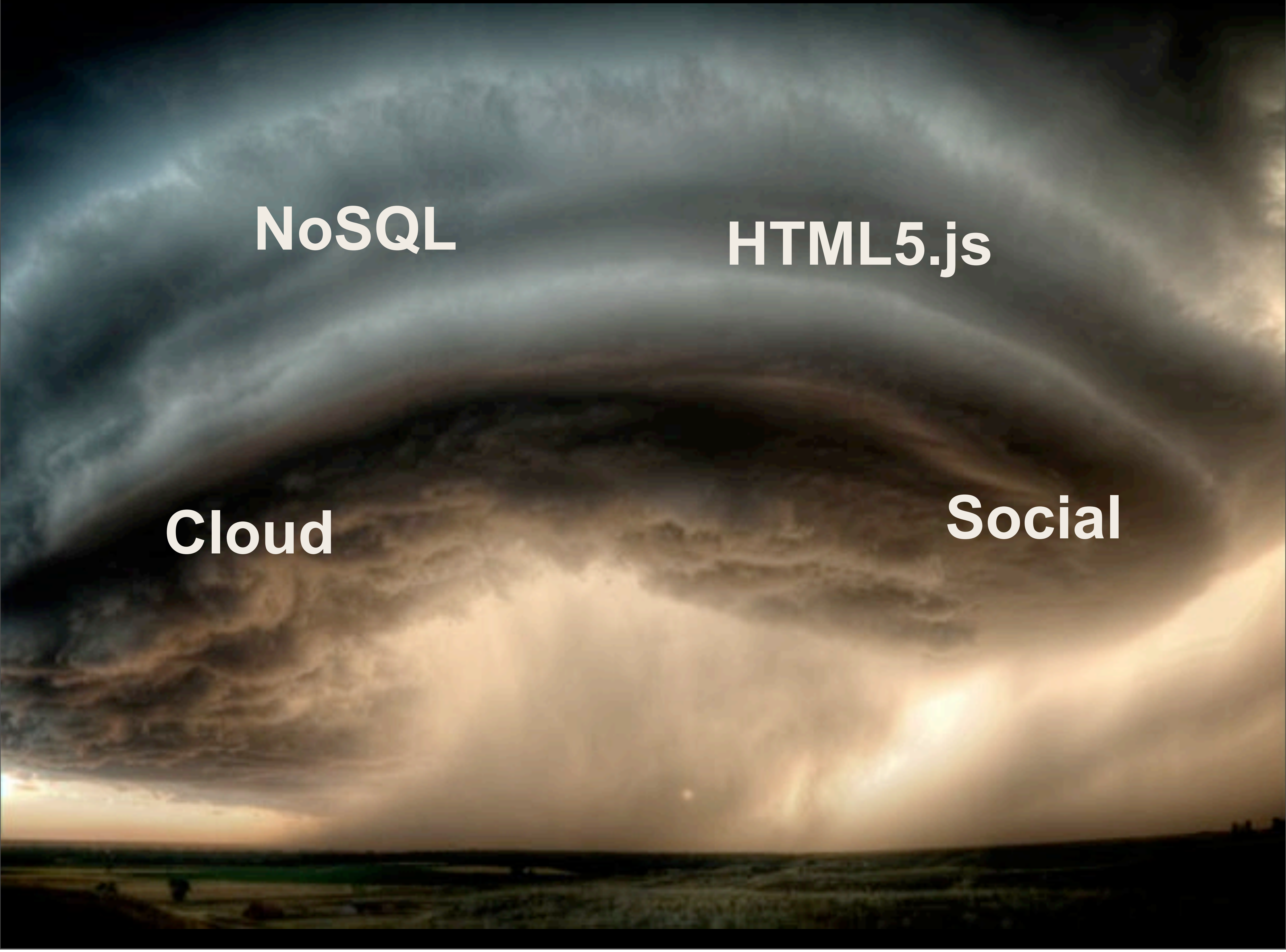
GR8Conf US 2013
July 21st-23rd
Minneapolis, MN

Grails and the World of Tomorrow

Peter Ledbrook

e: p.ledbrook@cacoethes.co.uk
t: @pledbrook



NoSQL

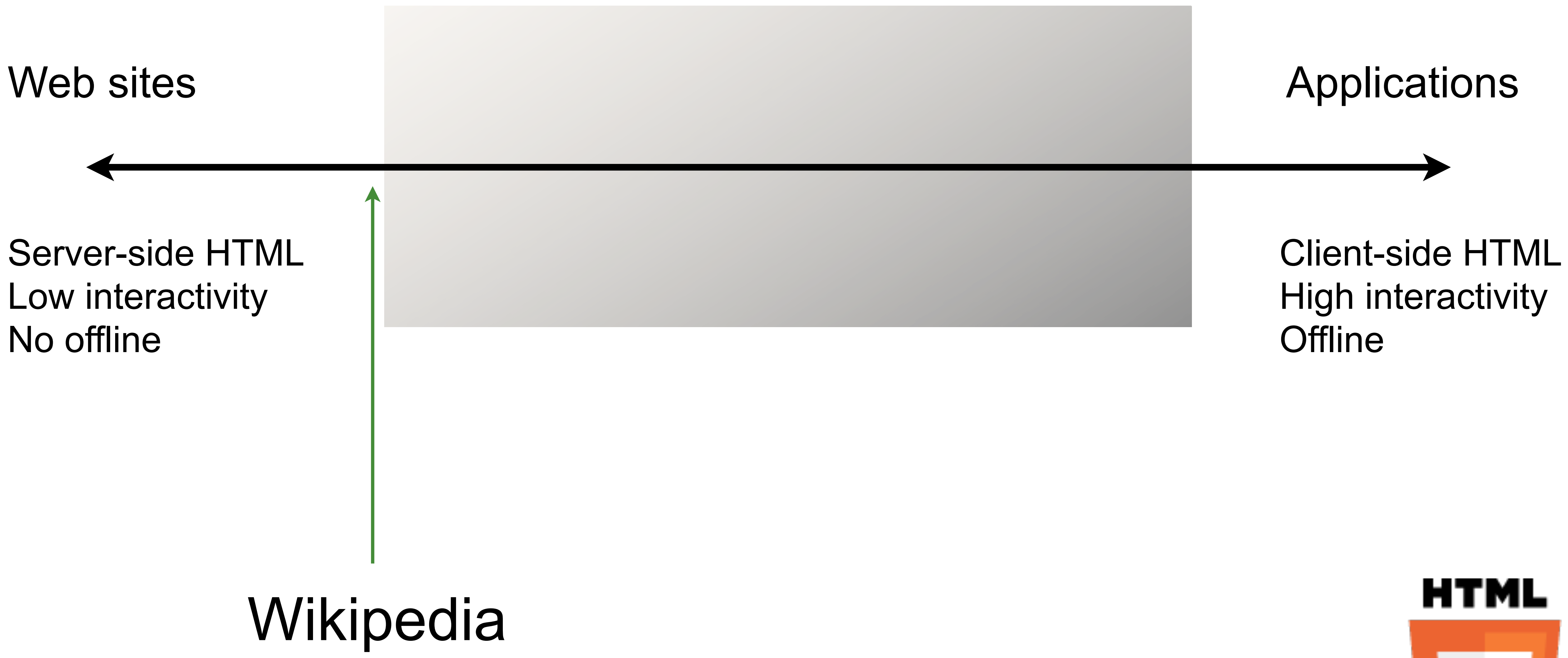
HTML5.js

Cloud

Social

Trends in web apps
Perfect storm of big changes

HTML5.js



Web sites

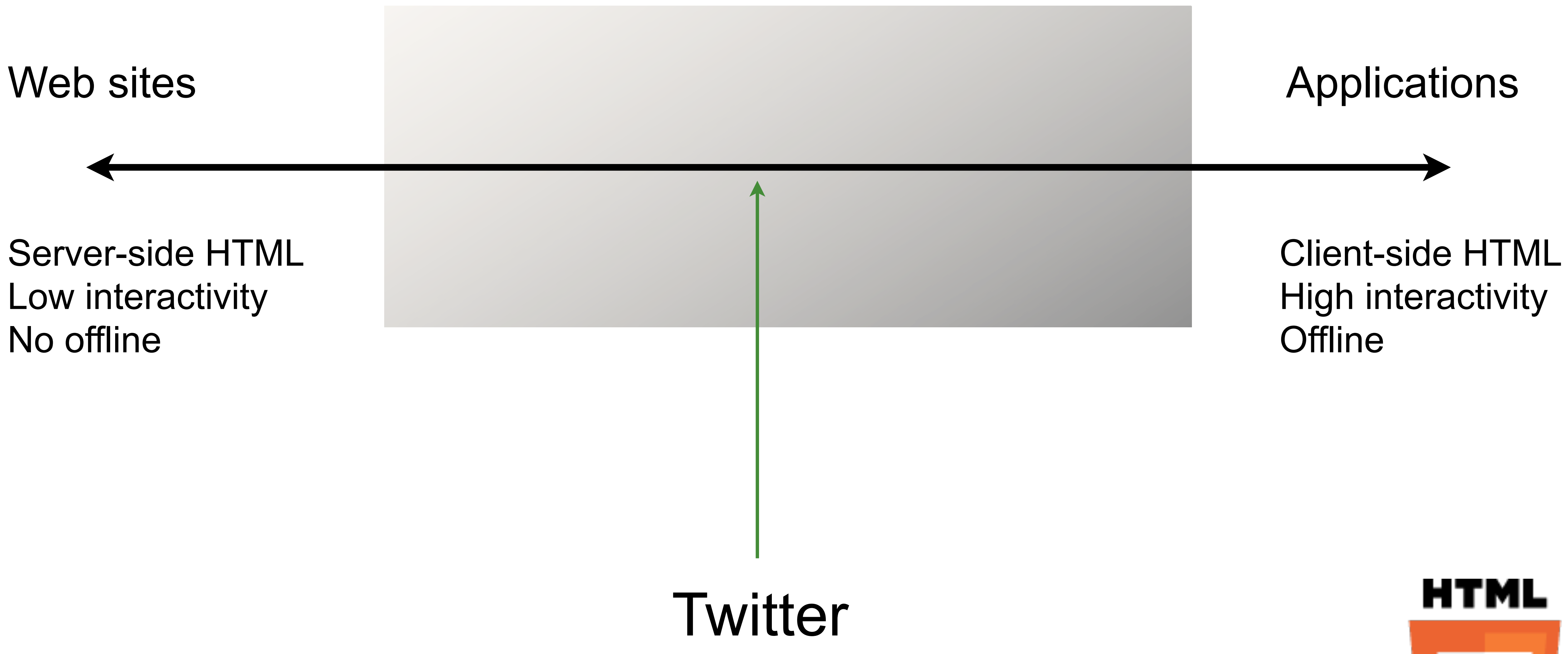
Applications

Server-side HTML
Low interactivity
No offline

Client-side HTML
High interactivity
Offline

Wikipedia

HTML5.js



Web sites

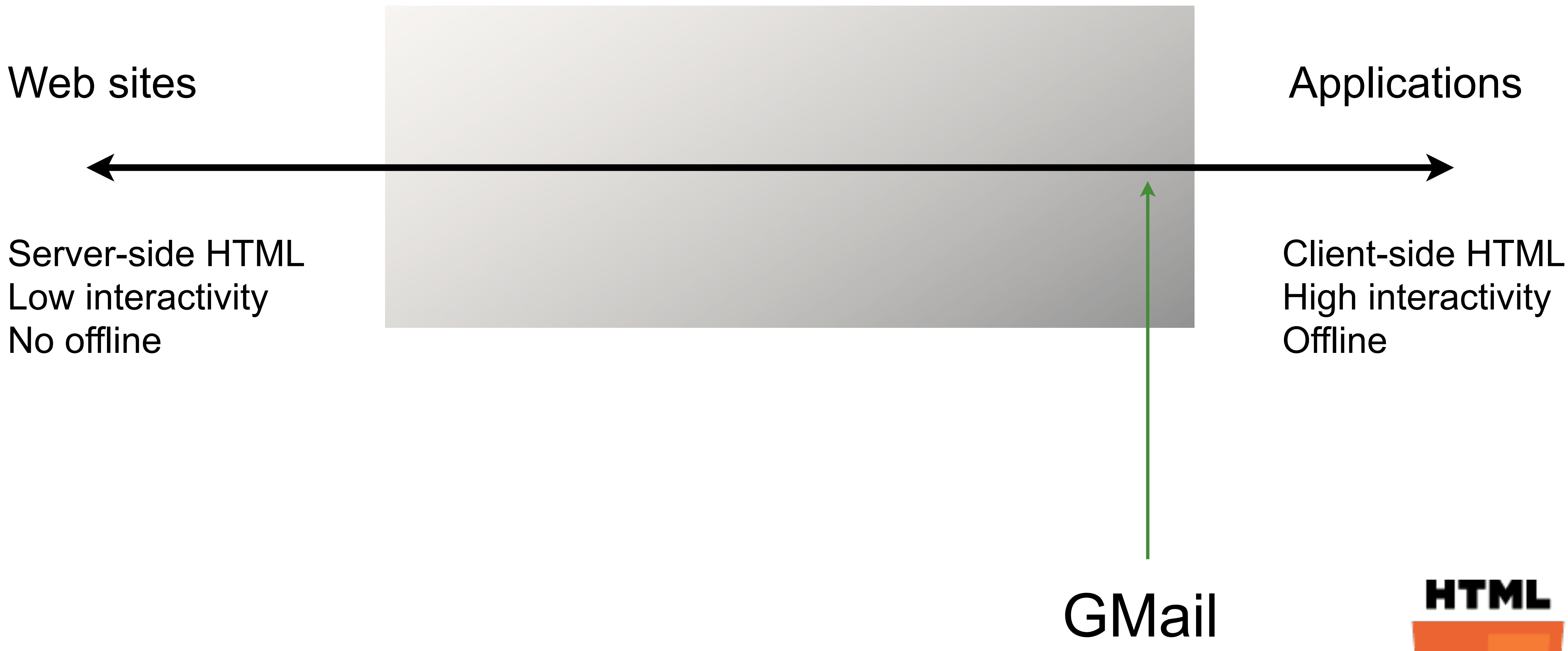
Applications

Server-side HTML
Low interactivity
No offline

Client-side HTML
High interactivity
Offline

Twitter

HTML5.js



Web sites

Server-side HTML
Low interactivity
No offline

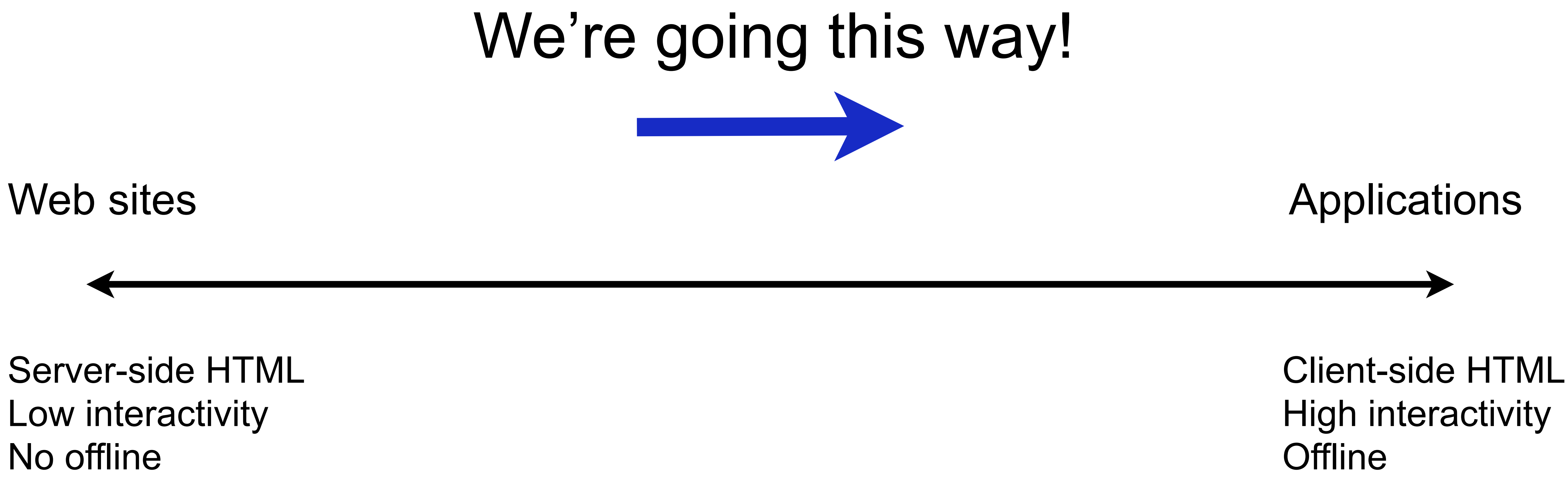
Applications

Client-side HTML
High interactivity
Offline

GMail

New Twitter

HTML5.js



We're going this way!

Web sites

Applications

Server-side HTML
Low interactivity
No offline

Client-side HTML
High interactivity
Offline

Example:

REST + JSON

AngularJS

Shift from low- to high- interactivity

HTML5.js

Websockets == persistent connections

Scalability more of an issue

Google I/O 2012

400 million

Android activations to date

Apple WWDC 2012

365 million

iOS devices sold to date

Let’s not forget Firefox OS

Is this the end for server-side templates?

Native apps
GSP == server-side ... does it have a place?
Let's not pronounce its death yet:
Rails 4 has large investment in fast, server-side apps
Does Wikipedia really need a highly dynamic site?
Fancy sites can be more frustrating than helpful
Low bandwidth effects

NoSQL / Big Data

Data the way you want it!

mongoDB
Neo4j
the graph database
redis
Cassandra

Not all data relational
Data structure is important!
Lots of choice
Each has its own advantages - what data are you storing? How are you querying it? Is performance the main concern?
How do we make best use of them?
Are they reliable enough?

NoSQL / Big Data

Now we even have to worry about how we store our data!

Not all data relational
Data structure is important!
Lots of choice
Each has its own advantages - what data are you storing? How are you querying it? Is performance the main concern?
How do we make best use of them?
Are they reliable enough?

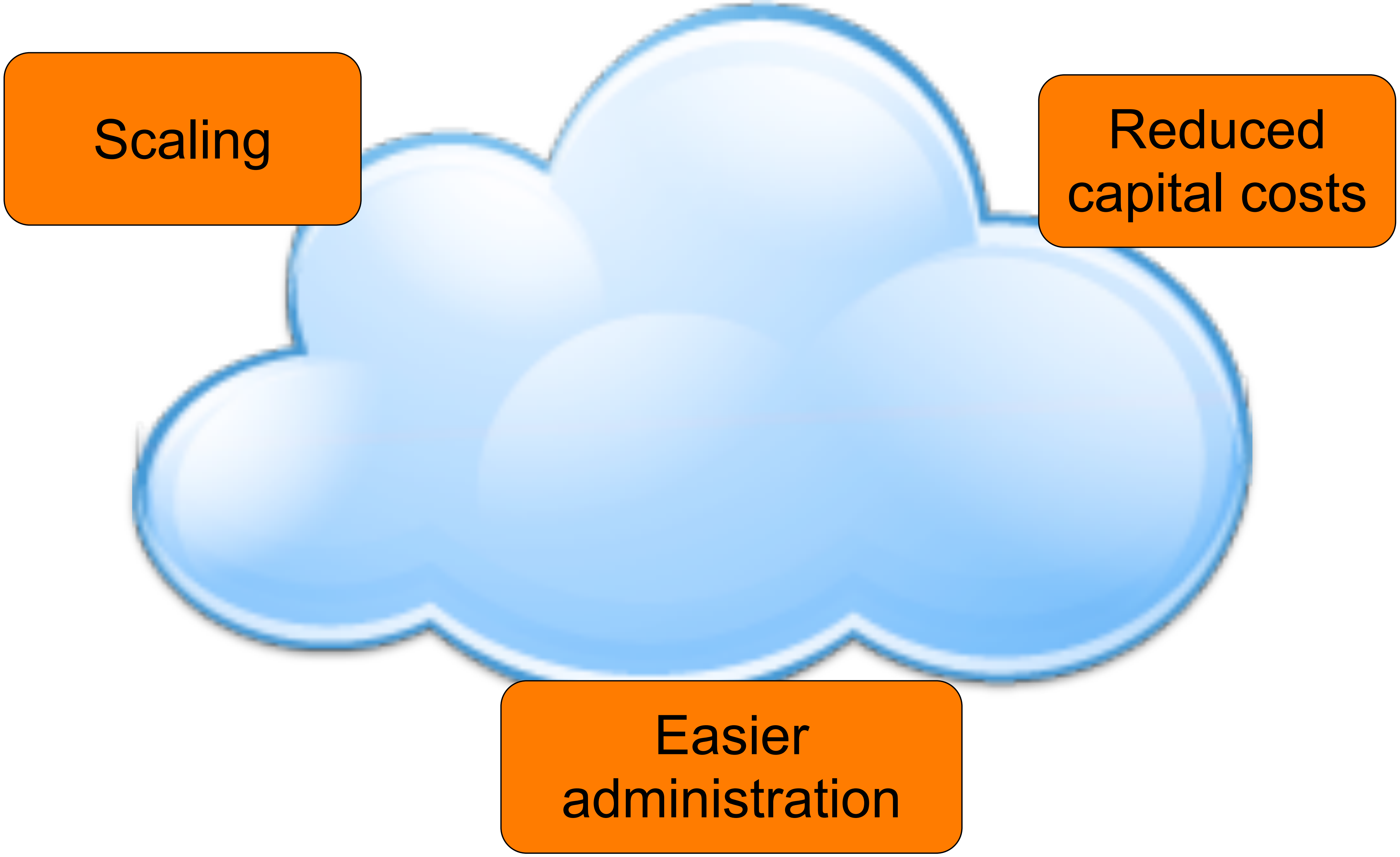
NoSQL / Big Data

What do we store?

Everyone has interesting data
Business Intelligence
Hadoop and the like for processing data
Who, what, when - maybe why?

Everyone’s networking now!
Why login with yet another username and password? Why do we have to provide our personal data *again* every time we use a new app/site
Consumer: Users want to integrate with their social network *platforms*
Enterprise: Marketing, customer relations -> Twitter and Facebook info
There is value in people’s networks

Cloud



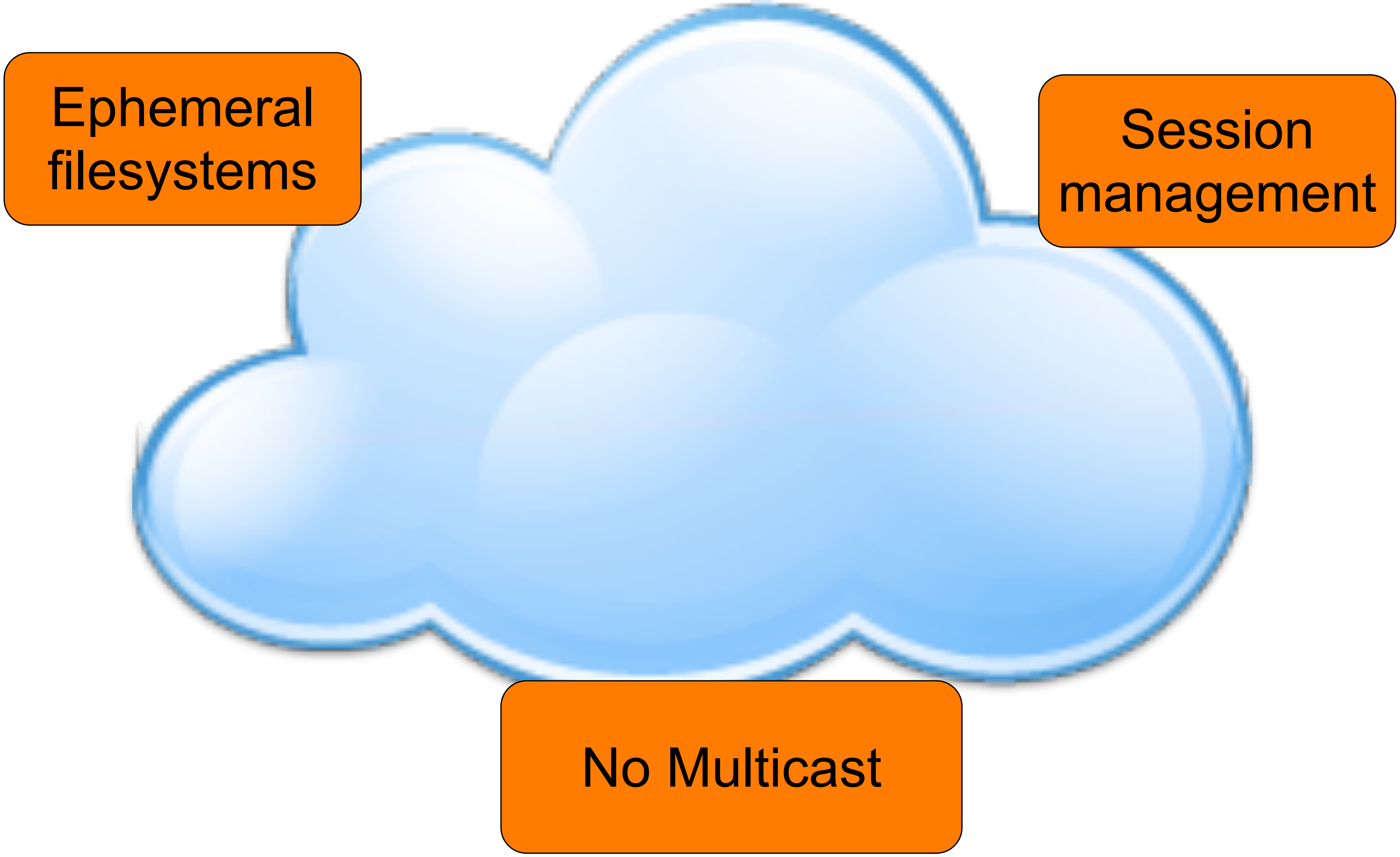
Scaling

Reduced capital costs

Easier administration

IaaS provides scalability and lower capital costs
PaaS provides simpler admin and easier development
But, cloud affects the way you architect applications

Cloud



Ephemeral filesystems

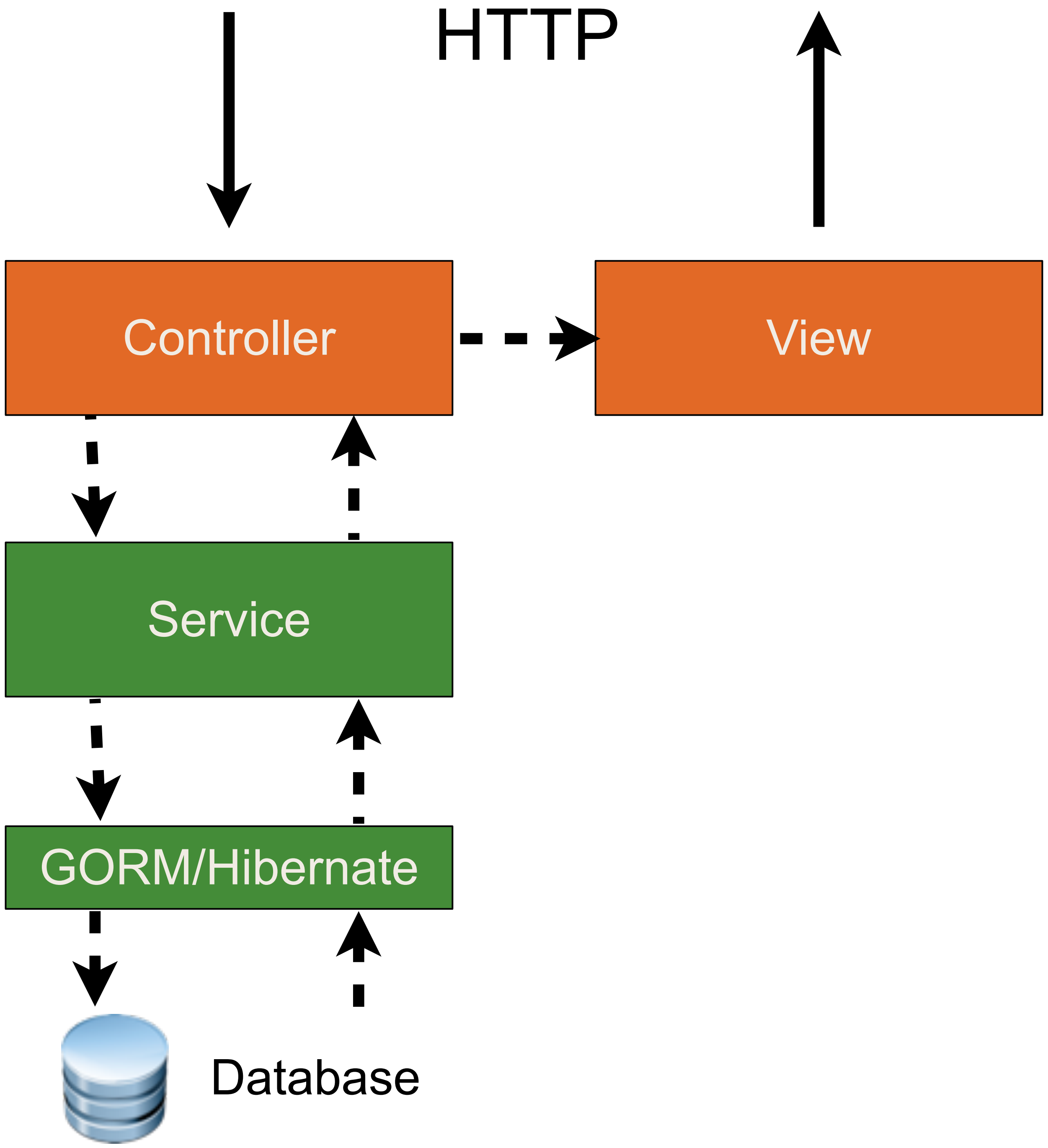
Session management

No Multicast

VMs may be started/restarted => ephemeral filesystem
Who does session management? Limited control.
No multicast => fewer options for distributed caches/heaps
Cloud relevant for the enterprise
Memory usage becomes important

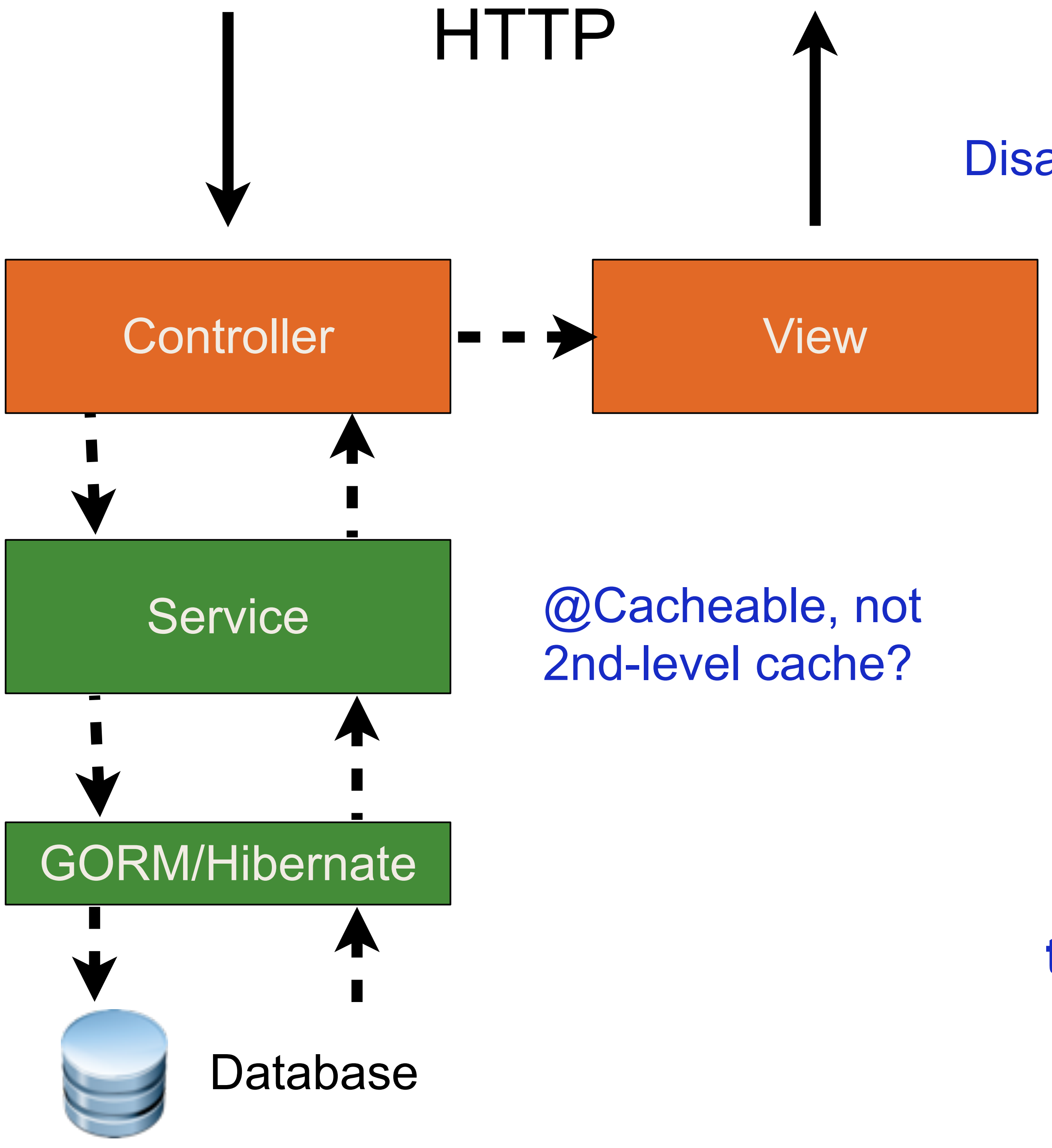
Exciting times, but hard on developers
Too much to learn!

A typical Grails app



Core Grails is full-stack, MVC framework with persistence

A typical Grails app



Disable OSIVI?

Command objects?

Don’t let HTTP session and Hibernate session mix?

Bind COs to domains?

importFrom(Domain)

@Cacheable, not 2nd-level cache?

Filtering and transforming data

What do these cars have in common?

VW Golf
Audi A3
Skoda Yeti
Seat Altea

Same platform, different components

How can Grails help - Rich UIs?

URL mappings for REST

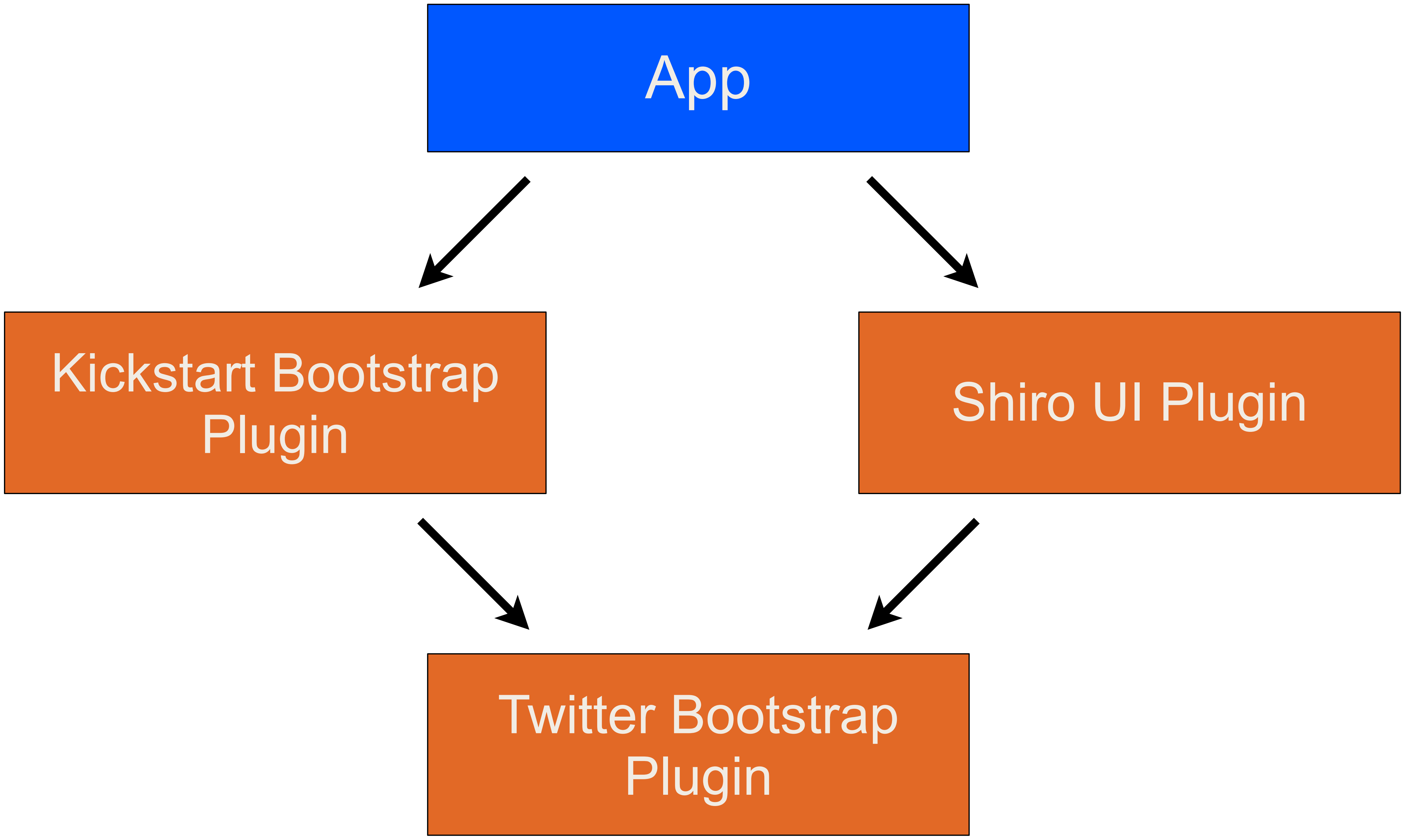
JSON & XML converters

Resources (for JS & CSS)

zipping, caching, compression

What can Grails do, since Rich UI is mostly JS & CSS?
Lots of churn in client-side frameworks

Static resource handling



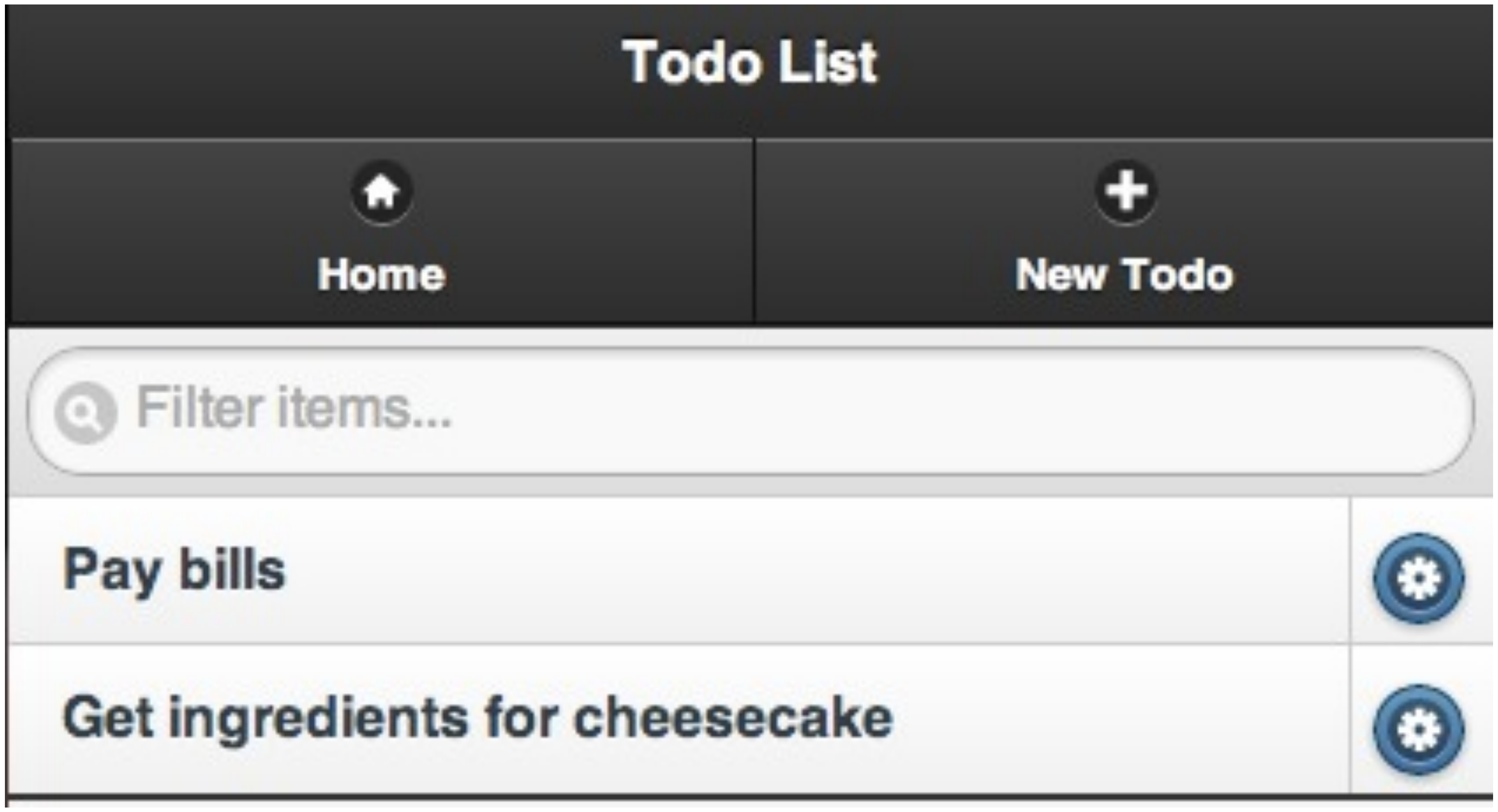
Only one instance of JS/CSS ever included in a page!
Advantage of wrapping JS libraries in plugins

Scaffolding libraries

```
> grails install-plugin jquery-mobile-scaffolding
> grails install-mobile-templates
> grails create-domain-class org.example.Todo

<edit Todo.groovy>

> grails generate-all org.example.Todo
> grails run-app
```



Pay bills

Get ingredients for cheesecake

Good for getting started and learning a JS framework

How can Grails help - Social?

Simplified OAuth via OAuth plugin:

```
oauth {
    providers {
        twitter {
            api = TwitterApi
            key = 'my-key'
            secret = 'my-secret'
            successUri = '/'
            failureUri = '/'
        }
    }
}
```

How can Grails help - Social?

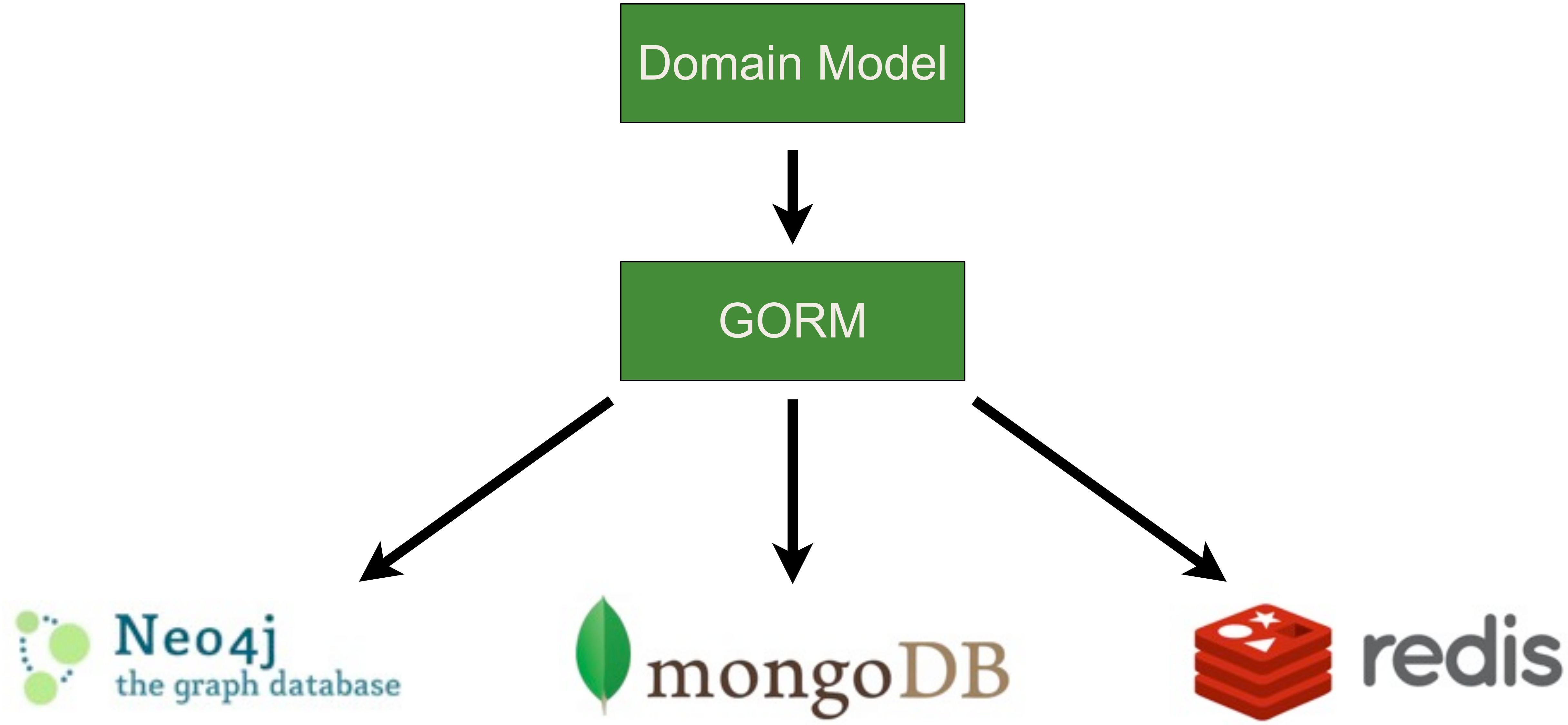
```
<oauth:connect provider="twitter">
  Connect to Twitter
</oauth:connect>
```

```
class MyService {
    def oauthService

    def myMethod() {
        def twitterUsers = oauthService.getTwitterResource(
                twitterAccessToken,
                "http://api.yourprovider.com/users/list")
        ...
    }
}
```

How can Grails help - NoSQL?

Accessible NoSQL via GORM



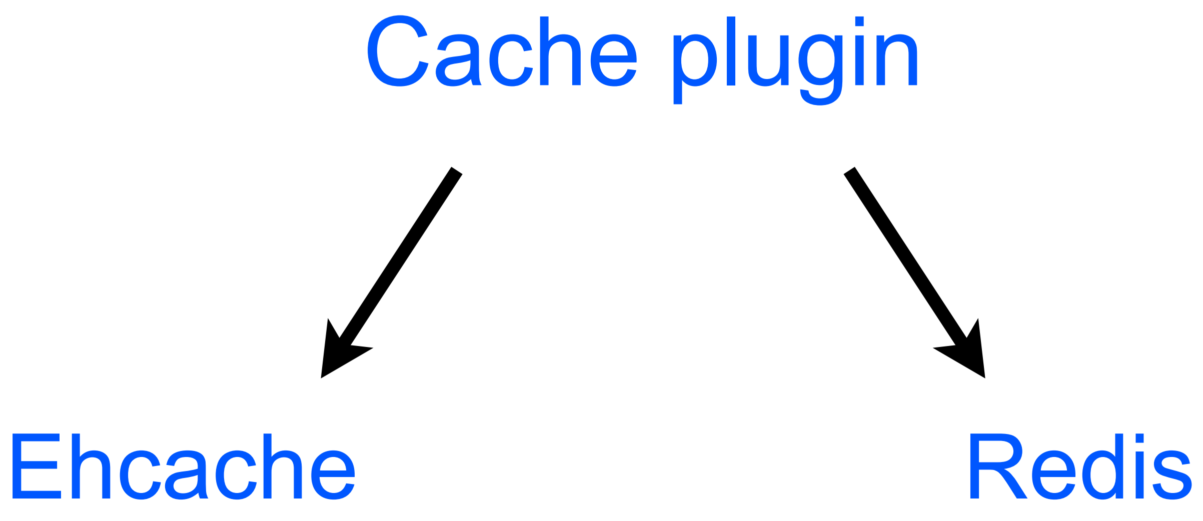
Domain Model

GORM

Single API for all data stores
But you will need to become familiar with the ones you use
Also low-level APIs to take advantage of datastore specialities
Schemaless + dynamic lang = good!

How can Grails help - Cloud?

Solve the caching problem with the Cache plugin

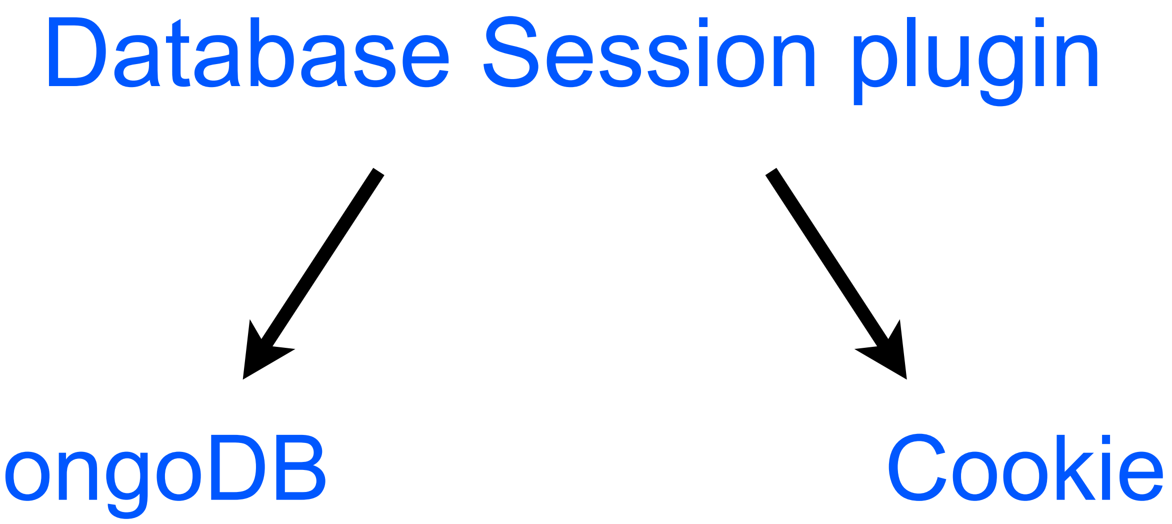


Cache plugin

Ehcache

Redis

For HTTP sessions, Database Session plugin

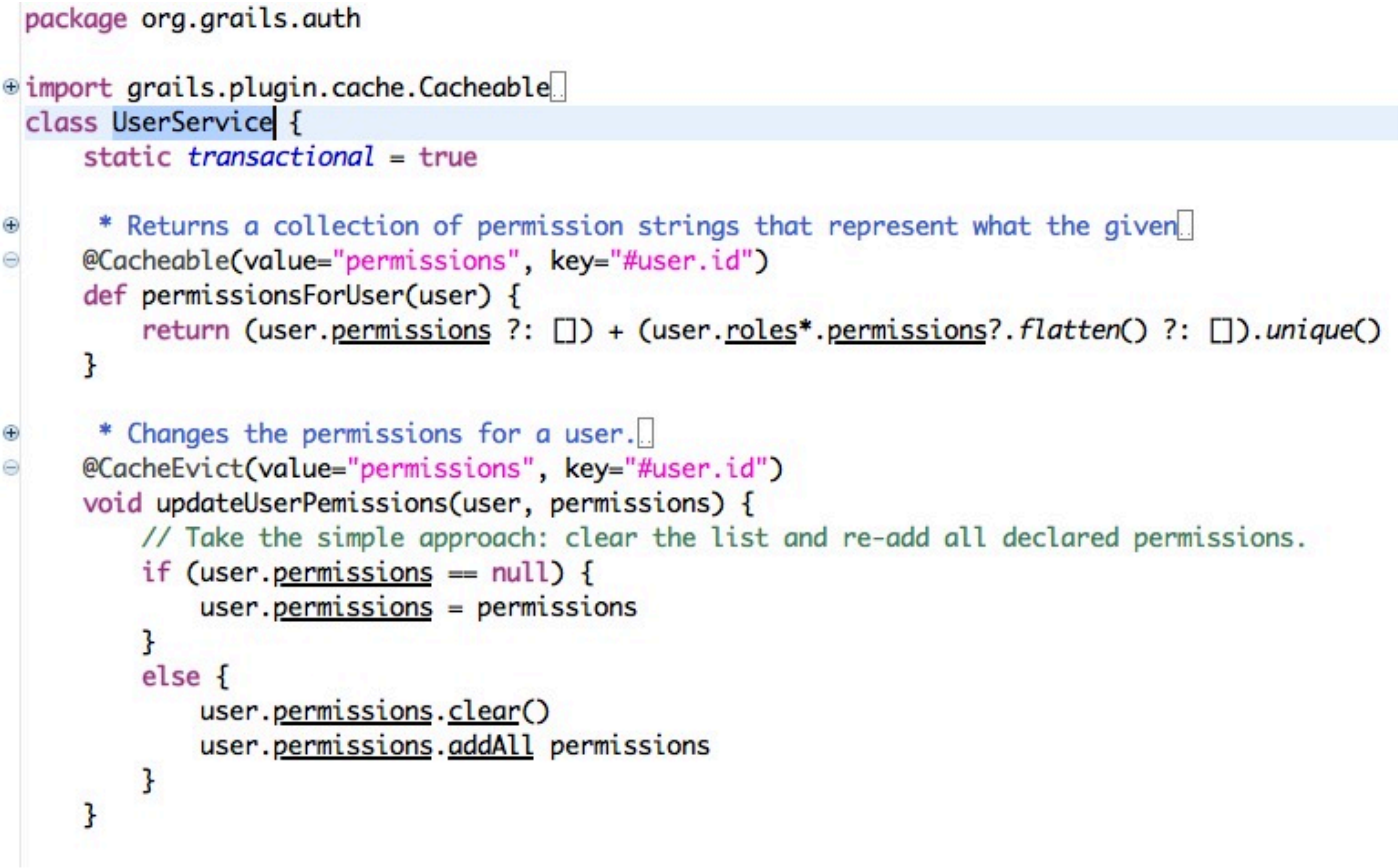


Database Session plugin

MongoDB

Cookie

Multiple application instances == shared cache



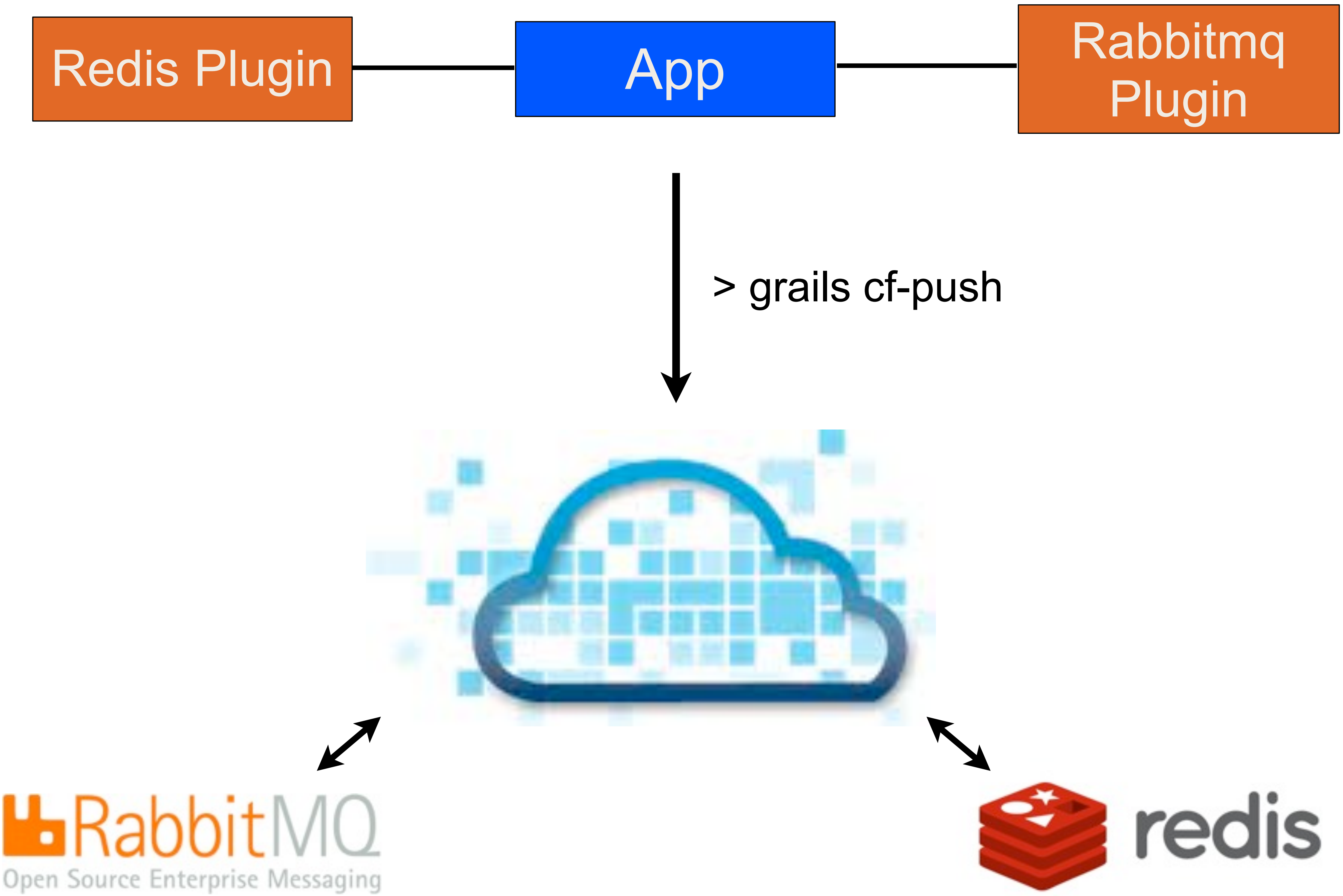
```
package org.grails.auth

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
        }
    }
```

PaaS deployment



Redis Plugin

App

Rabbitmq Plugin

> grails cf-push

Auto-reconfiguration of data source and template beans

Enabling plugin authors

More Platform Core

Security API

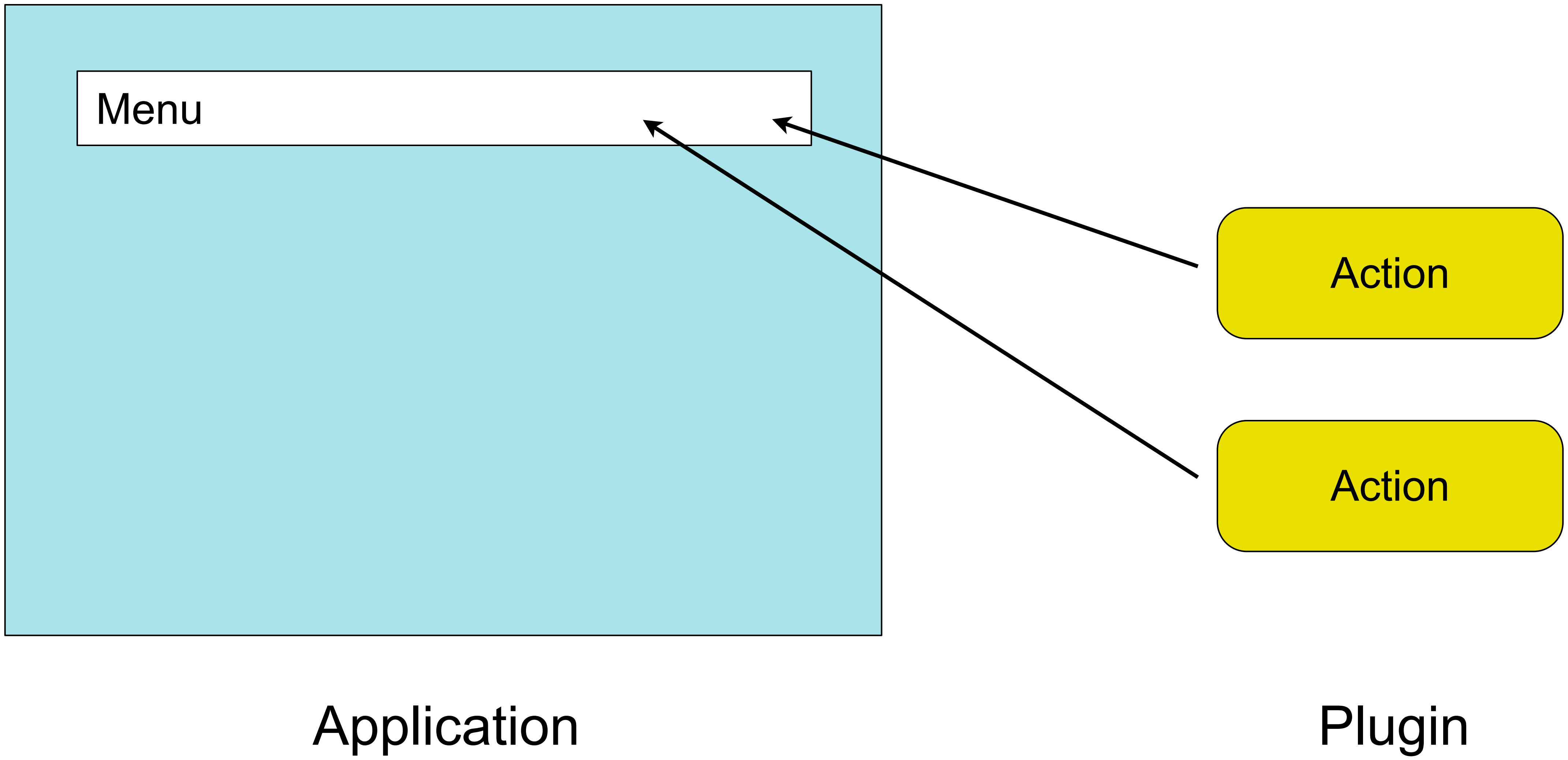
Who is the current user?

Does the user have a particular role?

Is user permitted to do something?

More Platform Core

Navigation API



More Platform Core

Config API

Declare config options

Automatic namespacing

Default values & automatic merging

More Platform Core

Convention API

e.g.

static searchable = { ... }

@Taggable

Convention overrides!

Platform UI

Themes

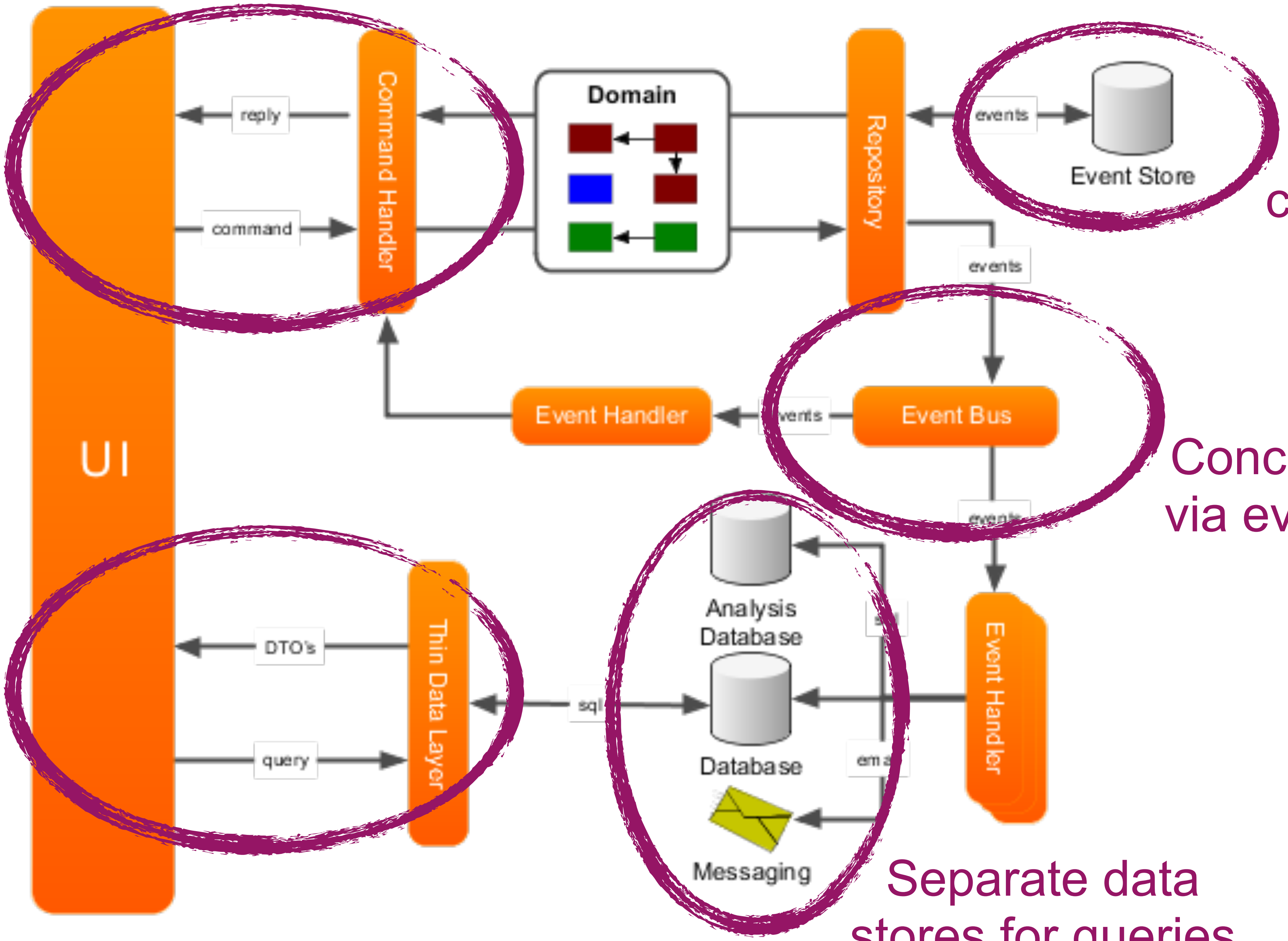
+

UI tags

e.g. App Info plugin + Bootstrap Kickstart

What about horizontal scalability?

Introducing CQRS



Updates

Store changes

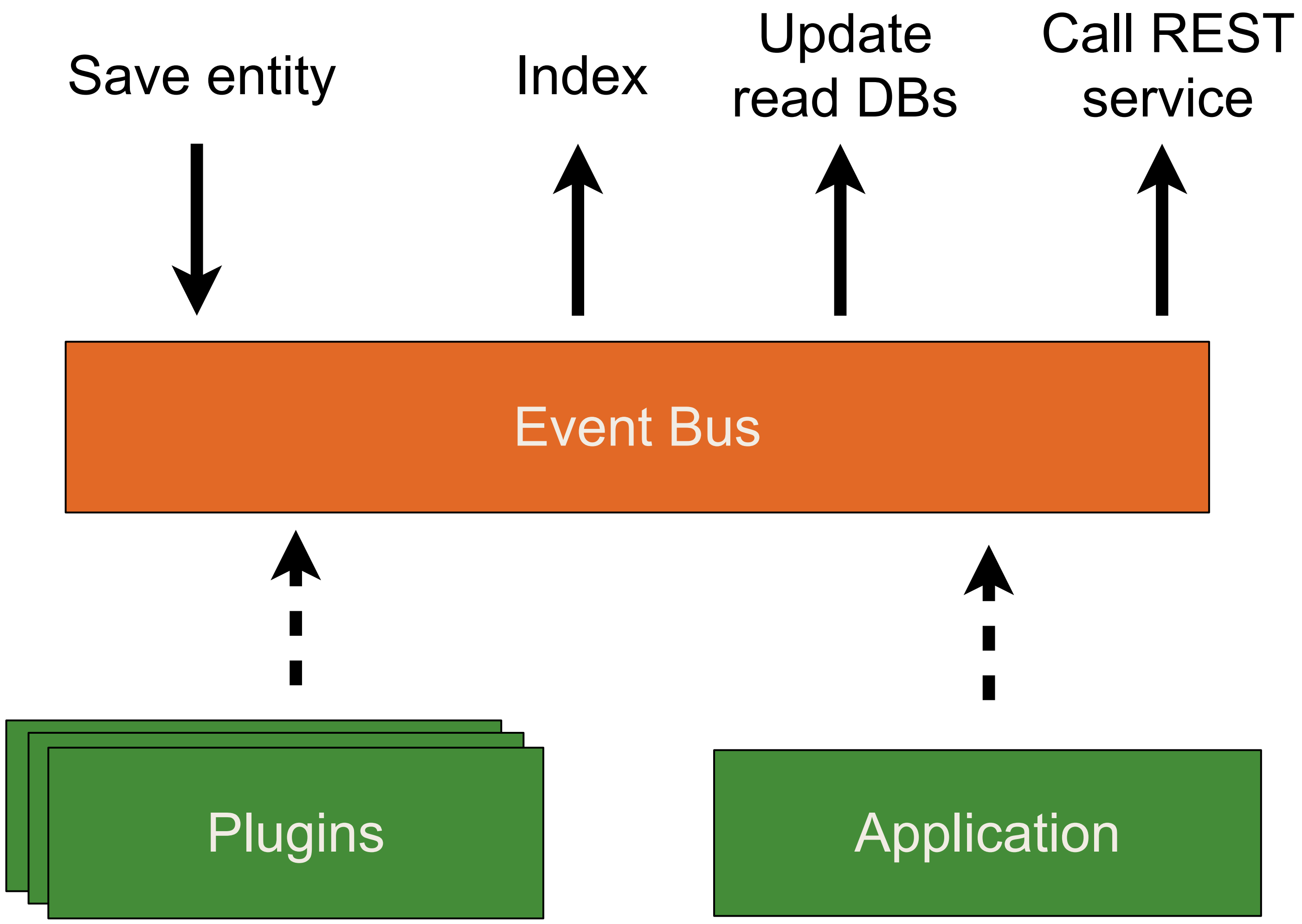
Concurrency via event bus

Views

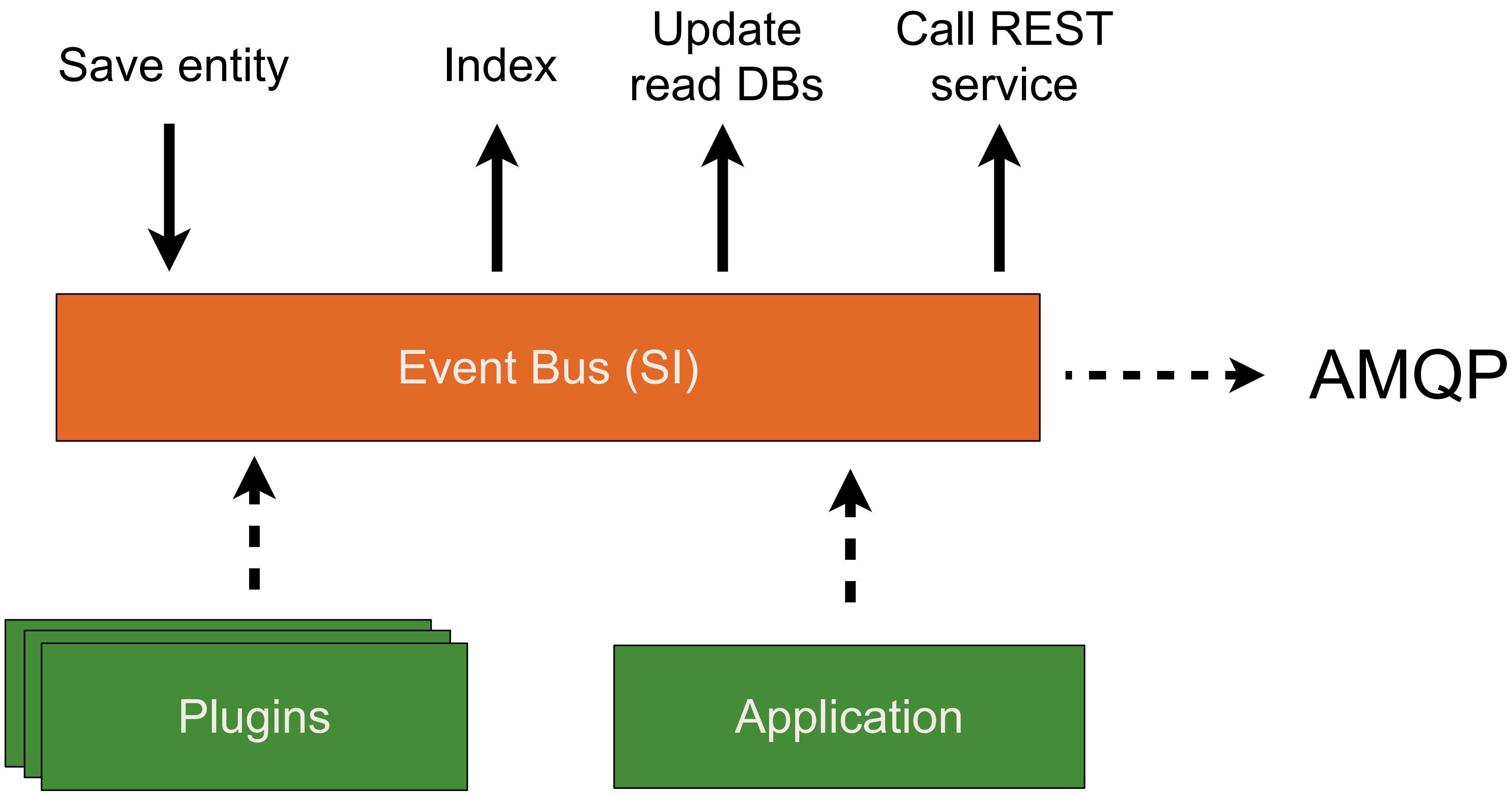
Separate data stores for queries

Why? Updates and querying often have different data requirements.
For example, Lanyrd use Redis structured data support
All read databases can be rebuilt from master events DB
CQRS designed for scale

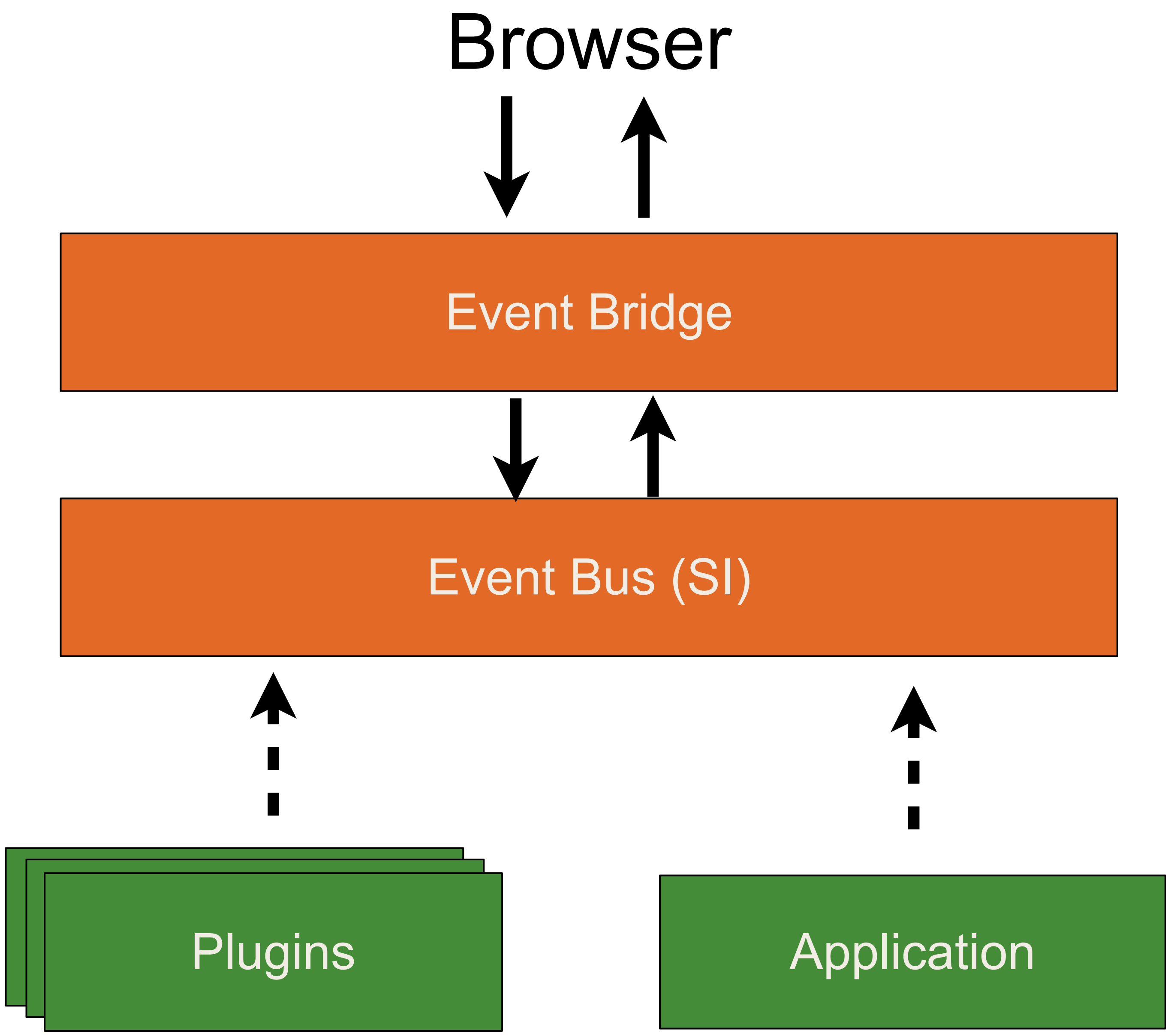
Event bus (Platform Core plugin)



Event bus (Events SI plugin)



Event bus (Events Push plugin)



Project archetypes?

SQL database

Multi page

CRUD

jQuery

SQL for write

Redis for read

REST endpoints

AngularJS

Events

Spring Integration

Web Sockets (with emulation)

Summary

- The way applications are architected *will* change
  - Websites will still be built (GSP not gone yet)
  - Not everyone will need the same architecture
  - Project archetypes and scaffolding!
- No single framework has everything you need
- Pick and choose the appropriate components for your

More info

- w: http://grails.org/
- f: http://grails.org/Mailing+Lists

- e: p.ledbrook@cacoethes.co.uk
- t: pledbrook
- b: http://www.cacoethes.co.uk/blog/

Q & A