The Node.js Ecosystem in Perspective

Dan Shaw, CTO of NodeSource

NODESOURCE®

November 7, 2016

Why so angry, dshaw?

Dan Shaw

CTO and Co-Founder of NodeSource.

Node.js startup veteran:
Storify, Spreecast, Voxer, ClassDojo.

Podcast host of NodeUp.

Created NodeBots Day, NodeBots SF, SFNode,
and EnterpriseJS.

Before Node.js did large-scale contracting
in Defense, Health Care and Education.
Primarily Java backend and JavaScript frontend.

 © 2016 NodeSource

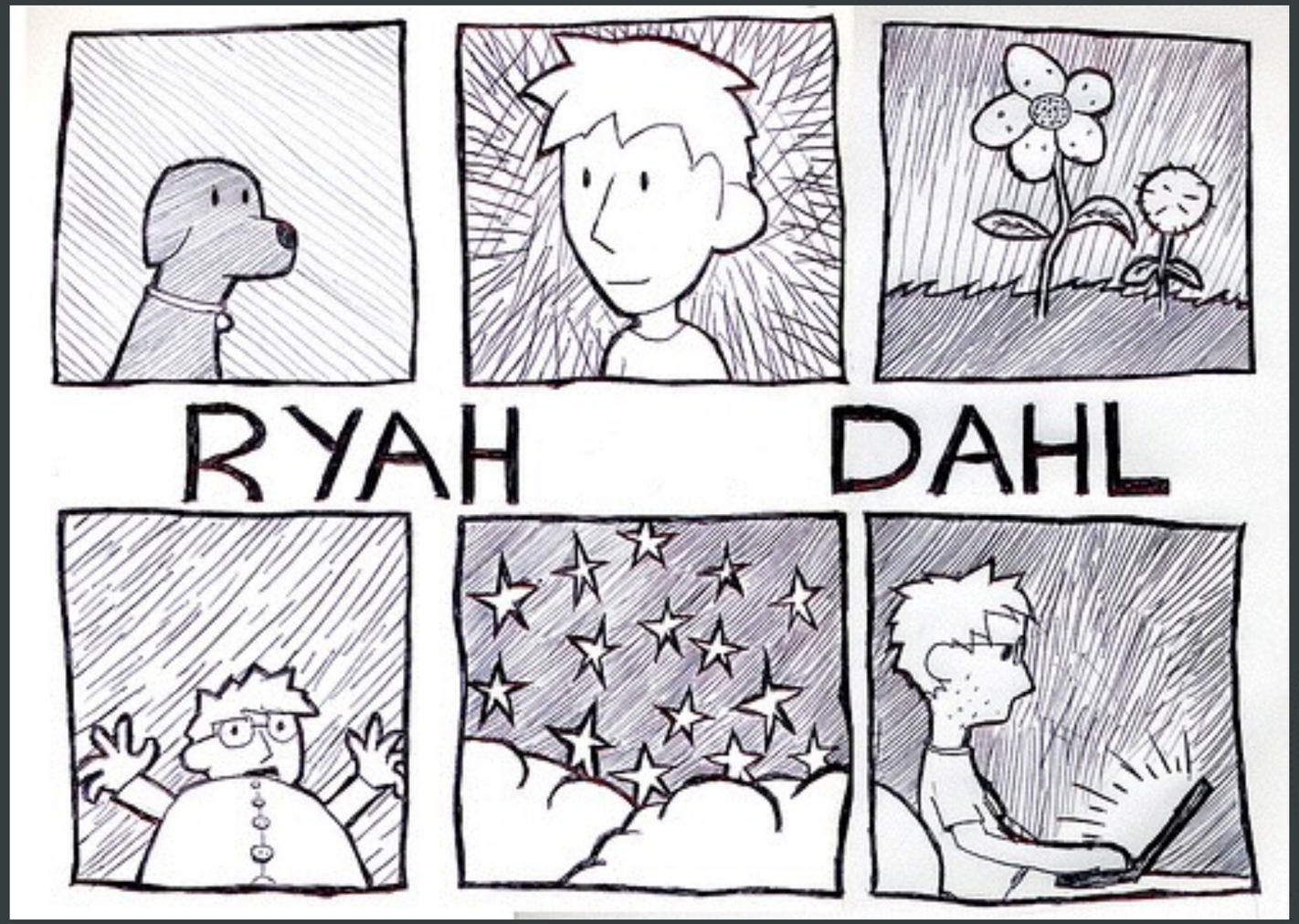
NodeSource is *the* Enterprise Node.js company offering the only commercial version of Node.js explicitly focused on the needs of Enterprise users of Node.js.

NODESOURCE ™

© 2016 NodeSource

In the beginning

© 2016 NodeSource



© 2016 NodeSource

Ryan Dahl

- Wants to improve application development
- Created on GitHub in the open
- Evented I/O
- C++ and JavaScript
- No package manager
- Mostly *nix-based

© 2016 NodeSource

© 2016 NodeSource

Key Platform Decisions

- Node.js ships with "batteries included”
- Node.js was built with a very liberal license
- Node.js has no standard library and encourages userland experimentation
- Node.js chose to standardize on a single package management solution with npm
- Node.js is fully cross-platform
- Node.js applications run everywhere (take that Java!)

© 2016 NodeSource

Node.js ships with "batteries included”

© 2016 NodeSource

Node.js ships with "batteries included”

- Node.js provides the platform and base API for application development
- JavaScript and Native interfaces provided by Google Chrome’s V8 VM
- V8 is bundled with Node.js
- Evented I/O originally built using libev
- libev replaced by libuv
- libuv was created for v0.6 to enable full Windows support

© 2016 NodeSource

Node.js ships with "batteries included”

- Other notable dependencies:
  - http-parser (by Ryan Dahl)
  - OpenSSL (Crypto)
  - C-Ares (DNS)
  - zlib (Compression)
- Notable exception: ICU (Internationalization) *Too big!

© 2016 NodeSource

Node.js ships with "batteries included”

© 2016 NodeSource

Node.js ships with "batteries included”

- This decision has caused some issues with other ecosystems like Debian
  - Debian packages are supposed to link to the dependencies rather than bundling them

© 2016 NodeSource

Node.js was built with a very liberal license

© 2016 NodeSource

Node.js was built with a very liberal license

- Node.js is licensed MIT
- The project identity is protected by copyright law
- Permissive license enabled io.js to happen
  (which is a good thing)

© 2016 NodeSource

Node.js has no standard library and encourages userland experimentation

© 2016 NodeSource

Node.js has no standard library and encourages userland experimentation

- “Standard libraries is where code goes to die” - ryah
- Intentionally small core
- Does it need to be in core to run?

© 2016 NodeSource

Node.js chose to standardize on a single package management solution with npm

© 2016 NodeSource

Node.js has no standard library and encourages userland experimentation

- There were originally multiple package managers
  - Kiwi by TJ Holowaychuk
  - npm by Isaac Schlueter
- With Node.js v0.6 we selected npm as the included package manager and built the first registry
- Now the largest registry of any language

© 2016 NodeSource

Node.js is fully cross-platform

© 2016 NodeSource

Node.js is fully cross-platform

- Node.js v0.6 shipped the first Windows release
- Thank you, Microsoft!
- Node.js still exposes some very *nix APIs even on Windows
- Nearly 50% of all Node.js developers run Windows

© 2016 NodeSource

Node.js applications run everywhere (take that Java!)

© 2016 NodeSource

Node.js applications run everywhere (take that Java!)

- JavaScript code runs everywhere
- Exception: Native code needs to be compiled for each target platform
- Companies like Netflix are explicitly limiting native code usage to maximize compatibility and interop

© 2016 NodeSource

Key Platform Decisions

- Node.js ships with "batteries included”
- Node.js was built with a very liberal license
- Node.js has no standard library and encourages userland experimentation
- Node.js chose to standardize on a single package management solution with npm
- Node.js is fully cross-platform
- Node.js applications run everywhere (take that Java!)

© 2016 NodeSource

These decisions shaped Node.js

© 2016 NodeSource

io.js was an important exploration

© 2016 NodeSource

Node.js is all grown up

© 2016 NodeSource

Today’s Node.js exists under the Node.js Foundation

© 2016 NodeSource

Modern Node.js

© 2016 NodeSource

Key Platform Decisions of the Present

- How best to support native Promises?
- Node.js <3 V8
- Node.js <3 ChakraCore
- Node.js <3 TC-39
- Node.js <3 ES Modules

© 2016 NodeSource

Native Promises Support in Node.js

© 2016 NodeSource

Native Promises Support in Node.js

- Love ‘em, hate ‘em or just simply can’t debug ‘em… PROMISES ARE HERE TO STAY
- Adoption is increasing rapidly
- Driven by fronted frameworks
- Bluebird is the library of choice for now
- Async/Await changes everything since it relies on native promises, making libraries challenging at best and obsolete at worst.

© 2016 NodeSource

Native Promises Support in Node.js

- Native Promise support in Node.js is provided by Google's V8 JavaScript VM
- In Node.js, the event loop must be aware of every event
- Currently, Promises are managed by microtask queue outside of Node.js visibility
- One solution is to replace built in Promises with a language-level implementation that Node.js provides

© 2016 NodeSource

Node.js <3 V8

© 2016 NodeSource

Node.js ❤️ V8

- V8 provides the JavaScript language capabilities
- V8 handles object allocation and memory management
- Node.js incorporates V8 releases far more rapidly
- V8 releases still can trigger a semver major
- VM Working Group working on ABI (bindings) stability
- Thanks to ChakraCore, Node.js has test suite

 © 2016 NodeSource

Node.js <3 ChakraCore

© 2016 NodeSource

Node.js ❤️ ChakraCore

- ChakraCore provides JavaScript language capabilities for Edge and Windows
- Time-travel debugging is amazing 😍
- ***test262*** provides cross-VM conformance tests
- Experimental features are easier to test with ChakraCore because it's more open

© 2016 NodeSource

Node.js <3 TC-39

© 2016 NodeSource

Node.js ❤️ TC-39

- The *Fetch API* being tied to async because Promise-based triggered concern
- Aligning Node.js Common.js and ES Modules has been challenging
- Node.js has over 5 million users who don’t want their code to break
- Node.js Technical Steering Committee now represented on TC-39 and will continue to maintain active membership

© 2016 NodeSource

Node.js <3 ES Modules

© 2016 NodeSource

Node.js ❤️ ES Modules

- Bradley Farias (neé Meck) did an incredible job assessing all possible solutions. 👏
- James Snell is working with TC-39 to sort out the details
- We will need to change the ES Modules spec to accommodate the backwards compatibility for Node.js
- JavaScript and Node.js are aligning on one true way to express JavaScript modules

© 2016 NodeSource

Node.js is Everywhere!

© 2016 NodeSource

Powered by Node.js

- Frontend tooling
- Web applications (SSR - Server-side Rendering)
- APIs
- Internet of Things
- Desktop Applications with Electron (Slack much?)

© 2016 NodeSource

Always bet on

© 2016 NodeSource

Thank you.

Dan Shaw

nodesource.com

@dshaw

NODESOURCE®