Introduction to MeteorJS

A Fully-Fledged JavaScript App Framework

Weiming Chen, June/2015

The “WOW” Moment

- When I encountered Ruby on Rails, while I was an ASP.NET developer

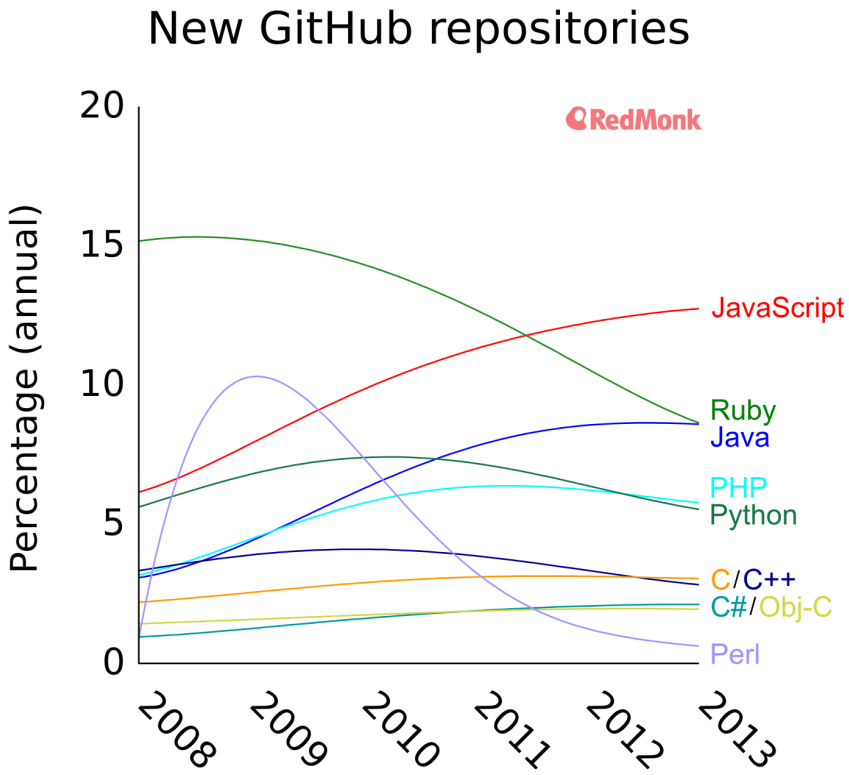
- The new “WOW” moment:

Meteor and RoR, in Common

1. Open source
2. Fully-fledged, “opinionated” web frameworks
3. Great package/library systems
   - Atmosphere vs. Rubygems
4. Super easy deployment
   - meteor.com vs. Heroku
5. Hot code reload on change
6. “Simplicity = Productivity”

What's the New "WOW" about?

- Data on the Wire
  - Built-in support for websockets
- One language
  - No need to switch between:
    - JavaScript
    - <insert name of your favourite server-side language>
    - SQL



New “WOW” (continued)

- Full Stack Reactivity
  - all layers, database to HTML templates, updates themselves automatically
- Database Everywhere
  - same data model and query interface on client, server and database
- Latency compensation
  - Prefetching data
  - Simulate actions to make it look like server methods return instantly
- More to be shown...

Demo Time!

Install Meteor:

```
$ curl https://install.meteor.com/
| sh
```

Create a simple chatroom app:

```
$ meteor create chatroom-demo
$ cd chatroom-demo
$ meteor
```

Demo (continued)

Code at: https://github.com/MingStar/meteorjs-chatroom-demo

git tags:

v0.1 - initial commit

v0.2 - accounts and log in

```
$ meteor add accounts-ui accounts-password
```

v0.3 - display and send messages

Demo (continued)

v0.4 - system message

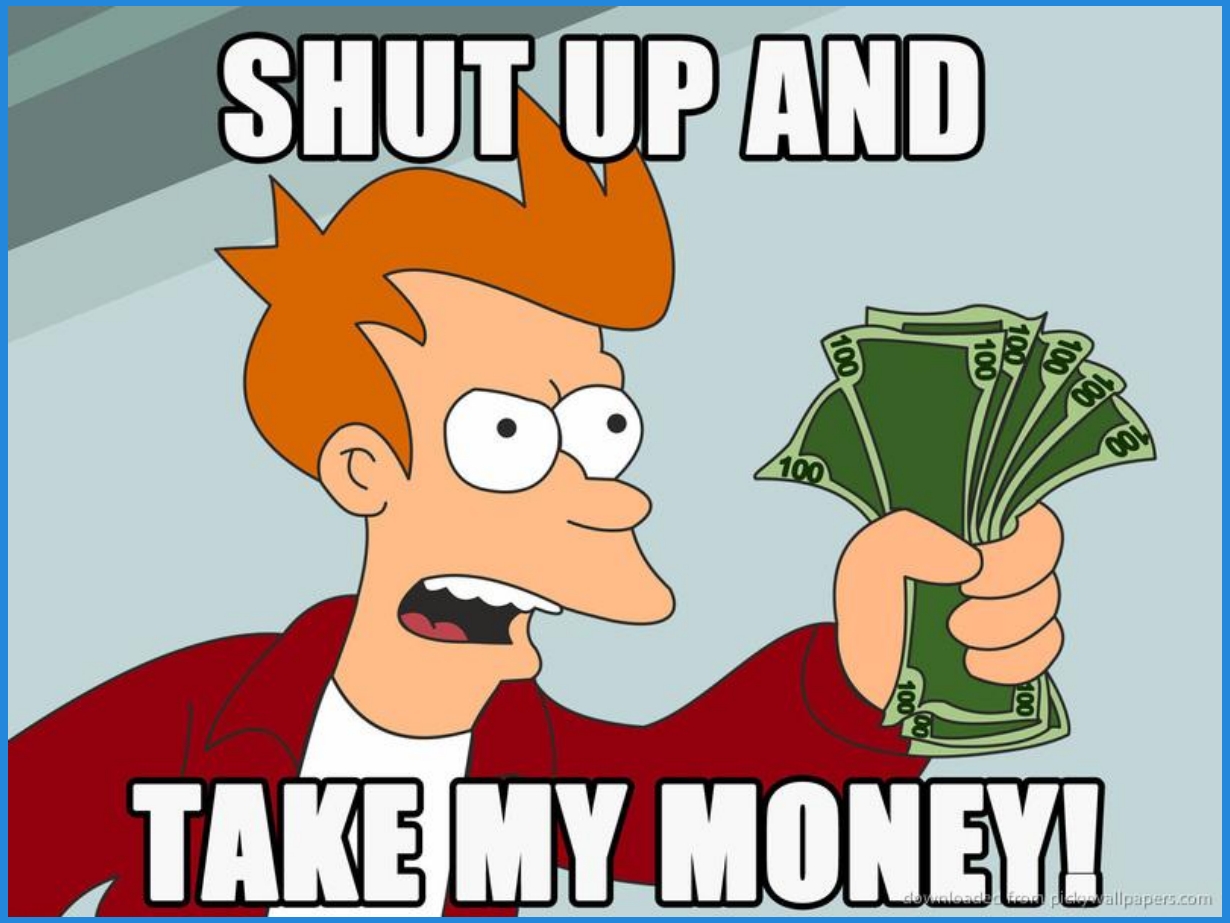
```
$ meteor mongo
meteor:PRIMARY> db.messages.insert({"username" : "system",
"text": "stop chatting, back to work!", "createdAt": new
Date()});
```

v0.5 - ios app!

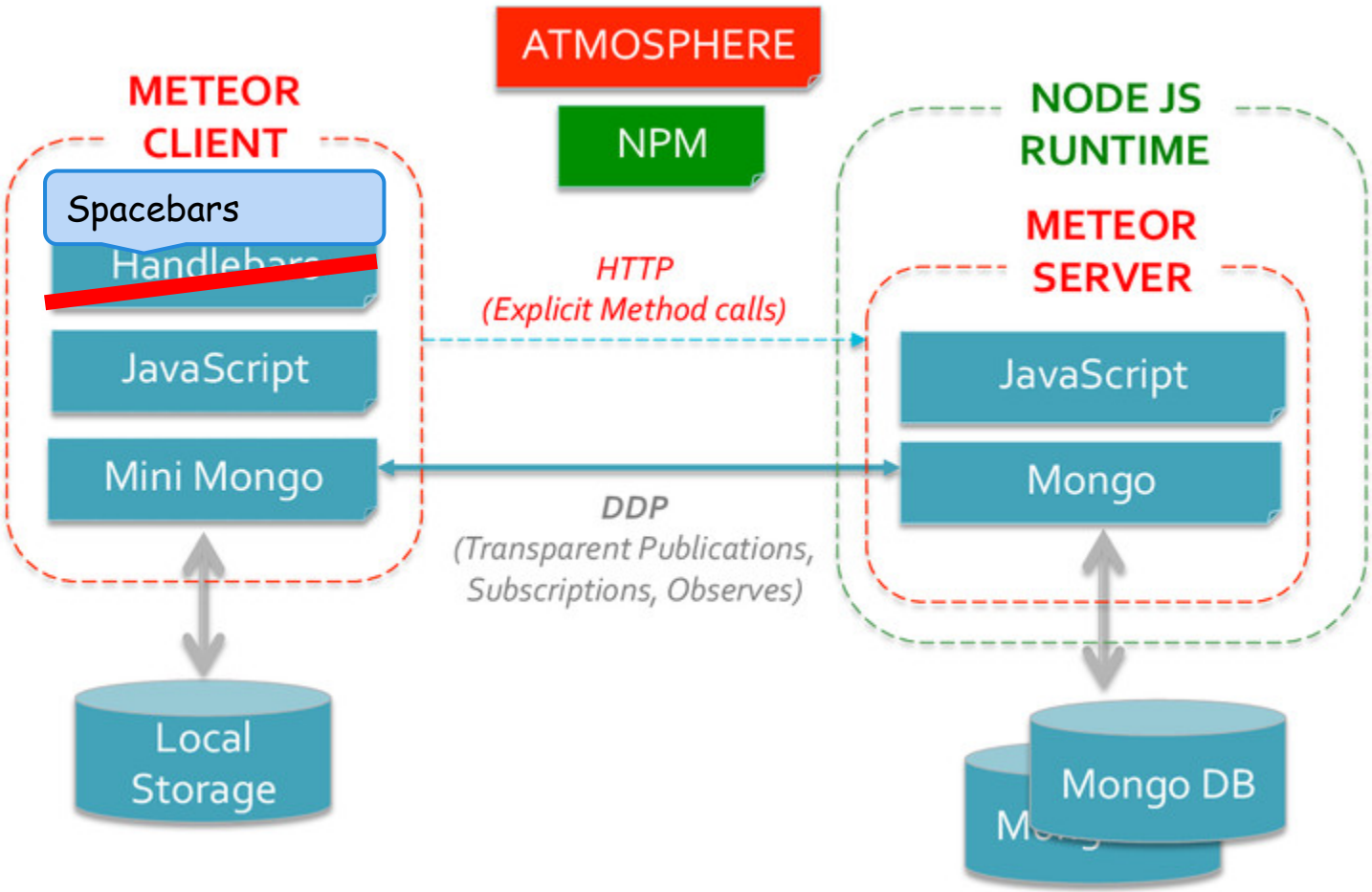
```
$ meteor install-sdk ios
$ meteor add-platform ios
$ meteor run ios
```

Finally, **deploy**!

```
$ meteor deploy chatroom-demo.meteor.com
```


SHUT UP AND
TAKE MY MONEY!

Meteor: Under the Hood



Meteor also Automagically:

- incorporates jQuery and underscoreJS
- compiles Coffeescript
- scopes your JS code by files
  - `var foo = bar; //foo is file scoped`
  - `foo2 = bar; //foo2 is app scoped`
- minifies and concatenates CSS and JS
- preserves the UI state (form inputs, scroll position, selection) b/t hot reloads

Useful Packages (out of 43,000+)

```
$ meteor add accounts-facebook
```

- accounts-twitter
- accounts-google
- accounts-linkedin

```
$ meteor add
meteoric:ionic-
sass
```

http://ionicframework.com/

Caution: Meteor is NOT a silver bullet

- New technology:
  - became 1.0 in October/2014
  - yet to see wide adoption as RoR
- Excellent for building responsive Web App
- Excellent for rapid prototyping
- Does Meteor Scale?
  - Factors to consider: routing, sessions, job processing, static file serving, and database
- How to Scale Meteor?

Thank You!

Questions?

Bonus: Does Websocket Scale?

- 620K idle concurrent websocket connections on only 1 m3.xlarge (4 core, 15GB RAM) EC2 instance.*

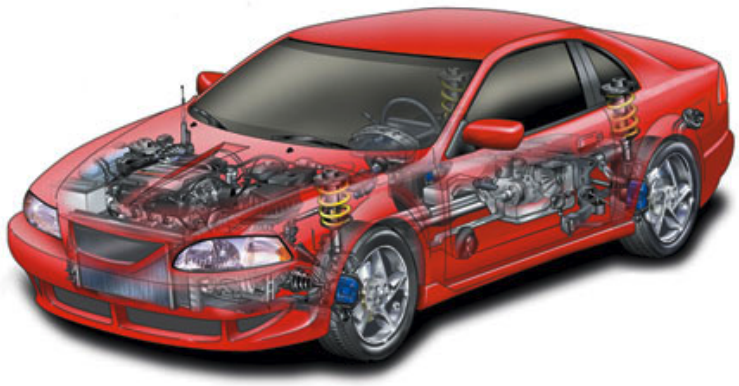
- Does your idea/business scale?

*http://www.jayway.com/2015/04/13/600k-concurrent-websocket-connections-on-aws-using-node-js/

Bonus: The MEAN stack vs. Meteor

Meteor gives you a lot more out of the box. The client and the server communicate data updates seamlessly and automatically, without you having to write any boilerplate data sync code.

The MEAN stack



Meteor.js

The MEAN stack is just **M**ongoDB, **E**xpress, **A**ngular and **N**ode.js bundled together, but there's nothing seamless about it. You have to do all the wiring-up yourself between MongoDB and Node.js, between Express and Angular, create REST endpoints and consume them etc. - all this just to get a

http://wiki.dandascalescu.com/essays/meteor_js_vs_the_mean_stack

Bonus: Meteor Project Structure

- **.meteor**: meteor project info, do not need modify it.
- **client**: all logics here are visible on the browser only.
- **lib**: all logic here will be loaded before anything.
- **private**: visible on server side only.
- **public**: for static content
- **server**: server side logic, not available on the browser.