

HTML5 Apps in Java & Scala with the Play Framework

@_JamesWard
jamesward.com

Modern Web Apps

- Stateless Web Tier
- Client/Server Web & Mobile Apps
- Transparent Real-time

Modern Web Apps - Why?

- Performance
- Faster development iterations
- Reuse code & services for mobile, web, etc

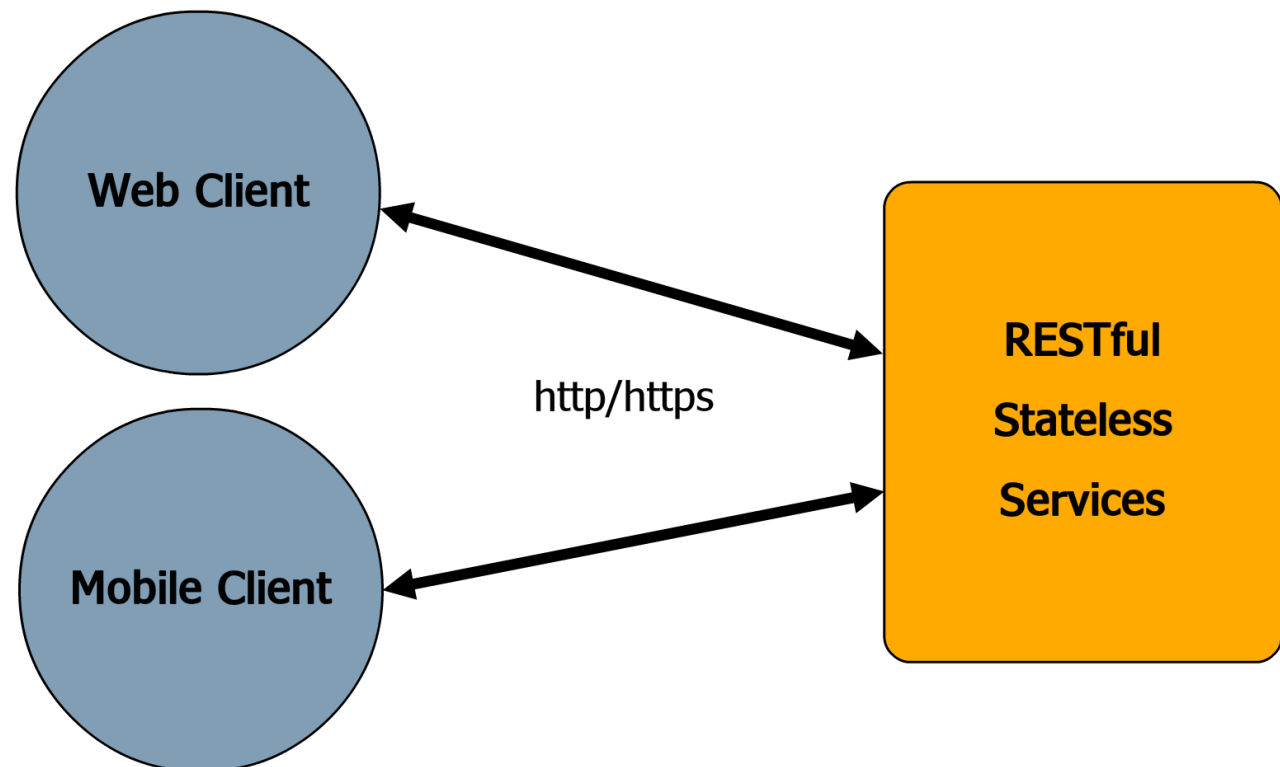


Stateless Web Tier

- Server affinity = Simple Scalability & Upgrades
- Continuous Delivery
- Browser back, forward, reload just work

Client/Server Web & Mobile Apps

- REST services (JSON)
- Web and/or Mobile Clients
- Client on a CDN



Transparent Real-time

- Keep client state in-sync
- Real-time notifications are now the norm
- Scalability?
- WebSockets?

HTML5

"HTML5 is a language for structuring and presenting content for the World Wide Web, and is a core technology of the Internet originally proposed by Opera Software. It is the fifth revision of the HTML standard (created in 1990 and standardized as HTML4 as of 1997) and as of March 2012 is still under development. Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices (web browsers, parsers, etc.). HTML5 is intended to subsume not only HTML 4, but XHTML 1 and DOM Level 2 HTML as well."

HTML5

The browser as an application platform.

HTML5

- New tags:

`<video>` `<section>` `<article>` `<header>` `<nav>`

- New APIs:

- Offline

- Drag & Drop

- Web Storage

- etc

- CSS3

- Faster JavaScript

HTML5 == Modern Web Apps

Evolving Web Tools / Libraries

- jQuery
- Twitter Bootstrap
- CoffeeScript
- Asset Compilers: Play 2, Rails, etc
- MVC Frameworks: Backbone, Angular, etc
- Client-side templating: Mustache, dust.js, etc

Play Framework

"Play is based on a lightweight, stateless, web-friendly architecture and features predictable and minimal resource consumption (CPU, memory, threads) for highly-scalable applications - thanks to its reactive model, based on Iteratee IO."

lightweight, stateless,

web-friendly

for

highly-scalable applications

My Top 10 Favorite Features

- 0) Simple
- 1) URL Routing
- 2) Parameter Binding
- 3) Class Reloading
- 4) Share-Nothing
- 5) Scala Support
- 6) Testing Support
- 7) JPA/EBean Support
- 8) NIO Server (Netty)
- 9) Instant Deployment on Heroku

Play 1 vs. Play 2

- No more magic
- Typesafe Stack
- Hibernate -> EBean (Java) || Anorm (Scala)
- Akka + Async
- Templates: Groovy (Java) || Scala (Scala) -> Scala Only
- Asset Compiler for JavaScript, CoffeeScript, and LESS
- Python CLI -> SBT
- IVY -> SBT
- Custom Config -> Typesafe Configuration

github.com/jamesward/play2bars-java

github.com/jamesward/play2bars-scala



Procfile

```
web: target/start -Dhttp.port=$PORT -DapplyEvolutions.default=true -Ddb.default.driver=org.postgresql.Driver -Ddb.default.url=$DATABASE_URL
```

project/Build.scala

```
val appDependencies = Seq(  
  "postgresql" % "postgresql" % "9.0-801.jdbc3"  
)
```

```
$ git init
```

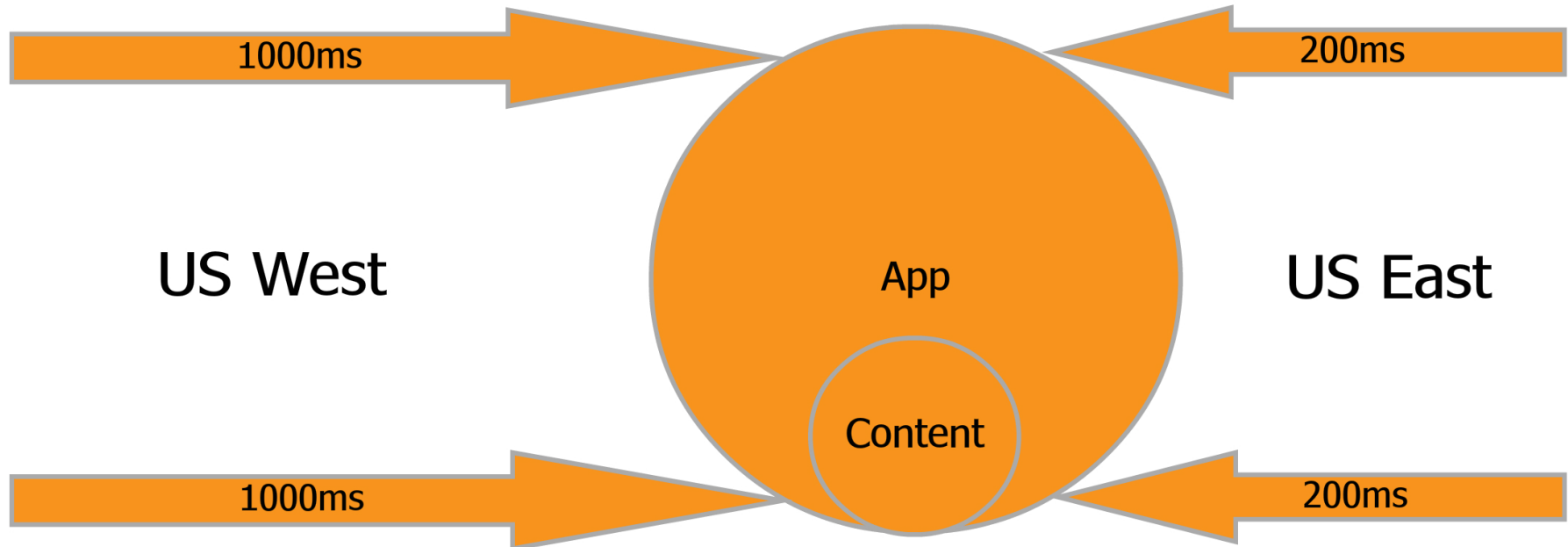
```
$ git add .
```

```
$ git commit -m init
```

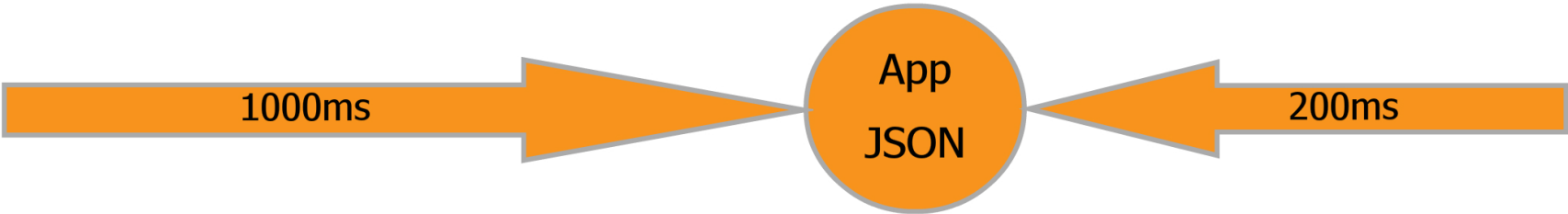
```
$ heroku create -s cedar
```

```
$ git push heroku master
```

Modern Web Apps - Traditional Web App

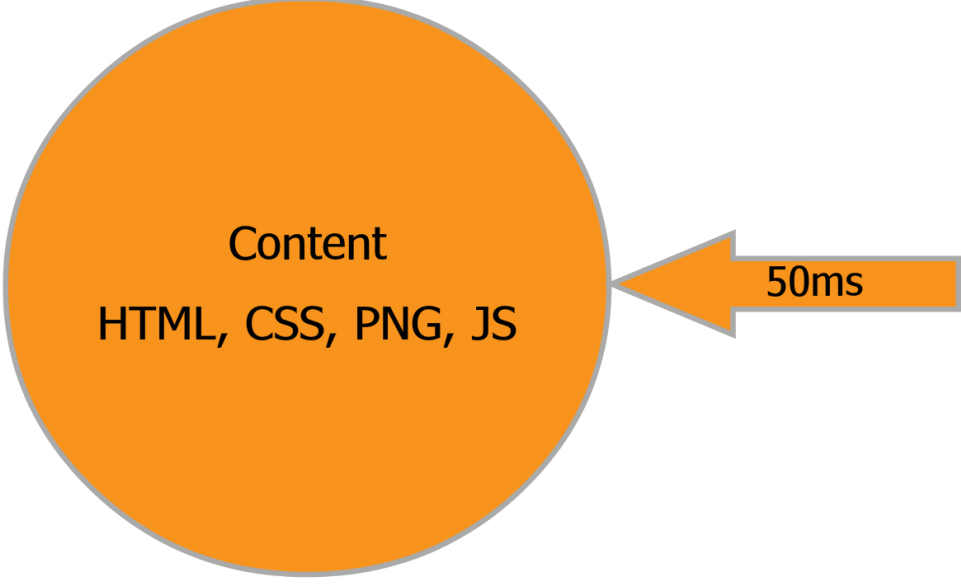
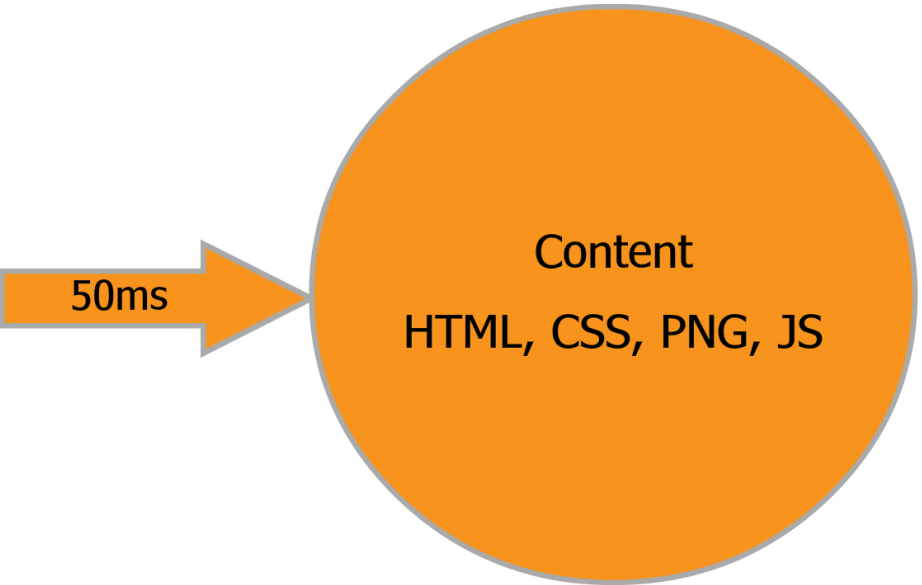


Modern Web Apps - Web Client on CDN



US West

US East



tryit.herokuapp.com

Comet + Akka

- Optional Actors
- JavaScript Callback onMessage from Server
- iframe

WebSockets + Akka

- Optional Actors
- Requires WebSocket API on client
- No built-in multi-node support

Real-time with Comet

```
public static Result liveClock() {  
    return ok(new Comet("parent.clockChanged") {  
        public void onConnected() {  
            clock.tell(this);  
        }  
    });  
}
```

playframework.org