



You come with just an idea -
we make great software for you!

Trust
Teamwork
Transparency



Isomorphic Javascript: new silver bullet

Vitaliy Medvedev, Arcadia



As true as steel to your desire

About me

Vitaliy Medvedev

Leading Software Engineer at JSC “Arcadia Inc.”

5+ years of experience in software development

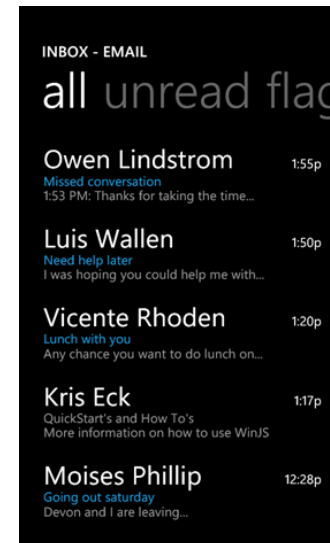
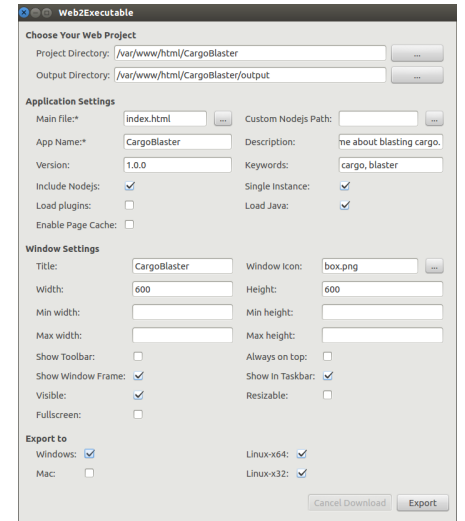
vitaliy.medvedev@arcadia.spb.ru

<http://twitter.com/imevs>

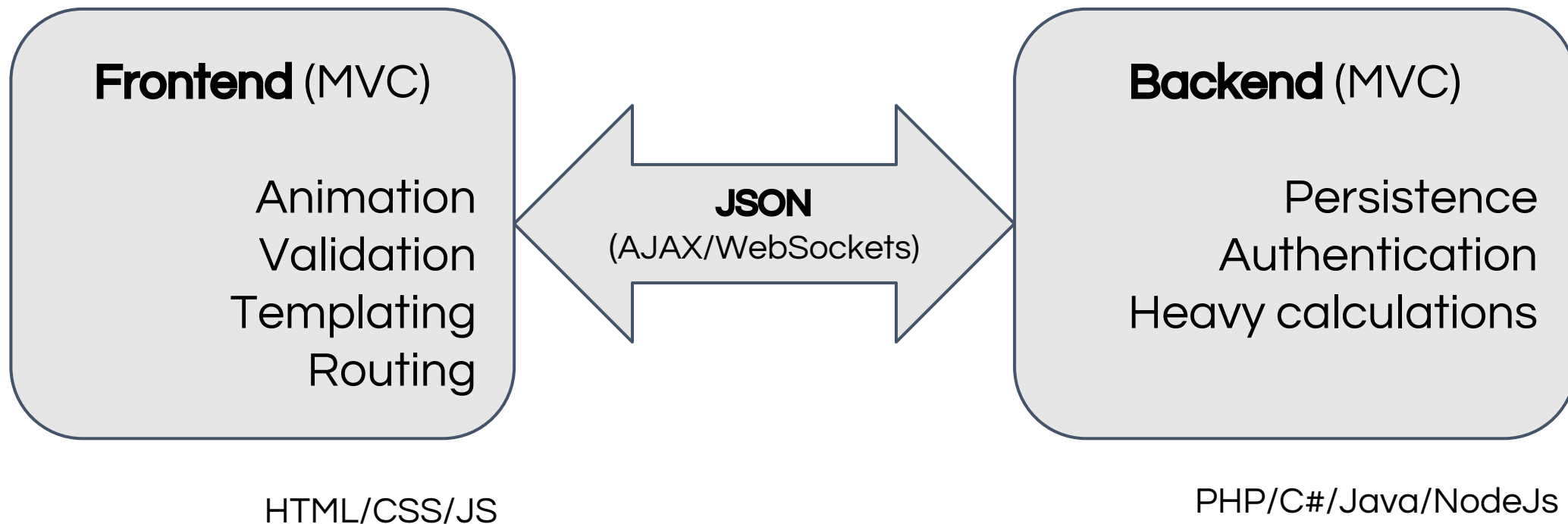


Where JS works

- client
- server
- desktop
- mobile
- quadcopters



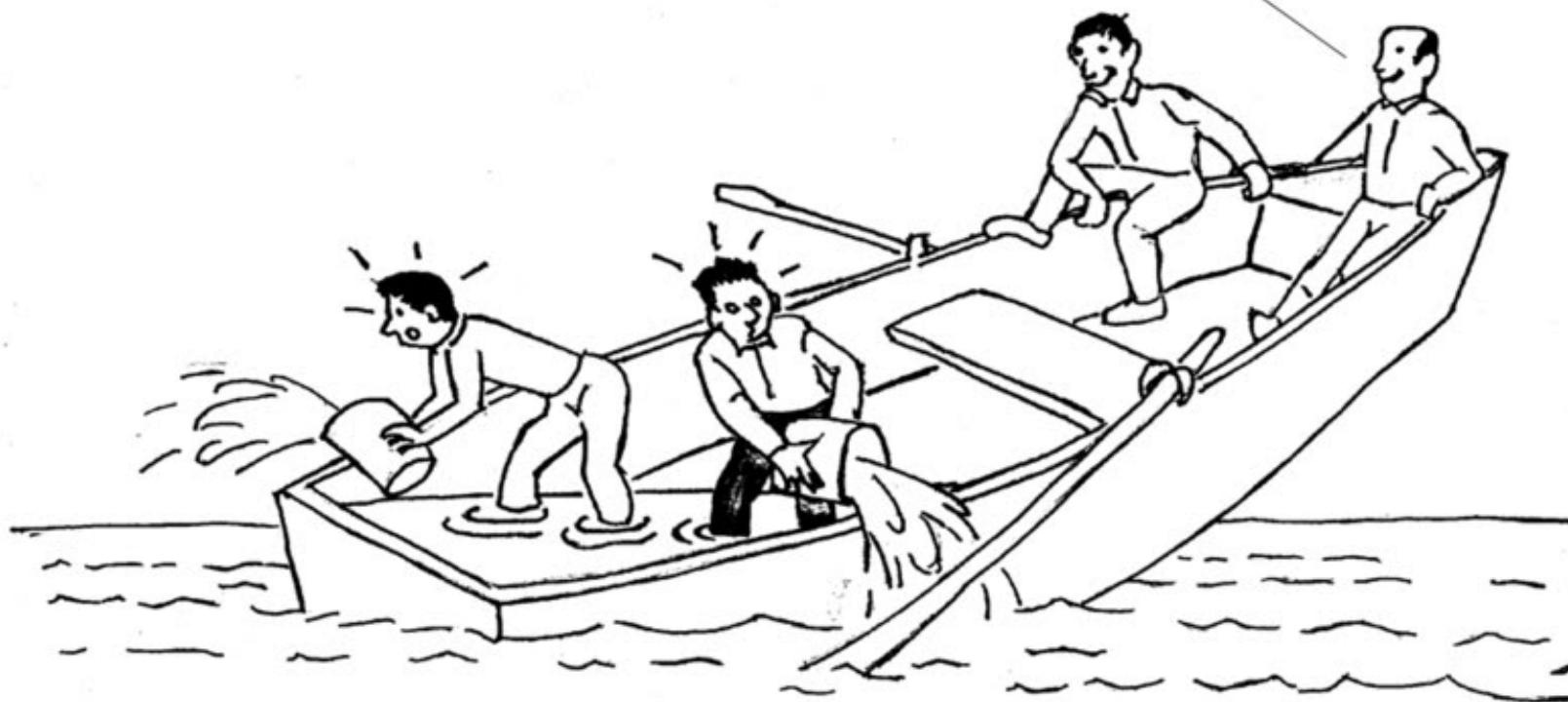
Single page application (SPA)



SPA problems

- loading time
- search engine indexing
- logic duplication
 - different languages - different ecosystems
 - cross-functional restrictions

Sure glad the hole isn't at our end.



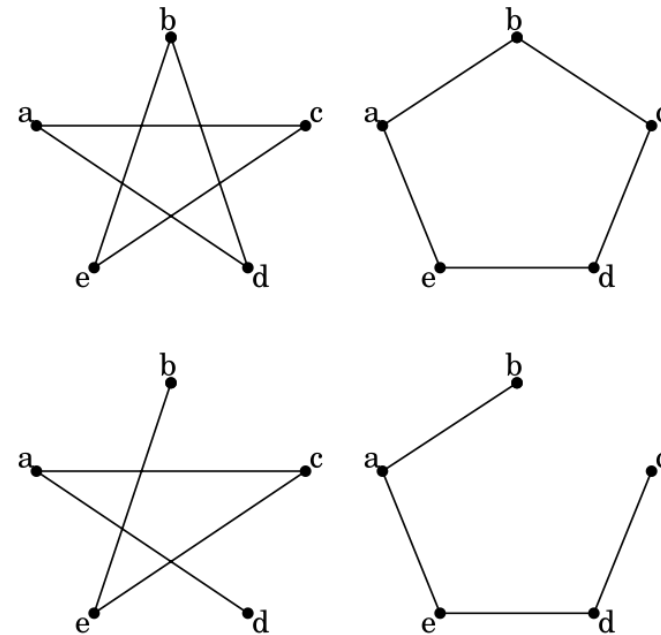
1. Solution!

What to do?

1. Write in the same language on both sides
2. Use same code on both sides

Isomorphic? WAT?!

- cross-platform
- multi-platform
- heteromorphic
-



Who invented isomorphism?

History:

1. 18 Oct 2011

Scaling Isomorphic Javascript Code

2. 01 Apr 2012

Yahoo! Mojito Framework

3. 08 Nov 2013

The future of web apps is — ready? — isomorphic JavaScript



2. Let's do isomorphism!

Using scripts



```
var module = require('./jquery');
```



```
<script src="script.js" [async] [defer]></script>
```



```
require(['jquery'], function ($) {  
    // do smth  
})
```

Differences between client & server JS

- Global objects
global / window
- Different available api
localStorage, historyApi, WebGL, Canvas, FileSystem
- Differences in API behavior
cookies, websockets, webworkers

Differences between client & server JS

- Environment
- Script loading
- Available api
- Engine version

Diving into isomorphism

- Understand that you can write as isomorphic
- Code kinds
 - Code that should not work at **frontend** side
 - Code that should not work at **backend** side
 - **Independent** of environment
 - **Dependent** of environment

Diving into isomorphism

- Abstraction from environment
browserify, typescript
- Environment specific unit tests
karma, mocha, phantomjs
- Static code analysis
jshint, jshint, eslint, tslint

Code before isomorphism

```
1 // Browser
2 function browserSum (a, b) { return a+b; }
3
4 // NodeJs
5 module.exports.serverSum = function(a,b) {
6     return a+b;
7 };
```

JavaScript

Simple isomorphic module

```
1 (function(exports) {  
2     exports.isomorphicSum = function(a, b) {  
3         return a + b;  
4     };  
5 })(typeof exports ? exports : window);|
```

JavaScript

What is isomorphic code good for?

1. Templating (handlebarsjs)
2. Business logic:
 - validation
 - filtering
 - calculations
3. Using common libraries (jquery, underscore)

Isomorphic libraries

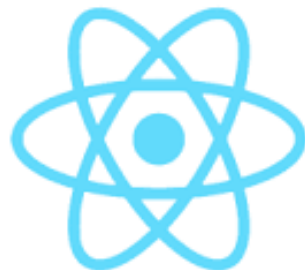
- jquery
- underscore
- backbonejs
- lodash
- cryptojs
- handlebarsjs
- async

3. What's next?

Full stack javascript frameworks



Meteor



React

Profit

- ✓ decrease loading time
- ✓ search engine indexing
- ✓ decrease logic duplication
- ✓ better code design
- ✓ code consistency

Useful links

Learning resources:

1. <http://venturebeat.com/2013/11/08/the-future-of-web-apps-is-ready-isomorphic-javascript/>
2. <http://www.slideshare.net/spikebrehm/a-28174727>
3. <http://isomorphic.net/>
4. <http://codewinds.com/podcast/009.html>
5. <https://github.com/spikebrehm/isomorphic-tutorial>
6. <http://jsfiddle.net/7o2mrby9/>

Production examples:

1. <http://airbnb.com/>
2. <https://lever.co/>

Q & A