

# THE ERA OF MODULE BUNDLERS

Arun Michael Dsouza

Software Engineer, AdPushup Inc.

[github.com/arunmichaeldsouza](https://github.com/arunmichaeldsouza)

[twitter.com/amdsouza92](https://twitter.com/amdsouza92)

JSConf Belgium 2017



Arun Michael Dsouza

📍 New Delhi, India

[github.com/arunmichaeldsouza](https://github.com/arunmichaeldsouza)

[twitter.com/amdsouza92](https://twitter.com/amdsouza92)



Ad Revenue Optimisation Platform

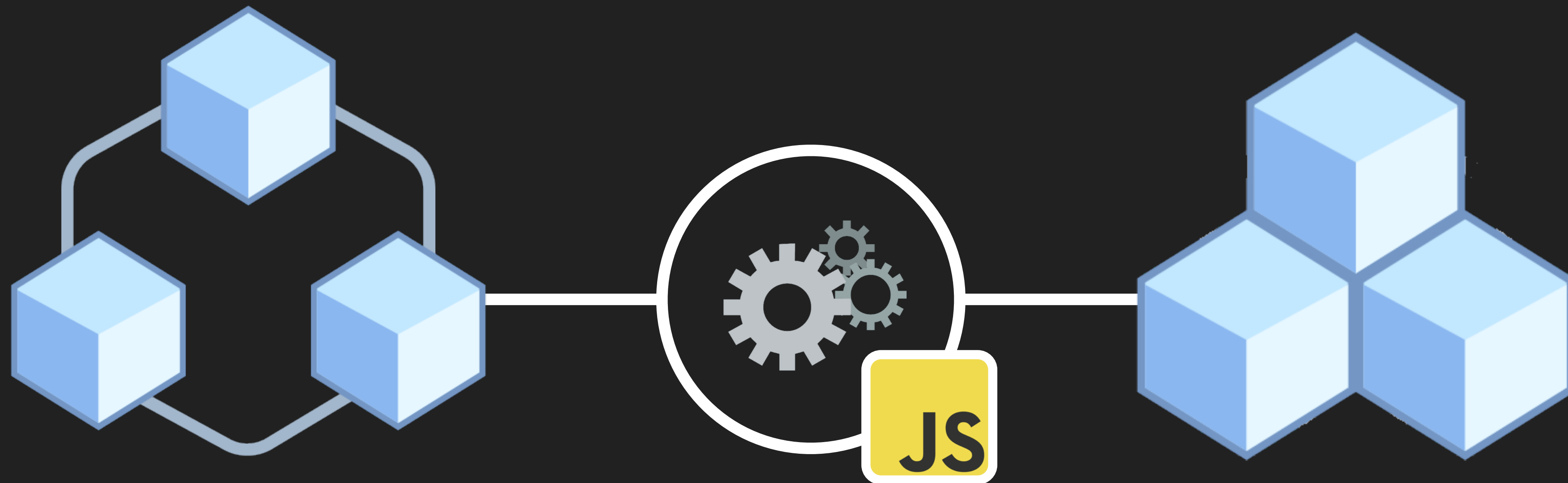
[adpushup.com](https://adpushup.com)

*adrecover*

Adblock Monetisation Platform

[adrecover.com](https://adrecover.com)

# THE ERA OF MODULE BUNDLERS

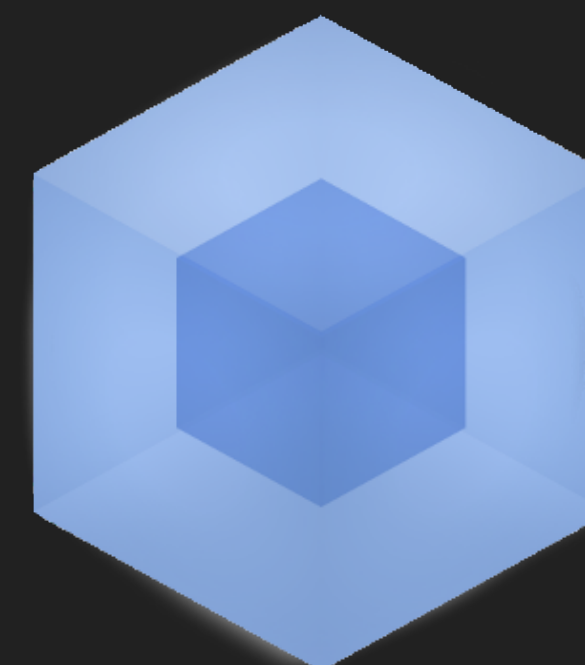


<http://bit.ly/Era-of-module-bundlers-demos>

## Task Runners



## Module Bundlers



# JAVASCRIPT MODULES

**MAINTAINABILITY**

**REUSABILITY**

**EASE OF TESTING**

# MODULE SYSTEMS IN JAVASCRIPT



```
(function() {  
    // Module code  
})();
```

Immediately Invoked Function Expression

```
(function() {  
    // Module code  
})();
```

```
{  
    // Module code  
}
```

<http://bit.ly/block-scoping-es6>

```
var math = (function() {  
    return {  
        sum: function() {  
            // Method code  
        },  
        product: function() {  
            // Method code  
        }  
    };  
})();
```

## Object Interface

<http://bit.ly/design-patterns-javascript>

# AMD

---

# COMMON JS

# AMD

```
define([], function() {  
    return {  
        sum: function() {  
            // Method code  
        },  
        product: function() {  
            // Method code  
        }  
    };  
});
```

math.js

# AMD

```
define([  
    'math',  
    'data'  
], function(math, data) {  
    return math.sum(2, 3);  
});
```

app.js

# COMMON JS

```
var math = {  
  sum: function() {  
    // Method code  
  },  
  product: function() {  
    // Method code  
  }  
};
```

```
module.exports = math;
```

math.js

# COMMON JS

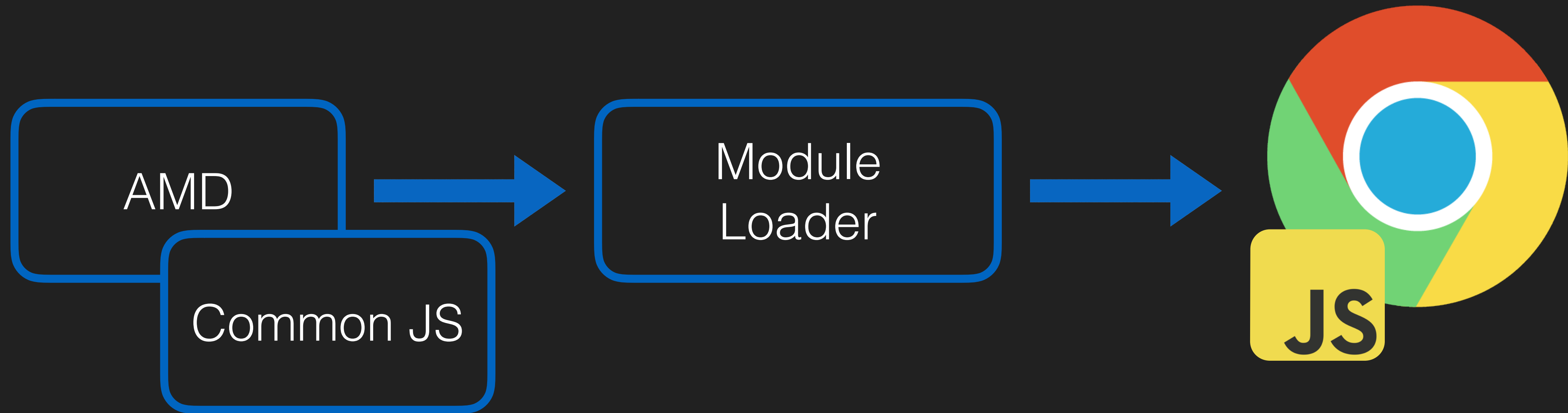
```
var math = require('./math');
```

```
var value = math.sum(2, 3);
```

app.js



# MODULE LOADERS



# REQUIRE JS

AMD Module Loader

<http://requirejs.org/>

# SYSTEM JS

Common JS / AMD Module Loader

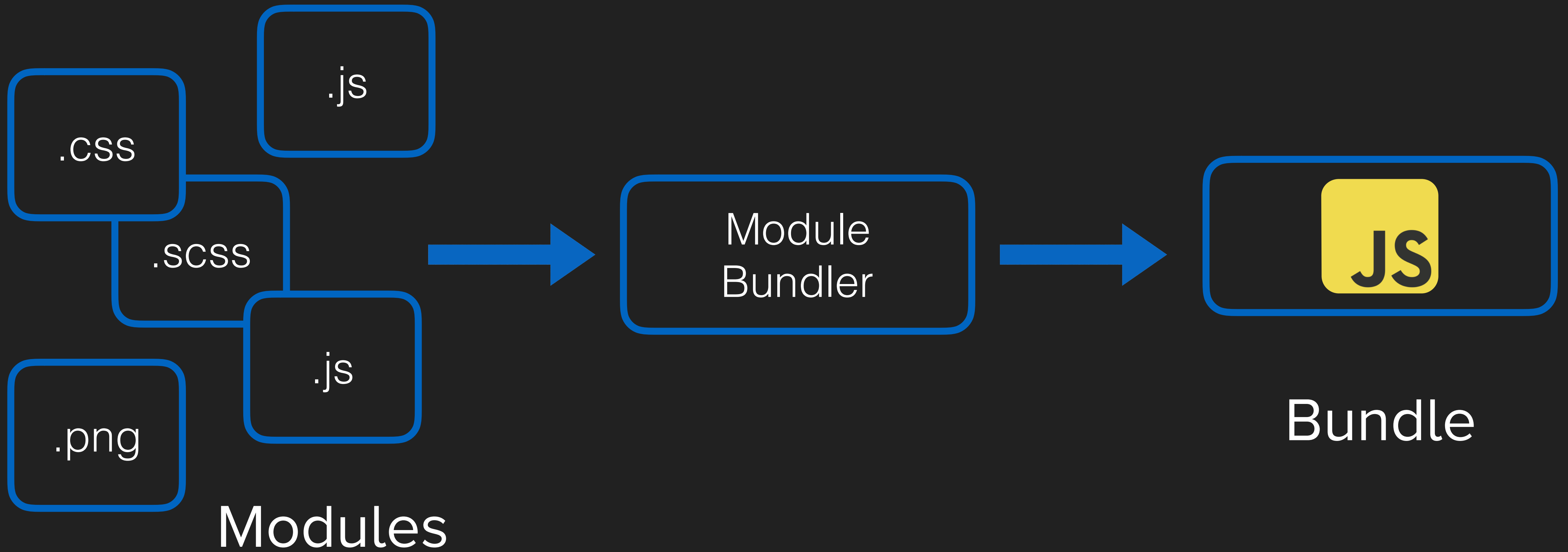
<http://bit.ly/system-js>

```
<html>
<head>
  <title>Page 1</title>
  <script data-main="js/page1" src="require.js">
  </script>
</head>
<body>

</body>
</html>
```

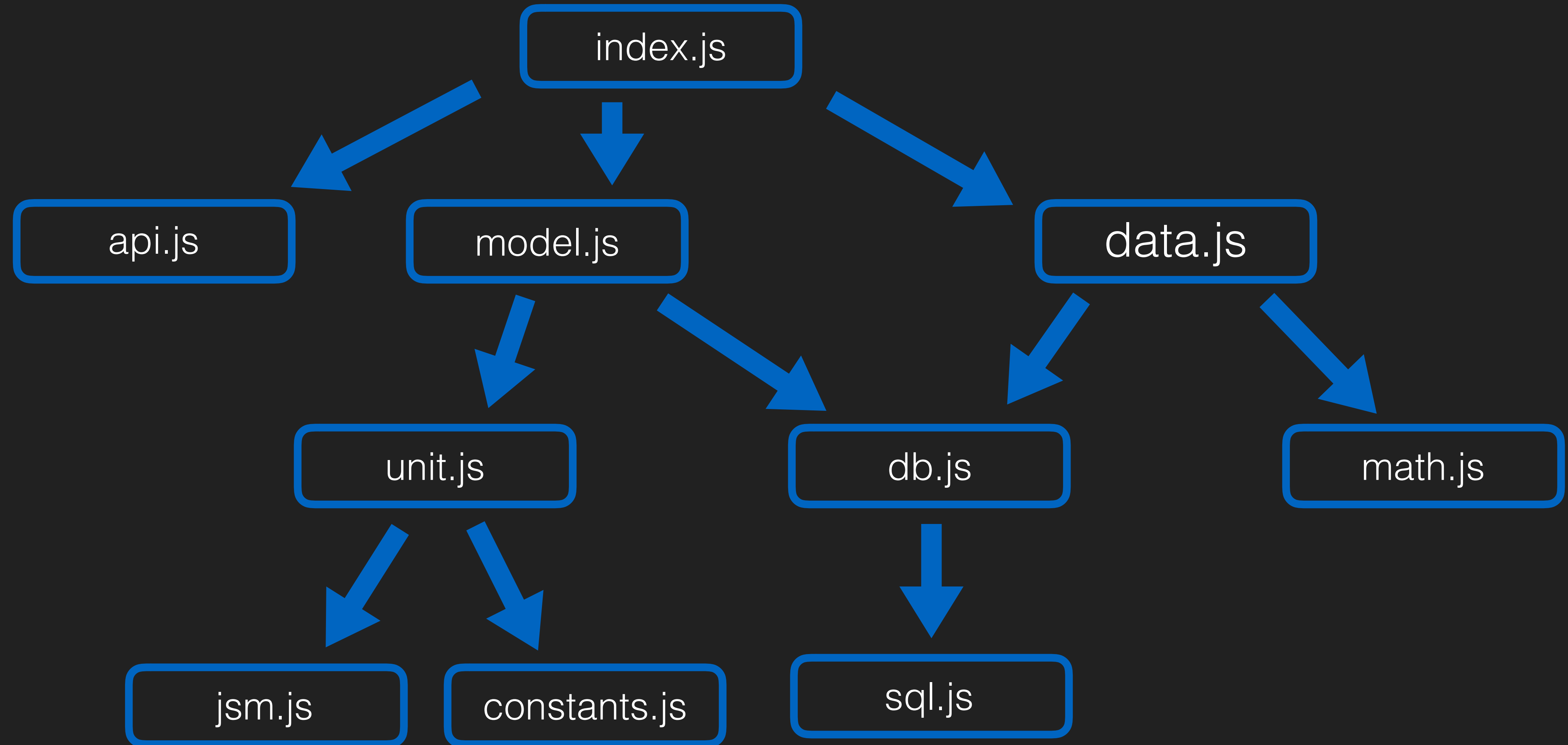
<http://bit.ly/require-js-examples>

# MODULE BUNDLING



```
<html>
  <head>My Webpage</head>
  <body>
    <script src="script1.js"></script>
    <script src="script2.js"></script>
    <script src="script3.js"></script>
    <script src="script4.js"></script>
  </body>
</html>
```

# HOW DO MODULE BUNDLERS WORK ?



Dependency Graph

# BROWSERIFY



NPM modules on the browser side! Yay!

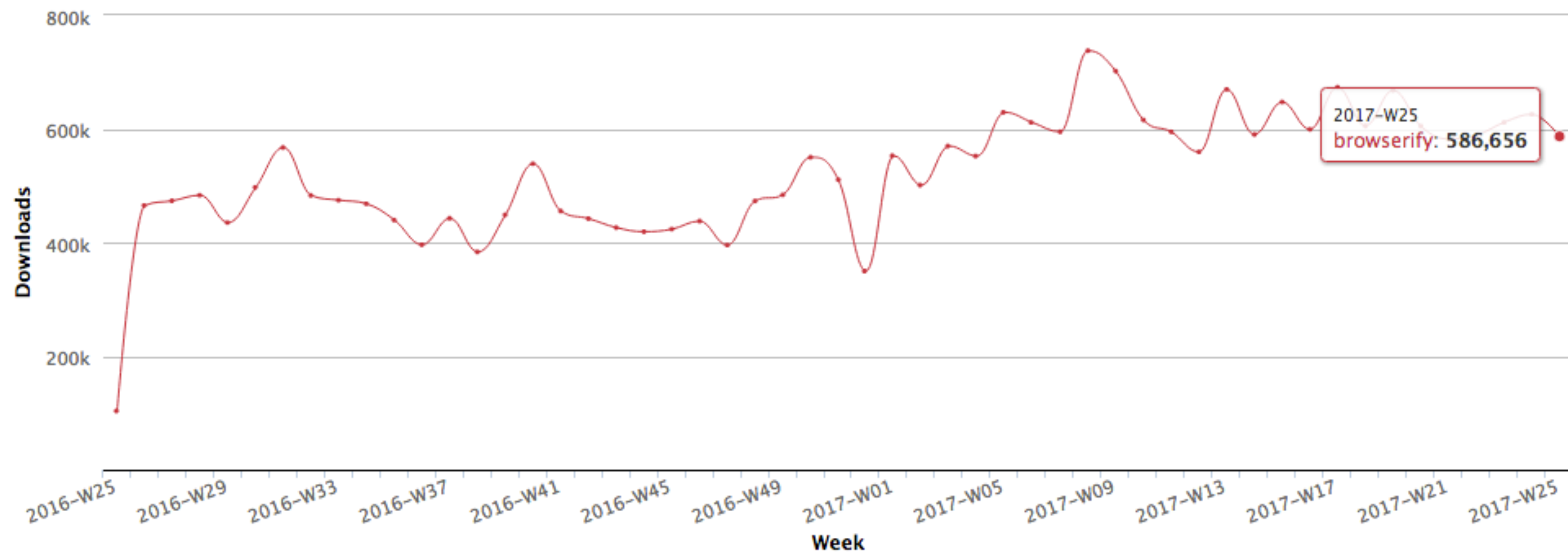






## Downloads per week

Click and drag in the plot to zoom in



# INSTALLATION

```
> npm install -g browserify
```

# USAGE

```
> browserify app.js -o bundle.js
```

# TRANSFORMS



<http://bit.ly/browserify-transforms>

## USING TRANSFORMS

```
> browserify app.js -o bundle.js -t [ babelify --presets  
    [ es2015 ] ]
```

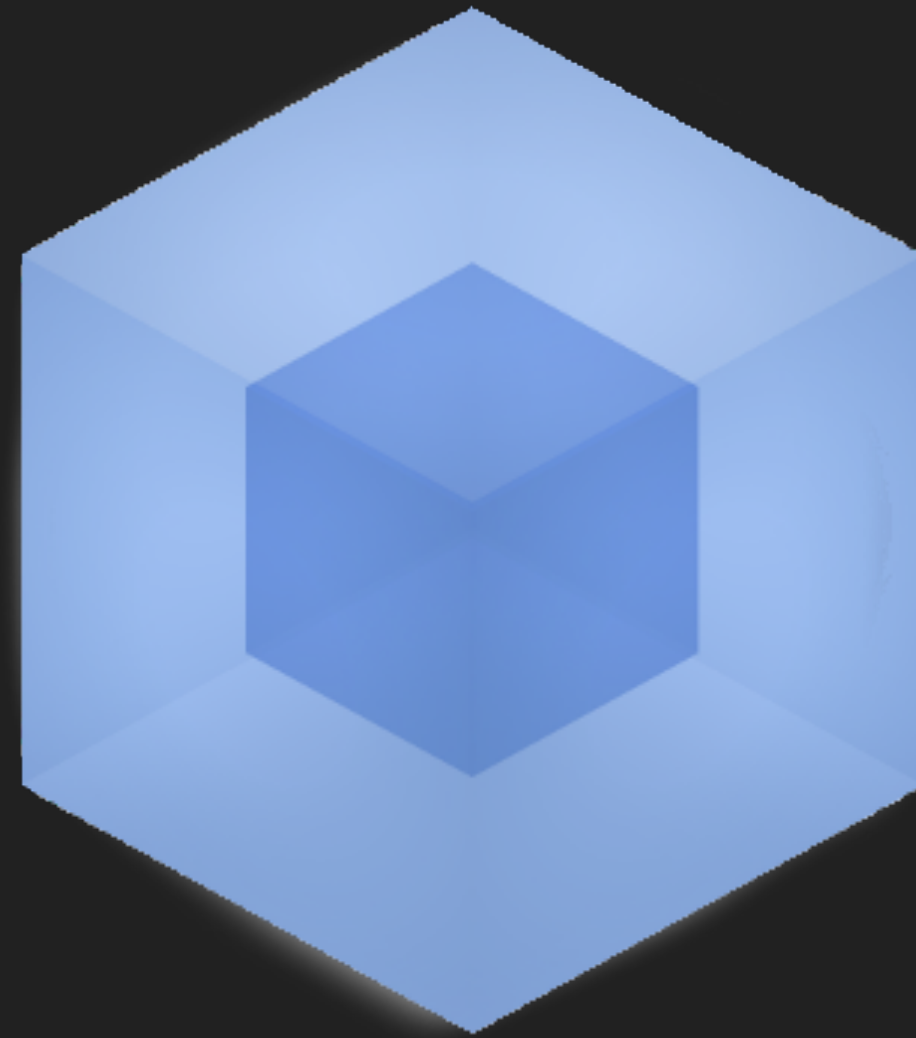
# TASK RUNNERS

```
grunt.initConfig({  
  watchFiles: {  
    // Watcher task  
  },  
  browserify: {  
    // Module bundling task  
  },  
  minifyCss: {  
    // CSS optimisation task  
  },  
  optimiseImages: {  
    // Image optimisation task  
  },  
  uploadOnCDN: {  
    // Deployment task  
  }  
});
```

# BABELIFY TRANSFORM

```
> npm install grunt grunt-browserify babelify babel-  
  preset-es2015 —save-dev
```

# WEBPACK





# INSTALLATION

```
> npm install -g webpack
```

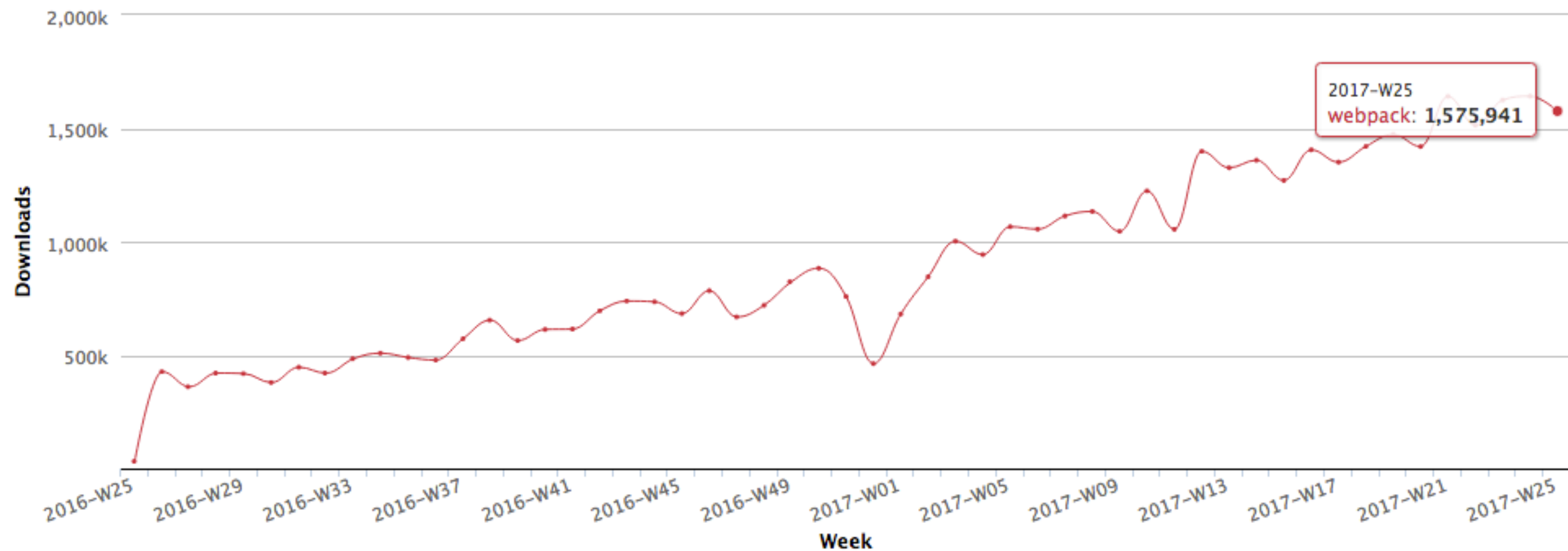
# USAGE

```
> webpack app.js bundle.js
```

# WEBPACK CONFIG FILE

```
module.exports = {  
  entry: './app.js',  
  output: {  
    filename: './bundle.js'  
  }  
};
```

Downloads per week  
Click and drag in the plot to zoom in



# LOADERS

Transform code from one form to another that is part of the bundle

# PLUGINS

Act on the bundle or bundle chunks for customisation and optimisation

# **BUNDLE OPTIMISATION WITH WEBPACK**

# BUNDLE ANALYSIS

# WEBPACK BUNDLE ANALYZERS

## NPM Module

<http://bit.ly/webpack-bundle-analyzer>

## Web-based Analysers

<http://bit.ly/webpack-visualizer>

<http://bit.ly/webpack-chart>

# WEBPACK BUNDLE ANALYZER

---

```
plugins: [  
  new BundleAnalyzerPlugin()  
]
```



# WEB-BASED BUNDLE ANALYZERS

```
> webpack --profile --json > stats.json
```

**PRODUCTION READY BUNDLE**

# UGLIFICATION

```
plugins: [  
  new webpack.optimize.UglifyJsPlugin()  
]
```

# NODE ENVIRONMENT VARIABLE

```
plugins: [  
  new webpack.DefinePlugin({  
    'process.env.NODE_ENV': JSON.stringify('production')  
  })  
]
```

# WEBPACK PRODUCTION FLAG

```
> webpack -p
```

# **BUILD-TIME GZIPPING**

# DYNAMIC GZIPPING

```
var compression = require('compression');  
app.use(compression());
```

# BUILD-TIME GZIPPING

```
> npm install compression-webpack-plugin --save-dev
```

<http://bit.ly/webpack-compression-plugin>



```
app.get('*.js', function (req, res, next) {  
  req.url = req.url + '.gz';  
  res.set('Content-Encoding', 'gzip');  
  next();  
});
```

# TREE SHAKING

Dead Code Elimination

Works with ES2015 Modules

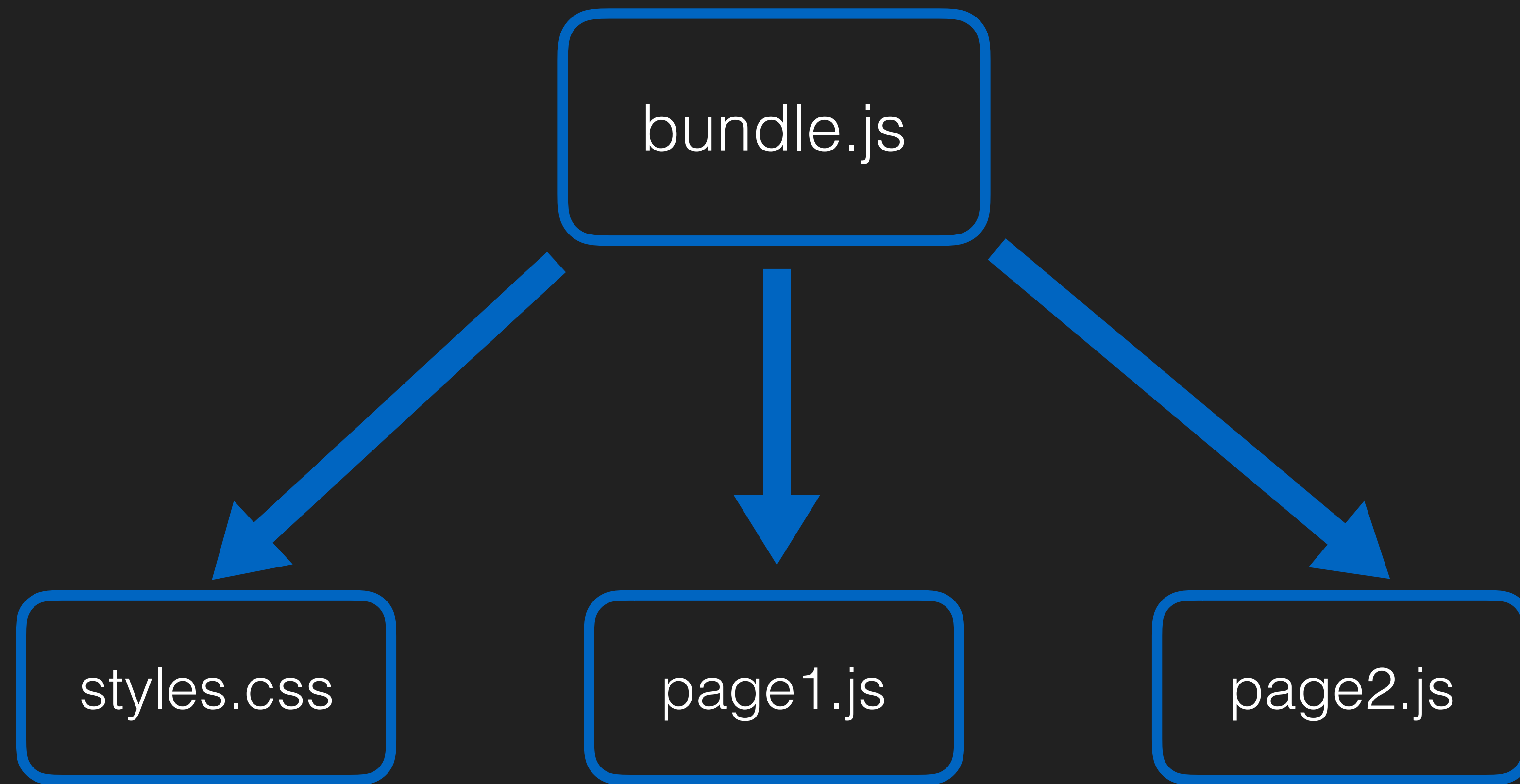
## **ES2015 MODULES (STATIC STRUCTURE)**

Can be tree-shaken

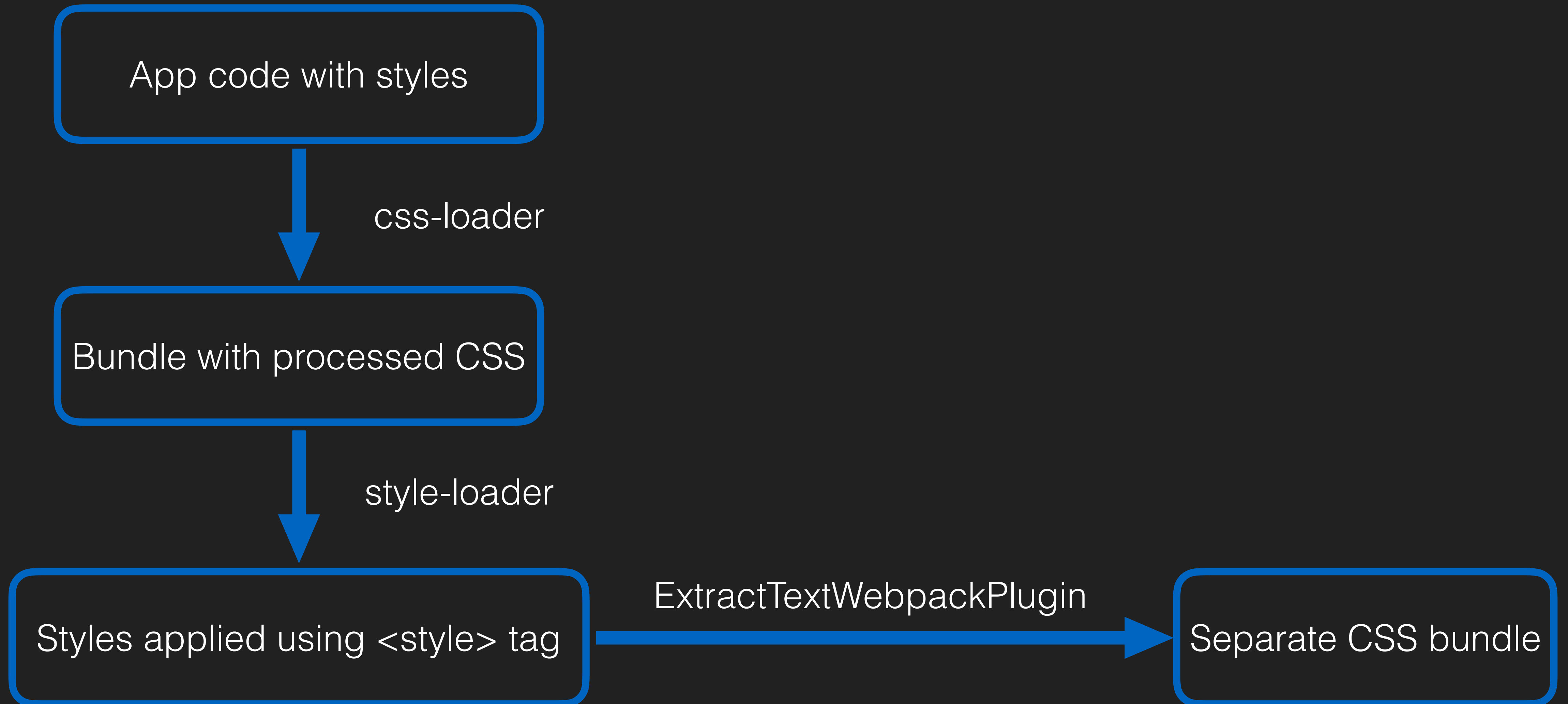
## **COMMON JS & OTHERS (DYNAMIC STRUCTURE)**

Cannot be tree-shaken

# CODE SPLITTING



# CSS CODE SPLITTING



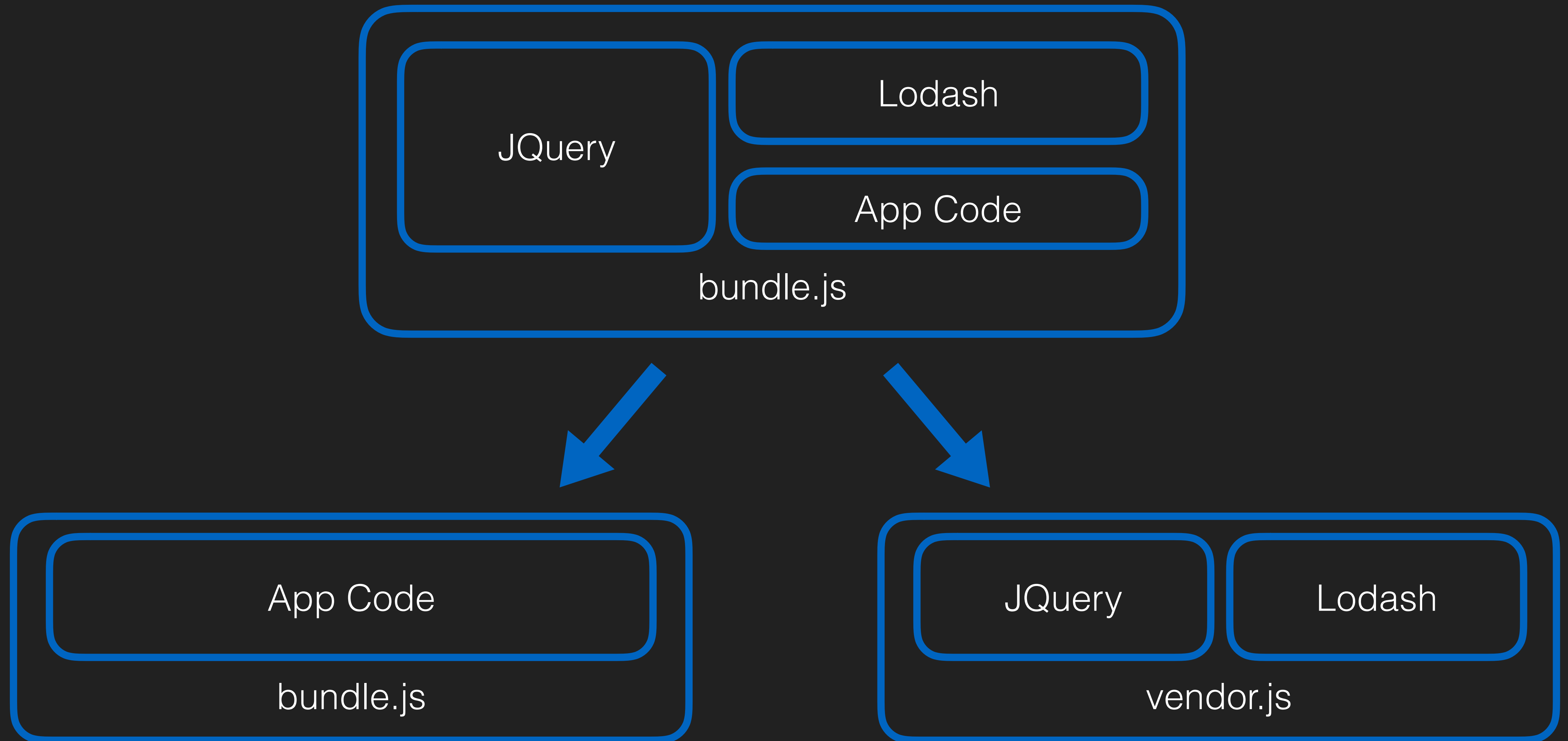
# INSTALLATION

```
> npm install —save-dev extract-text-webpack-plugin
```

# USAGE

```
plugins: [  
  new ExtractTextPlugin('./build/styles.css')  
]
```

# CODE SPLITTING LIBRARIES



# COMMON CHUNKS PLUGIN

```
plugins: [  
  new webpack.optimize.CommonsChunkPlugin(options)  
]
```

<http://bit.ly/common-chunks-plugin>



# IMPLICIT COMMON VENDOR CHUNK

```
plugins: [  
  new webpack.optimize.CommonsChunkPlugin({  
    name: 'vendor',  
    minChunks: function (module) {  
      return module.context && module.context.indexOf('node_modules') !== -1;  
    }  
  })  
]
```

# AUTOMATIC SPLITTING

```
import header from './header';  
import slider from './slider';  
  
require.ensure(['./productList'], function(require) {  
    require('./productList');  
});
```

# FUTURE OF JAVASCRIPT MODULES

Native ES2015 modules

import  
export

**THANK YOU**