REACT

AND THE PATH TO SANITY DRIVEN DEVELOPMENT

HI THERE

MY NAME'S ANTÓNIO CAPELO

Frontend Engineer at Moxy

@antoniocapelo

let's

TIME TRAVEL

a bit

Remember this?

Typical flow:

Browser asks for page

Server fetches data from DB

Server fills templates with data

Full markup is returned

Need some UI interactions?

Add a bunch of **JavaScript files**!

(back in 2010's...)

JAVASCRIPT

undefined is not a function

caps is not defined

cannot read property 'length' of undefined

...

OLD SCHOOL


prototype


jQuery
write less do more


mootools


dojo


Ext JS


script.aculo.us


Y!

MAIN PROBLEMS

- BIG PAYLOAD
- TEMPLATING LANGUAGES WITH SOME LIMITATIONS
- *PAIR* OUR JAVASCRIPT LOGIC WITH THE MARKUP
- WEAK (OR ABSENT) MODULE/COMPONENT SYSTEM
- BAD/LACK OF DEPENDENCY MANAGEMENT

Screenshot from the final part of a fellow developer's SO question:

And here's proof that I'm calling all of the files (I think):

```
<script src="/assets/jquery.js?body=1" type="text/javascript"></script>
<script src="/assets/jquery_ujs.js?body=1" type="text/javascript"></script>
<script src="/assets/jquery-ui.js?body=1" type="text/javascript"></script>
<script src="/assets/underscore.js?body=1" type="text/javascript"></script>
<script src="/assets/backbone.js?body=1" type="text/javascript"></script>
<script src="/assets/backbone-support/support.js?body=1" type="text/javascript"></sc
<script src="/assets/backbone-support/composite_view.js?body=1" type="text/javascrip
<script src="/assets/backbone-support/swapping_router.js?body=1" type="text/javascri
<script src="/assets/backbone-support.js?body=1" type="text/javascript"></script>
<script src="/assets/example_app.js?body=1" type="text/javascript"></script>
<script src="/assets/easing.js?body=1" type="text/javascript"></script>
<script src="/assets/modernizr.js?body=1" type="text/javascript"></script>
<script src="/assets/models/task.js?body=1" type="text/javascript"></script>
<script src="/assets/collections/tasks.js?body=1" type="text/javascript"></script>
<script src="/assets/views/task_view.js?body=1" type="text/javascript"></script>
<script src="/assets/views/tasks.js?body=1" type="text/javascript"></script>
<script src="/assets/views/tasks_index.js?body=1" type="text/javascript"></script>
<script src="/assets/routers/tasks.js?body=1" type="text/javascript"></script>
<script src="/assets/tasks/index.js?body=1" type="text/javascript"></script>
<script src="/assets/tasks/task.js?body=1" type="text/javascript"></script>
<script src="/assets/application.js?body=1" type="text/javascript"></script>
```

He's clearly not 100% sure if all of the 1000 necessary js files are being pulled

PAIRING STUFF?

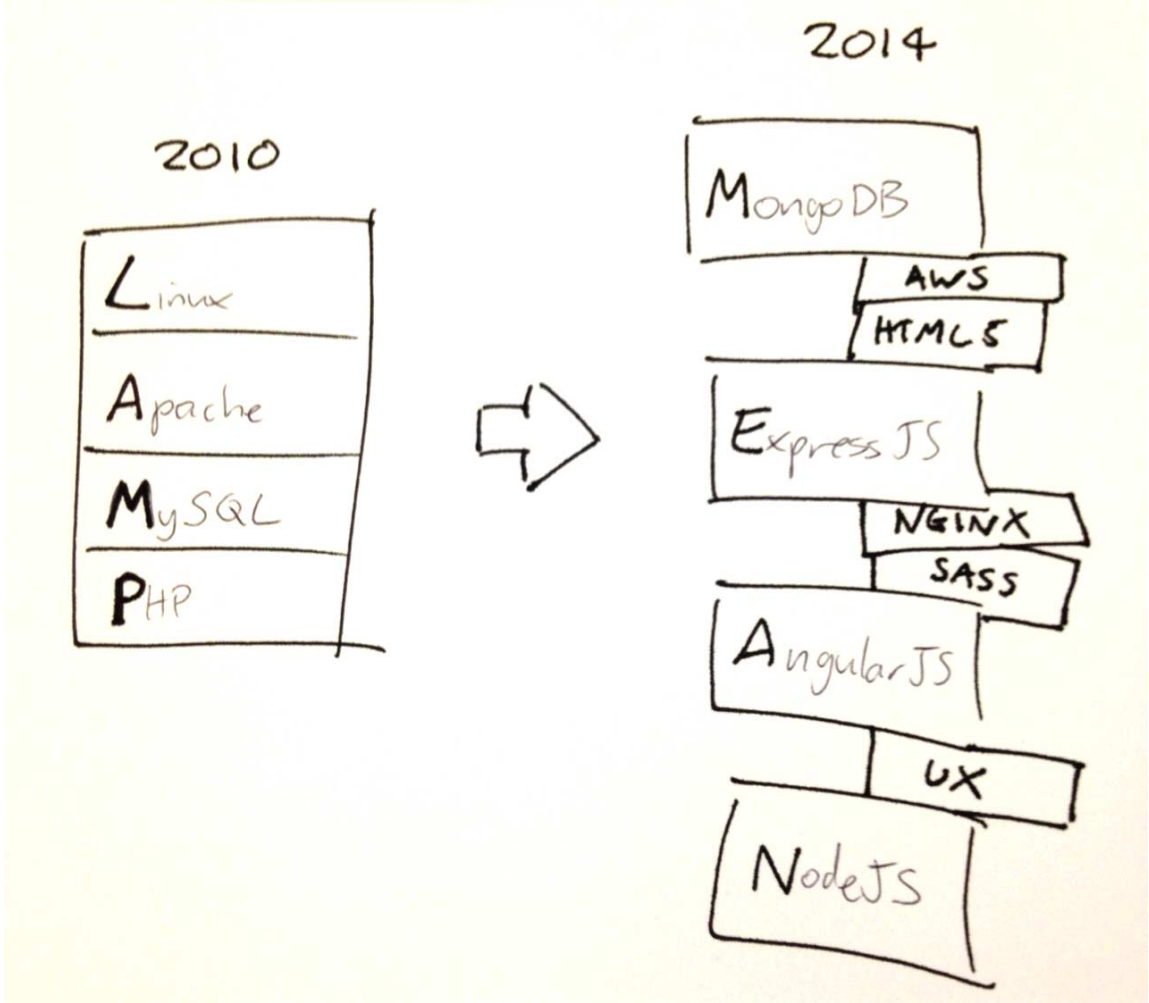
When rendering a simple list, we had to implement methods like:

- render()
- update();
- delete();

all these were 'glued' to the markup by class names, data attributes, etc

CLIENT SIDE

NEW STACKS


2010
Linux
Apache
MySQL
PHP
2014
MongoDB
AWS
HTML5
ExpressJS
NGINX
SASS
AngularJS
UX
NodeJS

NEW PLAYERS

The flow becomes:

Browser asks for page

Server returns minimal markup

Client (JS) picks up and does all the templating and additional requests

Pros

- ++ DEV. SPEED
- PROGRESSIVE LOADING
- RICHER INTERACTIONS
- QUICK RENDERING

Cons

- -- SEO
- FOC
- INITIAL LOAD MIGHT REQUIRE MORE TIME
- JS APPS BECOME MORE AND MORE COMPLEX

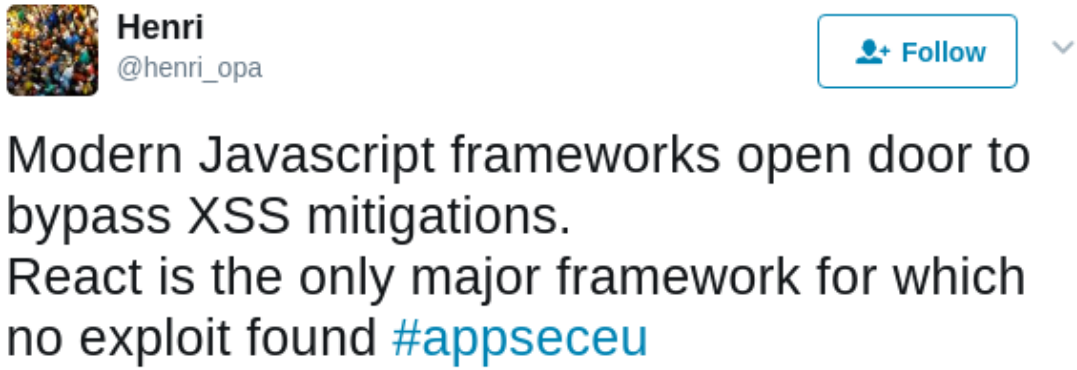
WHY REACT?

Wins

✔ Great adoption -> Strong dev community

✔ Great developer tools

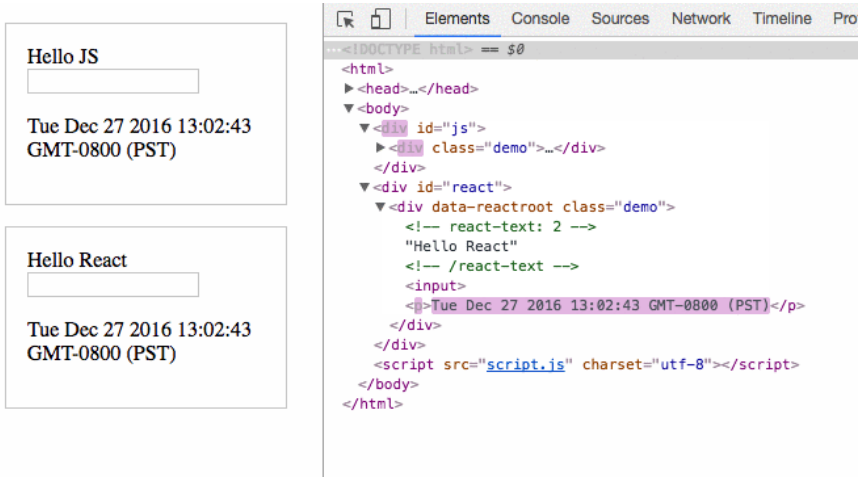
✔ Security features



Wins

✔ Understanding how react works -> React Native becomes a lot easier

✔ It's quick, thanks to the virtual DOM and *Tree Reconciliation*



Wins

JSX is cool

You don't need to learn a new templating syntax

```
import ListItem from './ListItem';
...
render() {
    return (
        <ul>
            { items.map(item => <ListItem key={ item.id } item={ ite
        </ul>
    );
}
```

Wins

✔ It's a library for building UI, not a framework

✔ Server Side Rendering

Plus...

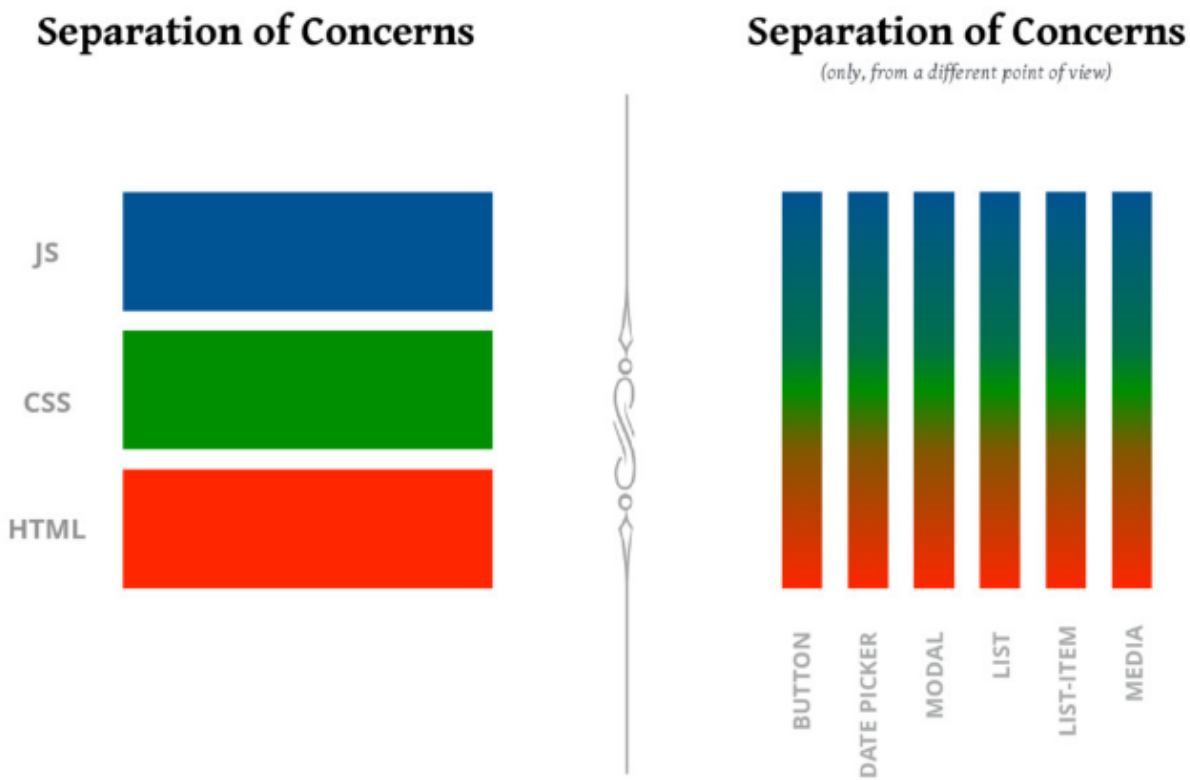
IT'S JUST FUNCTIONS

```
(props) => markup || more functions
```

Simplest React Component:

```
const ListItem = ({ name, id }) => (
    <li>
        { id } -{ name }
    </li>
);
```

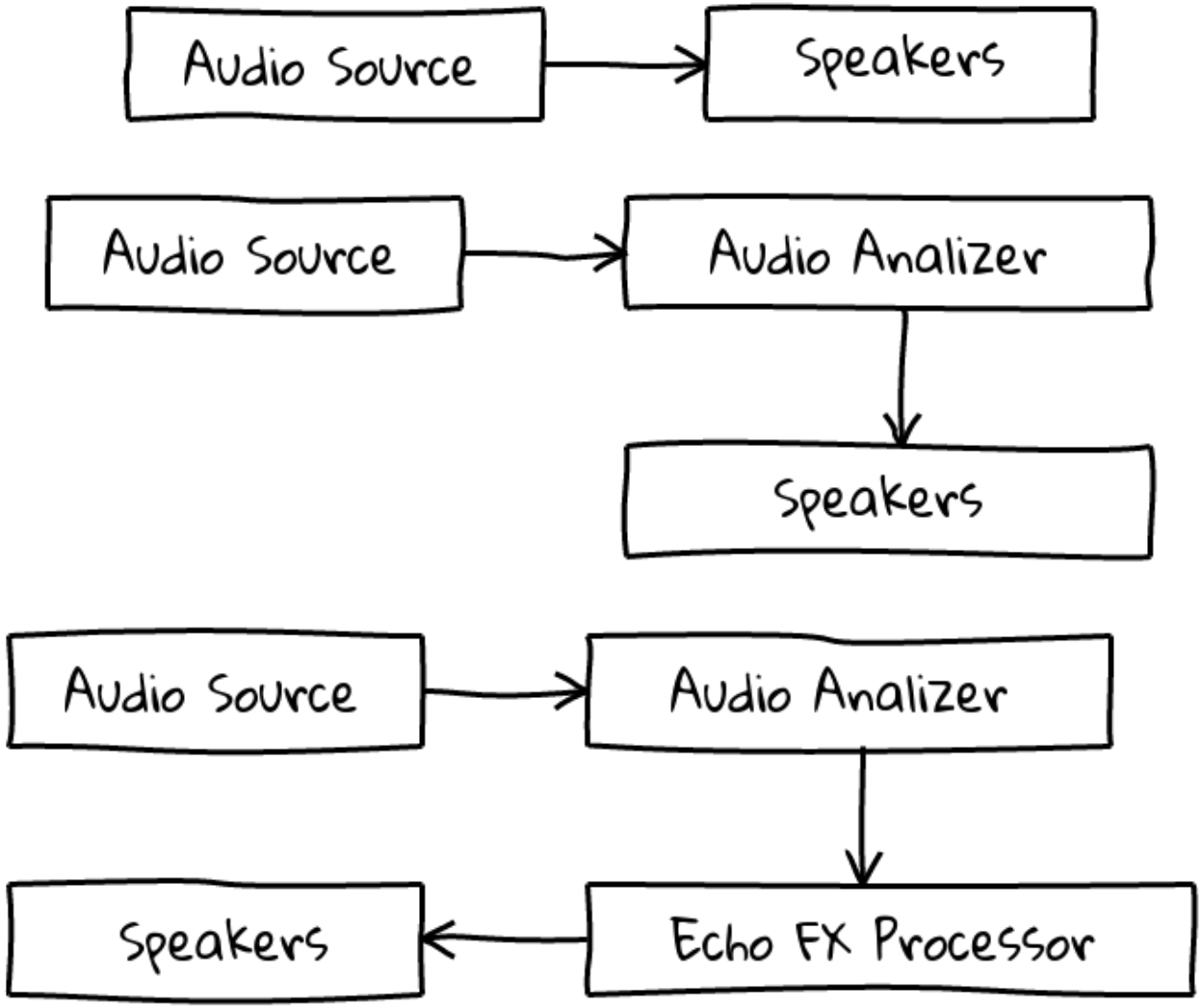
DEMO TIME



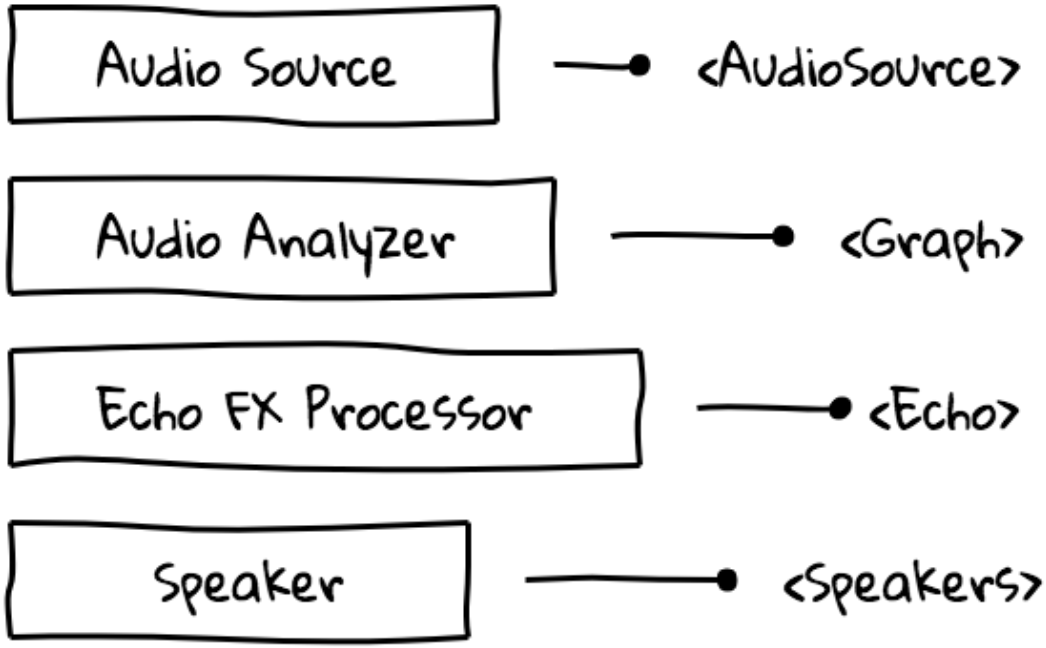
Great slide from Cristiano Rastelli's talk

COMPOSABILITY

COMPOSING (AUDIO) COMPONENTS



TRANSLATION TO REACT COMPONENTS



DEMO TIME

Thank you for listening!

QUESTIONS ?

@antoniocapelo