CLOUD FOUNDRY™

# Migrating to Cloud Foundry

Peter Ledbrook - VMware
t: @pledbrook
e: pledbrook@vmware.com

# What's happening with applications today?

**Frameworks are what really matter**

- Developer productivity and innovation
- Reduce time to market


**New application types**

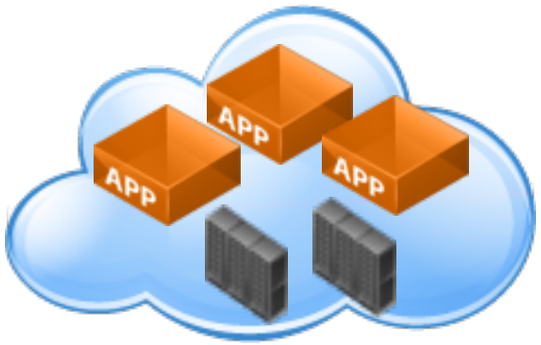
- Mobile, Social, SaaS
- Apps released early and often


**Data intensive**

- Emerging requirements: elasticity, multi-cloud
- Web orientation drives exponential data volumes


**Deployed on virtual and cloud infrastructures**

- Virtualization, Cloud, PaaS

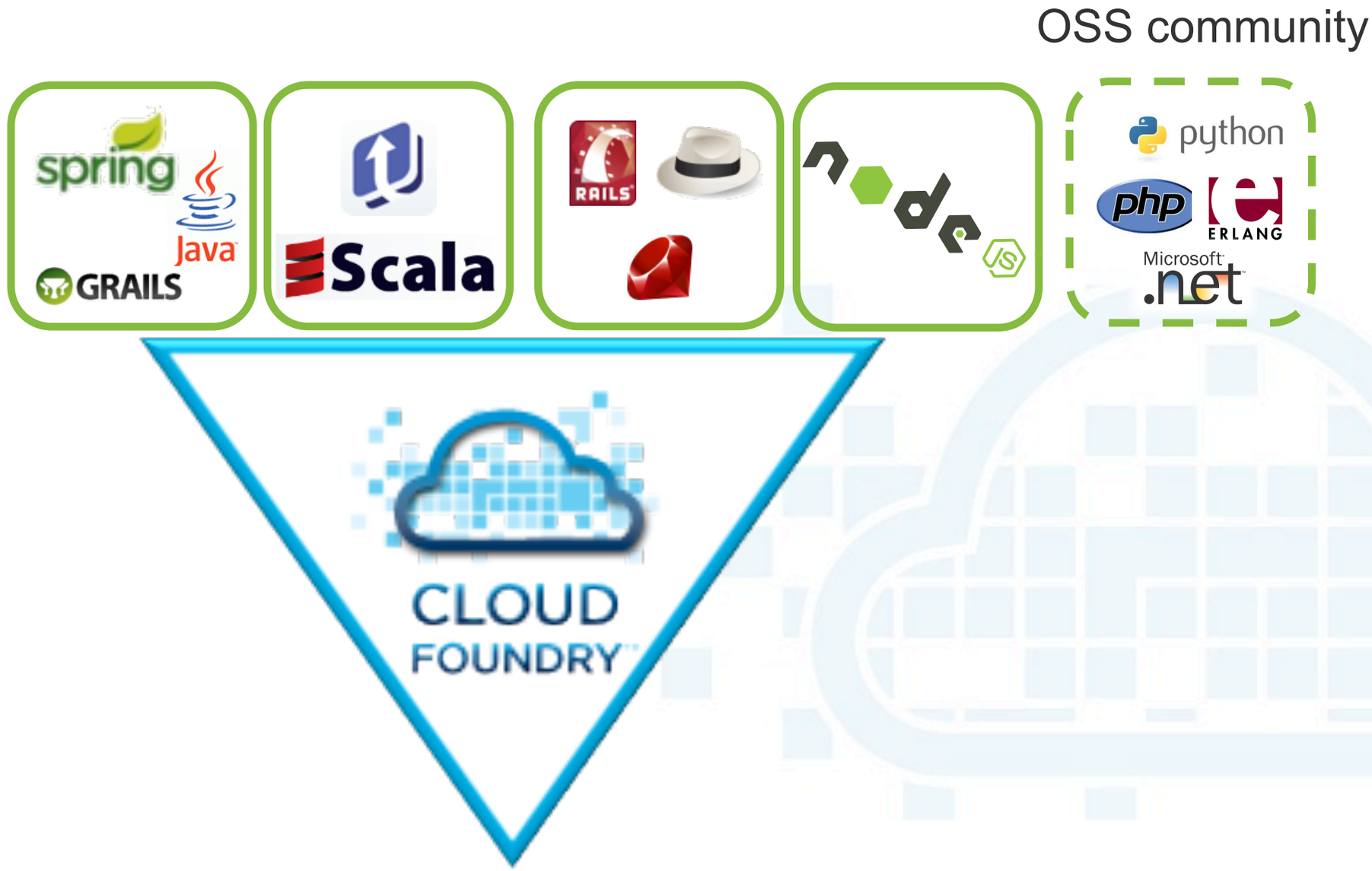



*The Open*
*Platform as a Service*

# Deploy and scale applications in seconds, without locking yourself into a single cloud

**Simple, Open,**
**Flexible, Scalable**

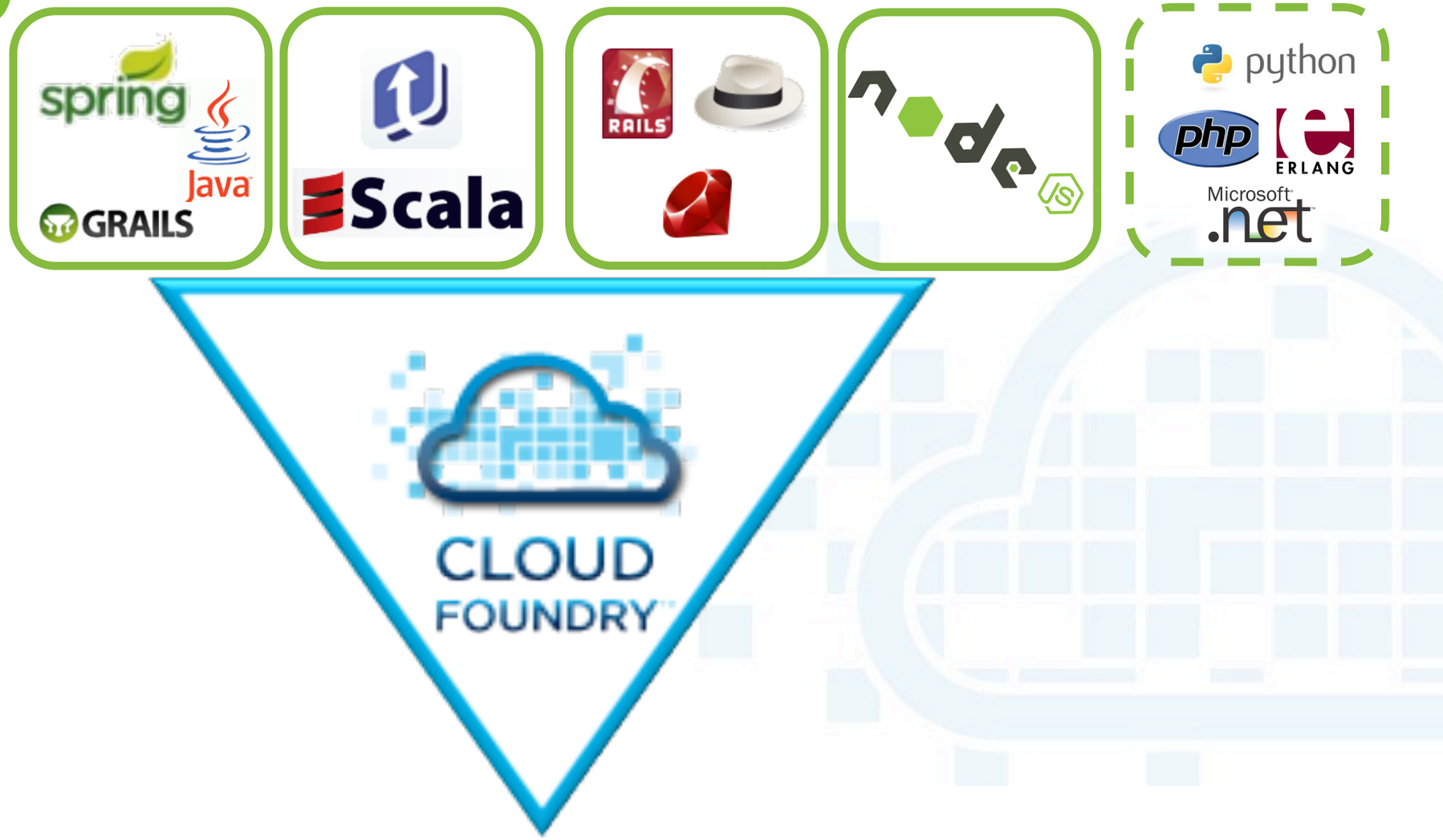
# Choice of frameworks, services & clouds



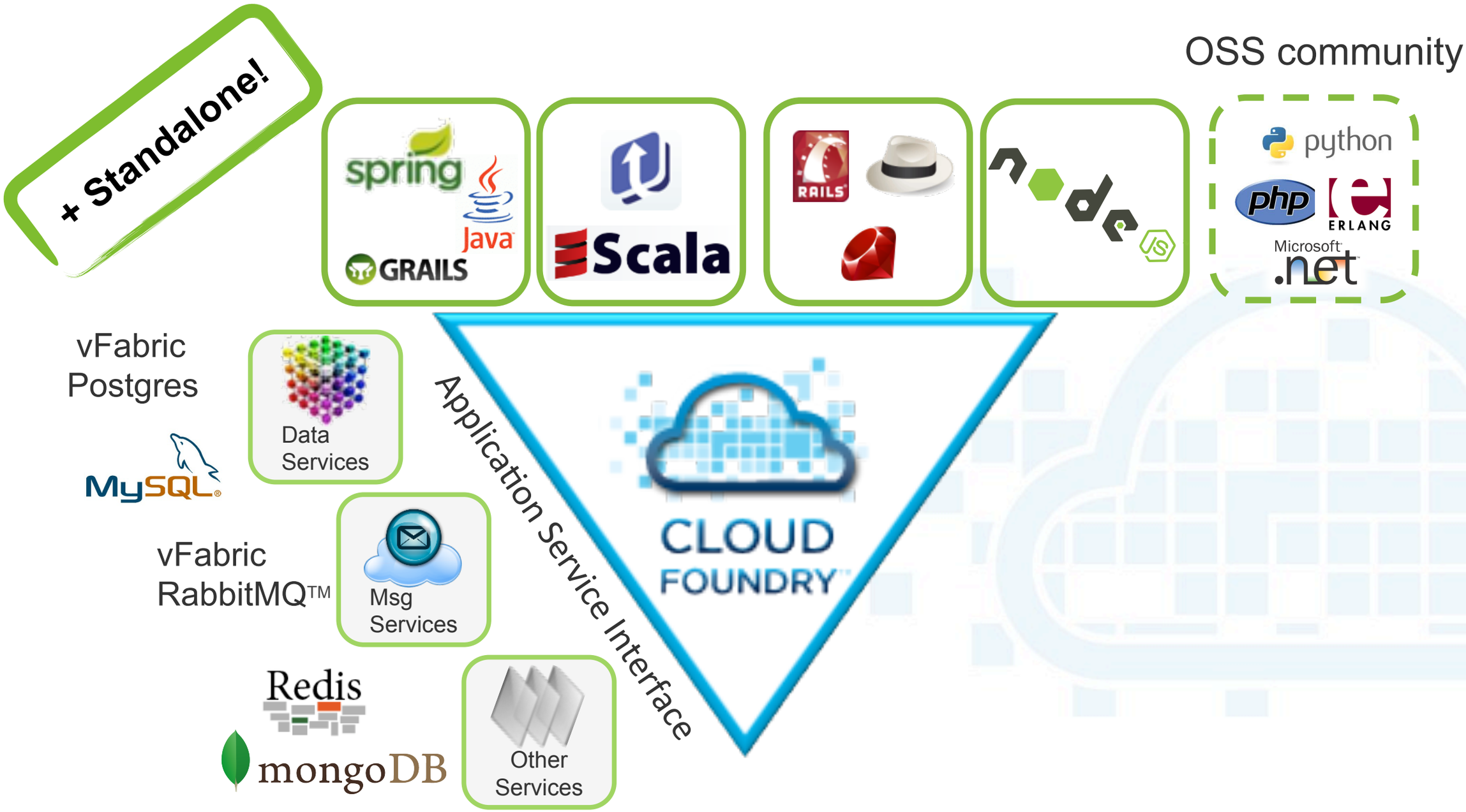
# Choice of frameworks, services & clouds

+ Standalone!

OSS community



# Choice of frameworks, services & clouds




Additional partners services …

# Choice of frameworks, services & clouds

+ Standalone!

OSS community

spring Java GRAILS

Scala

RAILS

node.js

python php ERLANG Microsoft .net

vFabric Postgres

MySQL

Data Services

vFabric RabbitMQ™

Msg Services

Redis

mongoDB

Other Services

Application Service Interface

CLOUD FOUNDRY™

Cloud Provider Interface

Private Clouds

Public Clouds

Micro Clouds

COMING SOON

Partners

CLOUD FOUNDRY.COM

Cloud on a Stick
cloudfoundry.com/micro

Additional partners services …

Avoid Lock-in

# Real choice of provider

## Choice of Private Cloud Distributions

ActiveState

DELL

Cloud on a Stick
cloudfoundry.com/micro

CANONICAL

ServiceMesh

nimbula

OPSCODE

SCALR

## Choice of Public Cloud Providers

appfog

CLOUD FOUNDRY.COM

ENSTRATUS

RIGHT SCALE

bluelock.

Tier 3
Tech Support

virtacore SYSTEMS

## Choice of Cloud Infrastructure

openstack

amazon web services

the rackspace cloud

vmware vCLOUD POWERED

VMware vSphere

Bare metal

Eucalyptus

# Real choice of provider

## Choice of Private Cloud Distributions

ActiveState

DELL

Cloud on a Stick
cloudfoundry.com/micro

CANONICAL

ServiceMesh

nimbula

OPSCODE

SCALR

## Choice of Public Cloud Providers

appfog

CLOUD FOUNDRY.COM

ENSTRATUS

RIGHT SCALE

bluelock.

Tier 3
Tech Support

virtacore SYSTEMS

## Choice of Cloud Infrastructure

openstack

amazon web services

the rackspace cloud

vmware vCLOUD POWERED


Bare metal

Eucalyptus

# Deployment tools

vmc (command line)




# Core services






# Developer Perspective

- Program to standard APIs
  - SQL drivers
  - Mongo client libraries
  - ...
- Connection settings from VCAP_SERVICES environment variable

# When you're ready to deploy...

```
$ vmc target <any cloud>

$ vmc push <my-app>
  > bind services? Yes

$ vmc instances <my-app> +100
```

# When you're ready to deploy...

```
$ vmc target <any cloud>

$ vmc push <my-app>
   > bind services? Yes

$ vmc instances <my-app> +100
```

That's all folks!

# The sample application

- http://grails.org
- Built with Grails
  - JVM web framework
  - Spring MVC under the hood
  - Groovy as the main development language

# Searchable Plugin - Mapping

**Get Started**
Installation
Quick Start
IDE Setup
Tutorials
Screencasts

**Reference**
Documentation
FAQs
Roadmap

**Community**
Sites using Grails
Testimonials
Contribute
Issue Tracker
Source code
Plugins
Mailing Lists
Nabble Forums
IRC Webchat

Find us on:

*Last updated by maurice 3 years ago*

Edit View Info

## Searchable Plugin - Mapping

A searchable class mapping describes how the class instance appears in the index and the data is searched.

This includes things like:

- Which properties are searchable
- How a property is processed during indexing - whether it is "analyzed" or not, for example
- How a property influences the search - whether it has a "boost", for example
- How associated searchable classes are linked and/or embed one another's searchable data

When you declare

```
static searchable = true
```

the plugin maps the class with built-in conventions.

GRAILS

springsource
A division of vmware

Products | Services | Training | Community | Downloads | Plugins | Documentation

Sign in to edit and +1 items.

# Searchable Plugin - Mapping

**Get Started**
Installation
Quick Start
IDE Setup
Tutorials
Screencasts

**Reference**
Documentation
FAQs
Roadmap

**Community**
Sites using Grails
Testimonials
Contribute
Issue Tracker
Source code
Plugins
Mailing Lists
Nabble Forums
IRC Webchat

Find us on:

Upload Image | Preview | Save | Cancel | View Info

```
h1. Searchable Plugin - Mapping

A searchable class mapping describes how the class instance appears in the index and the data is searched.

This includes things like:

* Which properties are searchable
* How a property is processed during indexing - whether it is "analyzed" or not, for example
* How a property influences the search - whether it has a "boost", for example
* How associated searchable classes are linked and/or embed one another's searchable data

When you declare

{code}
static searchable = true
{code}

the plugin maps the class with
built-in [conventions|Searchable Plugin - Mapping - Conventions].

You can override these conventions in a number of ways:

h2. The Mapping DSL

The [mapping DSL|Searchable Plugin - Mapping - Mapping DSL], is a bit like GORM's mapping DSL.

You can selectively override the built-in [conventions|Searchable Plugin - Mapping - Conventions] for specific
properties (or the class itself), and inherit the default behaviour for any properties you do no explicitly map.


h2. Native Compass Mappings
```

GRAILS

springsource
A division of vmware

Products | Services | Training | Community | Downloads | Plugins | Documentation

documentation

Welcome **pledbrook**
Profile | Logout

# Search Results

**Get Started**
- Installation
- Quick Start
- IDE Setup
- Tutorials
- Screencasts

**Reference**
- Documentation
- FAQs
- Roadmap

**Community**
- Sites using Grails
- Testimonials
- Contribute
- Issue Tracker
- Source code
- Plugins
- Mailing Lists
- Nabble Forums
- IRC Webchat

Find us on:

Showing **1** - **10** of **412** results for **documentation**

## Plugin > New Grails Documentation (2.x docs in 1.x)

This plugin is a backport of the additional functionality offered in the grails doc command in ...

## Wiki page > Jasper Plugin

documentation moved to http://grails.org/plugin/jasper

## Plugin > Grails Runtime Docs

An API Documentation plug-in for Grails Projects that also displays the dynamic methods and ...

## Wiki page > Korean QuickLinks

???? | ?? | ??? ??? | ?????

GRAILS

springsource
A division of vmware

Products | Services | Training | Community | Downloads | Plugins | Documentation

Sign in to edit and +1 items.

# Plugins

SEARCH or browse tags

Welcome to the Grails plugin portal. The place where you can find information about the latest plugins available for the Grails framework.

- All
- Featured
- Top Installed
- Highest Voted
- Recently Updated
- Newest
- Supported

Want to create a plugin?

If you are interested in creating and distributing a plugin in the Grails central repository, take a look at this

## Auto test plugin

(5)

TAGS: testing
GRAILS VERSION: 2.0.0 > *
CURRENT RELEASE: 1.0.1

Source Docs

MORE INFO Download

## Cloud Foundry Integration

supported by SpringSource

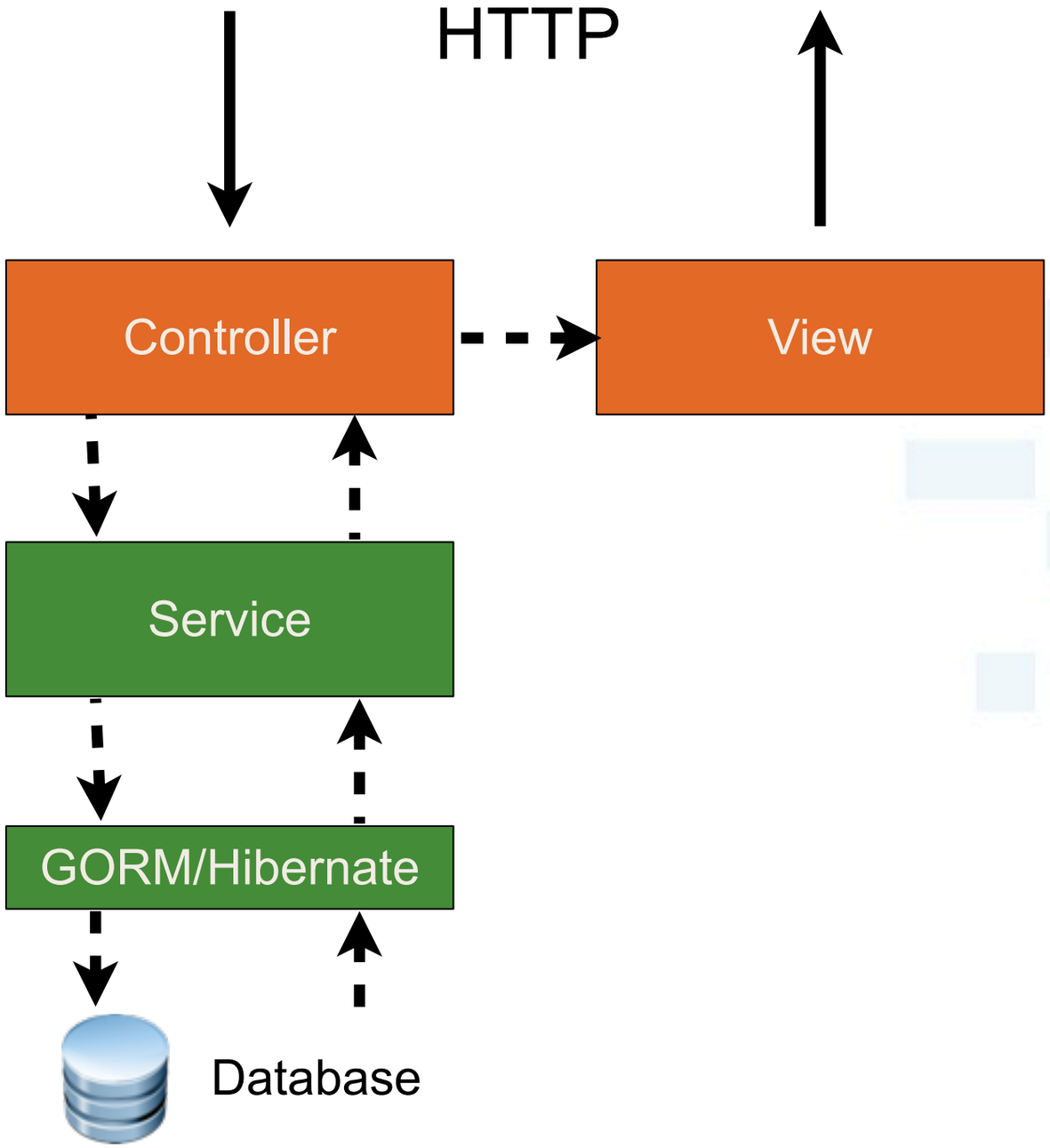
(24)

TAGS:
GRAILS VERSION: 1.3.3 > *
CURRENT RELEASE: 1.2.1

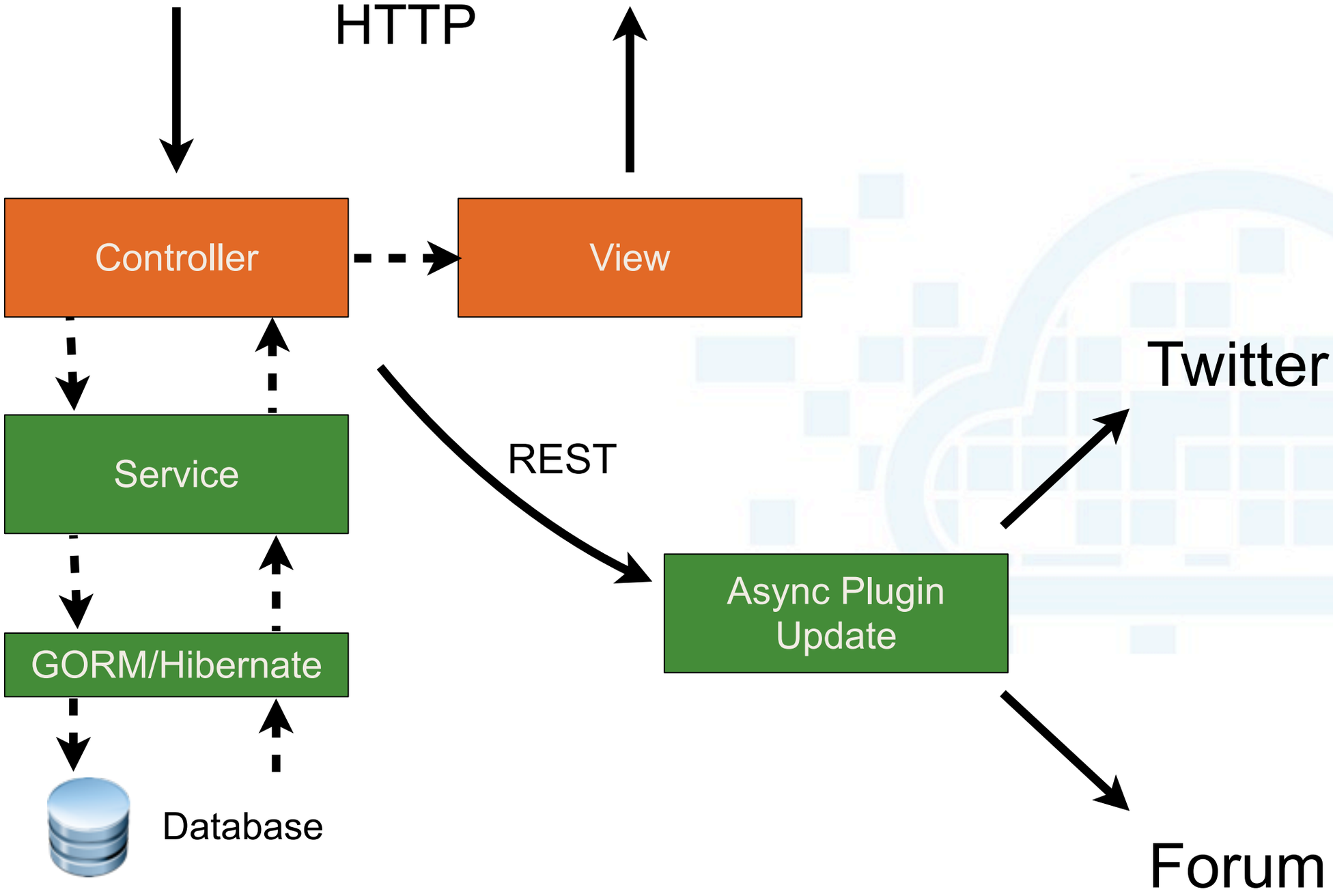
Source Docs Issues

MORE INFO Download

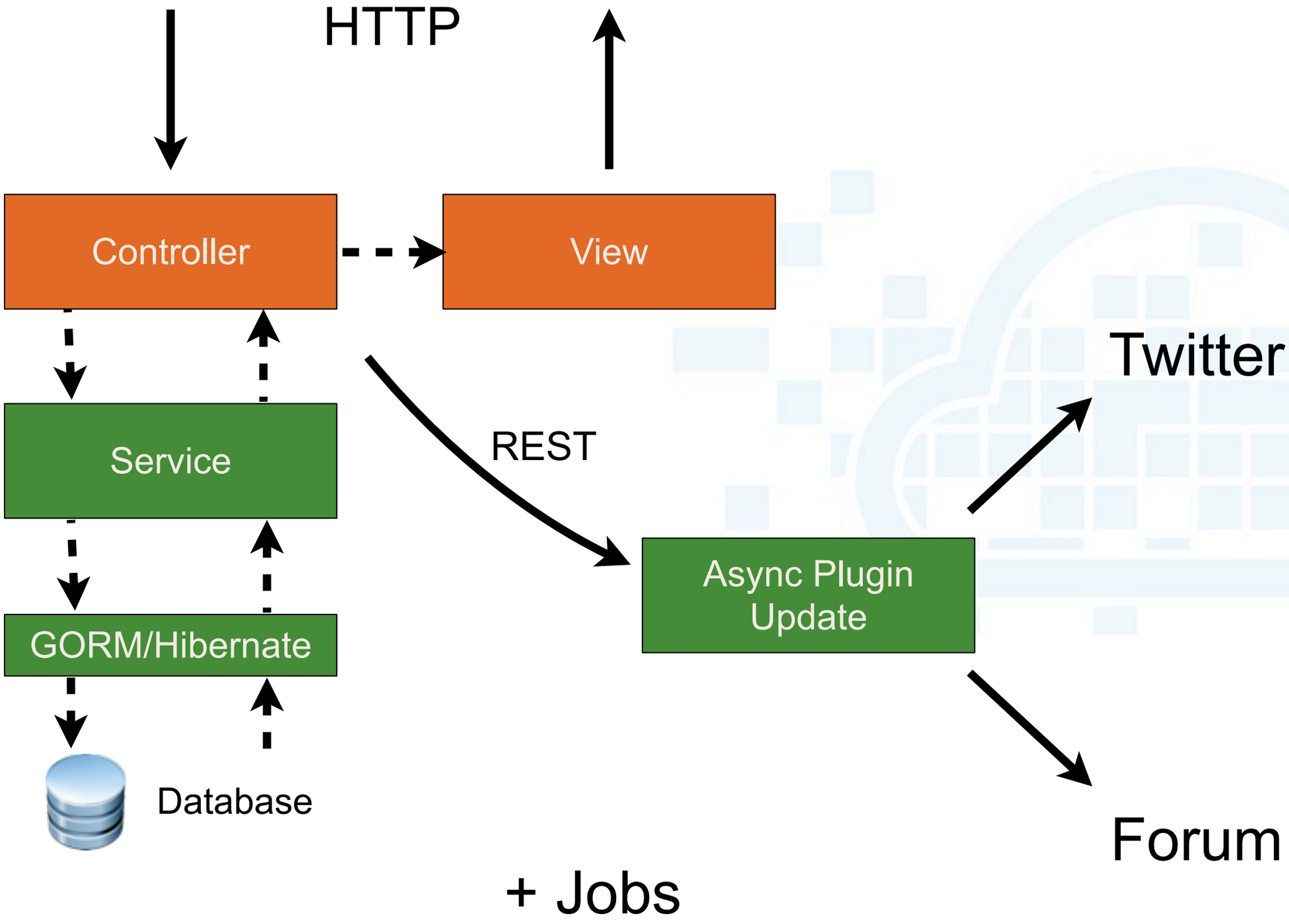
# Architecture



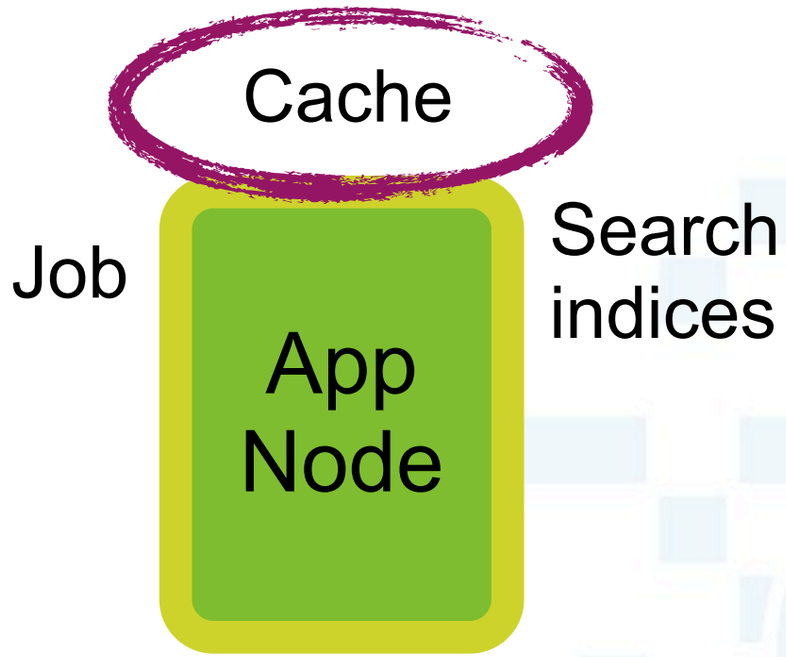
# Architecture



# Architecture

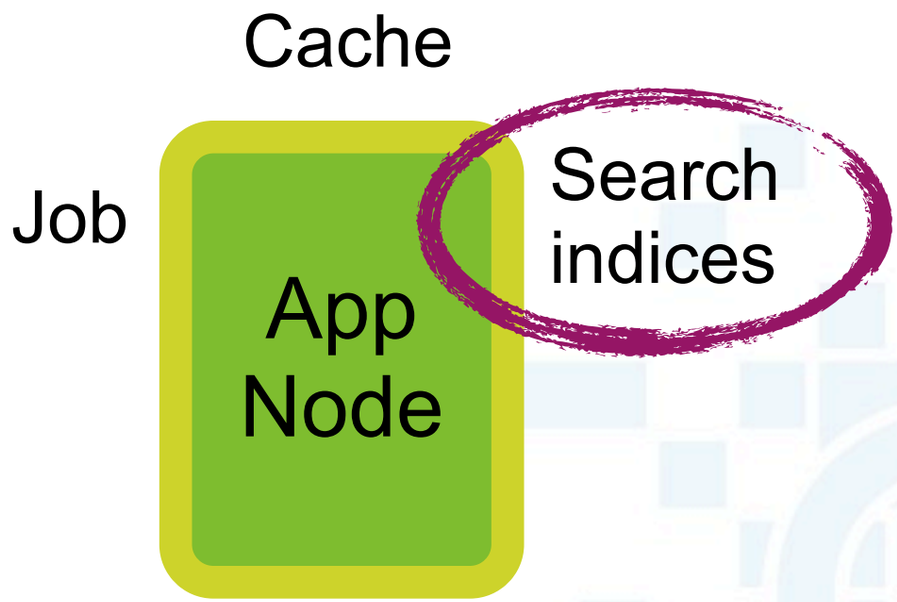


# Single instance



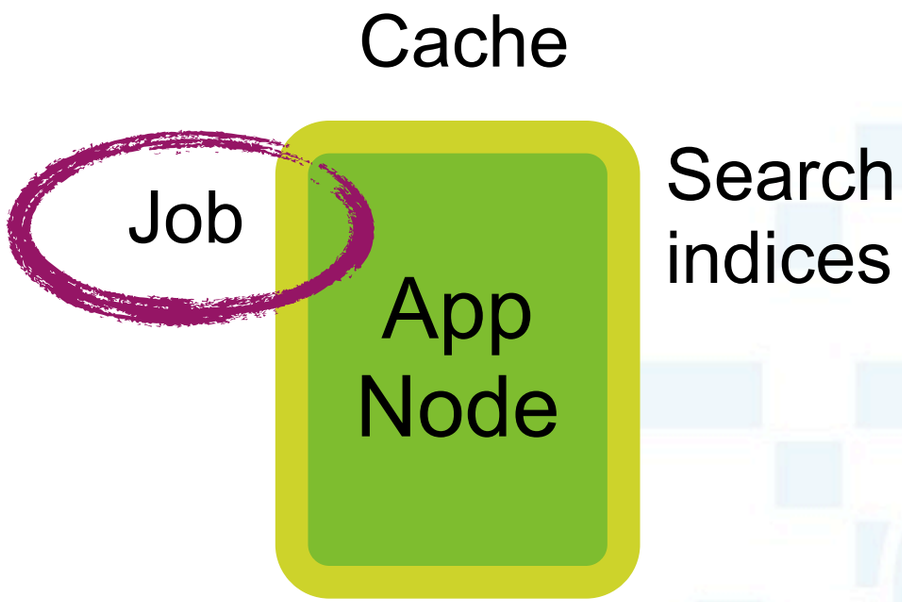

Local, embedded ehcache

# Single instance



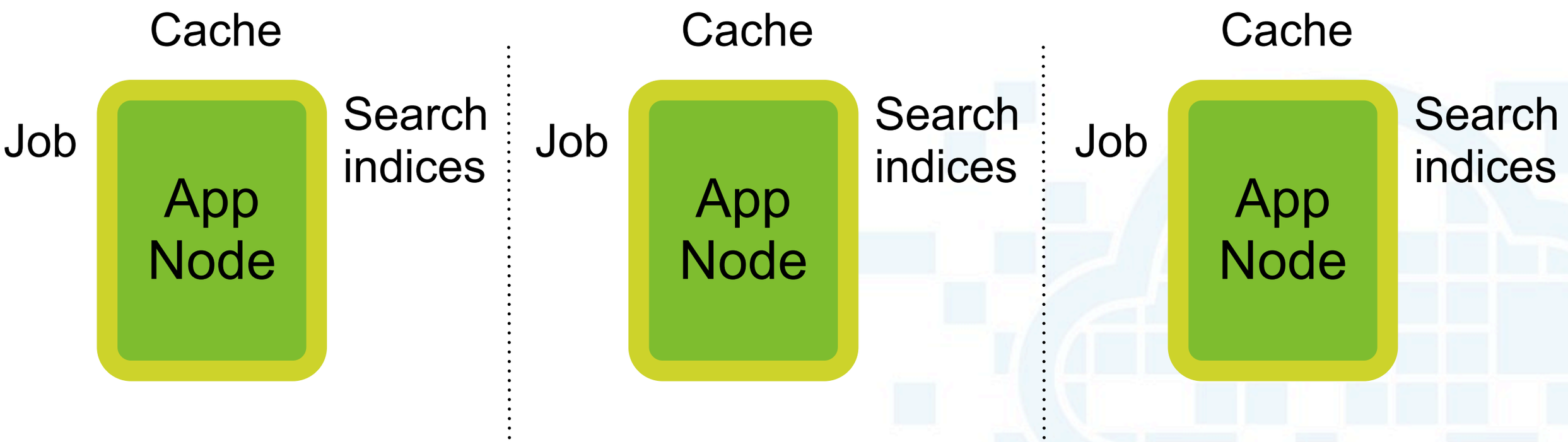

Local Lucene indexes

# Single instance

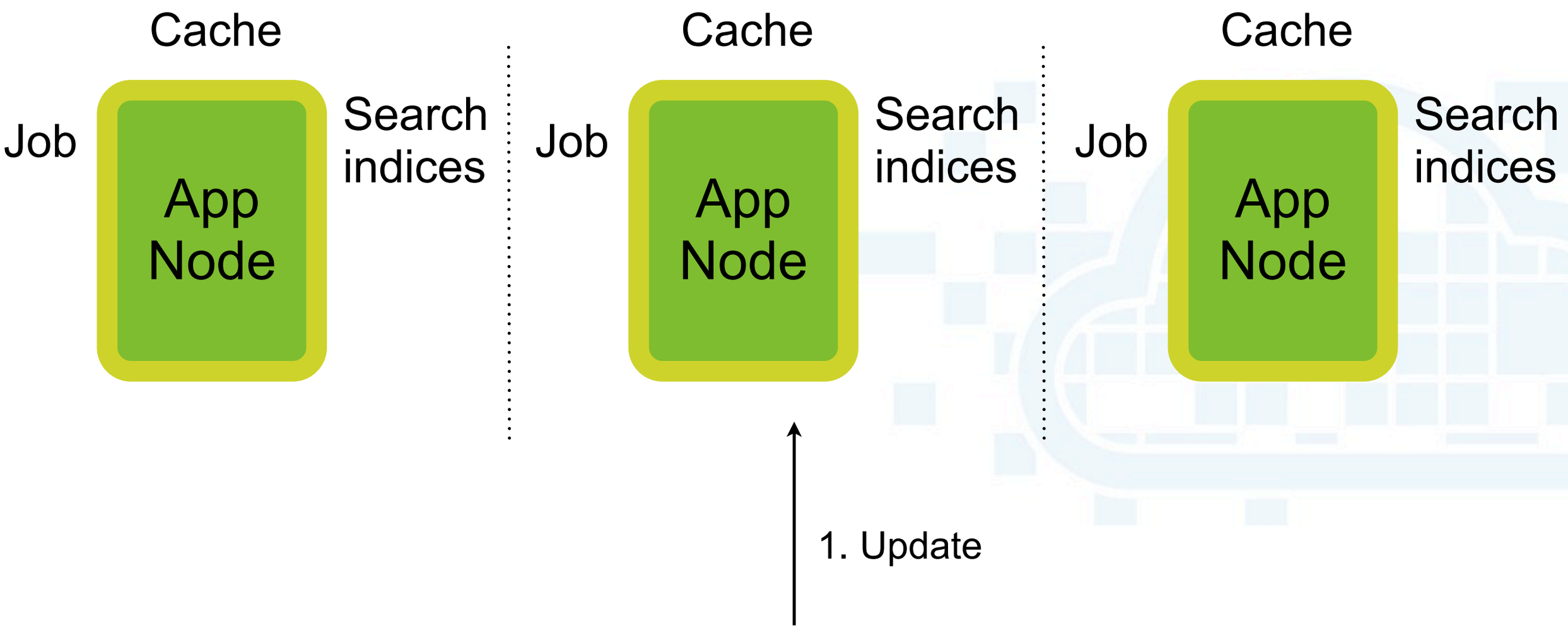



Single job of each type

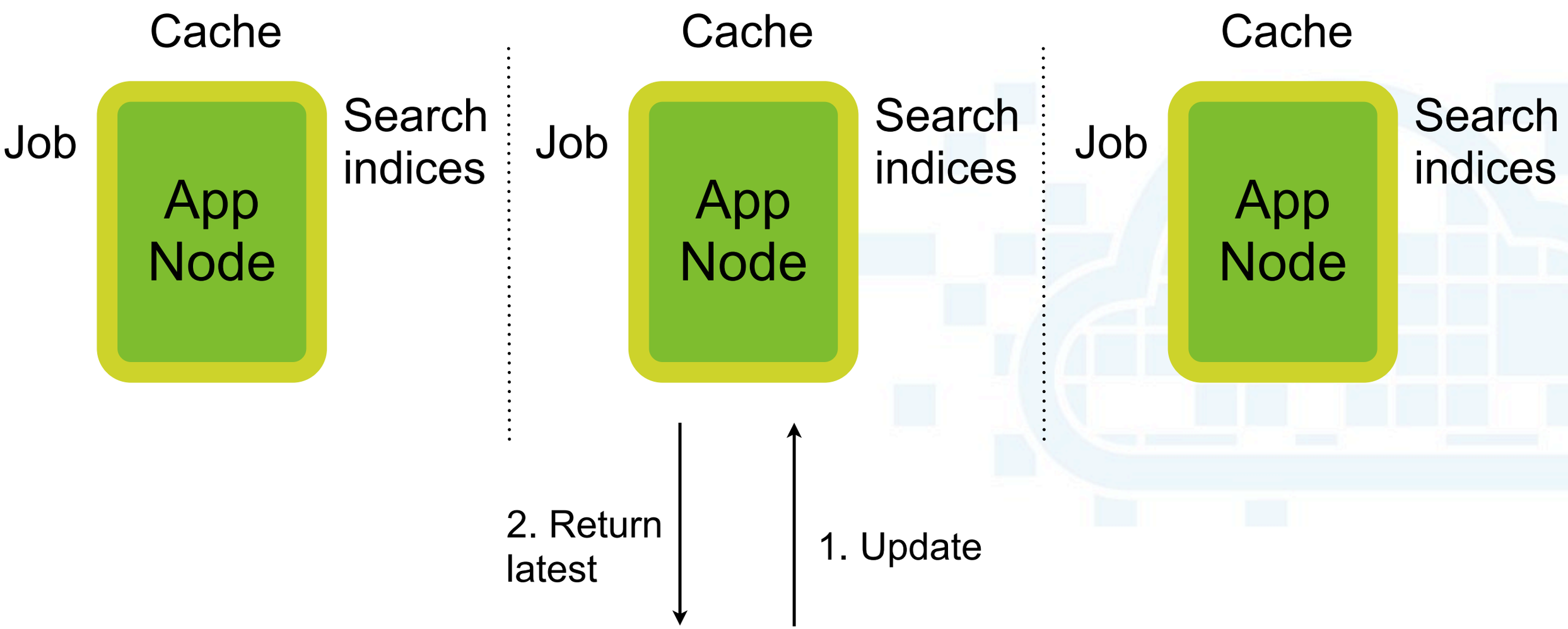
# Multi-instance



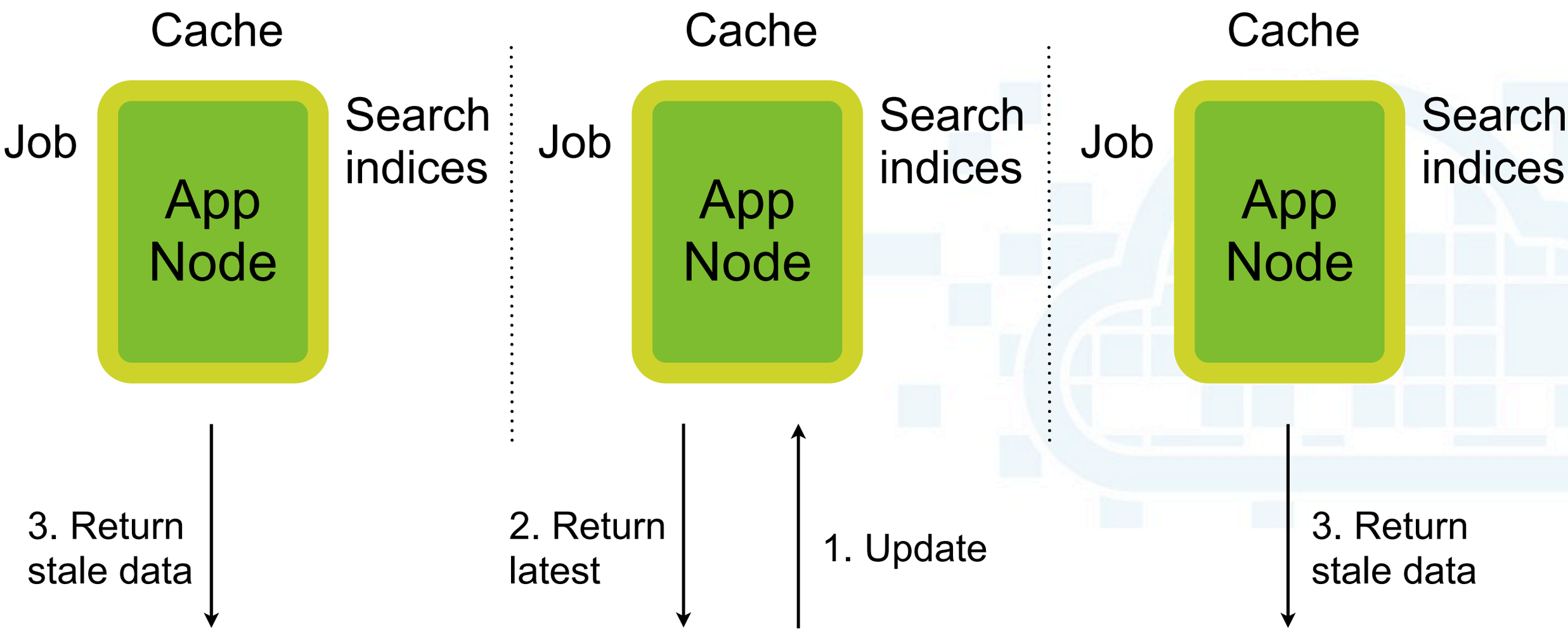
# Multi-instance



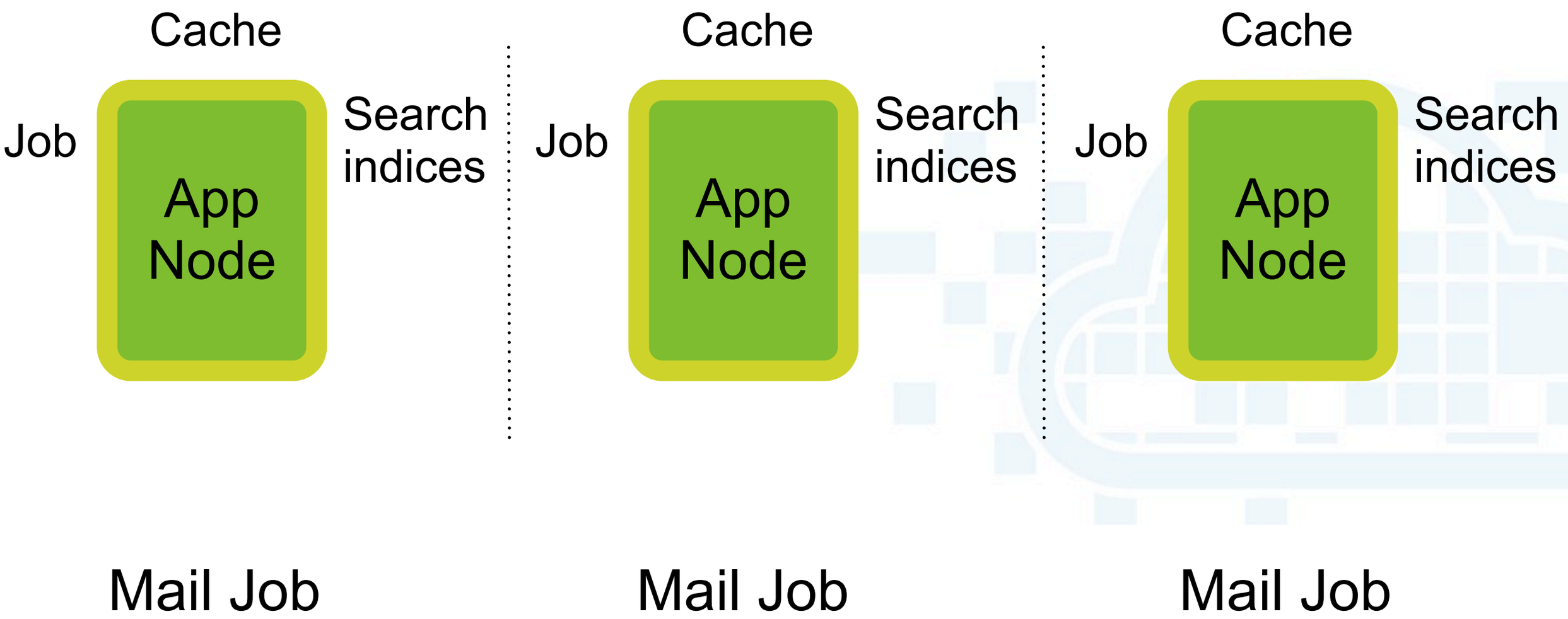
# Multi-instance



# Multi-instance



# Multi-instance




Jobs are executed three times?

# Caching

Caching shared data

=

Shared cache

# Ehcache

Simple & effective

Distributed mode

But...

# Ehcache

Simple & effective

Distributed mode

But...

No multicast on Cloud Foundry!

# So, what do we use?


- Key-value store
- In-memory with persistence (fast)
- Pub-sub
- Single service for all app instances

# Grails Cache plugin

- Based on Spring 3.1 cache abstraction
  - @Cacheable
  - @CacheEvict
- Ehcache & Redis providers
  - Spring Data RedisCacheManager
- <cache:block>
- <cache:render template=”...”>

```groovy
package org.grails.auth

import grails.plugin.cache.Cacheable
class UserService {
    static transactional = true

     * Returns a collection of permission strings that represent what the given
    @Cacheable(value="permissions", key="#user.id")
    def permissionsForUser(user) {
        return (user.permissions ?: []) + (user.roles*.permissions?.flatten() ?: []).unique()
    }

     * Changes the permissions for a user.
    @CacheEvict(value="permissions", key="#user.id")
    void updateUserPemissions(user, permissions) {
        // Take the simple approach: clear the list and re-add all declared permissions.
        if (user.permissions == null) {
            user.permissions = permissions
        }
        else {
            user.permissions.clear()
            user.permissions.addAll permissions
        }
    }
```

```groovy
package org.grails.auth

import grails.plugin.cache.Cacheable
class UserService {
    static transactional = true

    /**
     * Returns a collection of permission strings that represent what the given
     */
    @Cacheable(value="permissions", key="#user.id")
    def permissionsForUser(user) {
        return (user.permissions ?: []) + (user.roles*.permissions?.flatten() ?: []).unique()
    }

    /**
     * Changes the permissions for a user.
     */
    @CacheEvict(value="permissions", key="#user.id")
    void updateUserPemissions(user, permissions) {
        // Take the simple approach: clear the list and re-add all declared permissions.
        if (user.permissions == null) {
            user.permissions = permissions
        }
        else {
            user.permissions.clear()
            user.permissions.addAll permissions
        }
    }
}
```

```groovy
package org.grails.auth

import grails.plugin.cache.Cacheable
class UserService {
    static transactional = true

     * Returns a collection of permission strings that represent what the given
    @Cacheable(value="permissions", key="#user.id")
    def permissionsForUser(user) {
        return (user.permissions ?: []) + (user.roles*.permissions?.flatten() ?: []).unique()
    }

     * Changes the permissions for a user.
    @CacheEvict(value="permissions", key="#user.id")
    void updateUserPemissions(user, permissions) {
        // Take the simple approach: clear the list and re-add all declared permissions.
        if (user.permissions == null) {
            user.permissions = permissions
        }
        else {
            user.permissions.clear()
            user.permissions.addAll permissions
        }
    }
}
```

# HTTP session

Sticky sessions!

# HTTP session

Or consider Redis for distributed session store

# Jobs

- @Scheduled, Quartz, etc.
- Which node runs a job?

# Jobs

- @Scheduled, Quartz, etc.
- Which node runs a job?

Local data



Do nothing

# Jobs

- @Scheduled, Quartz, etc.
- Which node runs a job?

Local data



Do nothing

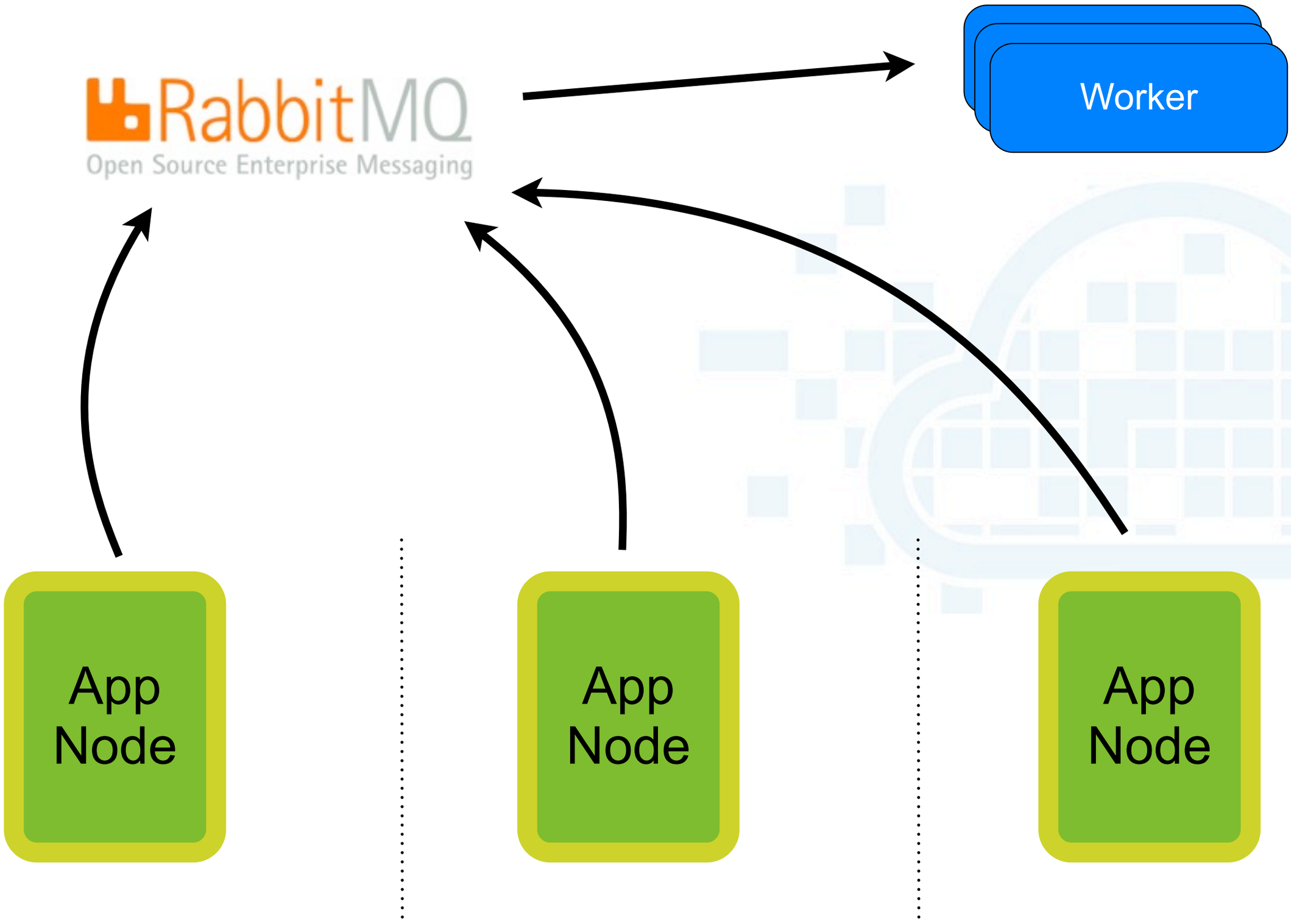
Shared data



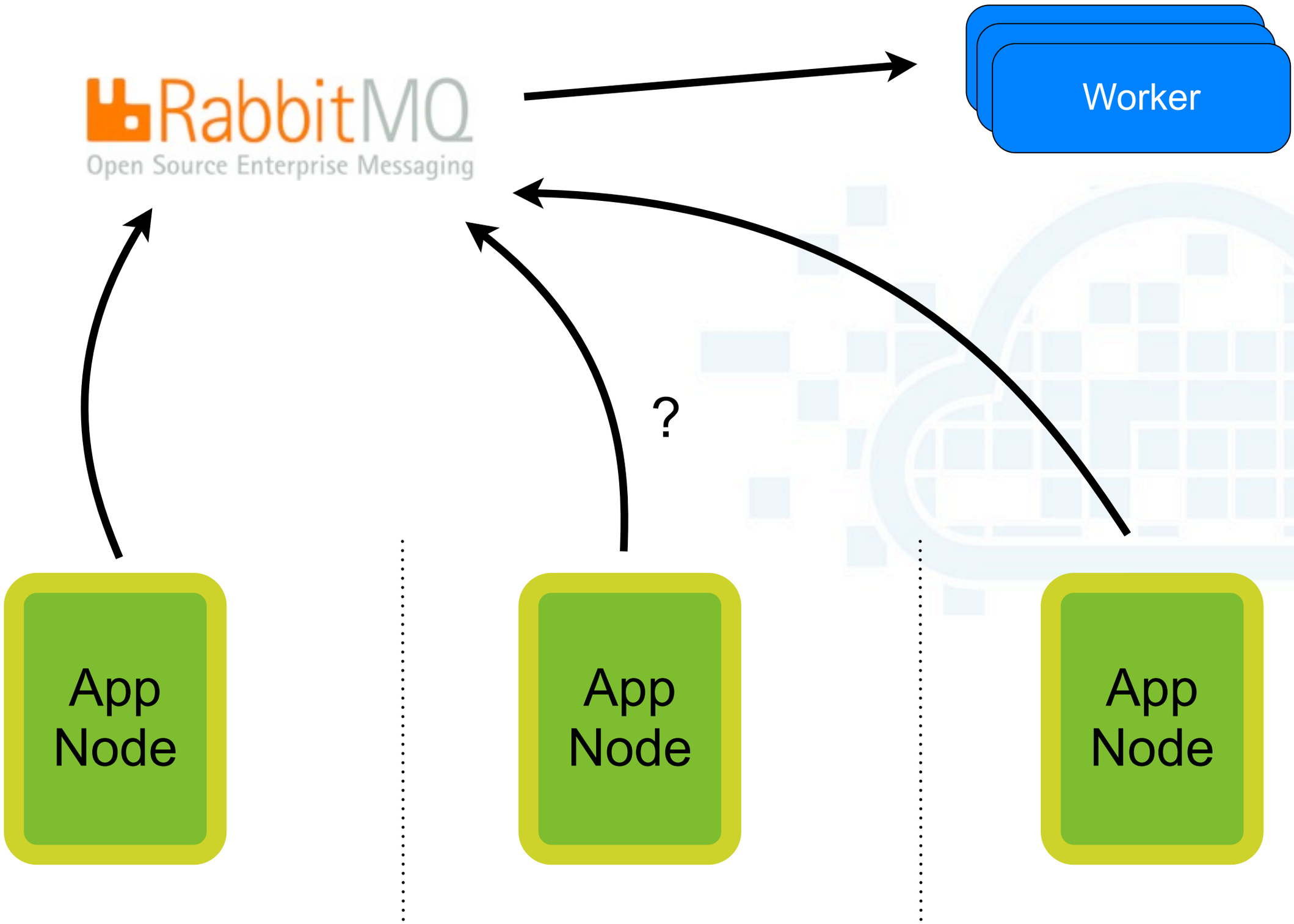
Quartz JDBC Store

Worker

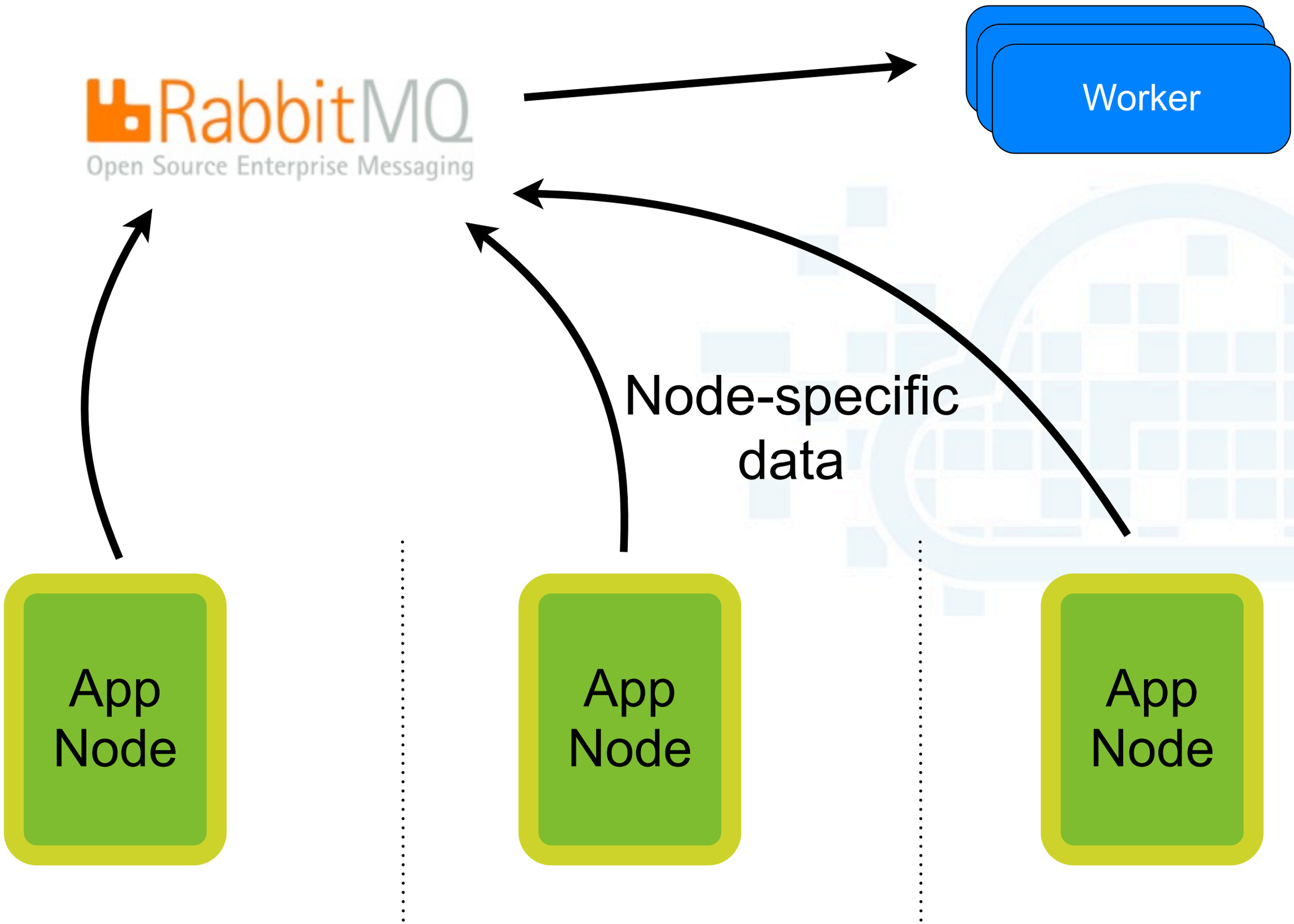
# Distribution with workers



# Distribution with workers



# Distribution with workers



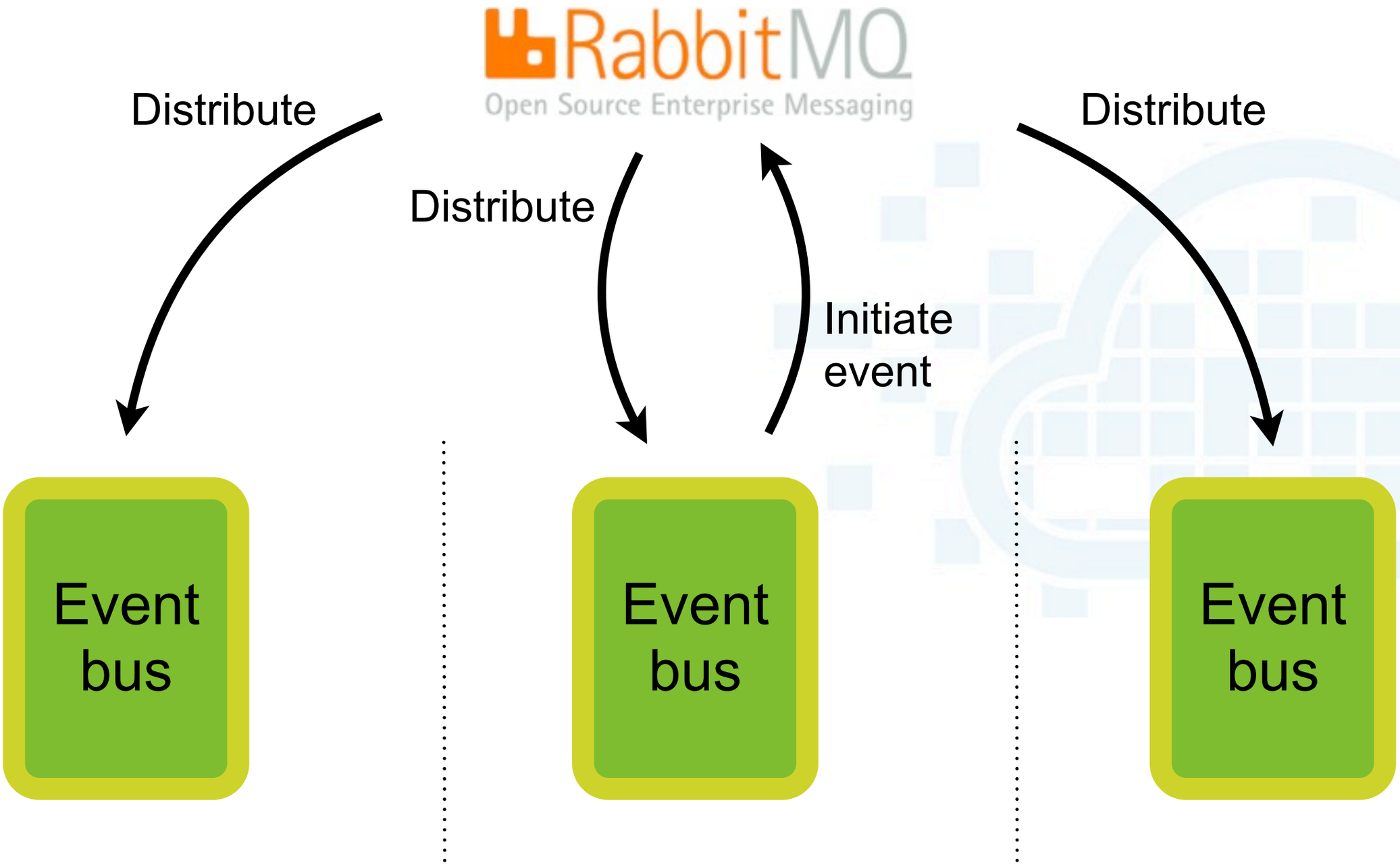
# Jobs

| | Pros | Cons |
|---|---|---|
| Quartz | Efficient in-JVM processing<br>Tried & tested<br>Relatively simple | Extra burden on web server |
| Workers | Use right language for the job<br>Distribute the workload | Counts against your quota |

# Search

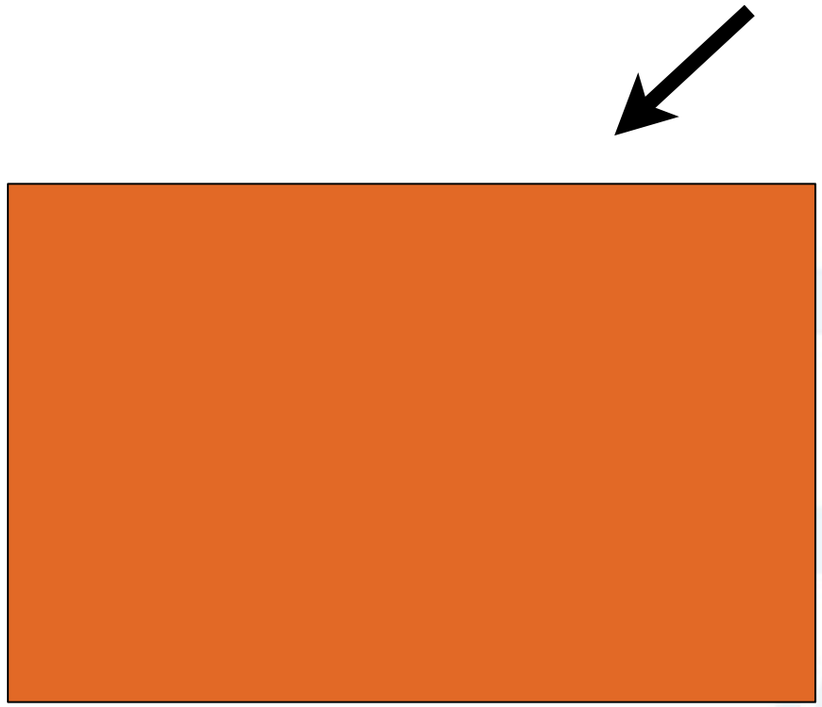
- Compass 2.0
  - Lucene indexes on local file system
  - Indexes database data
- > 1 node == stale indexes
- Search as a service
  - Solr
  - Elastic Search
  - Index Tank
  - Amazon Cloud Search
  - Not on Cloud Foundry (yet)
- Synchronise indexes across nodes!
  - Eventual consistency is good enough

# Synchronisation with an event bus



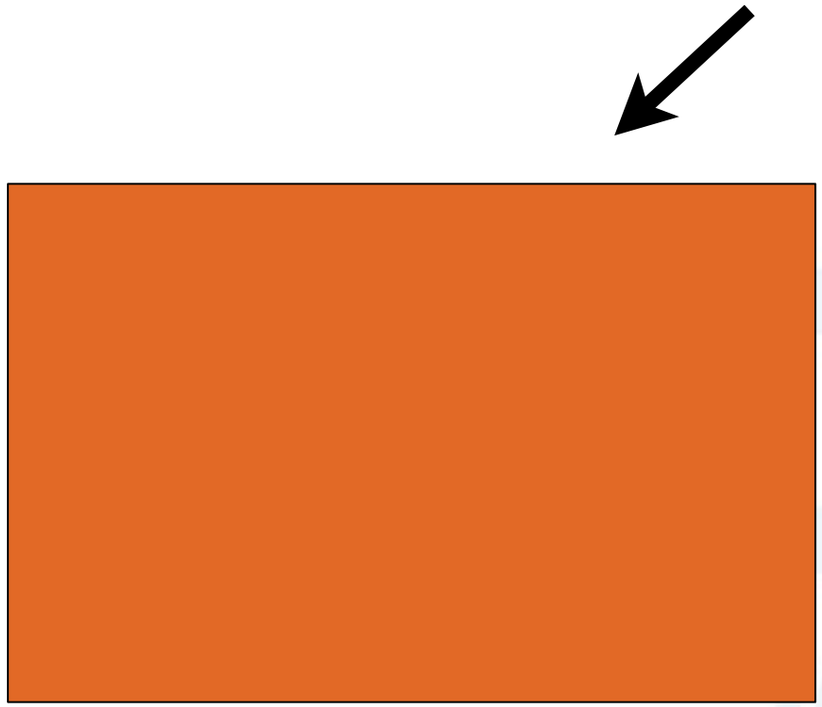
# Schrödinger's Cat

There's a cat in here!



# Schrödinger's Cat

There's a cat in here!



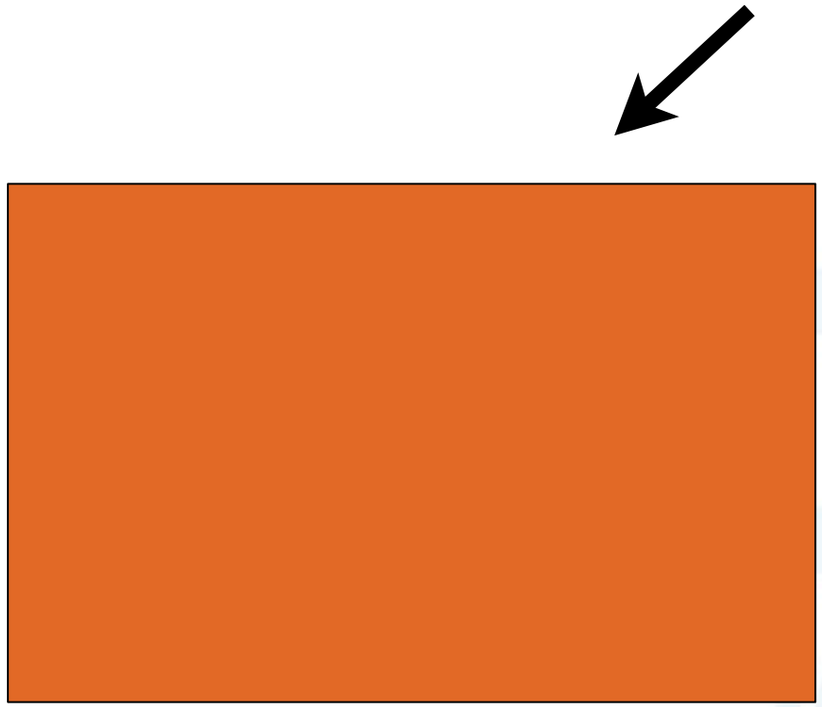
Is the cat alive or dead?

# Local filesystem

There's a file in here!

# Local filesystem

There's a file in here!

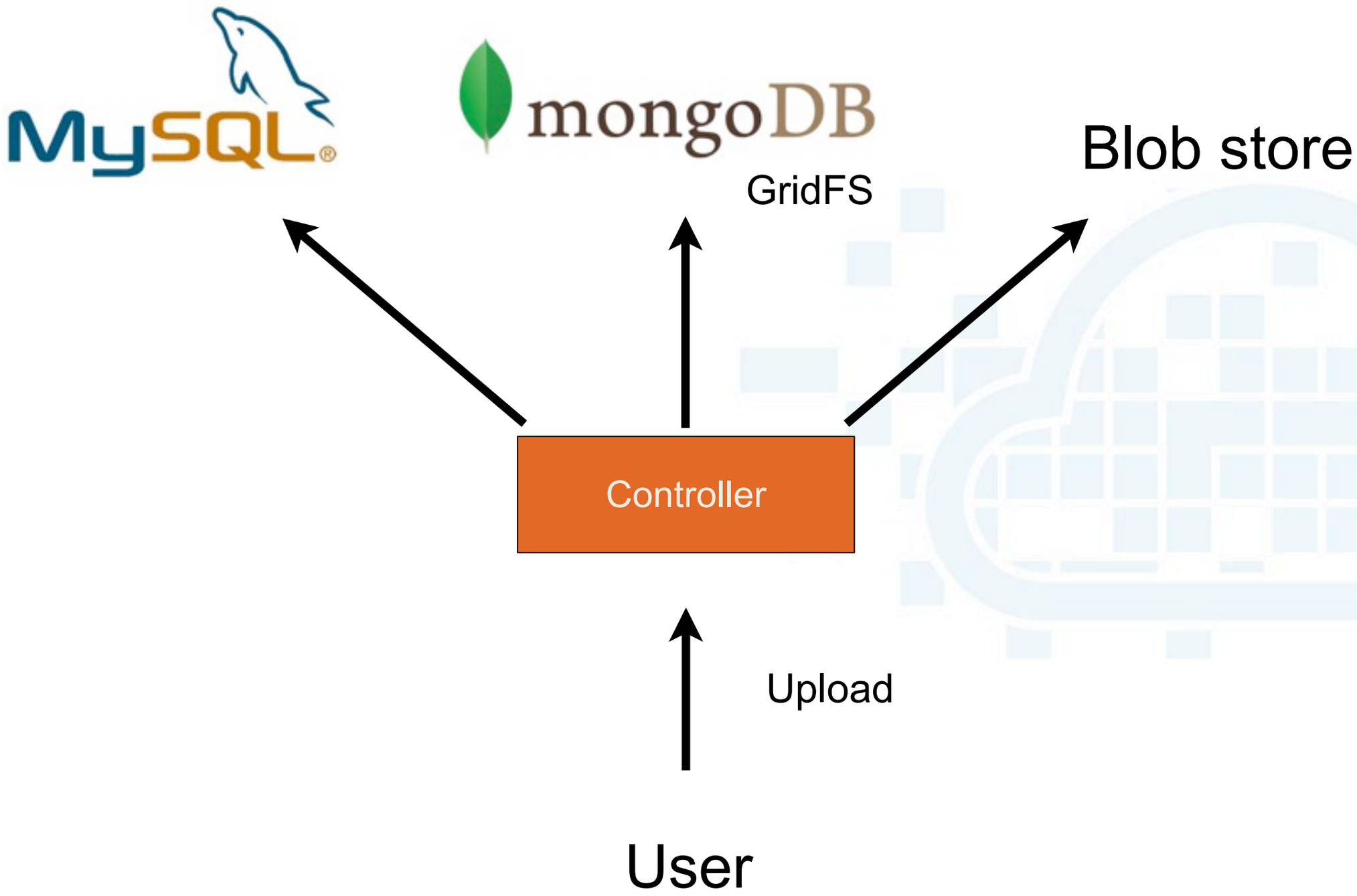


Does the file exist or not?

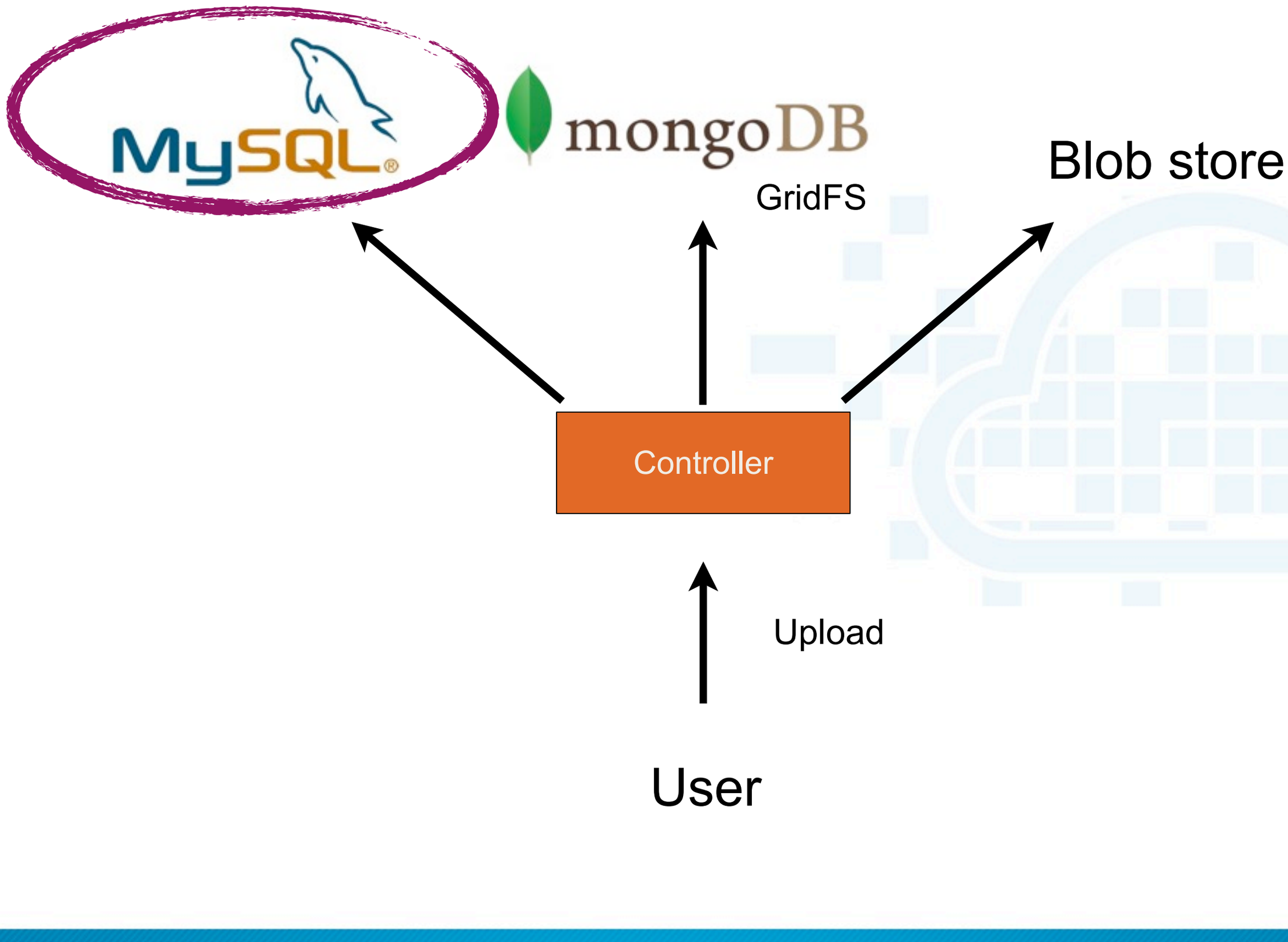
# Ephemeral file system

- VMs are created and destroyed
  - => application file system created and destroyed
  - You can write to the local file system...
  - ...but you will probably lose that data at some point
- Don’t store persistent data on local FS!
  - File uploads

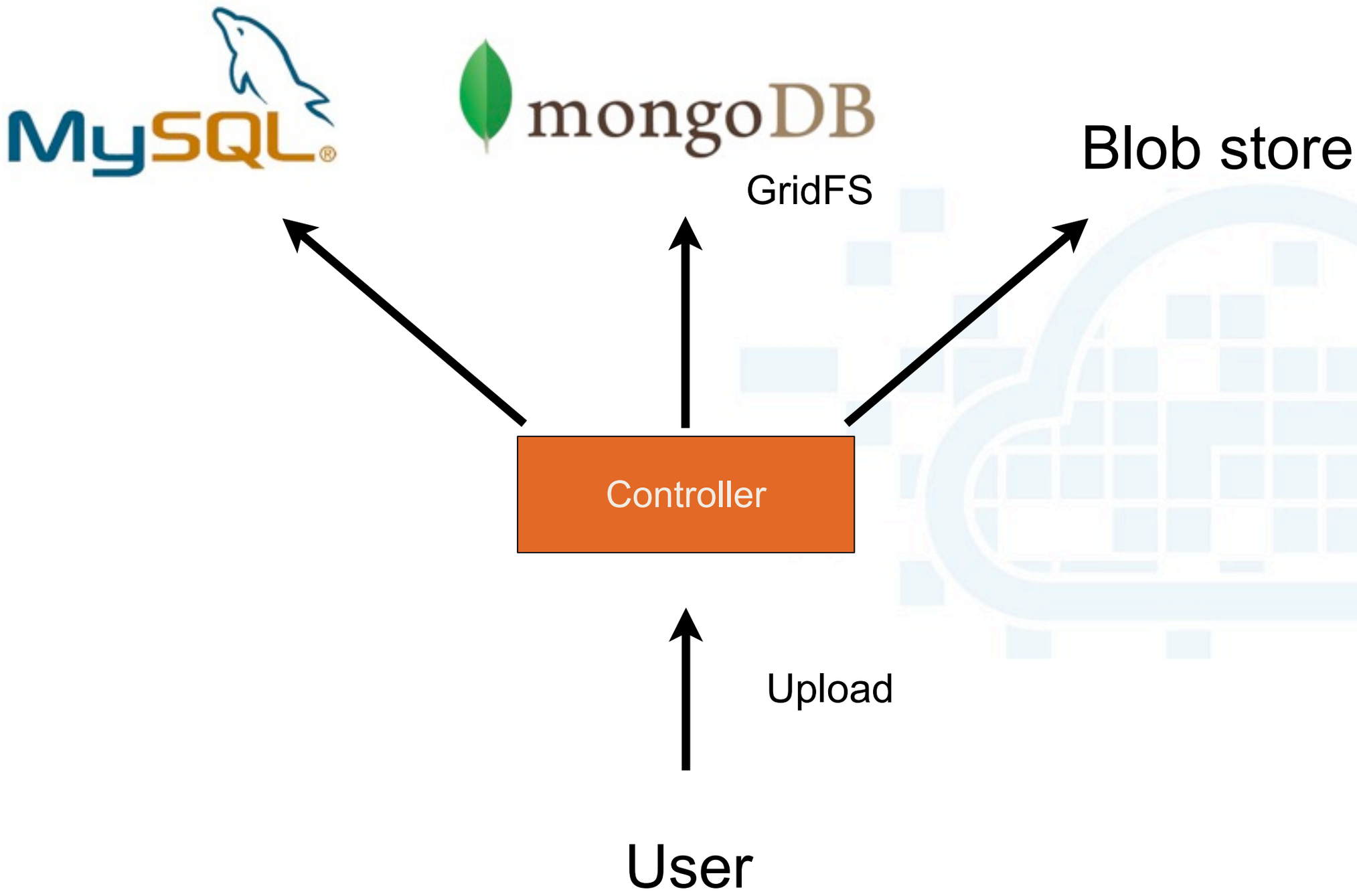
# Wiki images



# Wiki images



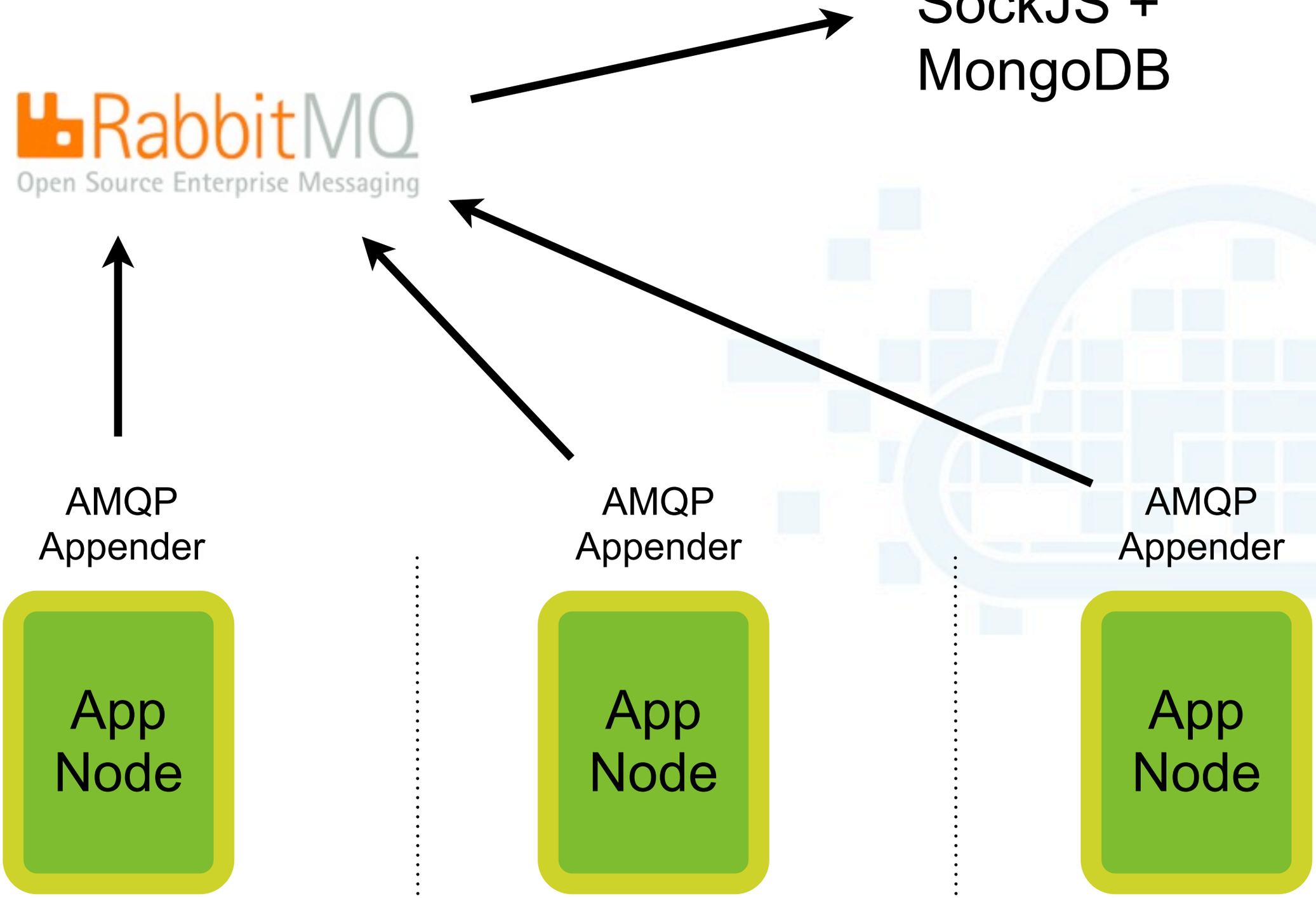
# Wiki images



# Logging

- View logs with `vmc logs`
- What if the application instance is restarted?
  - Logs are gone
- What about multiple application instances?
  - `vmc logs --instance 1` or `vmc logs --all`
  - but 30 days after deployment?

# Distributed logging



# Config

Package in the app?

Store in database?

Inject via environment variable?

# Dynamic config load

Config.groovy

```groovy
ConfigLoader.addEntries(loadJson(fetchJson()), this)
def fetchJson() { return System.getenv("GRAILS_APP_CONFIG") }
def loadJson(content) { return content ? grails.converters.JSON.parse(content) : [:] }
```

ConfigLoader.groovy

```groovy
class ConfigLoader {

    static void addEntries(Map data, obj = null) {
        data?.each { key, value ->
            if (value instanceof Map) {
                addEntries(value, obj.getProperty(key))
            }
            else obj.setProperty(key, value)
        }
    }
}
```

# Mail

- No SMTP - oh no!
- HTTP providers
  - SendGrid
  - Mailgun
  - Amazon Simple Email Service

# Easy HTTP Mail

resources.groovy

```
if (Environment.current.name == "cloud") {
    mailService(MailgunService) { bean ->
        bean.autowire = true
    }
}
```

MailerJob.groovy

```
mailService?.sendMail {
    title "Grails > ${myTitle}"
    from "wiki@grails.org"
    replyTo "wiki@grails.org"
    to email
    html text
}
```

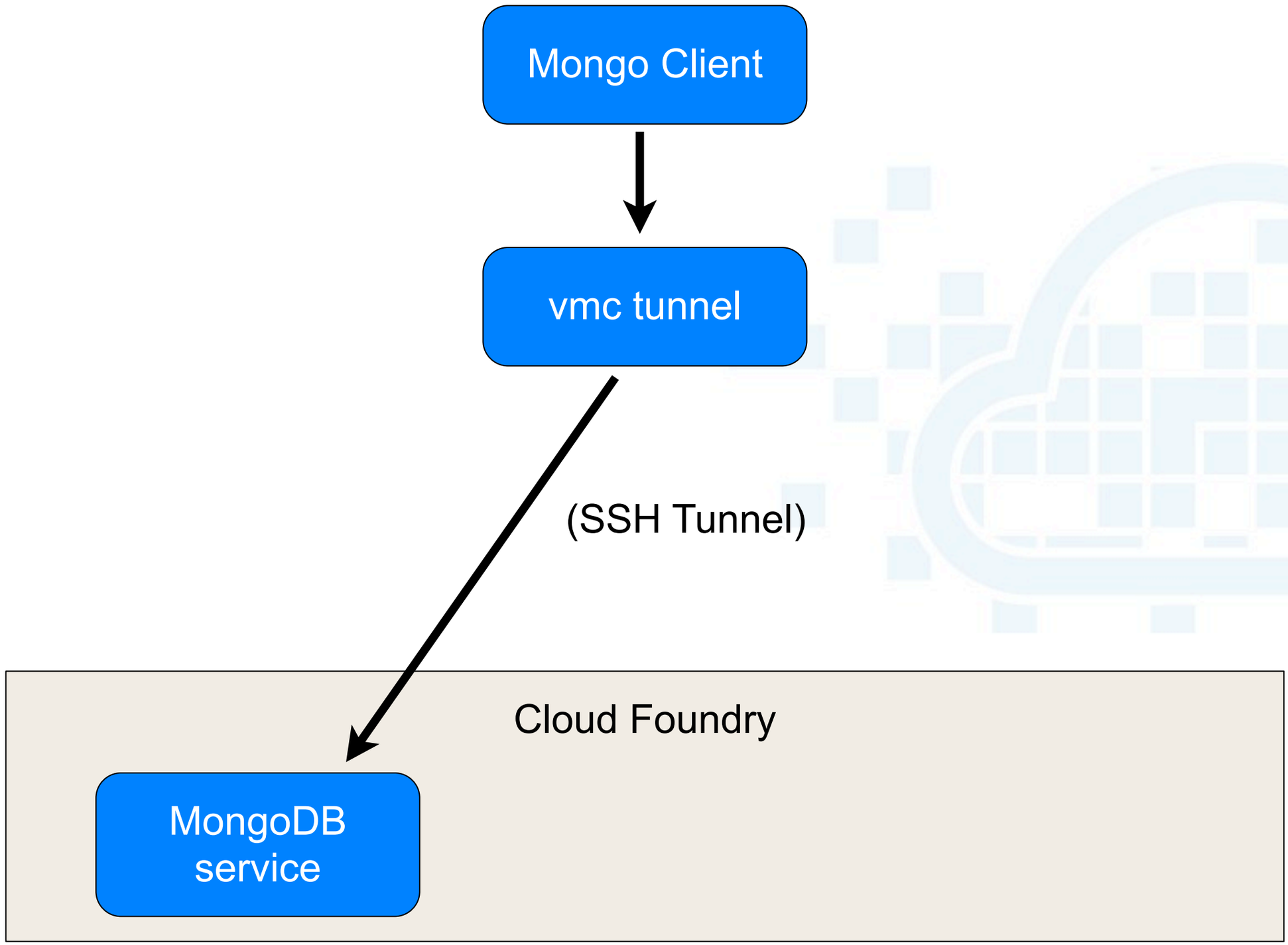
# Data

How do we load initial data?

Perform data migrations?

Back data up?

# Access your services



# Import/export

Import == mysql < data.sql

Export == mysqldump

# Application (WIP*)

Source https://github.com/grails-samples/grails-website

Live http://grails-website.cloudfoundry.com/

* Work in Progress

# Summary

- PaaS is the application platform for the Cloud era
- PaaS will change the way you write apps
  - Design for horizontal scalability
  - Account for ephemeral file system
  - Rich set of services
  - Go polyglot!
- Tools & libraries are important
  - Spring for the win!
- Can use Redis in place of RabbitMQ for some patterns

# What's next?

- Sign up - www.cloudfoundry.com
- Get the source code - www.cloudfoundry.org
- Download your Micro Cloud Foundry – my.cloudfoundry.com/micro
- Learn more on the Cloud Foundry blog - blog.cloudfoundry.com
- Follow us - @cloudfoundry

# Q&A