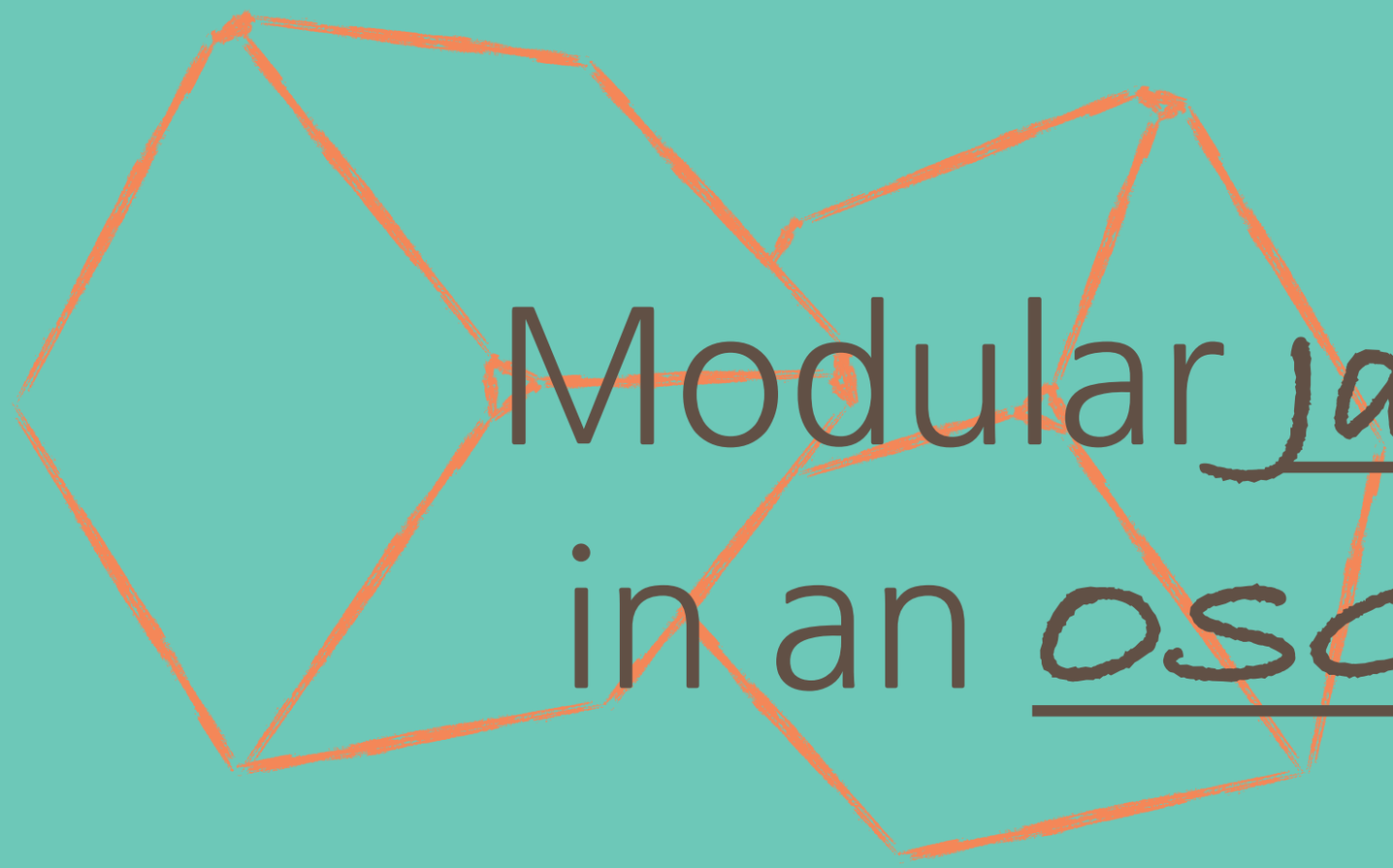
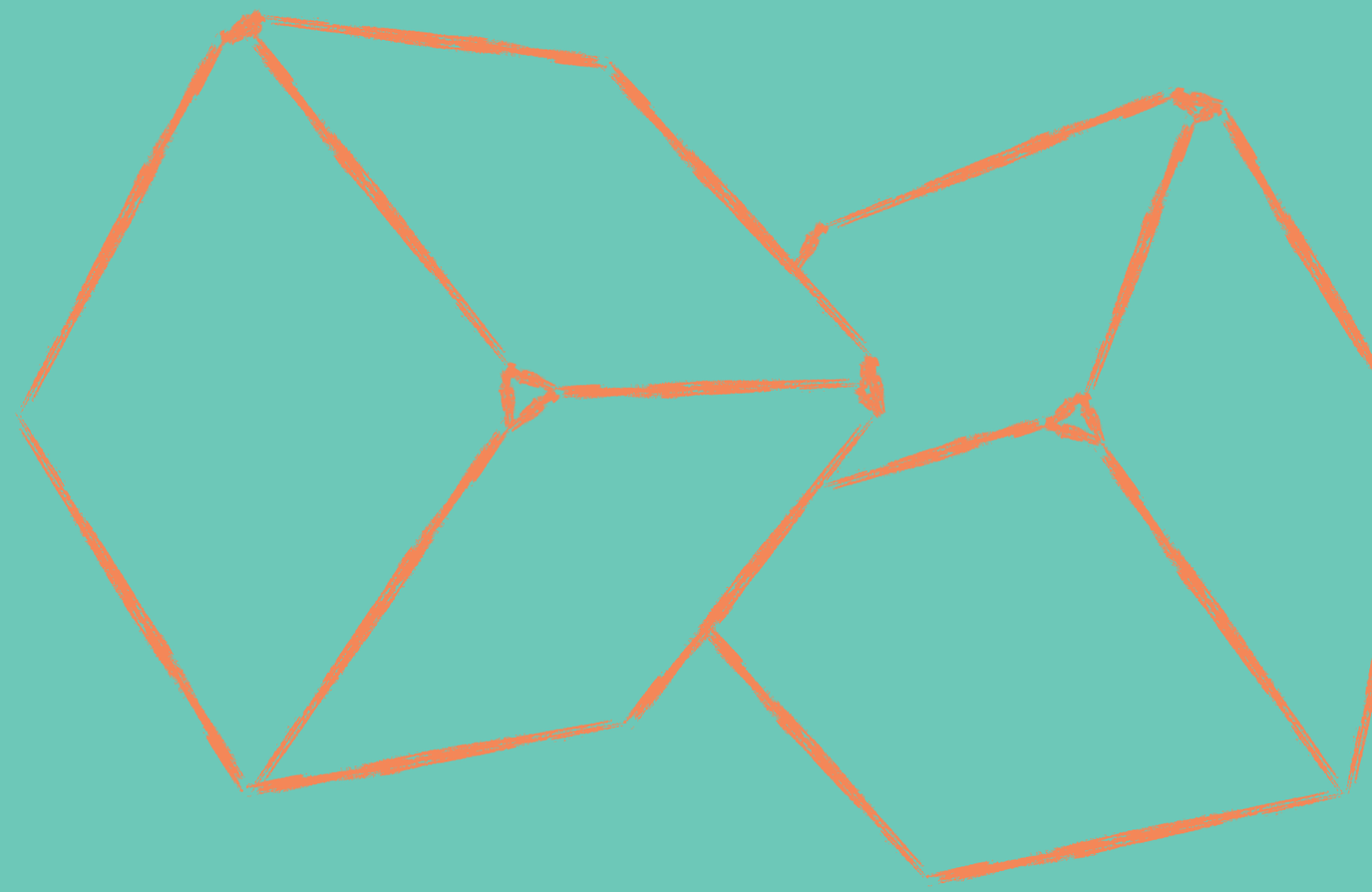


Modular JavaScript in an OSGi world

@sander_mak

luminis
Conversing worlds

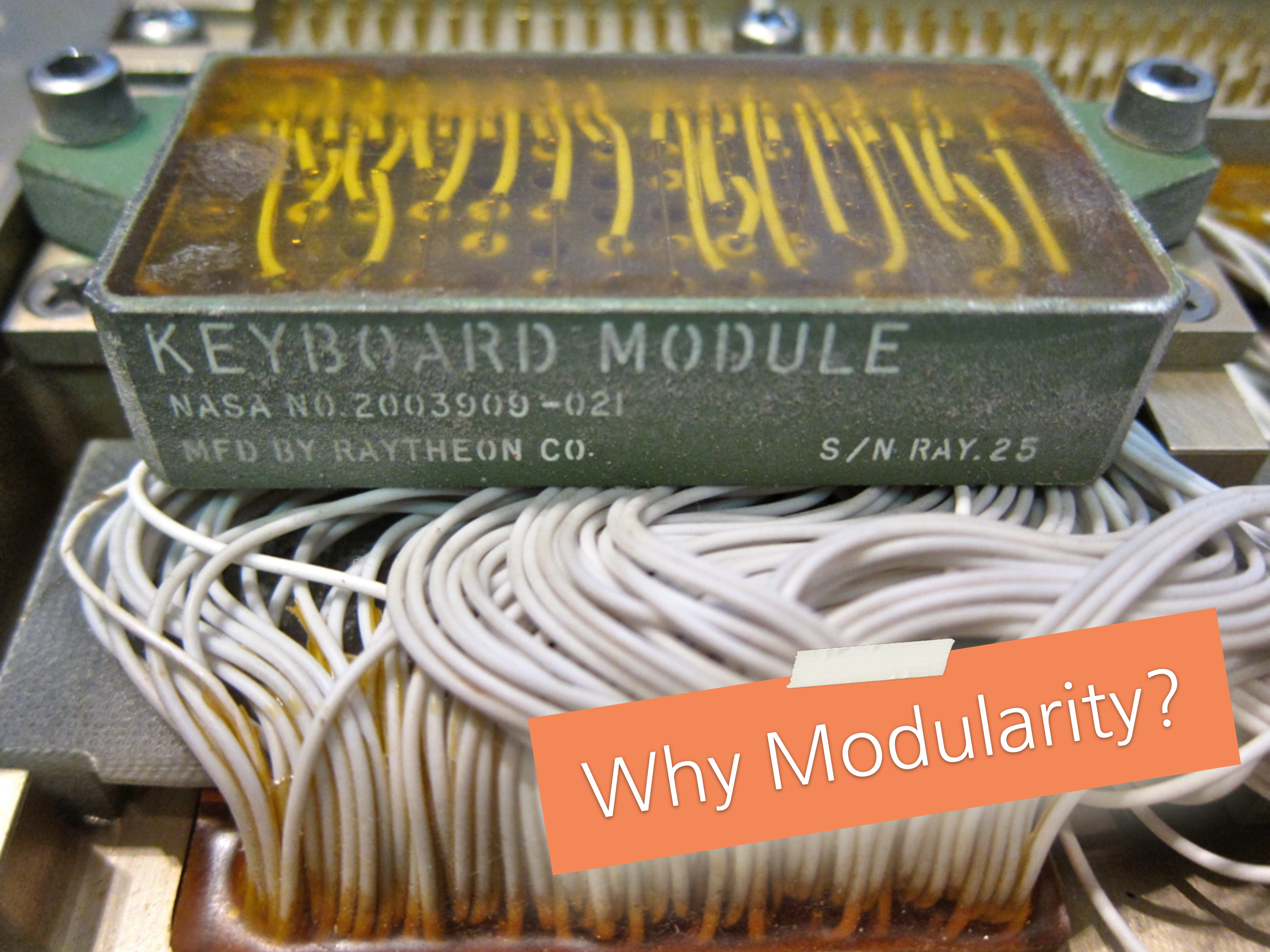


JavaScript?

JS Module Systems

JavaScript & OSGi

Future of modular webapps



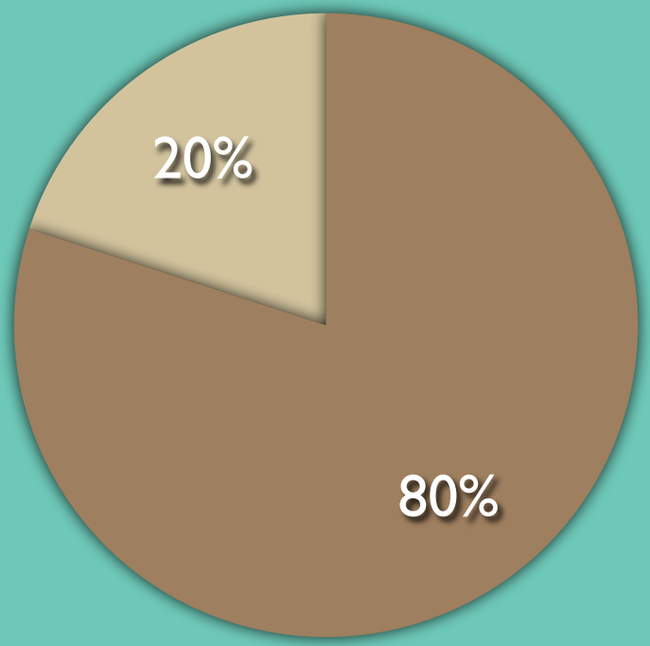
Why Modularity?

Modularity is

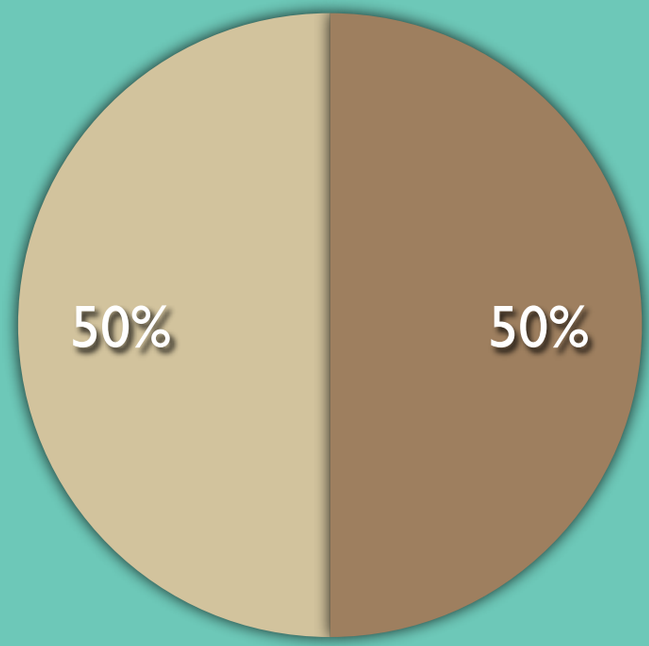
to maintainable code

Why modularity in JavaScript?

then



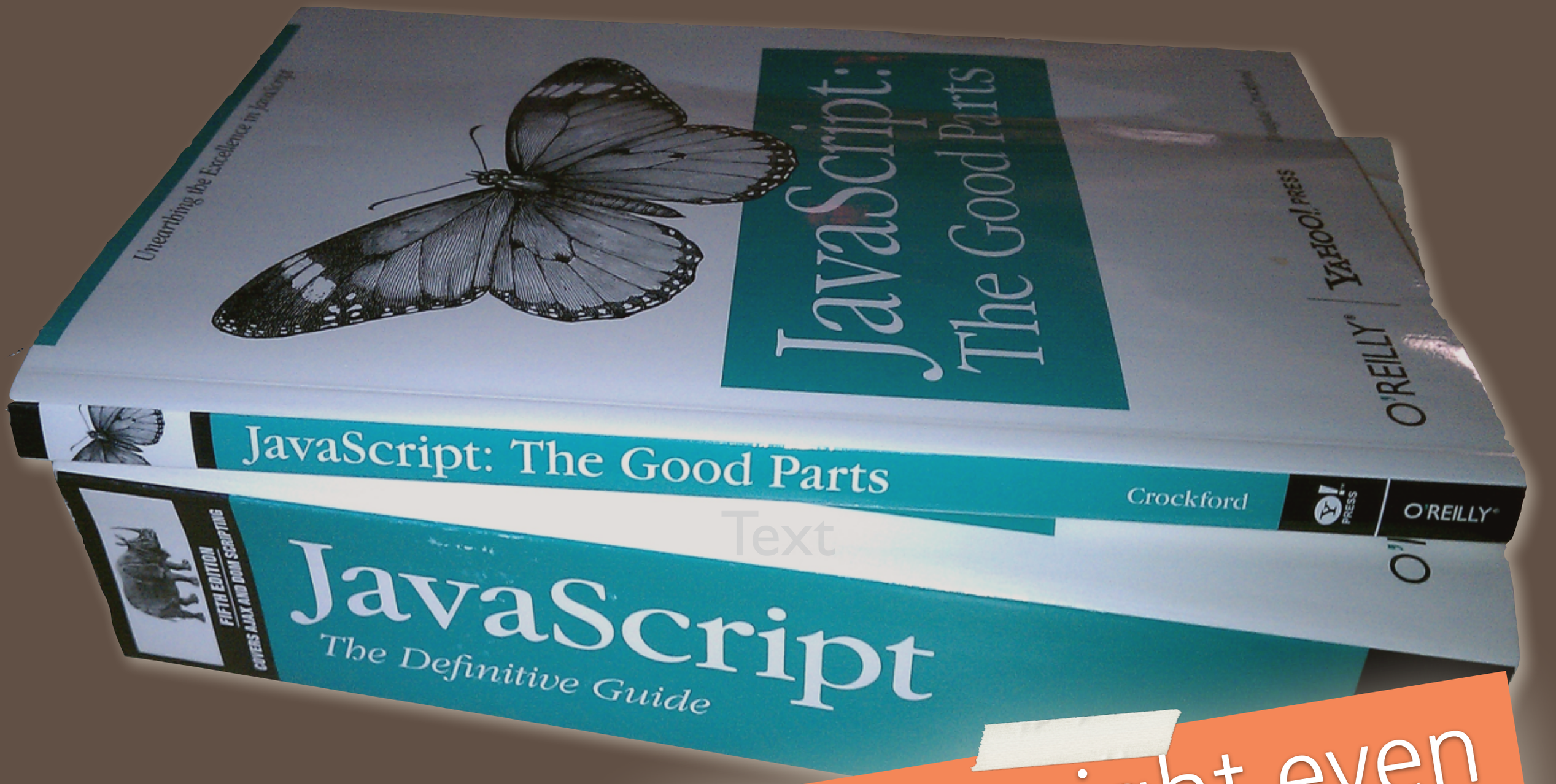
now



Java JavaScript

So...

We run into the same maintainability problems as on the server side...



And it might even be **worse**....

JavaScript Modularity 101

```
//Global variable
var myvar = "hello";

//Global function
function myfunction() {
}

//Global function defining a global value
function test() {
    somevalue = "still global";
}
```

Anonymous functions

Anonymous functions prevent putting something in global scope


```
(function() {
    console.log("This is private!");
}());
```

IIFE: 'Immediately invoked function expression'

Poor man's 'Modules'


```
//Anomynous function, not global
(function() {
    var myLibrary = {};

    //Now private to the anomynous function
    function myhelper() {
        // e.g. input validation
        console.log('helping');
    }

    myLibrary.add = function(x, y) {
        myhelper();

        return x + y;
    }

    window.myLibrary = myLibrary;
}());
```

myhelper is 'private'

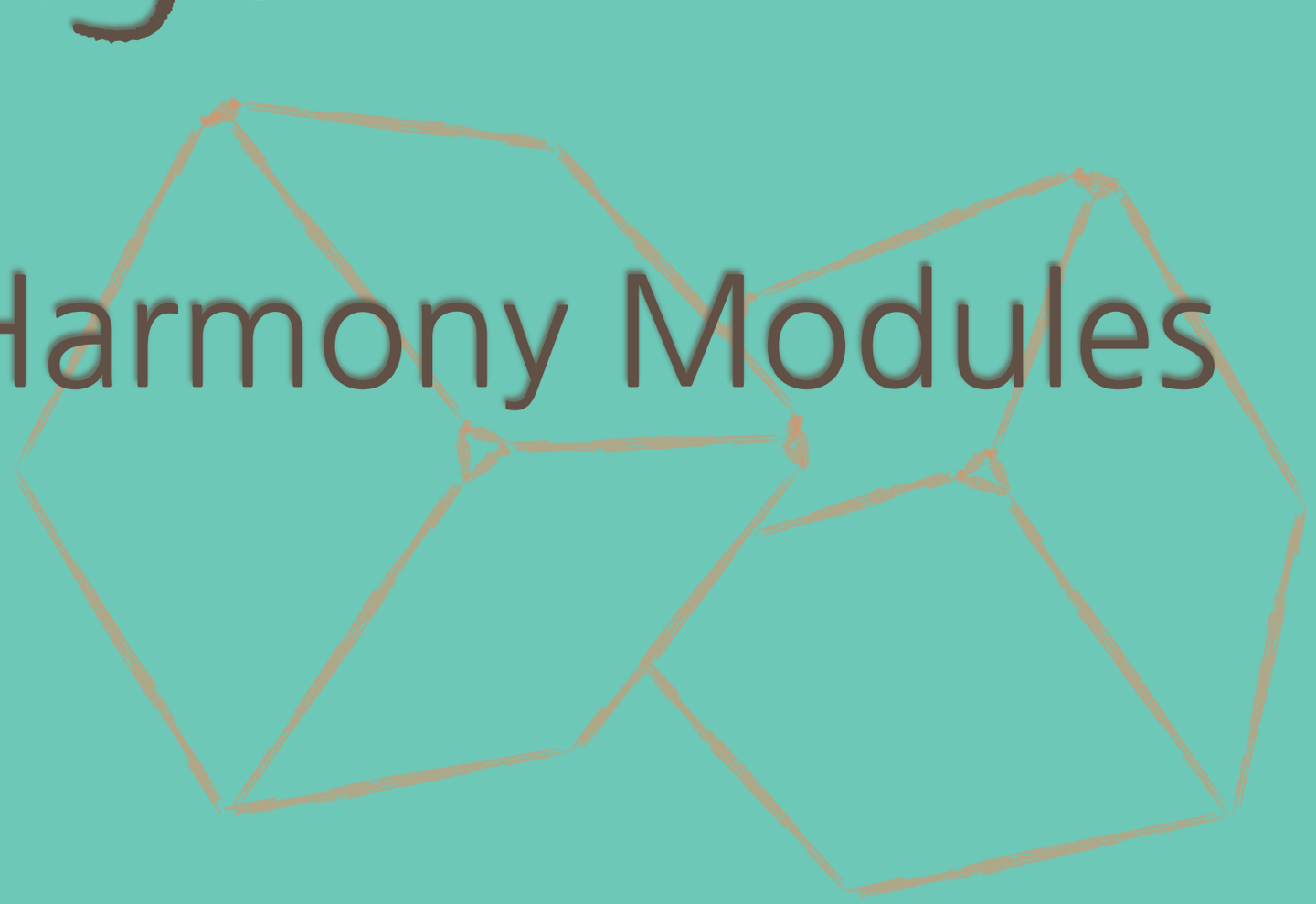
Comparing JavaScript Module Systems

Module options

Asynchronous Module Definition

CommonJS

ES6 Harmony Modules



Module comparison

	Spec?	Impls?	Environment
AMD	Spec-by-GitHub	RequireJS Curl.js …	Browser first
C.JS	Spec-by-Wiki	Browserify Node.js …	Server first
ES6	Ecma standard (in progress)	-	Both

Module comparison

AMD

```
// js/webshop.js
define(

  ['userservice', 'shoppingcart'],

  function (userservice, shoppingcart) {

    var webshopModule = {
      placeOrder: function() { ... }
    }

    return webshopModule;
  }
);
```

Module comparison

AMD

```
// js/webshop.js
define(
  ['userservice', 'shoppingcart'],

  function (userservice, shoppingcart) {

    var webshopModule = {
      placeOrder: function() { ... }
    }

    return webshopModule;
  }
);
```

Module comparison

AMD

```
// js/webshop.js
define(
  ['userservice', 'shoppingcart'],
  function (userservice, shoppingcart) {

    var webshopModule = {
      placeOrder: function() { ... }
    }

    return webshopModule;
  }
);
```

Module comparison

AMD

```
// js/webshop.js
define(

  ['userservice', 'shoppingcart'],

  function (userservice, shoppingcart) {

    var webshopModule = {
      placeOrder: function() { ... }
    }

    return webshopModule;
  }
);
```

Module comparison

AMD

```
// js/webshop.js
define(

  ['userservice', 'shoppingcart'],

  function (userservice, shoppingcart) {

    var webshopModule = {
      placeOrder: function() { ... }
    }

    return webshopModule;
  }
);
```

CommonJS

```
var userservice, shoppingcart;

userservice = require('userservice');
shoppingcart = require('shoppingcart');

function placeOrder() { ... }

exports.placeOrder = placeOrder;
```

Module comparison

AMD

```
// js/webshop.js
define(

  ['userservice', 'shoppingcart'],

  function (userservice, shoppingcart) {

    var webshopModule = {
      placeOrder: function() { ... }
    }

    return webshopModule;
  }
);
```

CommonJS

```
var userservice, shoppingcart;

userservice = require('userservice');
shoppingcart = require('shoppingcart');

function placeOrder() { ... }

exports.placeOrder = placeOrder;
```

Module comparison

AMD

```
// js/webshop.js
define(

  ['userservice', 'shoppingcart'],

  function (userservice, shoppingcart) {

    var webshopModule = {
      placeOrder: function() { ... }
    }

    return webshopModule;
  }
);
```

CommonJS

```
var userservice, shoppingcart;

userservice = require('userservice');
shoppingcart = require('shoppingcart');

function placeOrder() { ... }

exports.placeOrder = placeOrder;
```

Module comparison

AMD

```
// js/webshop.js
define(

  ['userservice', 'shoppingcart'],

  function (userservice, shoppingcart) {

    var webshopModule = {
      placeOrder: function() { ... }
    }

    return webshopModule;
  }
);
```

CommonJS

```
var userservice, shoppingcart;

userservice = require('userservice');
shoppingcart = require('shoppingcart');

function placeOrder() { ... }

exports.placeOrder = placeOrder;
```

ES6 Harmony

```
module 'webshop' {
  import { getUser } from 'userservice'
  module shoppingcart from 'http://url/shoppingcart.js'

  var placeOrder = function() { .. }
  export placeOrder
}
```

Module comparison

AMD

```
// js/webshop.js
define(

  ['userservice', 'shoppingcart'],

  function (userservice, shoppingcart) {

    var webshopModule = {
      placeOrder: function() { ... }
    }

    return webshopModule;
  }
);
```

CommonJS

```
var userservice, shoppingcart;

userservice = require('userservice');
shoppingcart = require('shoppingcart');

function placeOrder() { ... }

exports.placeOrder = placeOrder;
```

ES6 Harmony

```
module 'webshop'
  import { getUser } from 'userservice'
  module shoppingcart from 'http://url/shoppingcart.js'

  var placeOrder = function() { .. }
  export placeOrder
}
```

Module comparison

AMD

```
// js/webshop.js
define(

  ['userservice', 'shoppingcart'],

  function (userservice, shoppingcart) {

    var webshopModule = {
      placeOrder: function() { ... }
    }

    return webshopModule;
  }
);
```

CommonJS

```
var userservice, shoppingcart;

userservice = require('userservice');
shoppingcart = require('shoppingcart');

function placeOrder() { ... }

exports.placeOrder = placeOrder;
```

ES6 Harmony

```
module 'webshop' {
  import { getUser } from 'userservice'
  module shoppingcart from 'http://url/shoppingcart.js'

  var placeOrder = function() { .. }
  export placeOrder
}
```

Module comparison

AMD

```
// js/webshop.js
define(

  ['userservice', 'shoppingcart'],

  function (userservice, shoppingcart) {

    var webshopModule = {
      placeOrder: function() { ... }
    }

    return webshopModule;
  }
);
```

CommonJS

```
var userservice, shoppingcart;

userservice = require('userservice');
shoppingcart = require('shoppingcart');

function placeOrder() { ... }

exports.placeOrder = placeOrder;
```

ES6 Harmony

```
module 'webshop' {
  import { getUser } from 'userservice'
  module shoppingcart from 'http://url/shoppingcart.js'

  var placeOrder = function() { .. }
  export placeOrder
}
```

Can't we have it all?

Universal Module Definition

```
(function (root, factory) {
    if (typeof define === 'function' && define.amd) {
        // AMD. Register as an anonymous module.
        define(factory);
    } else if (typeof exports === 'object') {
        // Node. Does not work with strict CommonJS, but
        // only CommonJS-like enviroments that support module.exports,
        // like Node.
        module.exports = factory();
    } else {
        // Browser globals (root is window)
        root.returnExports = factory();
  }
}(this, function () {

    // Just return a value to define the module export.
    // This example returns an object, but the module
    // can return a function as the exported value.
    return {};
}));
```

Can't we have it all?

Universal Module Definition

```
(function (root, factory) {
    if (typeof define === 'function' && define.amd) {
        // AMD. Register as an anonymous module.
        define(factory);
    } else if (typeof exports === 'object') {
        // Node. Does not work with strict CommonJS, but
        // only CommonJS-like enviroments that support module.exports,
        // like Node.
        module.exports = factory();
    } else {
        // Browser globals (root is window)
        root.returnExports = factory();
  }
}(this, function () {

    // Just return a value to define the module export.
    // This example returns an object, but the module
    // can return a function as the exported value.
    return {};
}));
```

NO!

Pick one.
And stick with it.

(unless you are a library author)

Why RequireJS?

- Robust AMD implementation
  - Browser-based, no build step
  - Lazy-loading
- Optimizer: r.js
- Backwards compatible with globals
- jQuery supports AMD

RequireJS: *end-to-end*

index.html

```
<!DOCTYPE html>
<html>
<head>
    <script type="text/javascript" src="require.js" data-main="js/main"></script>
</head>
<body>
  <!-- the app -->
</body>
</html>
```

RequireJS: *end-to-end*

index.html

```html
<!DOCTYPE html>
<html>
<head>
    <script type="text/javascript" src="require.js" data-main="js/main"></script>
</head>
<body>
  <!-- the app -->
</body>
</html>
```

main.js

```javascript
require.config({
  paths: {
    'mymodule': 'mymodule-1.2'
  }
});

require(['mymodule'], function(mymodule) {
  document.write('<p>' + mymodule.message() + '<p>');
});
```

RequireJS: *end-to-end*

index.html

```html
<!DOCTYPE html>
<html>
<head>
    <script type="text/javascript" src="require.js" data-main="js/main"></script>
</head>
<body>
  <!-- the app -->
</body>
</html>
```

main.js

```javascript
require.config({
  paths: {
    'mymodule': 'mymodule-1.2'
  }
});

require(['mymodule'], function(mymodule) {
  document.write('<p>' + mymodule.message() + '<p>');
});
```

mymodule-1.2.js

```javascript
define([], function() {
  return {
    message: function() {
      return "Hello RequireJS!";
    }
  }
})
```

That. Is. All.

From legacy to modules

RequireJS Shims

```
require.config({
  paths: {
    'angular': '//ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js',
    'jquery': '//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js',
  },

  //  Tune dependencies of non-AMD scripts
  shim: {
    'angular': {
      exports: 'angular',
      deps: ['jquery']
    }
  }
});
```


```
require(['angular'], function(ng) {
  ng.module('MyAngularApp').service(...)
});
```

From legacy to modules

RequireJS Shims

```
require.config({
  paths: {
    'angular': '//ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js',
    'jquery': '//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js',
  },

  //  Tune dependencies of non-AMD scripts
  shim: {
    'angular': {
      exports: 'angular',
      deps: ['jquery']
    }
  }
});
```


```
require(['angular'], function(ng) {
  ng.module('MyAngularApp').service(...)
});
```

From legacy to modules

RequireJS Shims

```
require.config({
  paths: {
    'angular': '//ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js',
    'jquery': '//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js',
  },

  // Tune dependencies of non-AMD scripts
  shim: {
    'angular': {
      exports: 'angular',
      deps: ['jquery']
    }
  }
});
```


```
require(['angular'], function(ng) {
  ng.module('MyAngularApp').service(...)
});
```

Dependency injection

Module loading != Dependency Injection

Think **import** vs. **new**

package deps vs. OSGi services

Dependency injection

Module loading != Dependency Injection

Think **import** vs. **new**

package deps vs. **OSGi services**

Option: 

Dependency injection

ANGULARJS by Google

```
define('angular', function(ng) {

  var ngModule = ng.module('webshop.services', []);

  ngModule.factory('basket', function() {
    // Private initialization

    return {
      addProduct: function(product) {
        // add product to basket
      },

      // more methods...
    }
  }
});
```

All Angular modules must be known at ‘startup’

Dependency injection

ANGULARJS by Google

AMD Module

```
define('angular', function(ng) {
  var ngModule = ng.module('webshop.services', []);

  ngModule.factory('basket', function() {
    // Private initialization

    return {
      addProduct: function(product) {
        // add product to basket
      },

      // more methods...
    }
  }
});
```

All Angular modules must be known at ‘startup’

Dependency injection

ANGULARJS by Google

Angular Module

```
define('angular', function(ng) {

  var ngModule = ng.module('webshop.services', []);

  ngModule.factory('basket', function() {
    // Private initialization

    return {
      addProduct: function(product) {
        // add product to basket
      },

      // more methods...
    }
  }
});
```

All Angular modules must be known at ‘startup’

Dependency injection

ANGULARJS by Google

```
var ctrlModule = angular.module('webshop.controllers', ['webshop.services']);

ctrlModule.controller('ProductCtrl', ['$scope', '$routeParams', 'basket',
    function($scope, $routeParams, basket) {

      $scope.category = routeParams.category;

      $scope.inBasket = function(product) {
        basket.addProduct(product);
      }
    }]
);
```

Dependency injection

ANGULARJS by Google

Angular module dependency

```
var ctrlModule = angular.module('webshop.controllers', ['webshop.services']);

ctrlModule.controller('ProductCtrl', ['$scope', '$routeParams', 'basket',
    function($scope, $routeParams, basket) {

      $scope.category = routeParams.category;

      $scope.inBasket = function(product) {
        basket.addProduct(product);
      }
    }]
);
```

Dependency injection

ANGULARJS by Google

Injecting the service

```
var ctrlModule = angular.module('webshop.controllers', ['webshop.services']);

ctrlModule.controller('ProductCtrl', ['$scope', '$routeParams', 'basket',
    function($scope, $routeParams, basket) {

      $scope.category = routeParams.category;

      $scope.inBasket = function(product) {
        basket.addProduct(product);
      }
    }]
);
```

JavaScript & OSGi

Modularity across the *wire*

The high-level idea (I)

JS Module

JS Module

JS Module

REST

WebSockets

amdatu

OSGi bundles

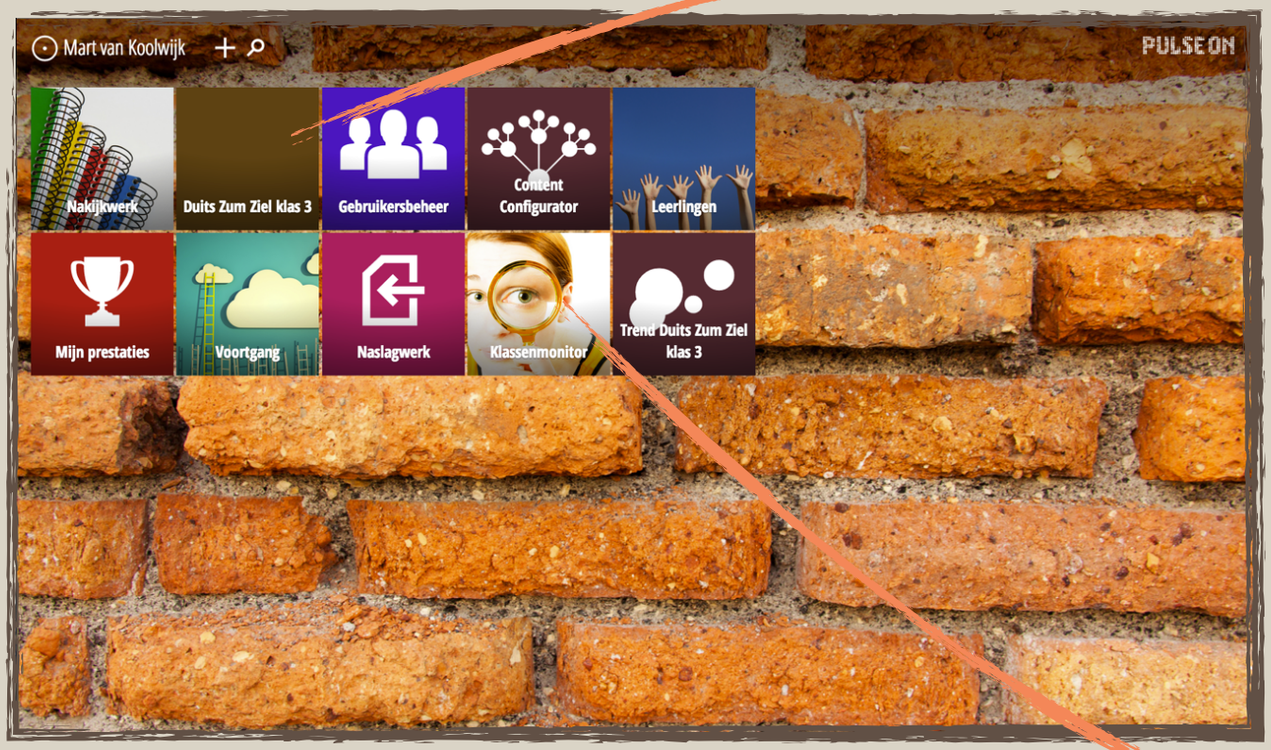
Apache Felix

The abstract idea (I)

We evolved through 3 implementations

- OpenSocial gadget container
- RequireJS + Angular module per bundle
- RequireJS + One ‘modular’ frontend bundle

OpenSocial gadgets



Dashboard

Gadget 1 (iFrame)

Gadget 2 (iFrame)

OpenSocial gadgets

Gadget:

- Amdatu OpenSocial (Apache Shindig)
- OSGi bundle
  - Serve static resources
  - Register Gadget as service (whiteboard)
- Standardized XML descriptor
  - Initial HTML views
  - Preferences

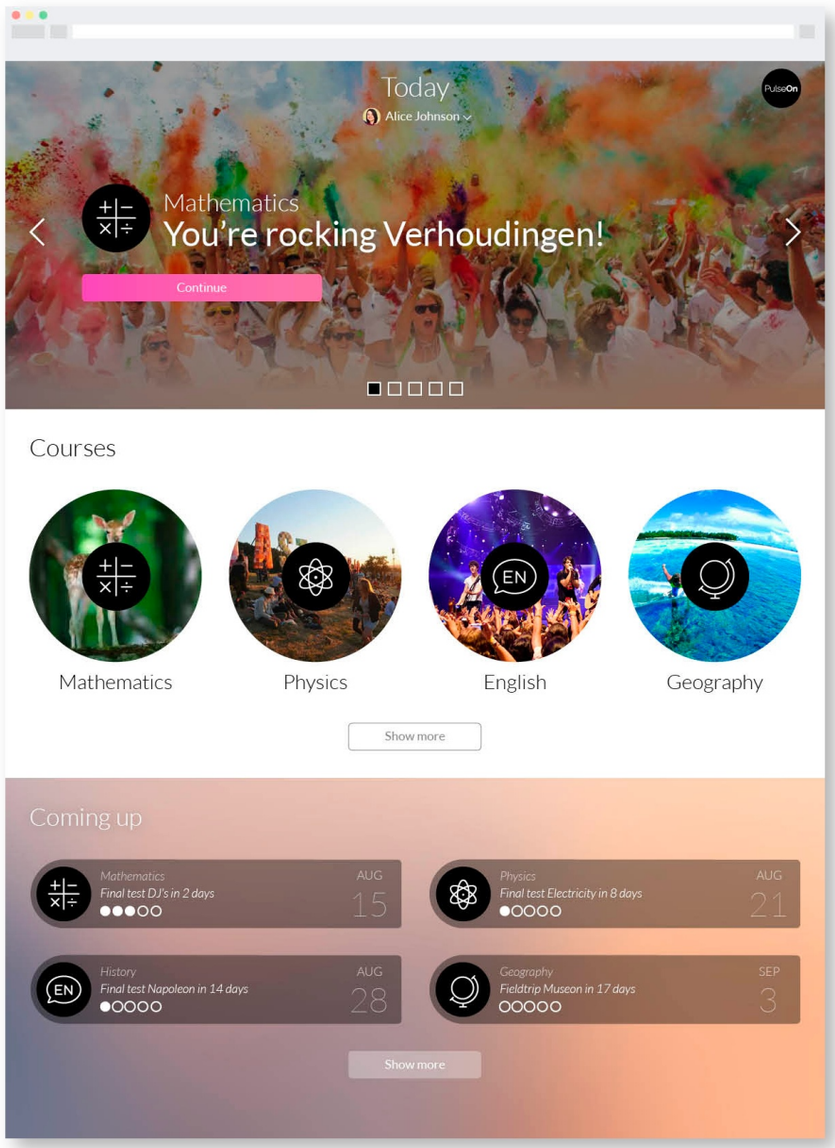
OpenSocial gadgets

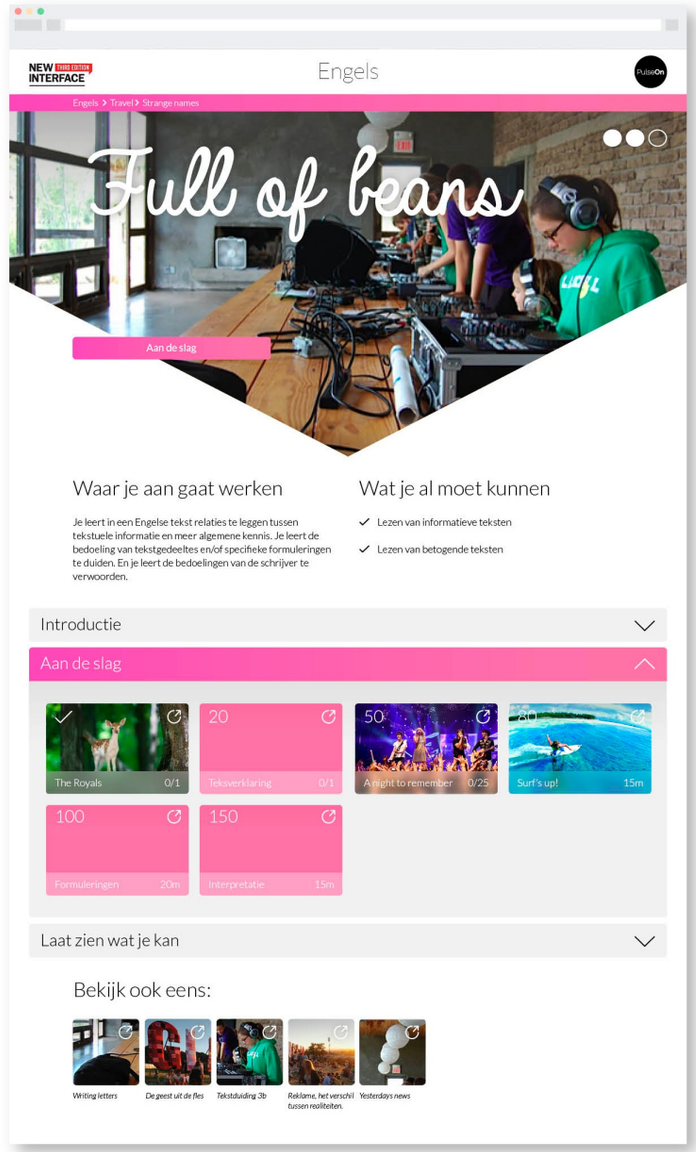
- Every gadget isolated (CSS, JS freedom)
- Frontend dynamics match OSGi

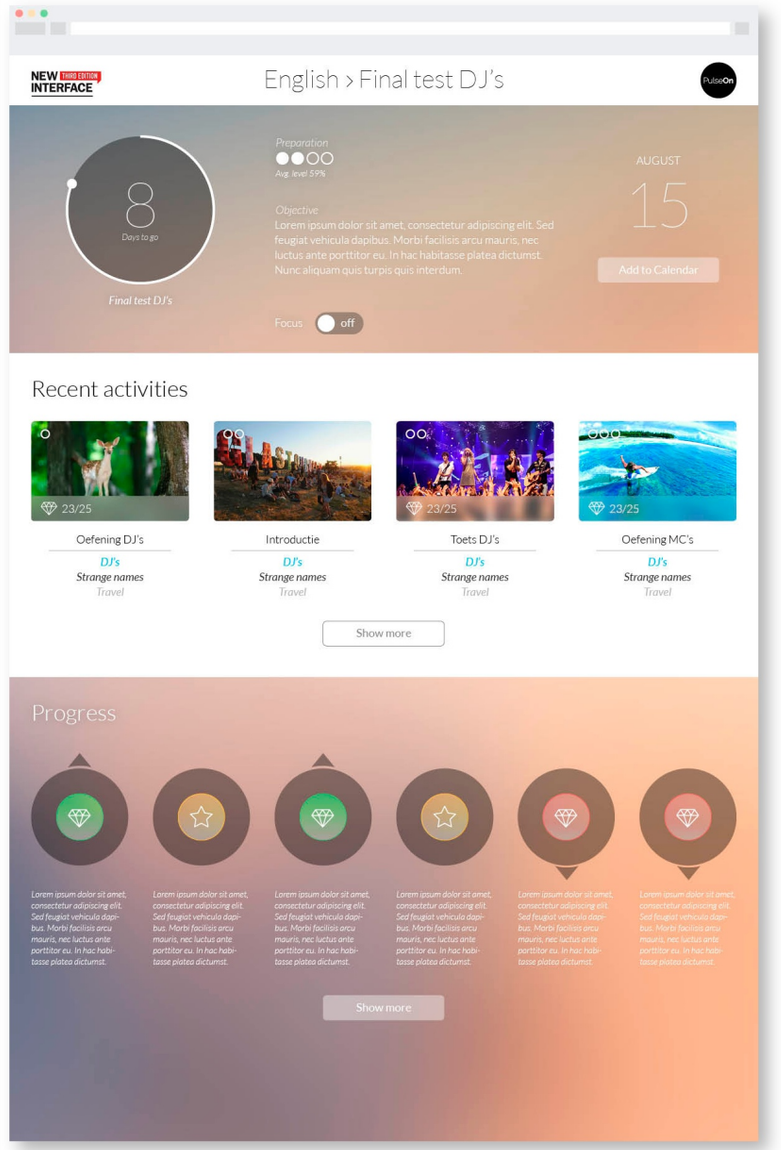
- Every gadget isolated (CSS/JS sharing)
- iFrames suck: sub-optimal user experience
- OpenSocial is dead

Angular **module per bundle**

Towards a true Single Page Web-application







No more iFrames...

Angular **module per bundle**

Bundle 1

/META-INF/MANIFEST.MF
/static/js/module.js
/static/js/..*.js
/static/templates
/static/css

Main frontend bundle

/META-INF/MANIFEST.MF
/static/js/angular.js
/static/js/require.js

Bundle 2

/META-INF/MANIFEST.MF
/static/js/module.js
/static/js/..*.js
/static/templates
/static/css

Angular **module per bundle**

Bundle 1

```
X-Web-Resource-Version: 1.1
X-Web-Resource: /profile;static
Include-Resource: static=static
X-NgApp: profile;module;profile.app
X-NgLink: myprofile
```

/META-INF/
/static/js/m
/static/js/..*
/static/templates
/static/css

Main frontend bundle

/META-INF/MANIFEST.MF
/static/js/angular.js
/static/js/require.js

Bundle 2

/META-INF/MANIFEST.MF
/static/js/module.js
/static/js/..*.js
/static/templates
/static/css

Angular **module per bundle**

Bundle 1

/META-INF/MANIFEST.MF
/static/js/module.js
/static/js/..*.js
/static/templates
/static/css

Main frontend bundle

/META-INF/MANIFEST.MF
/static/js/angular.js
/static/js/require.js

Generates

Bundle 2

```
define(['angular', #foreach($module in $modules)'/${module.path}/js/${module.moduleFileName}.js',#end], function(ng) {

  var appModule = ng.module('my.app', ['ngRoute', #foreach($module in $modules)'${module.moduleName}',#end], [
    '$locationProvider', '$routeProvider',
    function($locationProvider, $routeProvider) {
      $routeProvider.otherwise({redirectTo: '/curriculum'});
    }
  ]);

  return appModule;
});
```

Angular **module per bundle**

- Integrated single-page app experience
- Modular/maintainable JavaScript
- Separate bundles for frontend modules

- Dynamically generated JS defies optimization
- Addition of frontend bundle requires reload
- Frontend bundles have ‘hidden’ dependencies
- Concerns like (global) navigation tricky

Single frontend bundle

Single Page Web-application: UI-based tooling all the way

Concatenation **Minification** **generate AppCache manifest**

CSS Auto prefixing **CSS pre-processor (SASS/LESS)**

Frontend bundle

```
/META-INF/MANIFEST.MF
/static/module1/*.js
/static/module1/css/*.css
/static/module1/templates/*.html

/static/module2/*.js
/static/module2/css/*.css
/static/module2/templates/*.html

/static/main.js
```

Gulp/
Grunt

Bower

etc.

Processed Frontend bundle

```
/META-INF/MANIFEST.MF
/static/main.js
/static/main.css
```

Angular **module per bundle**

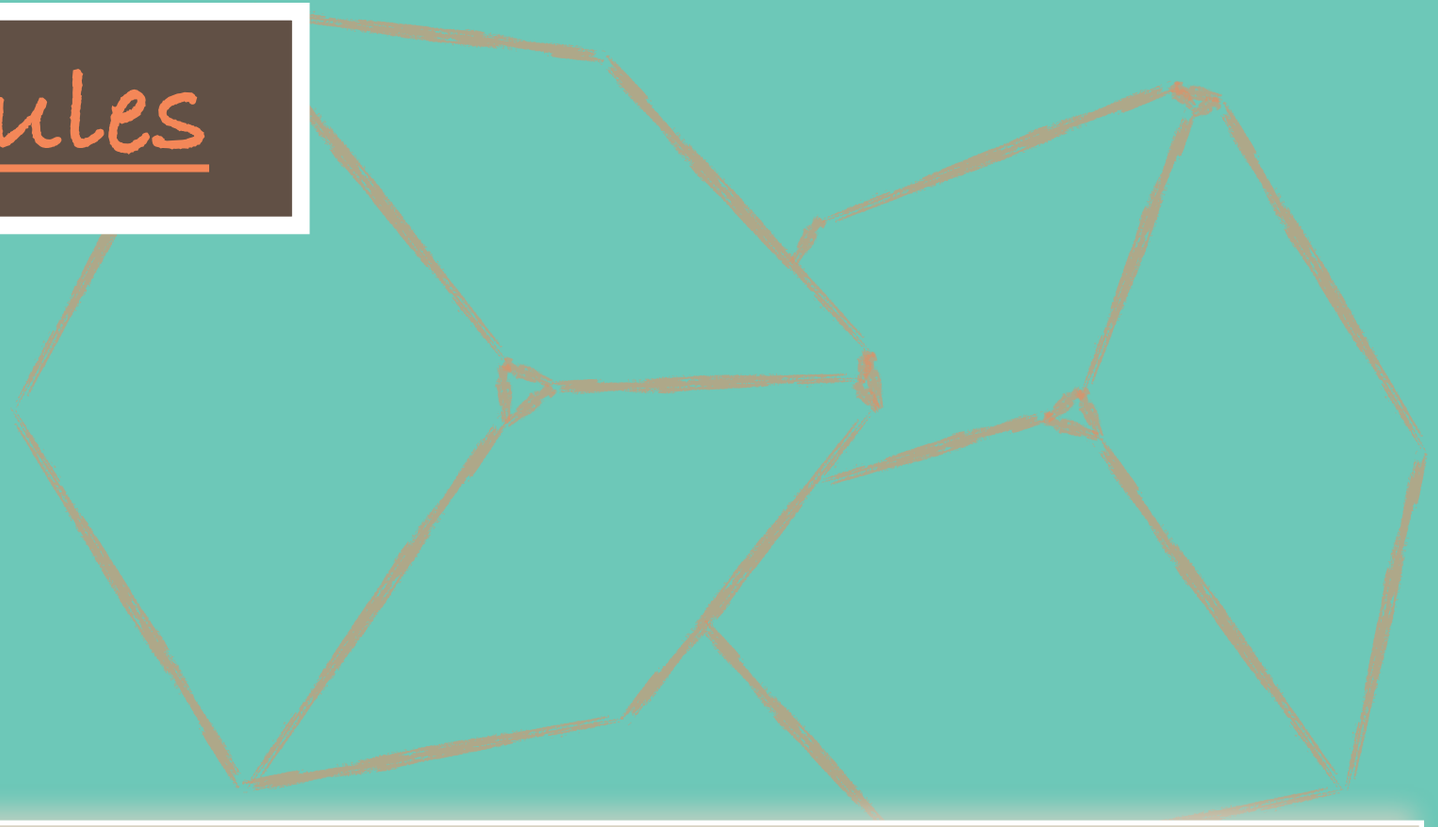
- Integrated single-page app experience
- Modular/maintainable JavaScript
- Use modern frontend toolchains

- Lose dynamic addition of frontend modules

What about...

‘The Future’

Future: ES6 *modules*



- ‘Transpile’ ES 6 modules
  - Google Traceur or Square’s ES transpiler
  - But: spec far from final

Future: JavaScript services

OSGi RFP 159

JavaScript Micro Services

https://github.com/osgi/design/raw/master/rfps/rfp-0159-JavaScript-Microservices.pdf

Future: JavaScript services

OSGi RFP 159

JavaScript Micro Services

-> No activity since August 2013

https://github.com/osgi/design/raw/master/rfps/rfp-0159-JavaScript-Microservices.pdf

Future: web components

W3C working draft

- Custom elements - **'the DOM is the framework'**
- Templates/Shadow DOM - **iFrameless encapsulation**
- HTML imports - **native #include for the web**
- Native 2-way databinding - **convenient & fast**

Future: web components

W3C working draft

- Custom elements - **'the DOM is the framework'**
- Templates/Shadow DOM - **iFrameless encapsulation**
- HTML imports - **native #include for the web**
- Native 2-way databinding - **convenient & fast**

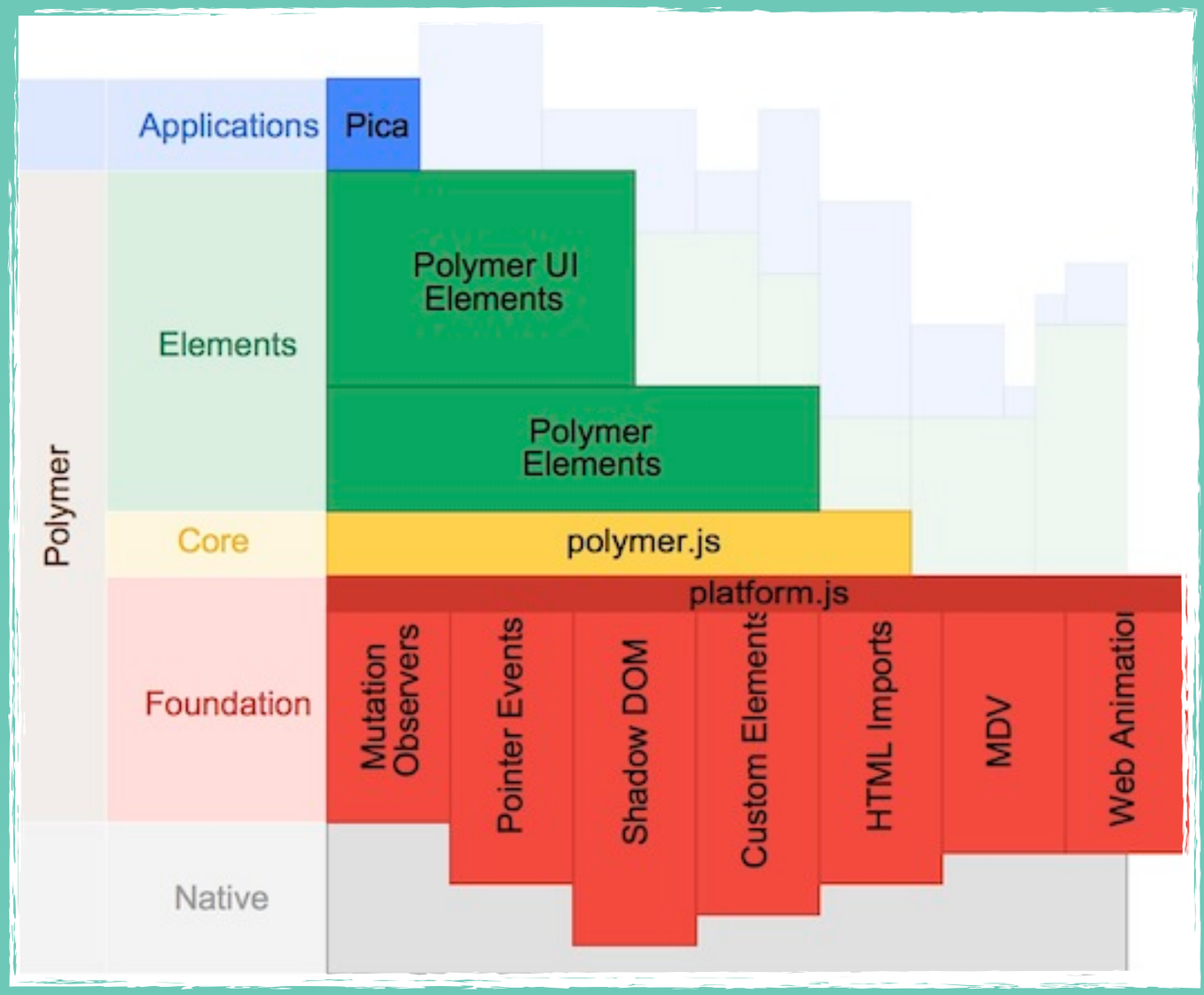
Not strictly about JS modularity:
supposed to piggyback on ES6 modules

Future: web components

Google Polymer

mozilla X-Tags

- Poly-fills
- Pre-defined custom elements
- AngularJS and Ember.js will migrate to web components



Future: Web Components

color-picker.html

```
<link rel="import" href="/components/polymer/polymer.html">

<polymer-element name="color-picker" attributes="owner color">
  <template>
    This is a <strong>{{owner}}</strong>'s color-picker.
    He likes the color <b style="color: {{color}}">{{color}}</b>.
  </template>
  <script>
    Polymer('color-picker', {
      // These default values are overridden
      // by the user's attribute values.
      color: "red",
      owner: "Daniel"
    });
  </script>
</polymer-element>
```

Future: Web Components

color-picker.html

```
<link rel="import" href="/components/polymer/polymer.html">

<polymer-element name="color-picker" attributes="owner color">
  <template>
    This is a <strong>{{owner}}</strong>'s color-picker.
    He likes the color <b style="color: {{color}}">{{color}}</b>.
  </template>
  <script>
    Polymer('color-picker', {
      // These default values are overridden
      // by the user's attribute values.
      color: "red",
      owner: "Daniel"
    });
  </script>
</polymer-element>
```

HTML import

Future: Web Components

color-picker.html

```
<link rel="import" href="/components/polymer/polymer.html">

<polymer-element name="color-picker" attributes="owner color">
  <template>
    This is a <strong>{{owner}}</strong>'s color-picker.
    He likes the color <b style="color: {{color}}">{{color}}</b>.
  </template>
  <script>
    Polymer('color-picker', {
      // These default values are overridden
      // by the user's attribute values.
      color: "red",
      owner: "Daniel"
    });
  </script>
</polymer-element>
```

Custom element definition

Future: Web Components

color-picker.html

```
<link rel="import" href="/components/polymer/polymer.html">

<polymer-element name="color-picker" attributes="owner color">
  <template>
    This is a <strong>{{owner}}</strong>'s color-picker.
    He likes the color <b style="color: {{color}}">{{color}}</b>.
  </template>
  <script>
    Polymer('color-picker', {
      // These default values are overridden
      // by the user's attribute values.
      color: "red",
      owner: "Daniel"
    });
  </script>
</polymer-element>
```

2-way databinding

Future: Web Components

color-picker.html

```
<link rel="import" href="/components/polymer/polymer.html">

<polymer-element name="color-picker" attributes="owner color">
  <template>
    This is a <strong>{{owner}}</strong>'s color-picker.
    He likes the color <b style="color: {{color}}">{{color}}</b>.
  </template>
  <script>
    Polymer('color-picker', {
      // These default values are overridden
      // by the user's attribute values.
      color: "red",
      owner: "Daniel"
    });
  </script>
</polymer-element>
```

Registration/initialization
(Polymer specific, ultimately calls into Web Components standard API)

Future: Web Components

color-picker.html

```
<link rel="import" href="/components/polymer/polymer.html">

<polymer-element name="color-picker" attributes="owner color">
  <template>
    This is a <strong>{{owner}}</strong>'s color-picker.
    He likes the color <b style="color: {{color}}">{{color}}</b>.
  </template>
  <script>
    Polymer('color-picker', {
      // These default values are overridden
      // by the user's attribute values.
      color: "red",
      owner: "Daniel"
    });
  </script>
</polymer-element>
```

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <script src="platform.js"></script>
    <link rel="import" href="color-picker.html">
  </head>
  <body>
    <color-picker owner="Scott" color="blue"></color-picker>
  </body>
</html>
```

Questions?

@sander_mak

bit.ly/modularjs