Continuous development pipeline example on wasdev.net



- http://ibm.biz/liberty-devops

Chef & Liberty Profile: no-charge downloads

- Chef is Infrastructure as Code
  - model your infrastructure as code. www.getchef.com
- WebSphere Liberty Profile
  - designed for Developers, ideal for Ops
- Visit us at WASdev and our GitHub org

standards.jsr352.jbatch

lib.rtcomm.clientjs

ci.maven.tools

ci.chef.wlp

github.com/WASdev

Liberty profile

September beta!

What's new

Connect & Contribute

GOTO: Aarhus, Denmark

wasdev.net

Questions?

We Value Your Feedback

- Don’t forget to submit your session and speaker feedback! Your feedback is very important to us – we use it to continually improve the event.

Thank You

Legal Disclaimer

-
- The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.
- References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
- IBM, the IBM logo, WebSphere, are trademarks of International Business Machines Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Chef