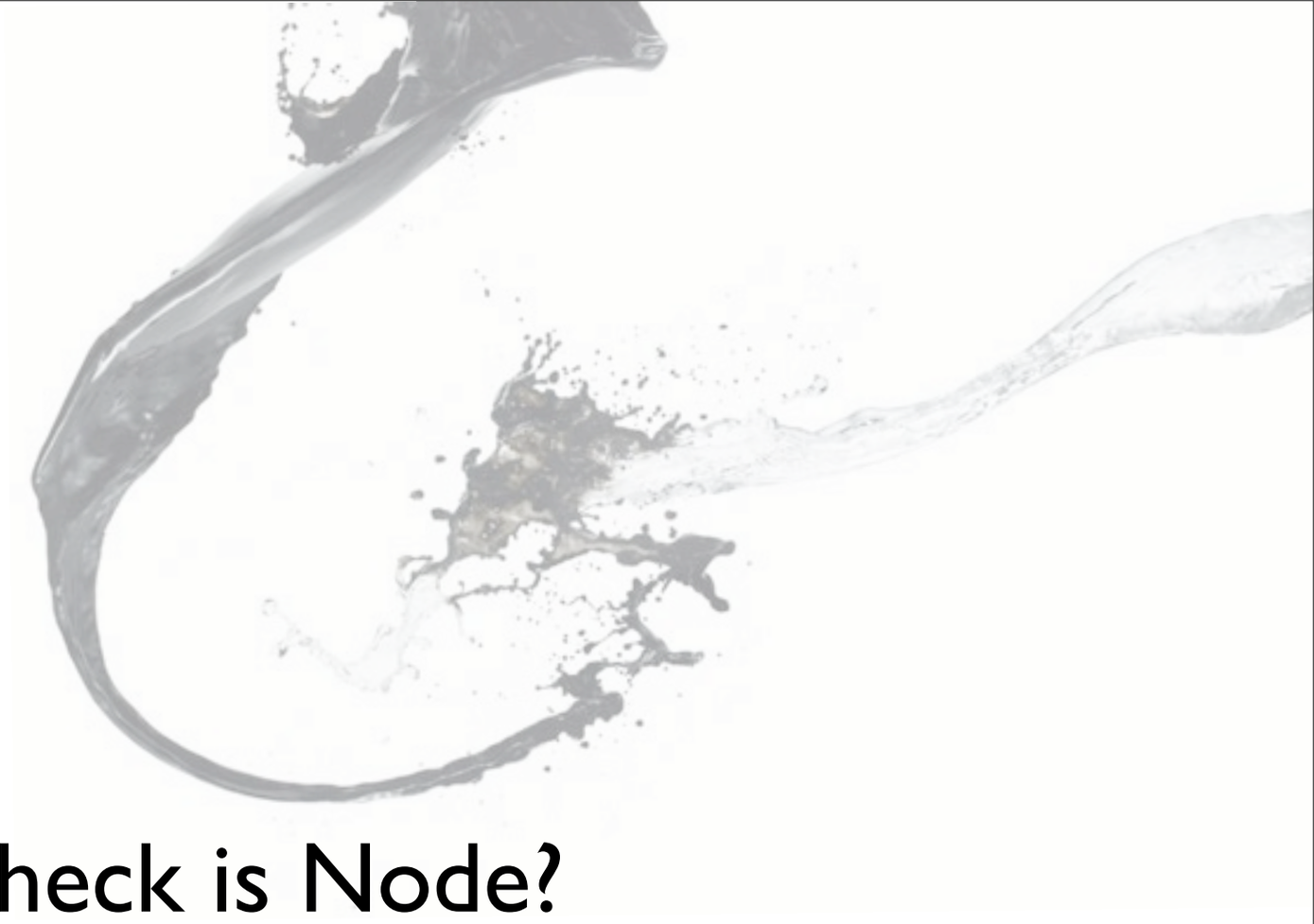
Node.js

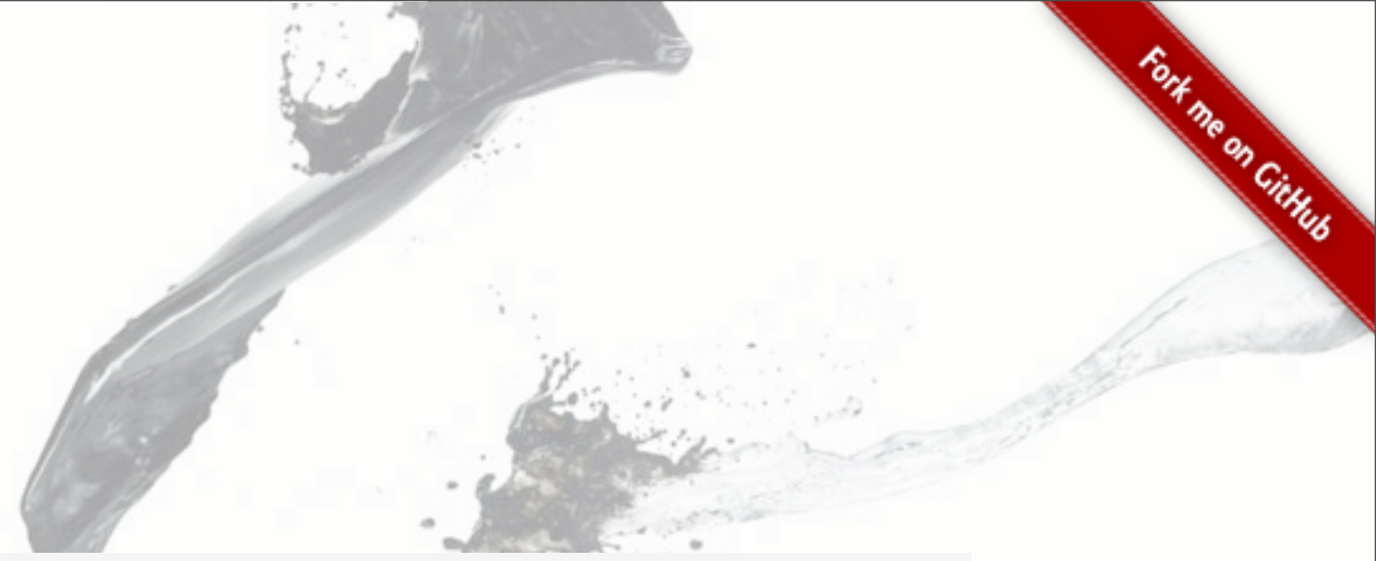


What the heck is Node?

Ryan Dahl “A bunch of Sugar on Top of V8”... focus is on doing networking correctly.

Node is a set of bindings to the google V8 JS engine.
Allows you to script programs that do IO in js
focused on hi performance

WESLEY REISZ



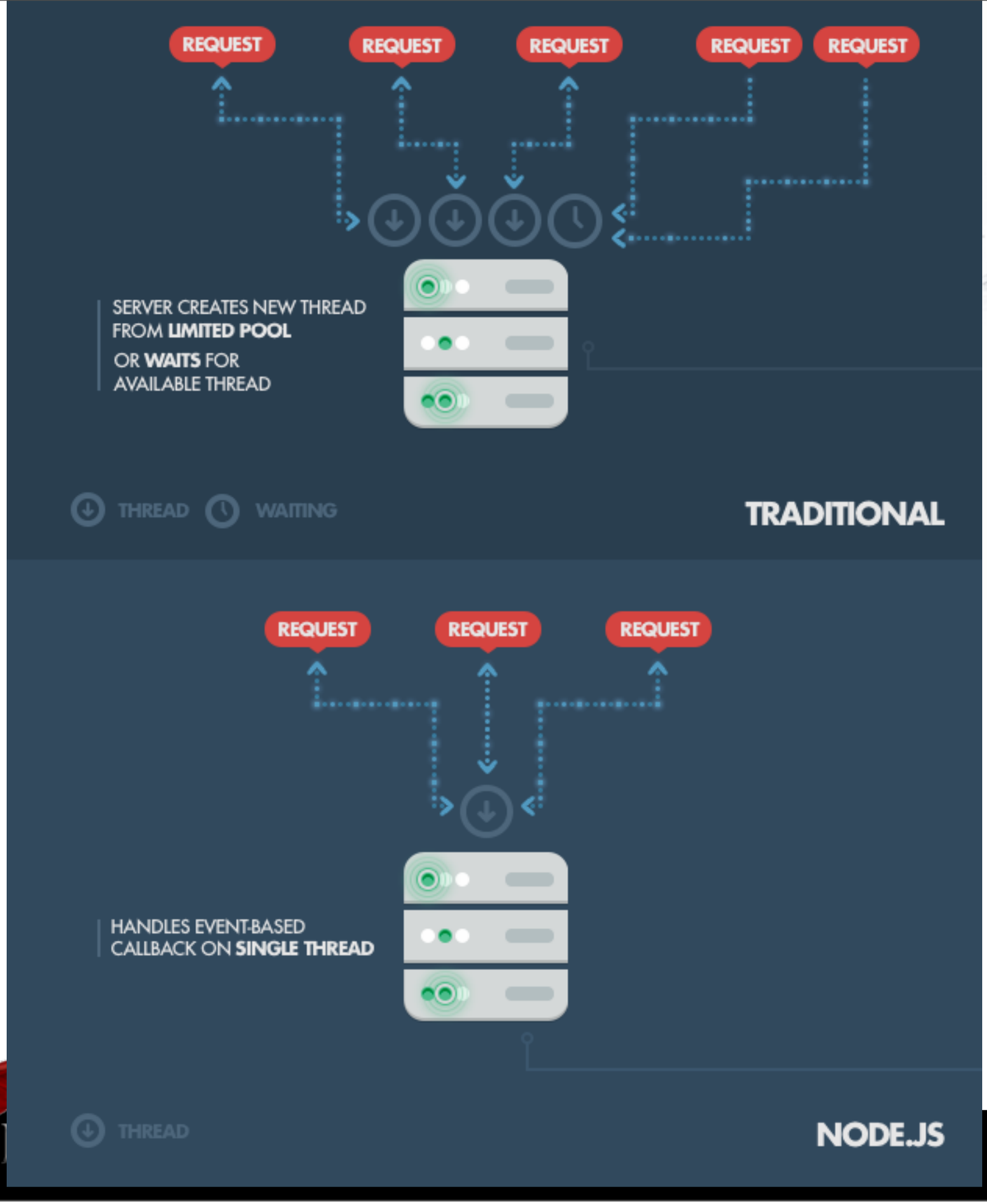
Fork me on GitHub

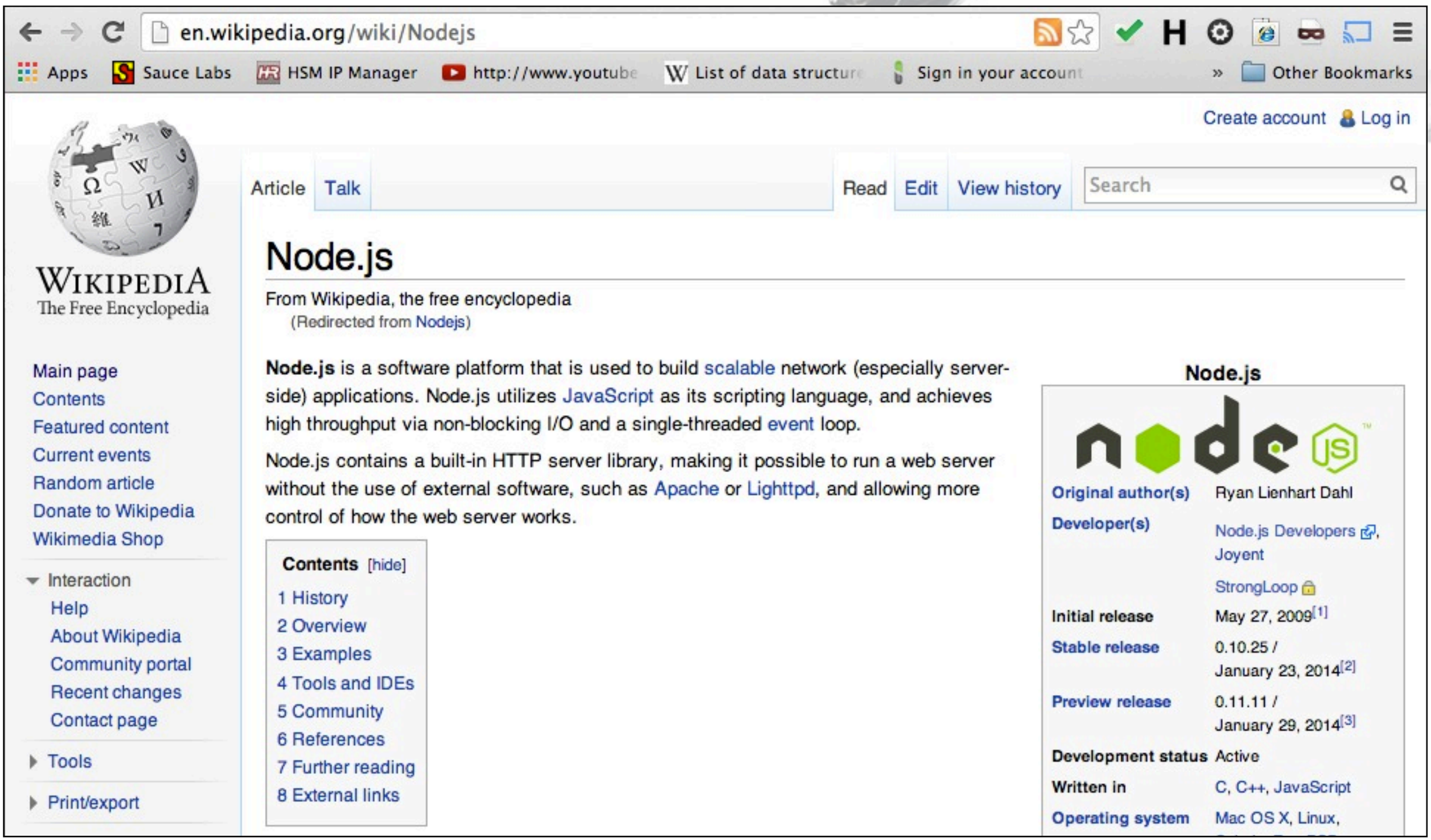
```
1
2  // Load the http module to create an http server.
3  var http = require('http');
4
5  // Configure our HTTP server to respond with Hello World to all requests.
6  var server = http.createServer(function (request, response) {
7      response.writeHead(200, {
8          "Content-Type": "text/plain"
9      });
10     response.end("Hello World\n");
11 });
12
13 // Listen on port 8000, IP defaults to 127.0.0.1
14 server.listen(8000);
15
16 // Put a friendly message on the terminal
17 console.log("Server running at http://127.0.0.1:8000/");
```

Node.js is a software platform that is used to build scalable network (especially server-side) applications. Node.js utilizes JavaScript as its scripting language, and achieves high throughput via non-blocking I/O and a single-threaded event loop.

Node.js is a software platform that is used to build **scalable** network (especially server-side) applications. Node.js utilizes JavaScript as its scripting language, and achieves high throughput via **non-blocking I/O and a single-threaded event loop**.

Demo:
interactive mode first... echo example
walk through step1 - step3
curl http://127.0.0.1:8000/
ab -n 100 -c 10 http://127.0.0.1:8000/
compare to java on tcat
curl -v http://127.0.0.1:8000/
show processes/threads


REQUEST
REQUEST
REQUEST
REQUEST
REQUEST
SERVER CREATES NEW THREAD
FROM LIMITED POOL
OR WAITS FOR
AVAILABLE THREAD
THREAD
WAITING
TRADITIONAL
REQUEST
REQUEST
REQUEST
HANDLES EVENT-BASED
CALLBACK ON SINGLE THREAD
THREAD
NODE.JS



http://en.wikipedia.org/wiki/Nodejs

Node.js is a lightweight web container that is a strong option for development of micro services and as a server to mobile and single-page web applications. Due to the asynchronous nature of node.js, developers are turning to promise libraries to simplify their application code.

As the use of promises mature within the node.js community, we expect to see more applications developed for node.js.

For those teams that are reluctant to try node.js in production, it is still worthwhile to consider node.js for development tasks like running JavaScript tests outside of the browser or generating static web content from tools like CoffeeScript, SASS, and LESS.

Where Can I find node?

WESLEY REISZ

Where Can I find node?

- https://github.com/joyent/node (of course)
- http://nodejs.org/dist/

- Walk thru a super simple chat application:
  - install node (http://nodejs.org/dist/)
  - write a chat client

```
var net = require('net');
var sockets = [];
var s = net.Server(function(socket){
   sockets.push(socket);

   socket.on('data',function(d){
     for(var i=0; i<sockets.length; i++){
         if(sockets[i] == socket){
            continue;
         }
         sockets[i].write(d);
     }
   });
   socket.on('end',function(){
     var i = sockets.indexOf(socket);
     sockets.splice(i,1);
   });
});

s.listen(8000);
```

- How about something more non trival
  - https://github.com/juchi/tictactoe
- Let's talk through a HTML5 (using websockets) and Node.js ticktactoe client realtime

"A" Benchmark

Java

Concurrent Requests	Average Response time (ms)	Requests/second
10	23	422
50	119	416
100	243	408
150	363	411

Node.js

Concurrent Requests	Average Response time (ms)	Requests/second
10	19	509
50	109	453
100	196	507
150	294	506

The Node.js is 20% faster than the Java EE solution for the problem at hand. That amazed me. An interpreted language as fast as a compiled language on a VM in which years of optimization have gone into. Not bad at all!

http://java.dzone.com/articles/performance-comparison-between?page=0,1

Using Node.Js PayPal Doubles RPS, Lowers Latency, With Fewer Developers, But Where Do The Improvements Really Come From?

WEDNESDAY, DECEMBER 11, 2013 AT 8:54AM

PayPal gives yet another glowing report of an app rewritten in node.js experiencing substantial performance improvements. PayPal rewrote their account overview page, one of the most trafficked apps on the website, which was previously written in King Java.

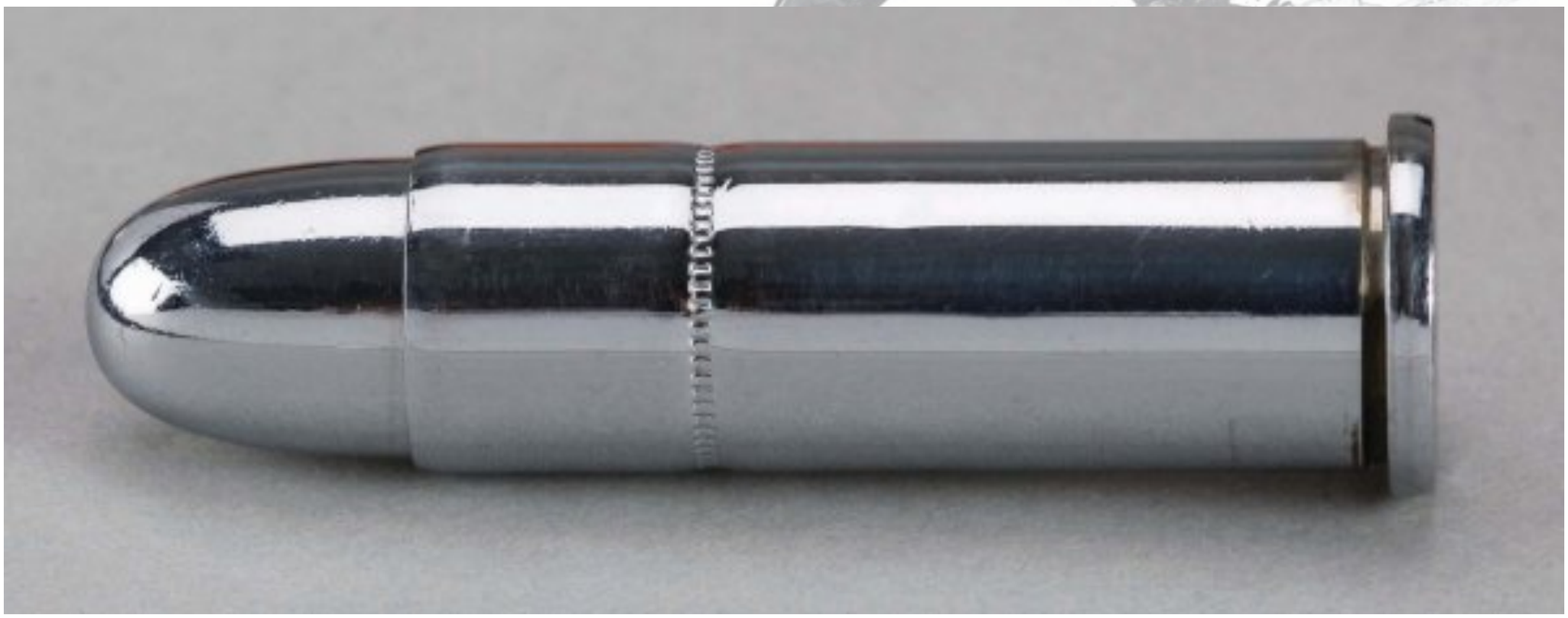
The benefits:

1. Full-stack engineers. Using JavaScript on both the front-end and the back-end removed an artificial boundary between the browser and server, allowing engineers to code both.
2. Built almost twice as fast with fewer people
3. Written in 33% fewer lines of code
4. Constructed with 40% fewer files
5. Double the requests per second vs. the Java application.
6. 35% decrease in the average response time for the same page.

- http://highscalability.com/blog/2013/12/11/using-nodejs-paypal-doubles-rps-lowers-latency-with-fewer-de.html

- Other Stuff (yes, that's a technical term):
  - Node exposes the full debugger from V8 (see debugger)
  - It's fast... it's really fast.
  - Easy move for Web Developers... that are WEB developers.
  - It's JavaScript
  - Extremely active Community
  - It has massive velocity!
  - Shared NOTHING model... if you want to talk to other instances, you have to open a socket and send it data. This means SCALE.
  - Express.js is the web framework of choice

- Now the Bad:
  - Node is single-threaded and doesn't automatically make use of more than a single core in your potentially multi-core machine. This means unless you design it differently, your application won't take full advantage of the available capacity the server hosting it has to offer.
  - It's a start from scratch. If you don't have a core driver.... you have to write one. ...or the ones that are out there are crap (not my words).
  - Node.js is NOT immature anymore, but is still evolving fast. Careful of the bleeding edge
  - Don't have a stack trace to follow back with as in threaded applications. You have short stacks in node. Can be difficult to see what called the error point.
  - Managing server state requires something like gossip or redis



No Silver Bullet, but...

WESLEY REISZ

- Resources:
  - http://www.youtube.com/watch?v=jo_B4LTHi3I
  - http://java.dzone.com/articles/performance-comparison-between
  - http://highscalability.com/blog/2013/12/11/using-nodejs-paypal-doubles-rps-lowers-latency-with-fewer-de.html
  - https://engineering.gosquared.com/optimising-nginx-node-js-and-networking-for-heavy-workloads
  - http://blog.argteam.com/coding/hardening-node-js-for-production-part-2-using-nginx-to-avoid-node-js-load/
  - https://github.com/wesreisz/nodeSamples