Node.JS Architecture

Presented by : Divya Gupta

Table of Content

Q. How Node.JS is Asynchronous, Event-Driven, & Single-Threaded at the same time.

- Introduction to Node.JS

3 key Concepts:

⟶ Asynchronous Programming
⟶ Event-Driven Architecture
⟶ Single-Threaded Execution

- Node.JS Server Architecture

- Components of NodeJS Architecture

Introduction

- Node.js (Node js) is an open-source and cross-platform JavaScript runtime environment.

- It runs on Chrome’s V8 JavaScript engine. It allows developers to run JavaScript code on the server.

- Node.js enables developers to get into the server-side world.

3 Key Concepts

Asynchronous & Non-Blocking

Event-Driven

Single-Threaded

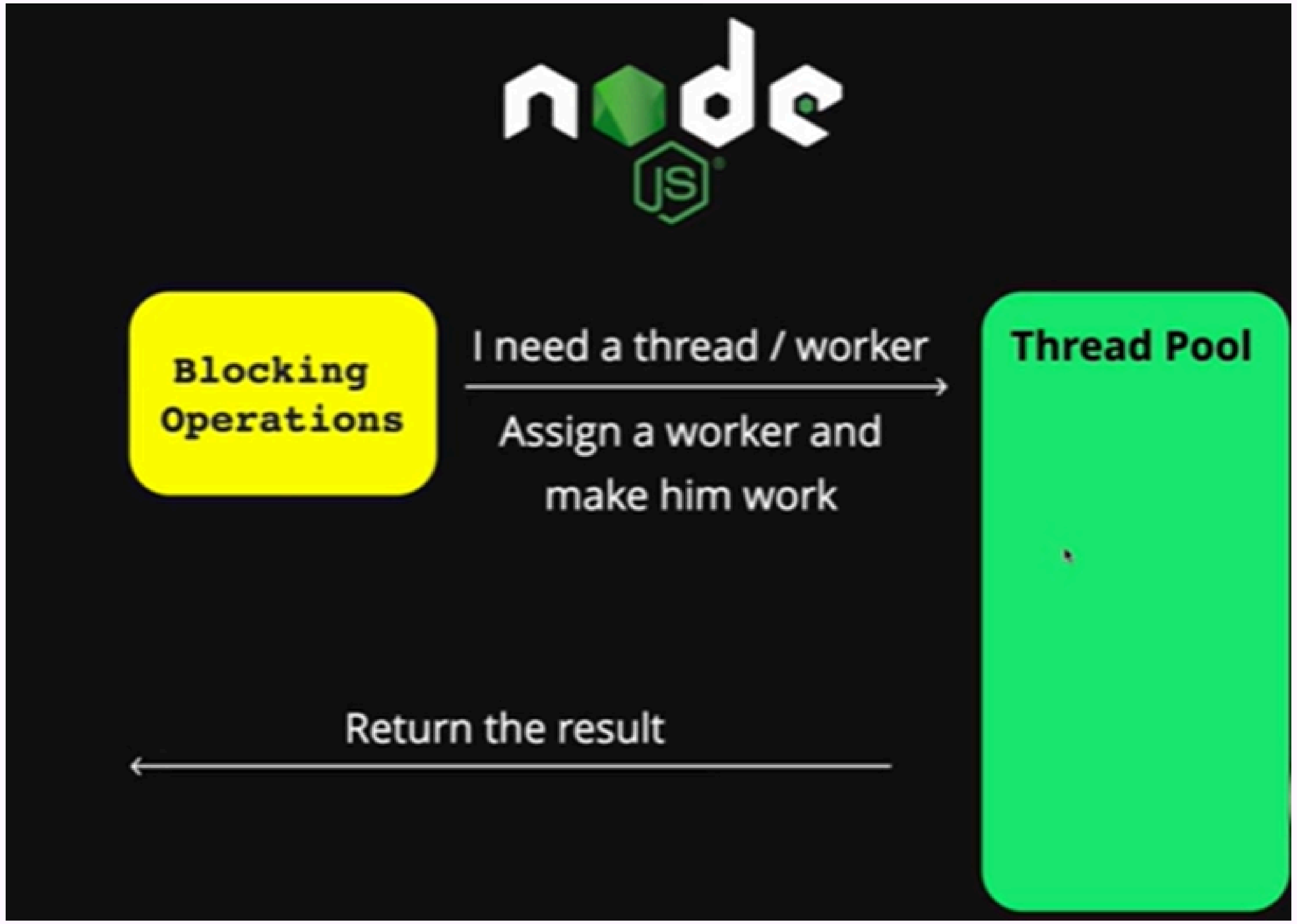
1.Asynchronous & Non-Blocking

***Asynchronous Operations*:**

Node.js doesn't wait for a task to complete before moving on to the next. Instead, it registers a callback function to be executed when the operation finishes.

Non-Blocking I/O:

Node.js uses asynchronous I/O operations, meaning it doesn't block the main thread while waiting for data from a file or network connection.



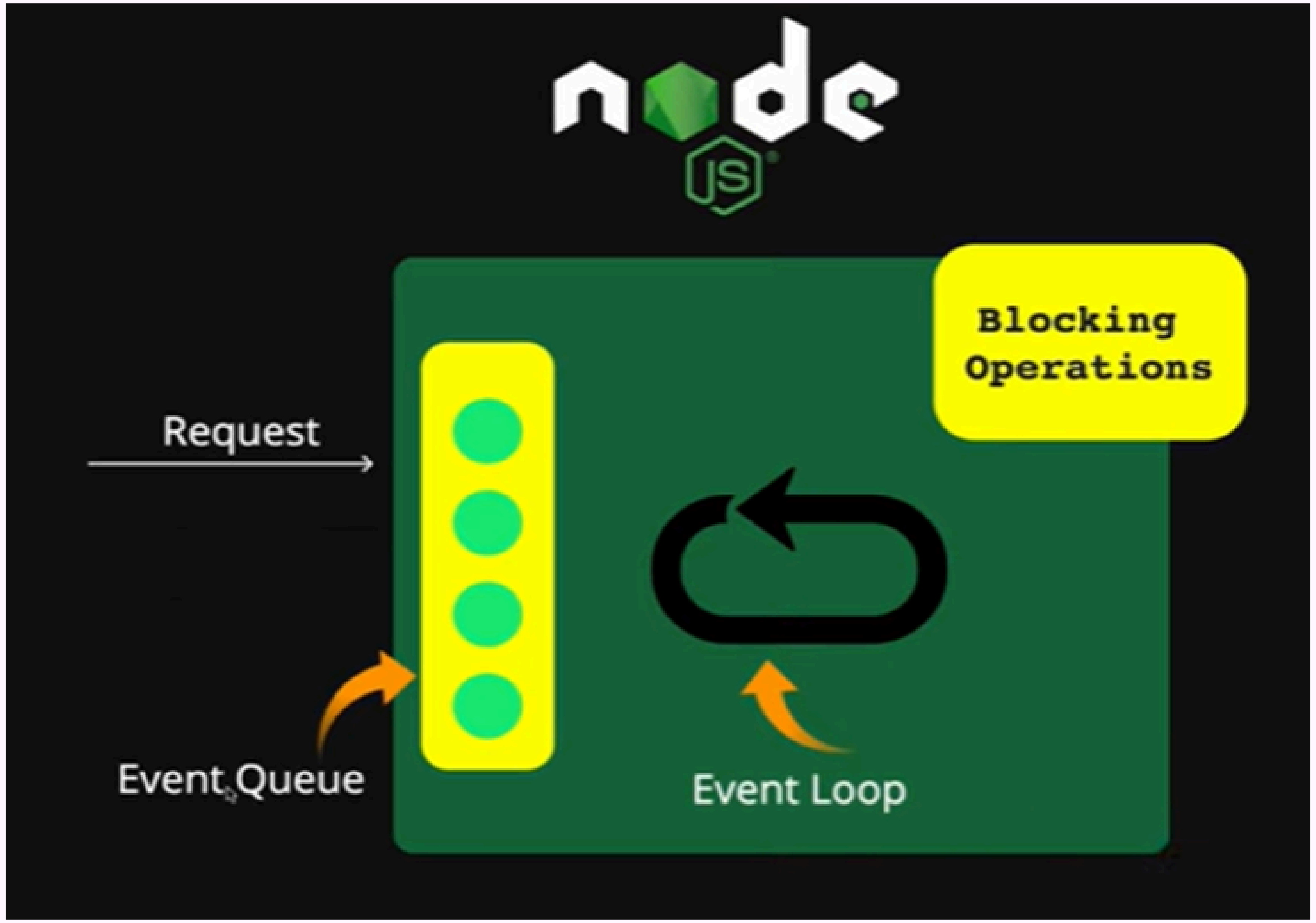
Blocking Operations

I need a thread / worker

Assign a worker and make him work

Thread Pool

Return the result



Request

Blocking Operations

Event Queue

Event Loop

2.Event-Driven

- ***Event Loop***

Node.js uses an event loop to process events and callbacks

- ***Event Queue***

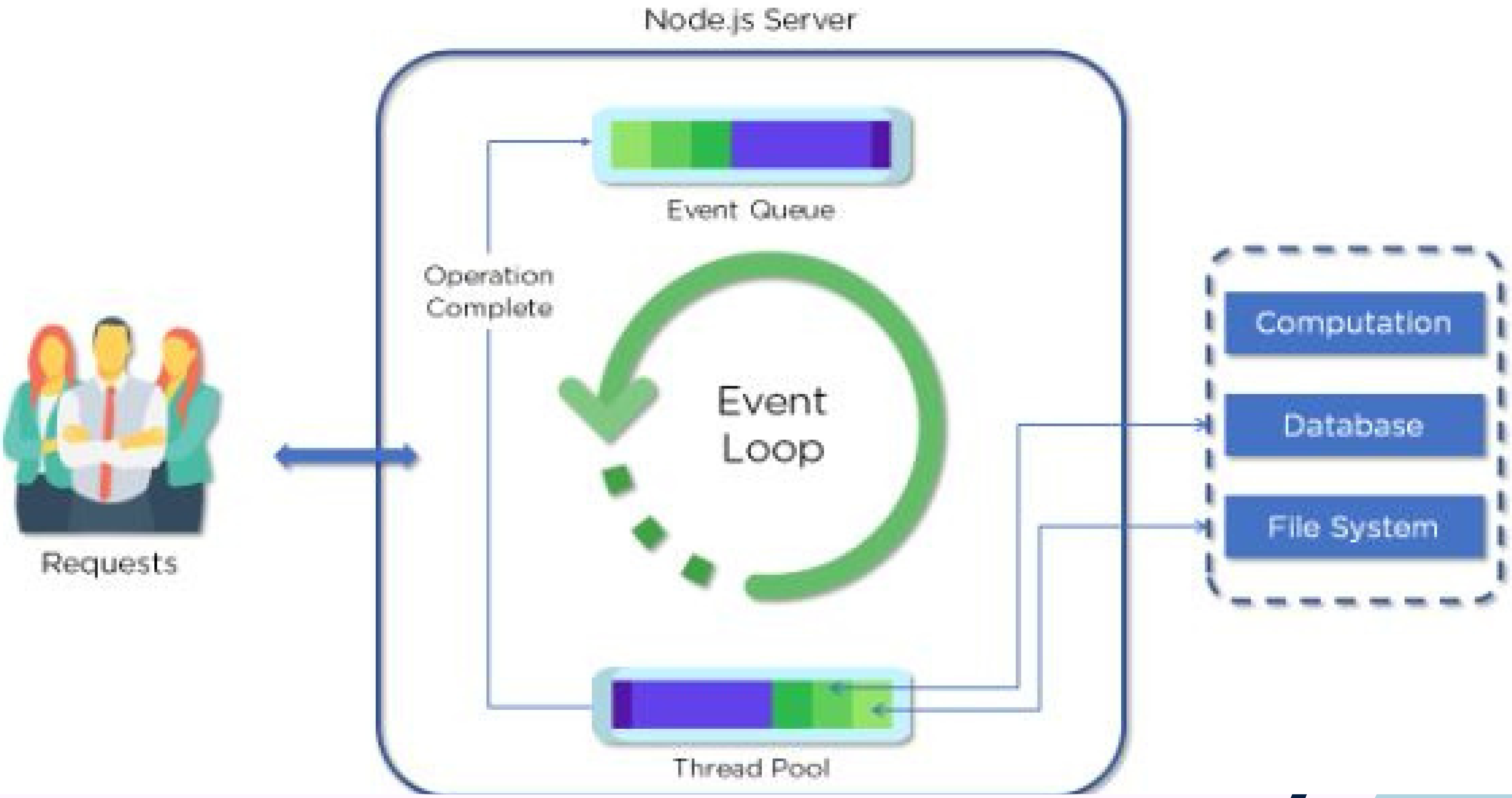
When an asynchronous operation completes, it adds a callback to the event queue.

.

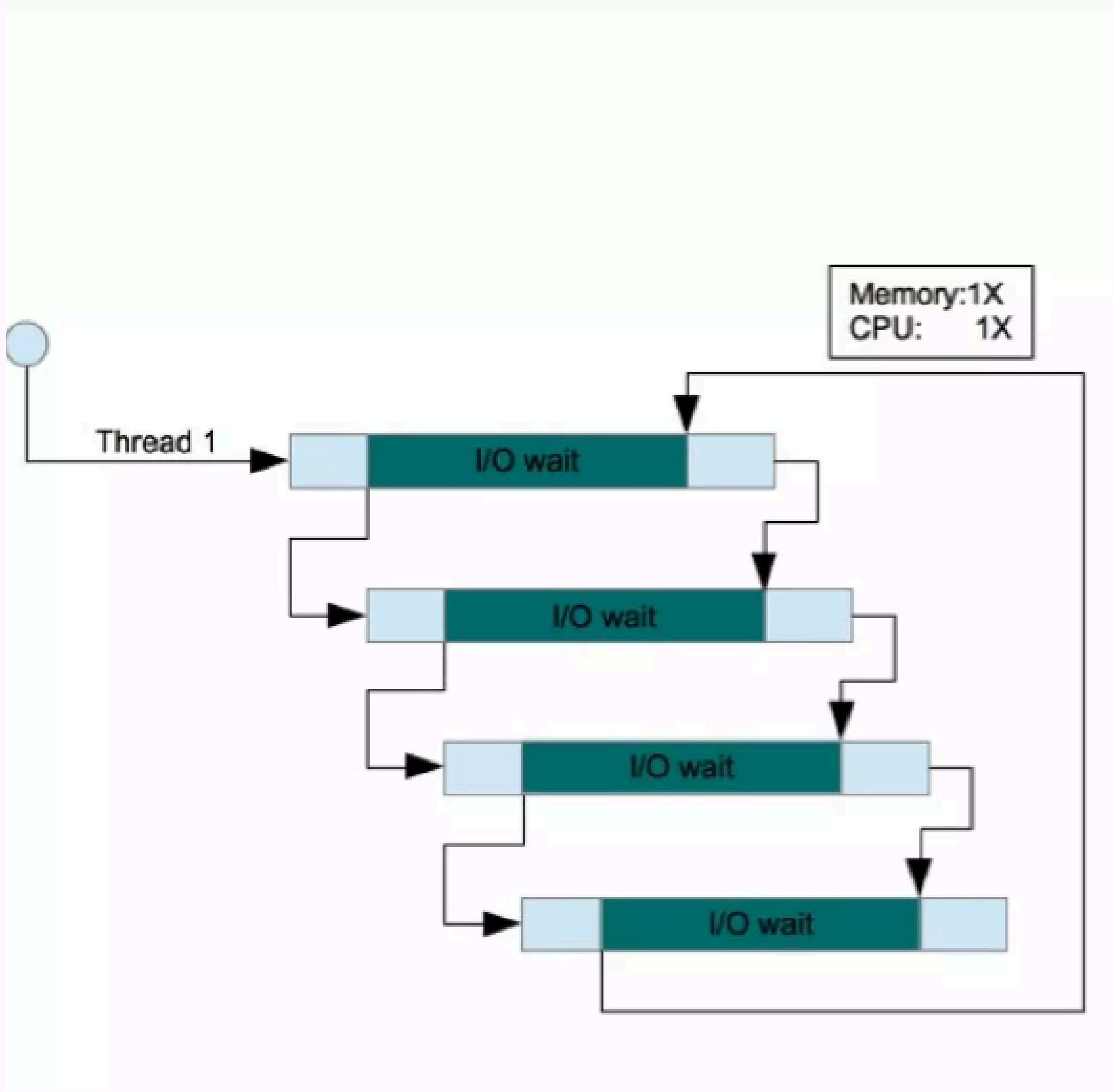
- ***Event Loop Processing***

The event loop continuously checks the event queue and executes callbacks.

.


Node.js Server
Event Queue
Operation Complete
Event Loop
Thread Pool
Requests
Computation
Database
File System

3.Single-Threaded Execution



Single Thread:

Node.js runs on a single thread of execution.

Non-blocking I/O:

The asynchronous nature of Node.js allows it to handle multiple concurrent requests efficiently without blocking the main thread.

Node JS Architecture

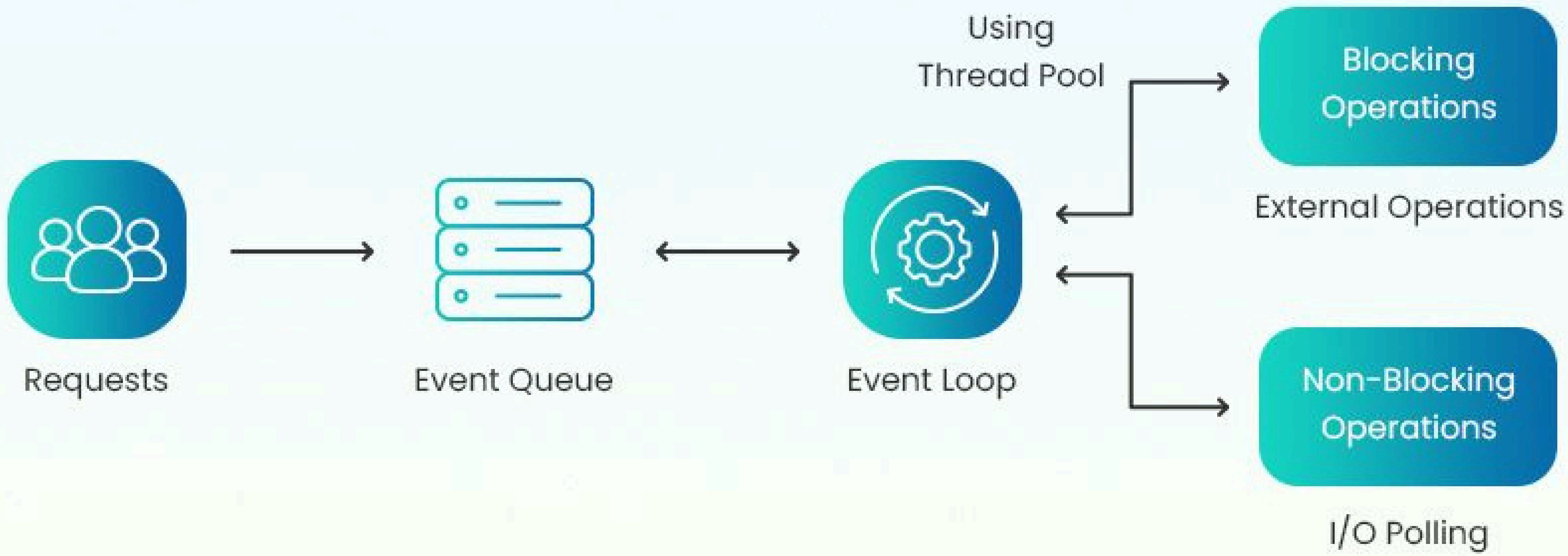
To manage several concurrent clients, Node.js employs a "Single Threaded Event Loop" design. The JavaScript event-based model and the JavaScript callback mechanism are employed in the Node.js Processing Model.

It employs two fundamental concepts:

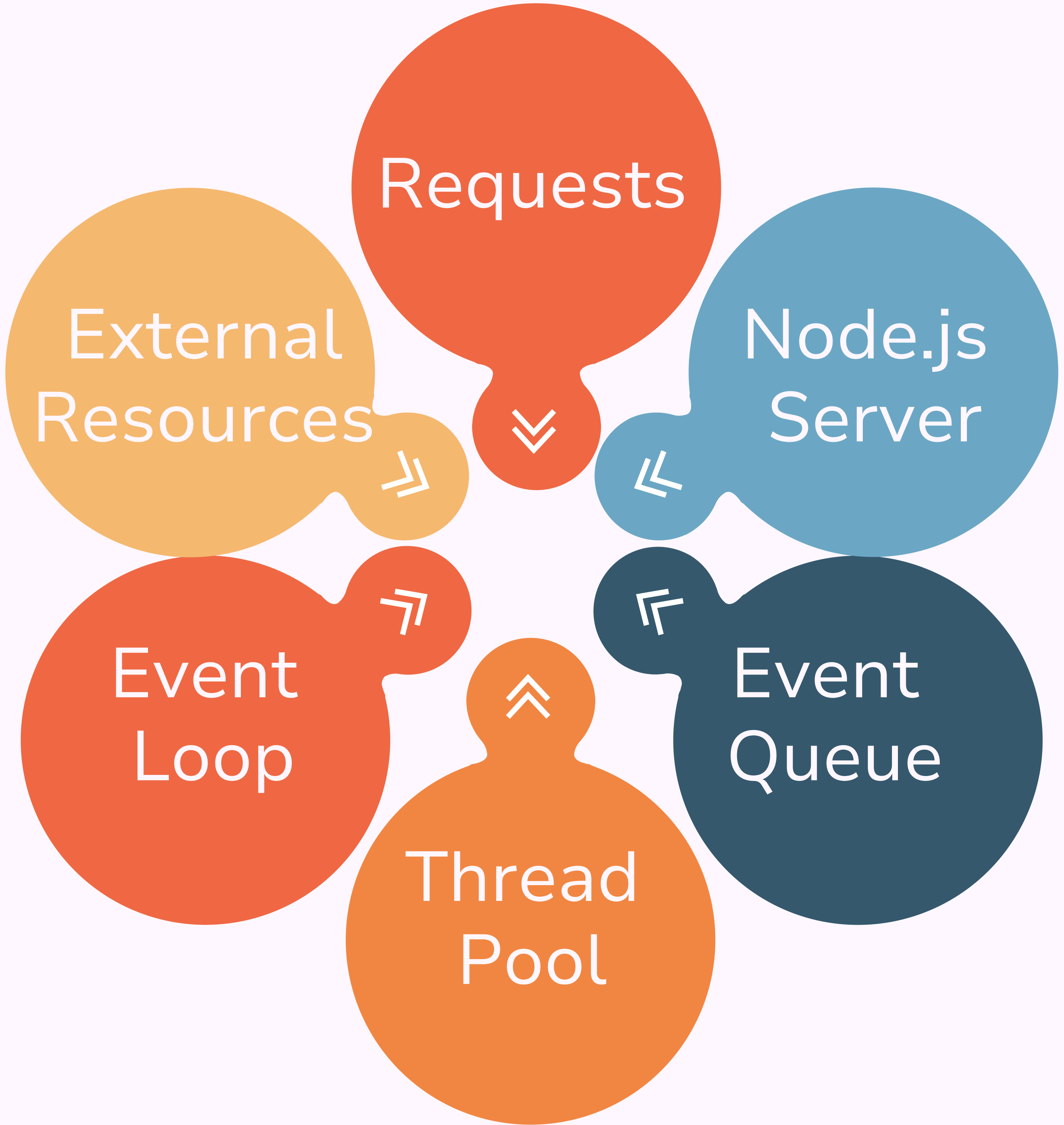
1) Asynchronous model

2) Non-blocking of I/O operations

These features enhance the scalability, performance, and throughput of Node.js web applications.


Using
Thread Pool
Blocking
Operations
External Operations
Requests
Event Queue
Event Loop
Non-Blocking
Operations
I/O Polling

Components of Node.JS Architecture



Bibliography

- https://radixweb.com/nodejs-architecture
- https://www.geeksforgeeks.org/node-js-web-application-architecture/
- https://www.scaler.com/topics/nodejs/node-js-architecture/
- https://nodejs.org/docs/latest/api/

Thank You