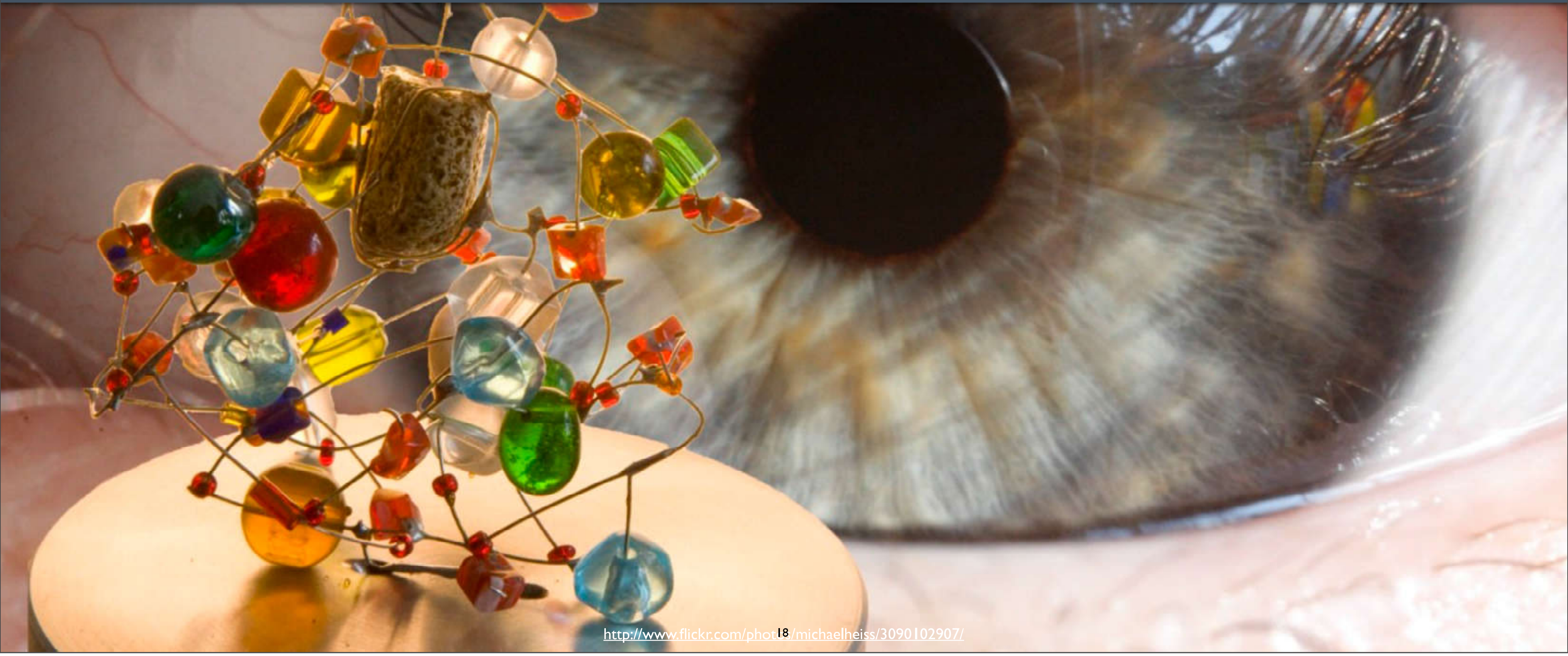
Overview of Chef

v2.1.6

Lesson Objectives

- After completing the lesson, you will be able to
  - Describe how Chef thinks about Infrastructure Automation
  - Define the following terms:
    - Node
    - Resource
    - Recipe
    - Cookbook
    - Run List
    - Roles
    - Search

Complexity



http://www.flickr.com/phot[illegible]/michaelheiss/3090102907/

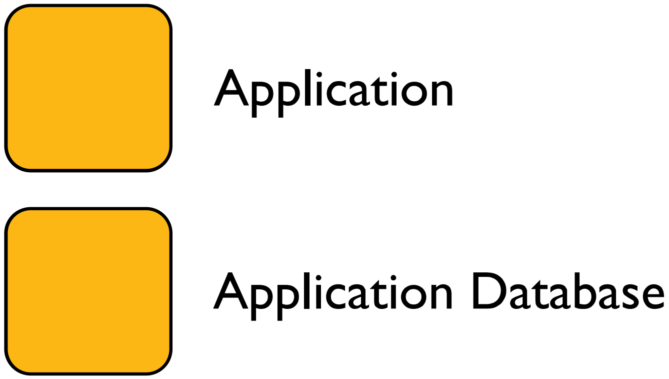
Items of Manipulation (Resources)

- Networking
- Files
- Directories
- Symlinks
- Mounts
- Registry Keys

- Powershell Scripts
- Users
- Groups
- Packages
- Services
- Filesystems

A tale of growth...

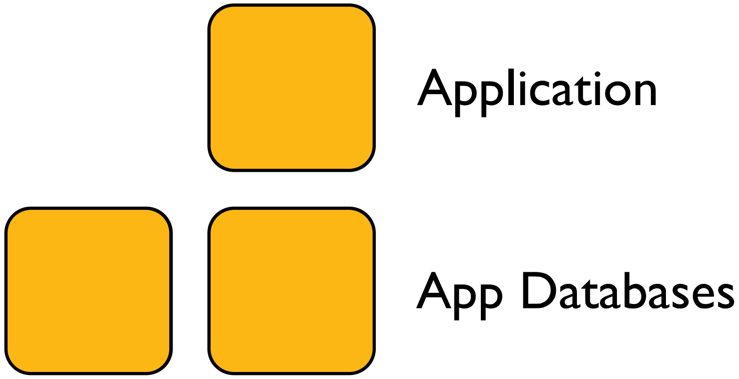
Add a database



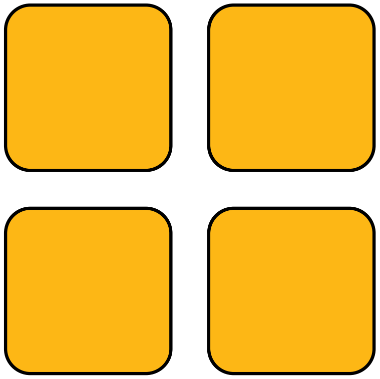
Application

Application Database

Make database redundant



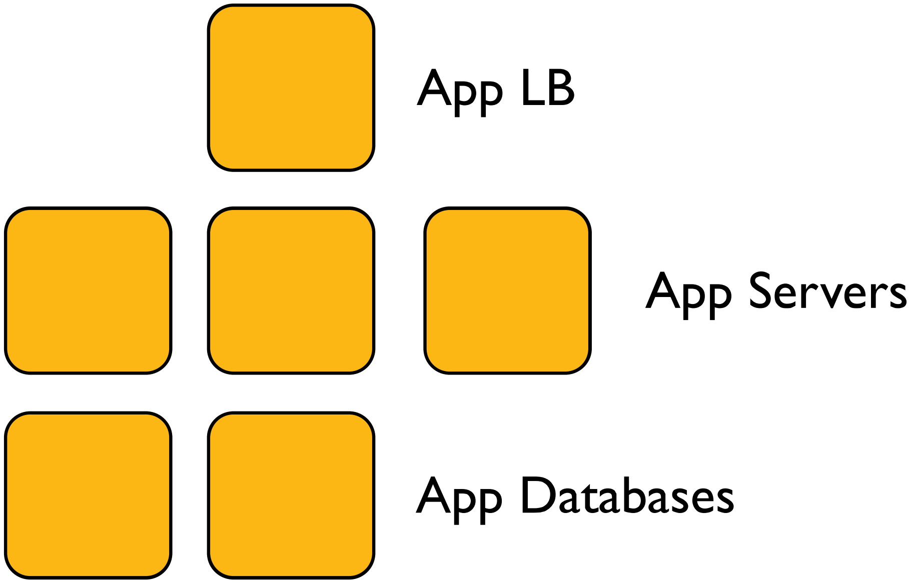
Application server redundancy



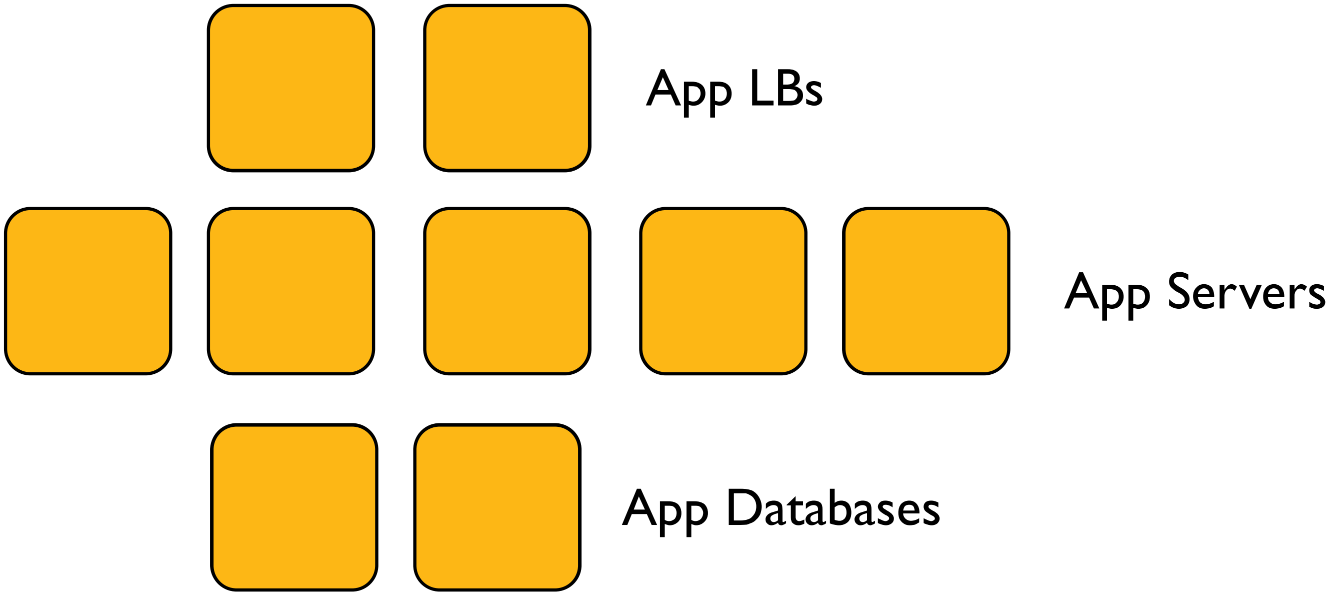
App Servers

App Databases

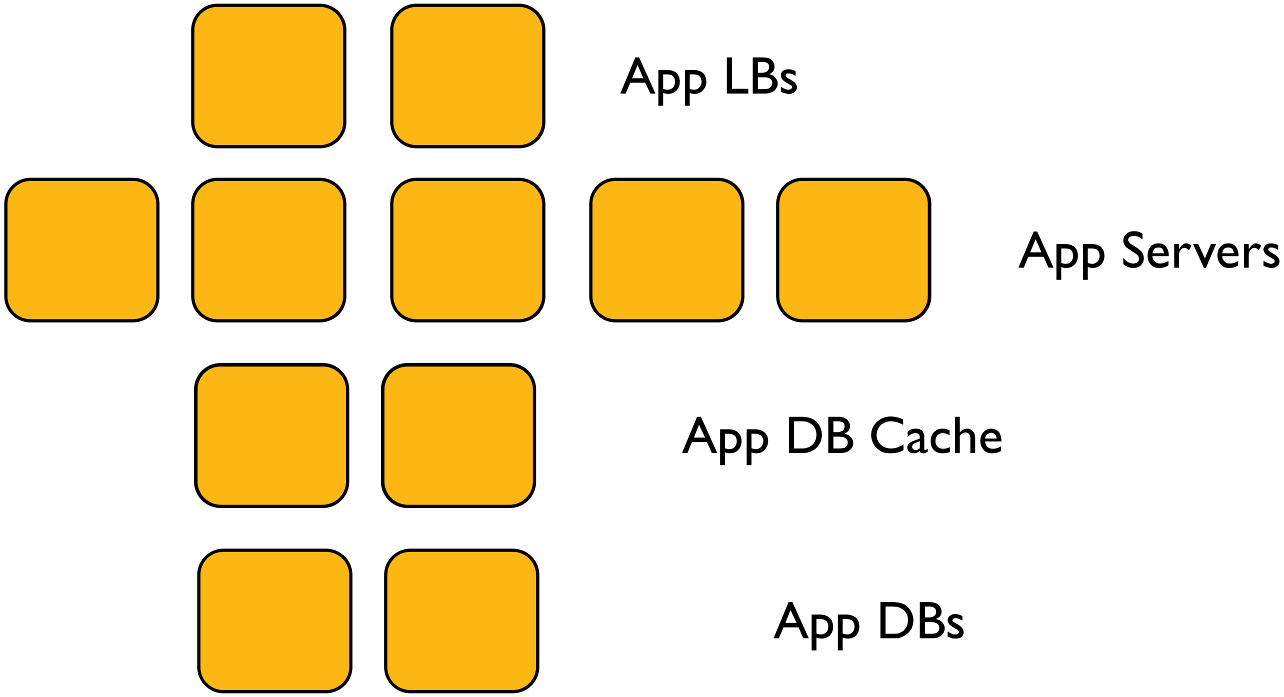
Add a load balancer



Webscale!



Now we need a caching layer



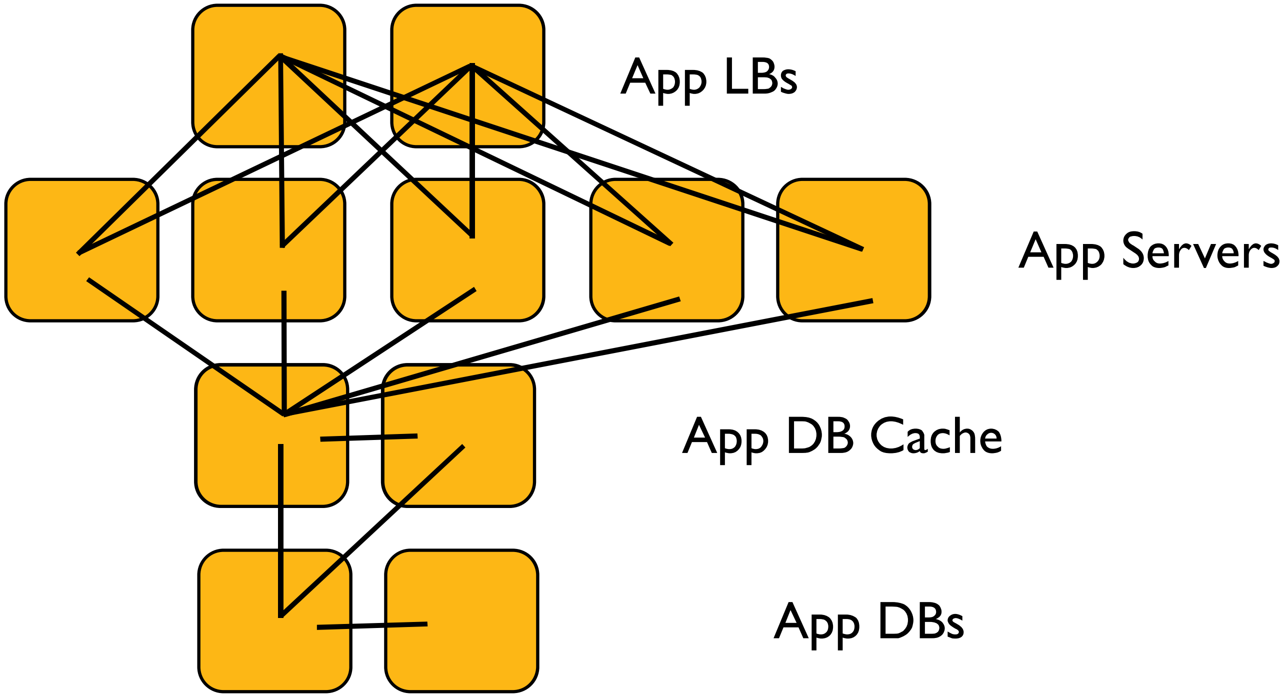
App LBs

App Servers

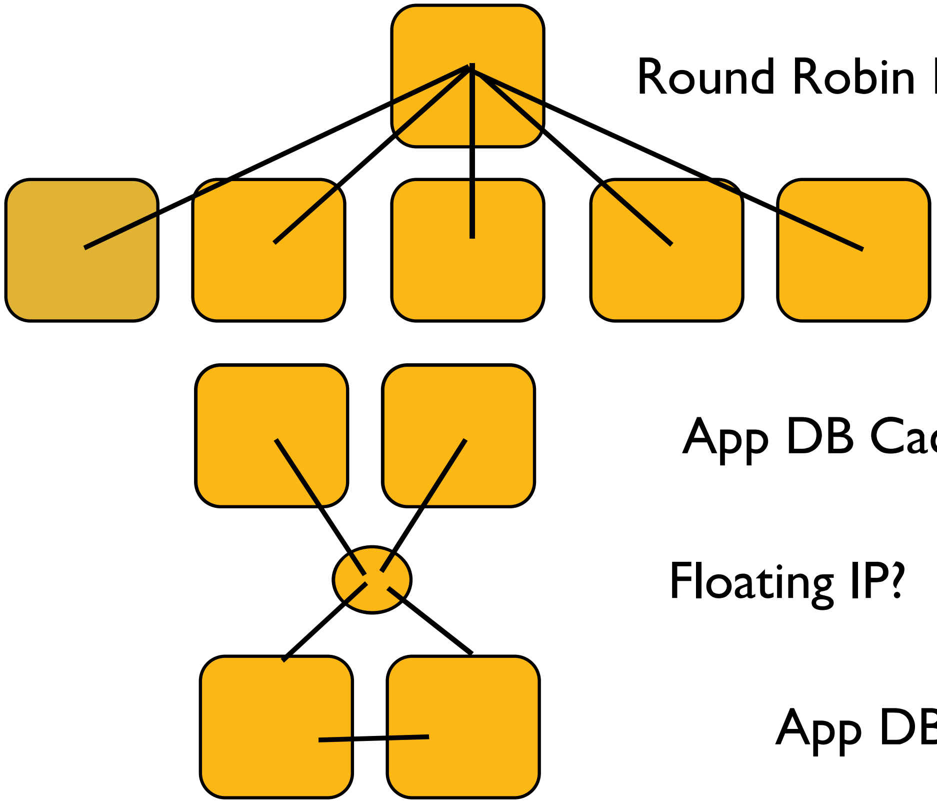
App DB Cache

App DBs

Infrastructure has a Topology



Your Infrastructure is a Snowflake



Round Robin DNS

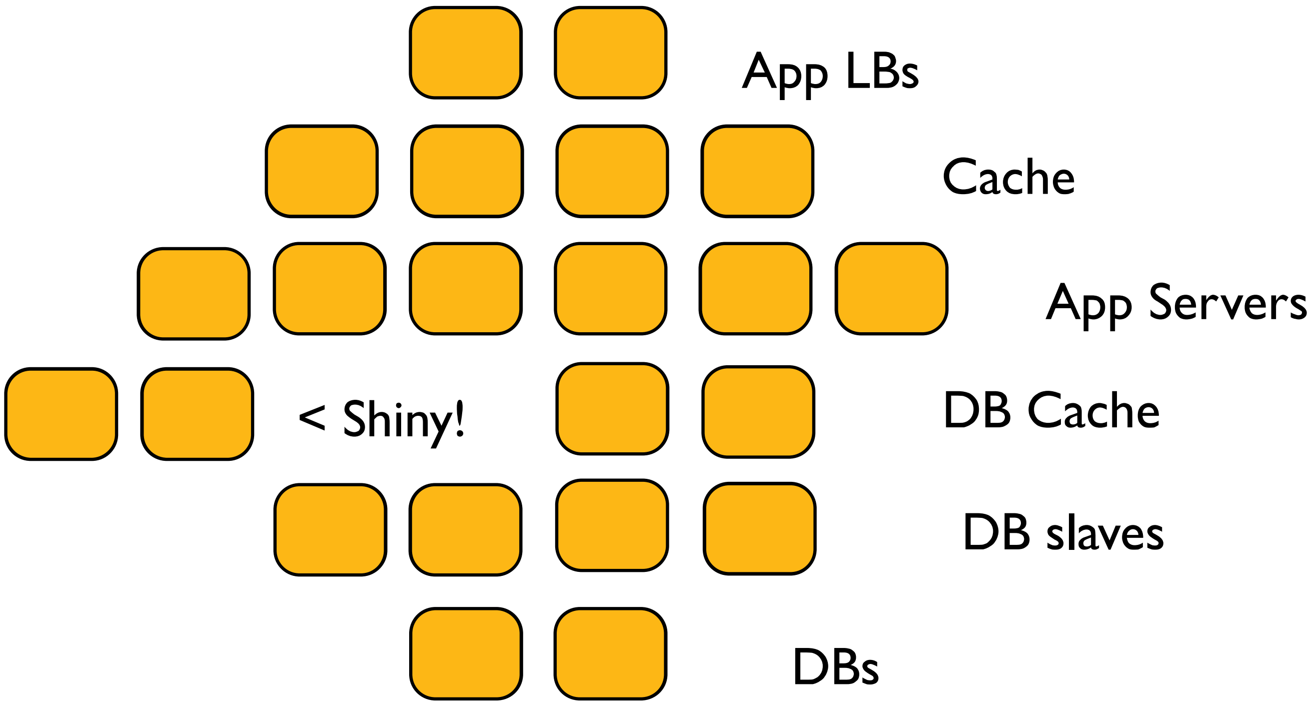
App Servers

App DB Cache

Floating IP?

App DBs

Complexity Increases Quickly



App LBs

Cache

App Servers

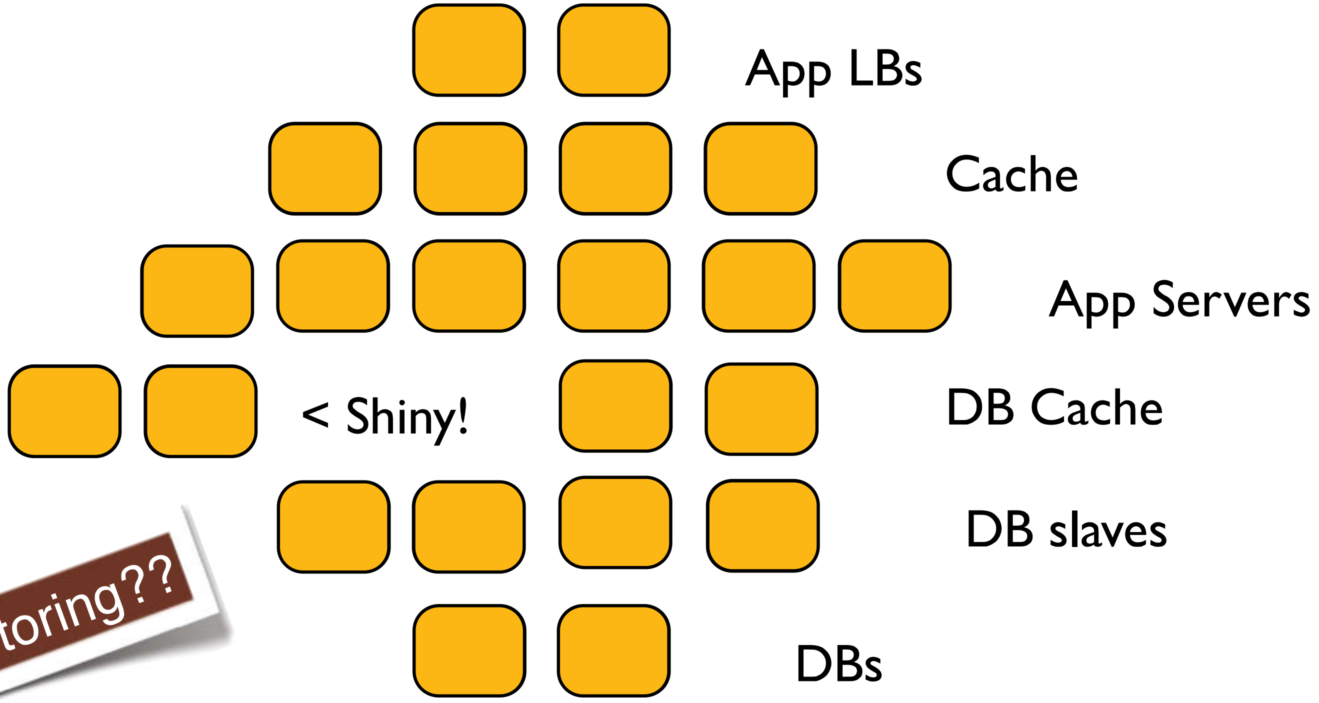
< Shiny!

DB Cache

DB slaves

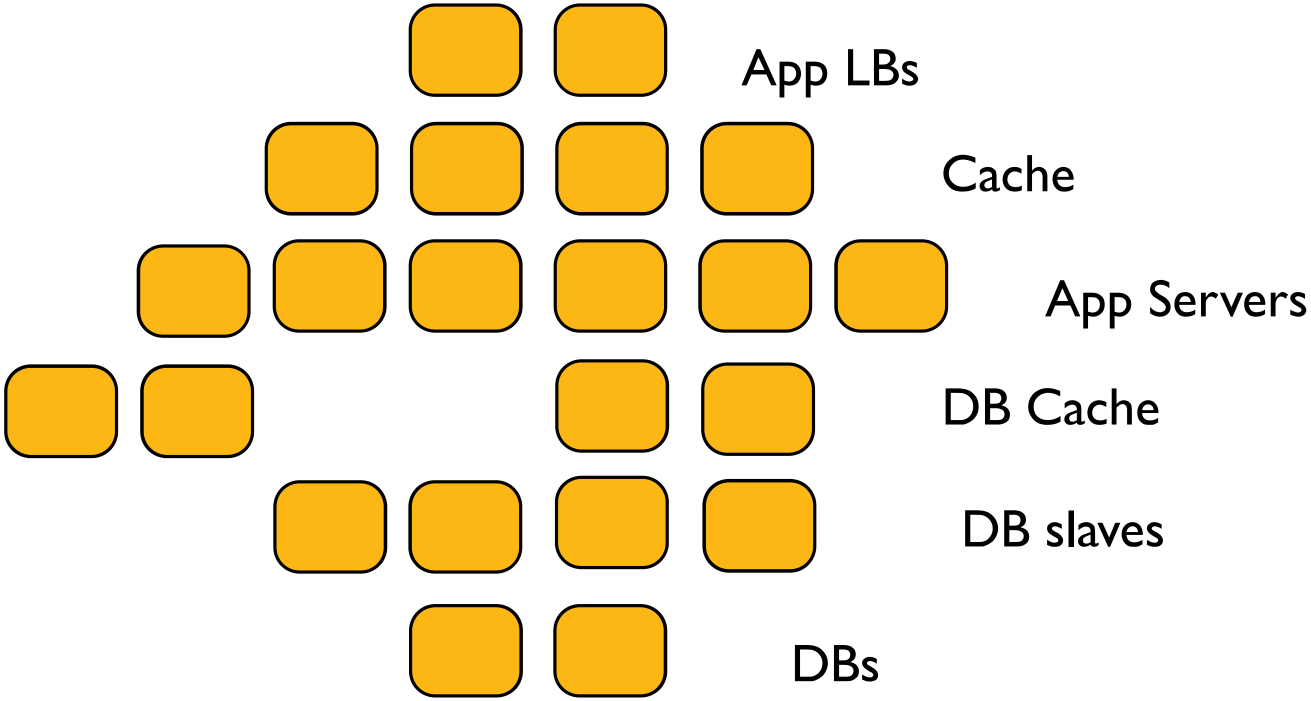
DBs

Complexity Increases Quickly



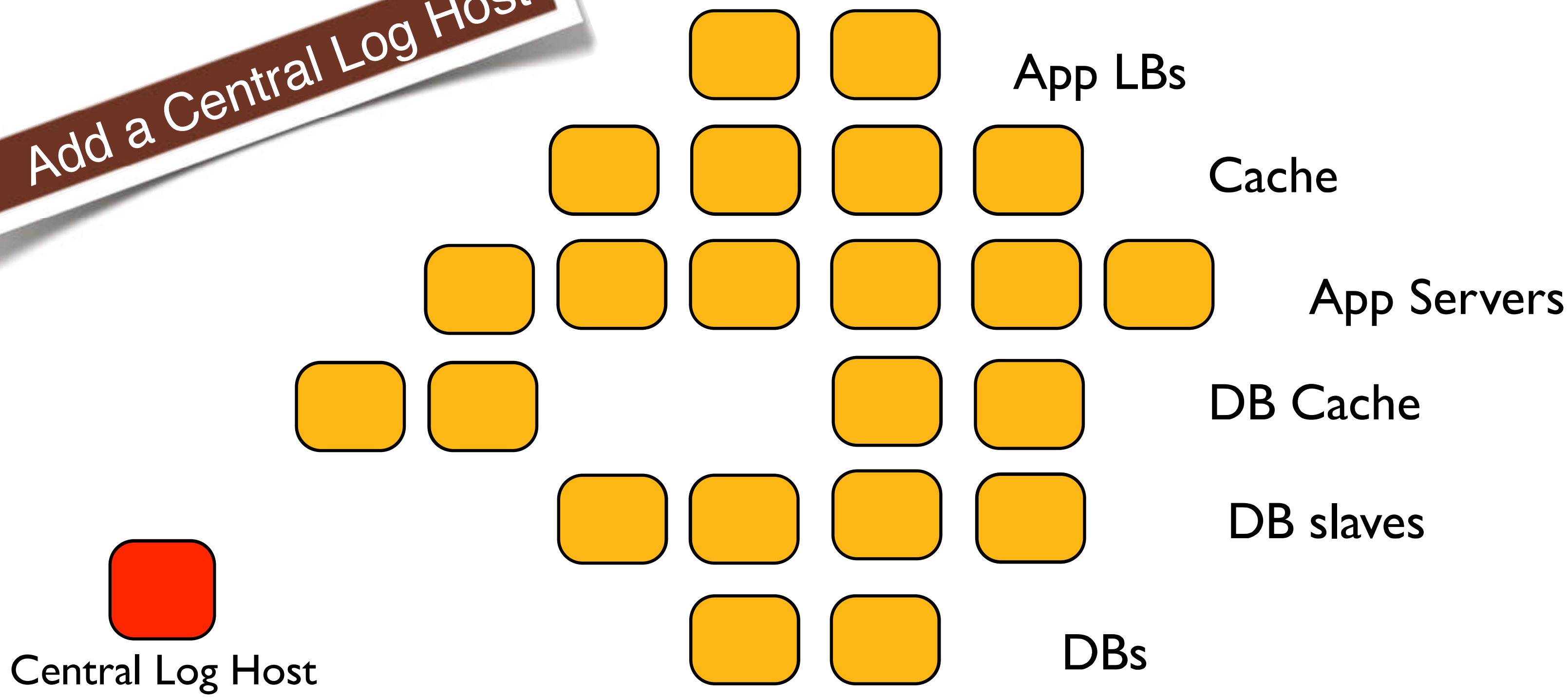
Are we monitoring??

...and change happens!



...and change happens!

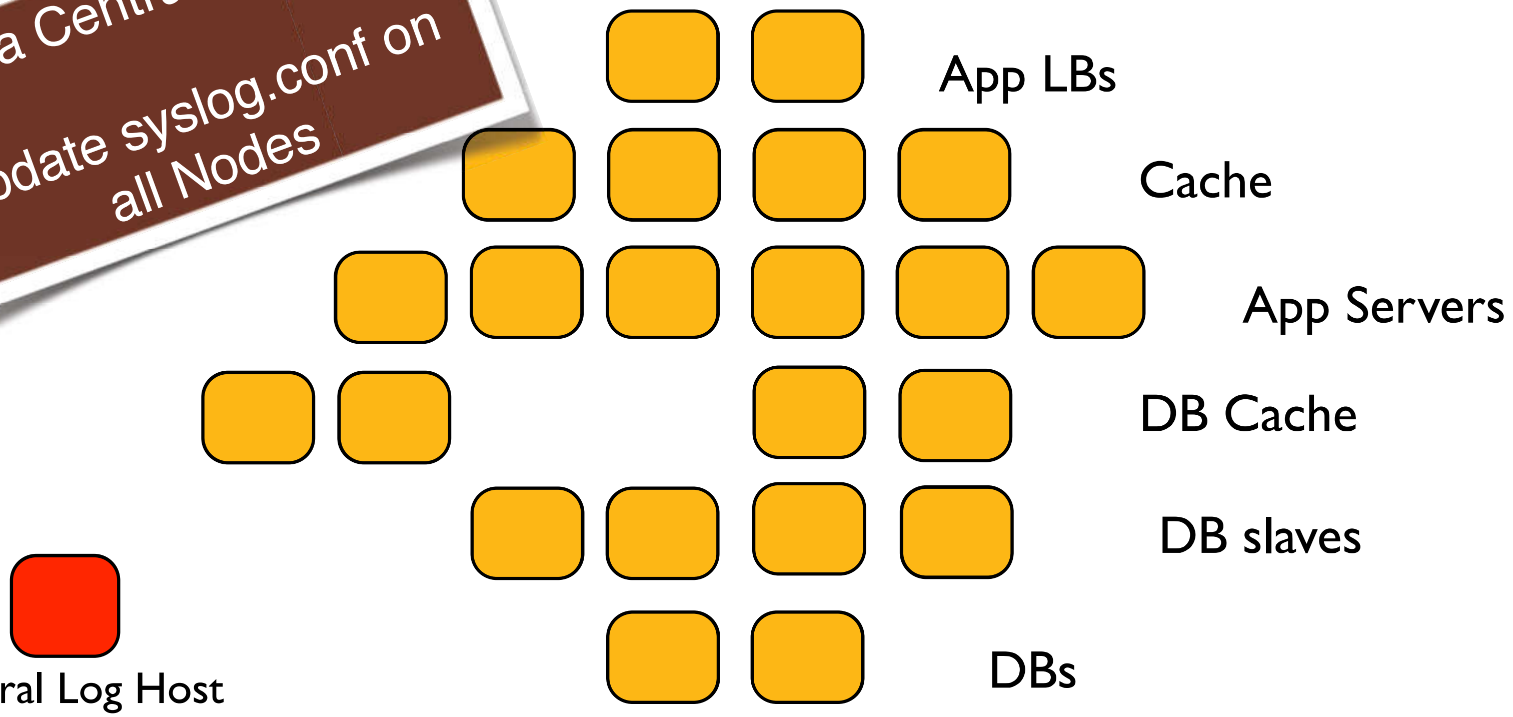
Add a Central Log Host



...and change happens!

Add a Central Log Host

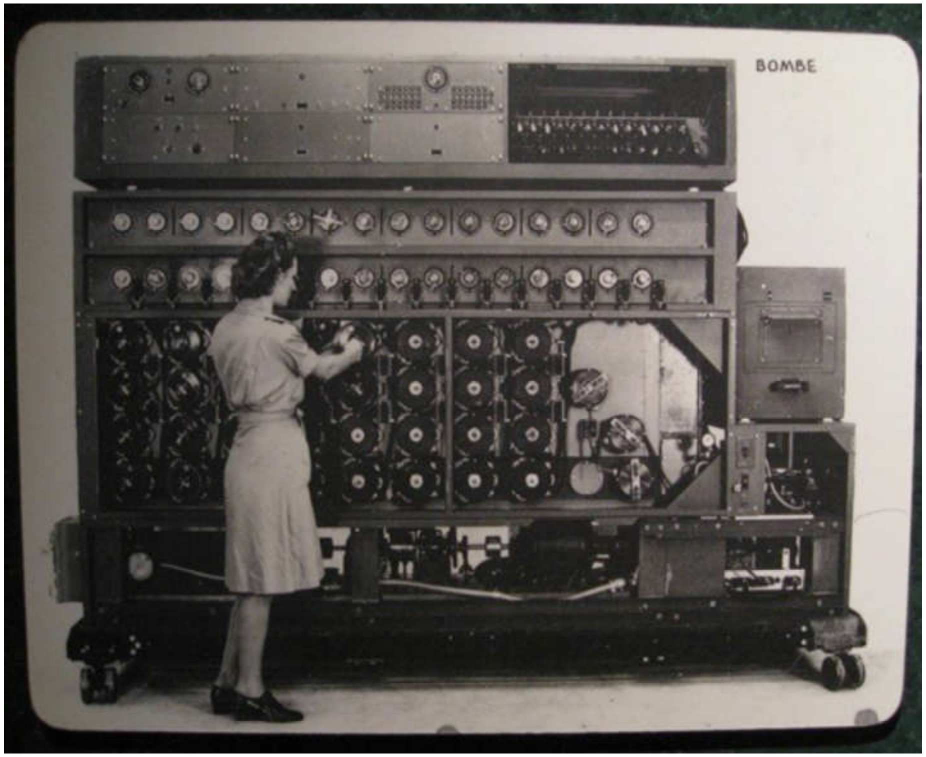
Update syslog.conf on all Nodes



Chef Solves This Problem

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

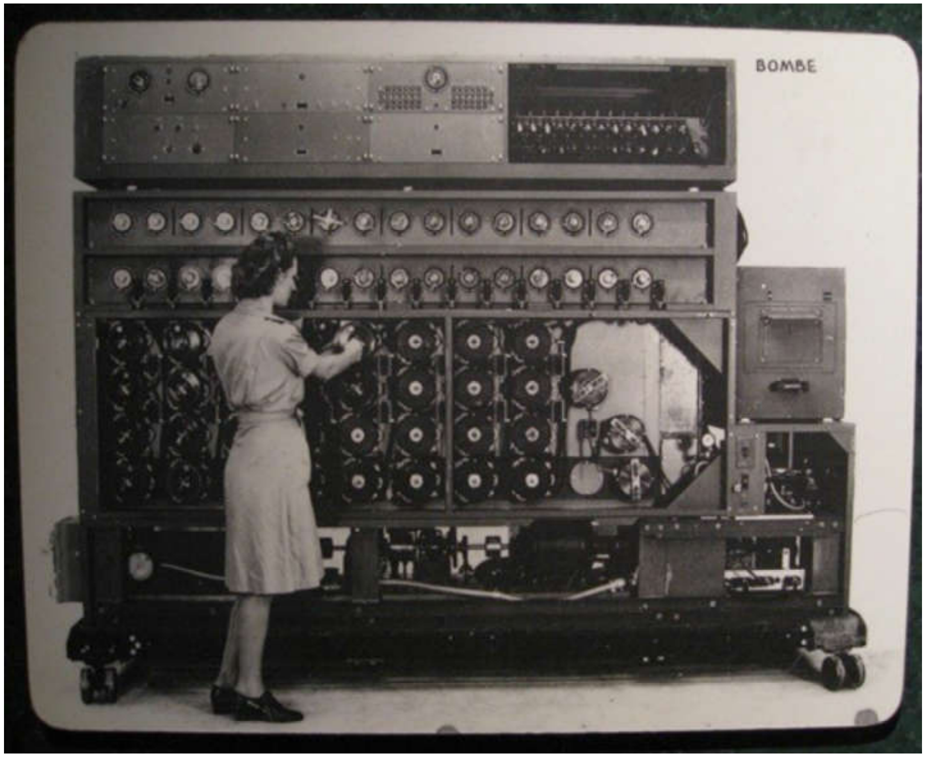
Chef is Infrastructure as Code



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

- Programmatically provision and configure components

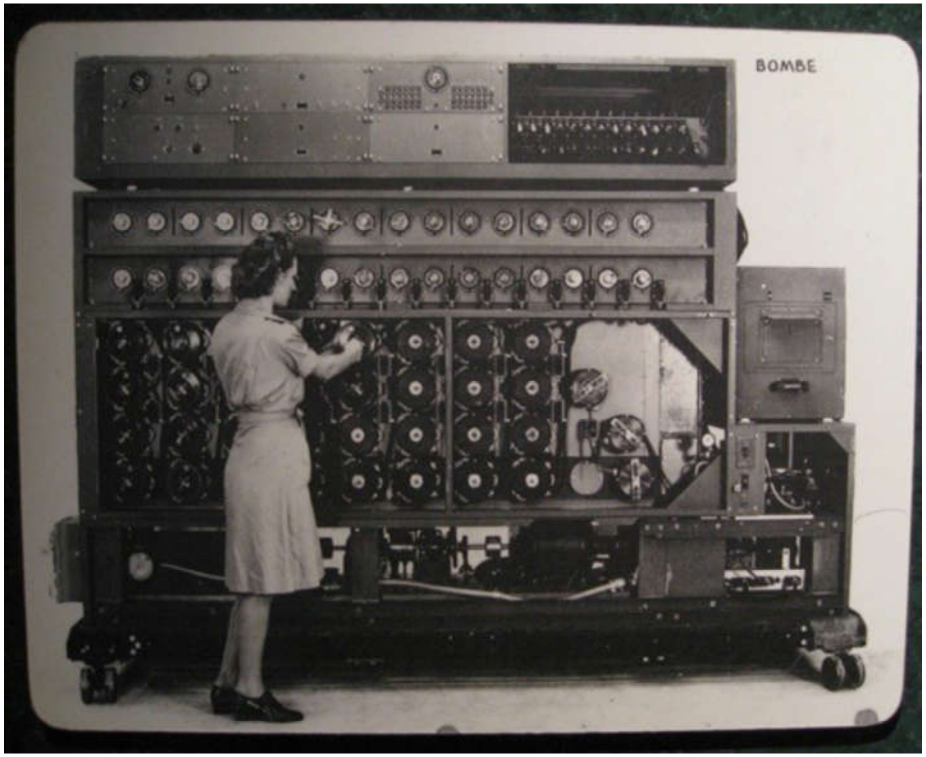
Chef is Infrastructure as Code



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

- Treat like any other code base

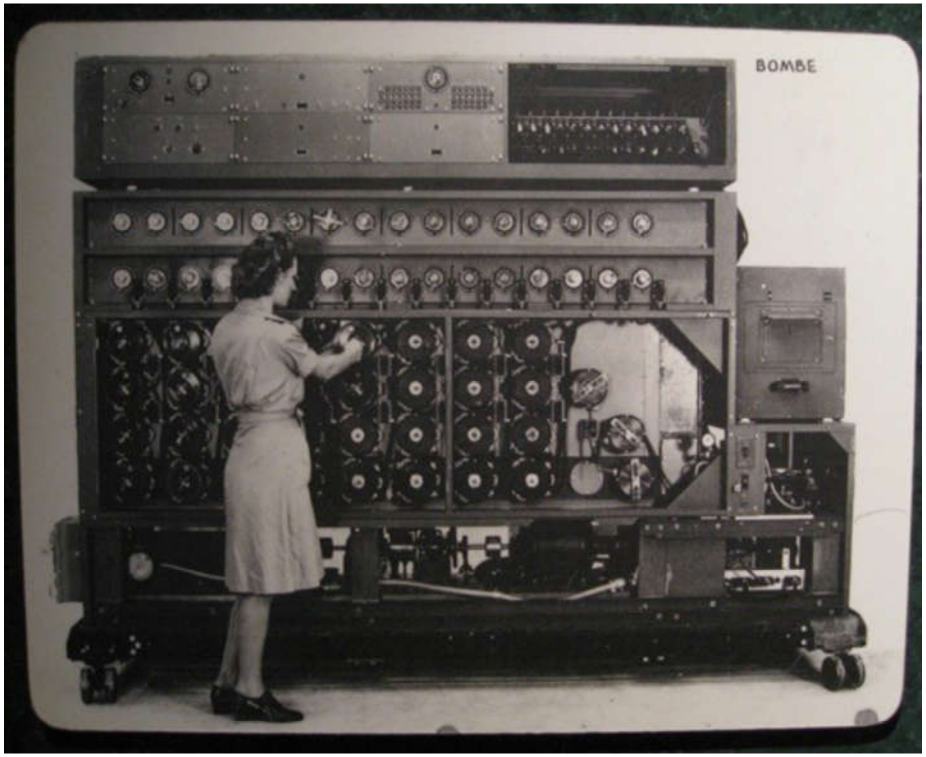
Chef is Infrastructure as Code



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

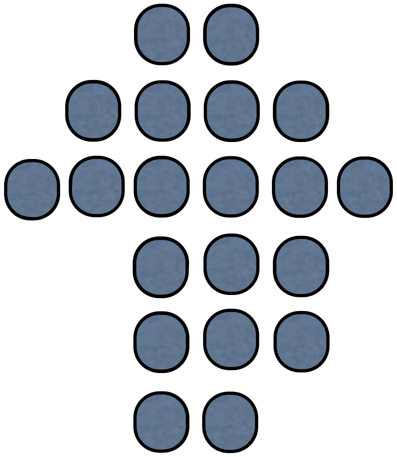
- You define the policy in your Chef configuration
- Your policy states what state each resource should be in, but not how to get there
- Chef-client will pull the policy from the Chef Server and enforce the policy on the Node

Managing Complexity

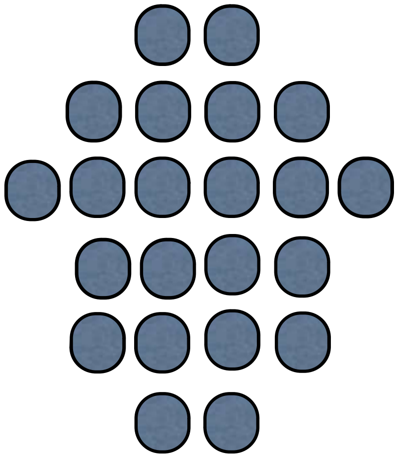
- Organizations
- Environments
- Roles
- Nodes
- Recipes
- Cookbooks
- Search

Organizations

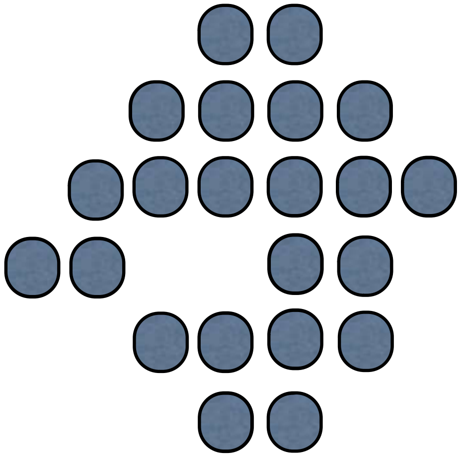
My Infrastructure



Your Infrastructure



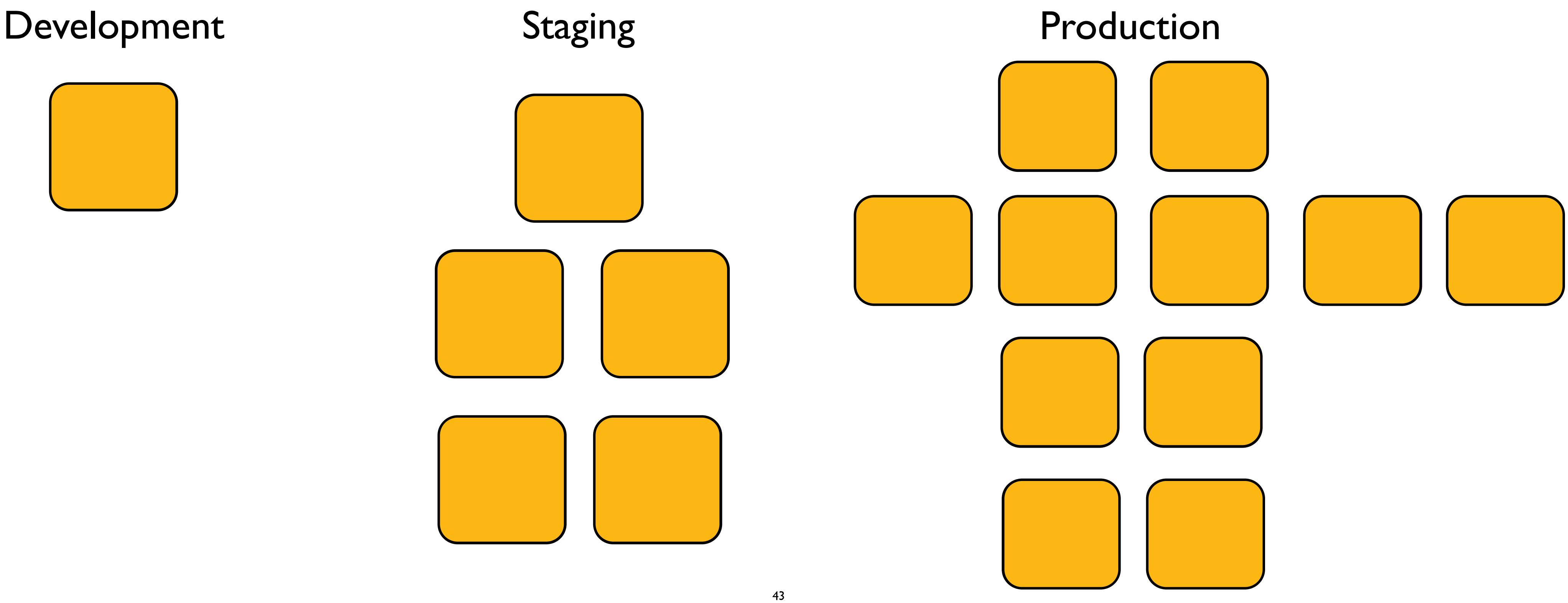
Their Infrastructure



Organizations

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

Environments



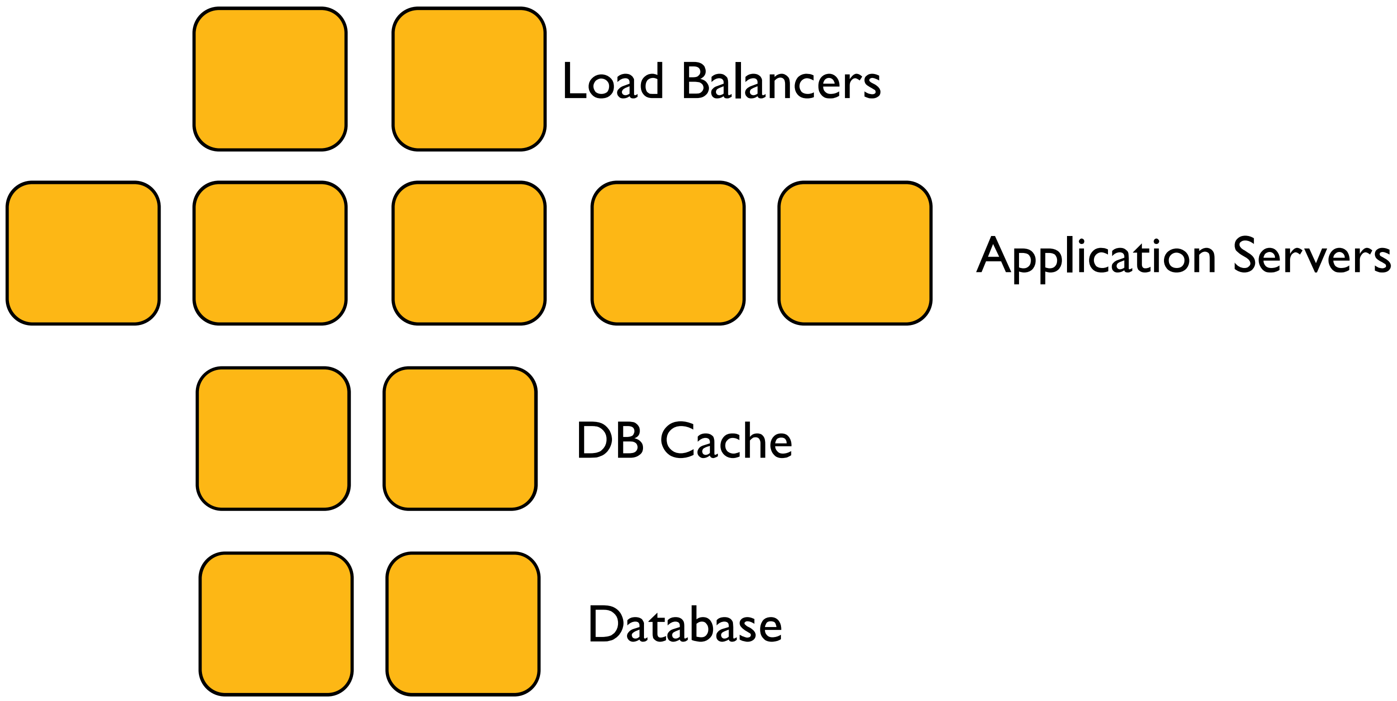
Environments

- 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



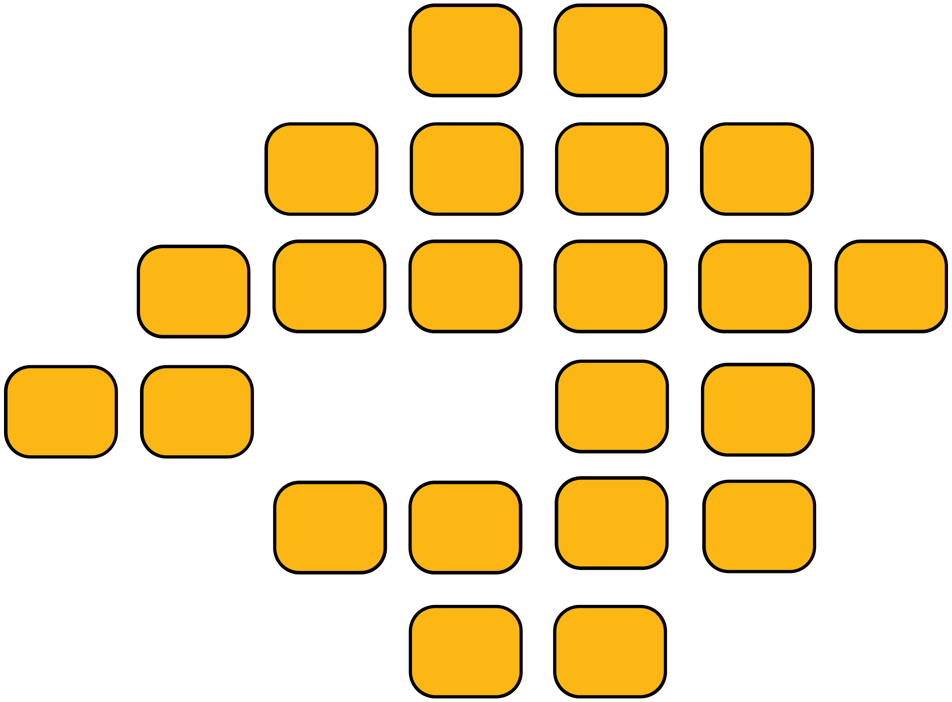
Roles

- 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

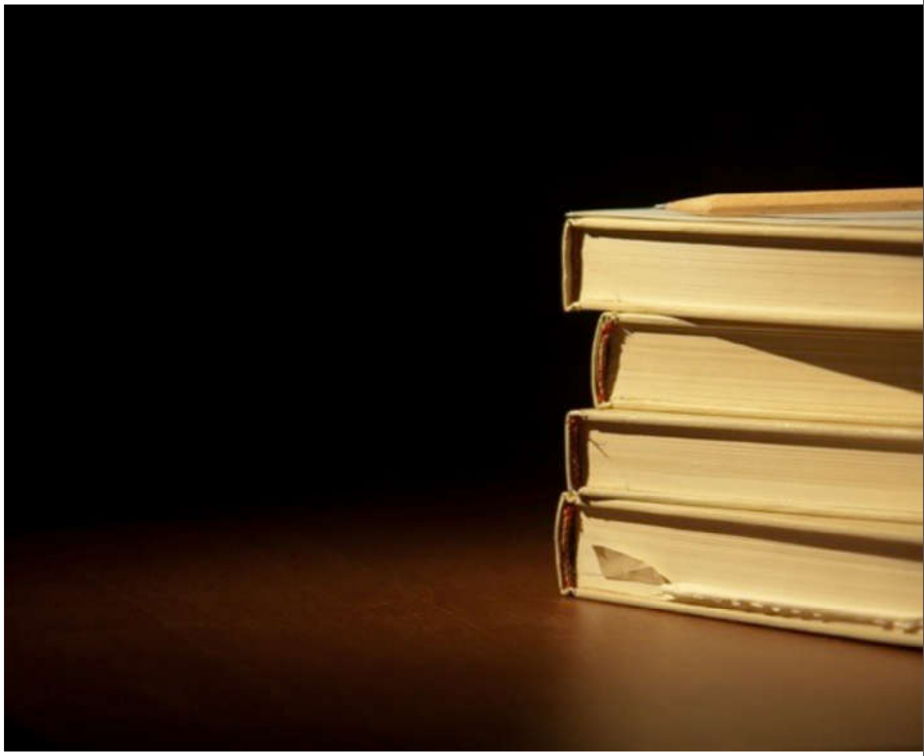
```
package "apache2"
```

```
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
```

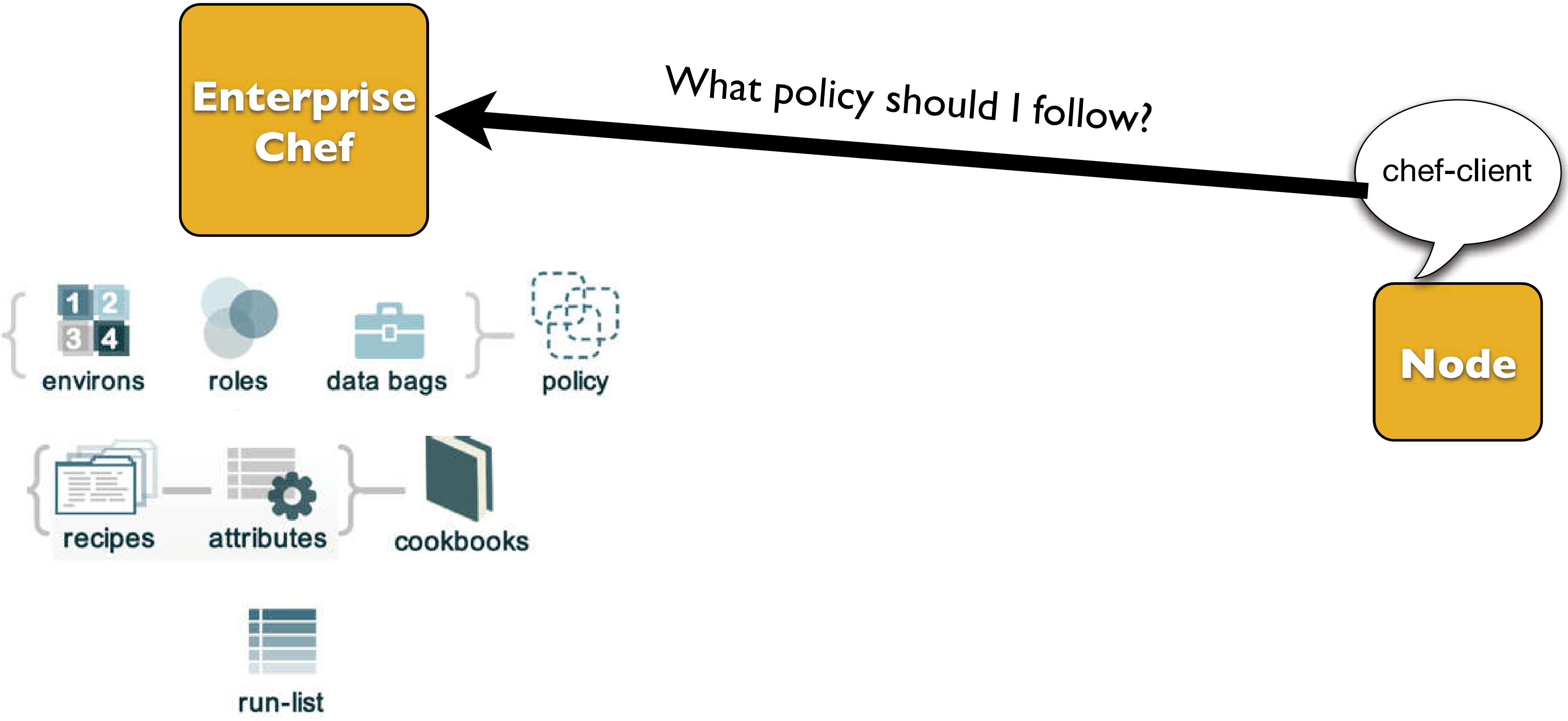
Cookbooks

- 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

environs

roles

data bags

policy

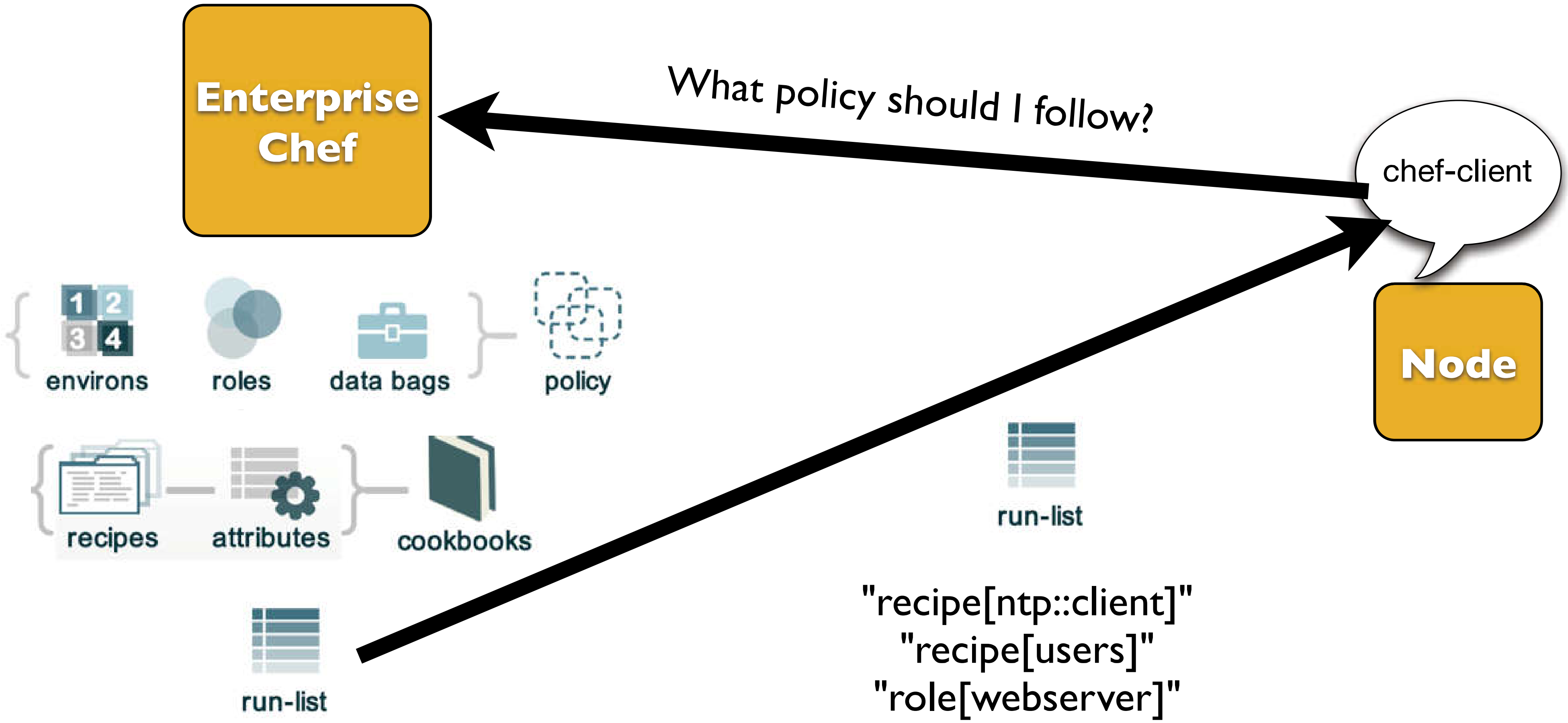
recipes

attributes

cookbooks

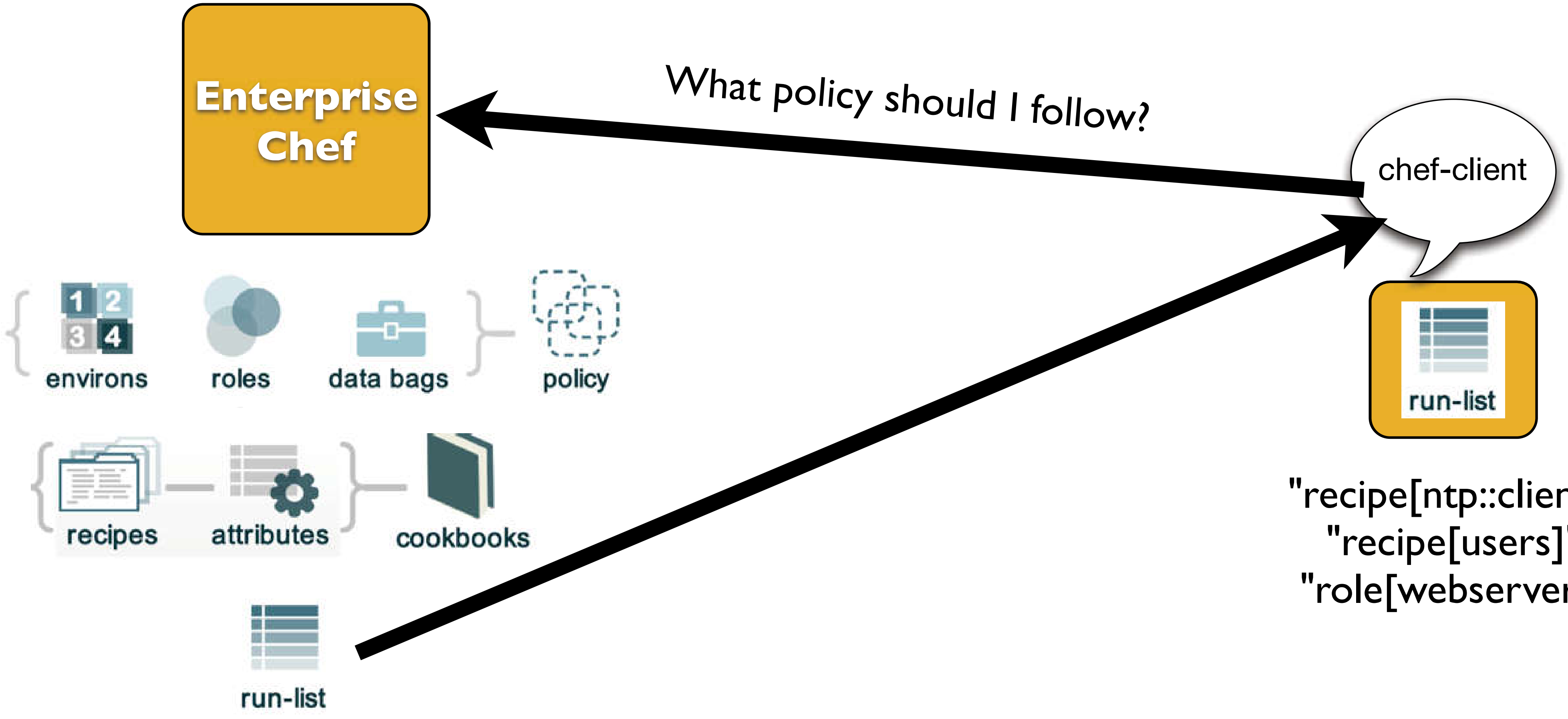
run-list

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