

Rafał Legiędź

Living on the Edge

Survive and Thrive with JavaScript and .NET running in-process

rafal.legiedz@gmail.com

[@rafek](#)

ABB



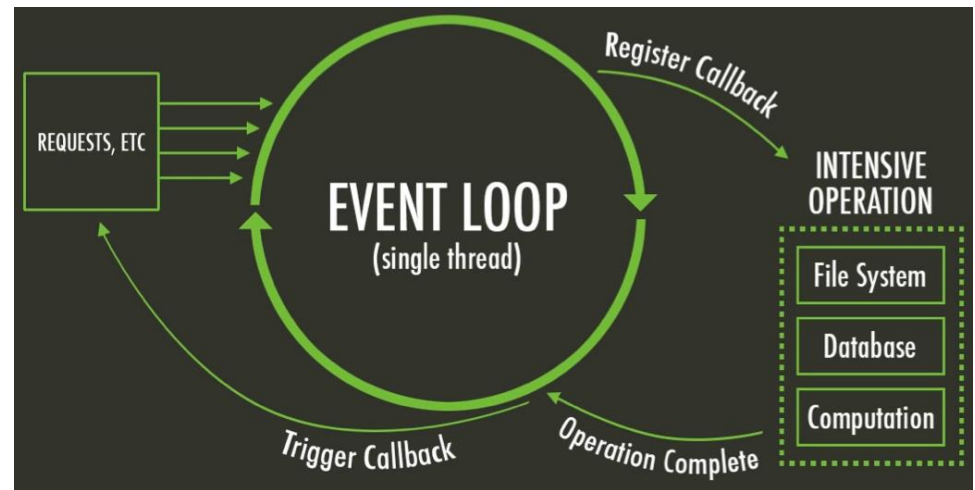
About me



Rafał Legiędź
rafal.legiedz@gmail.com
@rafek

Node.js

Platform built on Chrome's V8 JavaScript engine that provides an event-driven architecture and a non-blocking I/O API that optimizes an application's throughput and scalability.



Node.js

„[...] the only developer community that was built from day one around asynchronous, event-driven application logic as a core principle.”

~ Node.js is taking over the Enterprise

Node.js use cases



.NET

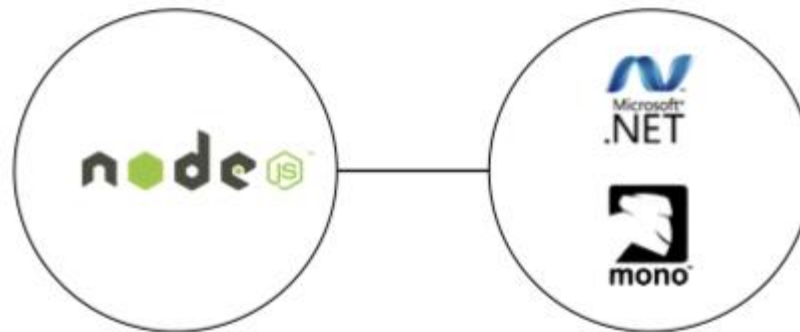
Advantages over Node.js:



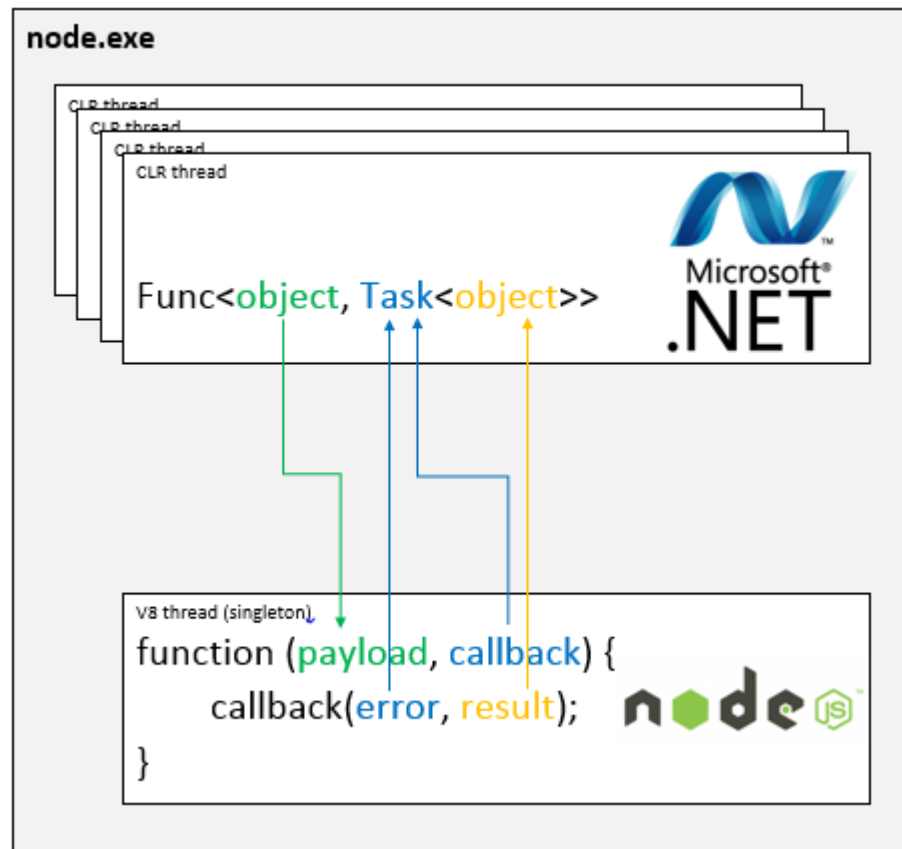
Edge.js

„[...] provides an asynchronous, in-process mechanism for interoperability between Node.js and .NET”

...on Windows, Linux and Mac OS X



Edge.js – interop model



Edge.js – possibilities

Enables or simplifies application scenarios which are hard or impossible to achieve with Node.js alone:

- implementing CPU-bound workloads in-process
- simplifying access to Windows specific functionality
- extending Node.js in .NET instead of native C

in any CLR language using inline code, separate files or assemblies.

What problem does it solve?



Tomasz Janczuk @tjanczuk · Apr 30

.NET vs Node.js? It is an AND not an OR. Now on Linux, Mac, and Windows.
bit.ly/12eLeVg #dotnet #nodejs #mono

← Reply ↻ Retweet ★ Favorite ⋮ More



Tessa @tesseractiv · Apr 30

@tjanczuk @shanselman Very cool, but dumb question: what problem does this solve? :-)

← Reply ↻ Retweet ★ Favorite ⋮ More



Scott Hanselman ✓

@shanselman

+ Follow

@tesseractiv @tjanczuk ah, whatever problem you have. If you have this problem, this solves it.

← Reply ↻ Retweet ★ Favorite ⋮ More



Edge.js – code

```
1 var edge = require('edge');
2
3 var hello = edge.func(function () { /*
4     async (input) => {
5         return "CSharp welcomes" + input.ToString();
6     }
7 */});
8
9 hello('Node.js', function (error, result) {
10     if (error) throw error;
11     console.log(result);
12 });
```

Installing Edge.js

- \geq Node.js 0.8.x
- .NET Framework 4.5 or Mono 3.4.0

```
C:\dev\sandbox
λ npm install edge
npm http GET http://registry.npmjs.org/edge
npm http 304 http://registry.npmjs.org/edge
npm http GET http://registry.npmjs.org/edge-cs/0.2.7
npm http 304 http://registry.npmjs.org/edge-cs/0.2.7

> edge@0.9.3 install C:\dev\sandbox\node_modules\edge
> node tools/install.js

Success: platform check for edge.js: node.js ia32 v0.10.26
edge@0.9.3 node_modules\edge
└─ edge-cs@0.2.7
```

.NET that scripts Node.js in-process

```
1  using EdgeJs;
2
3  async void Start() {
4      var createHttpServer = Edge.Func(@"
5          var http = require('http');
6
7          return function (port, cb) {
8              http.createServer(function (req, res) {
9                  res.end('Hello, world! ' + new Date());
10             }).listen(port, cb);
11         }
12     ");
13
14     await createHttpServer(8080);
15 }
```

Tips

- External class/assembly over inline code
- Separate modules
- Stay async
- Benchmark
- Node.js Tools for Visual Studio (Windows only)

Alternatives

- Stick with Node.js if possible
- Process invocation (Node.js's process api)
- Webservice (local/remote)
- *MQ

Resources

- [Edge.js presentation](#)
- [Source code](#)
- [This presentation](#)
- [The Birth & Death of JavaScript \(Gary Bernhardt\)](#)
- [Node.js is taking over the Enterprise](#)
- [Why Node.js is becoming the go-to technology in the enterprise](#)