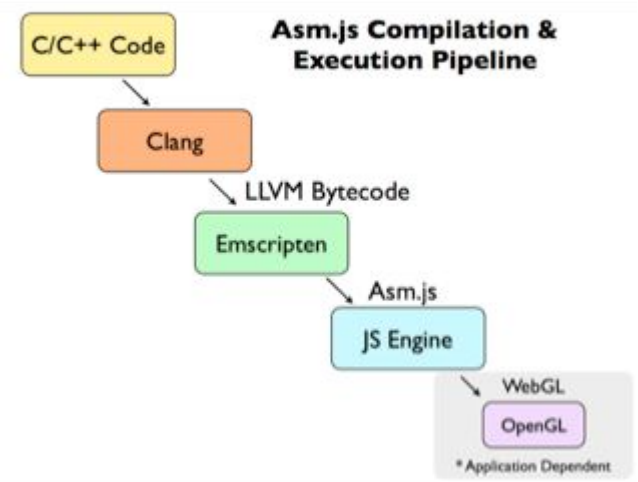
Introduction to Node.js

Red Babel.

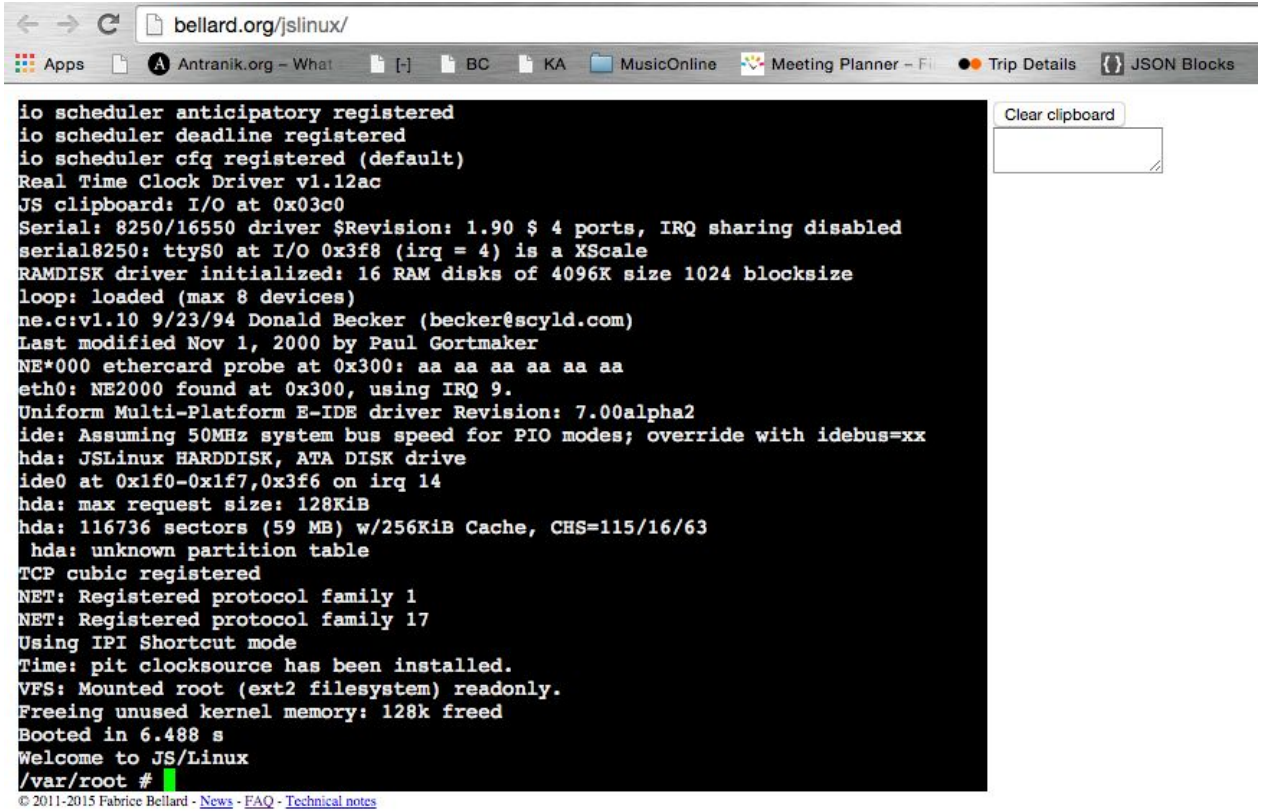
Atwood’s Law

“*any application that can be written in JavaScript, will eventually be written in JavaScript.*”

-- Jeff Atwood, Stack Overflow Co-founder. Source: http://blog.codinghorror.com/the-principle-of-least-power/



https://blog.mozilla.org/blog/2014/03/12/mozilla-and-epic-preview-unreal-engine-4-running-in-firefox/



http://bellard.org/jslinux/

ECMAScript 6

June 2015: http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf

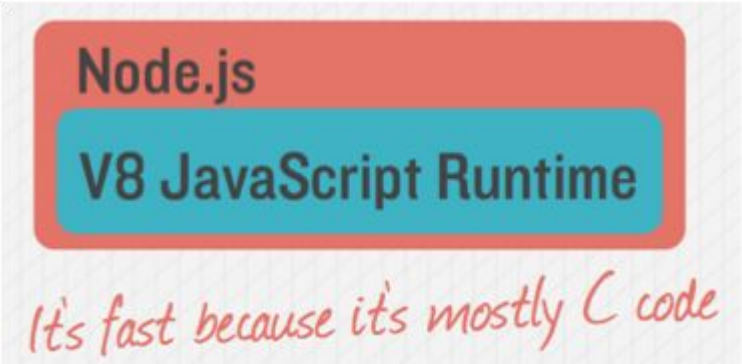
- **Fat arrows function**
- **Classes**
- **Destructuring**
- **Promises**
- **Modules**
- **Generators**

What is Node.js

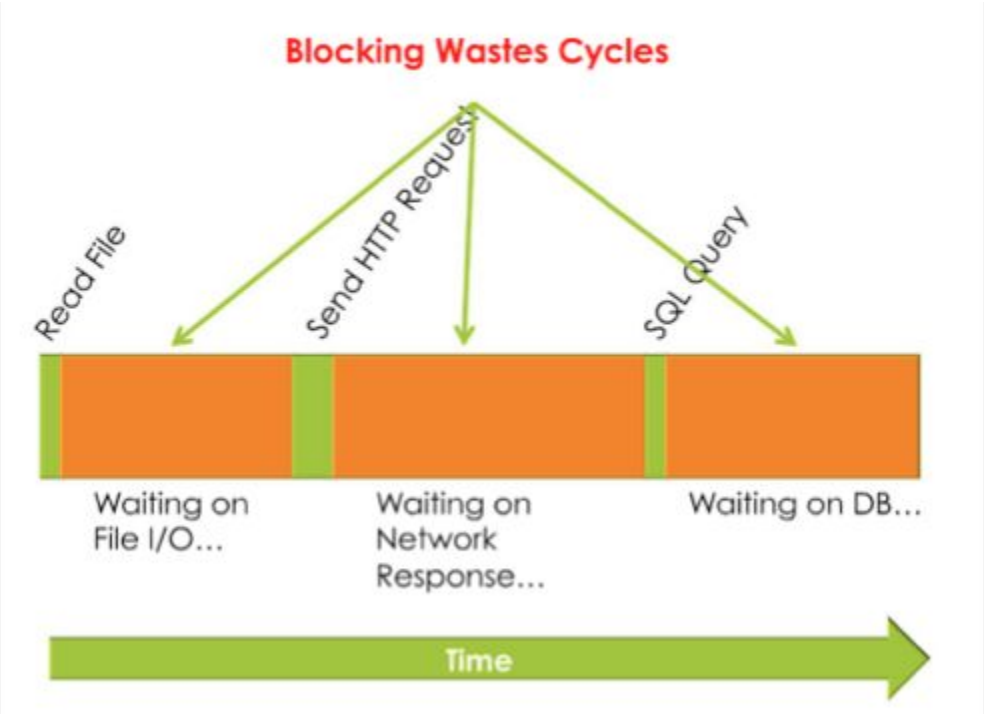
“Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Node.js uses an **event-driven**, **non-blocking I/O** model that makes it lightweight and efficient.”

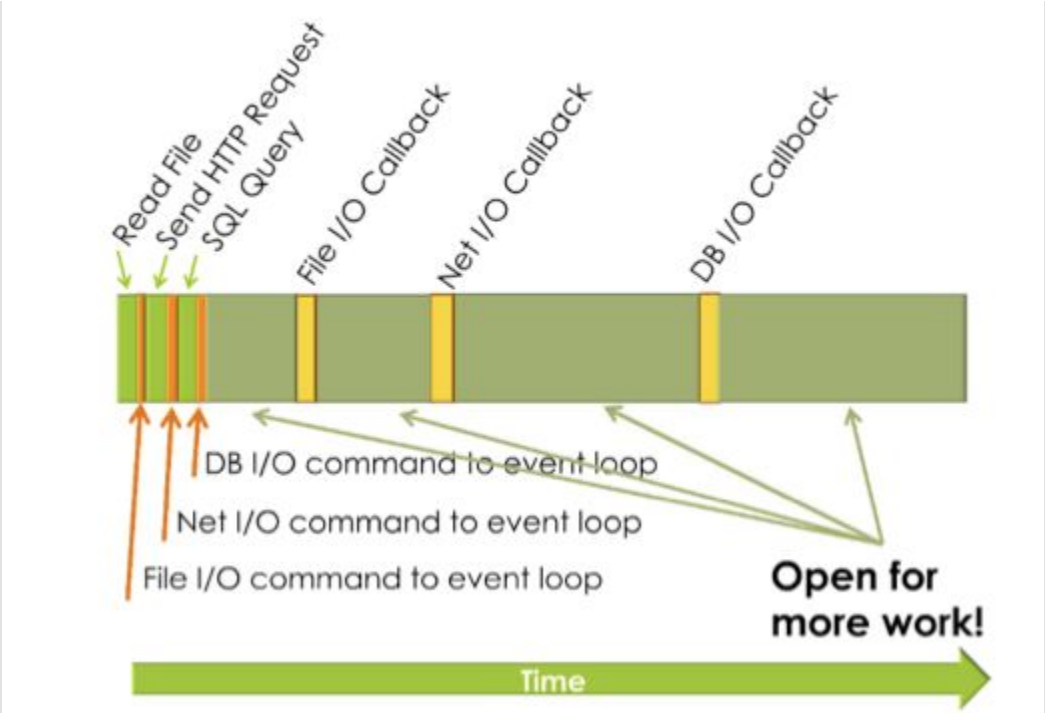
Allows you to build scalable network applications using JavaScript on the server-side.



Blocking


Blocking Wastes Cycles
Read File
Send HTTP Request
SQL Query
Waiting on File I/O…
Waiting on Network Response…
Waiting on DB…
Time

Non blocking



Node.js App

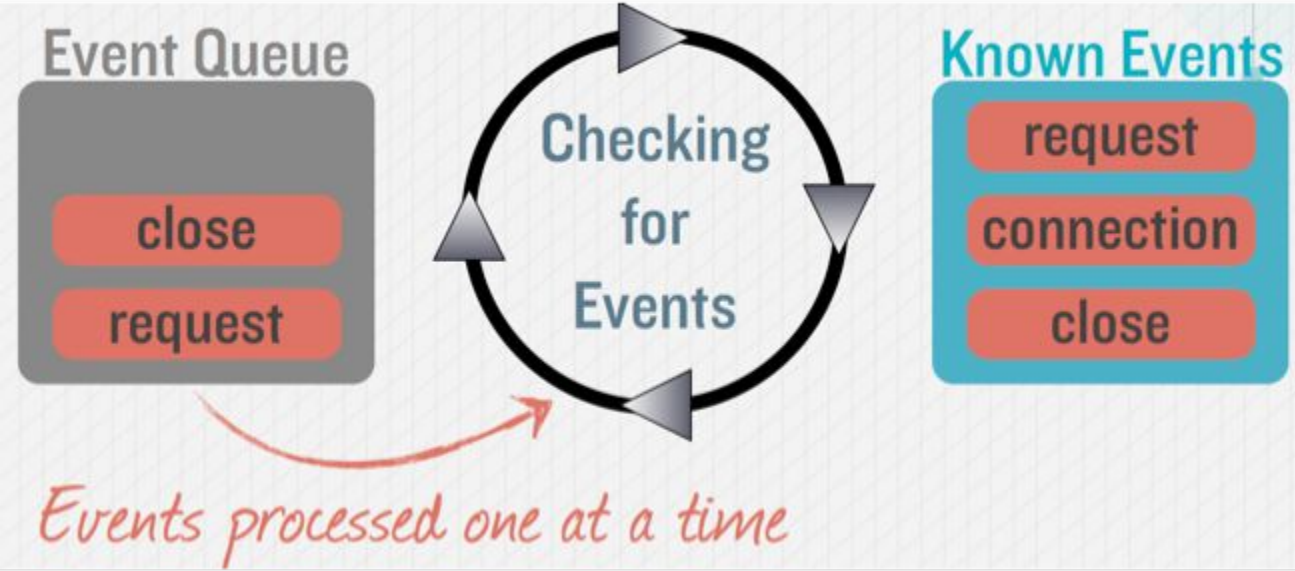
- Run entirely in a single thread
- Passes I/O requests to the event loop, along with callbacks

Your code then:

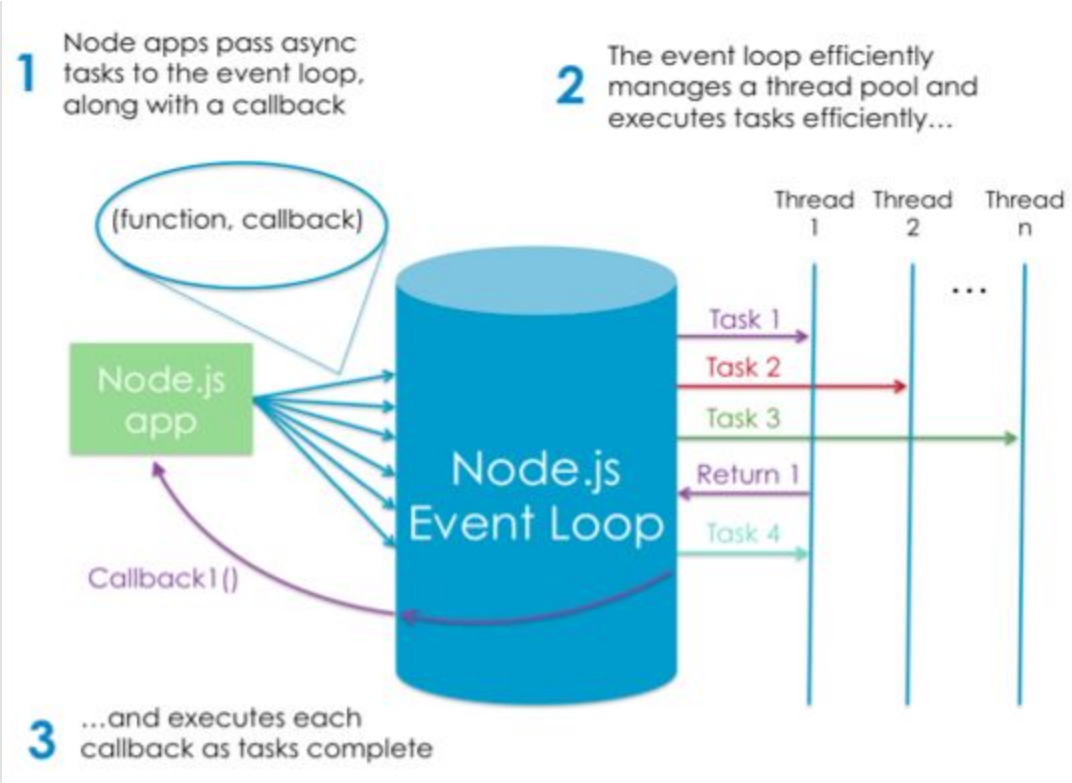
- Goes to sleep
- Uses no system resources

- Will be notified via callback when I/O is complete

Event Loop


Event Queue
close
request
Checking for Events
Known Events
request
connection
close
Events processed one at a time

Event Loop


1
Node apps pass async tasks to the event loop, along with a callback
2
The event loop efficiently manages a thread pool and executes tasks efficiently…
(function, callback)
Thread 1
Thread 2
Thread n
Node.js app
Node.js Event Loop
Task 1
Task 2
Task 3
Return 1
Task 4
Callback1()
3
…and executes each callback as tasks complete

Examples