The Evolution of the Node.js Ecosystem

Jacopo Daeli
Lead Software Engineer @ GoDaddy

Developer Week
San Francisco, Feb 12th 2020

Quickly build (almost) anytype of product

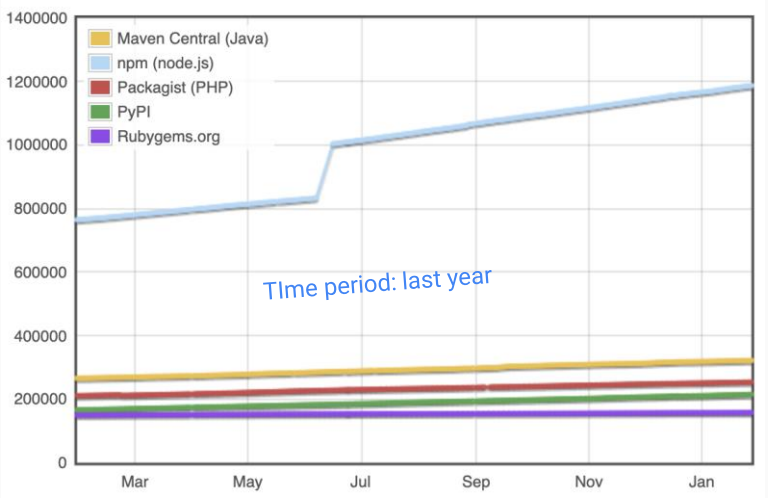
Agenda

- 35 mins presentation
- 15 mins Q&A

Node.js

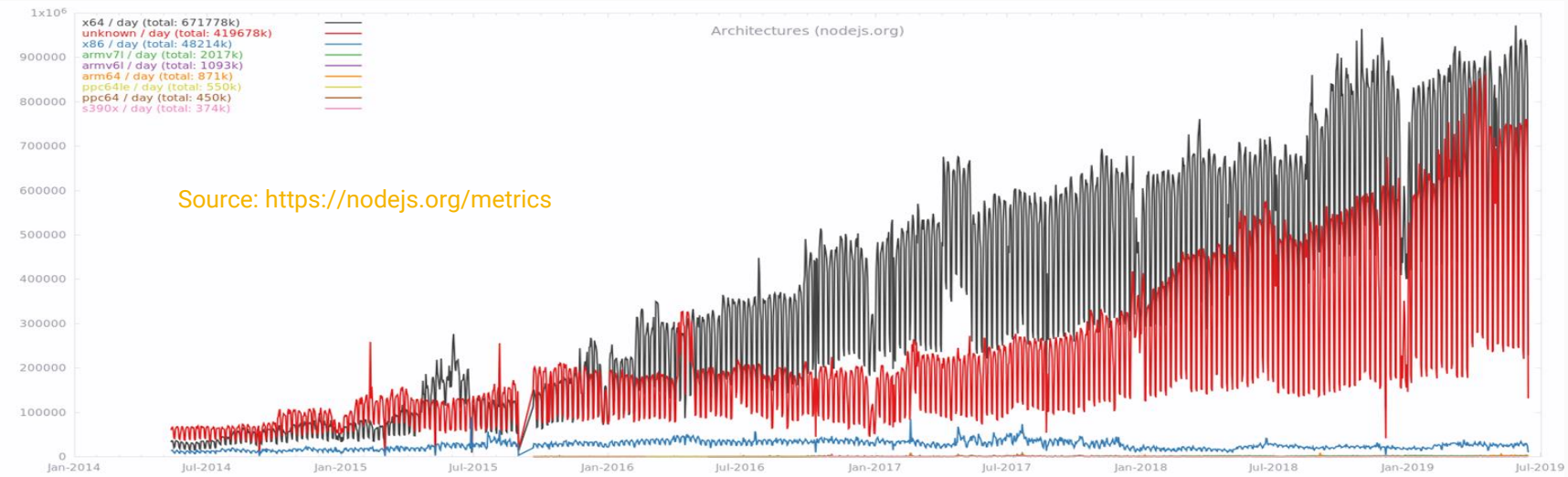
- **Fastest growing open source platform** in the world
- More than **twice** as many **modules** as the next competitor (Java)
- More than **one million modules** published
- More than **450 new packages** are published everyday

Source: http://www.modulecounts.com



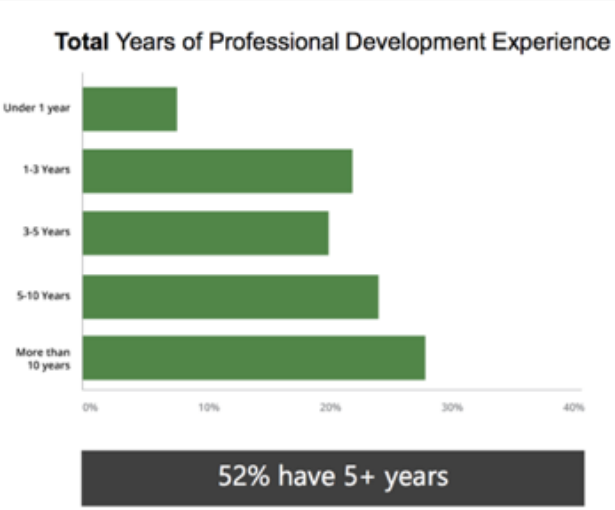
Node.js

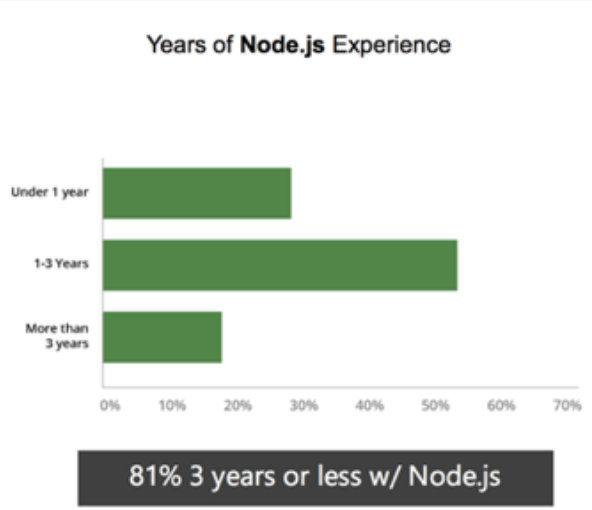
Node has an estimated **10M users** and is **growing at 50%** per year.



Node.js

Approximately 80% of the Node.js users have **3 years or less experience** with Node.





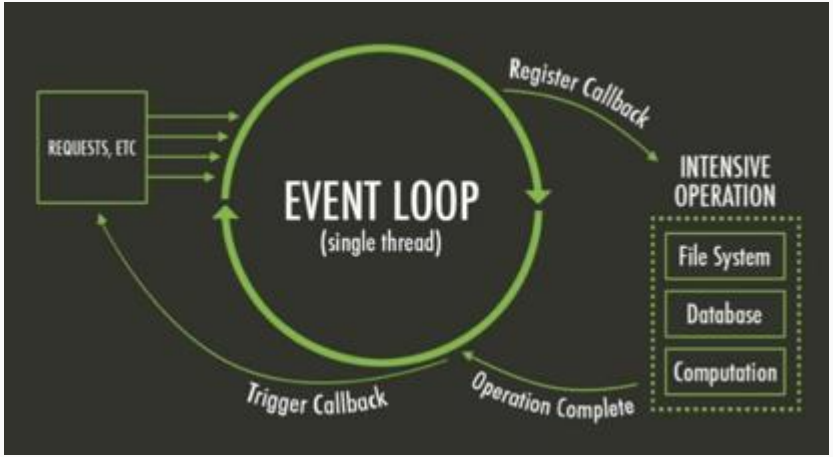
Node.js

Node.js

- **Ryan Dahl** started working on Node.js in 2009
- He was concerned about “**web servers speed**”
- Initial release supported only **Linux** and **Mac OS**
- February 2015, **Node.js Foundation** was announced
- September 2015, **Node.js v4.0** was released
- Feb 2020, **Node v12.15.0** was released

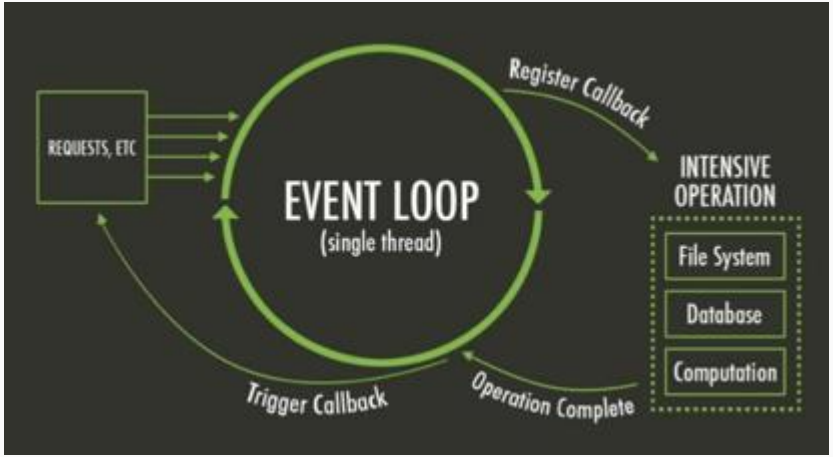
Node.js

- Node.js is a **runtime** for **JavaScript**
- You can run **JS code server side**
- It has **non-blocking** I/O operations
- I/O operations are **asynchronous**



Node.js Event Loop

- Offloads operations to the **System Kernel** whenever possible
- Modern Kernels are multi-threaded, they can **handle multiple operations** in background
- When an operation completes, the Kernel tells Node, and the associated **callback** is triggered to process the result



Node.js is a full stack ecosystem

- Web Frontend
- Mobile & Tablet
- Desktop Development
- Cloud Backends
- IoT Devices
- 3rd Party APIs & SOA
- Blockchain and DApps

Node.js is a full stack ecosystem

- **Perfect platform** to deal with all these components
- Node brings the **accessibility and ubiquity** of the web platform and of JavaScript to all these different areas
- You can **use Node** in a variety of different places **across teams**

Web Frontend

- Web Frontend development has **changed drastically** in the past 20 years
- The community developed **amazing tools** in Node.js and JavaScript to help developers build frontend applications

Web Frontend

- **Babel** for transpiling your JavaScript code
- **Standard** and **ESLint** to maintain your code quality with ease
- **JSHint** to help detecting errors and potential problems in your JavaScript code
- **Stylus**, **LESS**, etc. for better writing CSS

Web Frontend

- The **community wrote more tools** to help you manage all these tools
- Examples: **Gulp**, **Browserify**, **WebPack**, etc.

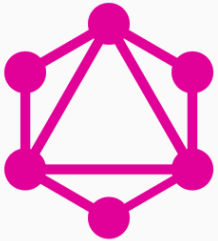
Web Frontend

React.js

- React is a JavaScript library for building **User Interfaces**
- Project started at **Facebook** around 2011 and publicly released in 2013
- It **rethought frontend frameworks** very differently
- The main idea is that a web framework should **exists in this compiled chain** where we have access to Node.js

GraphQL

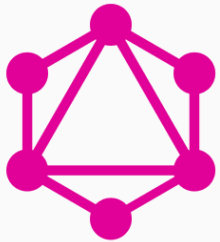
- GraphQL is a **data query language** initially developed internally by Facebook before being publicly released in 2015
- Currently the **Apollo Team** leads the GraphQL community
- Provides **an alternative to REST** and ad-hoc web service architectures



GraphQL

GraphQL

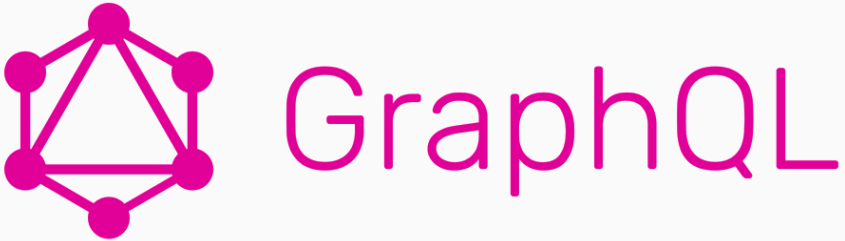
- Exposes three fundamental operations: **queries**, **mutations**, and **subscriptions**
- GraphQL is also a **runtime** for fulfilling those operations
- Gives clients the power to **ask for exactly what they need** and nothing more
- GraphQL APIs return **all the data** your app needs in a **single request**



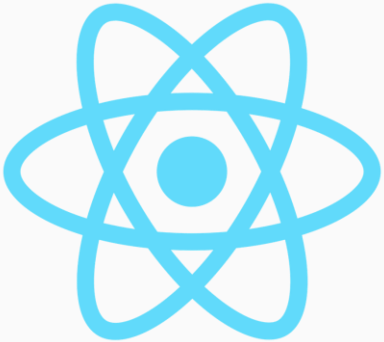
GraphQL

GraphQL

- GraphQL makes it **easier to evolve APIs** over time
- APIs are organized in terms of **types and fields**, not endpoints



Mobile & Tablet

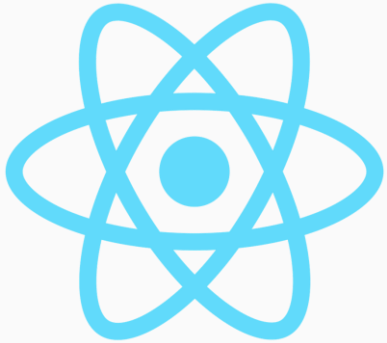


Apache Cordova

- **Apache Cordova**, previously known as **PhoneGap**
- Mobile apps with **HTML**, **CSS** and **JS**
- Cordova lets you build “**mobile web apps**”
- Target **multiple platforms** with one codebase
- **Free** and **open-source**

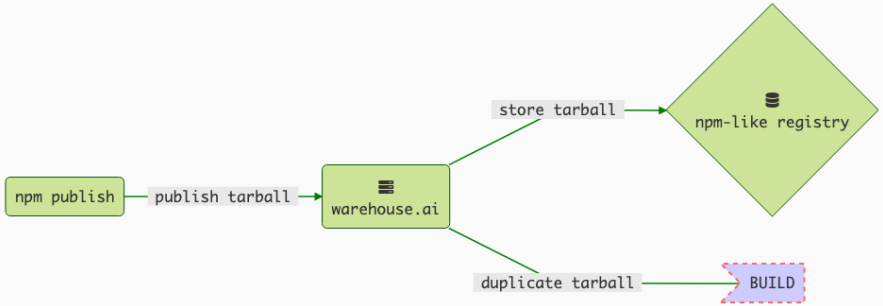
React Native

- React Native lets you build **Native apps** with only JavaScript
- Same **design** as React
- **Not** a mobile **web app** (or HTML app)
- Same **UI building blocks** as regular iOS and Android apps



Warehouse.ai

- A storage and developer **workflow** engine for **npm packages**
- Built internally at **GoDaddy** and now open-source
- Serving **built assets** for published modules through a **CDN target**
- **Build** and compile **npm packages** to run in the browser
  - **Webpack** as a Service
- https://github.com/godaddy/warehouse.ai



Desktop Development

Desktop Development is not dead!

Electron

- Electron “merges” **Chromium with Node.js**
- It brings the **Web Platform** and the entire **Node ecosystem** to Desktop Development
- It allows developers to build **cross-platform Desktop applications** using HTML, JavaScript and CSS
- Products built with Electron: **Slack**, **Atom**, etc.

Cloud Backends

- Node was built to build **Backends**
- It always had a great **Cloud story**
- Every well-known Cloud Provider **supports Node out of the box** and has pretty good Node support
- Google App Engine, AWS Elastic Beanstalk, Heroku, etc.

Google Cloud Platform

DigitalOcean

Serverless and Function as Service

- Nowadays we are moving into the **Function as a Service** (a.k.a **Serverless**) space
- Developers can build their applications out of these **discrete functions**
- Disclaimer : these **functions run on servers**, but developers don't have to worry about it

Serverless and Function as Service

- Node has been Cloud vendors' choice number one for Serverless because Node is **optimized to run in a single process**, with a **limited memory space**, and it is **great to handle a lot of I/O operations**
- Node **starts up quickly**
- These are very **important requirements** to successfully running in these Cloud Function services
- Indeed, we need a really **fast startup timing** to be really efficient in a space with very limited resources
- Moore's Law still **making computers faster**, but we are **keeping optimizing resources** using less and less of these computers to run our apps

Kubernetes External Secrets

- Integrate **external secret** management systems with **Kubernetes**
  - AWS **Secrets Manager**
  - **Vault** by HashiCorp
- Built internally at **GoDaddy**
- Project started during internal **company hackathon**
- Now **open-source**
  - 648 Stars and 77 forks

Kubernetes External Secrets

- Extends the **Kubernetes API** by
  - adding a `ExternalSecret` object Custom Resource Definition
  - a controller to implement the behavior of the object itself
- An `ExternalSecret` **declares how to fetch** the secret data
- The **controller converts** all `ExternalSecrets` to `Secrets`
- **Conversion** is completely **transparent** to `Pods` that can access `Secrets` normally

Kubernetes External Secrets

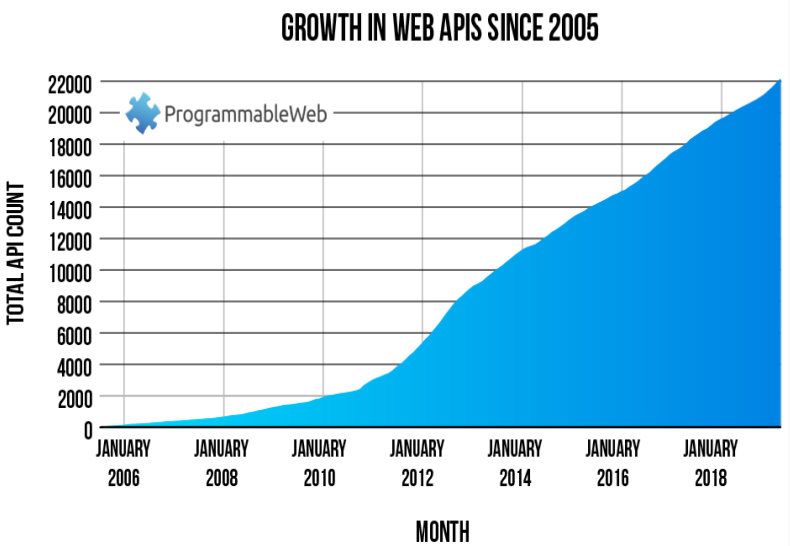
- Source code on GitHub
  - https://github.com/godaddy/kubernetes-external-secrets
- Very useful blog post
  - https://www.godaddy.com/engineering/2019/04/16/kubernetes-external-secrets
- Integrating AWS Secrets Manager with Kubernetes @ Paris.js slides
  - https://docs.google.com/presentation/d/17hJzgcRqke4gXjKQIF_U8tyR-ssBgi593ZqVKUBVB2M

IoT Devices

- Everyone wants a **button to do everything**
- Starting in 2012 the **Node ecosystem** started to see a big rise in **Robots**
- People working at NodeBots built a framework to **build Robots in JS** called “johnny-five”
- Over 75 developers contributed to the project

3rd Party APIs & SOA

- Everyone uses **APIs to build their applications**
- **Twilio** for Text Messages, **Mixpanel** for Analytics, **Google Speech** for Speech Recognition, etc.
- Most of these APIs have **Node SDK**
- Easy to use: **npm install**



Blockchain and DApps

A blockchain is a distributed list of **records**, called blocks, linked using **cryptography**.

A blockchain-network is a network of **computer nodes** that speak a **protocol** that decides **how to transact** on the blockchain (e.g., **Ethereum**).

A DApp is a computer program that runs on a **blockchain**-network.

DApps could be **disruptive**.

DApps have interesting properties

- Properties
  - Fault tolerant (probably exceeding AWS)
  - No central control after they're deployed (Google can't close up your favorite Dapp)
  - Open source (difficult not to be)
  - Incentivize with currency that's worth USD (you get $ if you help run Dapps)
- Examples
  - Golem (sell your idle CPU cycles)
  - Filecoin (get paid to store other users' files)
  - Stellar (services for low cost banking, microloans, microtransfers)
  - Escrow (sell your expensive stuff without an escrow)

Truffle

- Truffle is a **development environment**, testing framework and asset pipeline for Ethereum
  - It's written in JavaScript
  - Contracts are written in Solidity (contract-oriented, high-level language influenced by C++, Python and JavaScript)
  - Built-in **smart contract compilation**, linking, deployment and binary management
- Automated **contract testing** with **Mocha** and **Chai**
  - Write your tests in JavaScript

Web3.js

Web3.js is a **collection of libraries** which allow you to interact with a local or remote Ethereum node, using an HTTP, WebSocket or IPC connection.

Very useful for building functionalities in your app that needs to **communicate** with the Blockchain-network.

Why I love working with Node.js ?

Makes programming easier

Drastically reduces the entry barrier

Questions?

@JacopoDaeli

jacopo.daeli@gmail.com
www.jacopodaeli.com

Appendix 1 - Node.js Event Loop