

# Node.js Reloaded!

Julián Duque

A detailed image of the droid R2-D2 from Star Wars, positioned on the right side of the slide. The droid is white with blue accents and is standing on a grey surface. The background is a solid dark grey.

# Julián Duque

Developer Evangelist / Community Organizer / Engineer  
@julian\_duque



NODESOURCE

July 17, 2019

# What is Node.js?

# What is Node.js?

Node.js is an open source application runtime, similar to Oracle's Java or Microsoft's .NET, for building highly scalable and concurrent network and web applications.

Node.js uses JavaScript as its programming language thus unlocking a massive pool of engineers to write backend applications: web developers.



Node.js is a simple platform for developing

... network-centric

... modular

... JavaScript applications

... using asynchronous, event-driven  
programming



# A Simple Platform

- Node.js' core consists mostly of essentials: small-core
- The Python lesson: “A standard library is where code goes to die”
- npm and Node's modularity enables a vibrant ecosystem
- Open source competition begets quality and innovation

JavaScript

NODE.JS CORE API

NODE.JS BINDINGS

C / C++

V8  
JAVASCRIPT  
ENGINE

LIBUV

OPENSSL  
ZLIB  
HTTP\_PARSER  
CARES

## Networking and I/O

- More than ¼ of Node.js core is dedicated to networking
- Designed for I/O-driven, high-throughput workloads
- Used to power the modern web



# Modular

- npm: the world's largest open source package registry
- The Node.js module system solves most “dependency-hell” problems
- Encourages anti-monolith development
- The holy-grail of decoupled and reusable coding

## “archy”

- 700th most downloaded Node.js package
- Fits on a single slide, very focused concern
- Easily tested, grokred, documented, shared

This modular pattern is The Node.js Way

```
module.exports = function archy (obj, prefix, opts) {
  if (prefix === undefined) prefix = '';
  if (!opts) opts = {};
  var chr = function (s) {
    var chars = {
      '┌' : '┌',
      '┐' : '┐',
      '└' : '└',
      '┘' : '┘',
      '─' : '─',
      '│' : '│'
    };
    return opts.unicode === false ? chars[s] : s;
  };

  if (typeof obj === 'string') obj = { label : obj };

  var nodes = obj.nodes || [];
  var lines = (obj.label || '').split('\n');
  var splitter = '\n' + prefix + (nodes.length ? chr(' | ') : ' ') + ' ';

  return prefix
    + lines.join(splitter) + '\n'
    + nodes.map(function (node, ix) {
      var last = ix === nodes.length - 1;
      var more = node.nodes && node.nodes.length;
      var prefix_ = prefix + (last ? '' : chr(' | ')) + ' ';

      return prefix
        + (last ? chr('┌') : chr('└')) + chr('─')
        + (more ? chr('─') : chr('─')) + ' '
        + archy(node, prefix_, opts).slice(prefix.length + 2)
      ;
    }).join('')
  ;
};
```

# JavaScript on the Server

- Effectively bridge the server/browser divide
- JavaScript for approachability, productivity and developer joy
- JavaScript is not going anywhere!

# Asynchronous, Event-driven Programming

- Alternative programmer paradigm: Continuation Passing Style
- API focus on callbacks, events and streams
- Perfect for I/O-driven, high-throughput workloads
- Single-threaded encourages scalable design patterns


```
// callbacks  
fs.readFile('data.txt', 'utf8', countLines)
```

```
function countLines (err, data) {  
  let lines = data.split('\n').length  
  console.log(`${lines} lines`)  
}
```

```
// events  
server.on('connection', onConnection)
```

```
function onConnection (stream) {  
  console.log('someone connected!')  
}
```

Continuation passing  
“Do this when ready”



Thread doesn't “block” while waiting on I/O

Node.js is great for:

- Resource orchestration applications
- Network & I/O-driven applications—the web!

Node.js is not so great for:

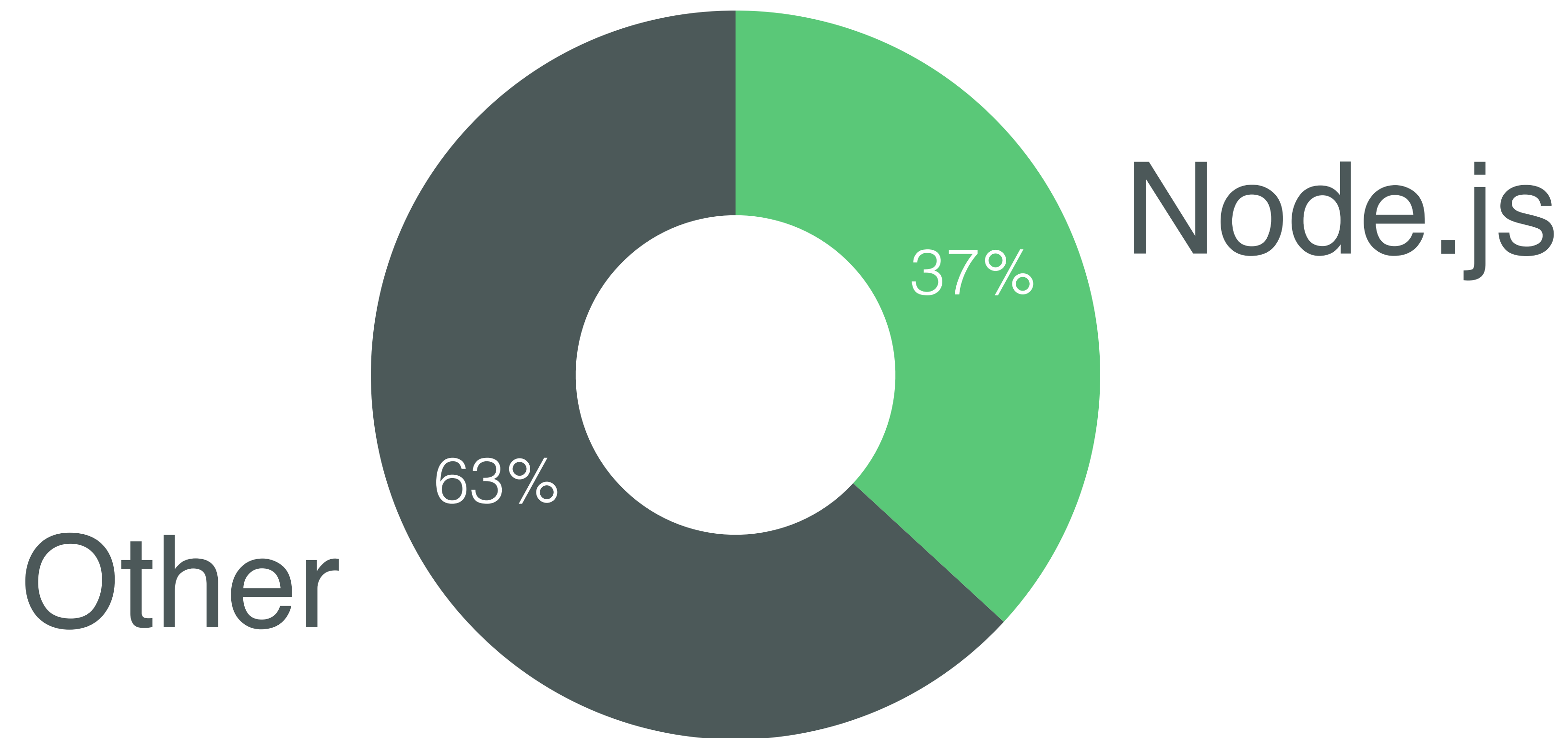
- CPU intensive workloads
- Systems programming



# Market Size

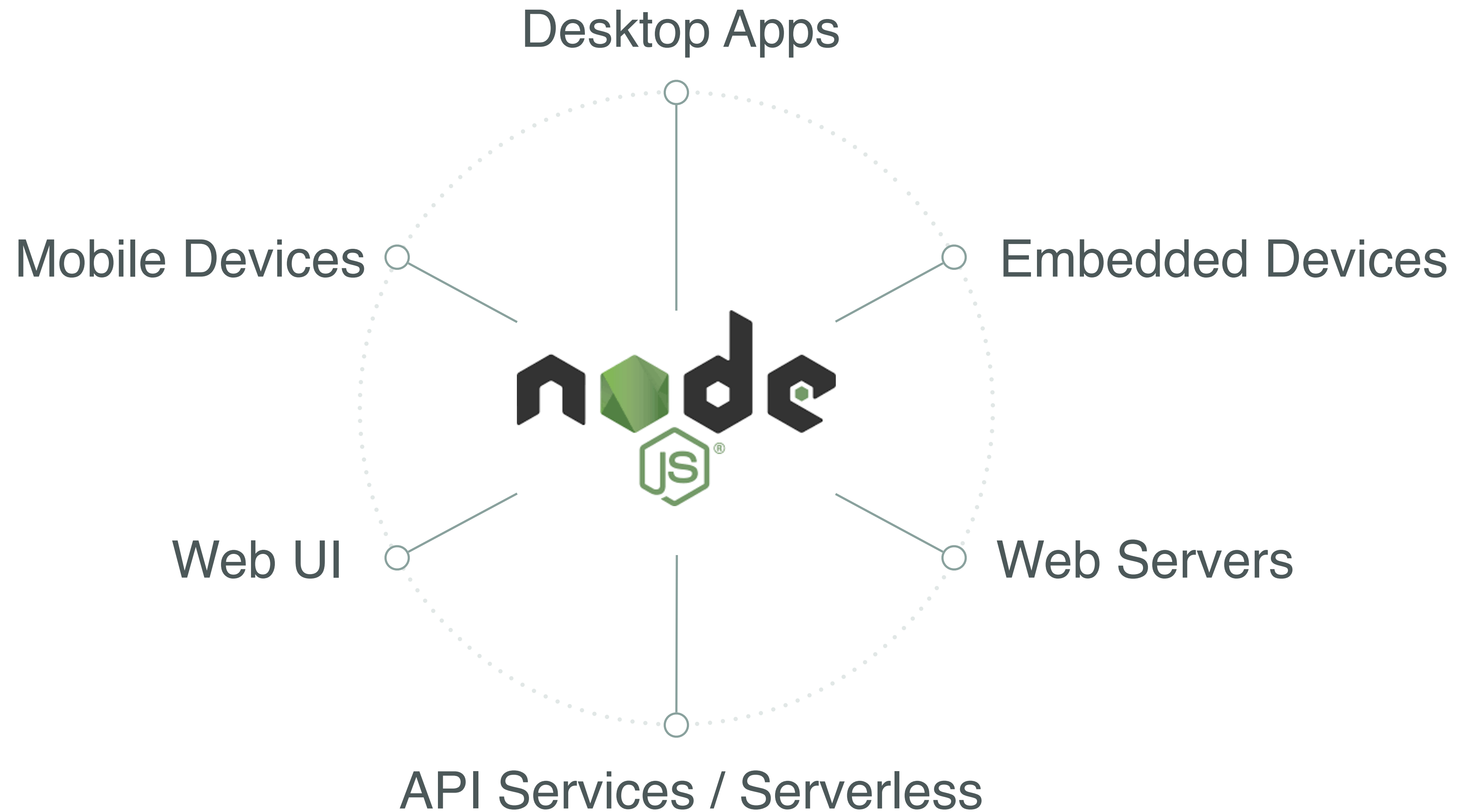
Node.js is the fastest growing  
open source project in the world.

Node.js has an estimated 7,000,000 users and is growing 100% year-over-year.



Note, only 19,000,000 known software developers worldwide as of 2016.

100% of the Fortune  
500 uses Node.js.



# Web Servers

- `express`
- `restify`
- `Hapi.js`
- `Sails.js`
- `Kraken.js`



## API Services / Serverless

- AWS Lambda
- Azure Functions

# Desktop Apps

- Electron
- NW.js
- Muon

# Embedded Devices

- NodeBots Ecosystem
- Tessel 2
- Node.js on ARM

# Web UI

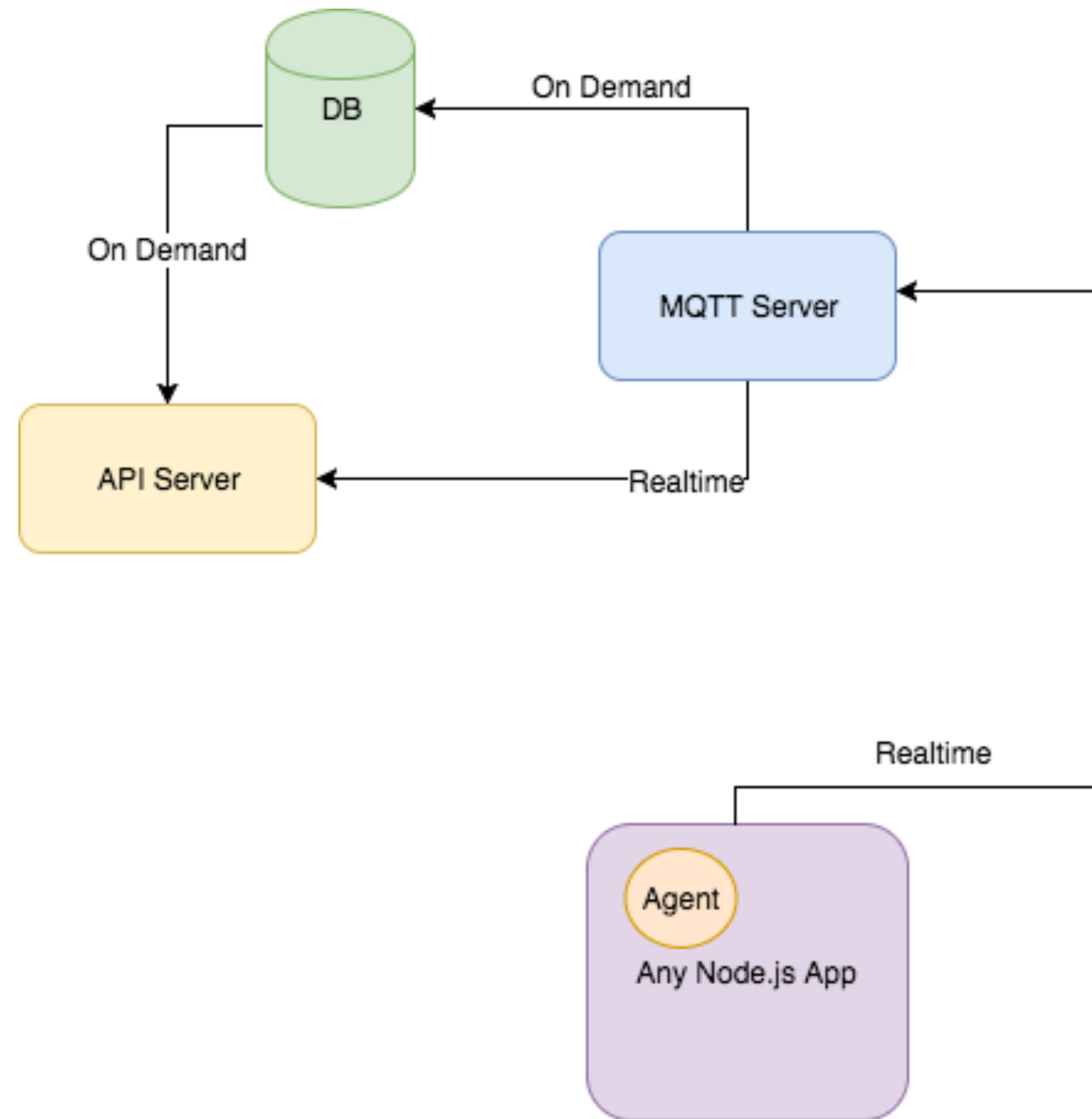
- Webpack
- Browserify
- Babel
- <Insert your Frontend Library here>

# Mobile Devices

- Apache Cordova / Phonegap
- React Native

<https://github.com/julianduque/platziverse>





# Thank you.

Julián Duque

[julian@nodesource.com](mailto:julian@nodesource.com)

[@julian\\_duque](https://twitter.com/julian_duque)