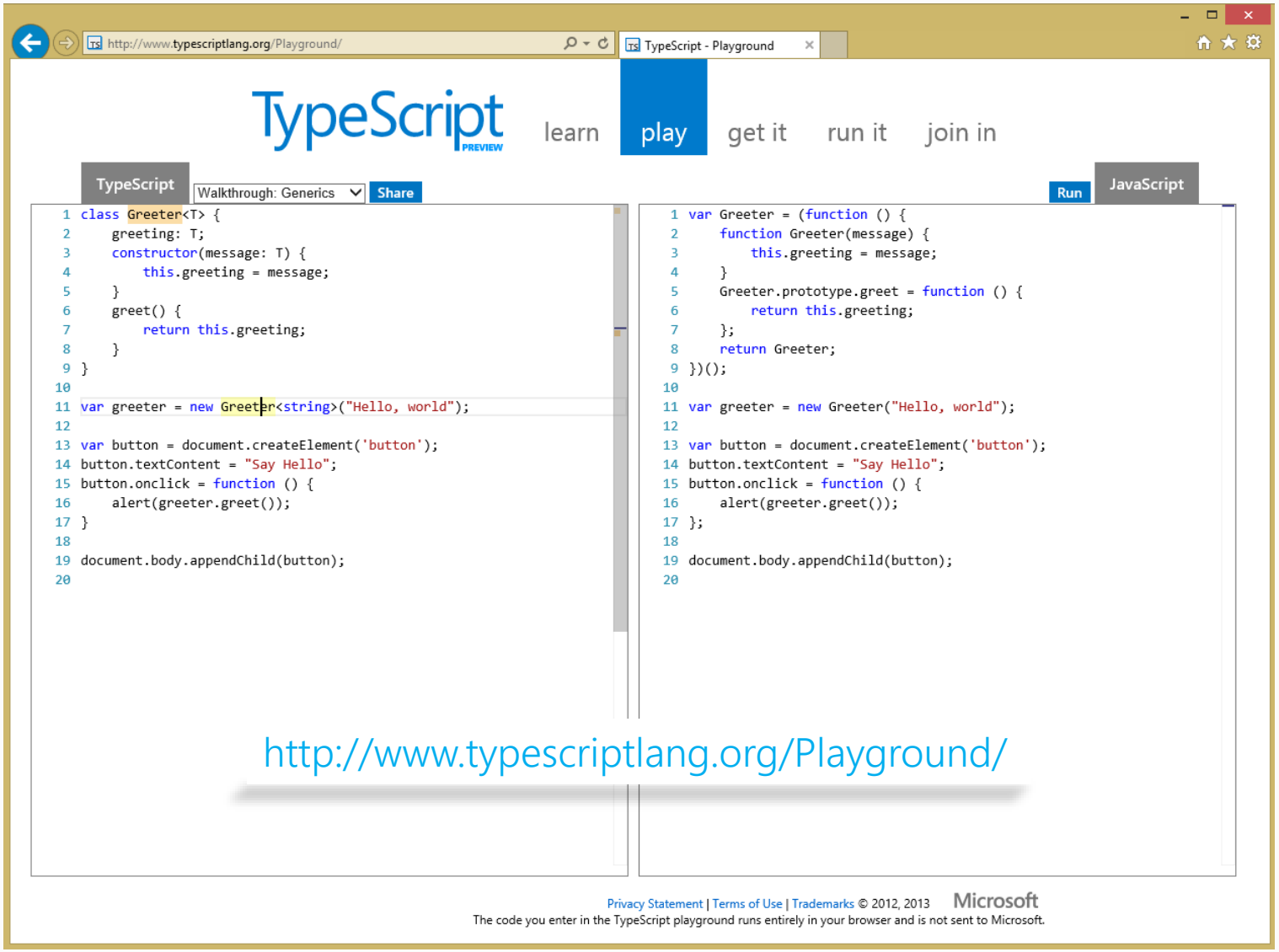
Building a large scale JavaScript application in TypeScript

Benjamin Pasero
Microsoft Monaco Team

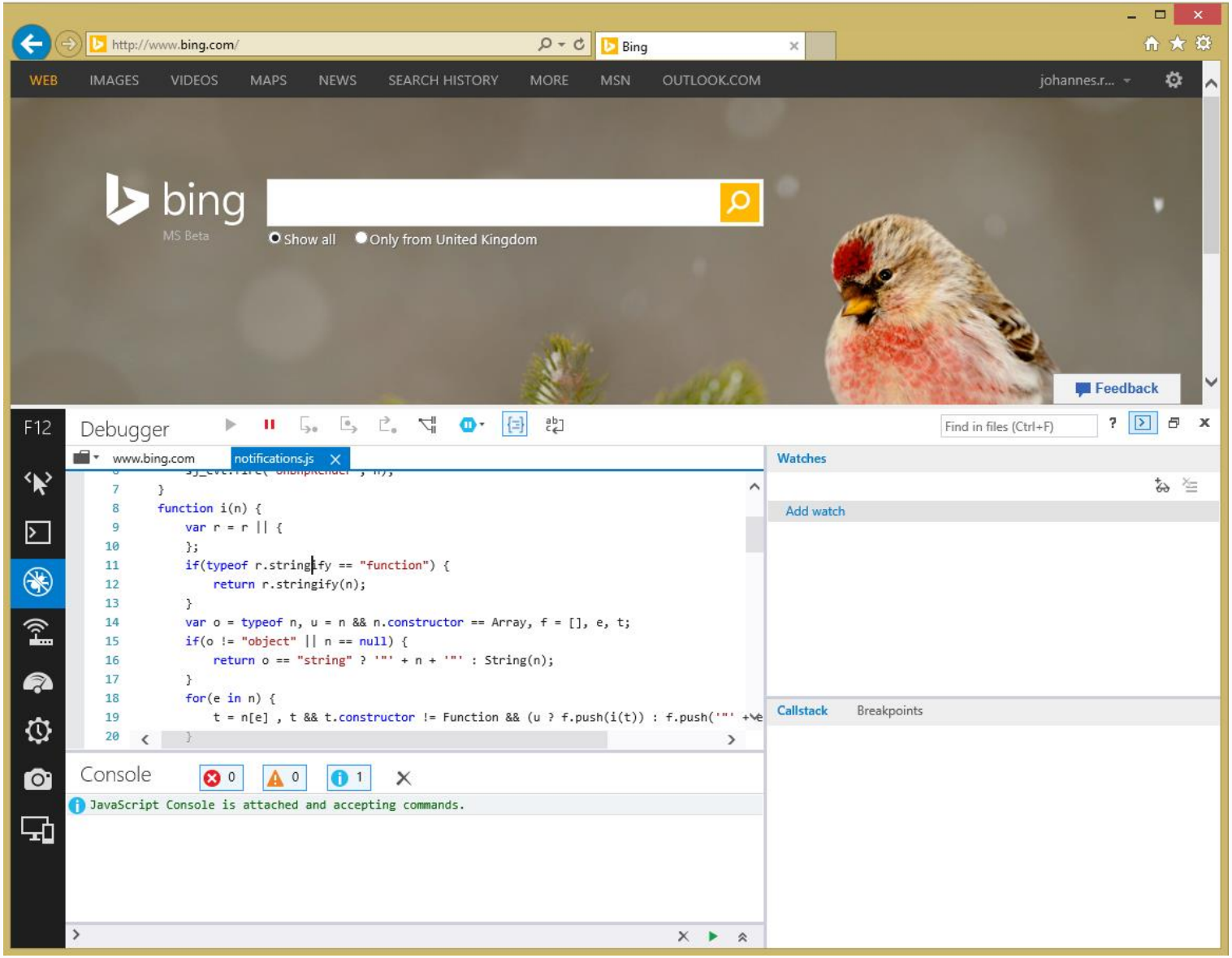
What do we do?

Modern, web standards based developer tools, platform, and cloud services

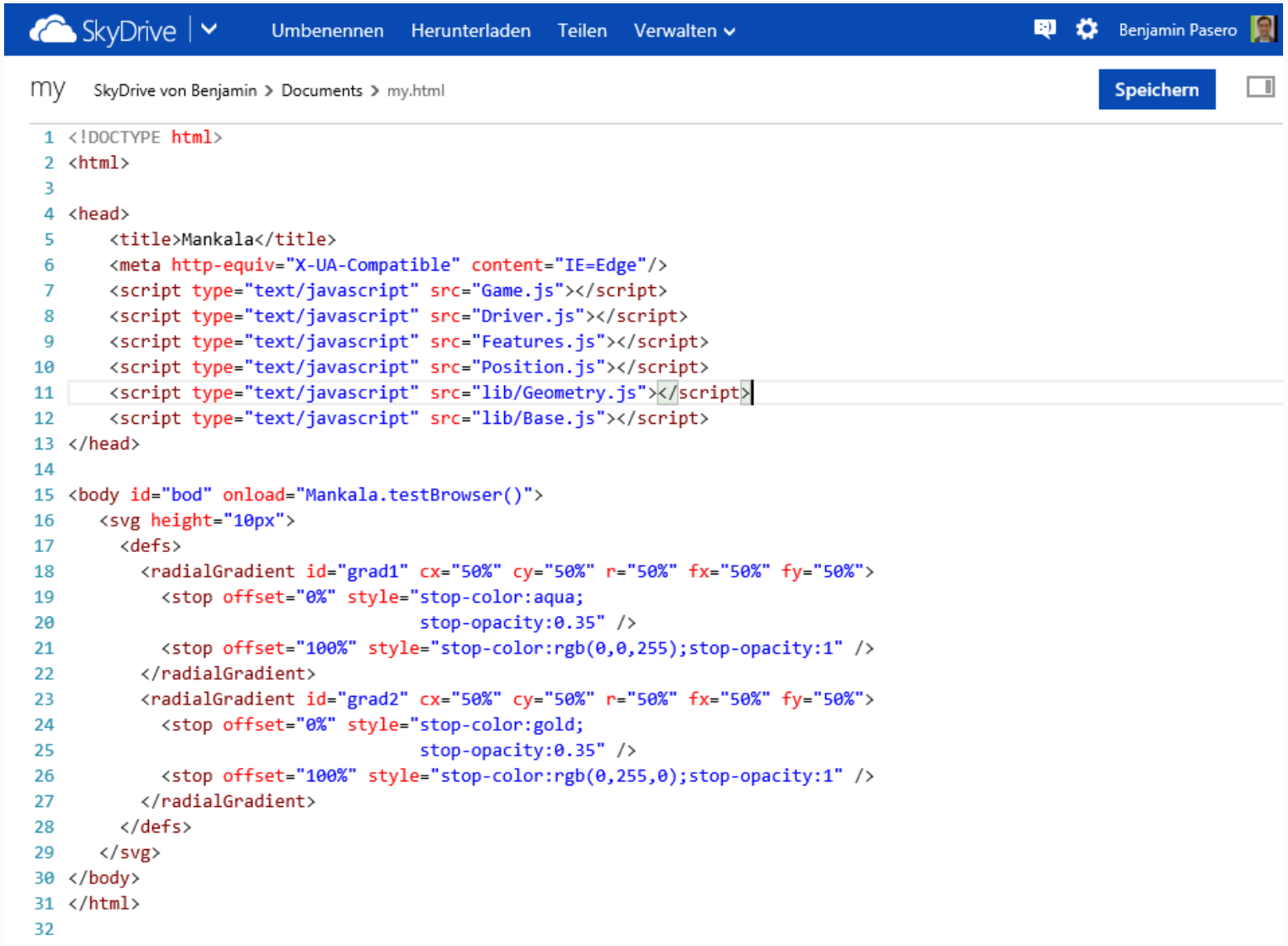
TypeScript Playground



IE F12

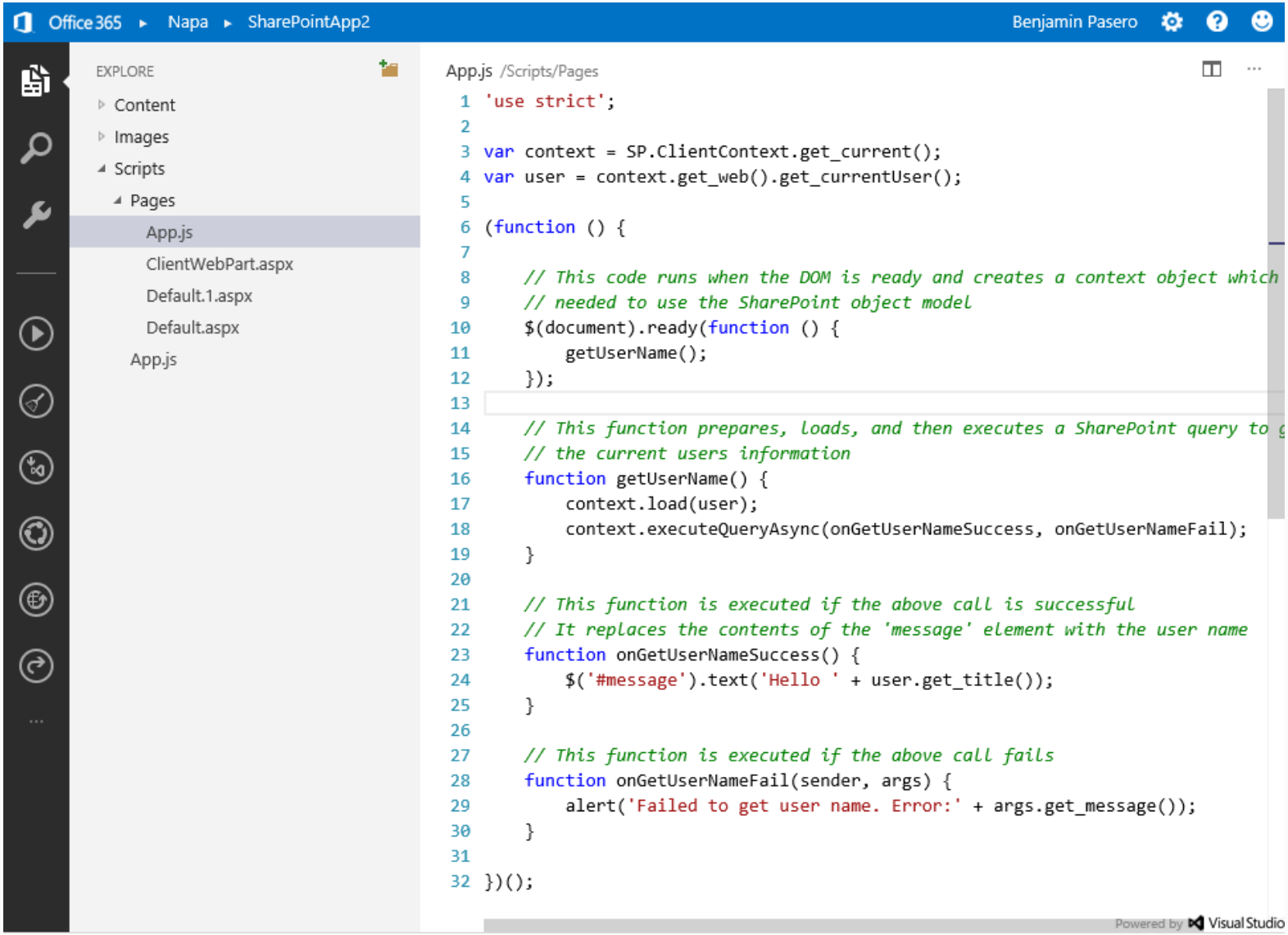

http://www.bing.com/
Bing
WEB
IMAGES
VIDEOS
MAPS
NEWS
SEARCH HISTORY
MORE
MSN
OUTLOOK.COM
johannes.r...
bing
MS Beta
Show all
Only from United Kingdom
Feedback
F12
Debugger
Find in files (Ctrl+F)
www.bing.com
notifications.js
function i(n) {
var r = r || {
};
if(typeof r.stringify == "function") {
return r.stringify(n);
}
var o = typeof n, u = n && n.constructor == Array, f = [], e, t;
if(o != "object" || n == null) {
return o == "string" ? '"' + n + '"' : String(n);
}
for(e in n) {
t = n[e] , t && t.constructor != Function && (u ? f.push(i(t)) : f.push('"' +
Watches
Add watch
Callstack
Breakpoints
Console
JavaScript Console is attached and accepting commands.

~~Sky~~Drive

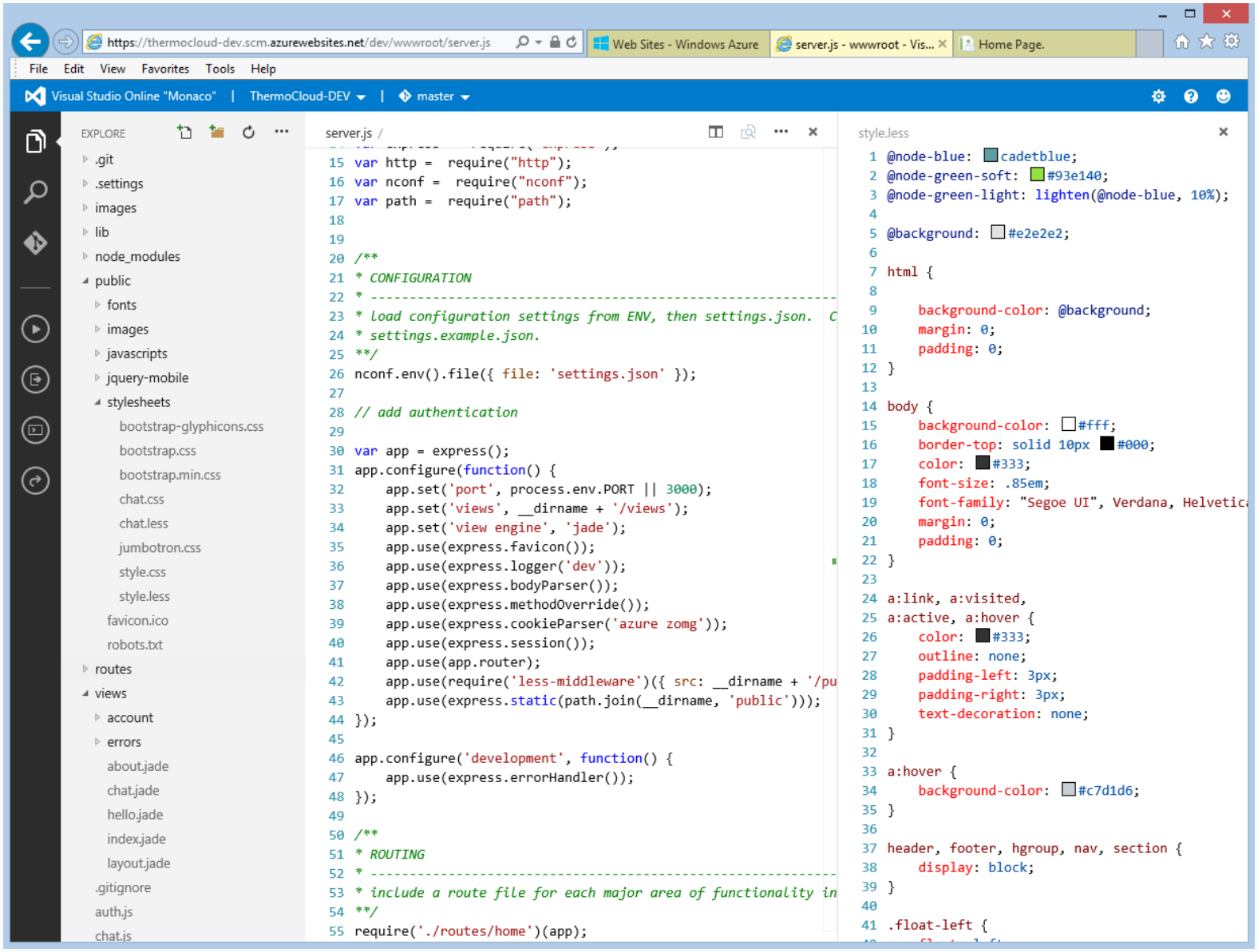



```
<!DOCTYPE html>
<html>

<head>
    <title>Mankala</title>
    <meta http-equiv="X-UA-Compatible" content="IE=Edge"/>
    <script type="text/javascript" src="Game.js"></script>
    <script type="text/javascript" src="Driver.js"></script>
    <script type="text/javascript" src="Features.js"></script>
    <script type="text/javascript" src="Position.js"></script>
    <script type="text/javascript" src="lib/Geometry.js"></script>
    <script type="text/javascript" src="lib/Base.js"></script>
</head>

<body id="bod" onload="Mankala.testBrowser()">
   <svg height="10px">
     <defs>
       <radialGradient id="grad1" cx="50%" cy="50%" r="50%" fx="50%" fy="50%">
         <stop offset="0%" style="stop-color:aqua;
                                  stop-opacity:0.35" />
         <stop offset="100%" style="stop-color:rgb(0,0,255);stop-opacity:1" />
       </radialGradient>
       <radialGradient id="grad2" cx="50%" cy="50%" r="50%" fx="50%" fy="50%">
         <stop offset="0%" style="stop-color:gold;
                                  stop-opacity:0.35" />
         <stop offset="100%" style="stop-color:rgb(0,255,0);stop-opacity:1" />
       </radialGradient>
     </defs>
   </svg>
</body>
</html>

```

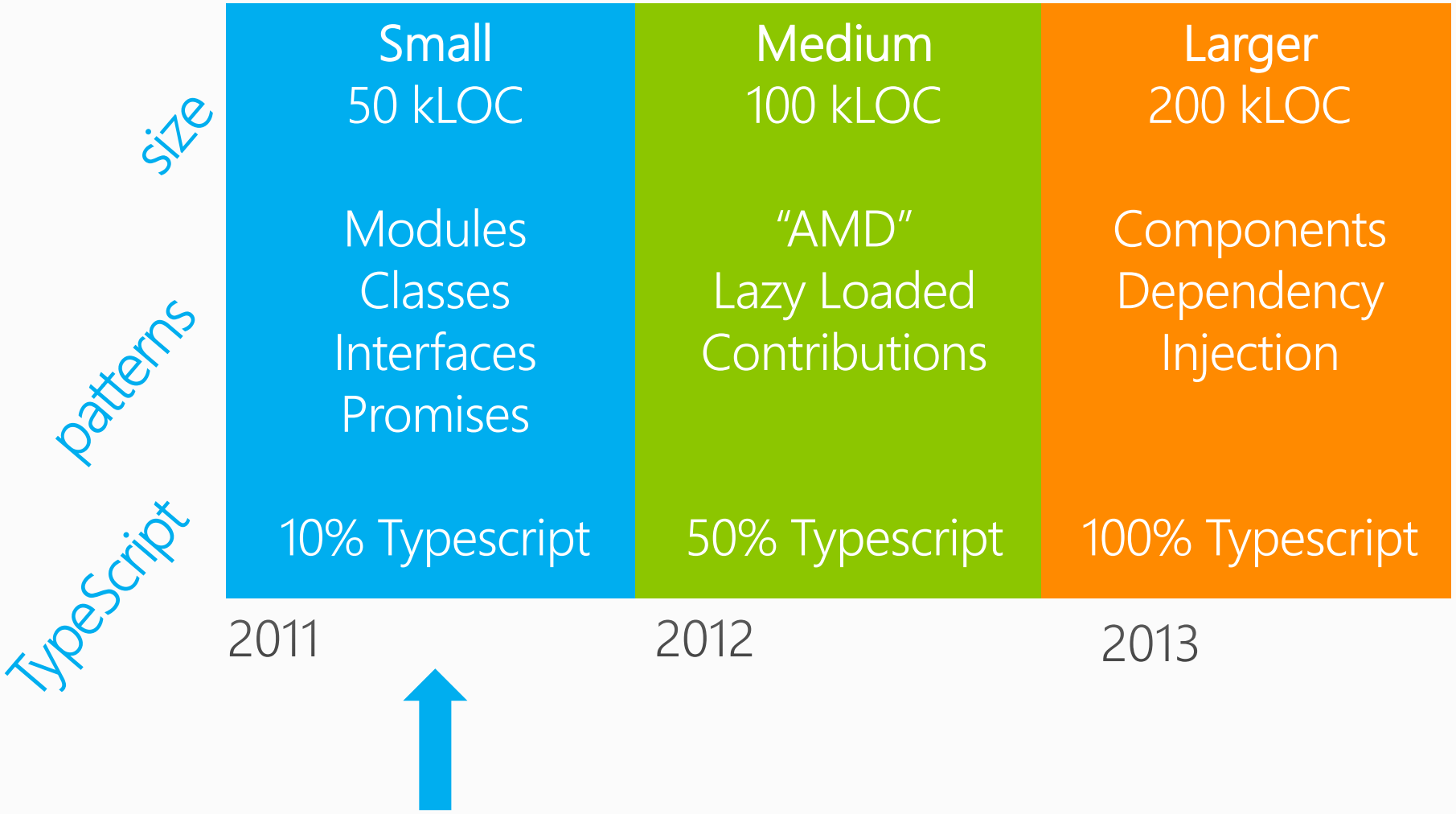
Napa Office 365



Azure Web Sites



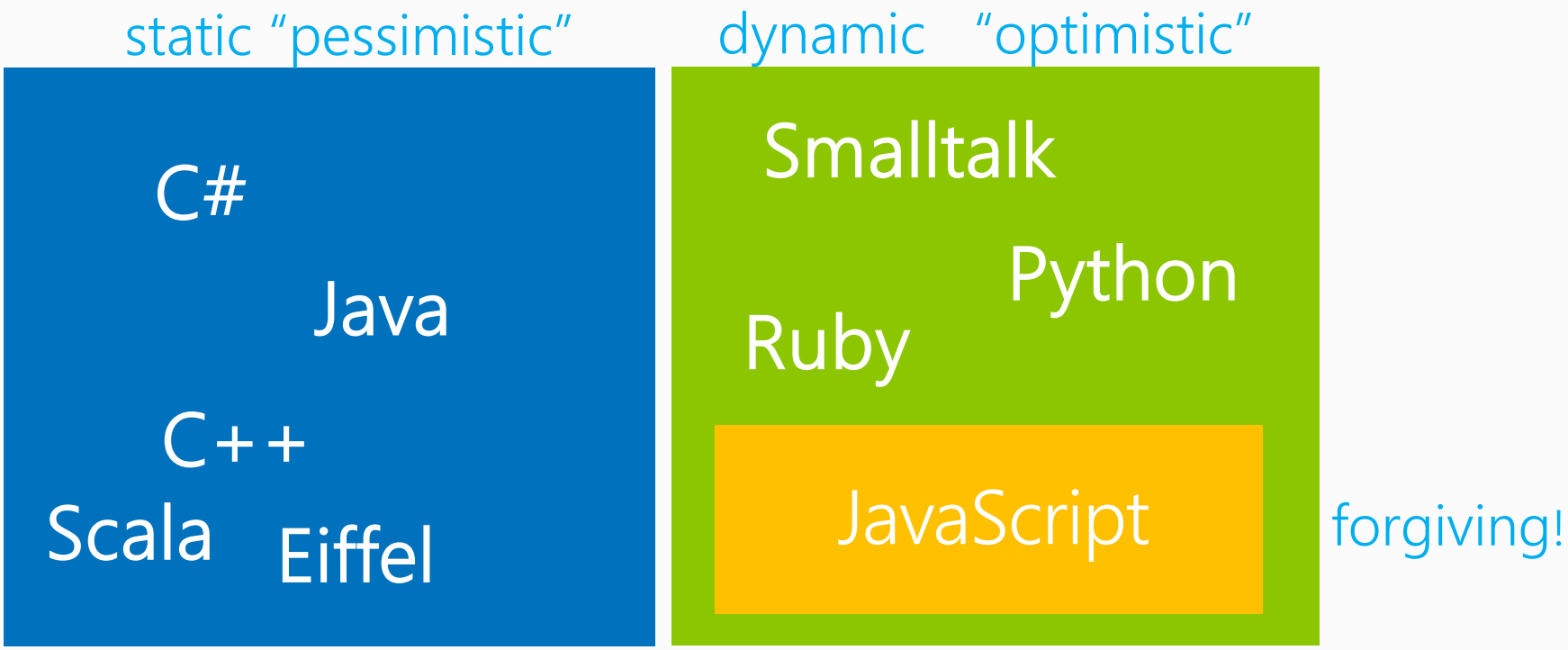
Our Journey (Part 1)



We enjoy programming in JS*

*But we had some concerns

Application Scale JavaScript?



Pains

Need to come up with "compensating" patterns for classes and modules/namespaces

Need patterns for managing callbacks

"JavaScript code 'rots' over time" --Ben

"Writing JavaScript code in a large project is like carving code in stone"
--Alex

TypeScript to rescue

TypeScript is a super set of JavaScript

Easy to convert existing code to TypeScript

Compiler / Library works cross browser

TypeScript compiles to plain JavaScript

TypeScript is open source

http://typescript.codeplex.com/

TypeScript Demo

Types, Interfaces, Classes, Generics, Modules

Writing better JavaScript

Formalization of common JavaScript patterns

Classes, modules

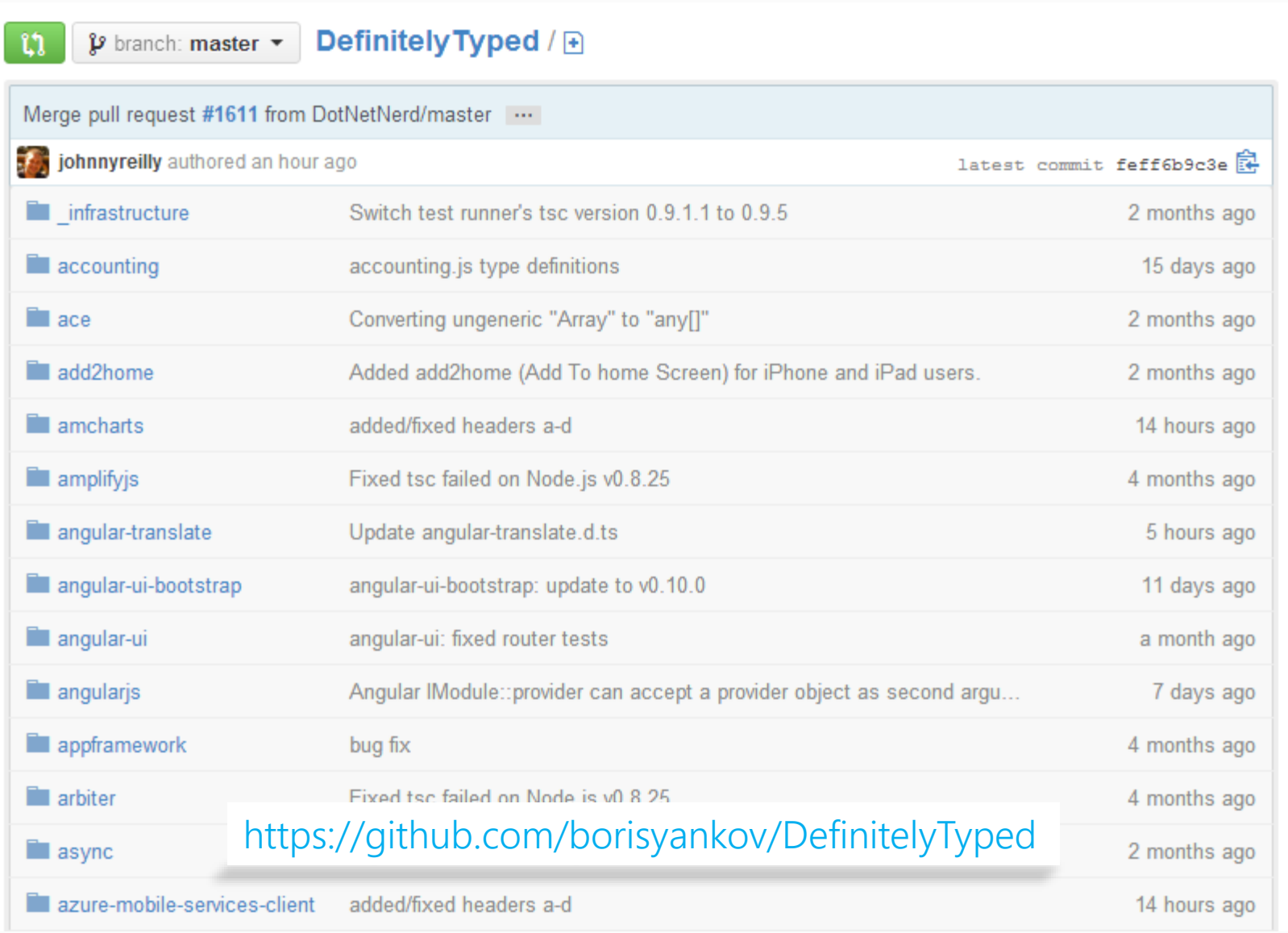
Type inference and structural typing

In practice very few type annotations are necessary

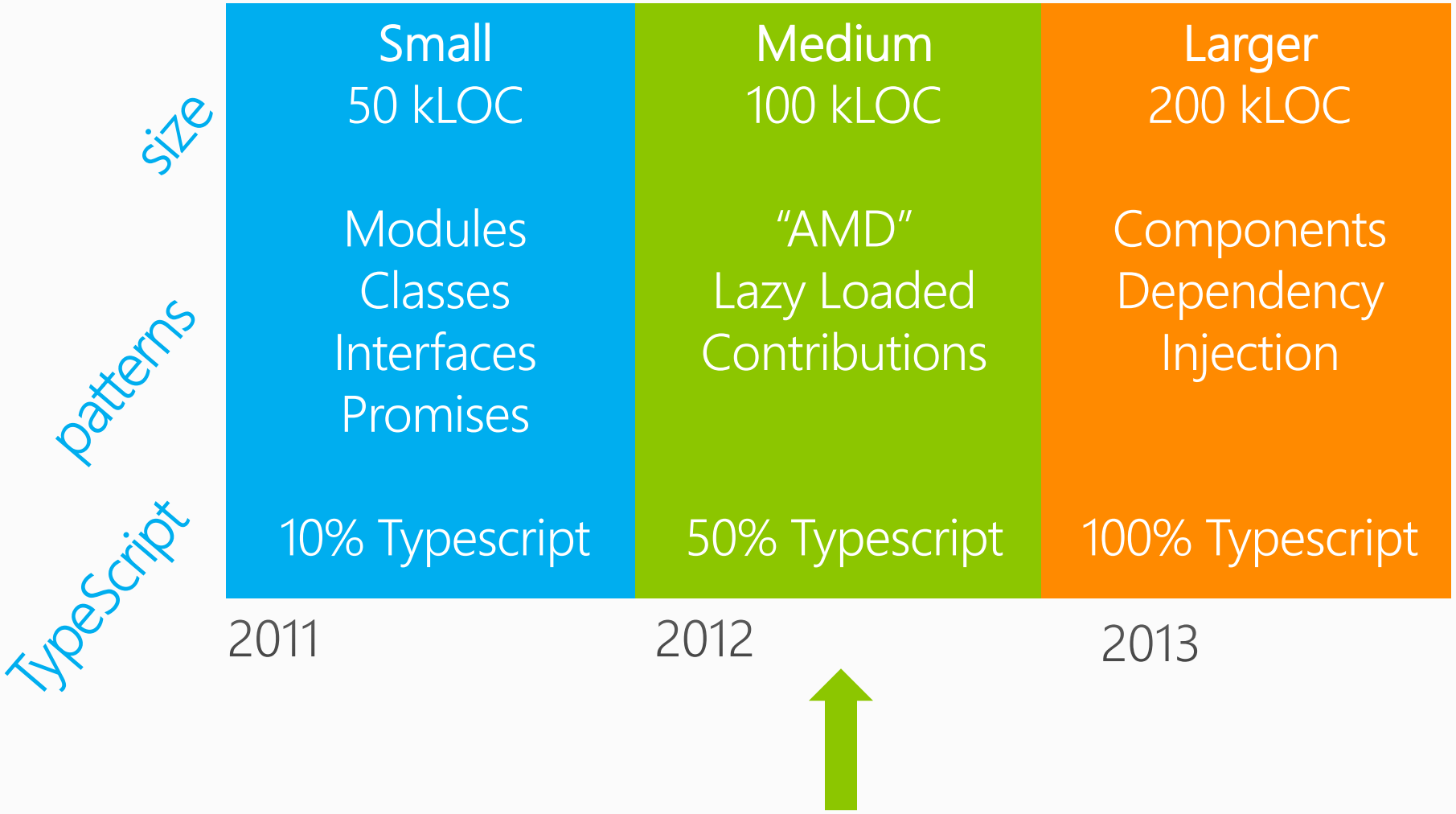
Works with existing JavaScript libraries

Declaration files can be written and maintained separately

TypeScript Type Definitions



Our Journey (Part 2)



Growing Pains

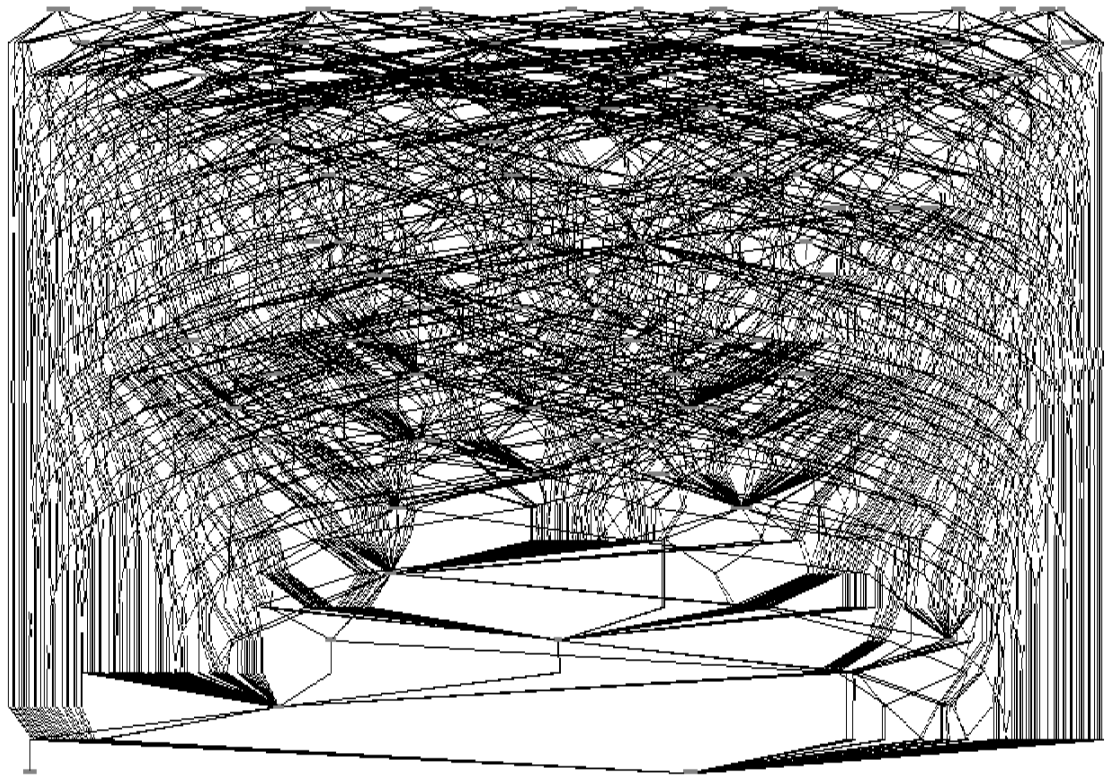
Managing scripts and their order

Manually edited lists of scripts

/ide/**manifest.json**

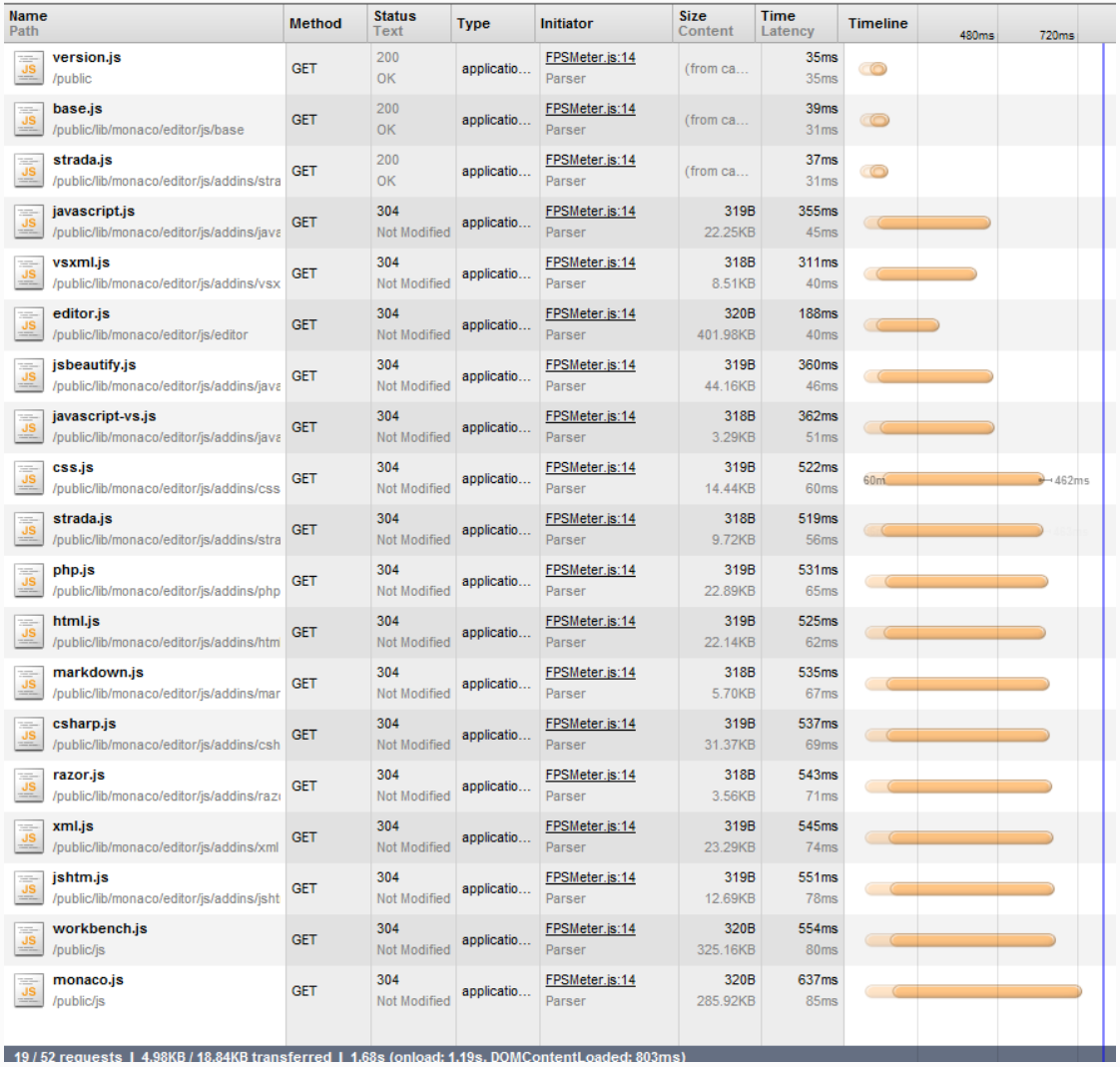
```
27
28      "src": {
29          "/monaco/ide/public/js/workbench.js": [
30              "public/js/constants.js",
31              "public/js/platform.js",
32              "public/js/events.js",
33              "public/js/core/context.js",
34              "public/js/core/storage.js",
35              "public/js/core/history.js",
36              "public/js/model/workspaceModel.js",
37              "public/js/ui/sitelet.js",
38              "public/js/ui/commands.js",
39              "public/js/ui/memento.js",
40              "public/js/ui/layout.js",
41              "public/js/ui/actions/action.js",
42              "public/js/ui/menu.js",
43              "public/js/ui/viewlets/viewlet.js",
44              "public/js/ui/parts/part.js",
45              "public/js/ui/parts/sidebarPart.js",
46              "public/js/ui/parts/editor/baseEditor.js",
47              "public/js/ui/parts/editor/editorModel.js",
48              "public/js/ui/parts/editor/editorInput.js",
49              "public/js/ui/parts/editor/editorOptions.js",
50              "public/js/ui/parts/editor/textEditor.js",
51              "public/js/ui/parts/editor/stringEditorModel.js",
52              "public/js/ui/parts/editor/stringEditorInput.js",
```

Growing Pains: Dependencies…



"our dependency graph was such a mess that each area had a dependency on just about every other area." – Nick (from another project)

Growing Pains: Eager loading



Name / Path	Method	Status / Text	Type	Initiator	Size / Content	Time / Latency	Timeline (480ms, 720ms)
version.js /public	GET	200 OK	applicatio…	FPSMeter.js:14 Parser	(from ca…	35ms 35ms	
base.js /public/lib/monaco/editor/js/base	GET	200 OK	applicatio…	FPSMeter.js:14 Parser	(from ca…	39ms 31ms	
strada.js /public/lib/monaco/editor/js/addins/stra	GET	200 OK	applicatio…	FPSMeter.js:14 Parser	(from ca…	37ms 31ms	
javascript.js /public/lib/monaco/editor/js/addins/java	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	319B 22.25KB	355ms 45ms	
vsxml.js /public/lib/monaco/editor/js/addins/vsx	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	318B 8.51KB	311ms 40ms	
editor.js /public/lib/monaco/editor/js/editor	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	320B 401.98KB	188ms 40ms	
jsbeautify.js /public/lib/monaco/editor/js/addins/java	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	319B 44.16KB	360ms 46ms	
javascript-vs.js /public/lib/monaco/editor/js/addins/java	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	318B 3.29KB	362ms 51ms	
css.js /public/lib/monaco/editor/js/addins/css	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	319B 14.44KB	522ms 60ms	60m… 462ms
strada.js /public/lib/monaco/editor/js/addins/stra	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	318B 9.72KB	519ms 56ms	
php.js /public/lib/monaco/editor/js/addins/php	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	319B 22.89KB	531ms 65ms	
html.js /public/lib/monaco/editor/js/addins/htm	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	319B 22.14KB	525ms 62ms	
markdown.js /public/lib/monaco/editor/js/addins/mar	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	318B 5.70KB	535ms 67ms	
csharp.js /public/lib/monaco/editor/js/addins/csh	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	319B 31.37KB	537ms 69ms	
razor.js /public/lib/monaco/editor/js/addins/raz	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	318B 3.56KB	543ms 71ms	
xml.js /public/lib/monaco/editor/js/addins/xml	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	319B 23.29KB	545ms 74ms	
jshtm.js /public/lib/monaco/editor/js/addins/jsht	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	319B 12.69KB	551ms 78ms	
workbench.js /public/js	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	320B 325.16KB	554ms 80ms	
monaco.js /public/js	GET	304 Not Modified	applicatio…	FPSMeter.js:14 Parser	320B 285.92KB	637ms 85ms	

19 / 52 requests | 4.98KB / 18.84KB transferred | 1.68s (onload: 1.19s, DOMContentLoaded: 803ms)

Solution: AMD

A module/file must declare what other modules it requires in order to function

```
define('id', ['moduleA'], function(moduleA) {
    // code goes here
});
```

TypeScript: External Modules

TypeScript supports code generation for **CommonJS** and the **AMD** module systems

```
import WinJS = require('vs/base/lib/winjs');
import ZoneWidget = require('vs/editor/zoneWidget');
```

```
tsc --module amd app.ts
```

```
tsc --module commonjs HttpServer.ts
```

Before/After

AMD in JavaScript

```
define([‘..../winjs.base‘, ‘..../zoneWidget’],
    function(WinJS, ZoneWidget) { ... }
);
```

TypeScript (AMD or CommonJS)

```
import WinJS= require('vs/base/lib/winjs');
import ZoneWidget = require('vs/editor/zoneWidget');
```

AMD Applied

Support à la carte consumption

Express CSS dependencies as well

CSS AMD loader plugin
TypeScript pragma

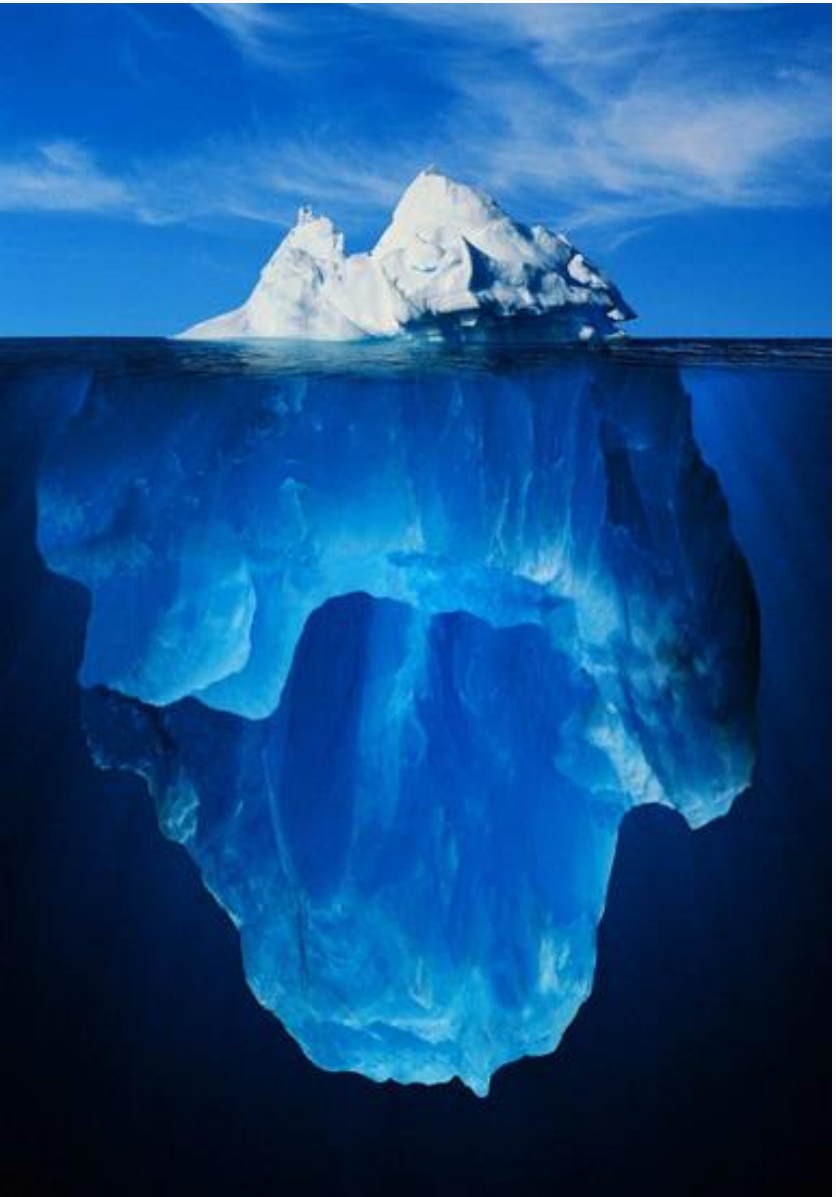
```
/// <amd-dependency path="vs/css!./actionbar" />
```

After the AMD Migration Impressions

"It feels like fresh showered. Self contained modules no more cycles no more globals with explicit dependencies" -Alex

"Happy - no more globals, no more cyclic dependencies, faster startup" -Dirk

Lazy Loading Contributions



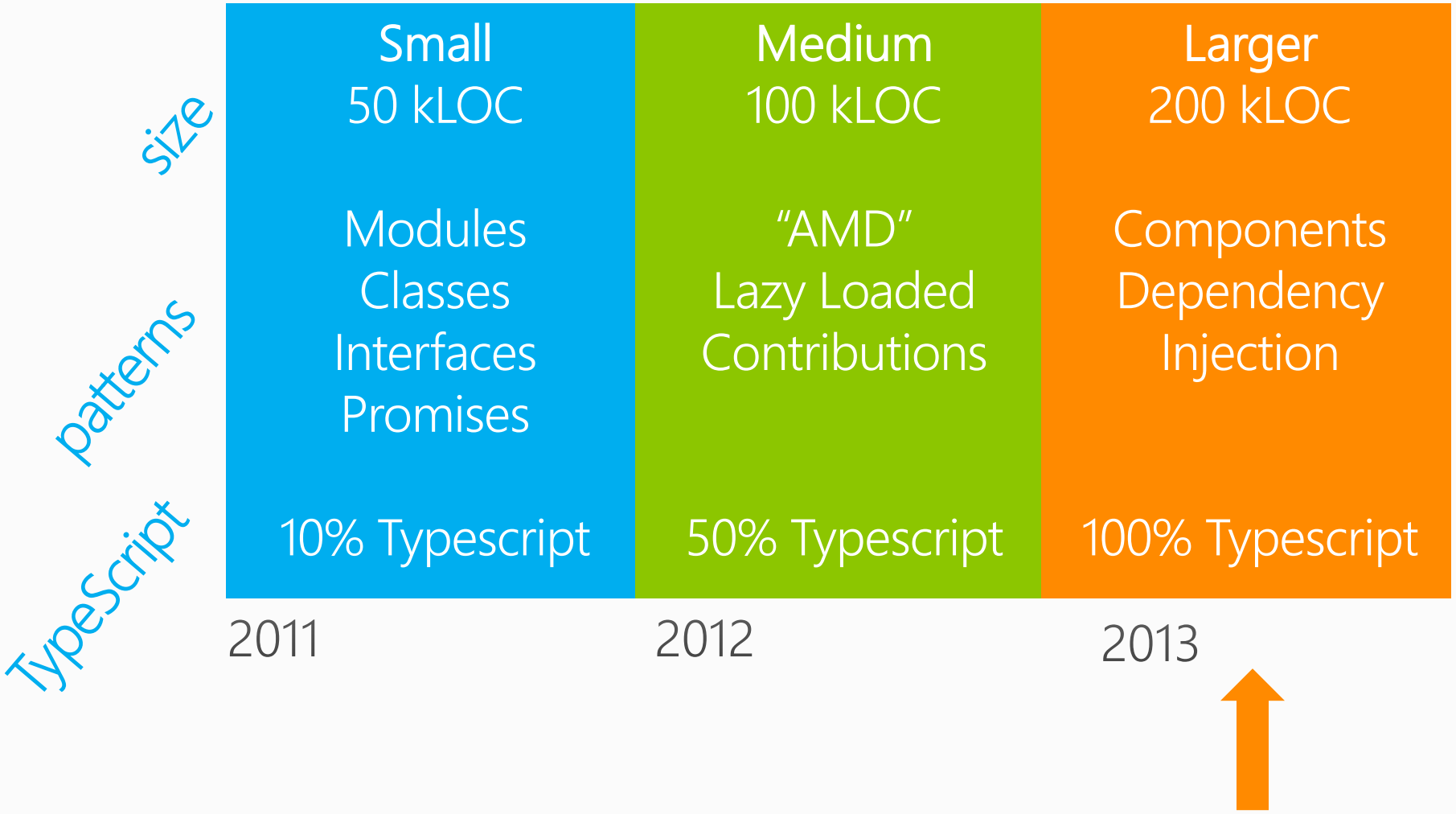
csharp.contribution.ts

```
modeRegistry.registerMode(
    [‘text/x-csharp'],
    new Platform.Descriptor(
        'vs/languages/csharp',
        ‘CSMode')
);
```

csharp.ts

```
export class CSMode extends
modesExtensions.AbstractMode {
    constructor() {
        super('vs.languages.csharp');
    }
   // lots of code ….
 }
```

Our Journey (Part 3)



Componentization

Reuse TypeScript code as 'binary' JS components with a declarations file

Example using TypeScript language services as a component

Compiler and IDE services > 30kLOC of TypeScript

```
tsc –declarations –out typescriptservices.js typescript.ts
```

API

Type the API surface

No inferred ‘any’ type in the API => “any hunting”

```
44      constructor(mode, buildType) {
type implicitly set to 'any'
```

JSDoc the API

```
/**
 * An direction aware selection.
 */
export interface IEditorSelection extends ISelection, IEditorRange {
    /**
     * Test if equals other selection.
     */
    equalsSelection(other:ISelection):bool;
    /**
     * Clone this selection.
     */
    clone():IEditorSelection;
    /**
     * Get directions (LTR or RTL).
     */
    getDirection(): SelectionDirection;
```

100% TypeScript

Migration is also code clean-up, but real work

Velocity around 300 LOCs per hour

"As I did conversions, I began typing various object literals I was passing around as interfaces. Soon enough, I realized how inconsistent I was, the same data was flowing around in at least 3 different formats. This is because of the easiness through which you can create literals in JavaScript …. Need some placeholder for data?... Just create a new literal object." --Alex

"In JavaScript, you are really at the mercy of your ability to spell. "

```
    delete this.markers[range.statMarkerId]; //start
```

--Joao

TS Retrospective

We were on the bleeding edge…

- …but we expected it and had plenty of band aid

We would do it again, the benefits outweigh the pains

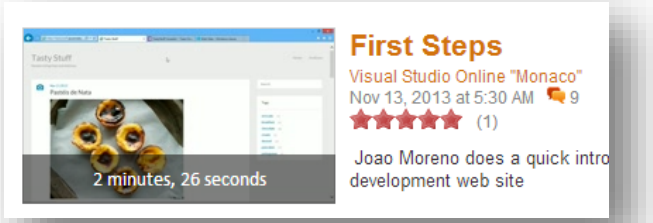
- Code readability, refactoring agility, tooling, fun

http://www.typescriptlang.org/

Monaco Links

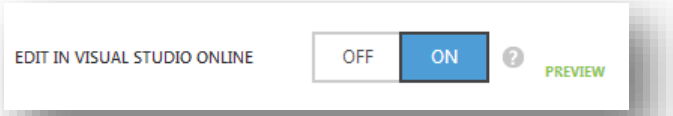
Learn

Get started with the Monaco Channel9 Series
Read Jgalloway's blog post on Monaco



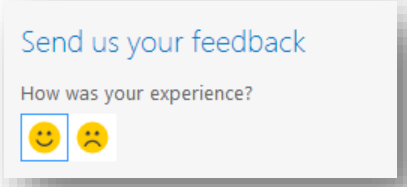
Try

Enable "Edit in Visual Studio Online" for your Azure Websites
Clone in code from Git repositories on Visual Studio Online
Develop, build, test and push to production



Feedback

Send a Smile!
#VSOMonaco



Thanks!