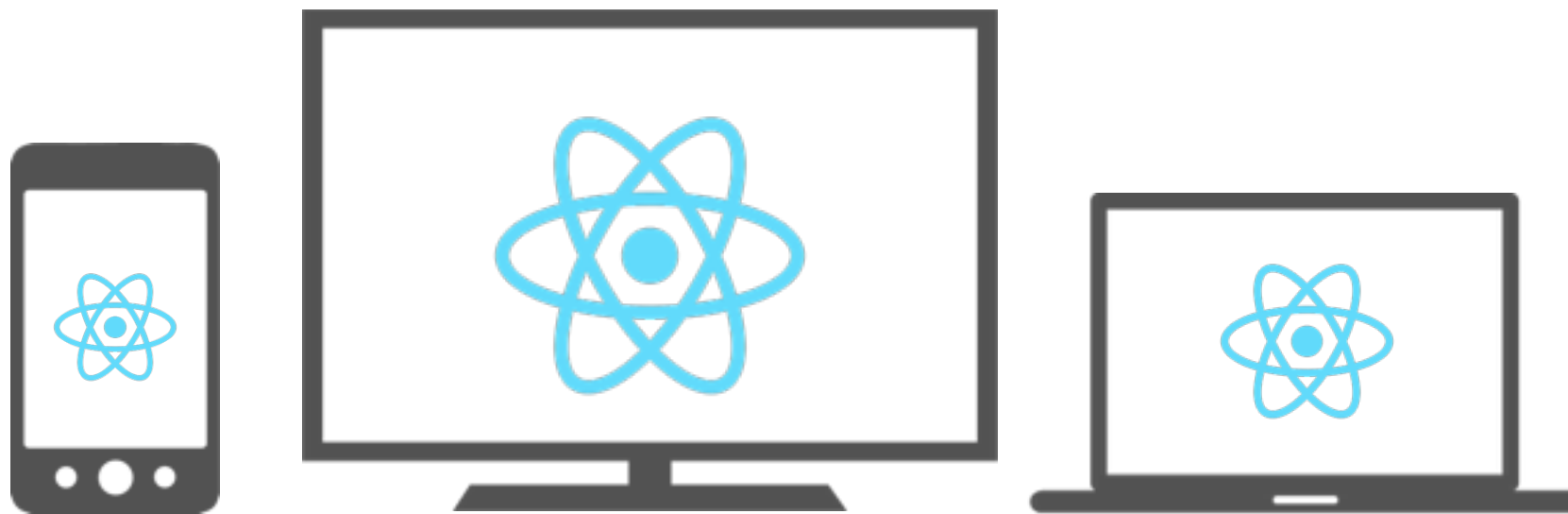


React on every screen

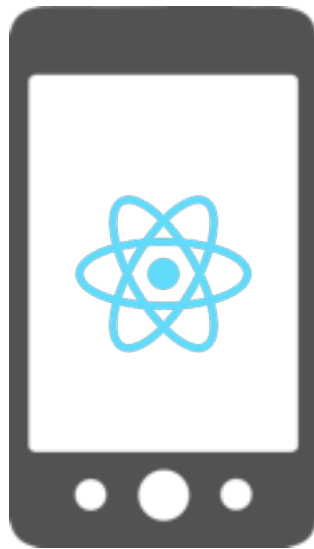


@teebot



Constraints

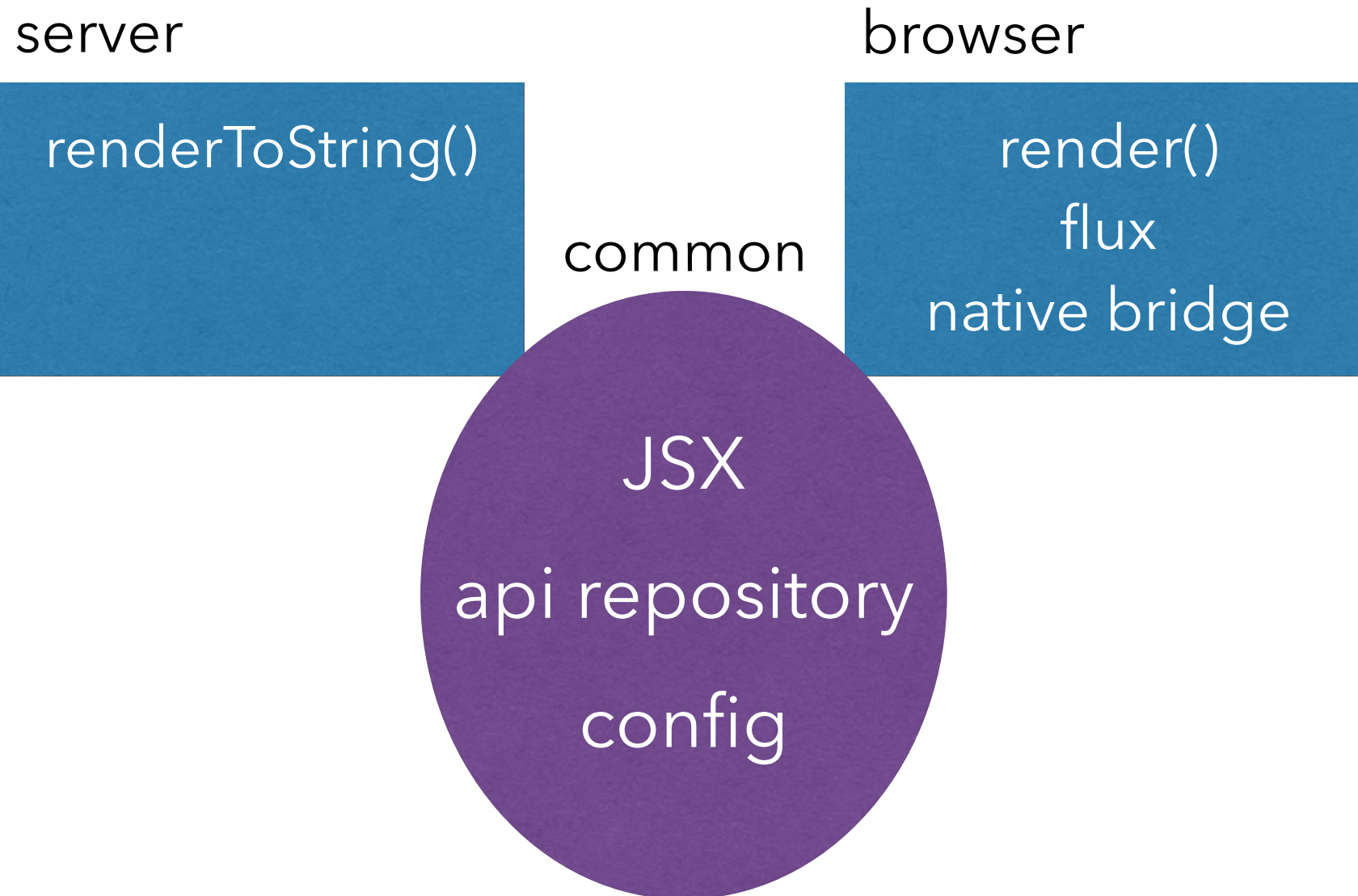
Mobile SDK



No control on its deployment

Expect ""Native experience""

Low bandwidth / spotty connection



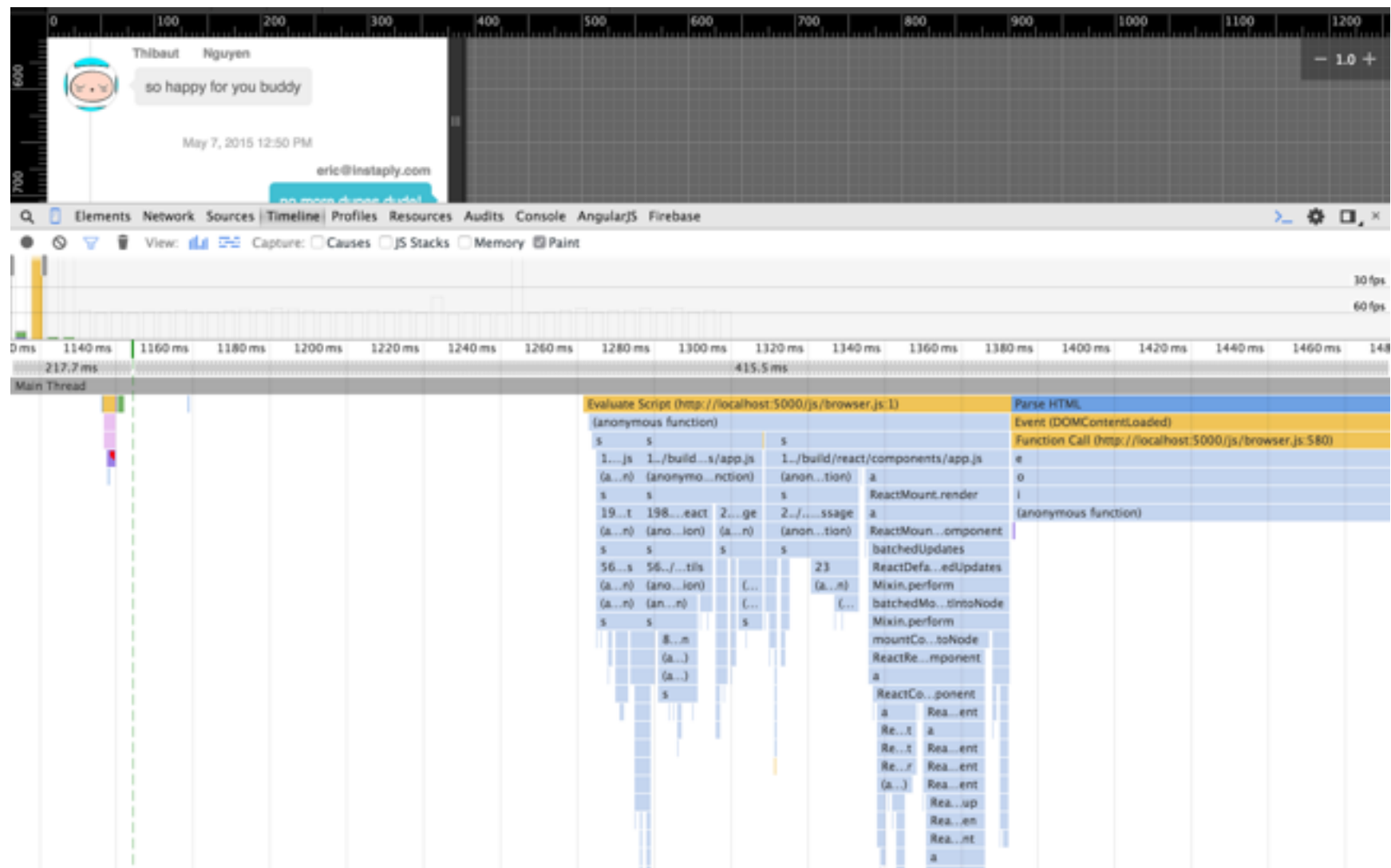
Enhance progressively

```
// helper function for rendering a view with React
function reactRender(res, ComponentClass, props, translations) {
  var component = new ComponentClass(props);
  res.render('index', {
    reactOutput: React.renderToString(component)
  });
}

var ReactApp = React.createFactory(require('./build/react/components/app.js'));
reactRender(res, ReactApp, conversation, translations);
```



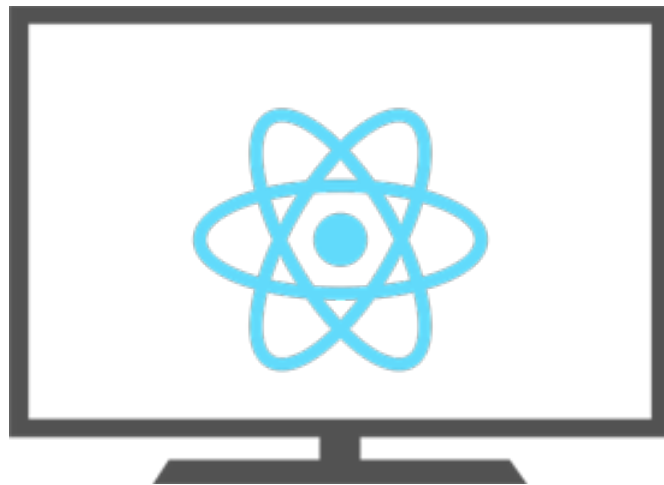
```
render() {  
  var messages = this.state.conversation.items.map(function (item, i) {  
    return (  
      <Message attachment={item.attachment} key={i} />  
    )  
  });  
  
  return (  
    <div id="conversation">  
      {notifWarning}  
      <ReactCSSTransitionGroup transitionName="message-animation">  
        {messages}  
      </ReactCSSTransitionGroup>  
    </div>  
  );  
}
```



Update optimistically

```
onPostMessage: function(message) {  
  // Update conversation optimistically  
  this.conversation.items = this.conversation.items.concat(message);  
  this.updateConversation(this.conversation);  
  
  var msgIndex = this.conversation.items.indexOf(message);  
  var self = this;  
  // once request is done, update its status if needed  
  repo.postMessage(this.token, message, function(error, result) {  
    if (error) {  
      message.failed = true;  
    } else {  
      message.failed = false;  
    }  
    self.conversation.items[msgIndex] = message;  
    self.updateConversation(self.conversation);  
  });  
},  
...|
```

InstaWall



Always on

No user interaction

Oh snaps you're right. Thx for the quick answer!

Hello. Are you open on Thanksgiving by any chance?

I'm glad I can reach you here. Is there any special deal on the barbecues this month by any chance? 😊

I'm kind of hopeful with that weather

```

const Conversation = React.createClass({
  mixins: [Reflux.listenTo(screenViewStore, 'onConversationsChange')],

  onConversationsChange: function() {
    this.scrollToBottom();
  },

  componentDidMount: function() {
    this.scrollToBottom();
  },

  render: function() {
    let typing;
    let messages = this.props.conversation.messages.map((item, i) => {
      return <Message attachment={item.attachment} key={item.href} text={item.text} />
    });

    if(this.props.conversation.isTyping) {
      typing = <div className='typing'>{this.props.conversation.typerName} is typing ...</div>
    }

    return (
      <div className={classString}>
        <TimeoutTransitionGroup enterTimeout={350} leaveTimeout={350} transitionName="message-animation">
          {messages}
        </TimeoutTransitionGroup>
        {typing}
      </div>
    )
  }
});

```

```

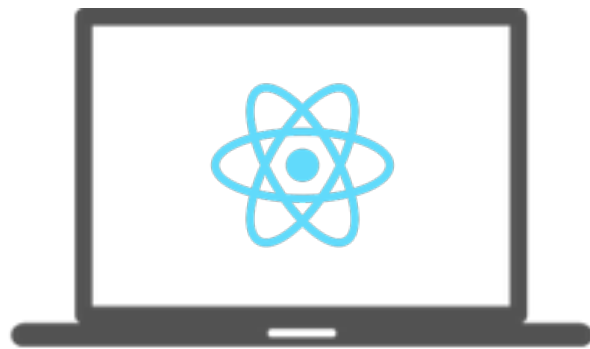
const ScreenViewStore = Reflux.createStore({
  init: function() {
    this.userAuth = JSON.parse(Cookies.get('userAuth'));
    this.connectSocket();
  },

  connectSocket: function() {
    if (this.userAuth.token) {
      let socket = io.connect(config.WS_URL);
      socket.on('connect', () => {
        socket.emit('authentication', {
          token: this.userAuth.token
        });
      });
      socket.on('new_message', (data) => {
        this.onReceivesNewMessage(data.messageHref);
      });

      socket.on('new_typing_event', (data) => {
        this.onMessageTyping({
          conversationHref: data.conversation.href,
          typerName: `${data.typer.firstName} ${data.typer.lastName}`
        });
      });
    }
  },
  ...

```

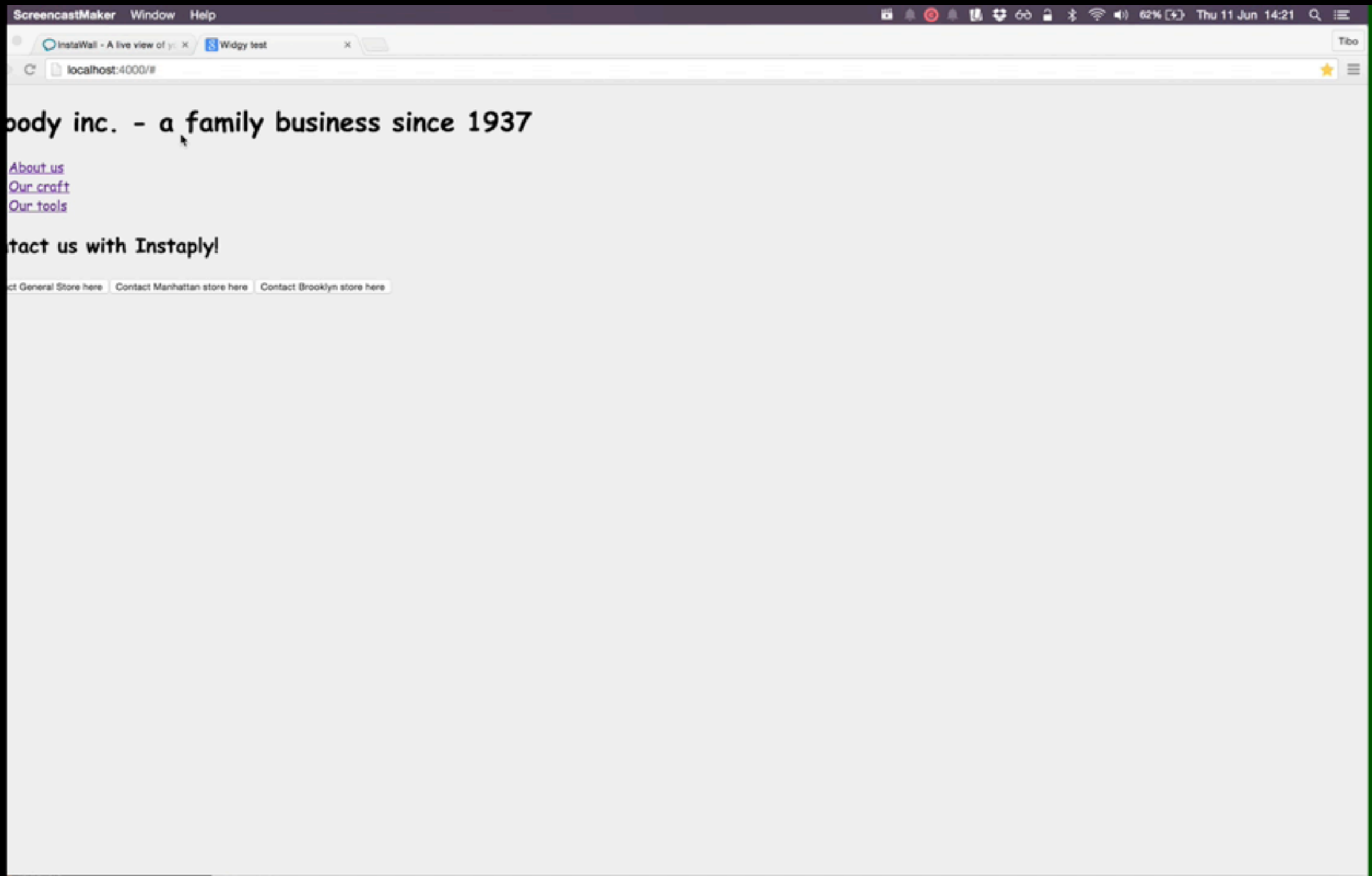
Widget



Cross Origin Requests

JS and CSS evals

Small footprint



server

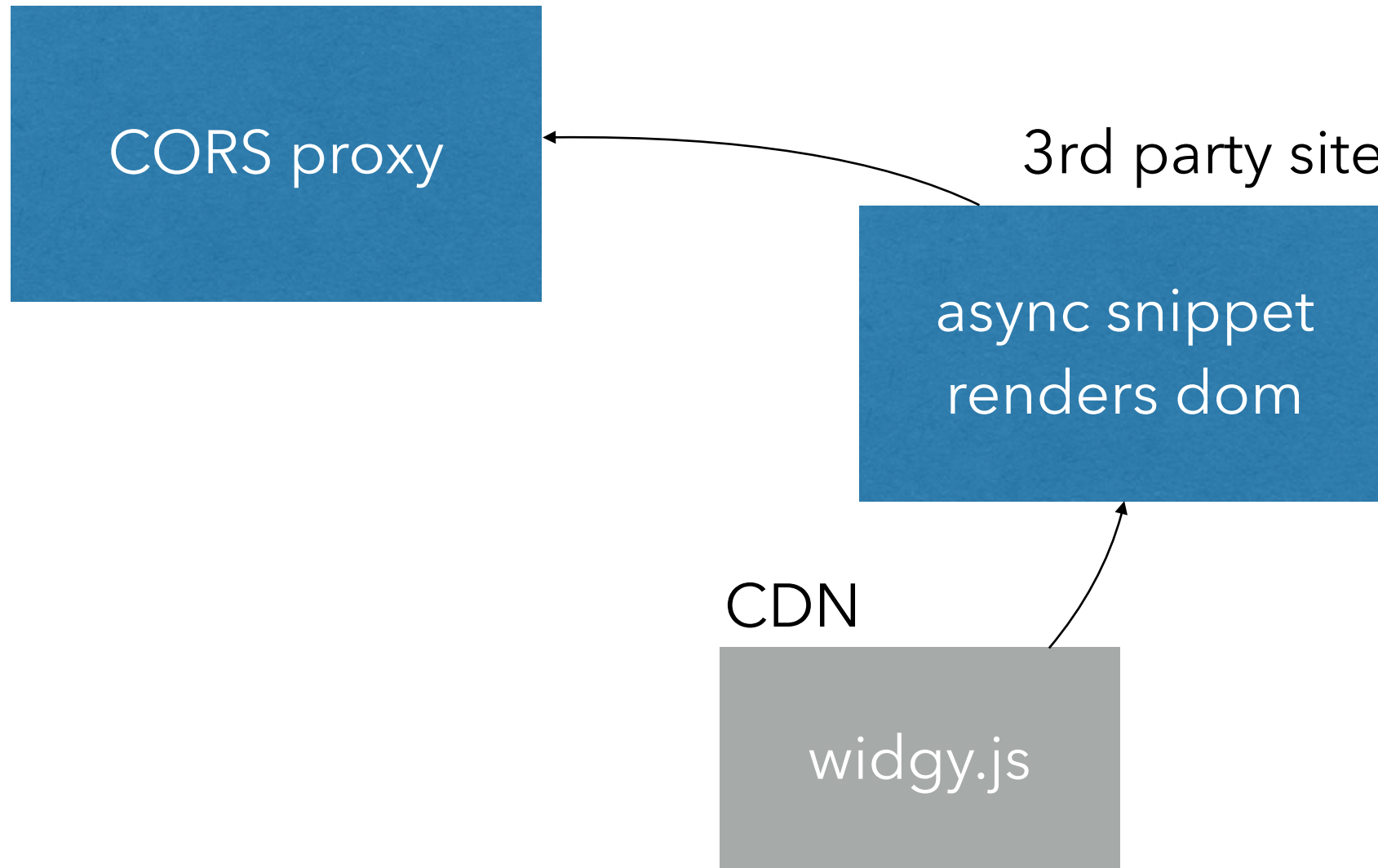
CORS proxy

3rd party site

async snippet
renders dom

CDN

widgy.js



```
<script>
var _iwq = _iwq || {};
_iwq.lang = 'fr';

(function() {
  var iw = document.createElement('script'); iw.type = 'text/javascript'; iw.async = true;
  iw.src = '//cdn.instaply.com/js/widgy.js';
  var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(iw, s);
})();
</script>
```

```
const React = require('react');
const repo = require('./react/repository');
const $ = require('jquery');
require('./css/styles.css');

const ReactApp = React.createFactory(require('./react/components/app.jsx'));

const removeWidget = (e) => {
  $('#iw-react-mount').fadeOut(200, function() {
    $(this).remove();
  });
};

const displayWidget = (businessId) => {
  window._iwqConfig = {
    businessId
  };

  let newDiv = $('<div id="iw-react-mount"></div>').css('display', 'none');
  $('body').append(newDiv);

  let appHtml = React.render(new ReactApp(), newDiv[0], bindCloseEvents);
  $(newDiv).fadeIn(200);
};
```

A word about the
tooling

BABEL

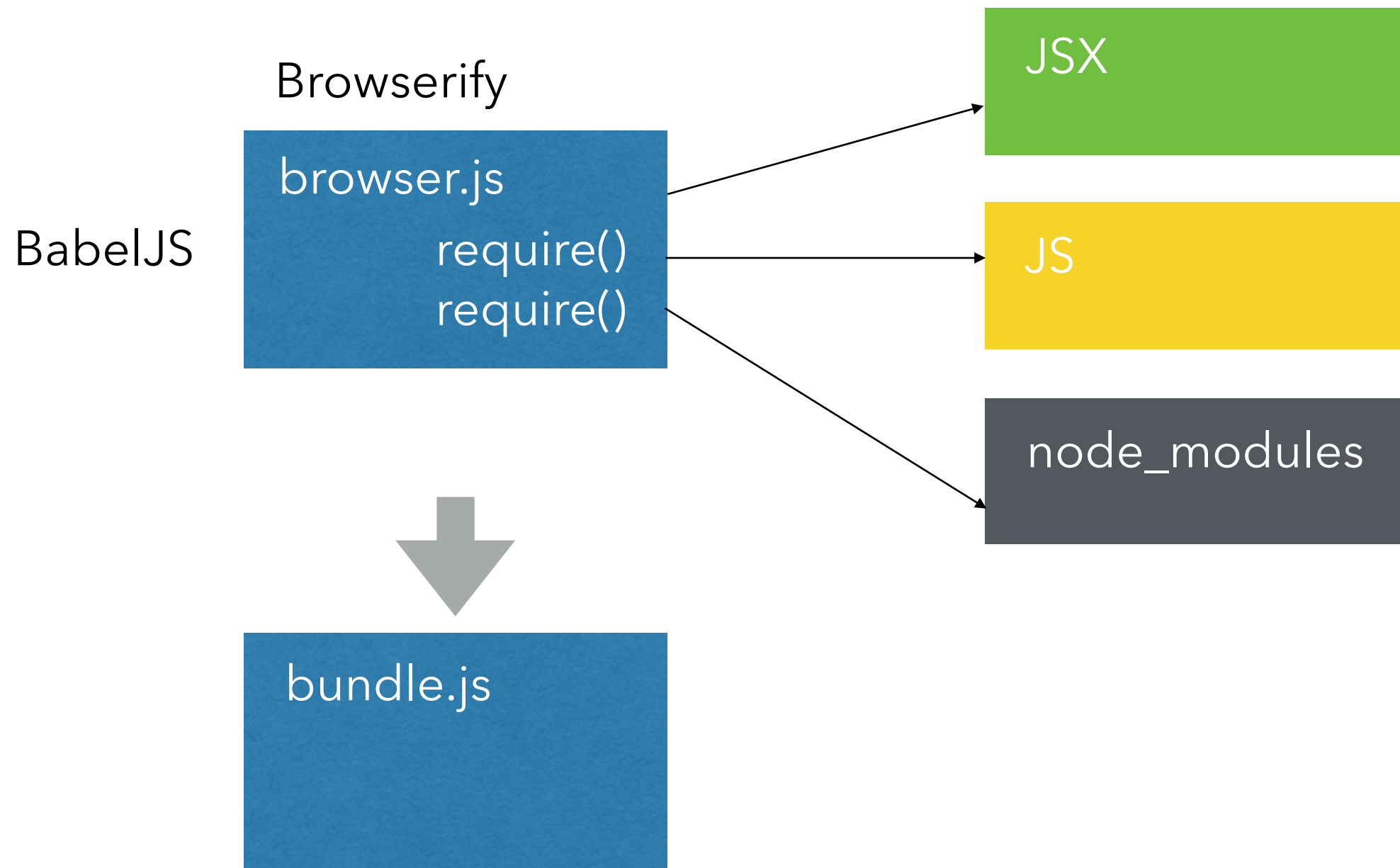
KUNG FURY

```
module.exports = function(grunt) {

  grunt.initConfig({
    browserify: {
      debug: {
        files: {
          'public/js/bundle.js': 'browser.js'
        },
        options: {
          transform: [['babelify', { 'stage': 0 }]]
        }
      },
      prod: {
        files: {
          'public/js/bundle.js': 'browser.js'
        },
        options: {
          transform: [['babelify', { 'stage': 0 }]],
          plugin: [['minifyify', { map: false }]]
        }
      }
    }
  });

  grunt.loadNpmTasks('grunt-browserify');

  grunt.registerTask('default', ['browserify:prod']);
  grunt.registerTask('debug', ['browserify:debug']);
};
```



Thanks