



NodeJS Events & Event loop

PAVEL YUKHNOVICH

April 2018

Agenda

- Using callbacks with async operations
- EventEmitter & Events
- What are memory leaks and how to avoid it?
- Event loop in NodeJS

A large, bold, black 'JS' is centered on a bright yellow rectangular background that occupies the right half of the slide.

Javascript

- Single-threaded

A large, bold, black 'JS' is centered on a solid yellow background. The letters are in a sans-serif font, with the 'J' and 'S' being slightly larger than the 'S'.

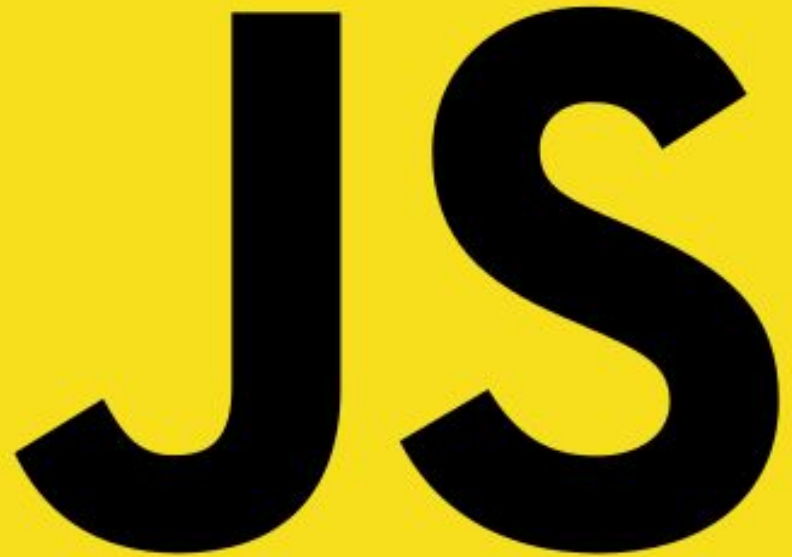
Javascript

- Single-threaded
- Asynchronous

A large, bold, black 'JS' is centered on a bright yellow rectangular background that occupies the right half of the slide.

Javascript

- Single-threaded
- Asynchronous
- Event-driven

A large, bold, black 'JS' is centered on a bright yellow rectangular background that occupies the right half of the slide.

Callback

```
function getLazySum(a, b, callback) {  
  setTimeout(() => {  
    callback(a + b);  
  }, Math.random() * 1000);  
}
```

```
function logMessage(message) {  
  console.log(message);  
}
```

```
getLazySum(1, 2, logMessage);
```

Callback

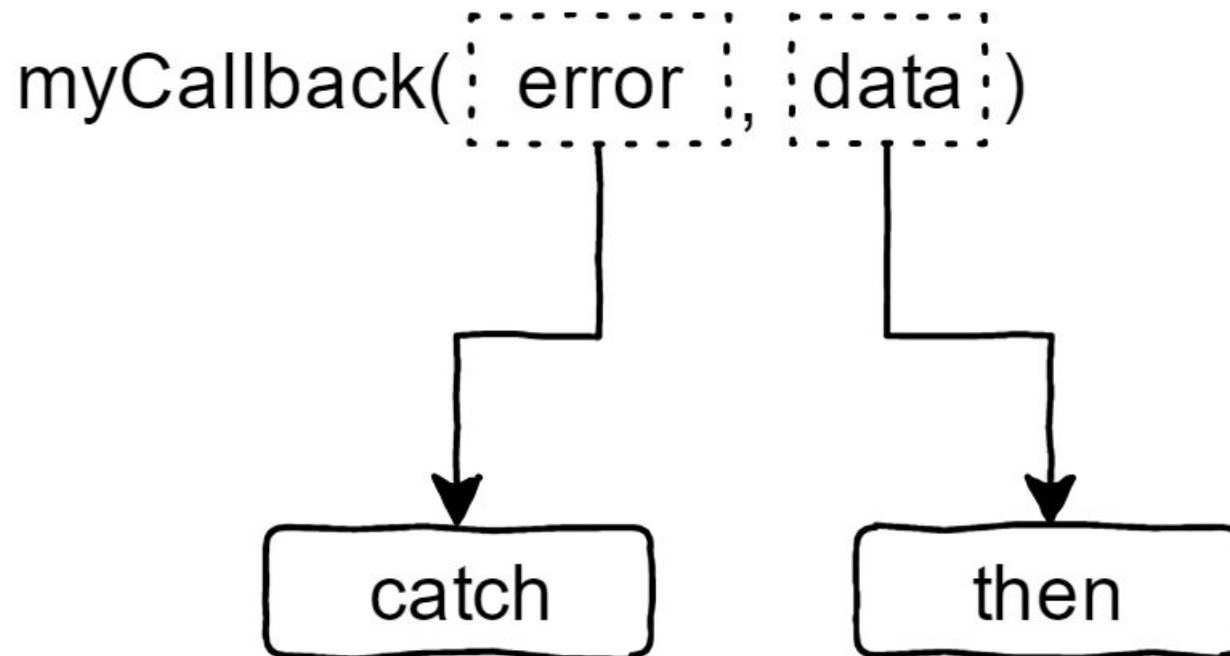
- `fs.open(path, flags, mode, callback)`
- `agent.createConnection(options, callback)`
- `server.listen(handle, callback)`
- `server.listen(path, callback)`
- `dns.resolve(hostname, rrtype, callback)`
- `fs.readFile(file, callback)`

Error-first callback

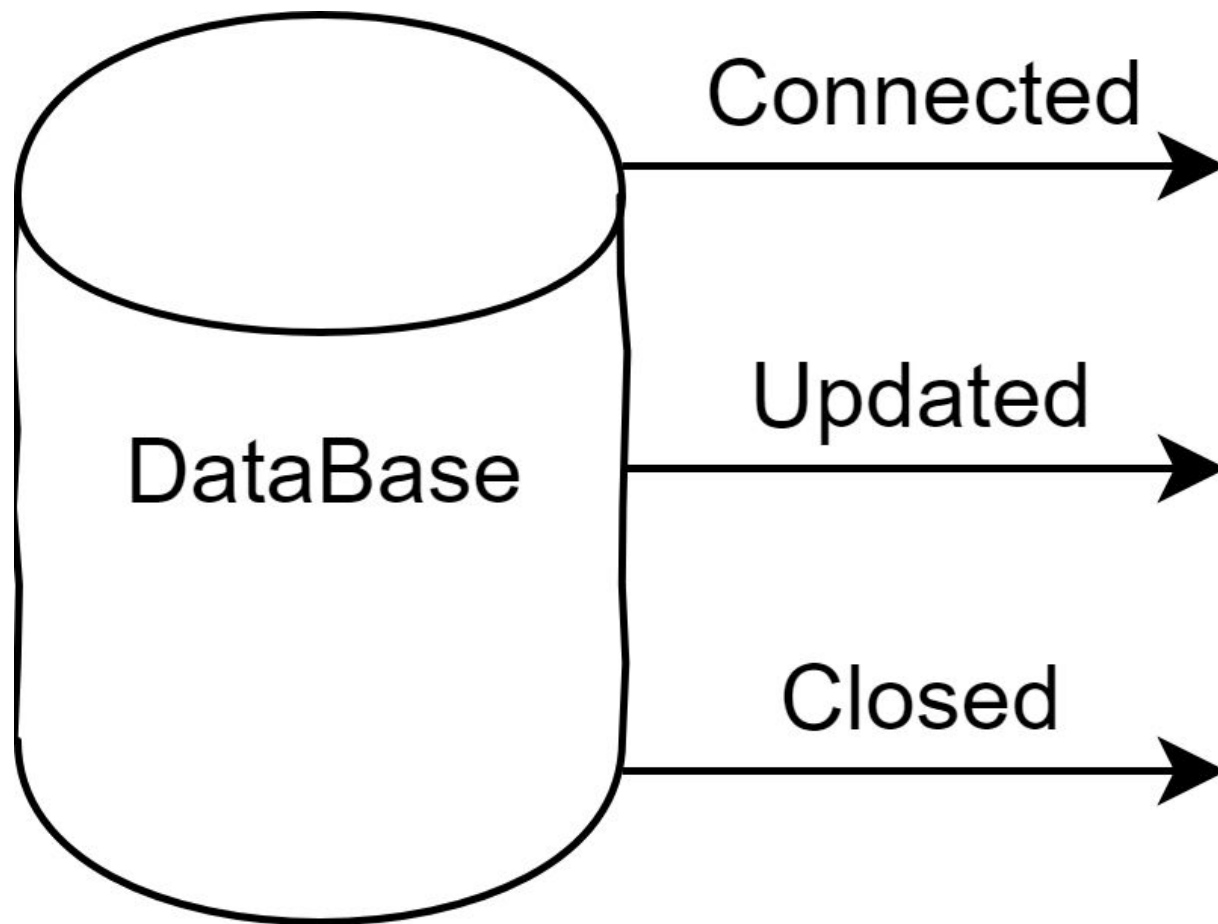
```
1 fs.readFile('/foo.txt', function (error, data) {  
2   |   if (error) { throw error; }  
3   |   console.log(data);  
4   | });
```

Error-first callback

```
1 fs.readFile('/foo.txt', function (error, data) {  
2   if (error) { throw error; }  
3   console.log(data);  
4 });
```



Events



EventEmitter

- **EventEmitter class in events**
- **pub/sub**

EventEmitter

- on
- once
- prependListener
- prependOnceListener
- emit
- removeAllListeners
- removeListener

EventEmitter

- Event: 'newListener'
- Event: 'removeListener'
- EventEmitter.listenerCount(emitter, eventName)
- EventEmitter.defaultMaxListeners
- emitter.addListener(eventName, listener)
- emitter.emit(eventName[, ...args])
- emitter.eventNames()
- emitter.getMaxListeners()
- emitter.listenerCount(eventName)
- emitter.listeners(eventName)
- emitter.on(eventName, listener)
- emitter.once(eventName, listener)
- emitter.prependListener(eventName, listener)
- emitter.prependOnceListener(eventName, listener)
- emitter.removeAllListeners([eventName])
- emitter.removeListener(eventName, listener)
- emitter.setMaxListeners(n)
- emitter.rawListeners(eventName)

EventEmitter

```
const EventEmitter = require('events');
```

```
class MyEmitter extends EventEmitter {};
```

```
const myEmitter = new MyEmitter();
```

>

```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();
```

```
const alerter = () => console.log('event happened!');
```

```
myEmitter.on('event', alerter);
```

```
myEmitter.emit('event');
```

event happened!


```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();  
  
const alerter = () => console.log('event happened!');  
  
myEmitter.on('event', alerter);  
  
myEmitter.emit('event');  
myEmitter.emit('event');  
myEmitter.emit('event');  
myEmitter.emit('event');
```

event happened!

event happened!

event happened!

event happened!

```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();
```

```
const alerter = () => console.log('event happened!');
```

```
myEmitter.once('event', alerter);
```

```
myEmitter.emit('event');  
myEmitter.emit('event');  
myEmitter.emit('event');  
myEmitter.emit('event');
```

event happened!

EventEmitter .addListener(eventName,listener)

```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();  
  
const alerter = () => console.log('event happened!');  
const helloer = () => console.log('hello there!');  
  
myEmitter.on('event', alerter);  
myEmitter.addListener('event', helloer);  
  
myEmitter.emit('event');
```

event happened!

hello there!

EventEmitter async vs sync

```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();  
  
const alerter = () => console.log('event happened!');  
const helloer = () => console.log('hello there!');  
  
myEmitter.on('event', alerter);  
myEmitter.addListener('event', helloer);  
  
myEmitter.emit('event');
```

event happened!

hello there!

EventEmitter async vs sync

```
const alerter = () => console.log("event1 happened!");  
const alerter2 = () => console.log("event2 happened!");
```

```
myEmitter.addListener("event", alerter);  
myEmitter.addListener("event2", alerter2);
```

```
myEmitter.on("event3", (a, b) => {  
  setTimeout(() => {  
    console.log("async event happened!");  
  }, 5000);  
});
```

```
myEmitter.emit("event");  
myEmitter.emit("event3", "a", "b");  
myEmitter.emit("event2");
```

event1 happened!

event2 happened!

async event happened!

EventEmitter .addListener(eventName,listener)

MAX LISTENERS NUM = 10

```
const alerter = () => console.log('event happened!');
const helloer = () => console.log('hello there!');
const byebyeer = () => console.log('bye bye');
const alerter2 = () => console.log('event happened!');
const helloer2 = () => console.log('hello there!');
const byebyeer2 = () => console.log('bye bye');
const alerter3 = () => console.log('event happened!');
const helloer3 = () => console.log('hello there!');
const byebyeer3 = () => console.log('bye bye');
const alerter4 = () => console.log('event happened!');
const helloer4 = () => console.log('hello there!');
const byebyeer4 = () => console.log('bye bye');
```

```
myEmitter.addListener('event', alerter);
myEmitter.addListener('event', alerter2);
myEmitter.addListener('event', alerter3);
myEmitter.addListener('event', alerter4);
myEmitter.addListener('event', helloer);
myEmitter.addListener('event', helloer2);
myEmitter.addListener('event', helloer3);
myEmitter.addListener('event', helloer4);
myEmitter.addListener('event', byebyeer);
myEmitter.addListener('event', byebyeer2);
myEmitter.addListener('event', byebyeer3);
myEmitter.addListener('event', byebyeer4);
```

```
myEmitter.emit('event');
```

```
event happened!
event happened!
event happened!
event happened!
hello there!
hello there!
hello there!
hello there!
bye bye
bye bye
bye bye
bye bye
```

(node:6156)

MaxListenersExceededWarning:
Possible EventEmitter memory
leak detected. 11 event listeners
added. Use
emitter.setMaxListeners() to
increase limit

EventEmitter .setMaxListeners(num)

```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();  
myEmitter.setMaxListeners(20);
```

```
const alerter = () => console.log('event happened!');  
const helloer = () => console.log('hello there!');  
const byebyeer = () => console.log('bye bye');  
const alerter2 = () => console.log('event happened!');  
const helloer2 = () => console.log('hello there!');  
const byebyeer2 = () => console.log('bye bye');  
const alerter3 = () => console.log('event happened!');  
const helloer3 = () => console.log('hello there!');  
const byebyeer3 = () => console.log('bye bye');  
const alerter4 = () => console.log('event happened!');  
const helloer4 = () => console.log('hello there!');  
const byebyeer4 = () => console.log('bye bye');  
myEmitter.addListener('event', alerter);  
myEmitter.addListener('event', alerter2);  
myEmitter.addListener('event', alerter3);  
myEmitter.addListener('event', alerter4);  
myEmitter.addListener('event', helloer);  
myEmitter.addListener('event', helloer2);  
myEmitter.addListener('event', helloer3);  
myEmitter.addListener('event', helloer4);  
myEmitter.addListener('event', byebyeer);  
myEmitter.addListener('event', byebyeer2);  
myEmitter.addListener('event', byebyeer3);  
myEmitter.addListener('event', byebyeer4);  
  
myEmitter.emit('event');
```

```
event happened!  
event happened!  
event happened!  
event happened!  
hello there!  
hello there!  
hello there!  
hello there!  
bye bye  
bye bye  
bye bye  
bye bye
```

EventEmitter .getMaxListeners()

```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();  
  
...  
  
console.log(myEmitter.getMaxListeners());
```

20

EventEmitter .listenerCount(eventName)

```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();  
  
...  
  
console.log(myEmitter.listenerCount('event'));
```

12

EventEmitter .eventNames()

```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();  
  
const alerter = () => console.log('event happened!');  
const alerter2 = () => console.log('event happened 2!');  
  
myEmitter.addListener('event', alerter);  
myEmitter.addListener('event2', alerter2);  
  
myEmitter.emit('event');  
myEmitter.emit('event2');  
  
console.log(myEmitter.eventNames());
```

event happened!
event happened 2!

['event', 'event2']

EventEmitter .prependListener(evName,listener)

```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();  
  
const alerter = () => console.log('event happened!');  
const helloer = () => console.log('hello there!');  
const byebyeer = () => console.log('bye bye');  
  
myEmitter.on('event', alerter);  
myEmitter.addListener('event', helloer);  
myEmitter.prependListener('event', byebyeer);  
  
myEmitter.emit('event');
```

bye bye!

event happened!

hello there!

EventEmitter .prependOnceListener

```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();  
  
const alerter = () => console.log('event happened!');  
const helloer = () => console.log('hello there!');  
const byebyeer = () => console.log('bye bye');  
  
myEmitter.on('event', alerter);  
myEmitter.addListener('event', helloer);  
myEmitter.prependOnceListener('event', byebyeer);  
  
myEmitter.emit('event');  
  
myEmitter.emit('event');
```

bye bye!

event happened!

hello there!

event happened!

hello there!

```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();  
  
const alerter = () => console.log('event happened!');  
  
myEmitter.on('event', alerter);  
  
myEmitter.emit('event');  
myEmitter.emit('event');  
myEmitter.removeListener('event', alerter);  
myEmitter.emit('event');
```

event happened!

event happened!

```
const EventEmitter = require('events');  
class MyEmitter extends EventEmitter {}  
const myEmitter = new MyEmitter();
```

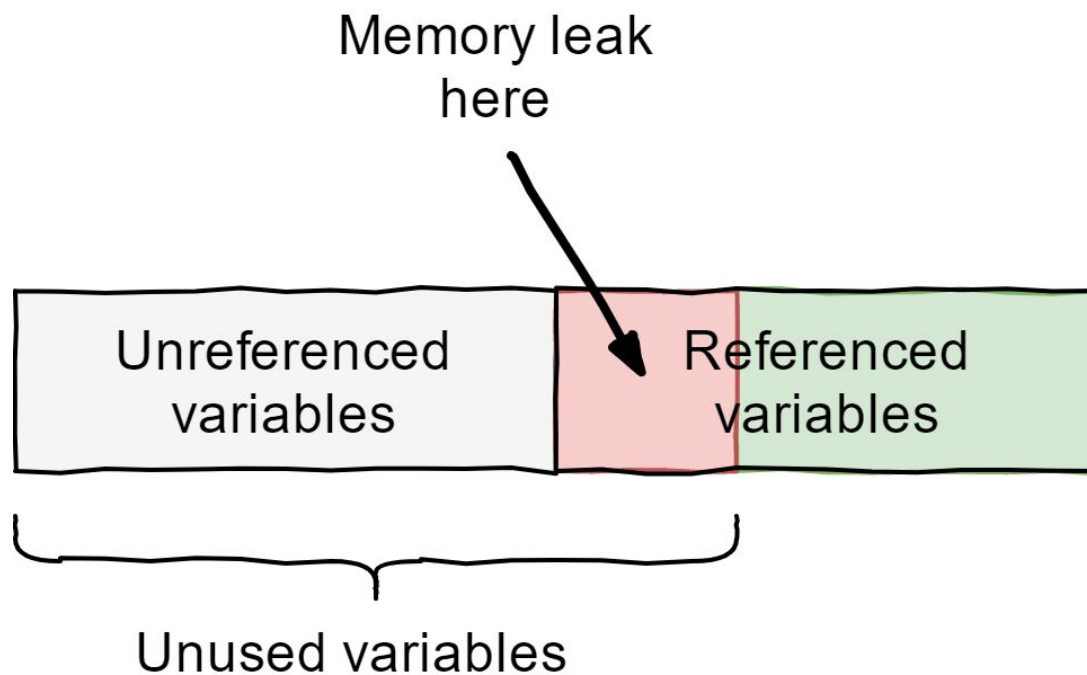
```
const alerter = () => console.log('event happened!');  
const helloer = () => console.log('hello there!');
```

```
myEmitter.on('event', alerter);  
myEmitter.emit('event');  
myEmitter.removeListener('event', alerter);  
myEmitter.addListener('event', helloer);  
myEmitter.emit('event');
```

event happened!

hello there!

Memory leaks



EventEmitters Zoo



EventEmitter

740 packages found

EventEmitters Zoo

- More methods
- Fast & optimized
- Lightweight
- Additional stuff

eventemitter2

- faster version of standard Event Emitter
- additional methods
- est. 7 years ago
- version 5.0.1
- > 1 mln downloads last week
- 945 dependents

E.g. `.many(eventName, times, listener)`
 `.onAny(listener)`

```
server.many('event', 4, () => {  
  console.log('event happened');  
});
```

```
server.onAny((event, value) => {  
  console.log('All events trigger this.');
```

eventemitter3

- faster version of standard Event Emitter
 - lightweight - a lot of functionality was removed
 - est. 2 years ago
 - 3.0.1 version
 - 1066 dependents
 - > 3 mln downloads last week
-
- **NO** `setMaxListeners`, `getMaxListeners`, `prependListener` and `prependOnceListener`
 - `removeListener` method removes all matching listeners, not only the first

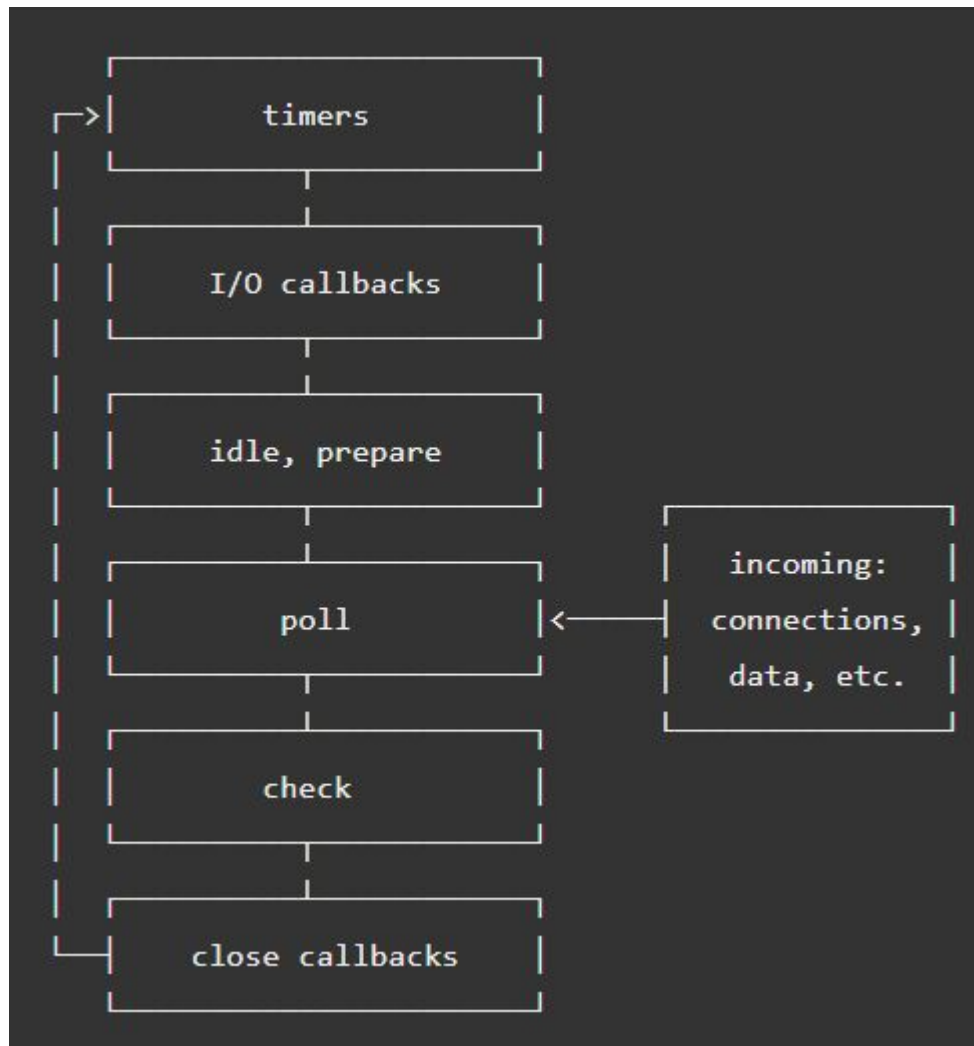
eventemitter5, eventemitter6,

- just wrappers
- doing completely nothing
- est. 1-2 years ago
- > 2 downloads last week

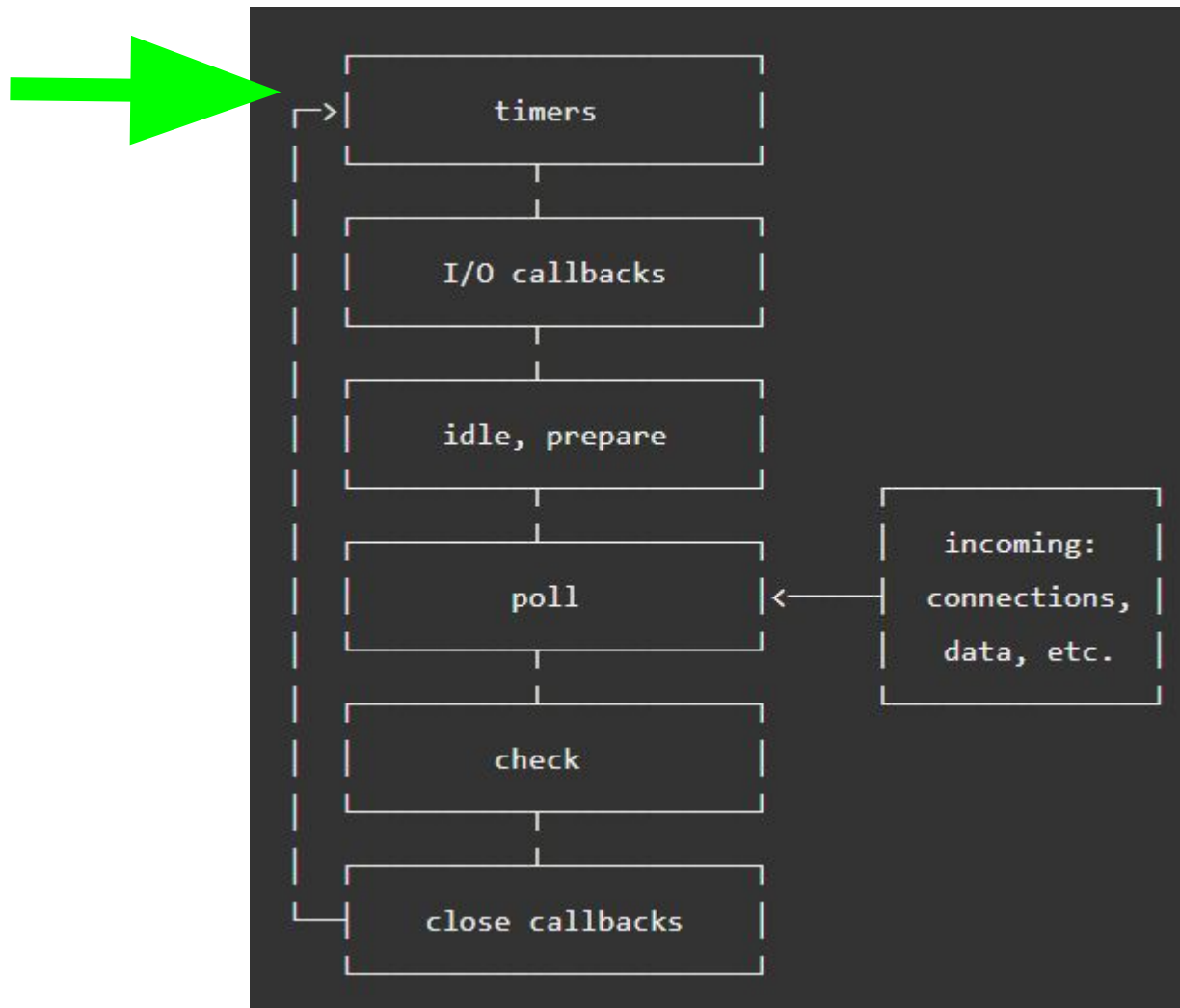
Event Loop



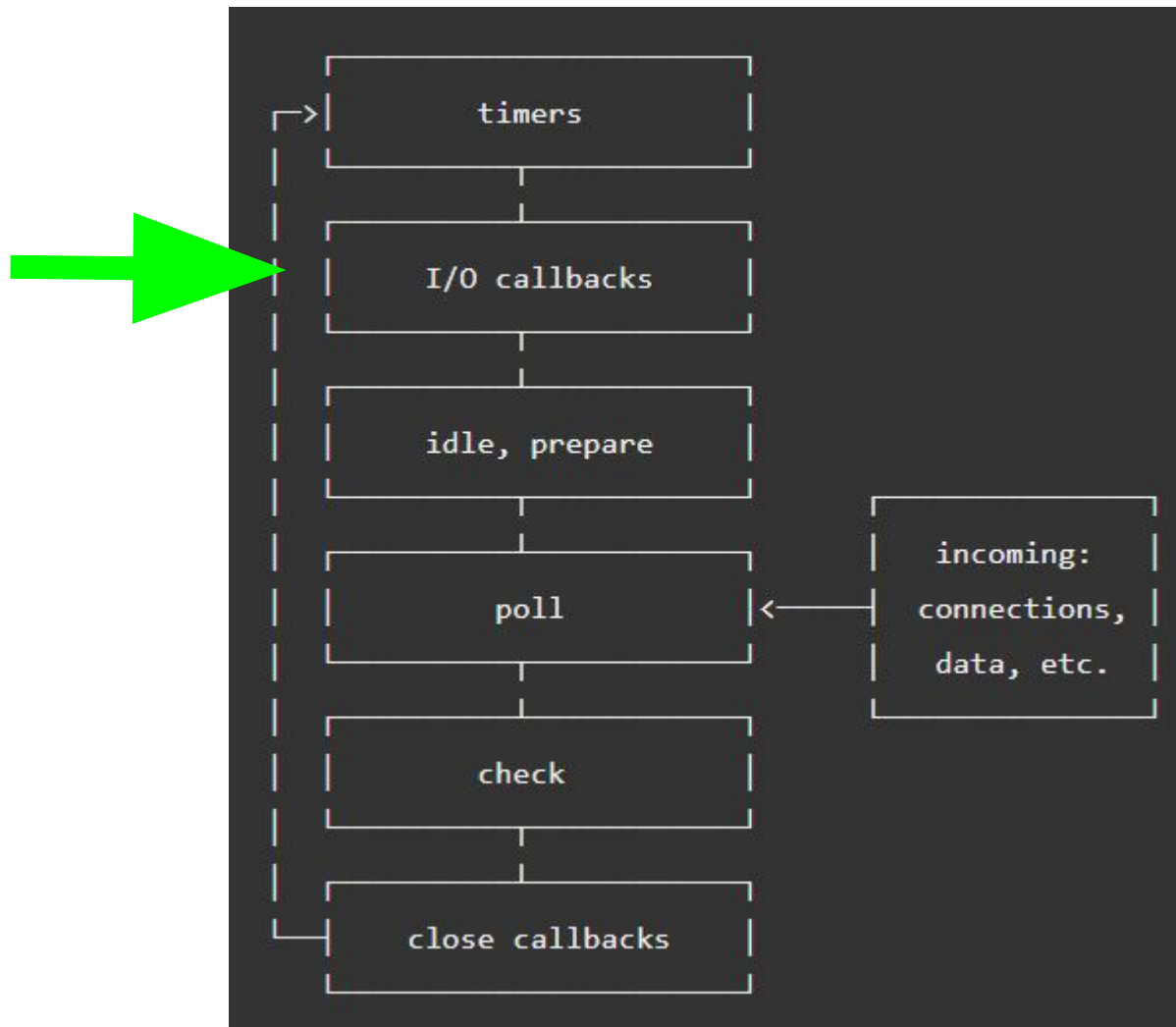
Event loop



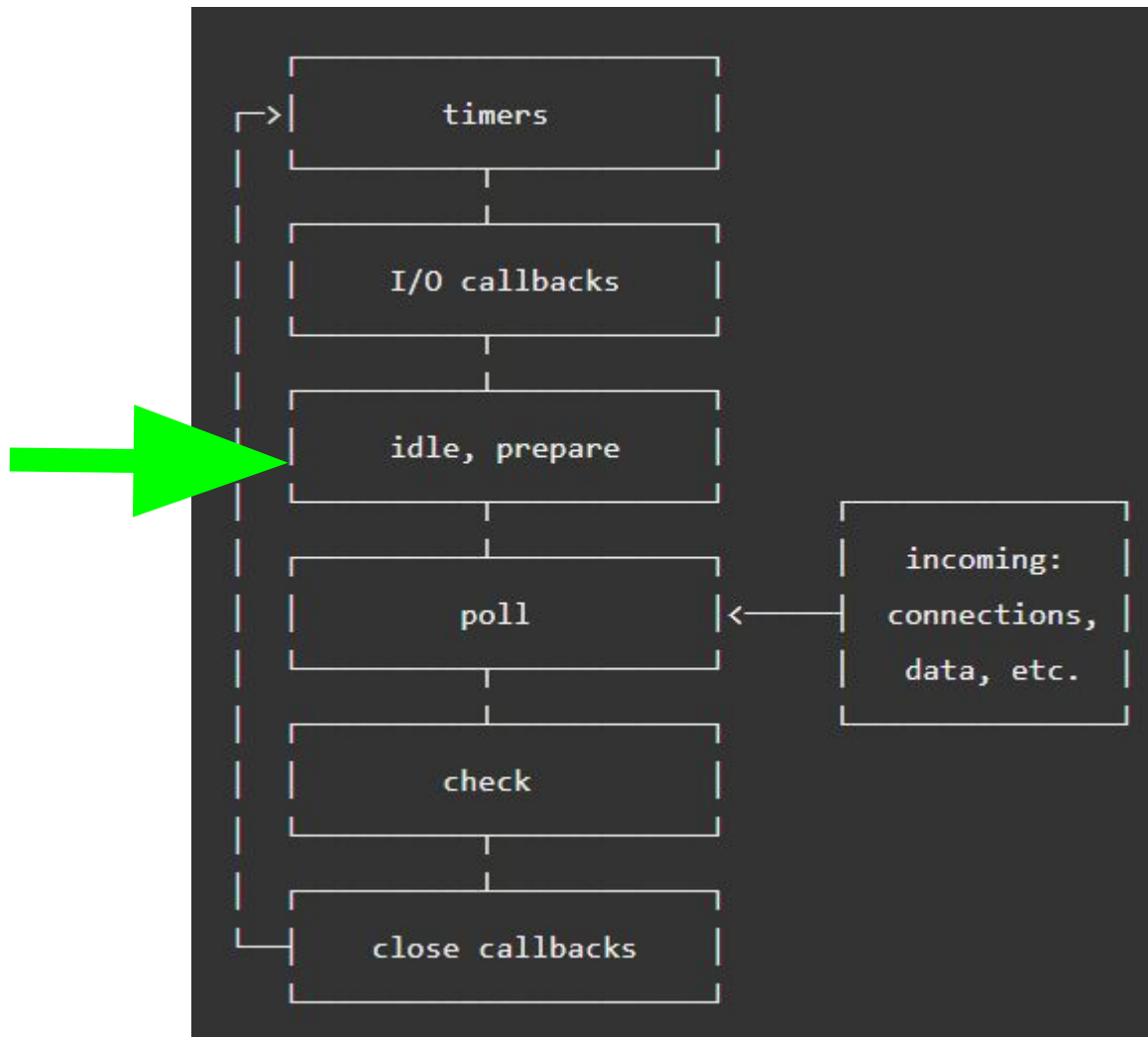
Event loop



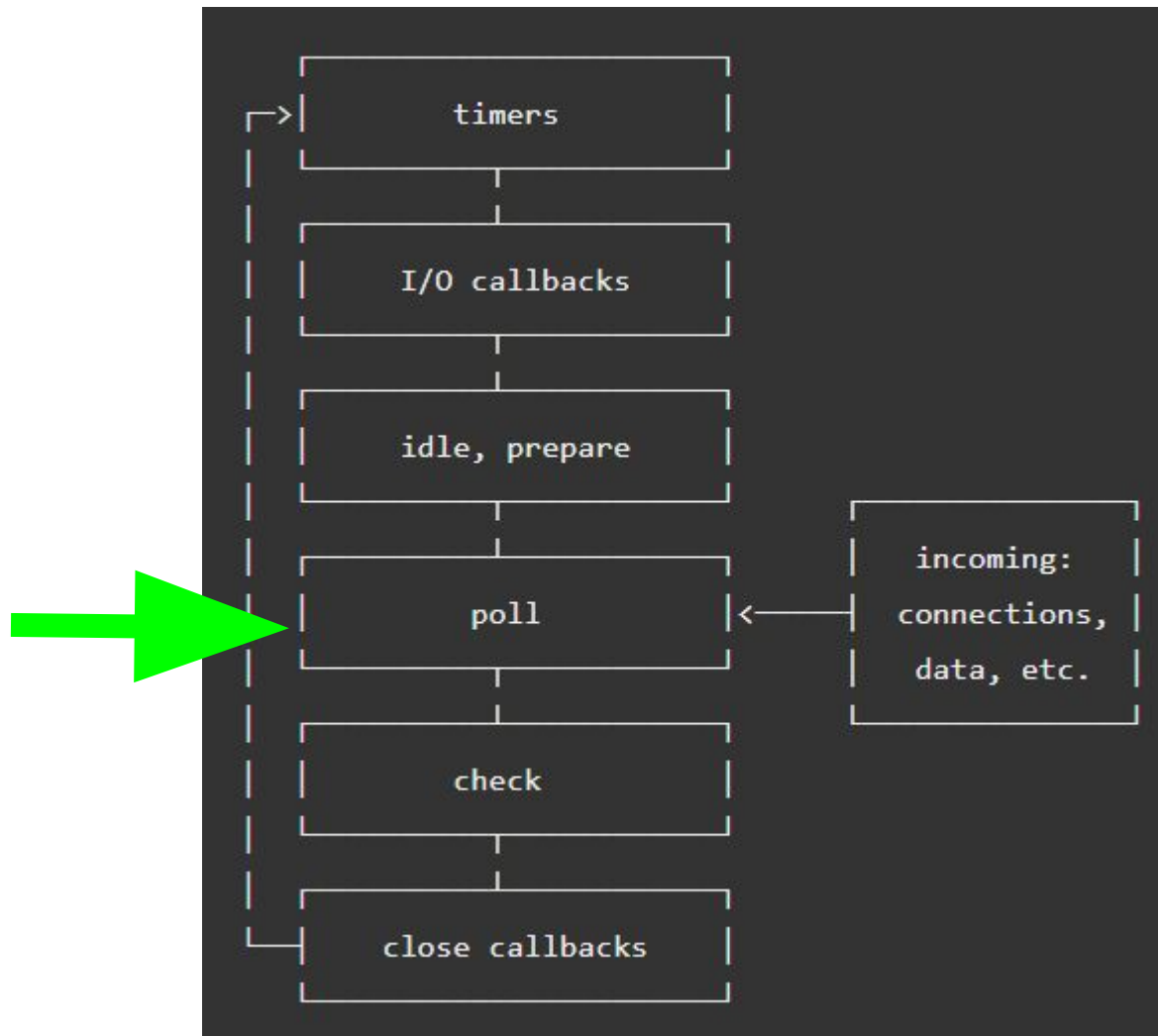
Event loop



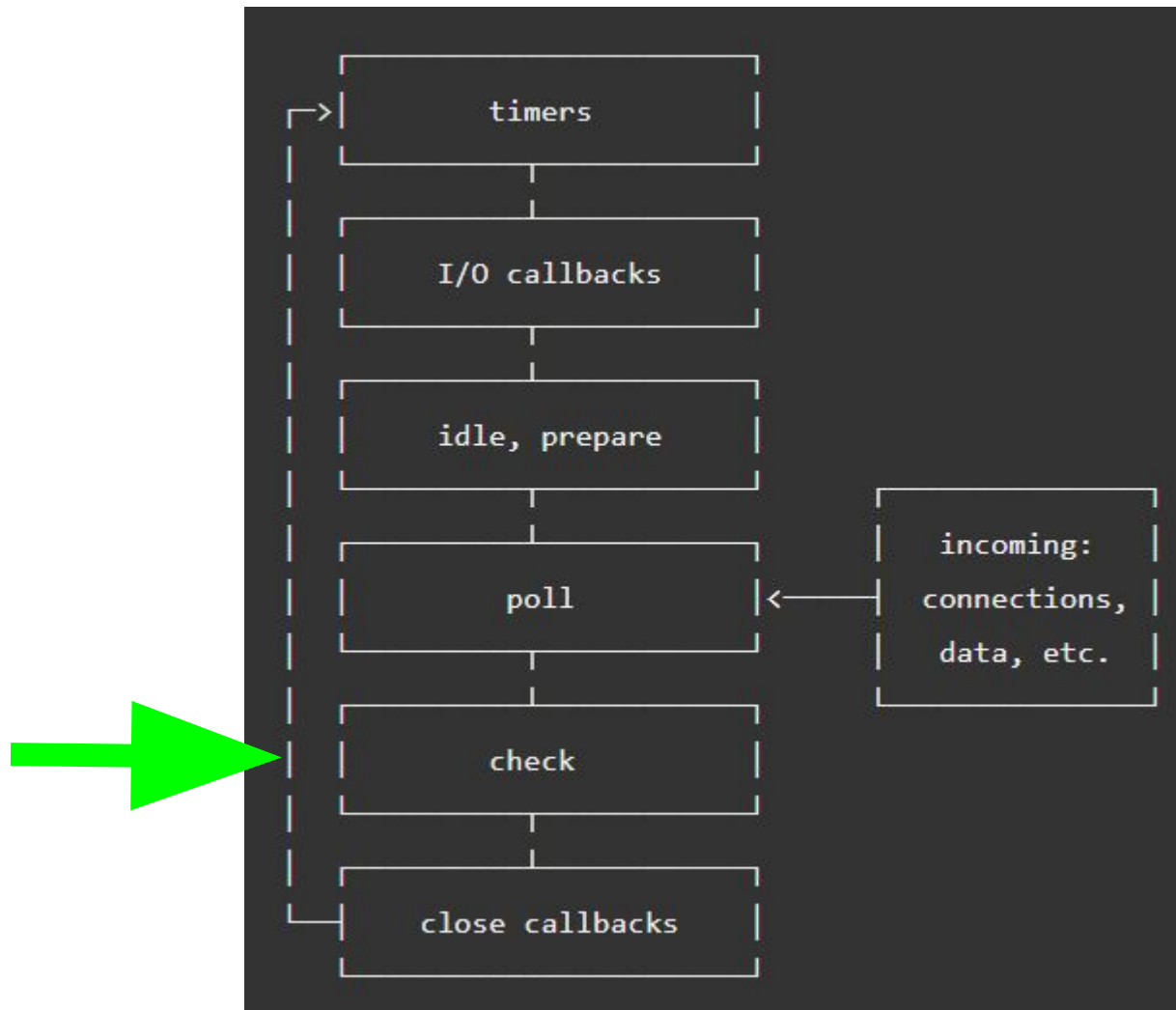
Event loop



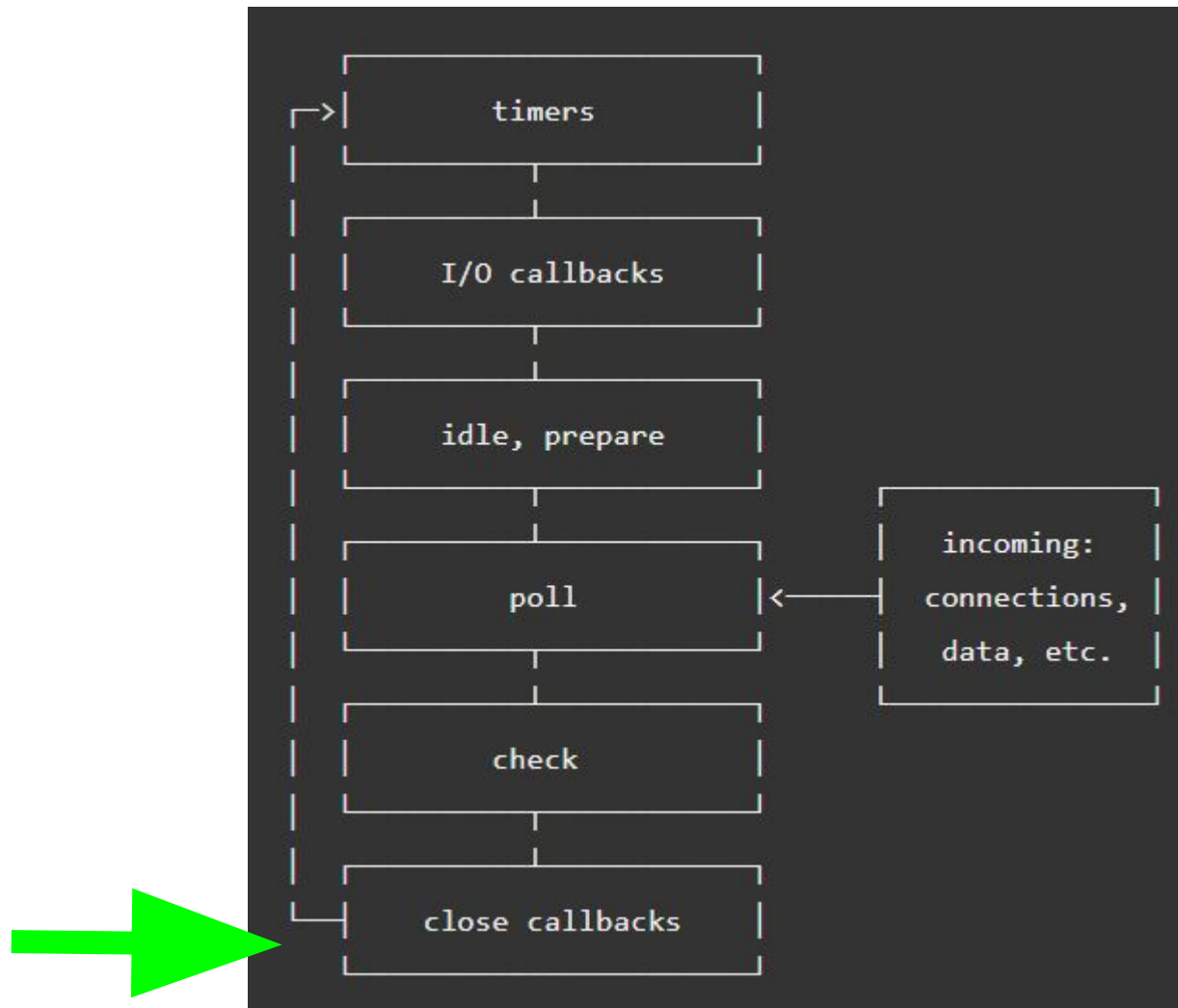
Event loop



Event loop



Event loop



setImmediate() vs setTimeout()

setImmediate()

designed to execute a script once the current **poll** phase completes.

setTimeout()

schedules a script to be run after a minimum threshold in ms has elapsed.

setImmediate() vs setTimeout()

not within an I/O cycle:

```
setTimeout(() => {  
  console.log("timeout");  
}, 0);
```

```
setImmediate(() => {  
  console.log("immediate");  
});
```

timeout
immediate

immediate
timeout

immediate
timeout

timeout
immediate

setImmediate() vs setTimeout()

within an I/O cycle:

```
const fs = require("fs");

fs.readFile(__filename, () => {
  setTimeout(() => {
    console.log("timeout");
  }, 0);
  setImmediate(() => {
    console.log("immediate");
  });
});
```

immediate
timeout

immediate
timeout



LibUV

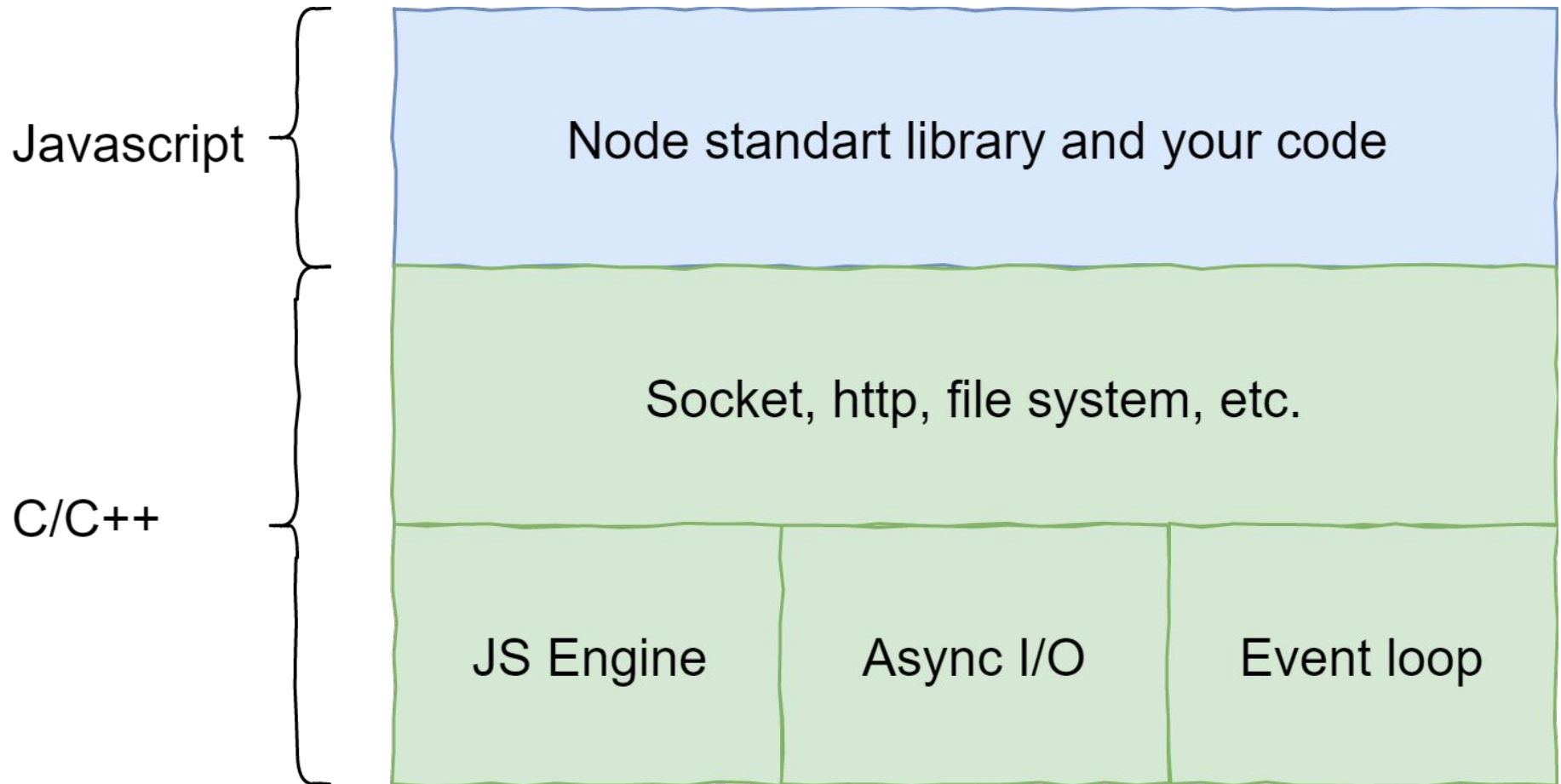
LibUV

Cross-platform support library which was originally written for NodeJS, but it's also used by pyUv, Luvit and others

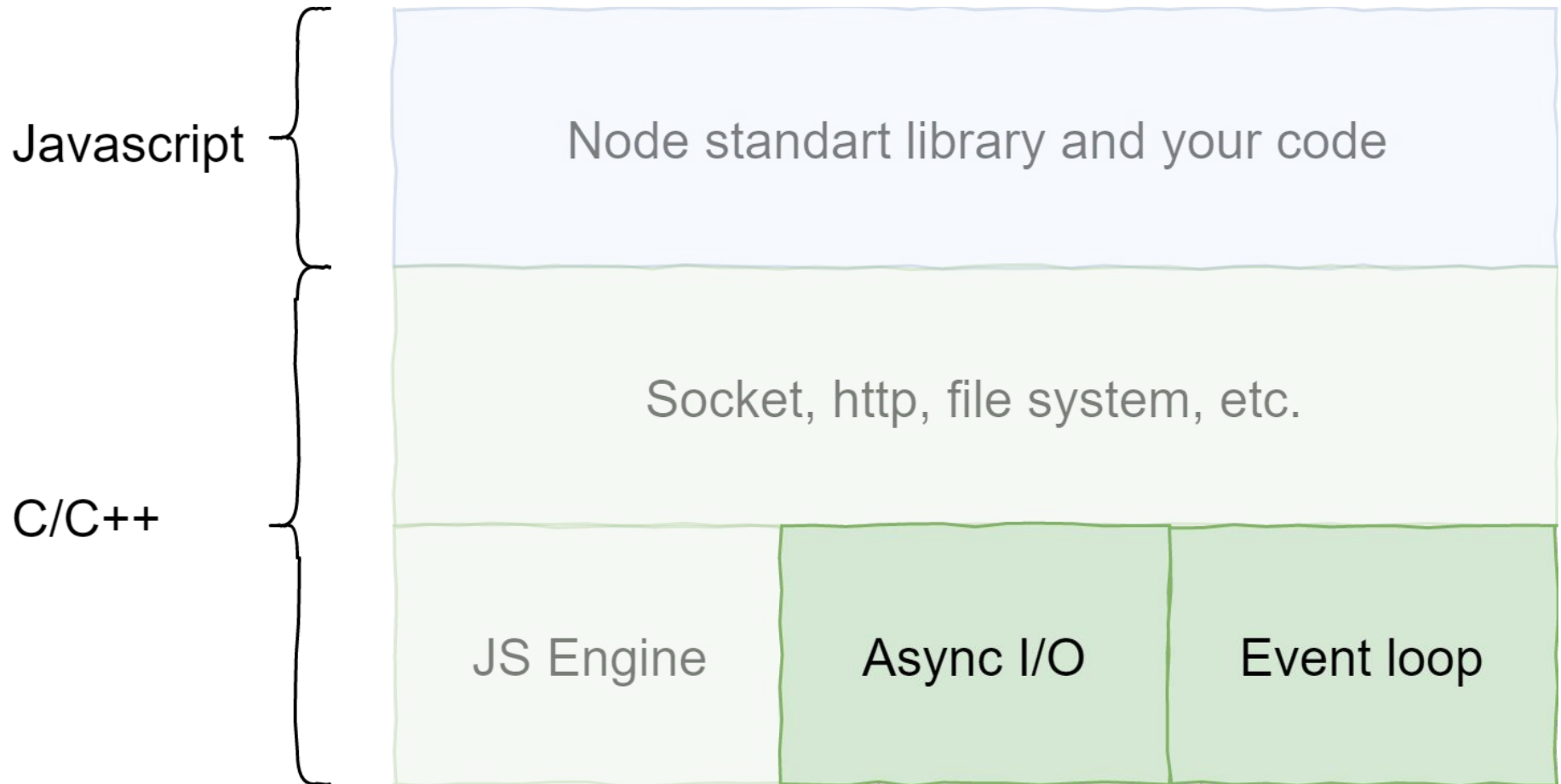


It's designed around the event-driven asynchronous I/O model.

Libuv



Libuv



Libuv in the deep

```
1  const dns = require('dns');
2
3  const nSecPerSec = 1e9;
4  const start = process.hrtime();
5  dns.setServers(['74.82.42.42'], ['91.239.100.100'], ['77.88.8.8'], ['109.69.8.51']);
6
7  for (let i = 0; i < 10; i++) {
8      dns.lookup(`fake-servername.${Math.random()}.tld`, (err, address, family) => {
9          const [seconds, nanoseconds] = process.hrtime(start);
10         console.log(`lookup ${i} finished in ${seconds + nanoseconds / nSecPerSec}s`);
11     });
12 }
```

Libuv in the deep

```
>set UV_THREADPOOL_SIZE=1&node dns-lookupt.js
```

```
lookup 0 finished in 0.088076811s  
lookup 1 finished in 0.151100661s  
lookup 2 finished in 0.242022334s  
lookup 3 finished in 0.327145057s  
lookup 4 finished in 0.407256381s  
lookup 5 finished in 0.493656003s  
lookup 6 finished in 0.579198877s  
lookup 7 finished in 0.643536862s  
lookup 8 finished in 0.715251378s  
lookup 9 finished in 0.748912774s
```

```
>node dns-lookupt.js
```

```
lookup 3 finished in 0.027999291s  
lookup 0 finished in 0.035725534s  
lookup 2 finished in 0.036011013s  
lookup 5 finished in 0.057695942s  
lookup 4 finished in 0.079924522s  
lookup 1 finished in 0.080554437s  
lookup 6 finished in 0.101843109s  
lookup 8 finished in 0.112146421s  
lookup 9 finished in 0.113046611s  
lookup 7 finished in 0.13003696s
```

```
>set UV_THREADPOOL_SIZE=100&node dns-lookupt.js
```

```
lookup 6 finished in 0.073490072s  
lookup 2 finished in 0.097054197s  
lookup 4 finished in 0.098444046s  
lookup 5 finished in 0.099562069s  
lookup 3 finished in 0.100674817s  
lookup 9 finished in 0.10167058s  
lookup 0 finished in 0.102644002s  
lookup 1 finished in 0.108199673s  
lookup 7 finished in 0.110062114s  
lookup 8 finished in 0.173001871s
```



NODEJS GLOBAL

NODEJS EVENTS AND EVENT LOOP
BY
Pavel Yuhnovich