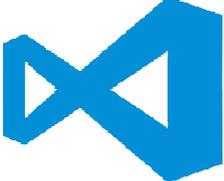


Visual Studio Code

Ein Open Source Code Editor in TypeScript und HTML 5

Dirk Bäumer
(@dbktw)





Visual Studio Code

A desktop tool that combines the **simplicity of a code editor** with what developers need for the core **code-build-debug-commit cycle**



↓ Windows

Windows 7, 8, 10



↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, CentOS



↓ OS X

OS X Yosemite, El Capitan



task.contribution.ts - vscode - Visual Studio Code

File Edit Selection View Go Debug Help

EXPLORER

- extensions
- feedback
- files
- html
- markers
- output
- performance
- preferences
- quickopen
- relauncher
- scm
- search
- snippets
- surveys
- tasks
 - browser
 - common
 - electron-browser
 - media
 - jsonSchema_v1.ts
 - jsonSchema_v2.ts
 - jsonSchemaCommon.ts
 - task.contribution.ts
 - terminalTaskSystem.ts
- node
- test
- terminal
- themes
- trust
- update
- views
- watermark
- welcome
- services
- test
 - buildfile.js
 - buildunit.json
 - css.d.ts
 - css.js
 - loader.js
 - monaco.d.ts
 - nls.d.ts
 - nls.js

```
884         let done = (value: TaskSet) => {
885             if (value) {
886                 result.push(value);
887             }
888             if (--counter === 0) {
889                 resolve(result);
890             }
891         };
892         let error = () => {
893             if (--counter === 0) {
894                 resolve(result);
895             }
896         };
897         if (this.getExecutionEngine() === ExecutionEngine.Terminal && this._providers.size > 0) {
898             this._providers.forEach((provider) => {
899                 counter++;
900                 provider.provideTasks().done(done, error);
901             });
902         } else {
903             resolve(result);
904         }
905     }).then((result) => {
906         return this.getWorkspaceTasks().then((workspaceTaskResult) => {
907             let workspaceTasksToDelete: Task[] = [];
908             let annotatingTasks = workspaceTaskResult.annotatingTasks;
909             let legacyAnnotatingTasks = workspaceTaskResult.set ? this.getLegacyAnnotatingTasks(workspaceTaskResult.set) : undefined;
910             if (annotatingTasks || legacyAnnotatingTasks) {
911                 for (let set of result) {
912                     for (let task of set.tasks) {
913                         if (annotatingTasks) {
914                             let annotatingTask = annotatingTasks.byIdentifier[task.identifier] || annotatingTasks.byName[task.name];
915                             if (annotatingTask) {
916                                 TaskConfig.mergeTasks(task, annotatingTask);
917                                 task._source.kind = TaskSourceKind.Workspace;
918                                 continue;
919                             }
920                         }
921                         if (legacyAnnotatingTasks) {
922                             let legacyAnnotatingTask = legacyAnnotatingTasks[task.identifier];
923                             if (legacyAnnotatingTask) {
924                                 TaskConfig.mergeTasks(task, legacyAnnotatingTask);
925                                 task._source.kind = TaskSourceKind.Workspace;
926                             }
927                         }
928                     }
929                 }
930             }
931             return result;
932         });
933     });
934 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: powershell.exe

```
Windows PowerShell
Copyright (c) 2016 Microsoft Corporation. All rights reserved.

Loading personal and system profiles took 6043ms.
P:\mseng\VSCode\vscode [master]>
```

Ln 1, Col 1 Tab Size: 4 UTF-8 LF TypeScript 2.3.2

Microsoft Edge

The screenshot displays the Microsoft Edge Developer Tools interface. The top bar shows the browser title "code.visualstudio.com - F12 Developer Tools - Microsoft Edge" and various tool tabs: F12, DOM Explorer, Console, Debugger, Network, Performance, Memory, Emulation, and Experiments. The address bar shows the URL "code.vis...udio.com" and "adblock...hrome.js".

The left sidebar contains the "Local Storage" and "Cookies" sections, listing various domains like "code.visualstudio.com", "cdnjs.cloudflare.com", "www.googleadservices.com", "www.google-analytics.com", "az416426.vo.msecnd.net", "www.googletagmanager.com", "c2076ic-live.com", "c.microsoft.com", "connect.facebook.net", "platform.twitter.com", "4056.xg4ken.com", "ghbtns.com", "www.google.ch", "(no domain)", "Dynamic scripts", and "EdgeExtension_BetaFishAdBlock".

The main area shows the "Debugger" tab with the following JavaScript code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <script>
5     var isProduction = window.location.hostname === 'code.visualstudio.com';
6     var isHomePage = window.location.pathname === '/';
7     var isNodejsLanding = window.location.pathname === '/nodejs';
8     var isDocsFirstStart = window.location.href.toLowerCase().split('.')[1] === '/docs?start=true';
9
10    function downloadExperiment(experimentId) {
11
12      // Copied code from Google analytics
13      // except for experimentId variable
14      function utmx_section(){}function utmx(){}(function(){var
15        k=experimentId,d=document,l=d.location,c=d.cookie;
16        if(!.search.indexOf('utm_expId='+k)>0)return;
17        function f(n){if(c){var i=c.indexOf(n+'=');if(i>-1){var j=c.
18          indexOf(';');return escape(c.substr(i+n.length+1,j<0?c.
19            length:j)}}}var x=f('__utmxx'),xx=f('__utmxx'),h=l.hash;d.write(
20          '<script src="'+http'+l.protocol+'https://ssl:
21            '//www'+'.google-analytics.com/ga_exp.js?'+utmkey='+k+
22            '&utmxx='+x?x:''+'&utmxx='+xx?xx:''+'&utmtime='+new Date().
23            valueOf()+h?&utmhash'+escape(h.substr(1)):'')+
24            '" type="text/javascript" charset="utf-8"></script>');
25
26    }
27
28    // Remember to create if statements in lower script as well.
29    if (isProduction) {
30      // Don't remove. This is the example to follow
31      // if (isHomePage) {
32        // downloadExperiment('101350005-35');
33      // }
34      if (isDocsFirstStart) {
35        // downloadExperiment('101350005-29');
36      }
37      if (isNodejsLanding) {
38        downloadExperiment('101350005-36');
39      }
40    }
41  }
42 </script>
```

The bottom status bar shows "Current window: code.visualstudio.com" and "Target: _top: code.visualstudio.com".



tasks - OneDrive task.contribution.ts - Or

onedrive.live.com/?v=TextFileEditor&id=B66B7E86BEC6C0C%21395406&cid=0B66B7E86BEC6C0C&partid=B66B7E86BEC6C0C%21395403

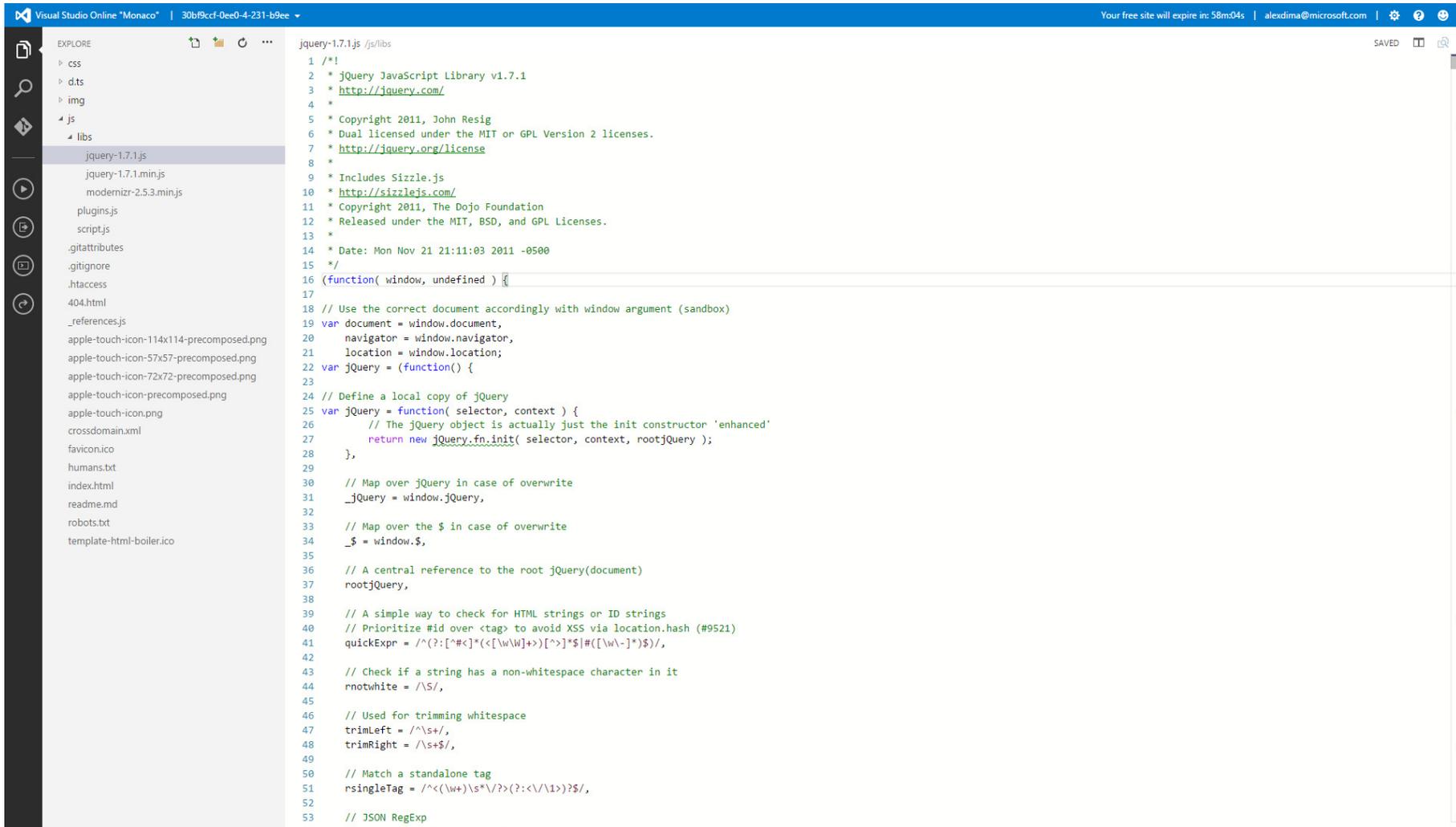
OneDrive Dirk Baeumer

Share Download Delete Rename Embed

Files > tasks > task.contribution

```
297     element.title = nls.localize('problems', "Problems");
298
299     Dom.addClass(error, 'task-statusbar-item-label-error');
300     error.innerHTML = '0';
301     label.appendChild(error);
302
303     Dom.addClass(warning, 'task-statusbar-item-label-warning');
304     warning.innerHTML = '0';
305     label.appendChild(warning);
306
307     Dom.addClass(info, 'task-statusbar-item-label-info');
308     label.appendChild(info);
309     $(info).hide();
310
311     callOnDispose.push(Dom.addDisposableListener(label, 'click', (e: MouseEvent) => {
312         const panel = this.panelService.getActivePanel();
313         if (panel && panel.getId() === Constants.MARKERS_PANEL_ID) {
314             this.partService.setPanelHidden(true);
315         } else {
316             this.panelService.openPanel(Constants.MARKERS_PANEL_ID, true);
317         }
318     }));
319
320     let updateStatus = (element: HTMLDivElement, stats: number): boolean => {
321         if (stats > 0) {
322             element.innerHTML = stats.toString();
323             $(element).show();
324             return true;
325         } else {
326             $(element).hide();
327             return false;
328         }
329     };
330
331
332     let manyMarkers = nls.localize('manyMarkers', "99+");
333     let updateLabel = (stats: MarkerStatistics) => {
334         error.innerHTML = stats.errors < 100 ? stats.errors.toString() : manyMarkers;
335         warning.innerHTML = stats.warnings < 100 ? stats.warnings.toString() : manyMarkers;
336         updateStatus(info, stats.infos);
337     };
338
339     this.markerService.onMarkerChanged((changedResources) => {
340         updateLabel(this.markerService.getStatistics());
341     });
```

Visual Studio Online "Monaco"



The screenshot displays the Visual Studio Online Monaco editor interface. The top status bar shows the project name "Visual Studio Online 'Monaco'", a unique identifier "30b9ccf-0ee0-4-231-b9ee", and a notification "Your free site will expire in: 58m:04s | alexdima@microsoft.com". The left sidebar contains an "EXPLORE" view with a file tree for a project named "jquery-1.7.1.js /js/libs". The main editor area displays the source code for "jquery-1.7.1.js /js/libs". The code is a JavaScript file defining the jQuery library, including comments about licensing and version information.

```
jquery-1.7.1.js /js/libs
1  /*!
2  * jQuery JavaScript Library v1.7.1
3  * http://jquery.com/
4  *
5  * Copyright 2011, John Resig
6  * Dual licensed under the MIT or GPL Version 2 licenses.
7  * http://jquery.org/license
8  *
9  * Includes Sizzle.js
10 * http://sizzlejs.com/
11 * Copyright 2011, The Dojo Foundation
12 * Released under the MIT, BSD, and GPL Licenses.
13 *
14 * Date: Mon Nov 21 21:11:03 2011 -0500
15 */
16 (function( window, undefined ) {
17
18 // Use the correct document accordingly with window argument (sandbox)
19 var document = window.document,
20     navigator = window.navigator,
21     location = window.location;
22 var jQuery = (function() {
23
24 // Define a local copy of jQuery
25 var jQuery = function( selector, context ) {
26     // The jQuery object is actually just the init constructor 'enhanced'
27     return new jQuery.fn.init( selector, context, rootjQuery );
28 },
29
30 // Map over jQuery in case of overwrite
31 _jQuery = window.jQuery,
32
33 // Map over the $ in case of overwrite
34 _$ = window.$,
35
36 // A central reference to the root jQuery(document)
37 rootjQuery,
38
39 // A simple way to check for HTML strings or ID strings
40 // Prioritize #id over <tag> to avoid XSS via location.hash (#9521)
41 quickExpr = /^(?:[#<]*([\w\W]+>)[>]*$|#[\w\-\_]*)$/,
42
43 // Check if a string has a non-whitespace character in it
44 rnotwhite = /\S/,
45
46 // Used for trimming whitespace
47 trimLeft = /^\s+/,
48 trimRight = /\s+$/,
49
50 // Match a standalone tag
51 rsingleTag = /^<(\w+)\s*/>(?:</\1>)?$/,
52
53 // JSON RegExp
54 ...
```

Inside Visual Studio Code

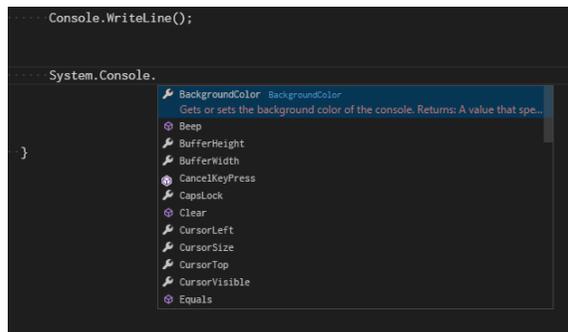
- Electron



- TypeScript



- Monaco Editor





ELECTRON

Build cross platform desktop apps with web technologies

Formerly known as Atom Shell. Made with ❤️ by GitHub.



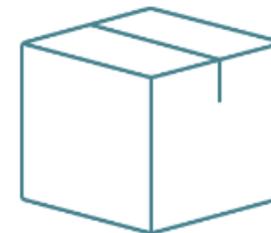
Web Tech

Use HTML, CSS, and JavaScript with Chromium and Node.js to build your app.



Open Source

Electron is open source; maintained by GitHub and an active community.

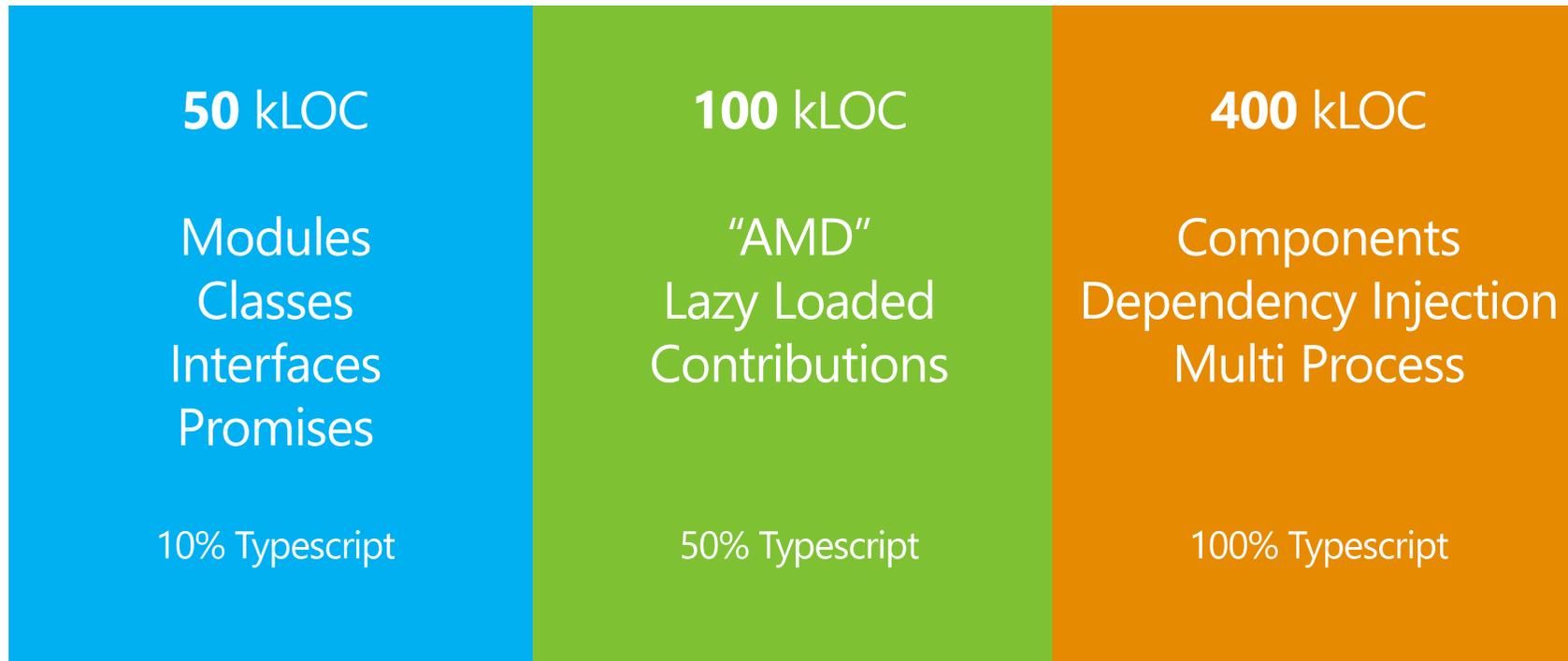


Cross Platform

Electron apps build and run on Mac, Windows, and Linux.



Growing the Code



autumn
2011

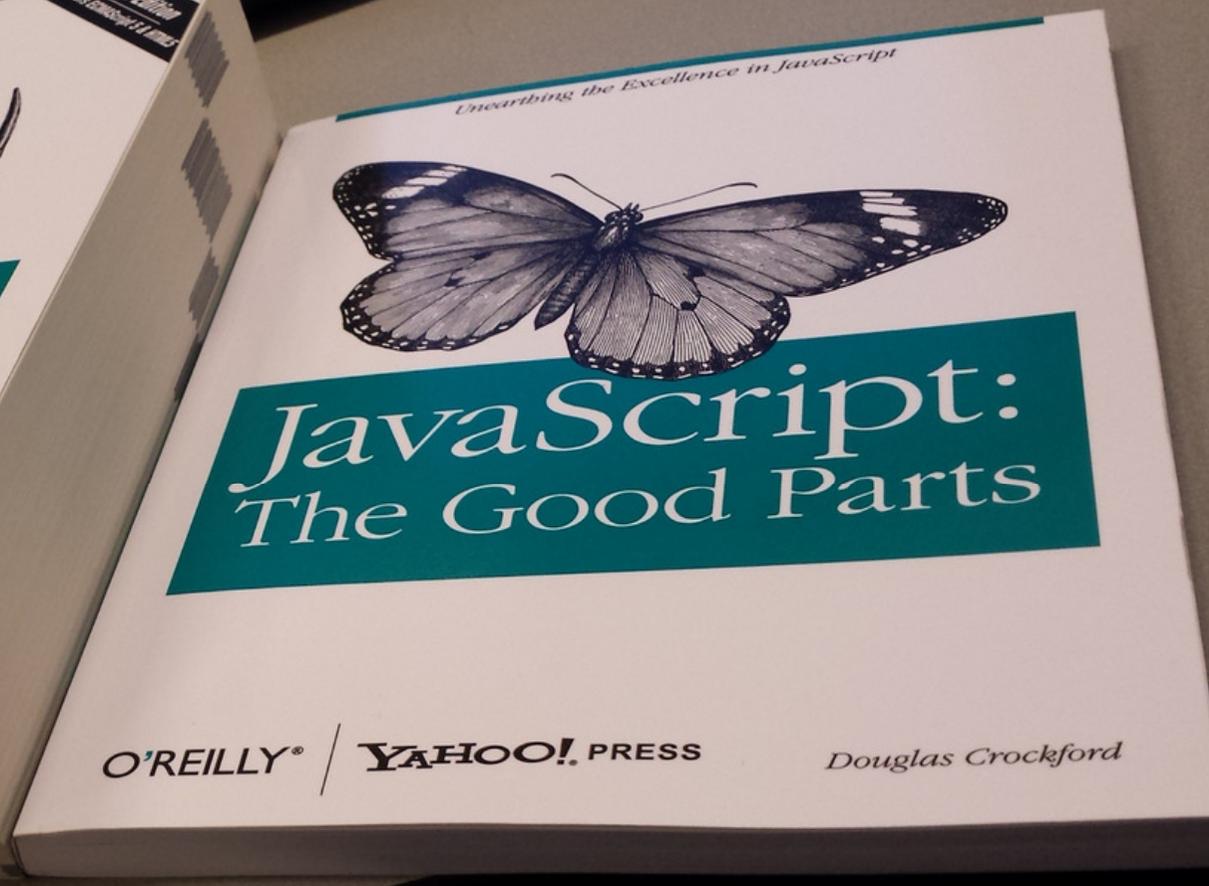
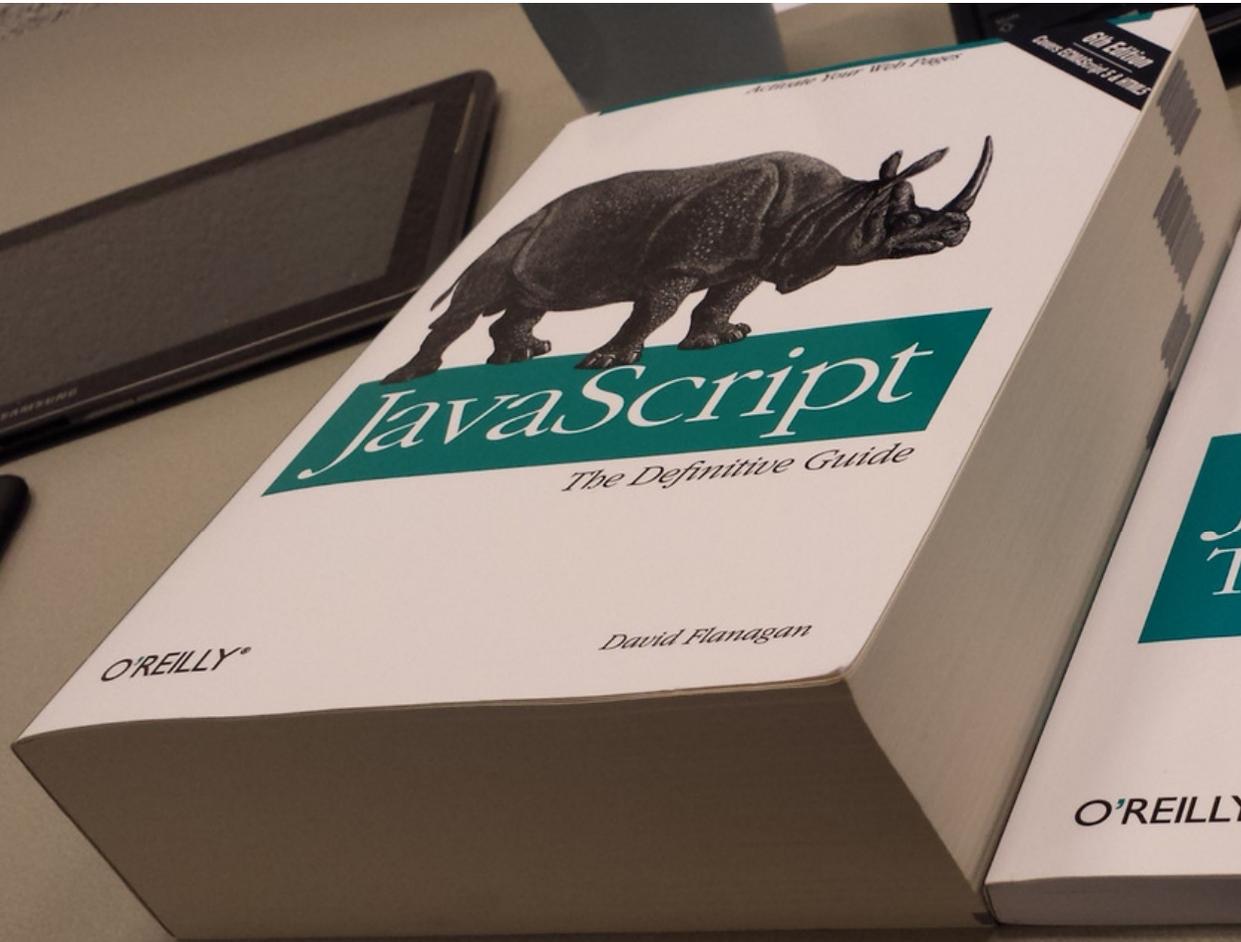
today
~800 LOC

Growing the Code

50 kLOC

Modules
Classes
Interfaces
Promises

10% Typescript



Large scale JavaScript
development is hard.

<https://www.destroyallsoftware.com/talks/wat>

JavaScript Pains

Programming JavaScript is like carving code in stone. Refactoring it is difficult. JavaScript code rots over time.

Comprehending JavaScript code is difficult as well (hard to browse and navigate)

Large code bases need to come up with compensating patterns for classes (es6), modules (amd or commonjs) and namespaces.

Describing APIs means keeping documentation synchronized with the implementation.

Starts with JavaScript

All JavaScript code is TypeScript code, simply copy and paste
All JavaScript libraries work with TypeScript

Optional static types, classes, modules

Enable scalable application development and excellent tooling
Zero cost: Static types completely disappear at run

TypeScript to the Rescue...

Structural typing, type inference, optional types
Zero run time cost, types disappear at run time

```
1 reference
1 interface Point {
2   0 references
3   - x: number;
4   0 references
5   - y: number;
6 }
7
8
9 let p = { x: 100, y: 100, description: "..."};
10 let point: Point = p;
```

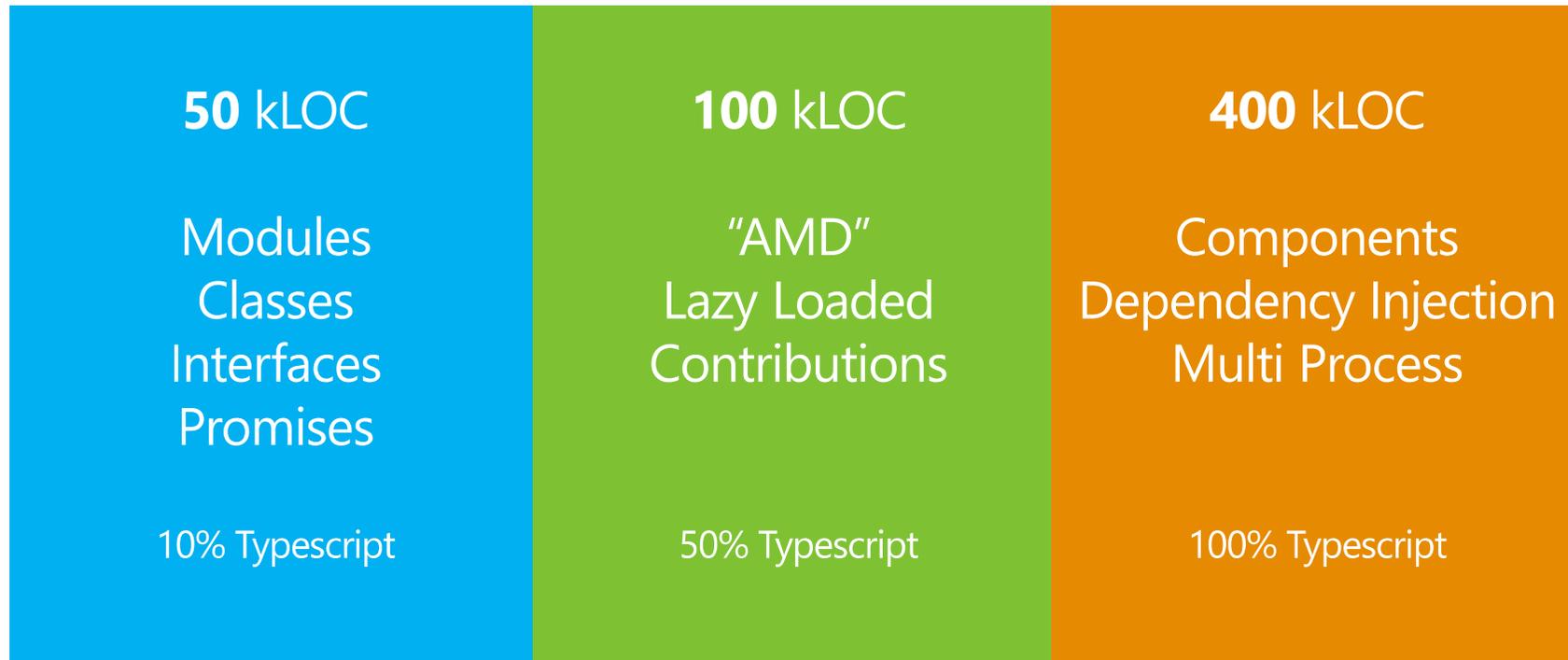
```
9 let person = {
10   cyclist: 'Johannes',
11   bikes: [
12     'Santa Cruz Nomad CC',
13     'Intense Carbine 29'
14   ]
15 };
16
17 let bikes = person.bikes;
18 let rider = person.rider;
(1/1) [ts] Property 'rider' does not exist on type '{ cyclist: string; bikes: string[]; }'.
```

Makes ES6 features available in ES3/5 runtimes

- ✓ Arrows
- ✓ Classes
- ✓ Modules (import / export)
- ✓ Let + Const
- ✓ For..Of
- ✓ Destructuring
- ✓ Template Strings
- ✓ Default + Rest

See also: <https://github.com/lukehoban/es6features>

Growing the Code



autumn
2011

today

Growing the Code

100 kLOC

“AMD”
Lazy Loaded
Contributions

50% Typescript

Code Organization: Keep code structured

```
1 var Namespace = {};  
2 Namespace.Util = {};  
3 Namespace.Util.Strings = {};  
4 Namespace.Util.Strings.trim = function() {  
5     /* etc */  
6 }
```

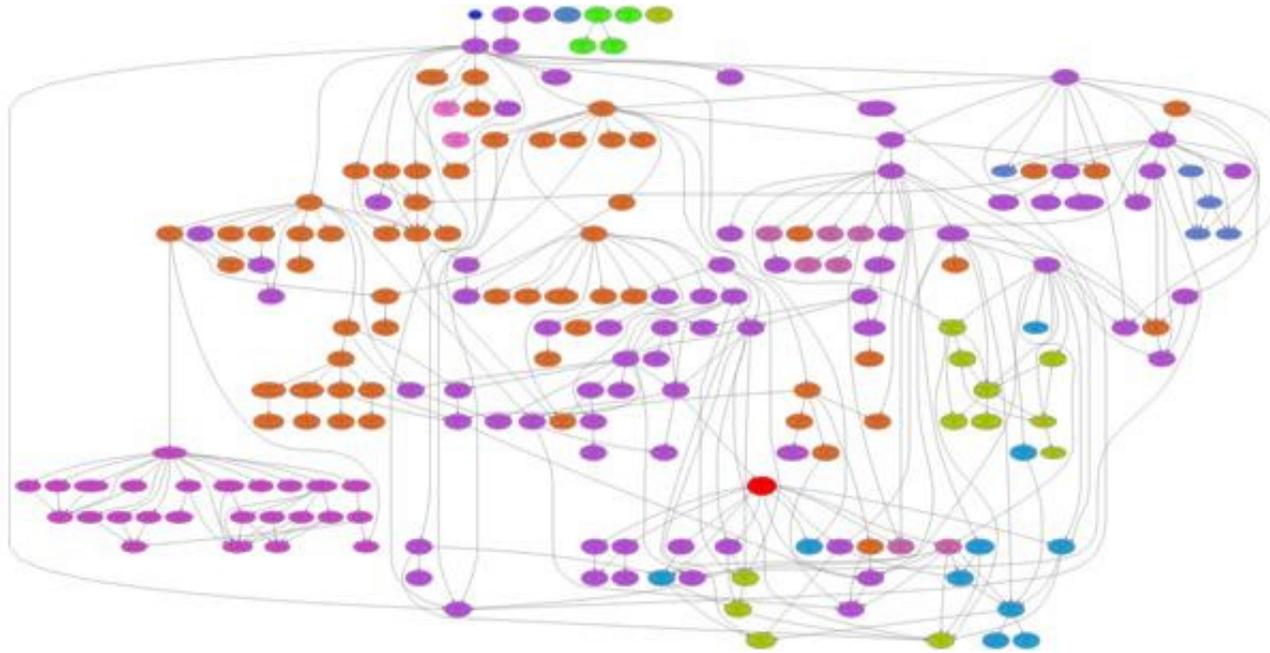
Global namespaces are a pain

No relationship to the source files on disk

Renaming things was a pain

Cycling dependencies were easily unnoticed

Code Organization: Dependencies



Our dependency graph was such a mess that each area had a dependency on just about every other area

Module Systems to the Rescue...

AMD

```
1 define('module_id', [  
2     'dependency_id'  
3 ], function(dependency) {  
4     // code  
5  
6     return {  
7         // exports  
8     };  
9 });
```

CommonJS

```
my_module.js  
1 var dependency = require('dependency_id');  
2  
3 // code  
4  
5 exports.myExports = {  
6     // exports  
7 }
```

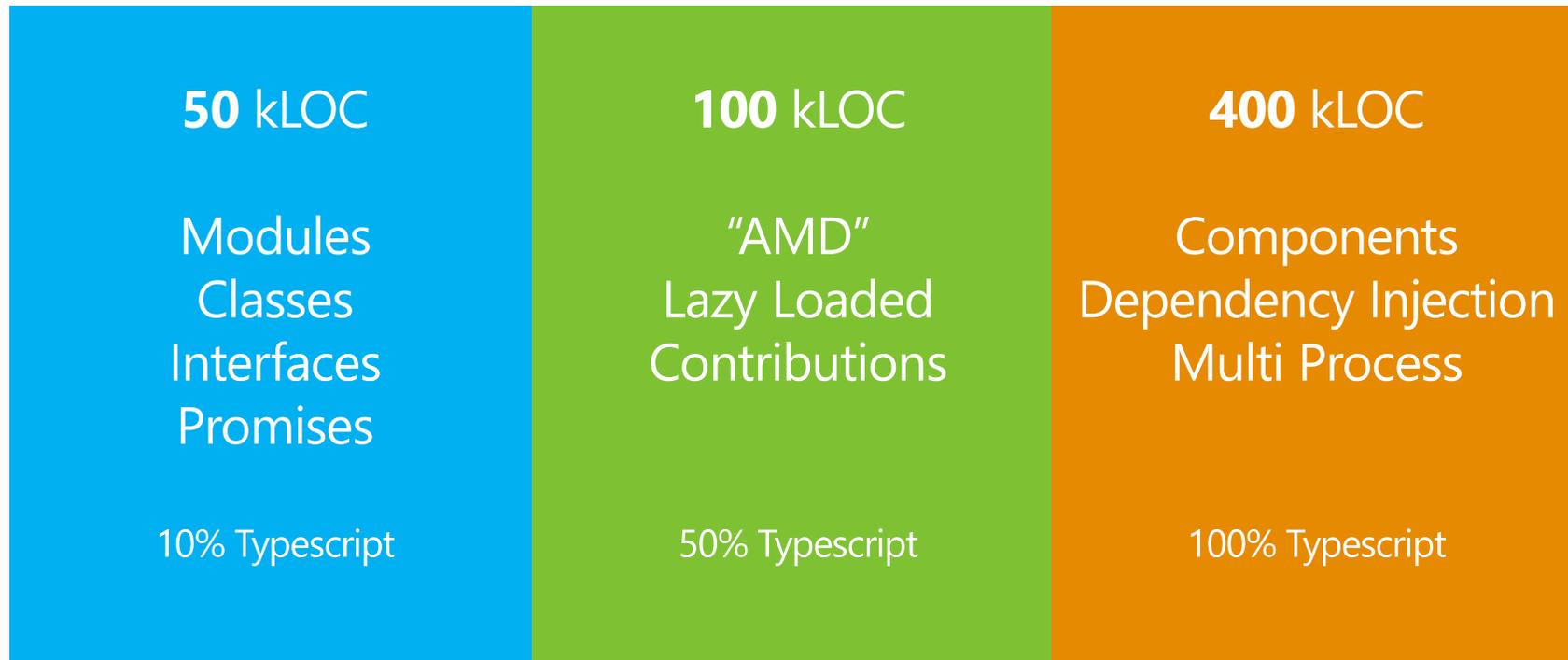
TypeScript: First class module support

Supports both AMD and CommonJS with one syntax

```
1 import { Dependency } from "./dependency_id";
2
3 // code
4
5 export function helper() {
6     // export
7 }
```

Sharing code between AMD and CommonJS is easy

Growing the Code



autumn
2011

today

Growing the Code

400 kLOC

Components
Dependency Injection
Multi Process

100% Typescript

today

100% TypeScript

Incremental conversion from JavaScript

Establish & enforce rules: no implicit any

Conversion is also code clean up and fixing

```
2 delete this.markers[range.statMarkerId];  
3 // or is it startMarkerId?
```

Dependency Injection

Constructor service injection in many places

```
1773 export class ShowActiveFileInExplorer extends Action {
1774
1775     constructor(
1776         id: string,
1777         label: string,
1778         @IWorkbenchEditorService private editorService: IWorkbenchEditorService,
1779         @IViewletService private viewletService: IViewletService,
1780         @IWorkspaceContextService private contextService: IWorkspaceContextService,
1781         @IMessageService private messageService: IMessageService
1782     ) {
1783         super(id, label);
1784     }
```

Extension Isolation

Controlled
extensibility

JavaScript or
TypeScript

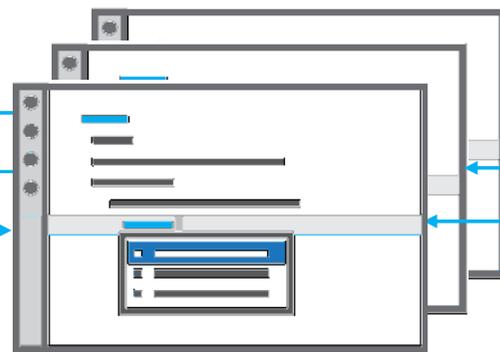
Reuse node modules

API: **vscode.d.ts**

Extension description:
package.json

Main
Process

IPC



Renderer process

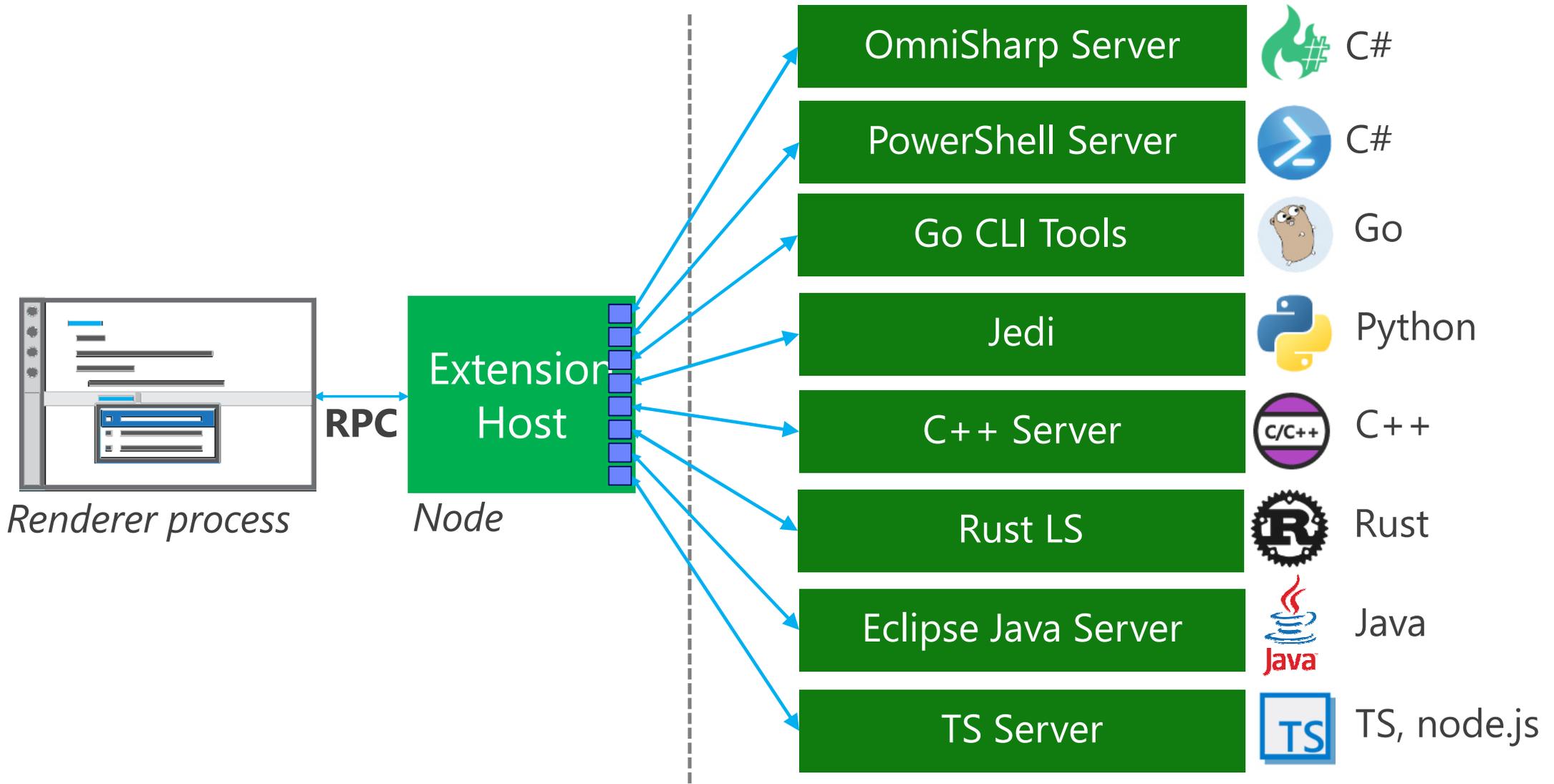
RPC

Extension
Host

node.js



Language Extensions: Language Servers



Thanks!

@code

<http://code.visualstudio.com>

<https://github.com/Microsoft/vscode/>

<http://www.typescriptlang.org/>

DIGICOMP

