Node.js Road to 1.0

Isaac Z. Schlueter – 2013

Brief History

- 2009, Ryan Dahl, JSConf EU, Joyent, etc.
- 2010.08.20 v0.2 "stable-ish"
- 2011.02.10 v0.4 "stuff works"
- 2011.11.04 v0.6 "stuff works on windows"
- 2012.06.25 v0.8 "stuff works better"
- 2013.03.11 v0.10 "streams2"

Node v0.10

- Consistent easily-extensible Stream API
- API polishing (domains, nextTick, IdleGC)
- Continuous Integration
- Arrival of several enterprise-focused Node offerings
- Continued exponential growth of public modules

streams2

- Change from "spew style" to "pull style"
- var data = stream.read() vs stream.on('data', fn)
- Classes for: Writable, Duplex, Transform, Passthrough
- pause() actually pauses now (No user-buffering)
- "Paving the cowpaths"

streams2

- Massive internal refactoring
- Nearly every API was touched deeply
- Still, 99% backwards compatible
  (one caveat: on('end') for un-consumed streams)

npm-www future plans

- We have data & momentum, need more mad science
- Integration with github repo data, github login
- Recommendation engine magic
- Module ranking: CPAN Kwalitee-style, ★'s

Deprecation

- The days of Node breaking APIs on purpose are pretty much over. We're growing up.
- If your program works today, we will try hard to make sure it works next year.
- No one expected Node to be this mature by now.
- This is **why** we've kept the core small!

ING: VAPORWARE WARNING: VAPORWARE WARNING: VAPORWARE WARNING: VAPO

Node v0.12

- TLS: Refactoring for performance
- HTTP cleanup: Client changes, but API-stable
- Cluster: Load balancing improvements
- Buffers: Address points of slowness
- Treat as a "v1.0-RC"

ING: VAPORWARE WARNING: VAPORWARE WARNING: VAPORWARE WARNING: VAPO

Buffers

- Persistent<T> and MakeWeak are too slow
- Investigating multiple avenues to attack this problem:
  - ThinBuffers: User-controlled allocation/deallocation
  - Make Buffer first-class V8-ism
- Improving this improves pretty much everything in Node

TLS

- Traditionally one of Node's weakest points.
- Too much pointless Buffer creation
  (Persistent object that immediately disappears)
- tls.CryptoStream will stick around for backwards-compatibility, but won't be used by default

```
                 +-------------------------------------------------------+
                 |  Userland JavaScript code                  +          |
                 +----------------------------------------^---|----------+
      +--------                                           |   |             BUFFER CREATION
      |          +----------------------------------------|---|----------+
      |          |  http_parser                           +   |          |
      |          +-------------------------------------^------|----------+
      |                                                |      |
      |          +-------------------------------------|------|----------+
      |          |  tls.CryptoStream pair              |      |          |
      |          |                                     |      |          |  BUFFER CREATION
      |          |      +------------------------+     |      |          |
node  |          |      |                       +------+      |          |
core  |          |      |  OpenSSL               |            |          |
      |          |      |           +           <-------------+          |
      |          |      +-------^---|------------+                       |
      |          |              |   |                                    |
      |          +--------------|---|------------------------------------+
      |                         |   |
      |          +--------------|---|------------------------------------+
      |          |  net.Socket  +   v      +                             |
      |          +---------------------^---|-----------------------------+
      |                                |   |                                BUFFER CREATION
      |          +---------------------|---|-----------------------------+
      |          |  system (libuv)     +   v                             |
      +--------  +-------------------------------------------------------+
```

ING: VAPORWARE WARNING: VAPORWARE WARNING: VAPORWARE WARNING: VAPO

http

- One file, did too much.
- Already split up, code much easier to manage
- Also has too much pointless buffer creation
- Client: Queueing is Fail (5 always the wrong number)
- KeepAlive for real

ING: VAPORWARE WARNING: VAPORWARE WARNING: VAPORWARE WARNING: VAPO

cluster

- Not balanced well enough
- One or two workers end up doing most work
- This causes problems in some of the exact cases where you need cluster to be working!
- Solution: round-robin scheduling.

ING: VAPORWARE WARNING: VAPORWARE WARNING: VAPORWARE WARNING: VAPO

Other Stuff

- execSync
- stream writev support
- Perhaps some breaking V8 API changes (ouch!)
- Better IPv6 support in more areas
- Specify DNS providers
- Bugfixes, etc.

ING: VAPORWARE WARNING: VAPORWARE WARNING: VAPORWARE WARNING: VAPO

Node v1.0

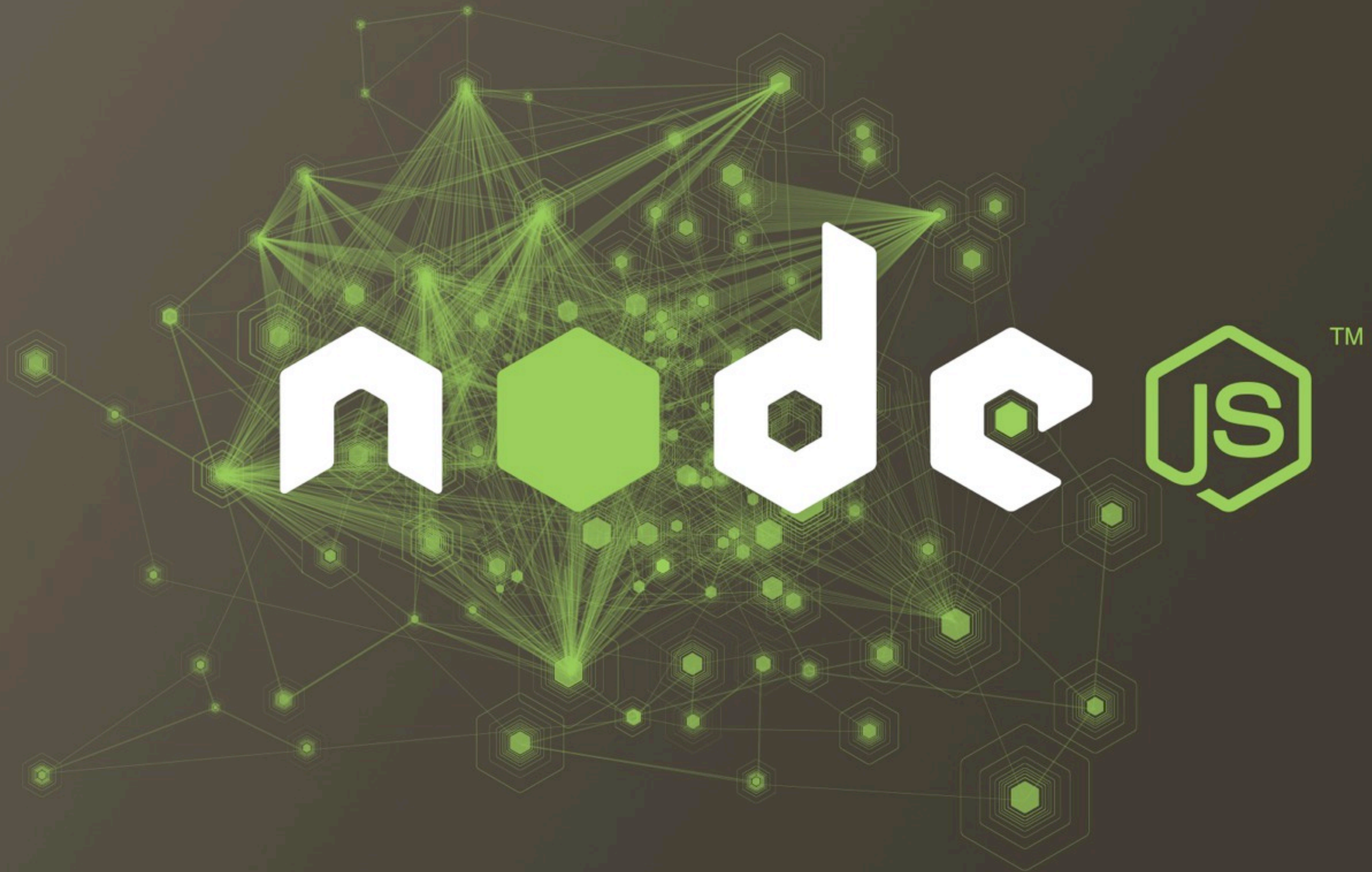
- Will be the next stable release after v0.12
- API is effectively 1.0 today, but internals still in motion
- No new features planned for post-v0.12
  should be mostly indistinguishable from v0.12

What's in a Number?

- Continual increases in Stability Indexes.
- Speed increases, V8 upgrades, new ES features, etc.
- Node is pretty stable now. Use it. It's fine, really.
- Nothing weird is going to happen.
- There will never be a Perl6/Python3 of Node.

Stability and Momentum

- Changes to core **slow down** Node-land
- Stable base best way to foster community innovation
- Change: antithesis of stability (Ahh, trade offs...)
- Crazily prolific and quickly growing community
- More node-core gets out of its way, the better!



http://j.mp/2013-road-to-node-10