



# Desktop UIs & Web Tech

Roman Liutikov  
@roman01la





👍 Ease of development 👍  
👍 Cross-platform 👍



👎 Executable size 👎  
👎 Memory & CPU consumption 👎  
👎 Low performance budget 👎

# Electron



=

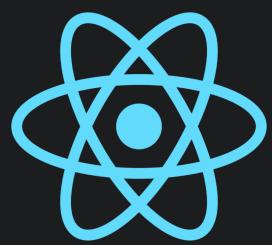
# Cordova





=

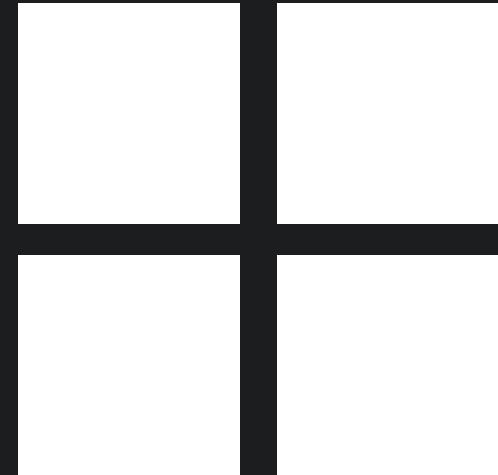




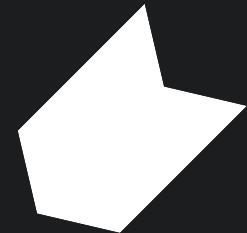
**React  
Native**



**Platform UI**



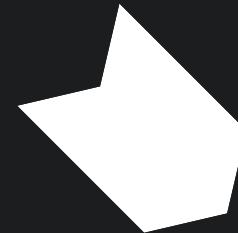
Platform UI



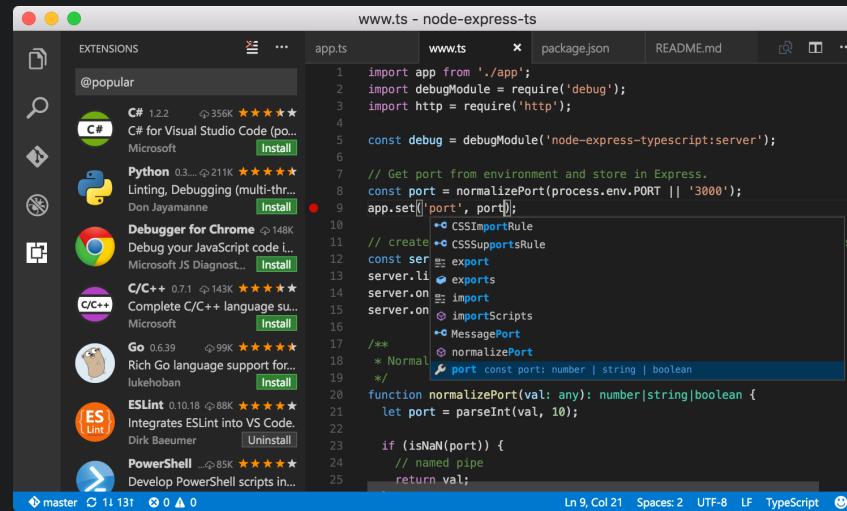
iOS



Android



WPF



The screenshot shows the Visual Studio Code interface. On the left, the Extensions sidebar is open, displaying popular extensions like C# (1.2.2), Python (0.3.0), Debugger for Chrome (0.7.1), Go (0.6.39), ESLint (0.10.18), and PowerShell (0.8.5). The main code editor window shows a file named www.ts containing TypeScript code for a node-express application. A tooltip is visible over the word 'normalizePort' in the code.

```
1 import app from './app';
2 import debugModule = require('debug');
3 import http = require('http');
4
5 const debug = debugModule('node-express-typescript:server');
6
7 // Get port from environment and store in Express.
8 const port = normalizePort(process.env.PORT || '3000');
9 app.set('port', port);
10
11 // create CSSImportRule
12 const server = export
13 server.li exports
14 server.on import
15 server.on importScripts
16
17 /**
18 * Normalize port
19 */
20 function normalizePort(val: any): number|string|boolean {
21   let port = parseInt(val, 10);
22
23   if (isNaN(port)) {
24     // named pipe
25     return val;
26   }
27
28   if (port < 0) {
29     throw new Error(`Port ${port} is less than zero`);
30   }
31
32   if (port >= 65535) {
33     throw new Error(`Port ${port} is greater than the maximum allowed value of 65535`);
34   }
35
36   return port;
37 }
```

## Cross-Platform UI

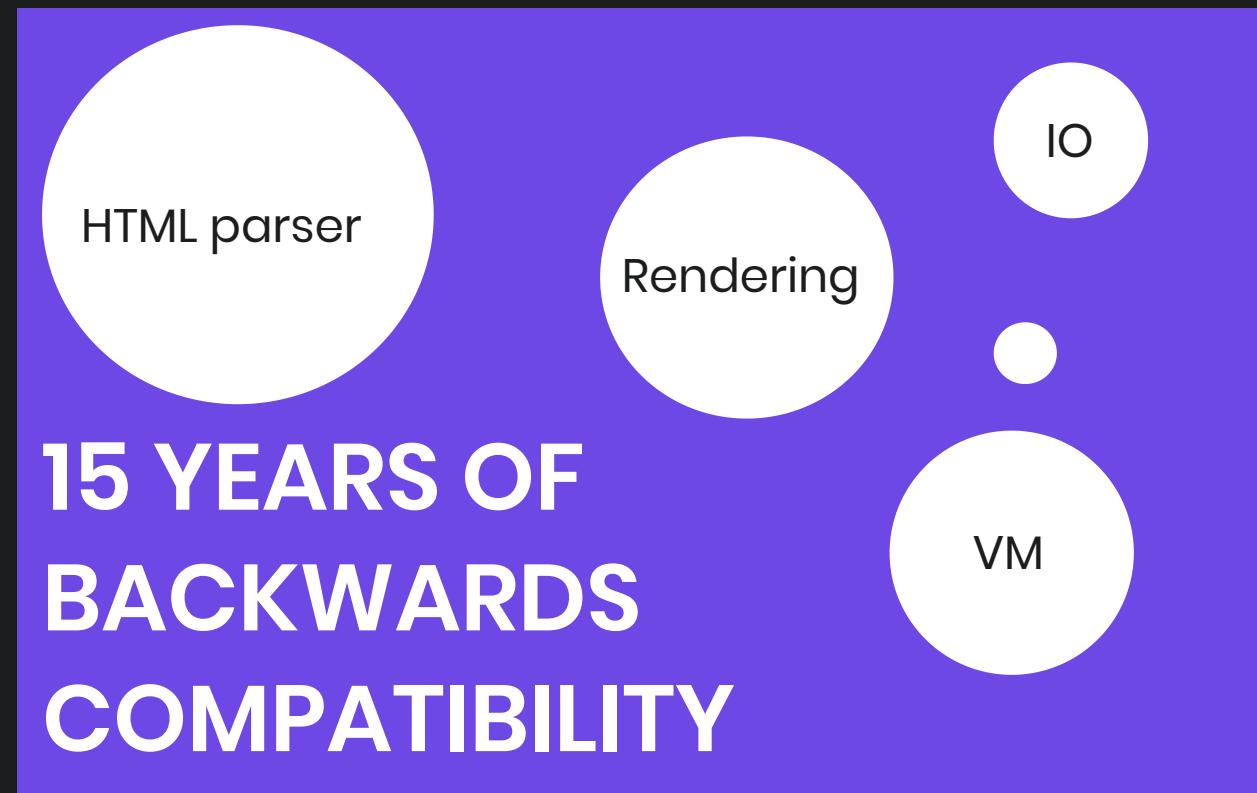
macOS

Windows

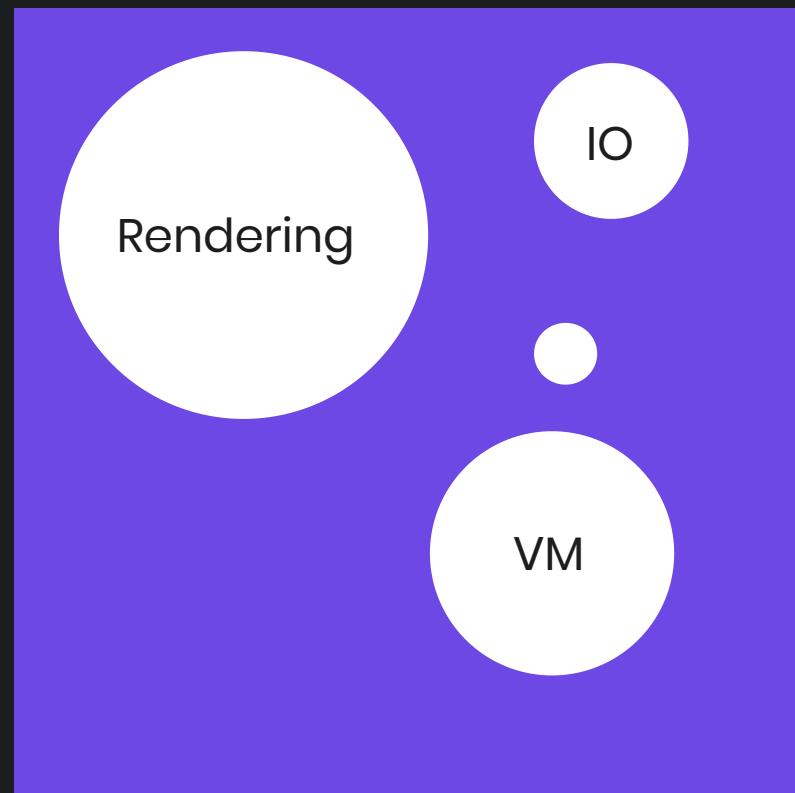
Linux

# Modern Browser

## Uncontrolled env

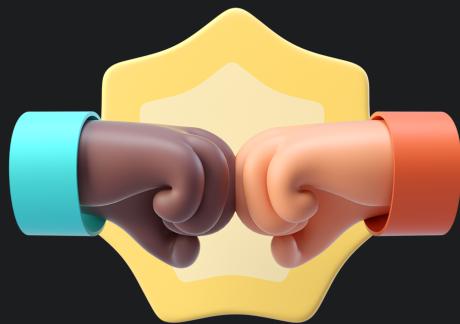


# Desktop Controlled env

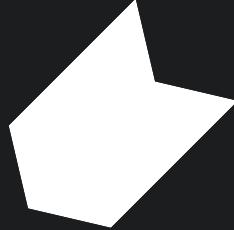


# Rendering

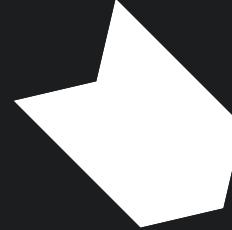
## CPU vs GPU



# C++/Rust/oCaml/...



Native



WASM

**Performance**



**Ease of development**







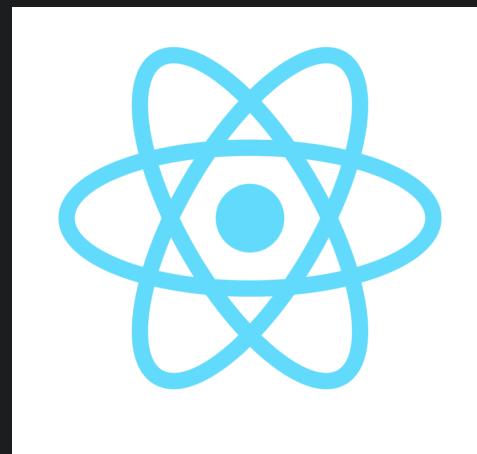
Dart



**JS**



**SK*IA***



**Desktop UI  
primitives**

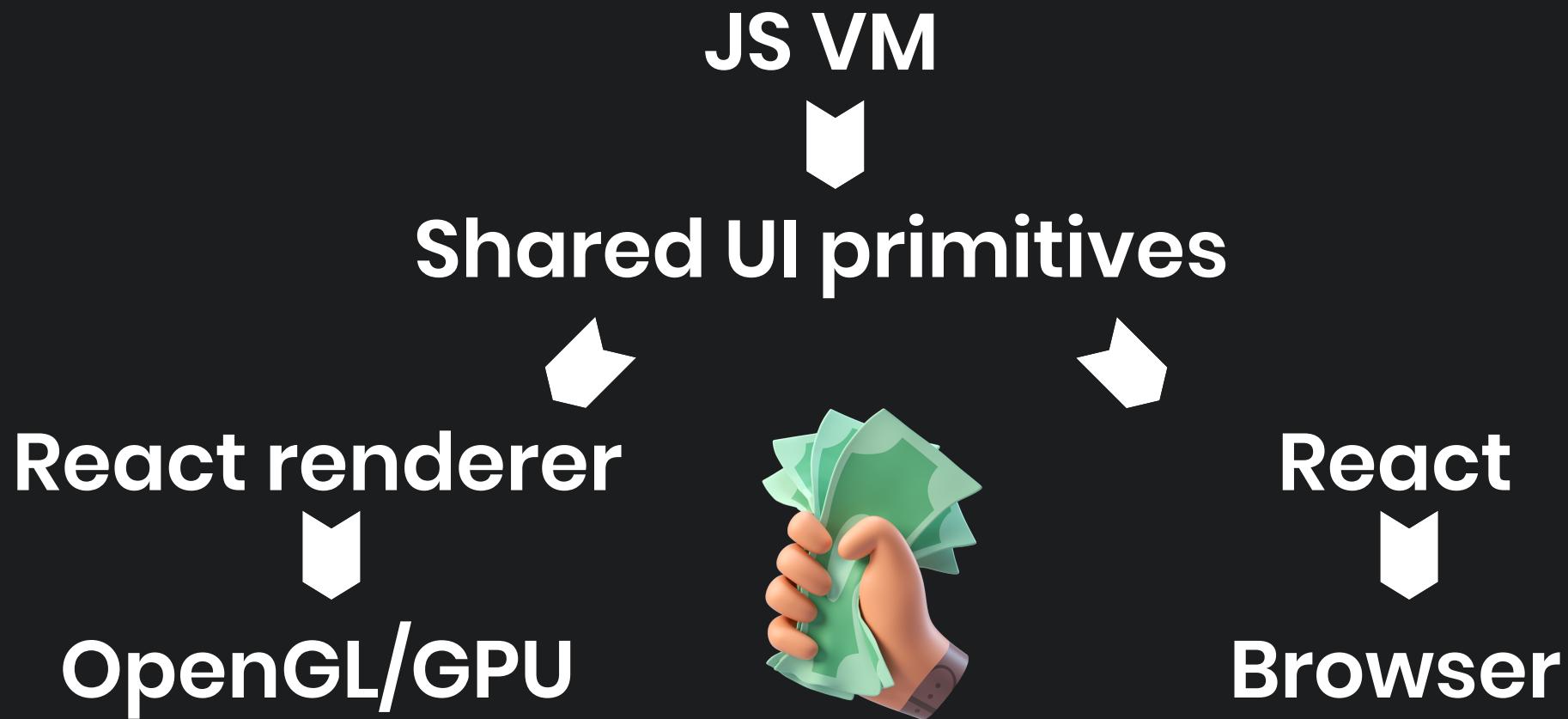


**Desktop**

**Web**

React Native  
for  
Web Desktop  
?

# “Ideal” stack





✓ Rendering engine  
+  
? Layout engine  
+  
? Accessibility  
+  
? Platform APIs Bindings



@roman01la