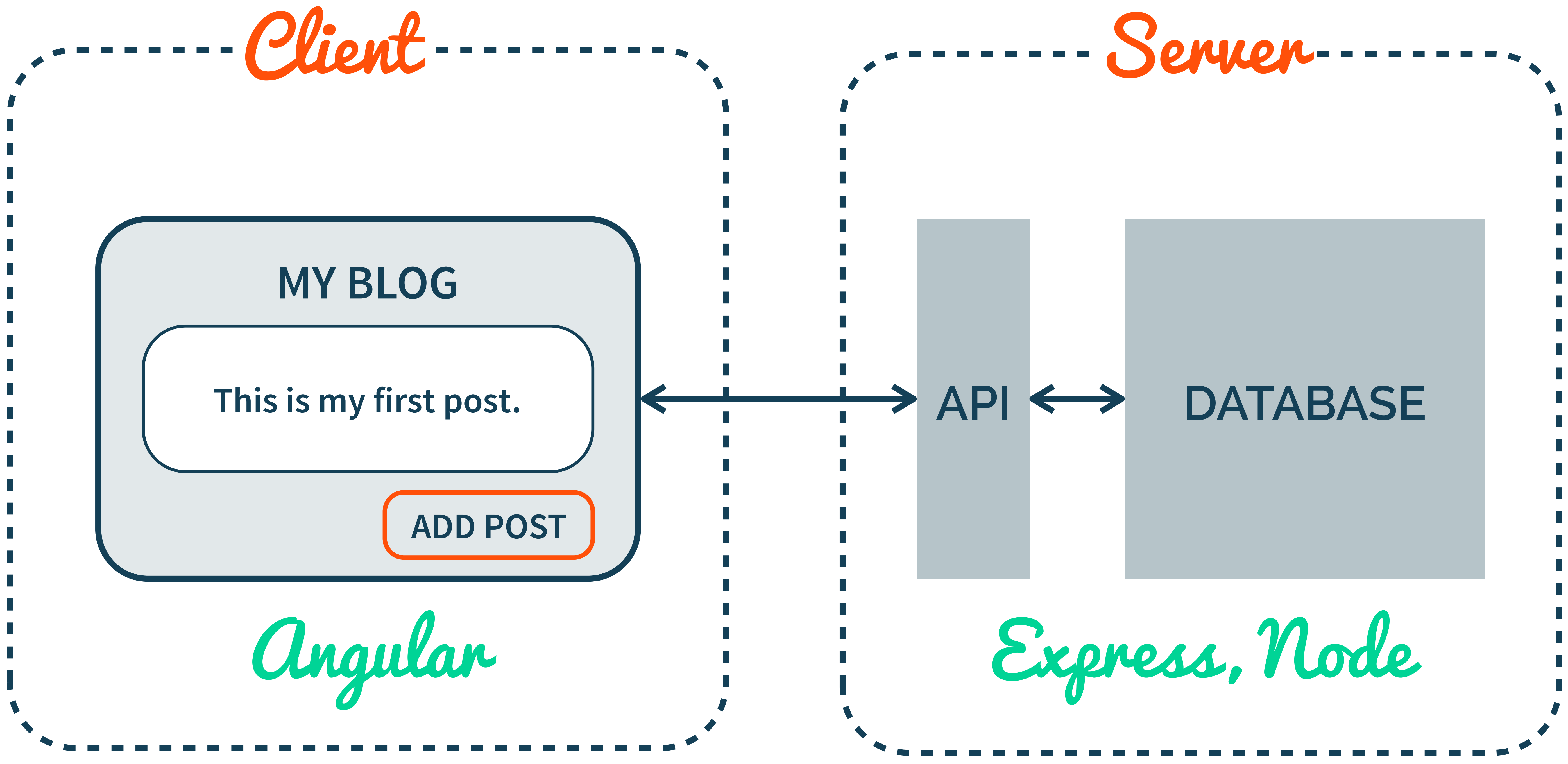
NODE & EXPRESS

SERVER-SIDE JAVASCRIPT



NODE

"Node.js is a platform built on ***Chrome's JavaScript runtime*** *for easily building fast, scalable network applications. Node.js uses an* ***event-driven, non-blocking I/O model*** *that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices"*

nodejs.org

NON-BLOCKING

Asychronous

I/O operations are slow

non-blocking while one process is waiting for I/O, let another process make use of CPU

TRADITIONAL SERVERS

Apache, IIS

servers serve several clients at the same time: how?

multi-process or multi-threaded

each connection results in the creation of a dedicated child process/thread

parent process/main thread remains available, listening for new connections

EVENT-DRIVEN CONCURRENCY

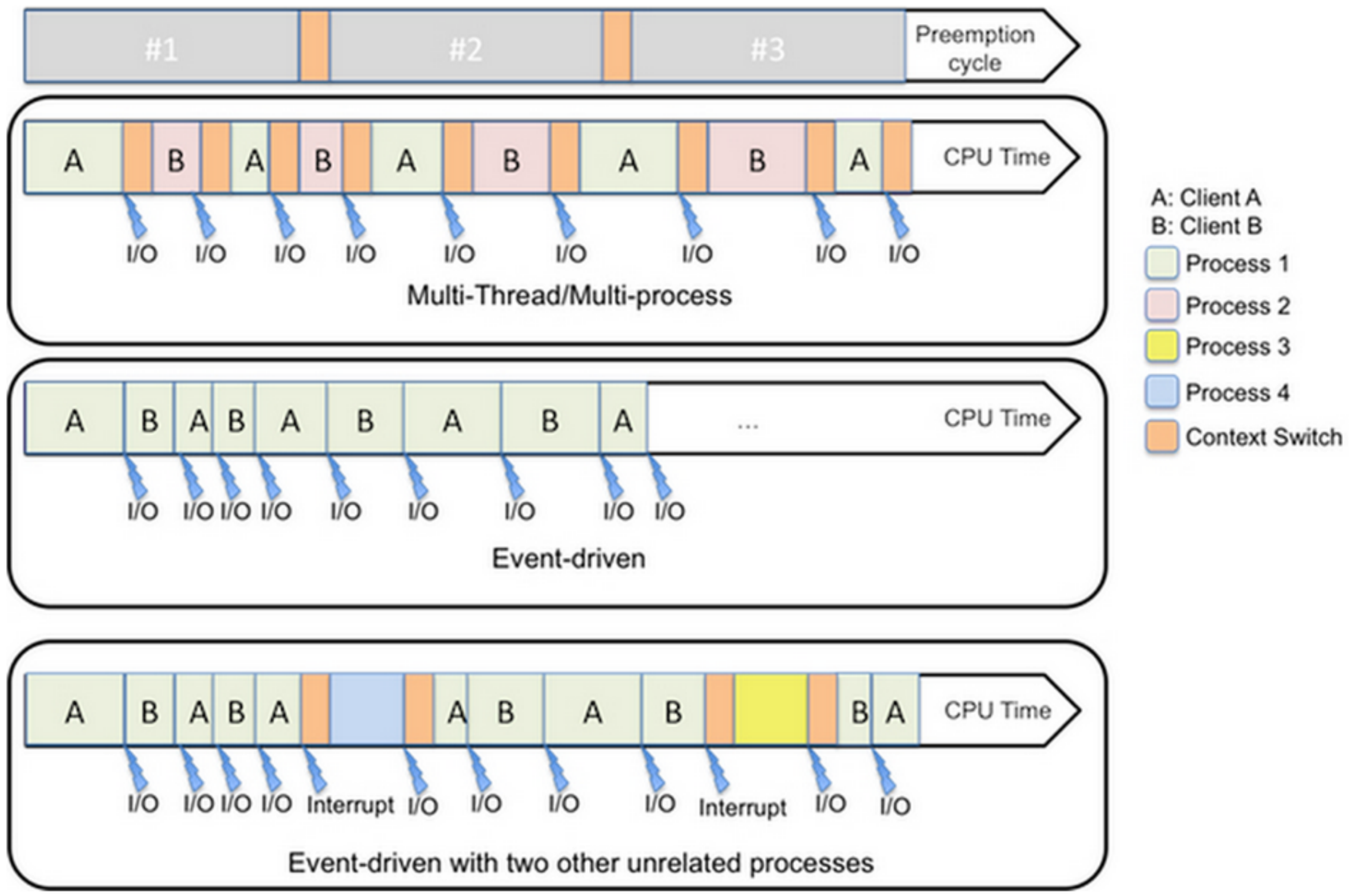
Node, nginx, Twisted, EventMachine

everything runs in one process, one thread

a event is emitted and the appropriate callback for that event is invoked

Once an event is treated, the process is ready to treat another event

www.teenycloud.com/blog/2013/11/30/node-event-driven-programming/



www.teenycloud.com/blog/2013/11/30/node-event-driven-programming/

THE EVENT-LOOP

all the events are processed by event-loop queue

event-loop fetches next event to process and dispatches the corresponding handler

anyone blocking the event-loop will prevent the other events from being processed

single-threaded: DO NOT BLOCK THE EVENT LOOP

Node API is non-blocking (with the exception of some file system operations which come in two flavors: asynchronous and synchronous)

www.w3.org/TR/2004/REC-webarch-20041215/

JAVASCRIPT AND I/O

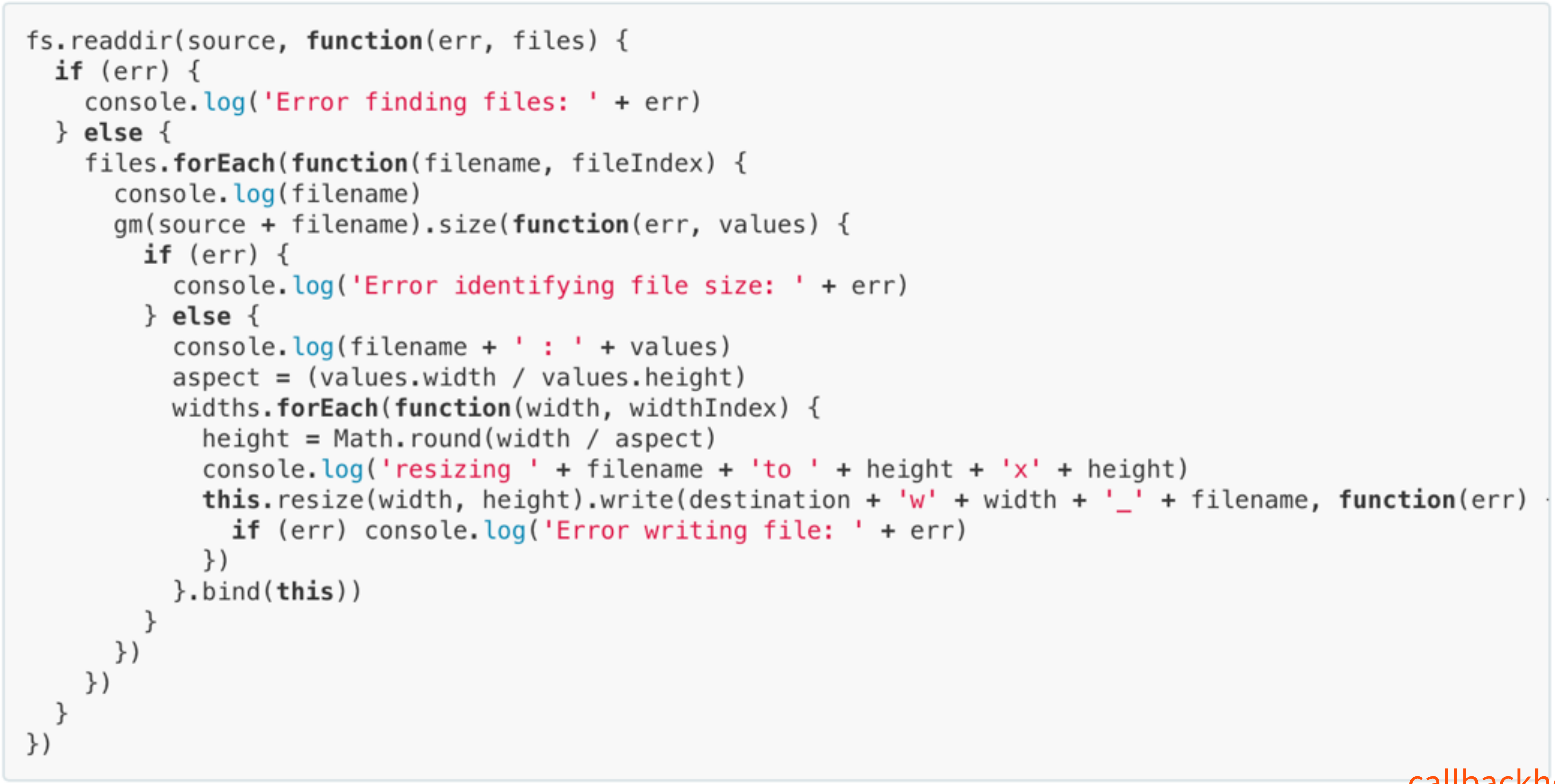
JavaScript was designed for being used inside a browser; missing basic I/O libraries (such as file operations)

Node: Javascript + I/O API

could make I/O natively non-blocking in Node

www.w3.org/TR/2004/REC-webarch-20041215/

CALLBACK STYLE PROGRAMMING



```
fs.readdir(source, function(err, files) {
  if (err) {
    console.log('Error finding files: ' + err)
  } else {
    files.forEach(function(filename, fileIndex) {
      console.log(filename)
      gm(source + filename).size(function(err, values) {
        if (err) {
          console.log('Error identifying file size: ' + err)
        } else {
          console.log(filename + ' : ' + values)
          aspect = (values.width / values.height)
          widths.forEach(function(width, widthIndex) {
            height = Math.round(width / aspect)
            console.log('resizing ' + filename + 'to ' + height + 'x' + height)
            this.resize(width, height).write(destination + 'w' + width + '_' + filename, function(err)
              if (err) console.log('Error writing file: ' + err)
            })
          }.bind(this))
        }
      })
    })
  }
})
```

callbackhell.com

EXPRESS

Web application framework built on top of Node

provides scaffolding: “a thin layer of fundamental web application features”

minimal, flexible

serve simple pages, APIs, etc.

expressjs.com

NEXT CLASS: MIDTERM REVIEW

courses.engr.illinois.edu/cs498rk1/