CSC309 Winter 2016
Lecture 8

Larry Zhang

Announcement

- Assignment 3 is out
- Important dates:
    - **March 8, 10:00pm: proposal submitted to MarkUs**
    - March 18, 10:00pm: Checkpoint 1 submitted to MarkUs (finish front-end implementation)
    - March 29, in lecture, showcase
    - April 1, 10:00pm: Final submission

Proposal

- a one / two page PDF that describes what game you want to implement
- may include mock-up pictures of your game
- filename: proposal.pdf

The overall setup

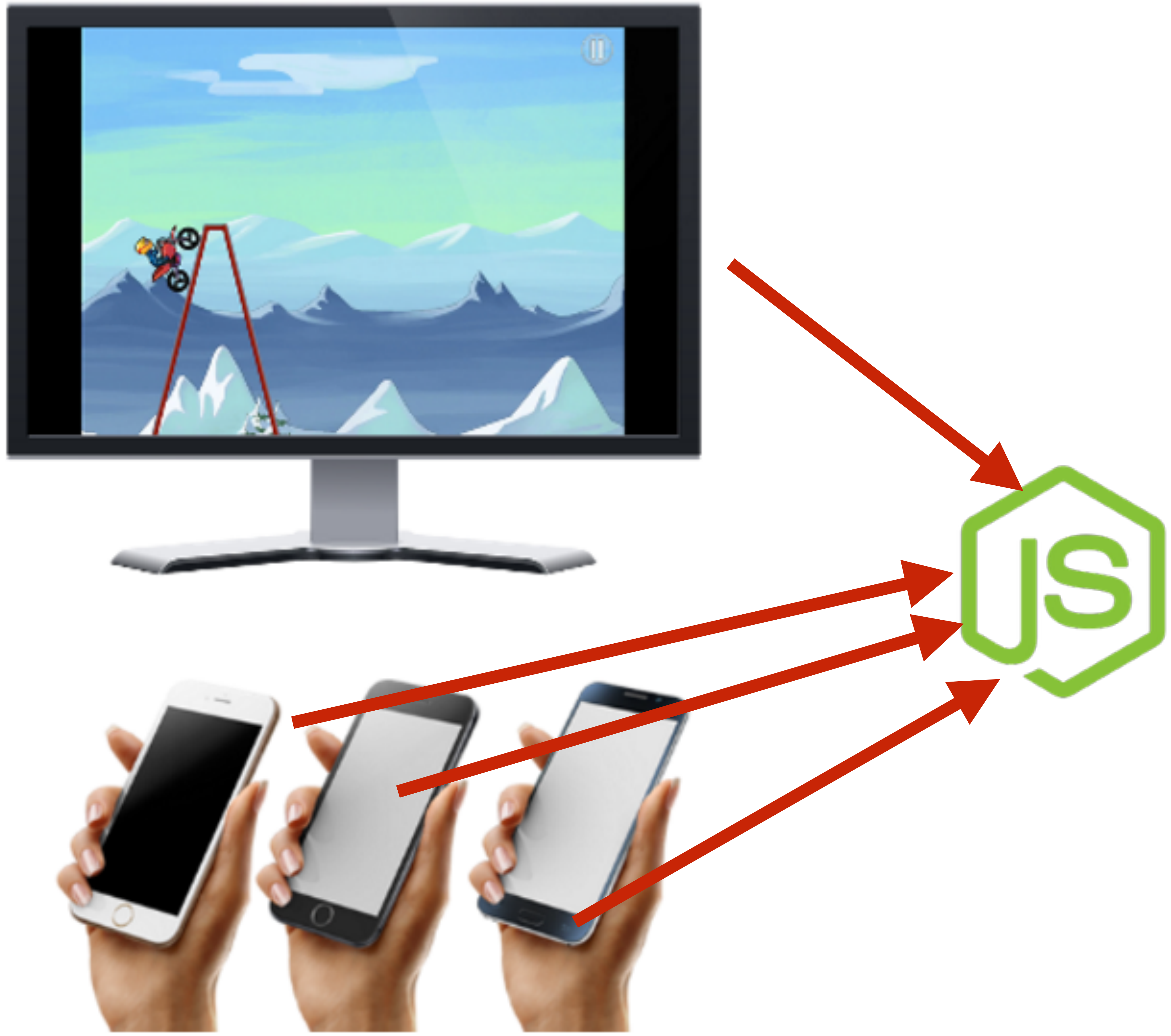
- Multiple players staring at the same screen and play against each other (or cooperatively) using their phone

- A NodeJS / WebSocket server

- game display page, with a WebSocket client

- game control page, with a WebSocket client

- plus other things like database if needed



Evaluation

- We expect to see “3 weeks’ worth of work” into your product
- Basic requirement is minimal
- Big part of it would be additional / creative features
  - We have a list possible features updated on Piazza’s A3 FAQ

Today’s topic: Node.js

What is Node.js

- NodeJS is a server side runtime environment for web applications.
- Written in Javascript
- Use Google’s V8 JavaScript engine as interpreter
- Originally written in 2009 by Ryan Dahl.
- A package manager npm, introduced for Node.js in 2011.

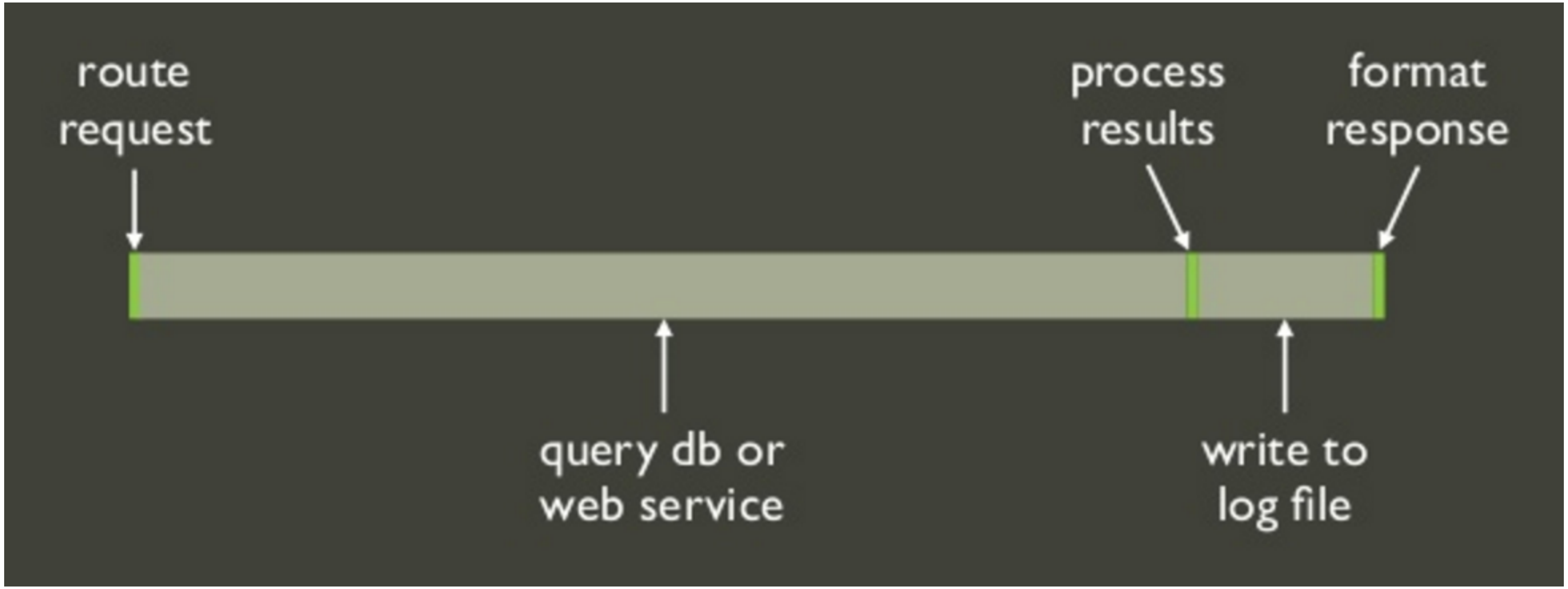
We already know PHP, why bother Node.js?

- Node.js has an event-driven, **non-blocking I/O** model
- lightweight and efficient
- good for data-intensive real-time applications, like games
- a single-threaded web server that handle concurrent requests well

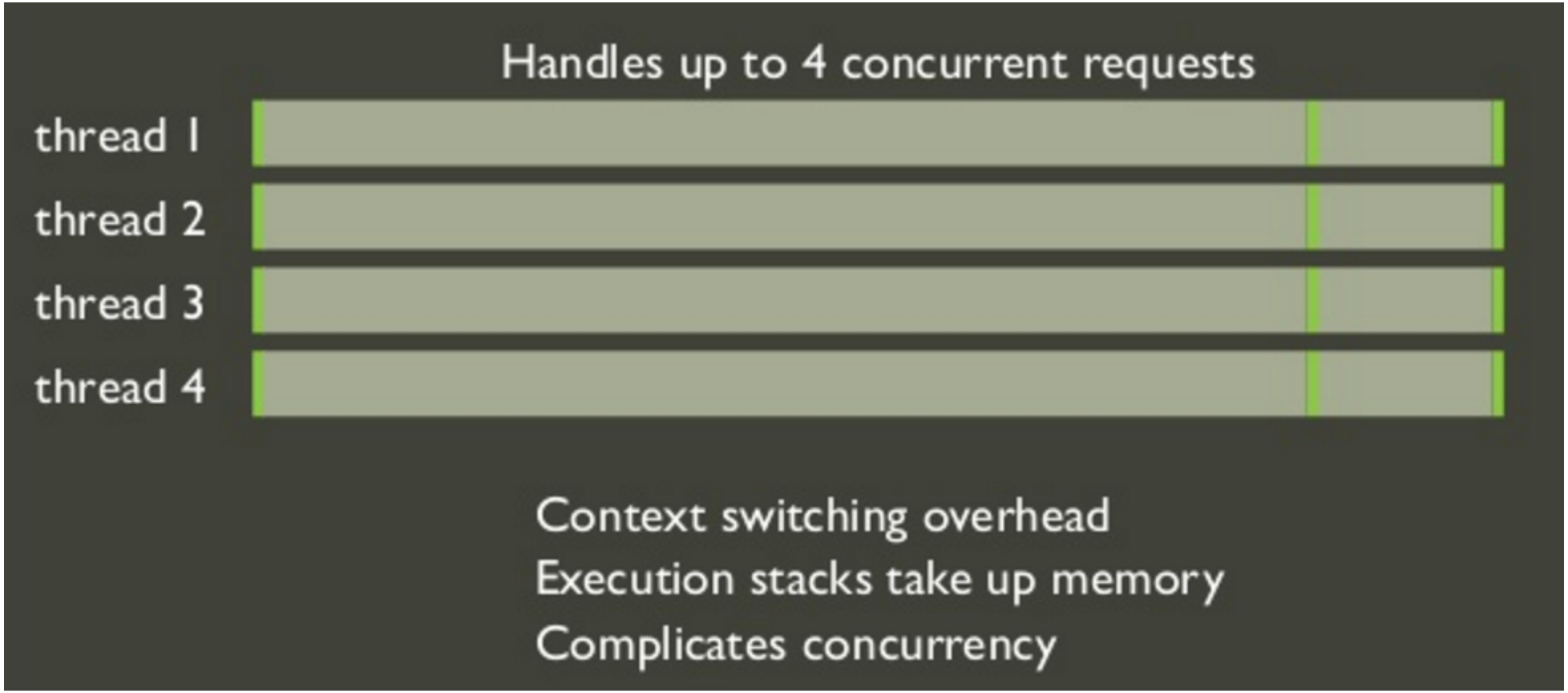
I/O takes time

- For a typical request handling at a web server, most of the time we are waiting for some I/O to complete.

- Waiting means blocking, i.e., you can do nothing else while waiting (assuming single thread)



Use threads! (e.g., Apache)



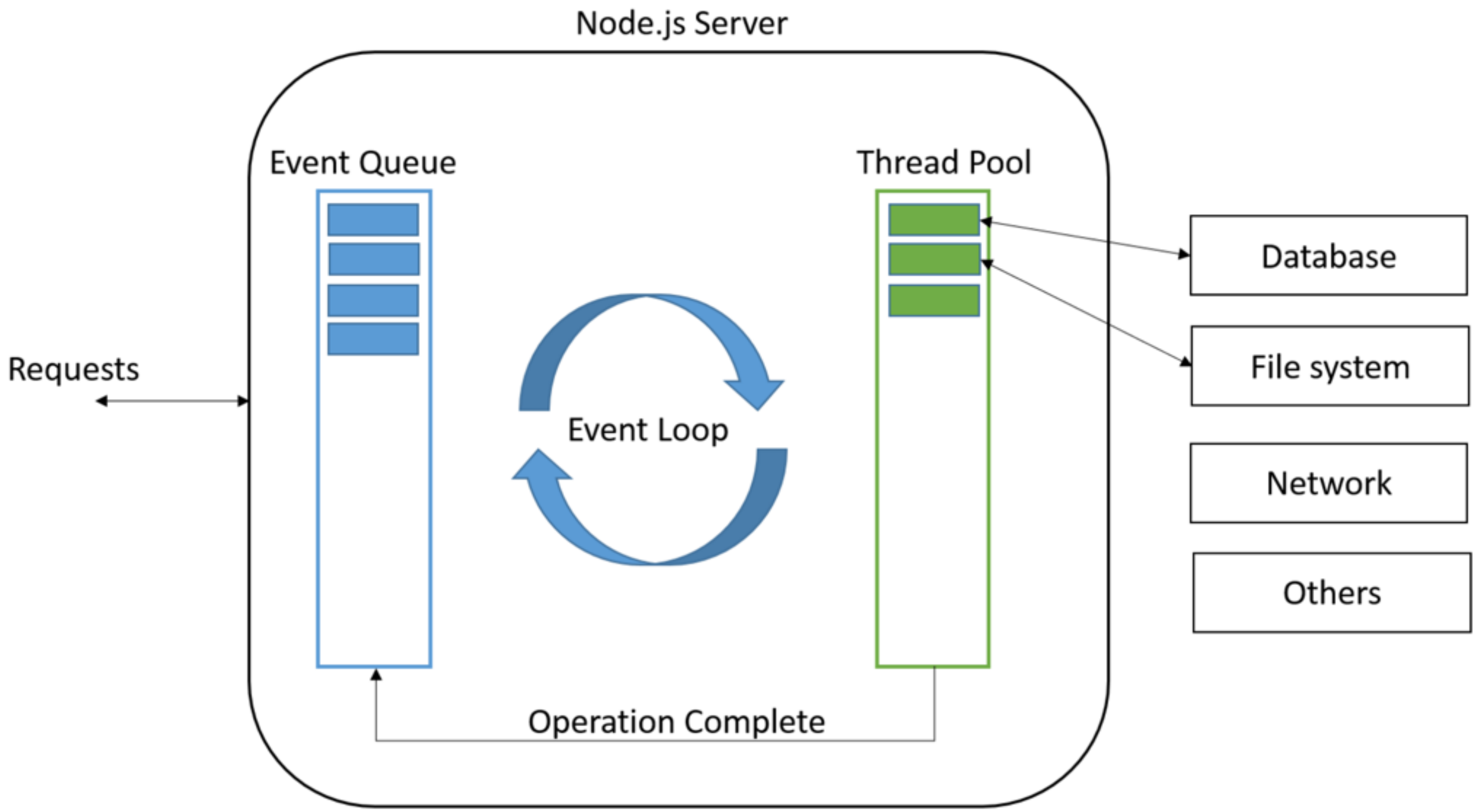
Context switching overhead
Execution stacks take up memory
Complicates concurrency

Some people, when confronted with a problem, think, “I know, I’ll use threads!” Now they have 10 problems.

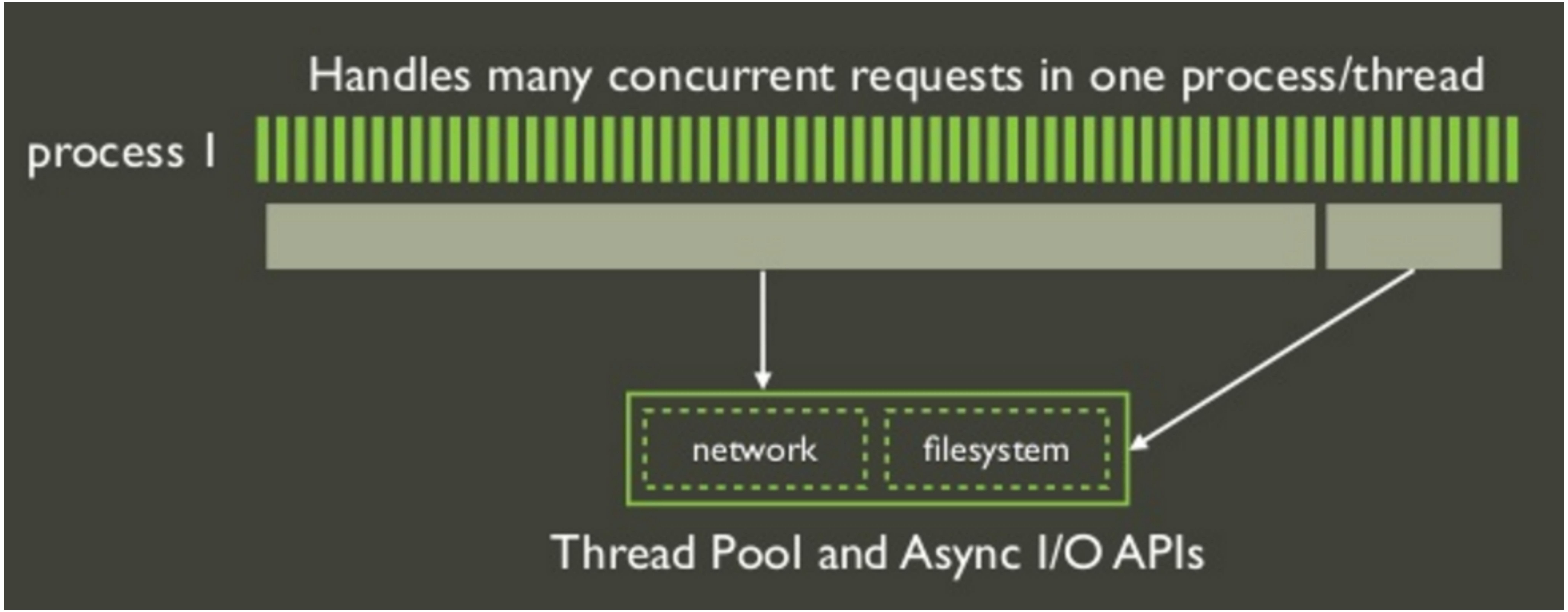
-- Bill Schindler

Node.js: use an Event Loop

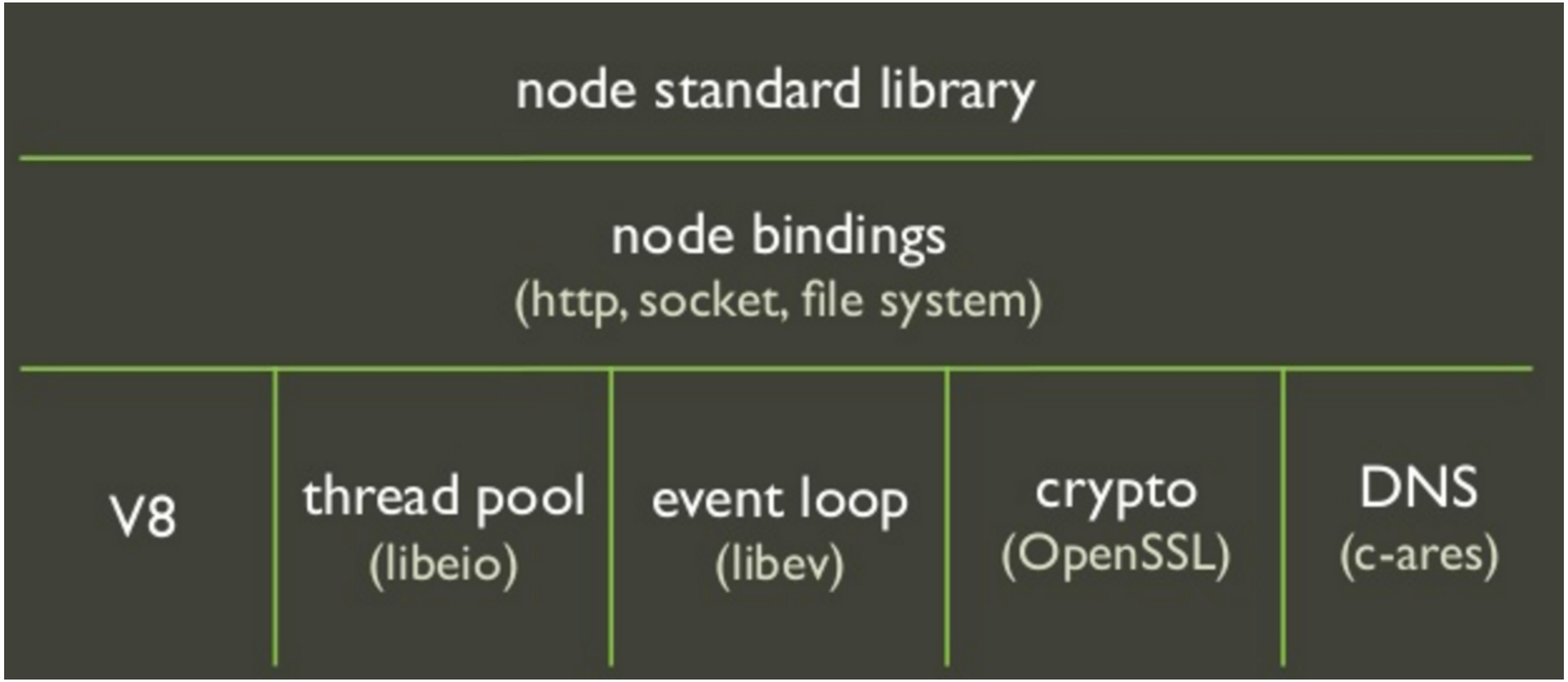
- Web server is **single thread**
- The main web server thread is never blocking
- The main thread keeps looping through the event loop to poll for completed operations, then execute the stored callback functions.
- I/O are handled by the thread pool



Node.js: use an Event Loop


Handles many concurrent requests in one process/thread
process 1
network
filesystem
Thread Pool and Async I/O APIs

Node.js platform



node standard library

node bindings
(http, socket, file system)

V8

thread pool
(libeio)

event loop
(libev)

crypto
(OpenSSL)

DNS
(c-ares)

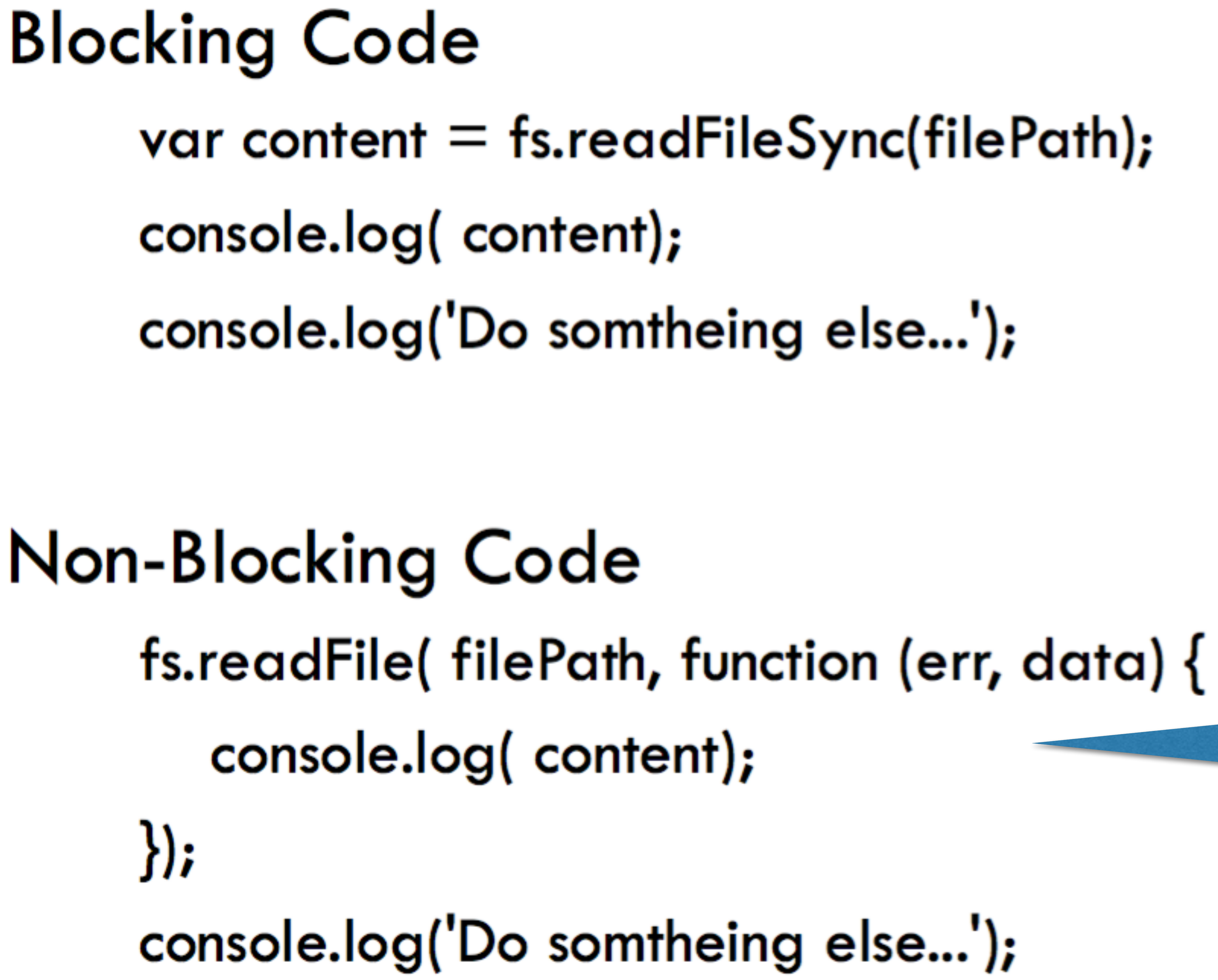
Benefits of Node.js

- Multiplatform: supports Windows, Mac, Linux
- Light-weight: since it's single threaded
- Easy concurrency: since it's single threaded
- JavaScript on both client and server, don't have switch minds back and forth
- Vibrant community, lots of packages available
- Fast, especially for data-intensive, real-time apps, like games, chatting, steaming, etc.

Node.js: Basic Syntax

It’s basically JavaScript

Blocking / Non-blocking code



Blocking Code

```
var content = fs.readFileSync(filePath);
console.log( content);
console.log('Do somtheing else...');
```

Non-Blocking Code

```
fs.readFile( filePath, function (err, data) {
    console.log( content);
});
console.log('Do somtheing else...');
```

In Node.js, do non-blocking code all the time for I/O.

Trivia

- There is no “sleep” function in JavaScript.

Creating a HTTP server

```
var http = require('http');
var server =
http.createServer(function(req, res) {
  res.writeHead(200);
  res.end('Hello World');
});
server.listen(8080);
```

Unlike PHP, we need to specify a listening port number for each Node.js server

Reference:
https://nodejs.org/en/

demo

http://www.cs.toronto.edu/~ylzhang/csc309w16/node/

npm: the Node.js package manager

- Lots of useful packages for Node.js, everything you can think of.
- To use a package, just do “npm install package-name”
- Reference:
  - https://www.npmjs.com/

Brief mention: Express.js

- A web framework for Node.js
- The native Node.js alone is quite low level, Express.js is a framework that makes it easy to create web applications.
- Often work together with a **template engine** (e.g., Jade, Mustache, EJS) to generate web pages with dynamic content.
- Native support of MVC
- Reference:
    - http://expressjs.com/
    - http://jade-lang.com/

Next week

- WebSocket