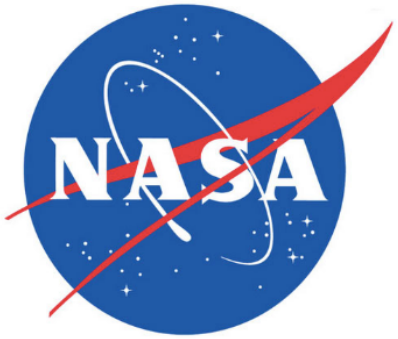
Introduction to

by Anastasiia Derkach

softserve

Companies that Used Node.js



softserve

Node.js is open source, cross-platform, and since its introduction in 2009 by Ryan Dahl

If GitHub stars are one popularity indication factor, having 58000+ stars means being very popular.

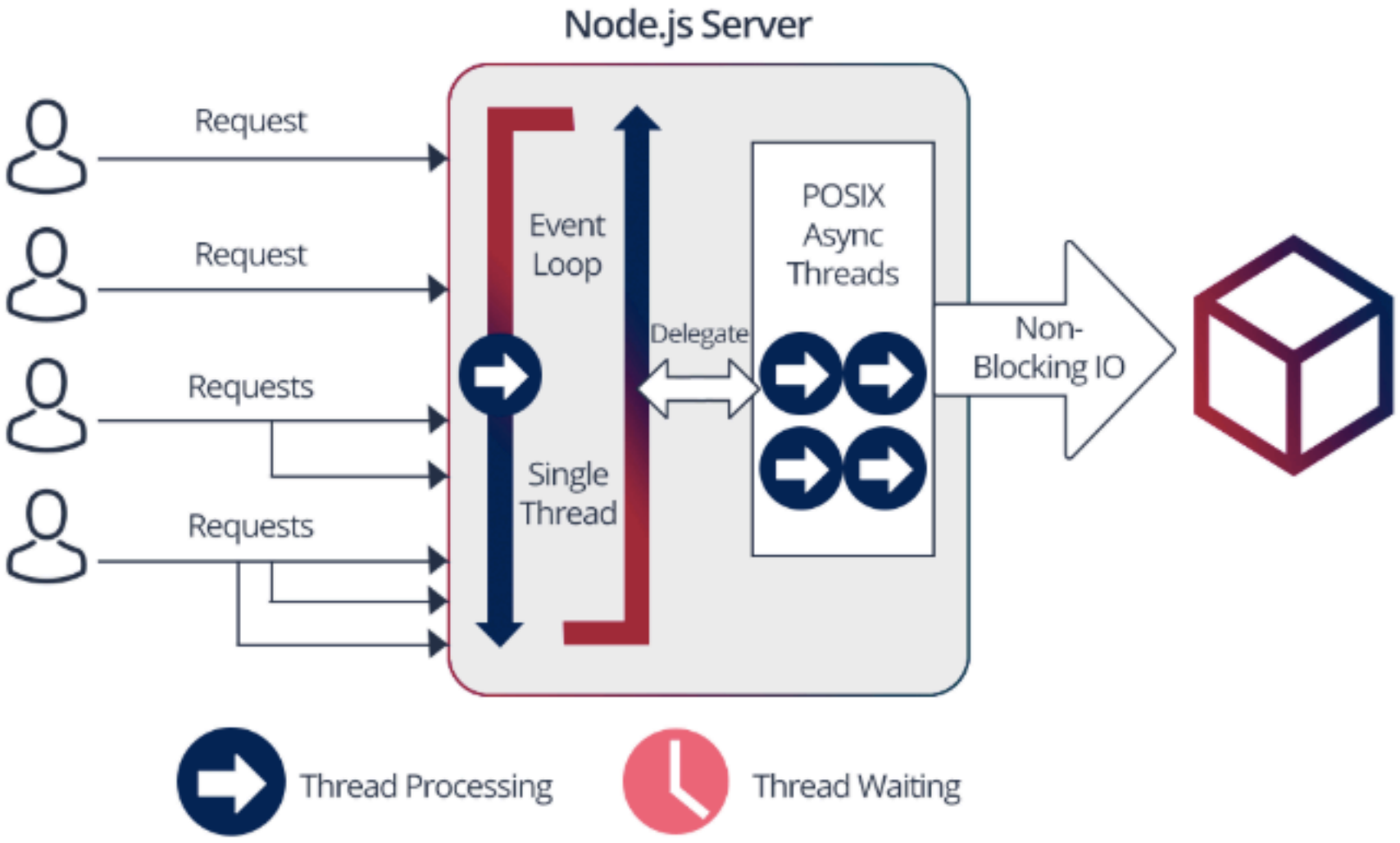
Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser.

A Node.js app is run in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm.

This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency, which could be a significant source of bugs.

The following concepts are also key to understand asynchronous programming, which is one fundamental part of Node.js:

- Asynchronous programming and callbacks
- Timers
- Promises
- Async and Await
- Closures
- The Event Loop


Node.js Server
Request
Request
Requests
Requests
Event Loop
Single Thread
Delegate
POSIX Async Threads
Non-Blocking IO
Thread Processing
Thread Waiting

softserve

WHY TO USE

softserve

One development language JS on client and server side.

Open source technology.

Many NPM package available on npmjs.org.

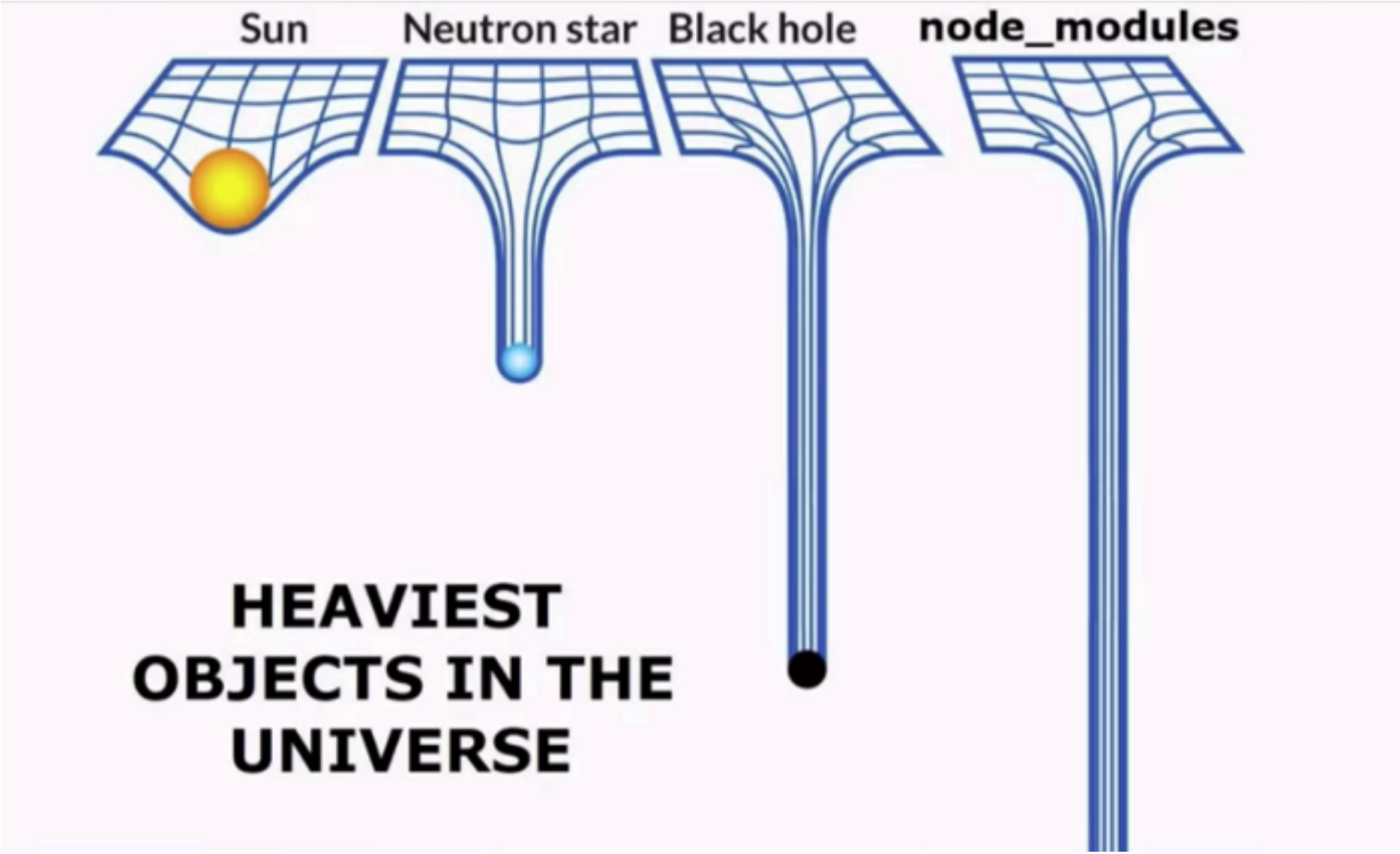
Short learning curve(Event loop theory, JavaScript, Evented programming).

One development language JS on client and server side.

Node.js has a unique advantage because millions of frontend developers that write JavaScript for the browser are now able to run the server-side code and frontend-side code without the need to learn a completely different language.

NPM

npm with its simple structure helped the ecosystem of node.js proliferate and now the npm registry hosts almost 500.000 open source packages you can freely use.


Sun
Neutron star
Black hole
node_modules
HEAVIEST OBJECTS IN THE UNIVERSE

You control the environment

Unless you are building an open source application that anyone can deploy anywhere, you know which version of Node.js you will run the application on. Compared to the browser environment, where you don't get the luxury to choose what browser your visitors will use, this is very convenient

This means that you can write all the modern ES6-7-8-9 JavaScript that your Node.js version supports.

WHERE TO USE

softserve

- I/O bound Applications
- Data Streaming Application
- Data Intensive Real-time Applications (DIRT)
- JSON APIs based Applications
- Single Page Applications

Node.js frameworks and tools

- **Express,** one of the most simple yet powerful ways to create a web server. Its minimalist approach, unopinionated, focused on the core features of a server, is key to its success
- **Meteor**, an incredibly powerful full-stack framework, powering you with an isomorphic approach to building apps with JavaScript, sharing code on the client and the server. Once an off-the-shelf tool that provided everything, now integrates with frontend libs React, Due and Angular. Can be used to create mobile apps as well.
- **Koa**, built by the same team behind Express, aims to be even simpler and smaller, building on top of years of knowledge. The new project born out of the need to create incompatible changes without disrupting the existing community.
- **Next.js**, a framework to render server-side rendered React applications.
- **Micro** a very lightweight server to create asynchronous HTTP microservices.
- **Socket.io**, a real-time communication engine to build network applications.

WHERE NOT TO USE NODE.JS?

softserve

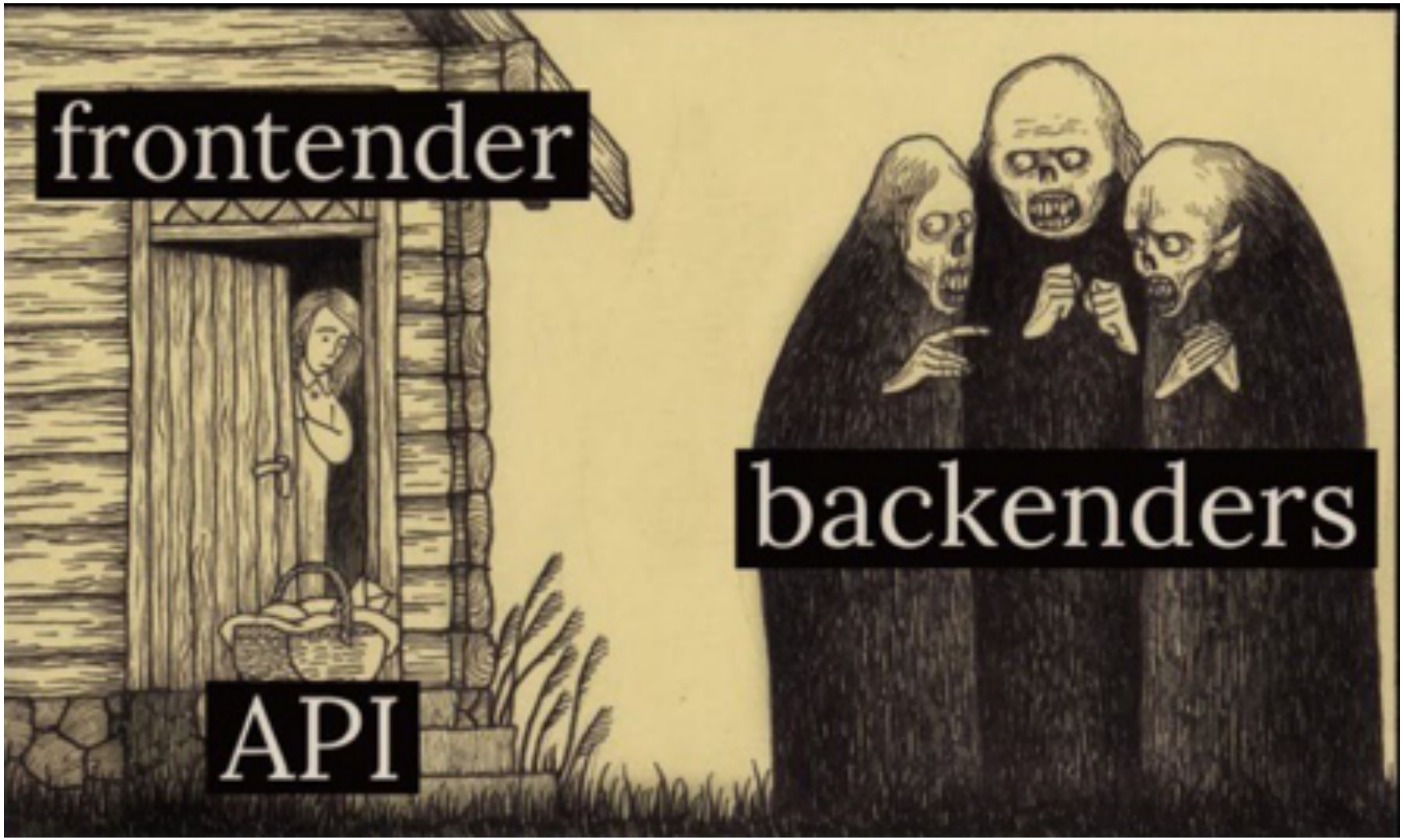
It is not advisable to use Node.js for CPU intensive applications.

```
const http = require('http')

const hostname = '127.0.0.1'
const port = 3012

const server = http.createServer((req, res) => {
  res.statusCode = 200
  res.setHeader('Content-Type', 'text/plain')
  res.end('Hello World')
})

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`)
})
```



softserve

Failures in Node.js

softserve

RESOURCES USED

- https://flaviocopes.com/nodejs/
- https://nodejs.dev/
- https://nodejs.org/