Modern Architectures with Spring and JavaScript

Martin Lippert, VMware
mlippert@vmware.com, @martinlippert

about me

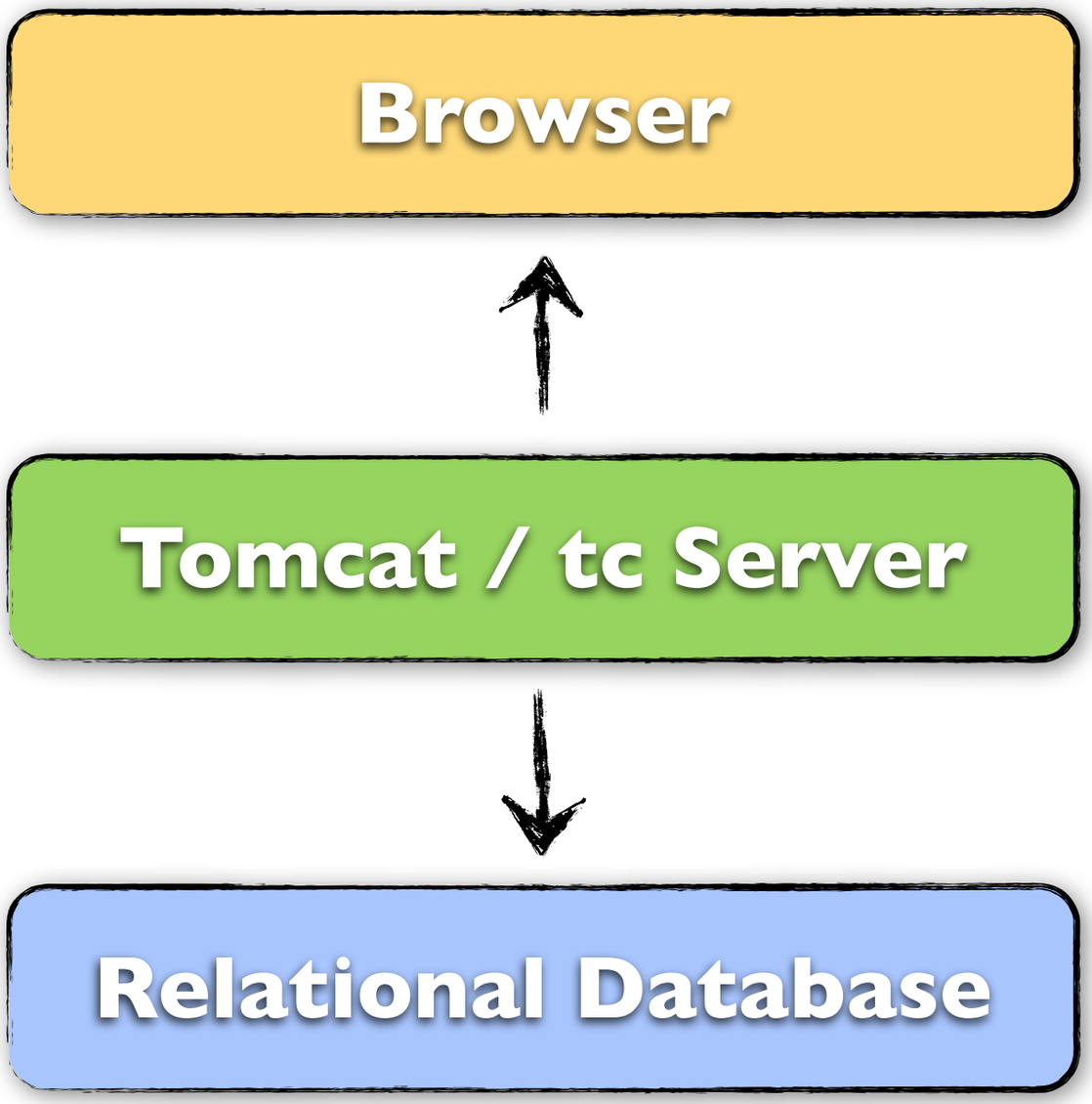
Martin Lippert

Staff Engineer, R&D, at SpringSource/VMware and lead of Spring Tool Suite development

Working areas

- Eclipse Tooling for the Spring platform
- Open Source Committer
- Aspect-Weaving for OSGi
- Cloud IDEs and JavaScript
- Agile software development

Typical Runtime Structures

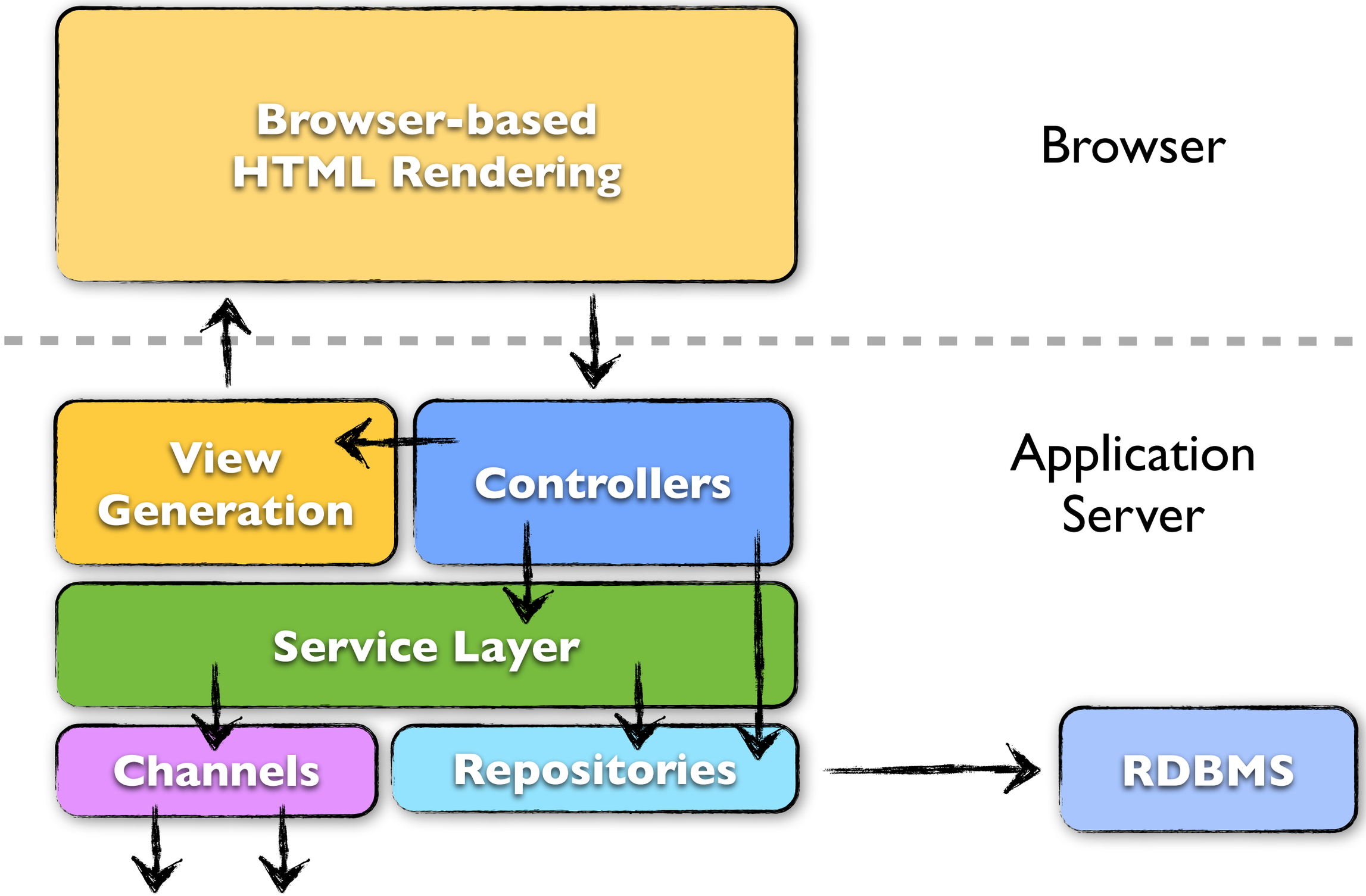


render HTML

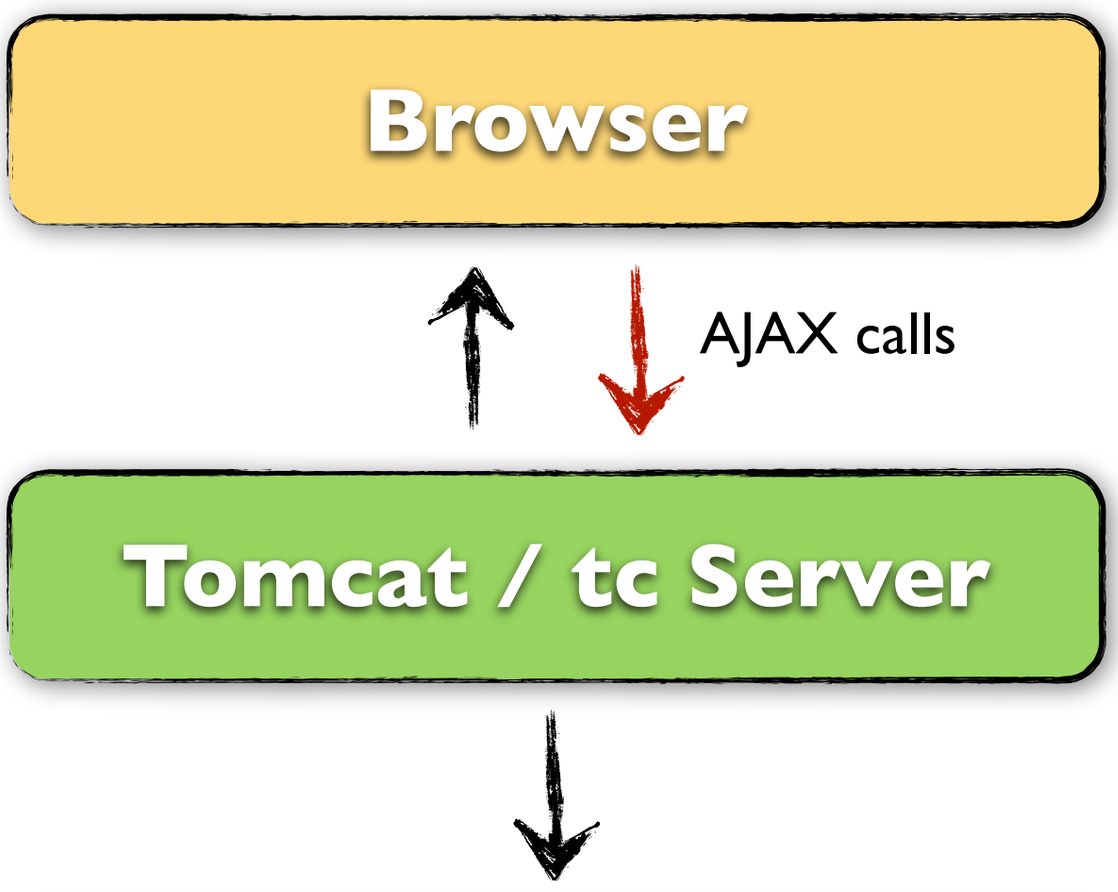
business logic &
page rendering

contains data

More Detail...


Browser-based
HTML Rendering
Browser
View
Generation
Controllers
Service Layer
Channels
Repositories
Application
Server
RDBMS

What happens?

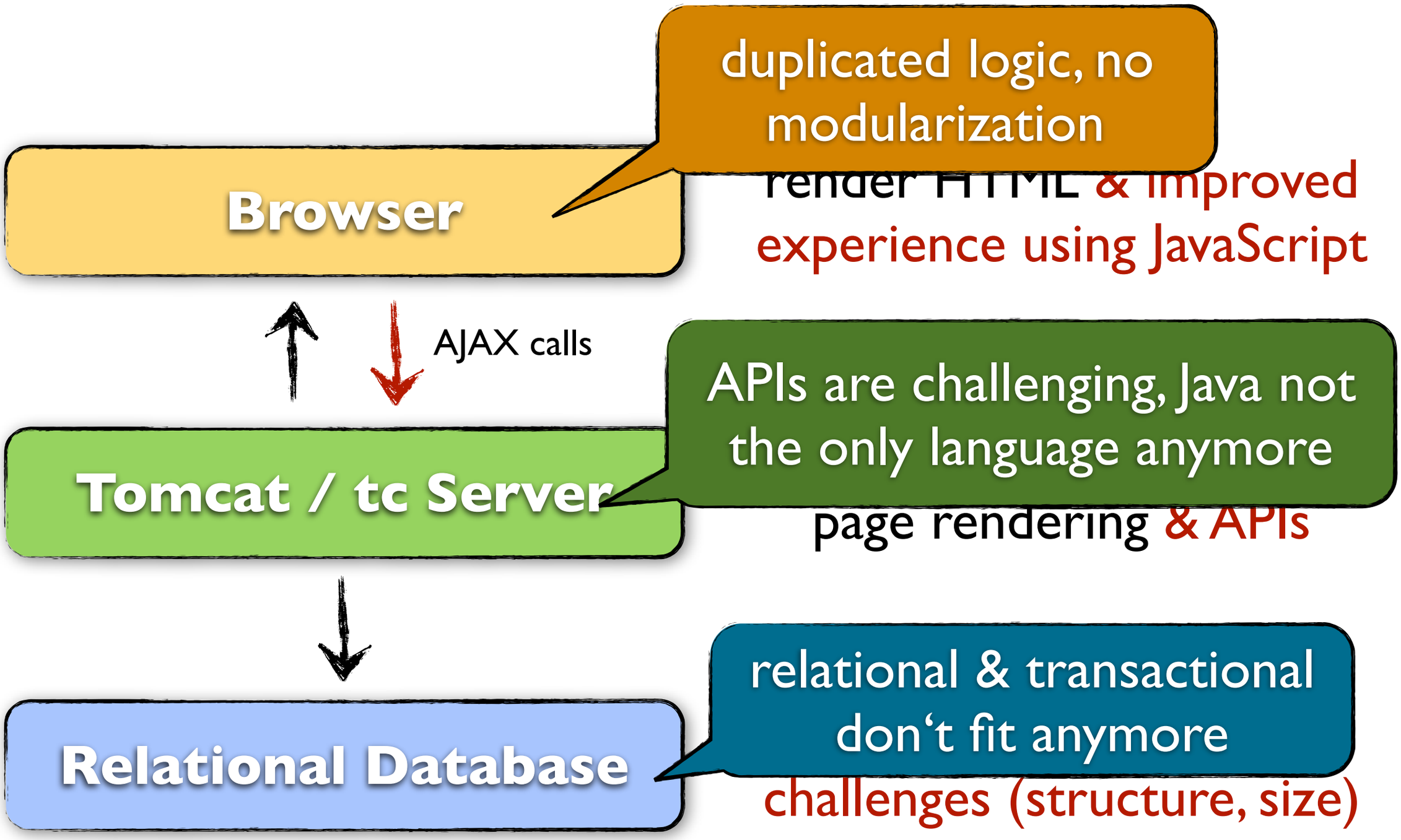


render HTML & improved experience using JavaScript

business logic & page rendering & APIs

contains data & new challenges (structure, size)

A few observations



Different pictures

AWS

node.js

NoSQL

JavaScript

Hadoop

CoffeeScript

modern apps

Scala

Ruby/Rails

Clojure

PaaS

HTML5/CSS3

HTML/css

Java

JavaScript

old style apps

Application Server

RDBMS

Where do we go from here?

The Client Side

Innovation happens here

The JavaScript Story

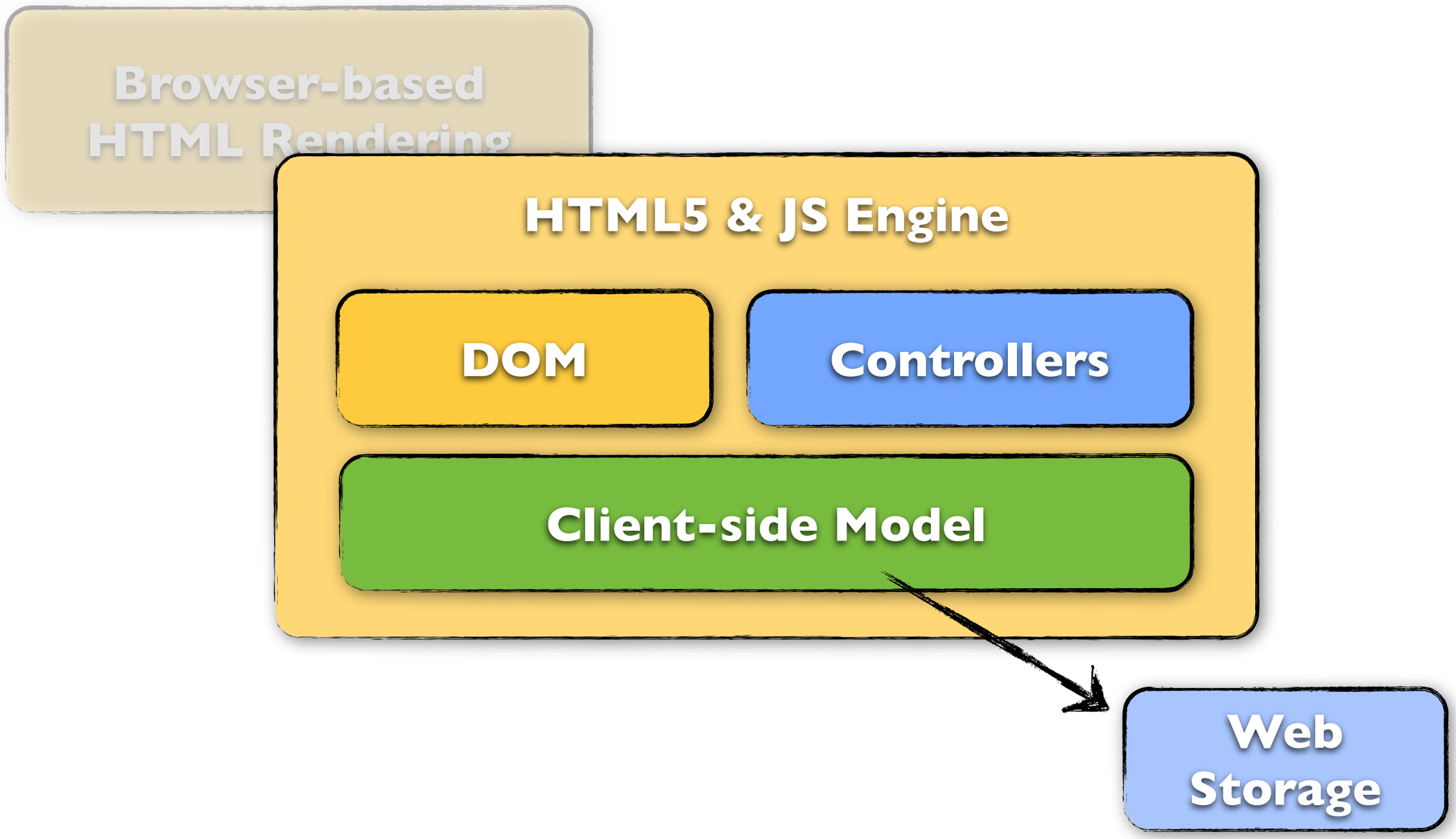
http://www.maztek.com/blog/wp-content/uploads/javascript.jpg

My assumptions
- on the client side -

Browser only (HTML5/CSS3)
JavaScript only

„The browser-based application written in JavaScript becomes the new rich client architecture“

More Detail...


Browser-based
HTML Rendering
HTML5 & JS Engine
DOM
Controllers
Client-side Model
Web
Storage

Existing JavaScript libs are UI centric

(focus on making life with the DOM easier)

most prominent:

jquery

JavaScript versions of „good old rich client patterns“ begin to appear

(and are highly necessary)

Examples

backbone.js

angular.js

ember.js

...

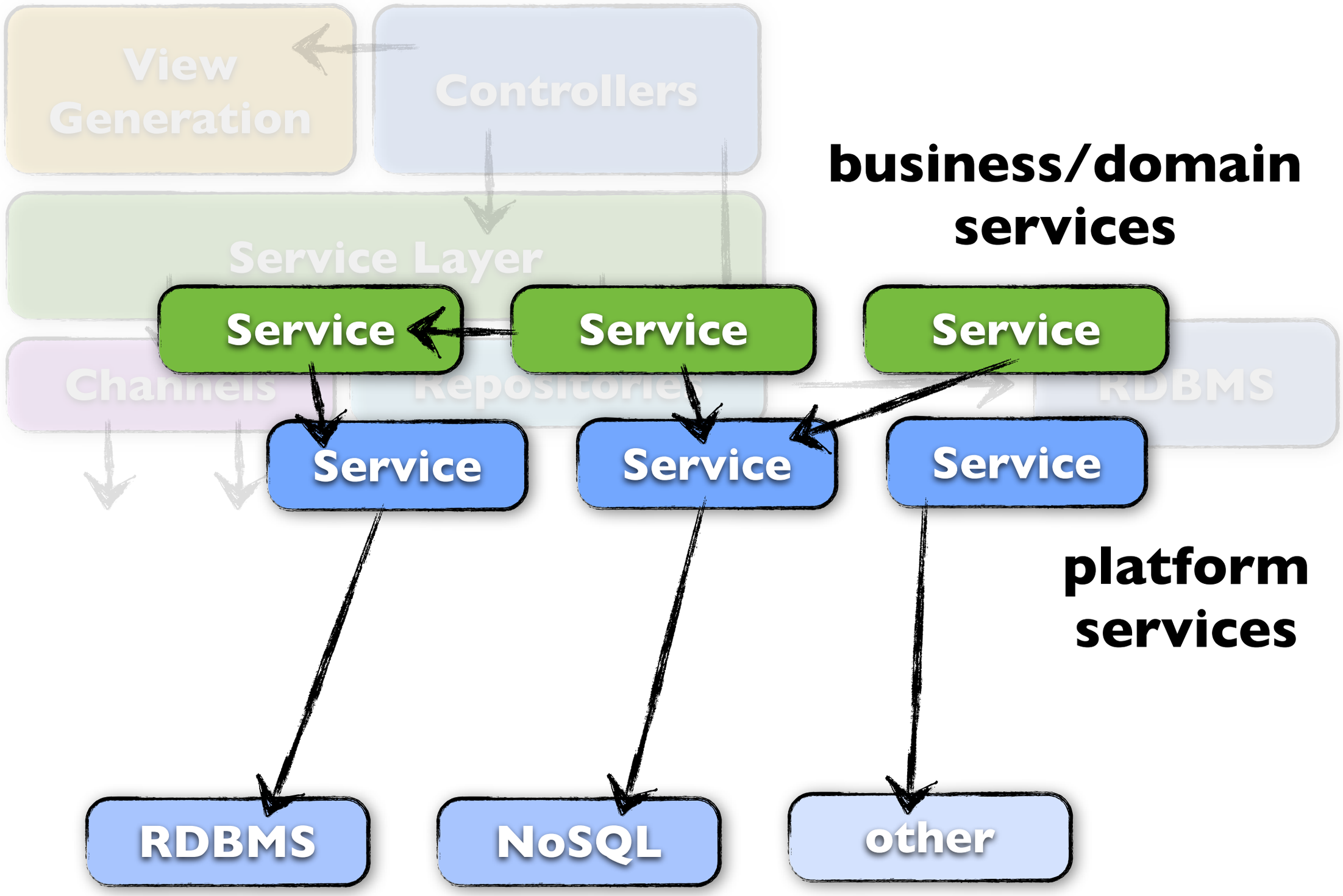
Browser App (JavaScript)

maybe also CoffeeScript, TypeScript, Dart

maybe GWT, but likely not

forget about JSF ;-)

The Server Side


View Generation
Controllers
Service Layer
Channels
Repositories
RDBMS
business/domain services
Service
Service
Service
Service
Service
Service
platform services
RDBMS
NoSQL
other

this is where **Spring** is really powerful
ready to run „in the cloud“ (scalability)
(no client-side rendering or logic)

Service Service Service Service Service

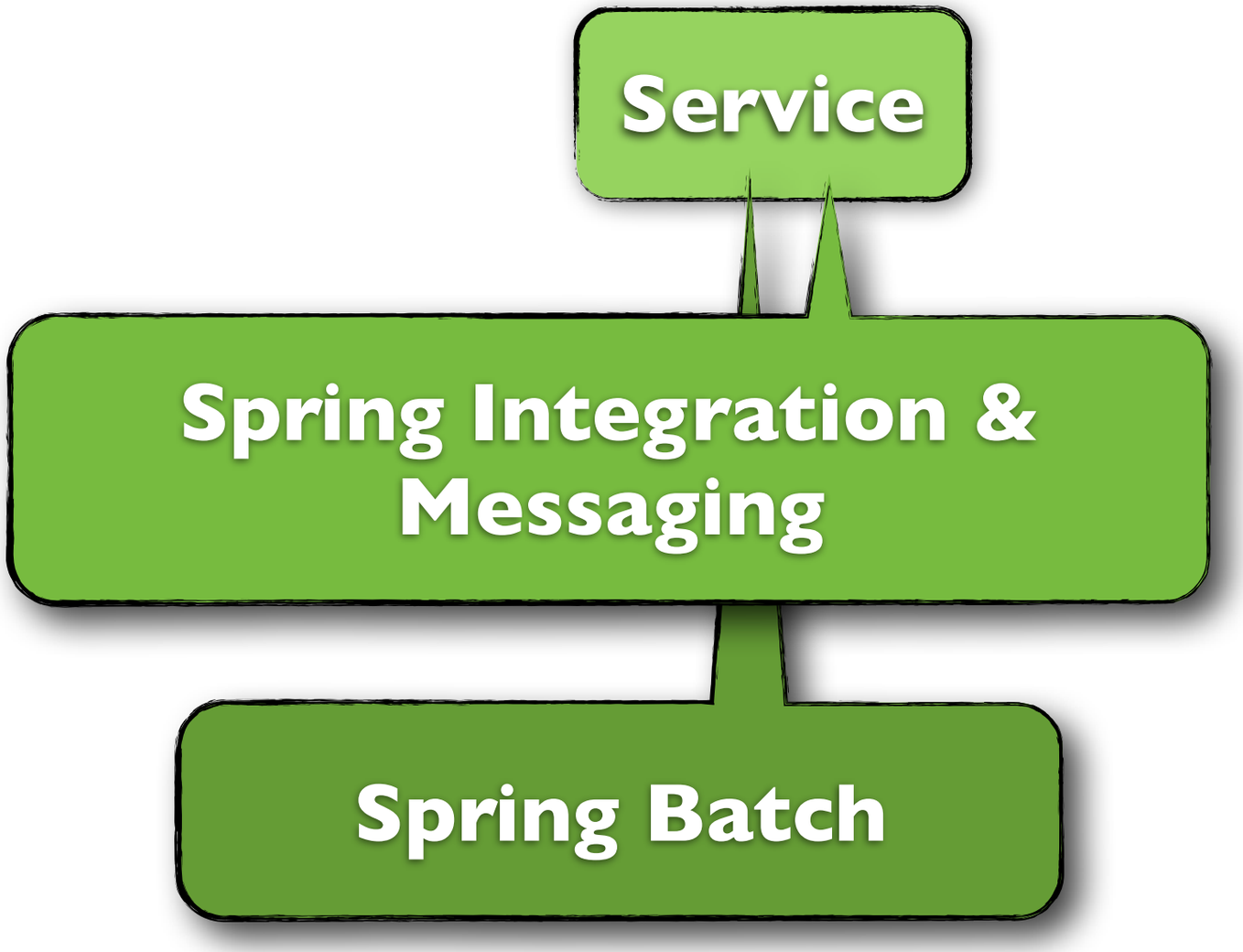
Service

Spring MVC is the easiest way to implement RESTful APIs and services

APIs are JSON and HATEOAS based

Spring MVC + Spring HATEOAS is a powerful combination

more on Spring HATEOAS:
https://github.com/SpringSource/spring-hateoas


Service
Spring Integration &
Messaging
Spring Batch

My assumptions

- server side languages -

many different languages in use

choose the right language for the right job

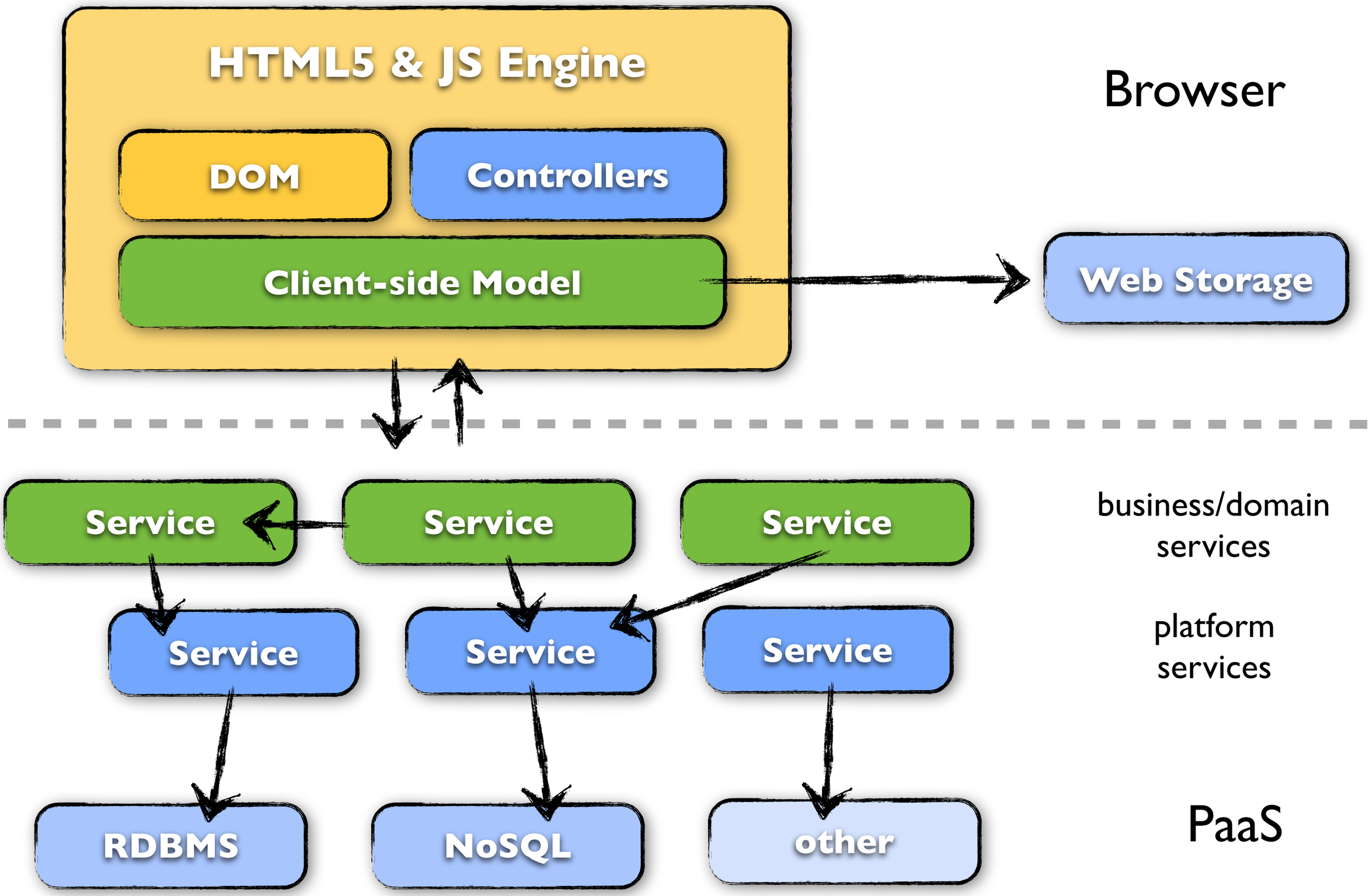
don‘t use a new language for fun

My assumptions

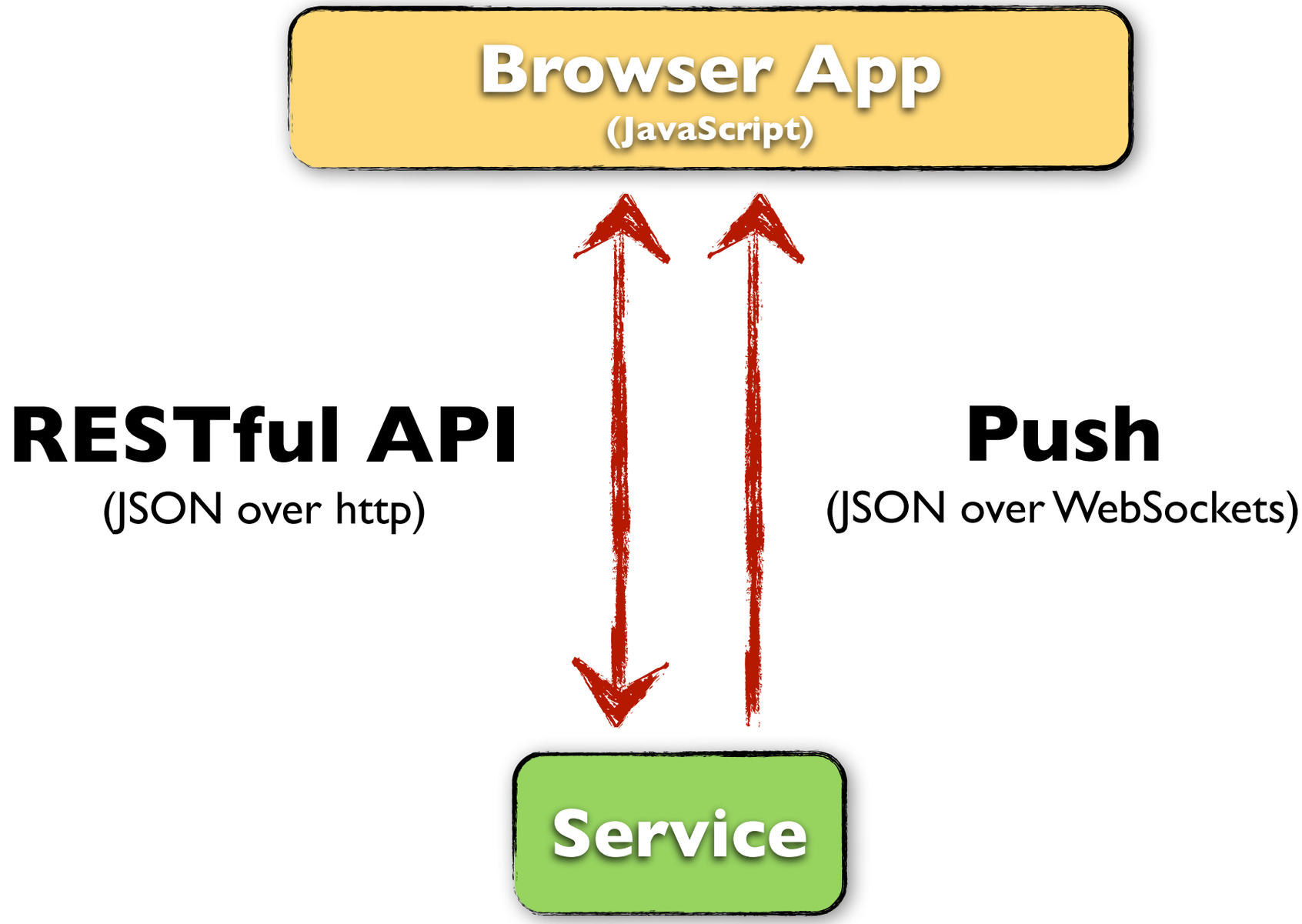
- data storage -

more and more data (big data)
different storage techniques combined
(rdbms, nosql, graph databases)
scalability is important

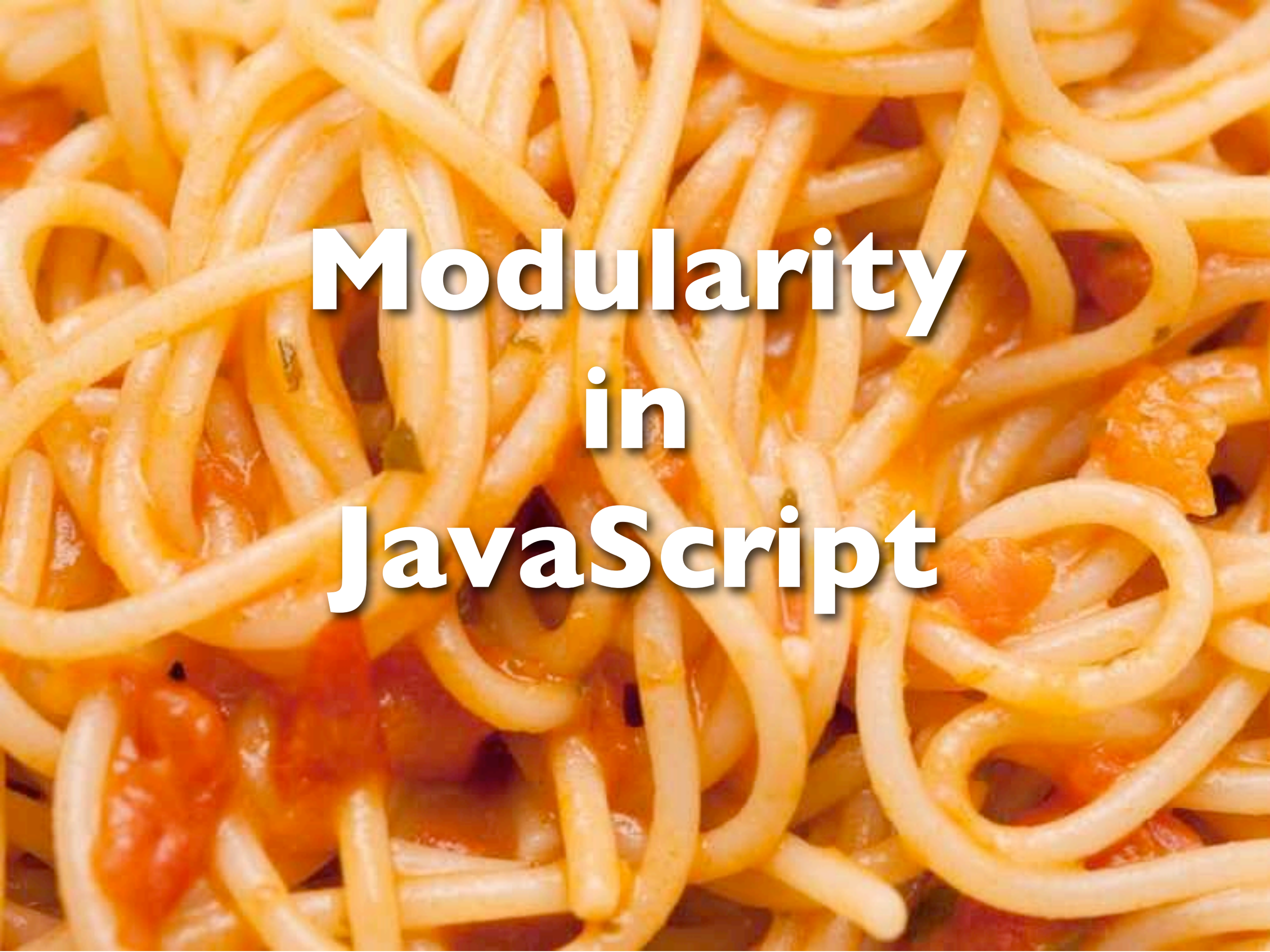
The picture


HTML5 & JS Engine
DOM
Controllers
Client-side Model
Web Storage
Browser
Service
Service
Service
Service
Service
Service
business/domain services
platform services
RDBMS
NoSQL
other
PaaS

Communication



The Challenges


Modularity
in
JavaScript

AMD
(asynchronous module definition)

wire.js
(Dependency Injection for JavaScript)

Micro Services for JavaScript
(OSGi services written in JavaScript)

More Challenges

offline
cloud-ready services
define good APIs
versioned APIs
TDD for JavaScript

more information

Adrian Colyer on Application Development in the Cloud Era
http://www.youtube.com/watch?v=axOPJbrIjkY

Example app using Spring for providing RESTful APIs and JavaScript for a rich client and mobile app
https://github.com/SpringSource/html5expense

Asynchronous Module Definition for JavaScript (AMD)
https://github.com/amdjs/amdjs-api
http://requirejs.org/docs/whyamd.html

wire.js
https://github.com/cujojs/wire

hello world with wire.js
https://github.com/briancavalier/hello-wire.js

more advanced example for wire.js
https://github.com/briancavalier/piratescript

Cloud Foundry PaaS
http://www.cloudfoundry.com
http://www.cloudfoundry.org

Q&A

and thank you for your attention

Martin Lippert, VMware
mlippert@vmware.com, @martinlippert