THE PROJECTOR WORKS

MODULES EVERYWHERE

CertSimple

EV SSL:

Barclays Bank PLC [GB] https://bank.barclays.co.uk

Stripe, Inc. [US] https://stripe.com/gb

CurrencyTransfer Ltd [GB] https://app.currencytransfer.com

A Medium Corporation [US] https://medium.com

ACTUALLY CHECKS WHO YOU ARE, SHOWS THAT IN BROWSER

40X FASTER

HOW CERTSIMPLE IS BUILT

We use npm on the **backend**

We use npm on the **frontend**
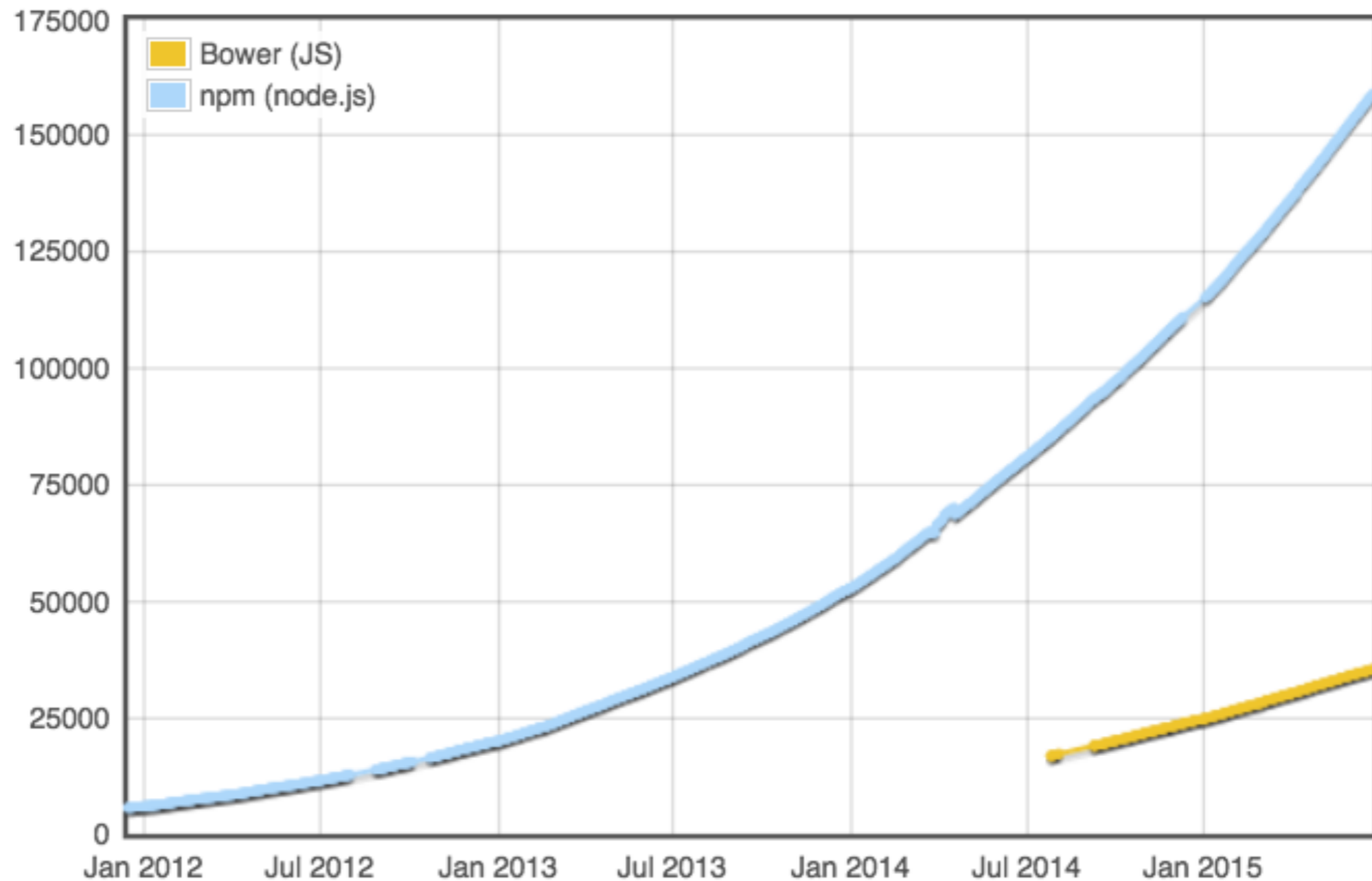
All our code is npm modules

ALL THE THINGS WE WISH WE'D KNOWN EARLIER

Why we chose npm

How you can use npm

Things to avoid

More packages

Every frontend package
CertSimple needed was **officially maintained** on npm

stripe

jQuery
write less. do more.

tags-input

Ractive.js

selectize.js

AGAVE.JS

Attitude to npm over time

General JS things **frontend needed** also on npm

country-data

es6-shim

alpha-sort

array-includes

superagent

USE REAL PROJECTS, NOT STACKOVERFLOW

Other module formats?

- window **globals** are dead

- **RequireJS** never had a module repository - dead

- **UMD** only exists because of globals and Require - just dead code at this point.

- Very few **ES6** packages, and they'll be on npm anyway

Building,
& bundling npm modules

	gulp	npm run-script	webpack
Build Tasks	gulp	npm run-script build	webpack
Bundle	browserify	browserify	webpack

Webpack?

- Can bundle npm packages

- Replaces build system as well

- 8-10% smaller modules + hot reload

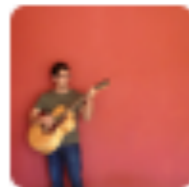- We use a bunch of existing gulp tools, so benefit **not yet** worth it

modules everywhere

Mike MacCana @mikemaccana — 18/05/2015

@izs didn't this used to have 'no lib dir, everything is a module?' in it? Does that still apply? docs.npmjs.com/misc/coding-st...
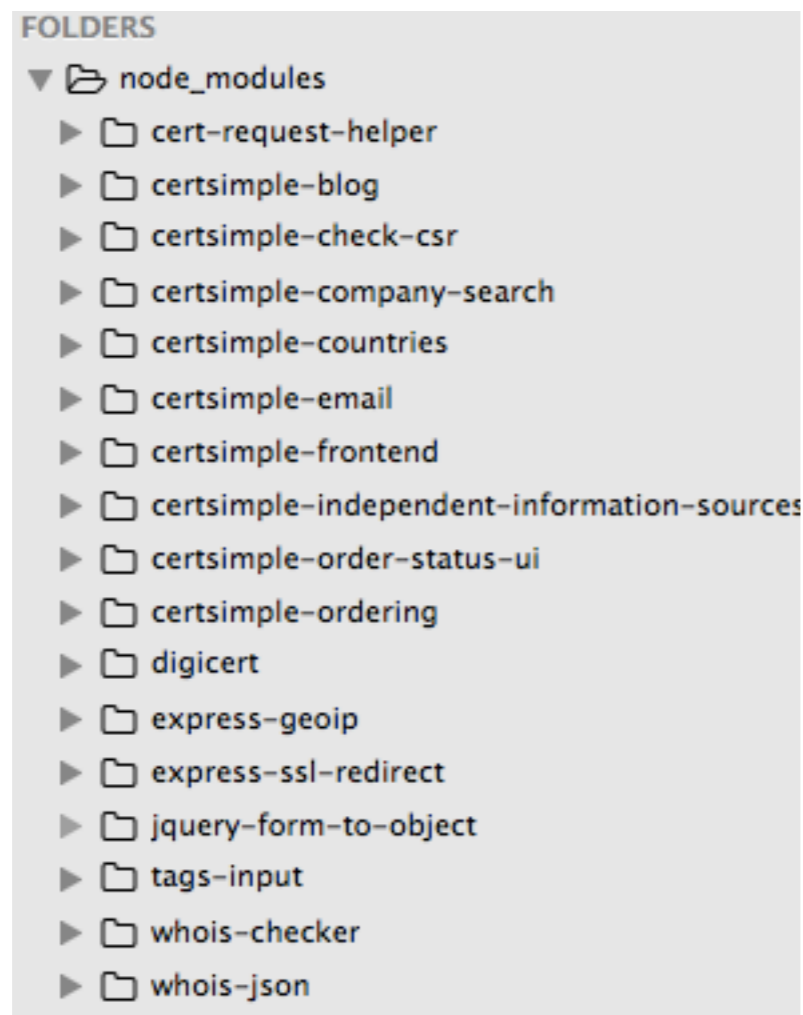
THAT isaac
@izs

@mikemaccana I DO usually do that. But not all the time, and npm's initial design predates npm existing ;)

18/05/2015 16:21

Why just modules?

FOLDERS
- ▼ 📂 node_modules
  - ▶ 📁 cert-request-helper
  - ▶ 📁 certsimple-blog
  - ▶ 📁 certsimple-check-csr
  - ▶ 📁 certsimple-company-search
  - ▶ 📁 certsimple-countries
  - ▶ 📁 certsimple-email
  - ▶ 📁 certsimple-frontend
  - ▶ 📁 certsimple-independent-information-sources
  - ▶ 📁 certsimple-order-status-ui
  - ▶ 📁 certsimple-ordering
  - ▶ 📁 digicert
  - ▶ 📁 express-geoip
  - ▶ 📁 express-ssl-redirect
  - ▶ 📁 jquery-form-to-object
  - ▶ 📁 tags-input
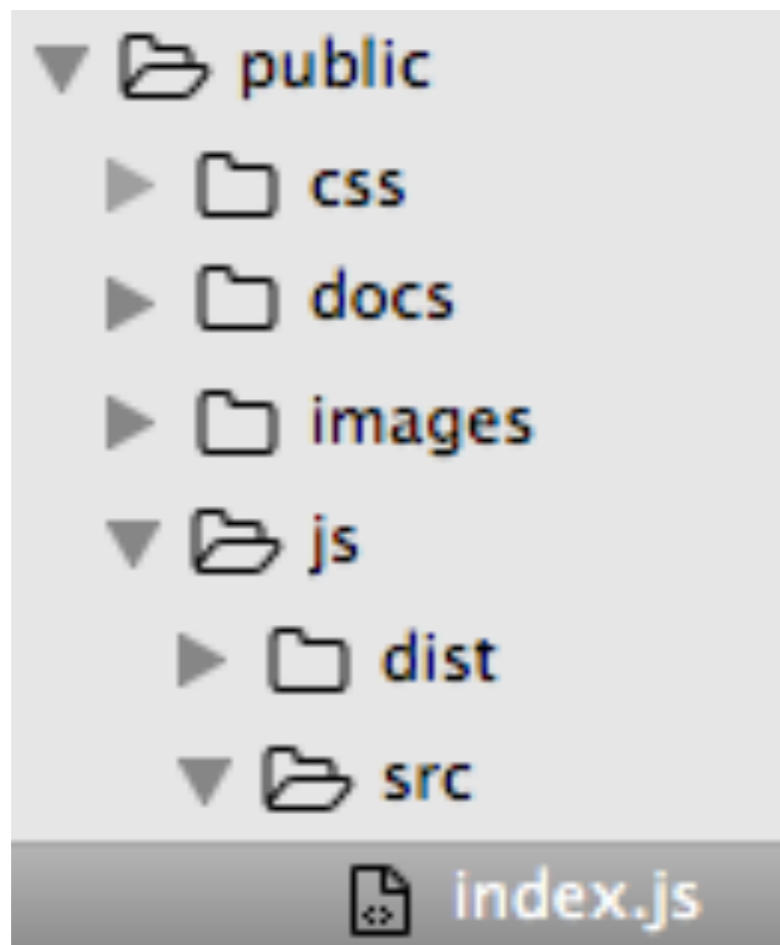 - ▶ 📁 whois-checker
  - ▶ 📁 whois-json

- Simple imports

- modules are a reasonable unit of granularity

- Unlike /lib, every module has 'test', README.md, packages.json, etc.

- Easier to open source

Making **your** modules prominent

```
certsimple.sublime-project  ×

{
    "folders":
    [
        {
            "folder_include_patterns":
            [
                "cert-request-helper",
                "certsimple-frontend",
                "certsimple-blog",
                "certsimple-ordering",
                "certsimple-check-csr",
                "certsimple-independent-information-sources",
                "certsimple-pki",
                "certsimple-company-search",
                "certsimple-digicert-scrape",
                "certsimple-order-status-ui",
                "certsimple-email",
                "certsimple-countries",
                "express-ssl-redirect",
                "express-geoip",
```

Automatically build a Sublime .project file that prioritises private npm modules

Bundling

```
index.js                    ×
require('certsimple-frontend')
```

This is the **entire** js/src/index.js

```
public
  css
  docs
  images
  js
    dist
    src
      index.js
```

```javascript
gulp.task('js', function() {

  // Browserify/bundle the JS.
  browserify({
    entries: './public/js/src/index.js',
    insertGlobals : true,
    fullPaths: true, // For discify
    debug: ! isProduction
  }).bundle()
    .pipe(source('index.js'))
    .pipe(buffer())
    .pipe(gulpIf(isProduction, uglify()))
    .pipe(gulp.dest('./public/js/dist'))
```

Makes a bundle in js/dist/index.js

The 'frontend' module:

```javascript
var Ractive = require("ractive"),
  $ = require("jquery"),
  superagent = require("superagent"),
  scroll = require('scroll'),
  tagsInput = require('tags-input'),
  allCountries = require('certsimple-countries'),
  paymentTemplateHelpers = require('payment-template-helpers');

// https://github.com/stripe/jquery.payment/issues/152
window.$ = $;
require("jquery.payment");
require('selectize');
require('browser-debug-mode');
require('es6-shim');

// Only enable debug when unminified
Ractive.DEBUG = /unminified/.test(function(){/*unminified*/});

var SPACE = 32,
  PAGE_UP = 33;
```
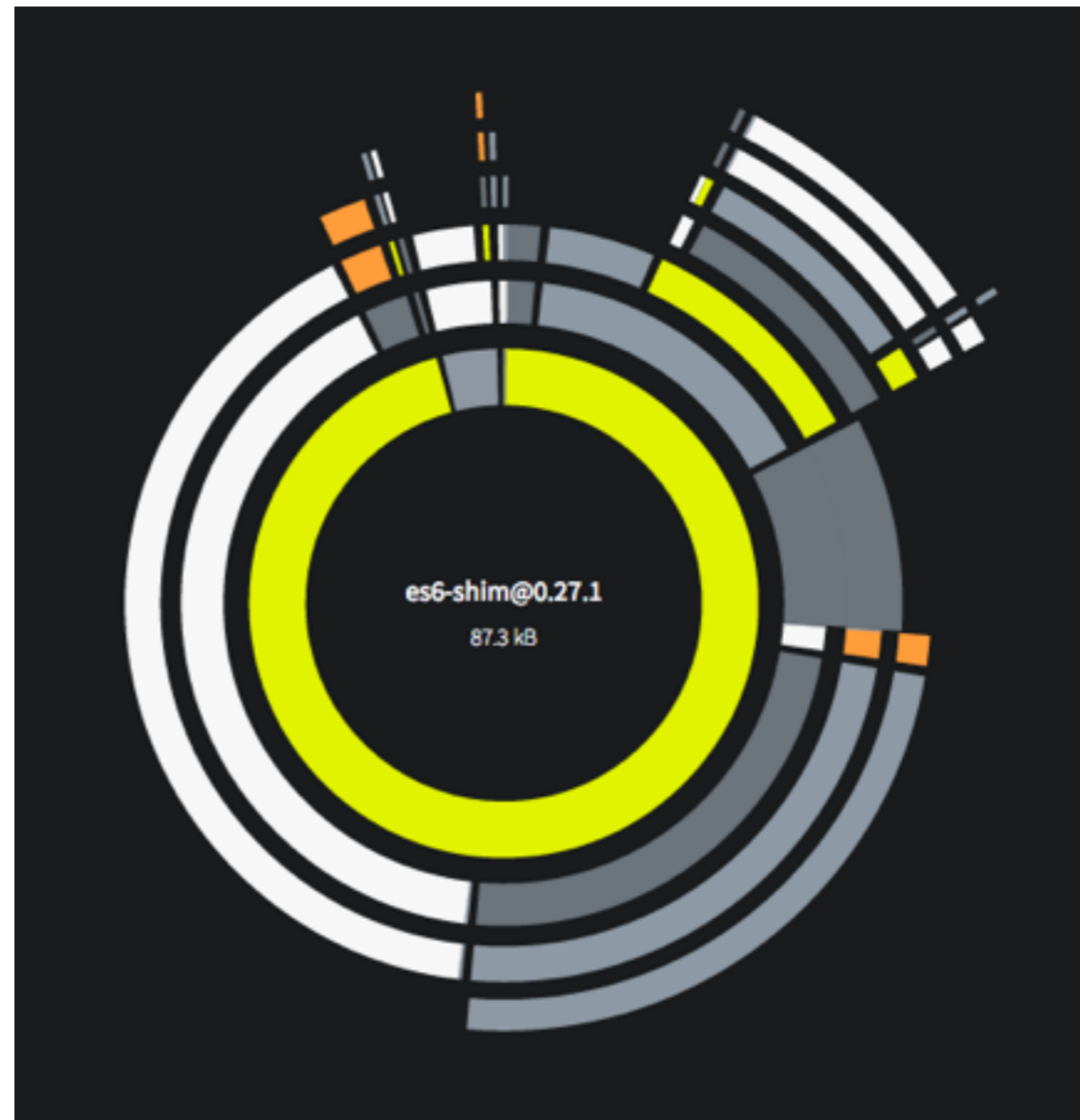
How to bundle templates?

```json
package.json          ×
{
  "name": "certsimple-frontend",
  "version": "1.0.0",
  "private": true,
  "browserify": {
    "transform": [
      "brfs"
    ]
  },
```

```javascript
// Note this can't be merged into another var statement, due to the way brfs works.
var fs = require("fs");

var evCertificatePartial = fs.readFileSync(__dirname + '/ev-certificate.mustache', 'utf8');

var signupStepsPartial = fs.readFileSync(__dirname + '/signup-steps.mustache', 'utf8');
```

Analysing bundle size:

```
index.js                    ×
require('certsimple-frontend')
```


es6-shim@0.27.1
87.3 kB

Deploying

- package.json only specified version for your immediate dependencies

  - a version 1.0.0 - only use b version 1.2.0

    - b version 1.2.0 - use version c 1.0.0 or newer

      - c - could be any version

- package.json alone isn't enough to ensure C has a consistent version

npm shrinkwrap

- Provides **consistent deploys**

- Specifies the full tree, all the way down

- Built into npm

- Avoids overhead of committing node modules

THANKS!

Links!

Module counts
http://www.modulecounts.com/

Discify
http://hughsk.io/disc/

Sublime prioritise private modules
https://gist.github.com/mikemaccana/fe5c1a83bcbdc0157534

Browserify + gulp
https://github.com/gulpjs/gulp/blob/master/docs/recipes/browserify-uglify-sourcemap.md

Webpack vs Browserify
https://github.com/k88hudson/browserify-webpack#results