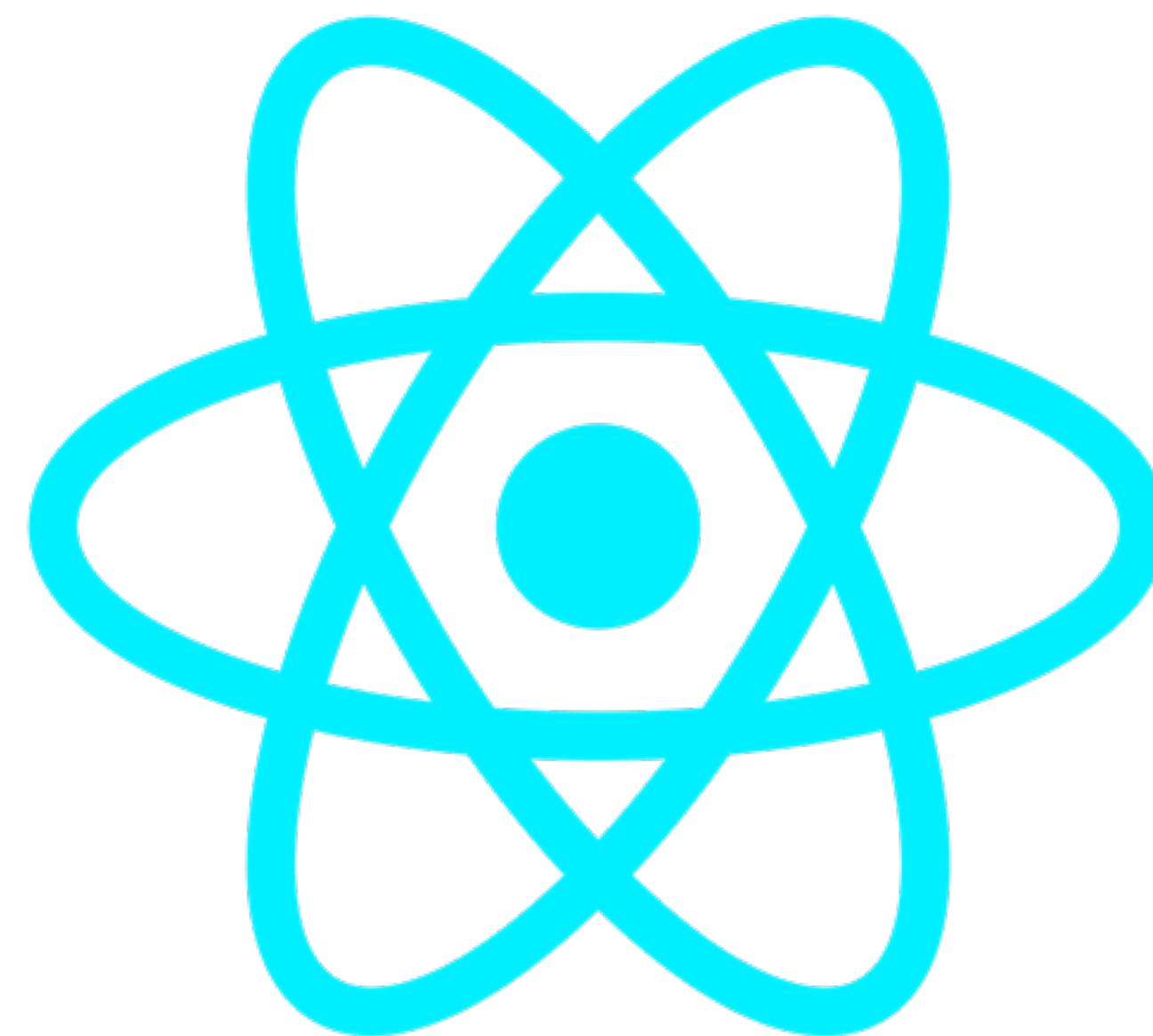




SINGLE-PAGE APPS

JACK TOLD US "WHY"

HOW



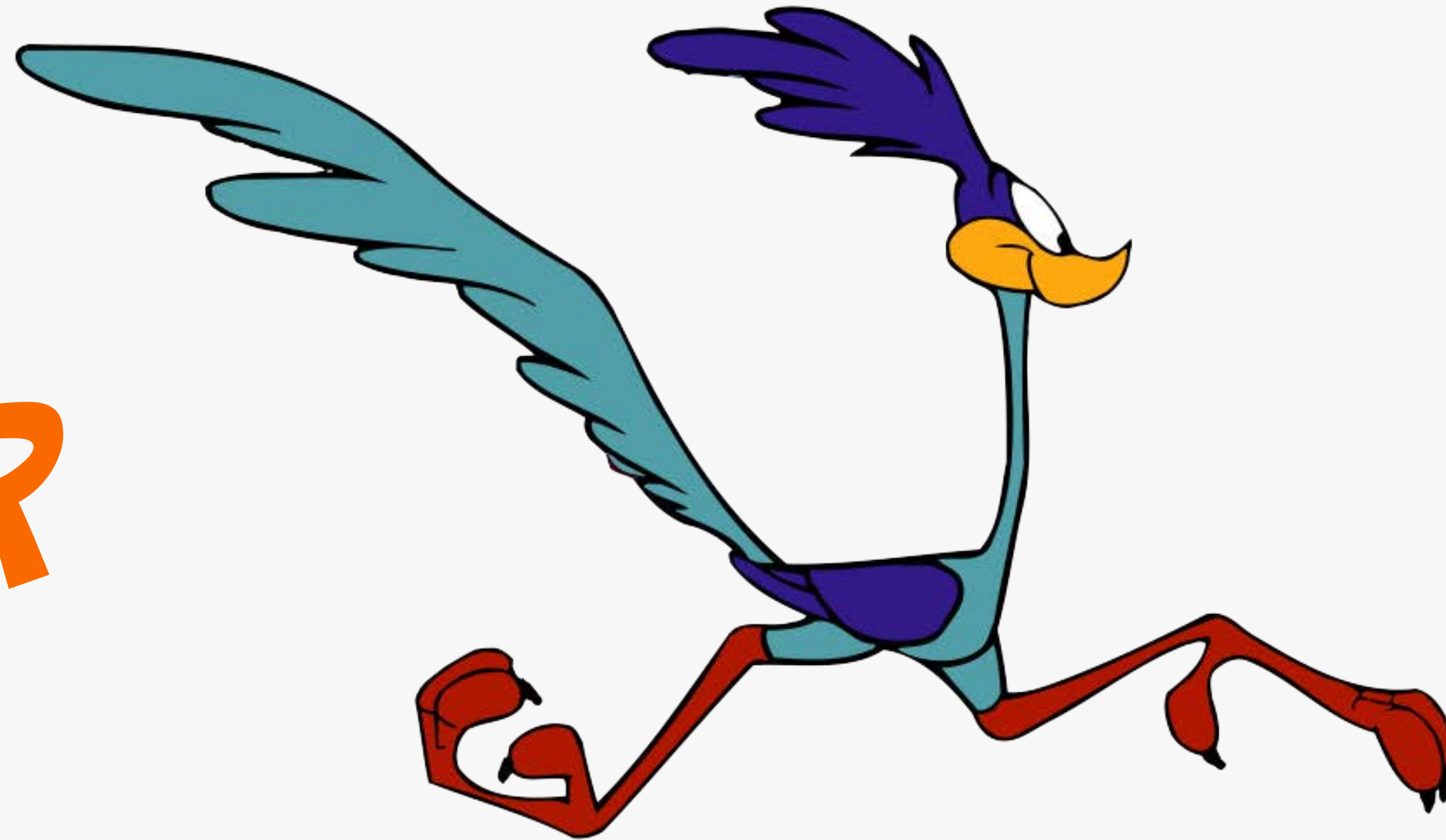
NIKOLAY BACHIYSKI, AUTOMATTIC, WORDCAMP VIENNA, APRIL 11, 2015

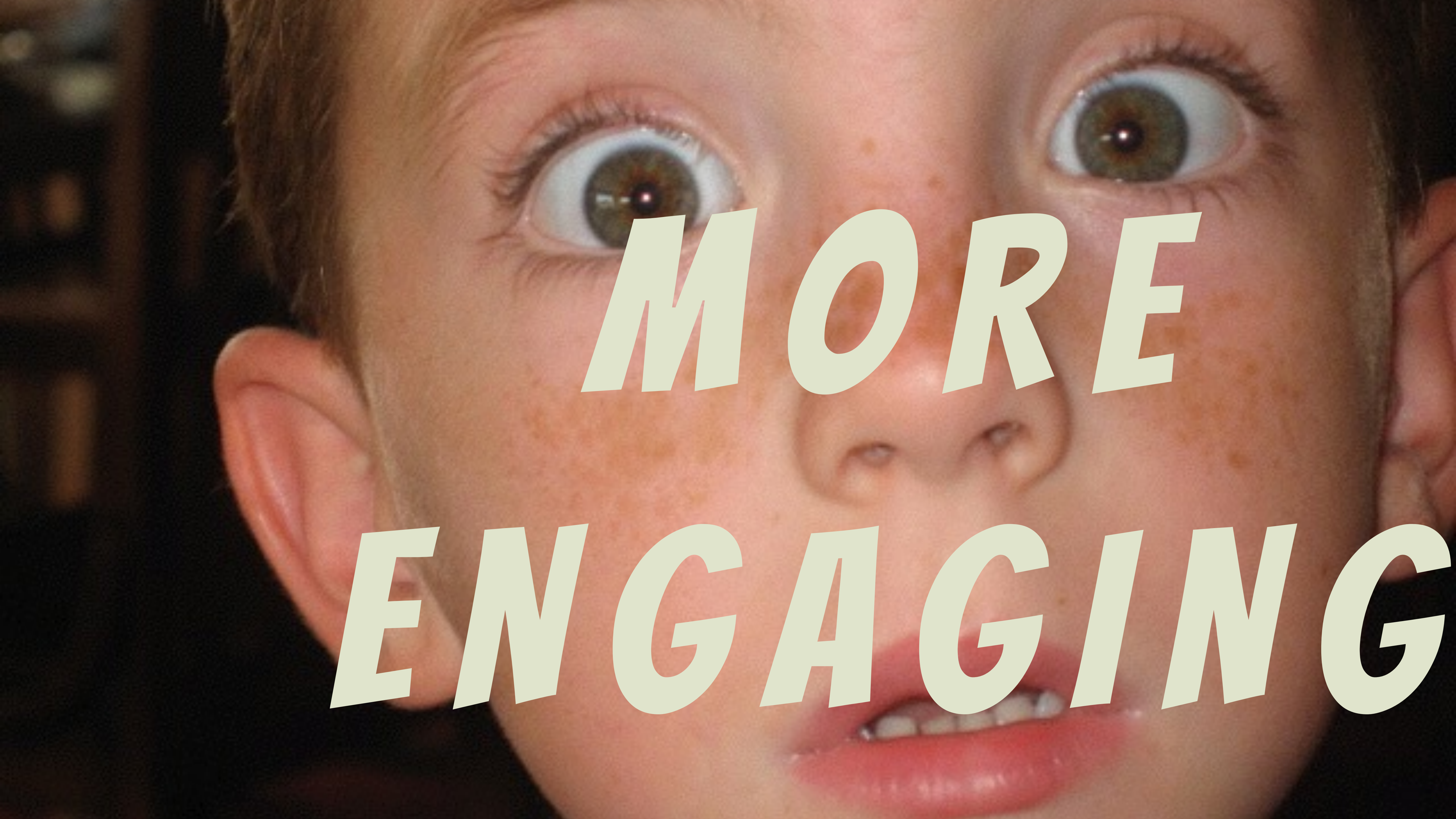
AUTOMATTIC

EXTRAPOLATE.ME

@NIKOLAYB

FASTER





MORE

ENGAGING

LET'S TALK ABOUT

CHANGE



We hate the new profile page...please get it back!

Facebook
thousa
start Fa
the New

By DANIEL BATES and
UPDATED: 20:04 GMT, 22



I hate FB Timeline, and want to disable it ASAP

Like

Timeline

About

Photos

Likes

Videos

PEOPLE

22,210 likes

Post

Photo / Video

Write something on the

POLICE TENSION

SCIENCE OF YOU

CGI

VIDEOS

SECTIONS

ELITE DAILY

90% Hate Facebook's Timeline:
Controversial Feature Is Supported By
One In Ten Users

I Hate the New Facebook
Profile
Community

Timeline

About

Photos

PEOPLE

16,965 likes

Post

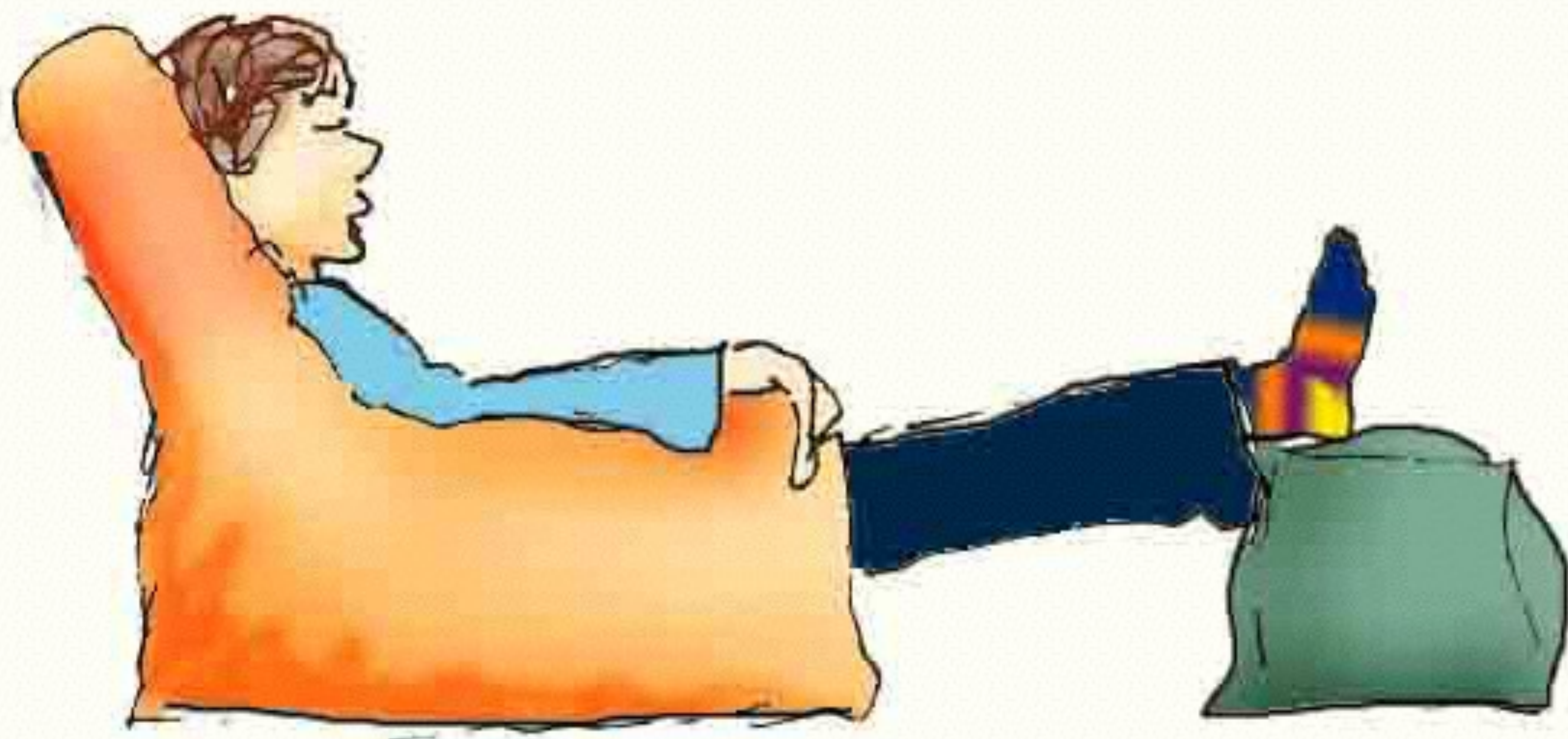
Photo / Video

Write something on the

LET'S TALK ABOUT

STATE





OLD-SCHOOL

VS.

SINGLE-PAGE APPS

Visual	Text
	Sales increased steadily from 2010 to 2015, reaching a peak of 10 million dollars in 2015.
	The number of books sold was highest in the Mystery genre (90) and lowest in the Non-Fiction genre (60).
	The Sales department accounts for the largest portion of the company's revenue (40%).

Text

B *I* ABC ☰ ☱ ☲ ☳ ☴ ☵ ☶ ☷ “ — ≡ ≡ ≡ 🔗 🔒 ☰ ☱ ☲ ☳ ☴ ☵ ☶ ☷

1

Preview

[Edit](#)[Edit](#)[Edit](#)

Publish

1

Standard

○

Aside

 Image

 **Video**

○

Quote

 [Link](#)

○

 Gallery

Status

○

 Audio

Chat

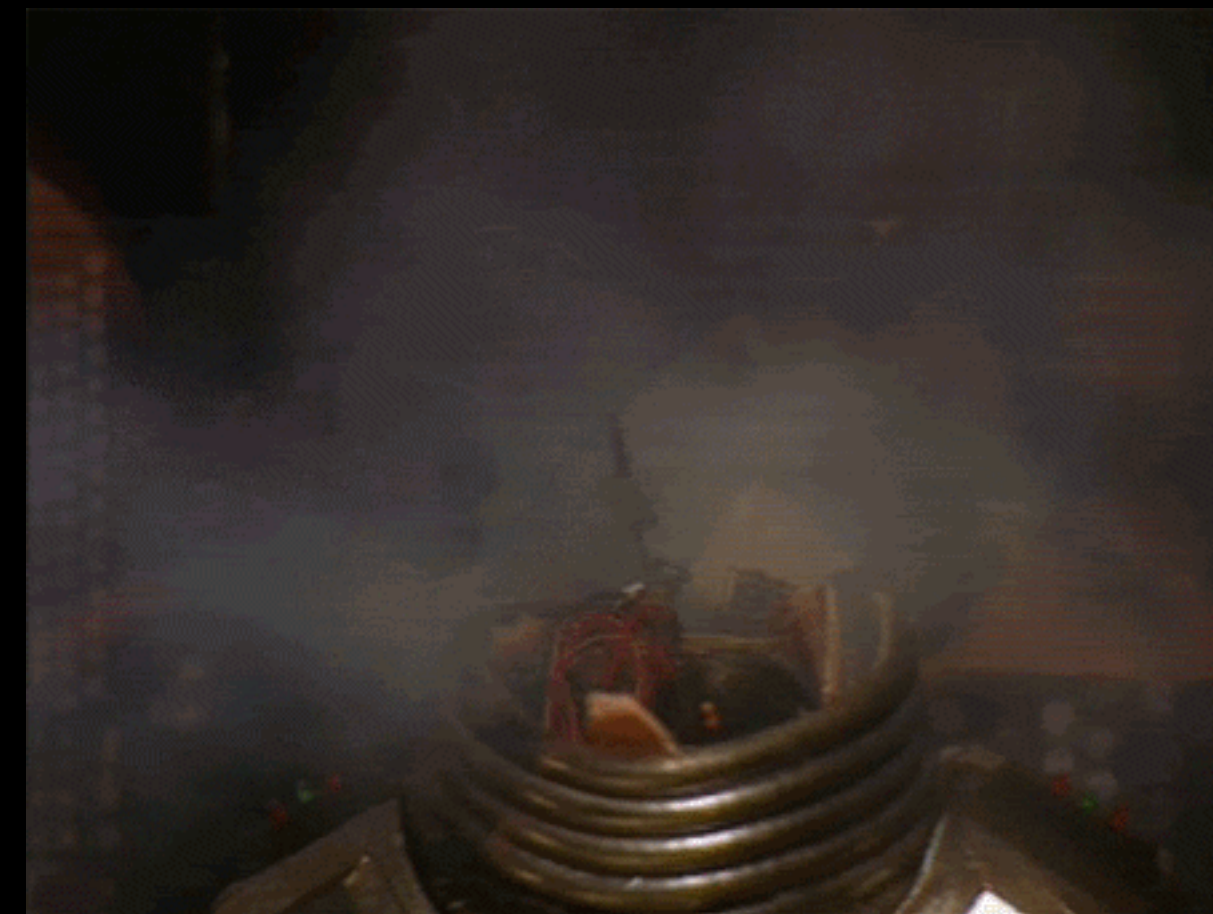
Custom Fields

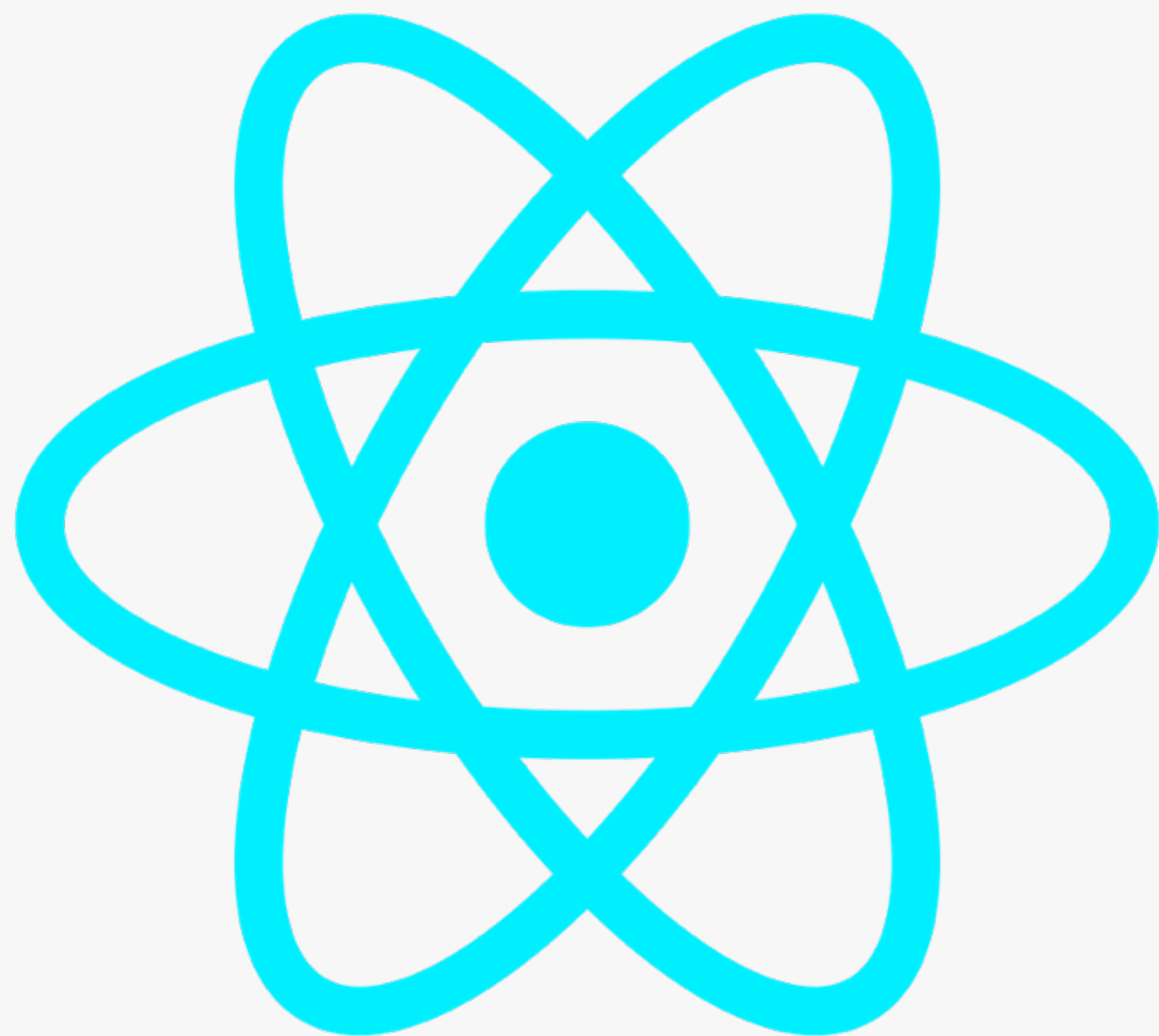
OLD-SCHOOL

CHANGE ⇒ REFRESH

SINGLE-PAGE

CHANGE ⇒

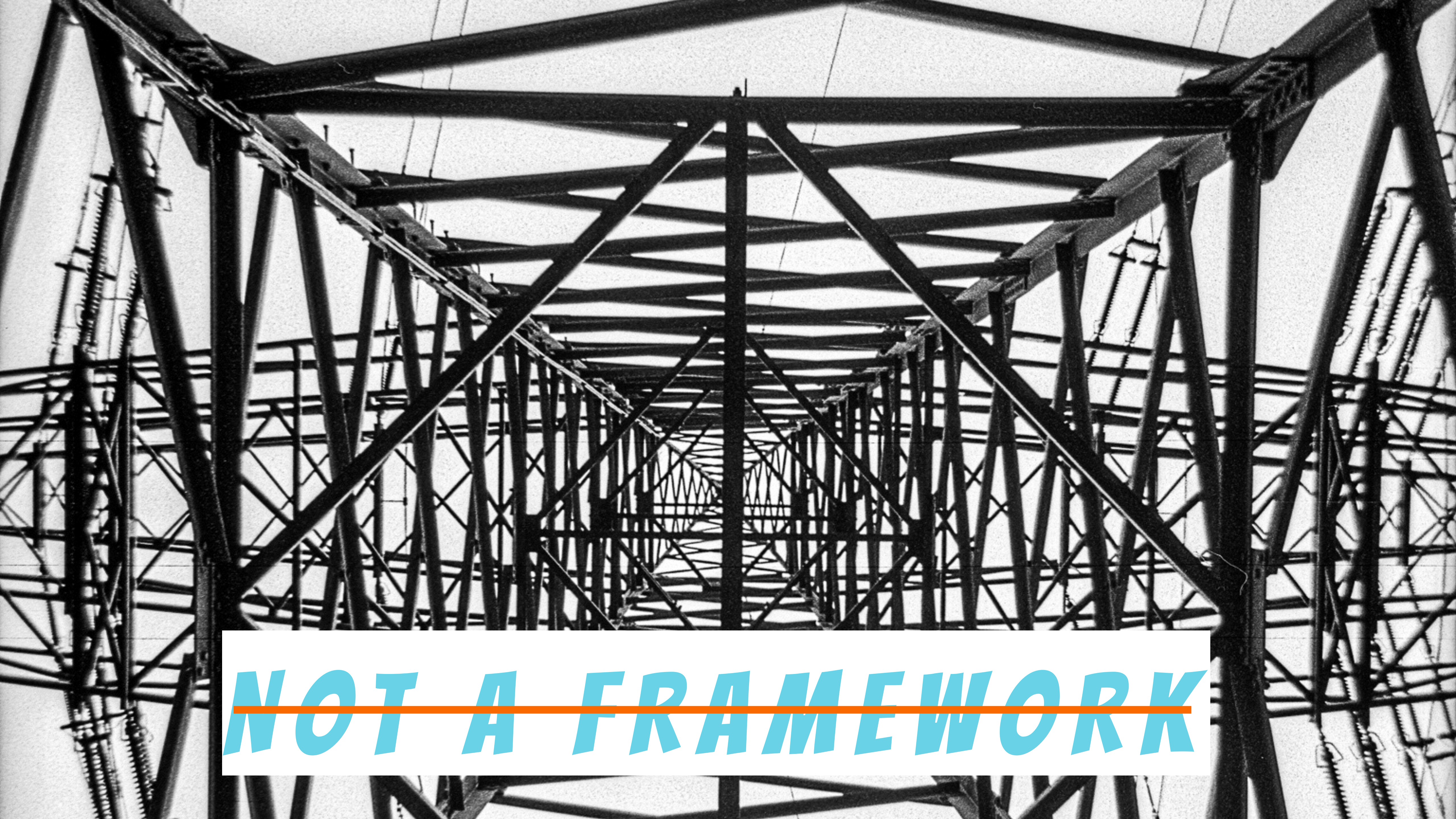








UI LIBRARY



NOT A FRAMEWORK

```
var Post = React.createClass( {  
  function render() {  
    return (  
      <div className="post">  
        <p className="content">  
          { this.props.post.content }  
        </p>  
        <PostMeta post={ this.props.post } />  
      </div>  
    );  
  }  
} );
```

```
var Post = React.createClass( {  
  function render() {  
    return (  
      <div className="post">  
        <p className="content">  
          { this.props.post.content }  
        </p>  
        <PostMeta post={ this.props.post } />  
      </div>  
    );  
  }  
});
```

JSX

```
var Post = React.createClass( {  
  function render() {  
    return (  
      React.DOM.div(  
        {className: 'post'},  
        [ React.DOM.p(  
          null,  
          this.props.post.content  
        ),  
        PostMeta( {post: this.props.post} )  
      )  
    );  
  }  
} );
```

```
var Post = React.createClass( {  
  function render() {  
    return (  
      <div className="post">  
        <p className="content">  
          { this.props.post.content }  
        </p>  
        <PostMeta post={ this.props.post } />  
      </div>  
    );  
  }  
} );
```

```
var PostMeta = React.createClass( {
  function getInitialState() {
    return { hidden: true };
  },
  function onToggle() {
    this.setState( { hidden: ! this.state.hidden } );
  },
  function render() {
    return (
      <div className="meta">
        { this.props.post.author }
        <button onClick={ this.onToggle }>
          { this.state.hidden ? 'More' : 'Hide' }
        </button>
        ( this.state.hidden ?
          <p>...extra meta...</p> :
          null
        )
      </div>
    );
  }
} );
```

```
var PostMeta = React.createClass( {
  function getInitialState() {
    return { hidden: true };
  },
  function onToggle() {
    this.setState( { hidden: ! this.state.hidden } );
  },
  function render() {
    return (
      <div className="meta">
        { this.props.post.author }
        <button onClick={ this.onToggle }>
          { this.state.hidden ? 'More' : 'Hide' }
        </button>
        ( this.state.hidden ?
          <p>...extra meta...</p> :
          null
        )
      </div>
    );
  }
} );
```

```
var PostMeta = React.createClass( {
  function getInitialState() {
    return { hidden: true };
  },
  function onToggle() {
    this.setState( { hidden: ! this.state.hidden } );
  },
  function render() {
    return (
      <div className="meta">
        { this.props.post.author }
        <button onClick={ this.onToggle }>
          { this.state.hidden ? 'More' : 'Hide' }
        </button>
        ( this.state.hidden ?
          <p>...extra meta...</p> :
          null
        )
      </div>
    );
  }
} );
```

```
var PostMeta = React.createClass( {  
  function getInitialState() {  
    return { hidden: true };  
  },  
  function onToggle() {  
    this.setState( { hidden: ! this.state.hidden } );  
  },  
  function render() {  
    return (  
      <div className="meta">  
        { this.props.post.author }  
        <button onClick={ this.onToggle }>  
          { this.state.hidden ? 'More' : 'Hide' }  
        </button>  
        ( this.state.hidden ?  
          <p>...extra meta...</p> :  
          null  
        )  
      </div>  
    );  
  }  
} );
```

RE-RENDERS

LET'S TALK A BIT MORE ABOUT

COMPONENTS

COMPOSABLE

```
<MasterBar>  
  <UserGreeting user="...">  
    <Avatar login={ this.props.user.login } />  
    Howdy, { this.props.user.nickname }  
    <LogOutLink id={ this.props.user.id } />  
  </UserGreeting>  
</MasterBar>
```

***BUILDING BLOCKS OF
OUR APPLICATION***

```
<MasterBar />
<MastHead />
<Navigation active="archive" />
<Posts date="2015-04-01" author="crumb">
  <Post>
    ...
  </Post>
  <Post>
    ...
  </Post>
  ...
</Posts>
```

```
<MasterBar />
<MastHead />
<Posts date="2015-04-01" author="crumb">
  <Post post="...">
    <p class="content">
      { this.props.post.content}
    </p>
    <PostMeta post="..." />
  </Post>
  <Post>
    ...
  </Post>
  ...
</Posts>
```

[illegible]

TESTABLE

PREVENTS XSS

```

```



```
<p>{ this.props.content }</p>
```



```
&lt;img src=&quot;javascript:alert(&#39;XSS&#39;);&quot;&gt;
```

```
<p dangerouslySetInnerHTML={{ __html: this.props.content }} />
```



NOT TEMPLATES

```
<div class="posts">
  {{#each posts}}
    <div class="post">
      {{#if author}}
    </div>
  {{/each}}
</div>
```

REUSABLE

```
<Comment>  
  <Avatar email={ this.props.comment.author.email} />  
  ...  
</Comment>
```

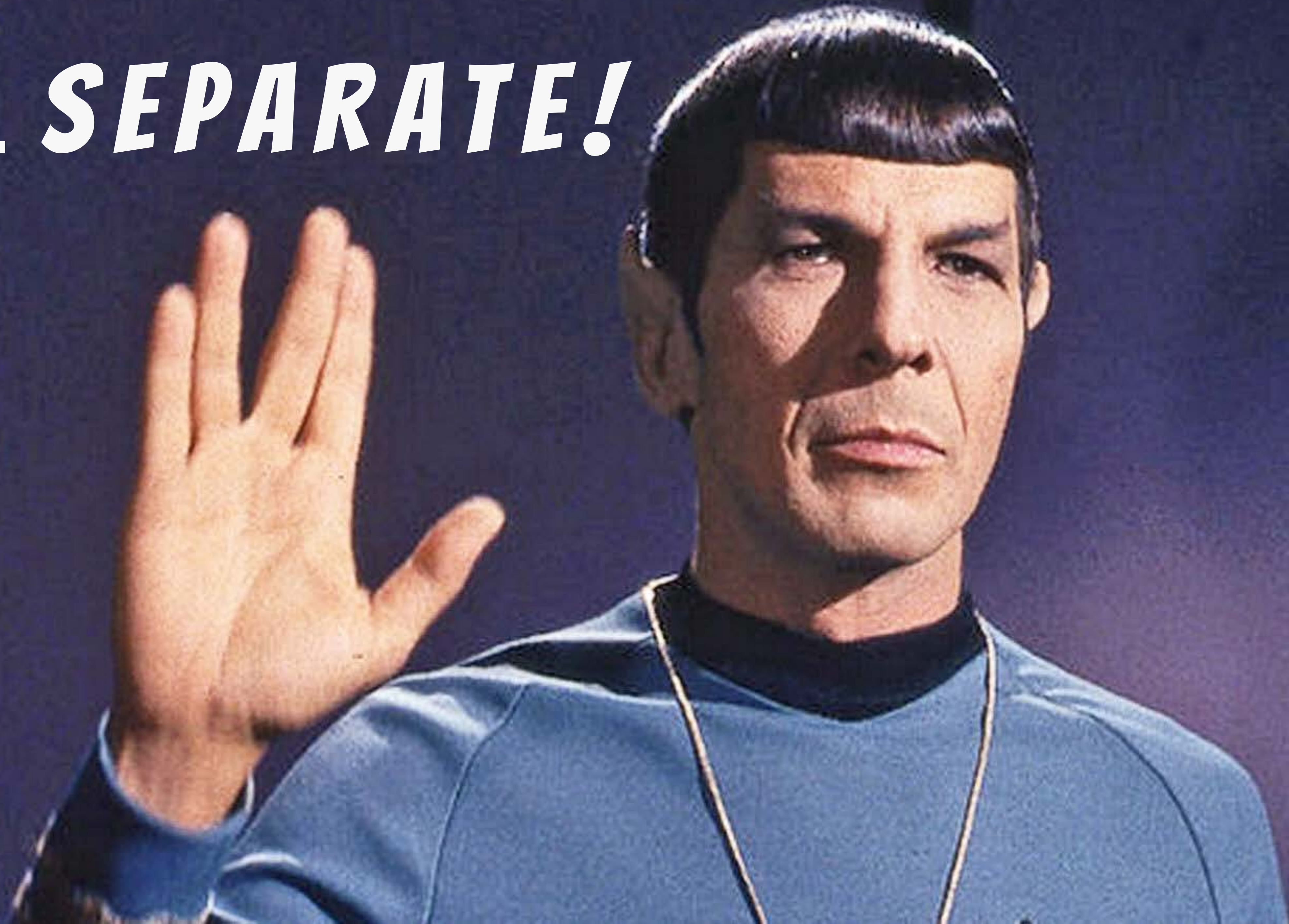
```
<PostMeta>  
  <Avatar email={ this.props.comment.author.email} />  
  ...  
</PostMeta>
```

WAIT!!

A POPULAR BEST PRACTICE SAYS...

KEEP LOGIC AND MARKUP...

... *SEPARATE!*



***SEPARATION OF CONCERNS
IS GOOD, RIGHT?***

onClick()



***WHEN DO WE NEED AN EVENT HANDLER
WITHOUT THE DOM ELEMENT IT IS BOUND TO?***

<button>

onClick()



***WHEN DO WE NEED A DOM ELEMENT WITHOUT
THE EVENT HANDLERS FOR ITS ACTIONS?***

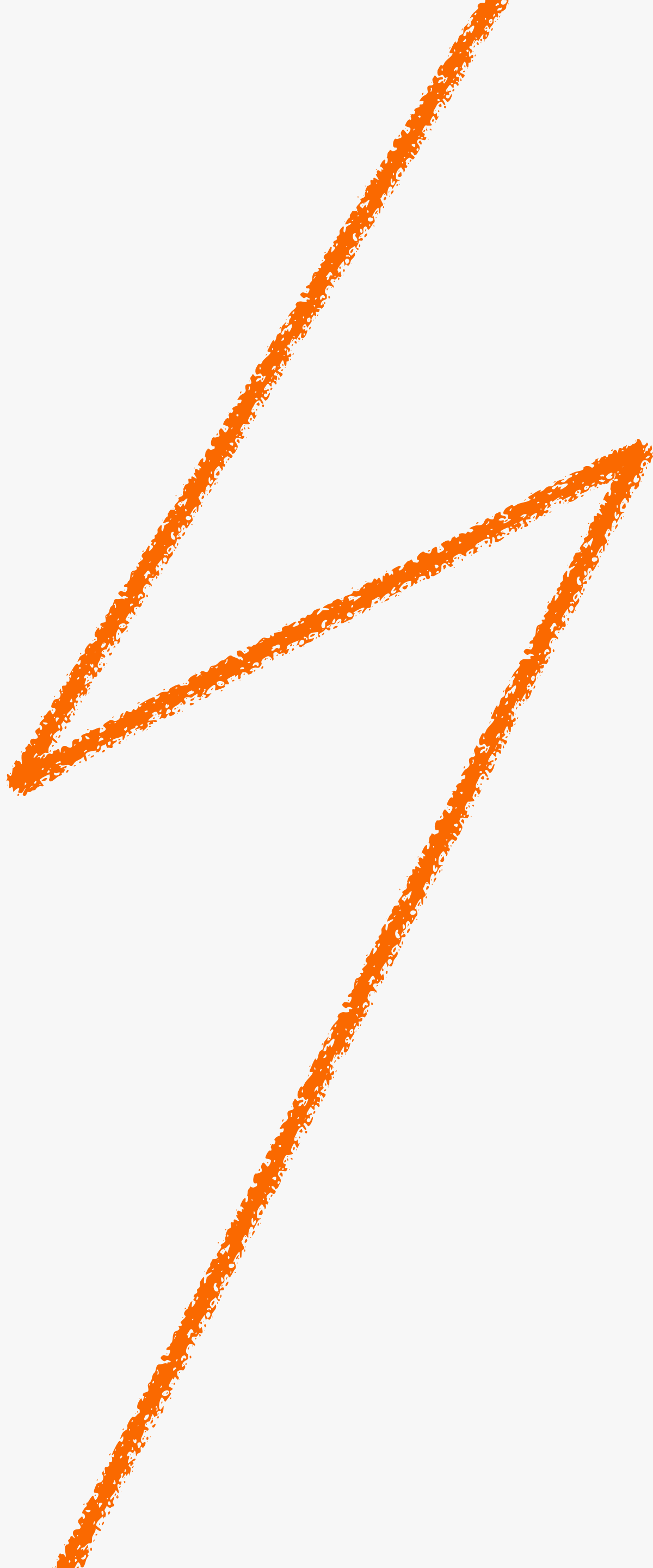
<button>

HTML

DISPLAY LOGIC

***THE SAME
CONCERN***

JS



HTML

***THE SAME
CONCERN***





RE-RENDERING





BeStaFvOnU2Be

THE ASPCA'S VET LIAISON

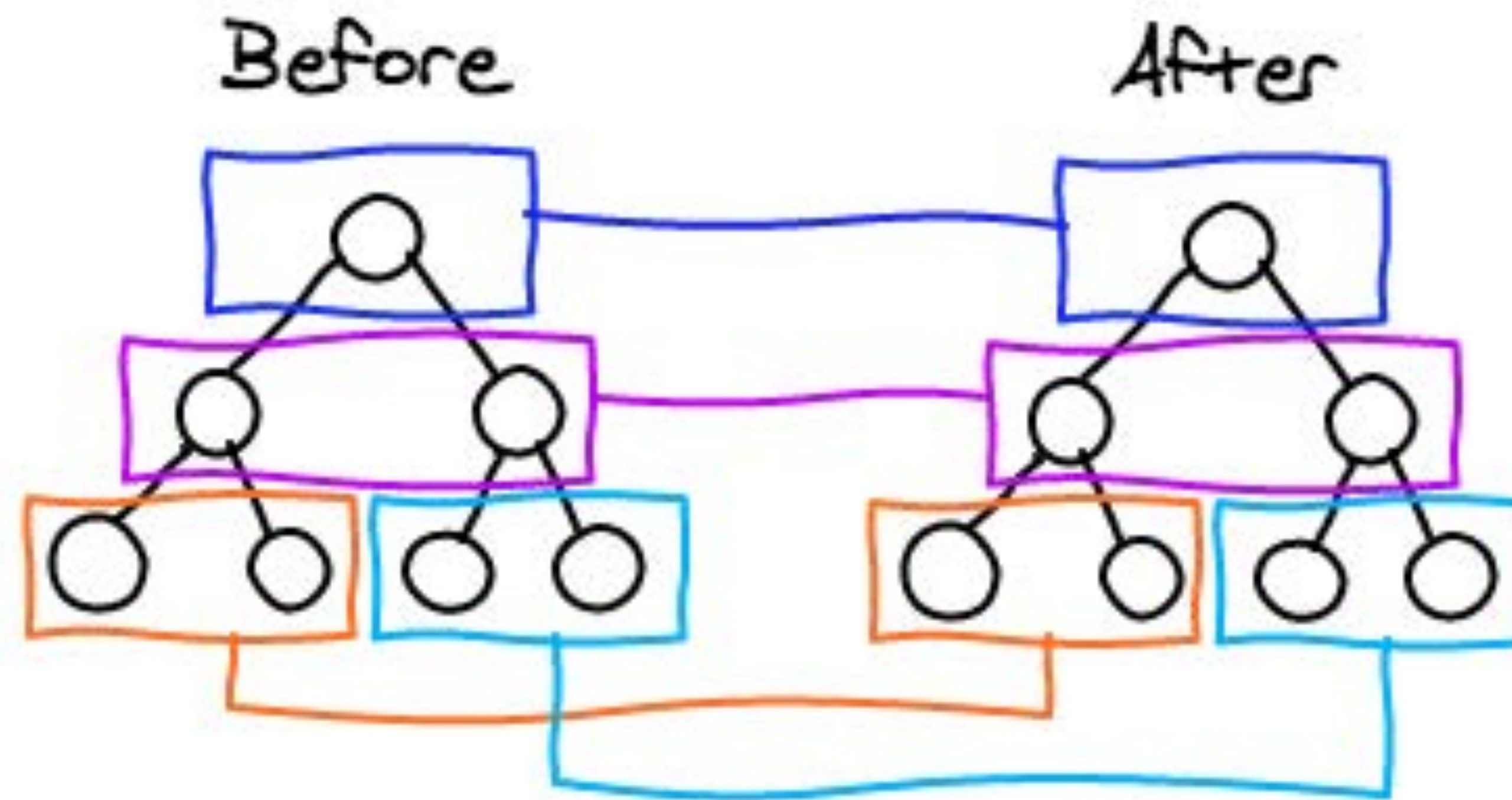
VIRTUAL DOM

ON EVERY UPDATE

0. RUN render()

ON EVERY UPDATE

1. DIFF WITH PREVIOUS TREE



<http://calendar.perfplanet.com/2013/diff/>

ON EVERY UPDATE

***2. COMPUTE MINIMUM SET
OF DOM CHANGES***

ON EVERY UPDATE

***3. BATCH-EXECUTE ALL
DOM UPDATES***

***VIRTUAL DOM:
NOT ONLY SPEED!***

SERVER-SIDE RENDERING

***SVG, VML,
CANVAS***

***RUNNING IN A
WEB WORKER***



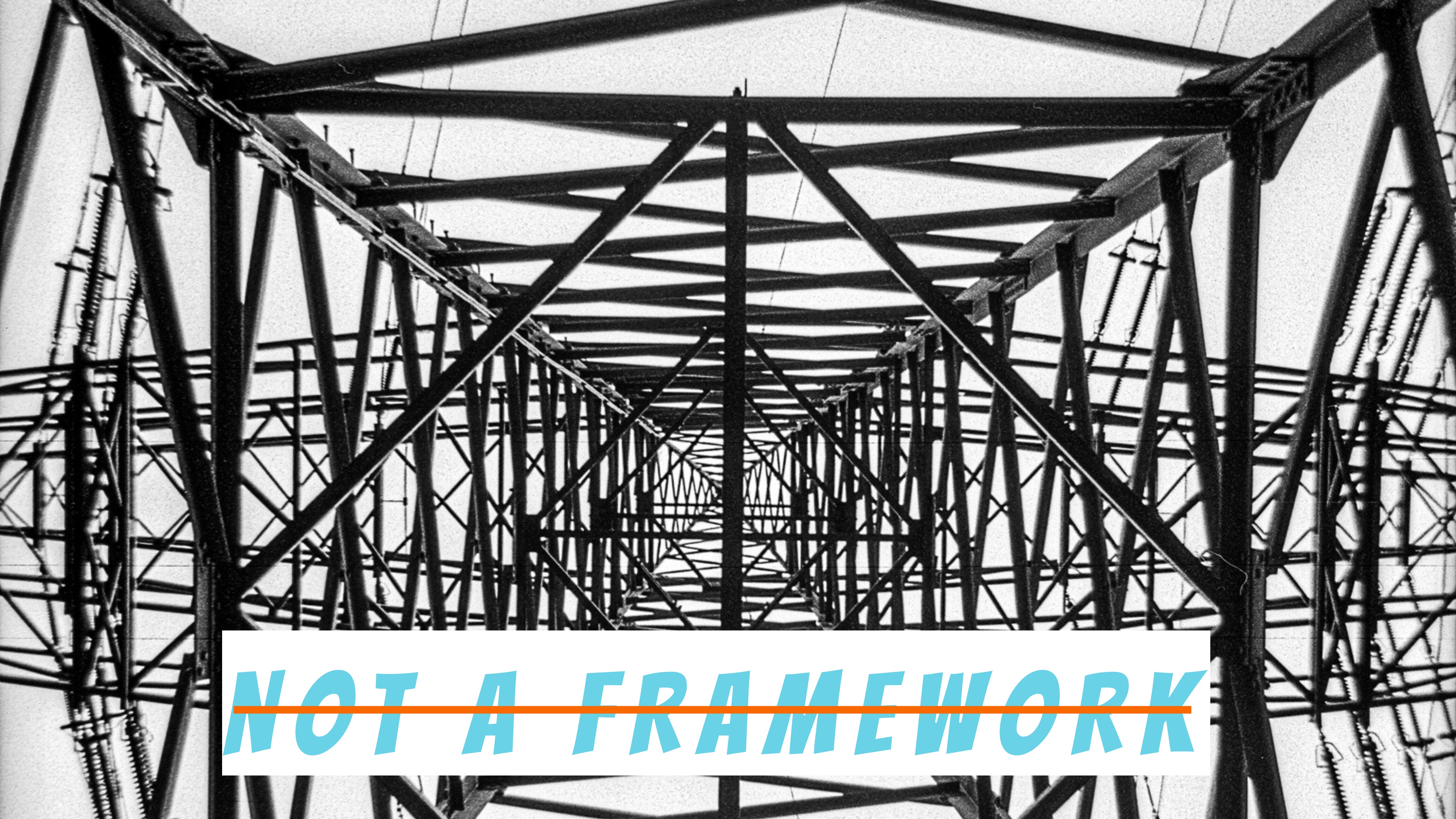


FLUX

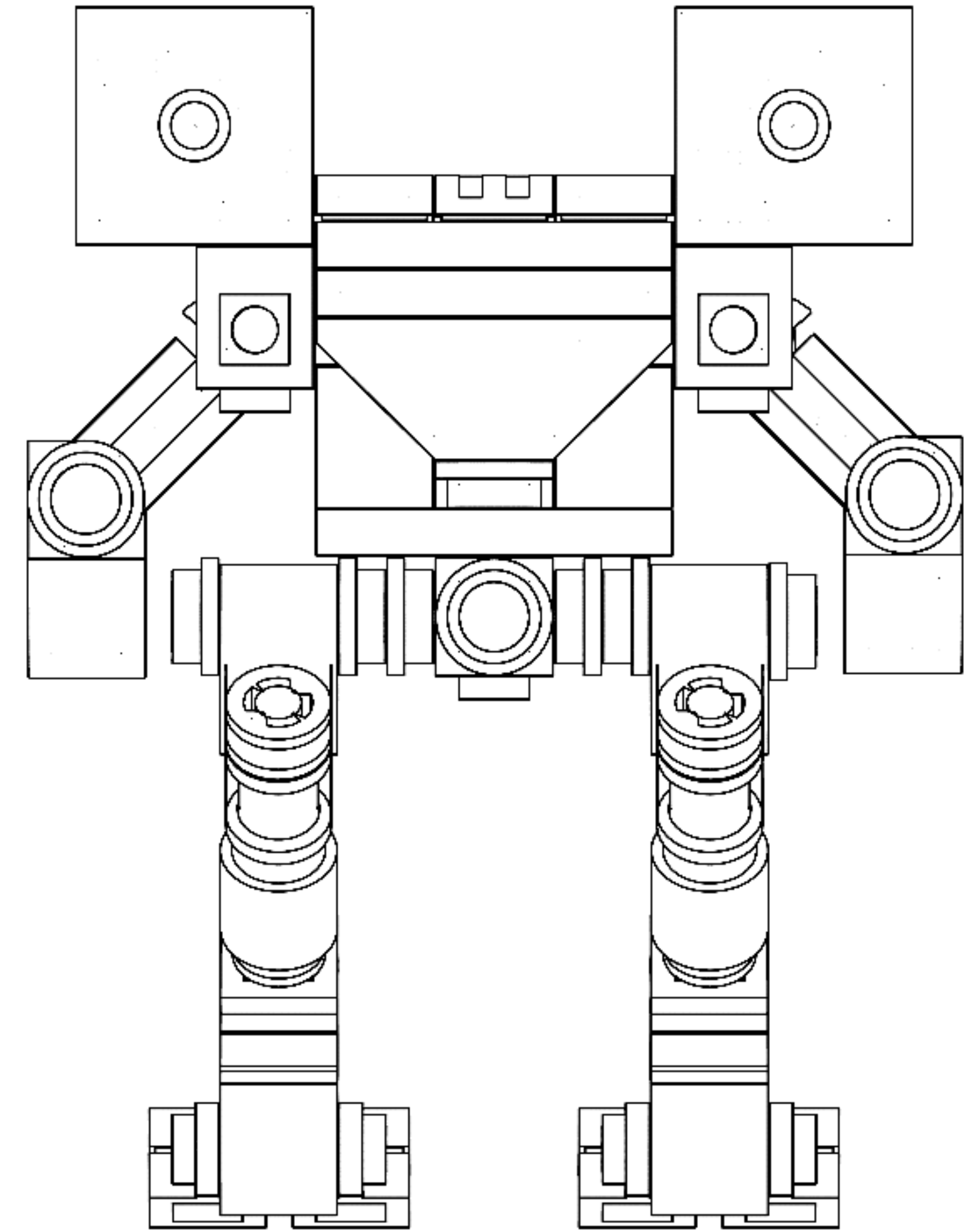
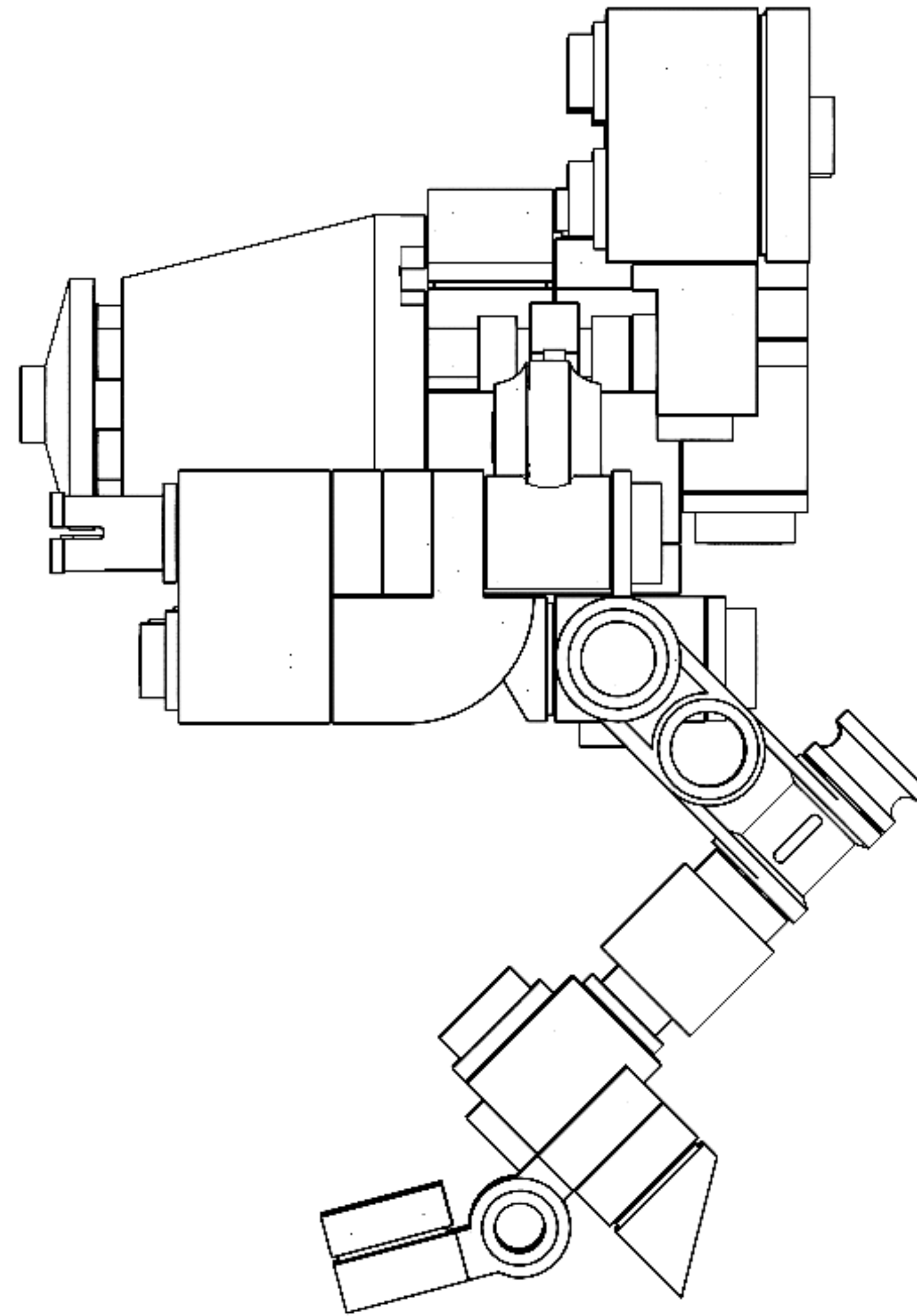
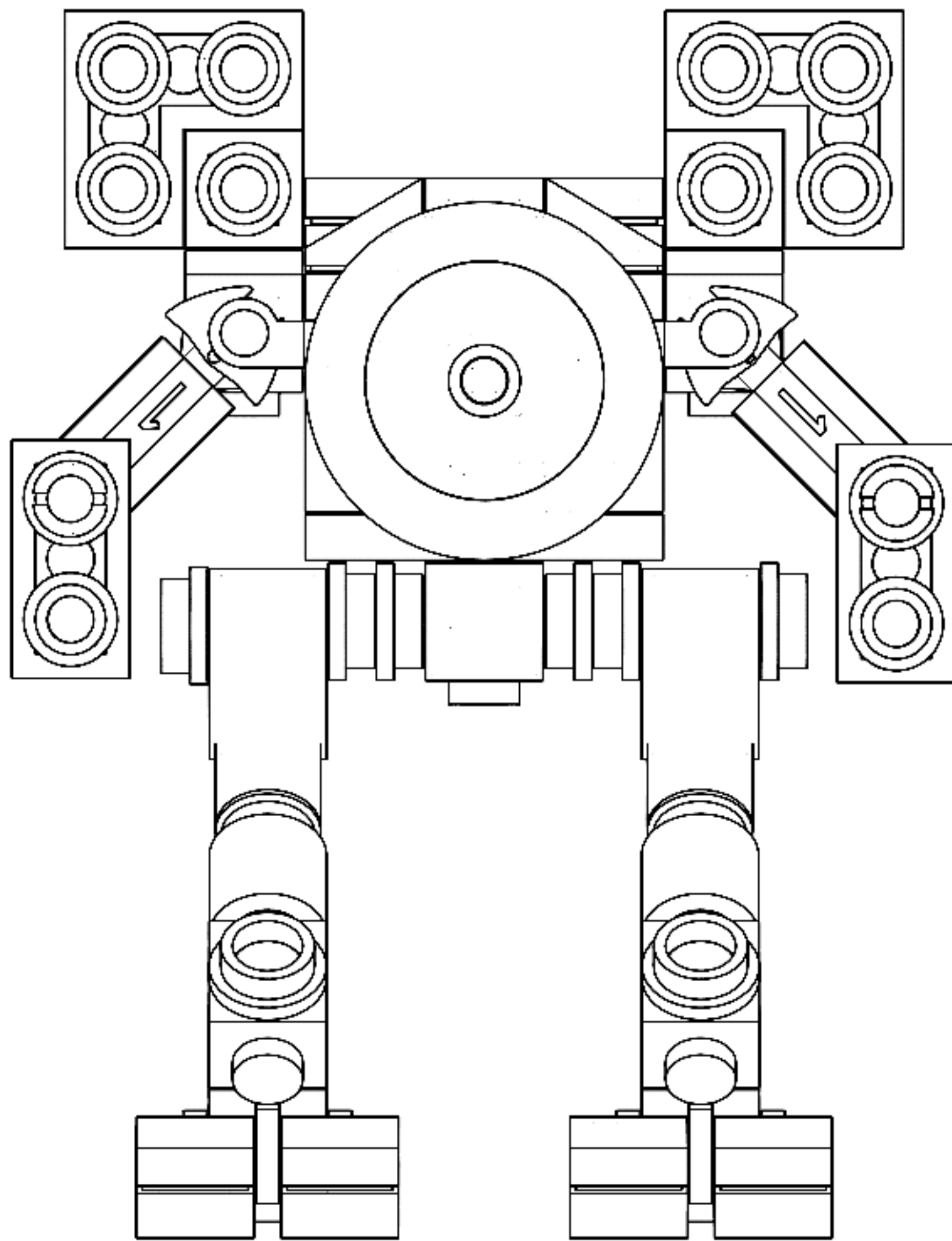
<https://facebook.github.io/flux/>

<http://blog.andrewray.me/flux-for-stupid-people/>

<https://scotch.io/tutorials/getting-to-know-flux-the-react-js-architecture>



NOT A FRAMEWORK



ARCHITECTURE BLUEPRINT



LET'S CHAT

SUMMARY

COMPONENTS,

NOT TEMPLATES

RE-RENDER,

DON'T TOUCH

THE DOM

LET'S TALK ABOUT

VIRTUAL DOM

IS SIMPLE,

FAST, FUN

QUESTIONS?

@NIKOLAYB