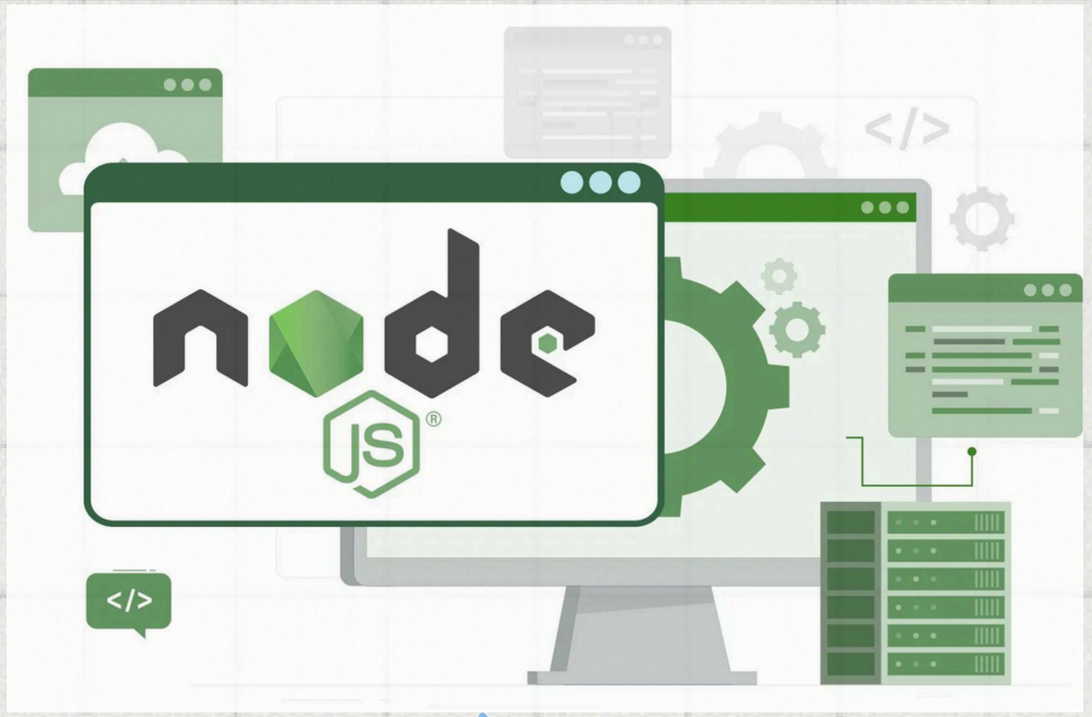
NODEJs

Presented by AKSHIMA KAKKAR
22105042
B.VOC WEB DESIGNING

Table of Content

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



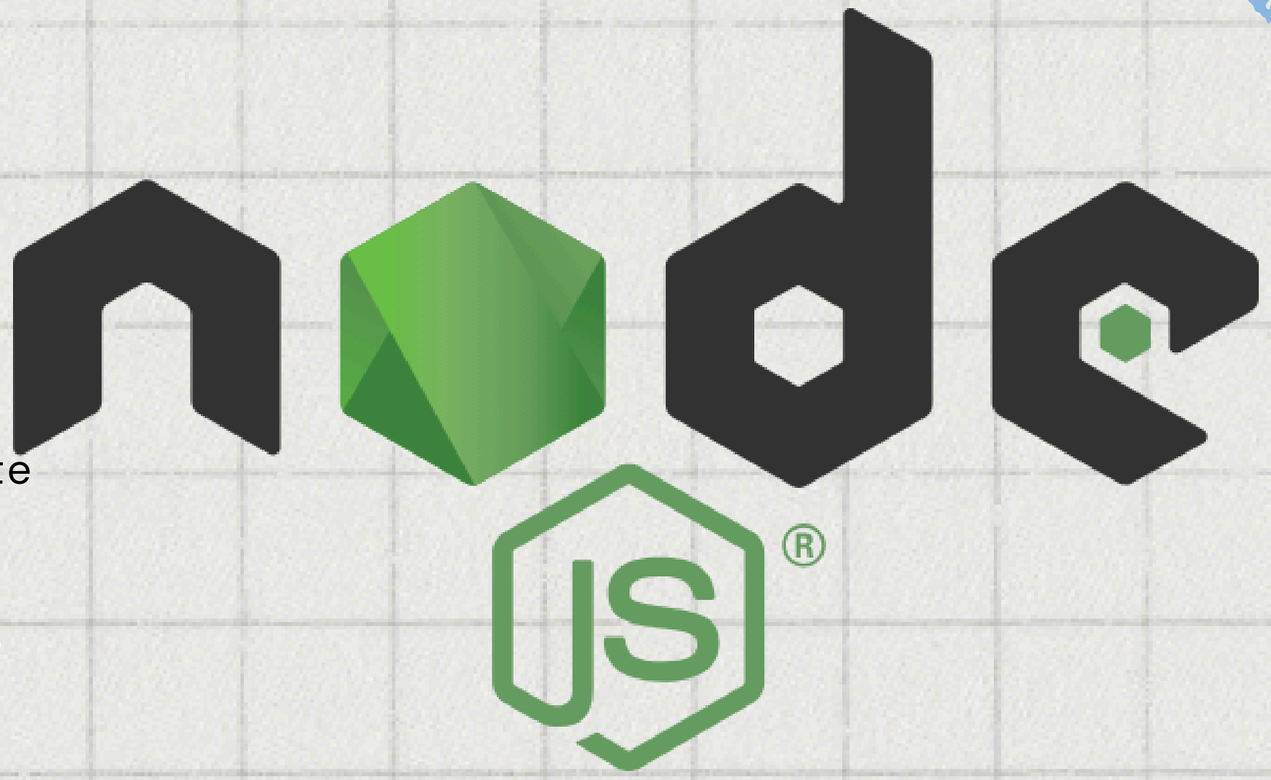
- *Introduction to NodeJS*

3 Key Concepts

- *Asynchronous*
- *Event-driven*
- *Single-threaded*

- *How These Concepts Work Together in Node.js*

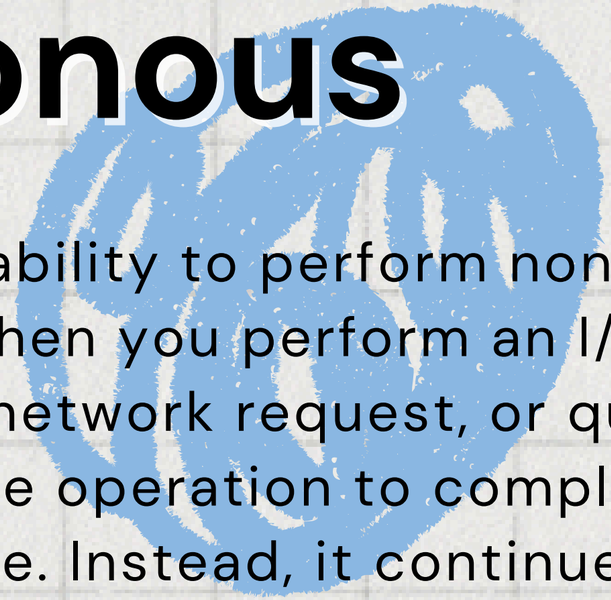
INTRO TO NODEJs



Node.js is a powerful and popular runtime environment that allows developers to execute JavaScript code outside of a web browser.

It's widely used for real-time applications like chat services and APIs.

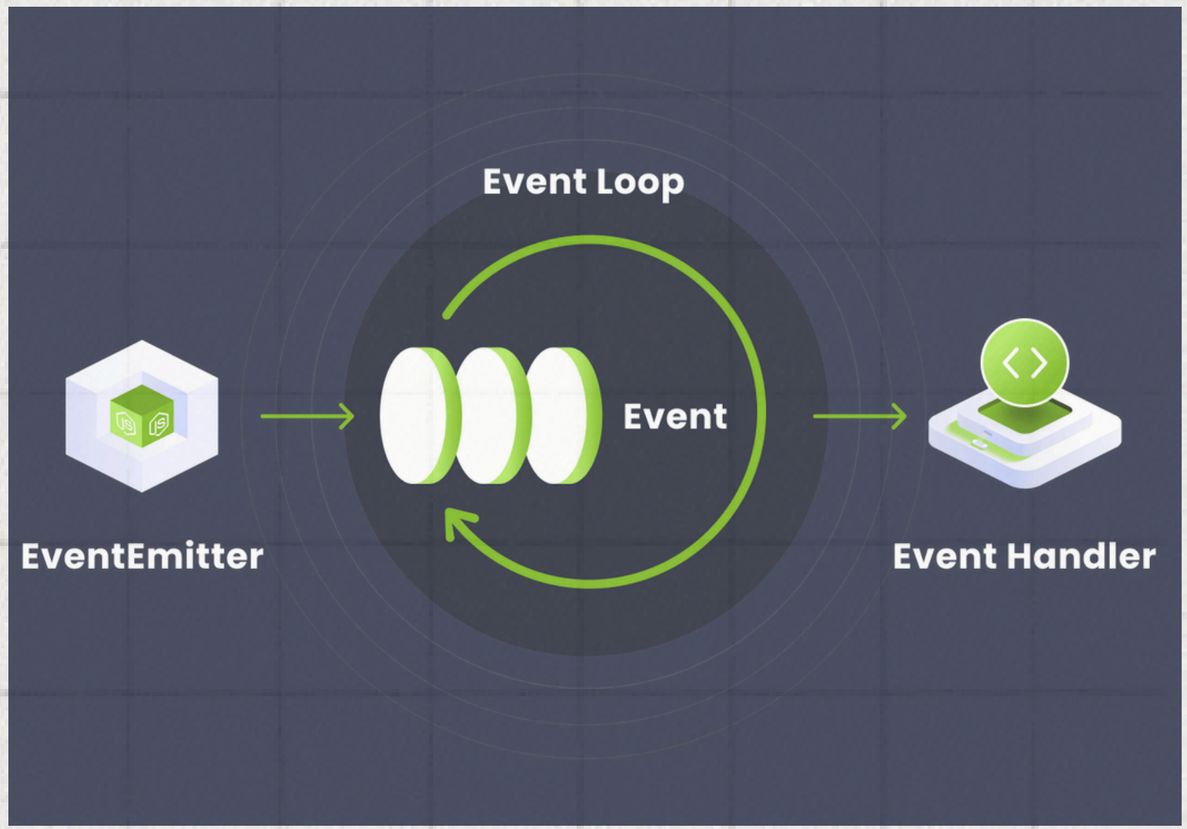
Asynchronous



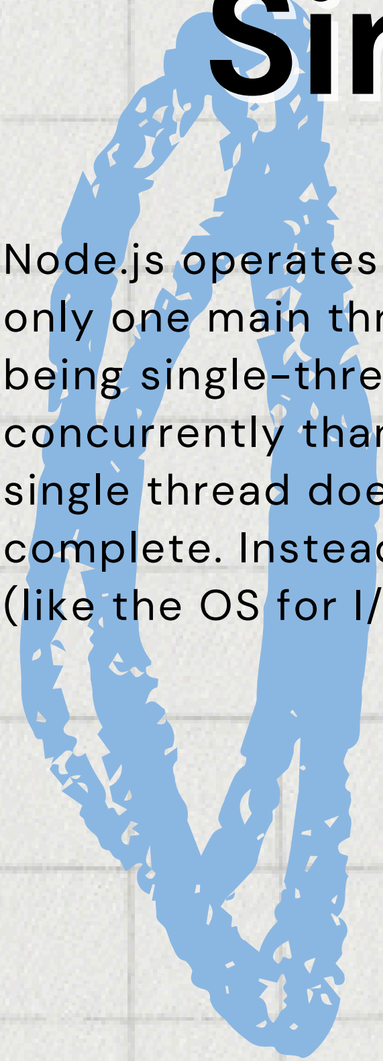
In Node.js, asynchronous refers to the ability to perform non-blocking operations. This means that when you perform an I/O operation (like reading a file, making a network request, or querying a database), Node.js doesn't wait for the operation to complete before moving on to execute other code. Instead, it continues to execute the next lines of code and deals with the result of the I/O operation once it's ready (typically through a callback, promise, or async/await).

Event-driven

INode.js is event-driven, meaning that its execution is driven by events. When an asynchronous operation completes, an event is emitted. Node.js has an event loop that listens for these events and, when an event is detected, it triggers the corresponding callback functions. This is the core of how Node.js handles asynchronous operations without blocking the main thread.

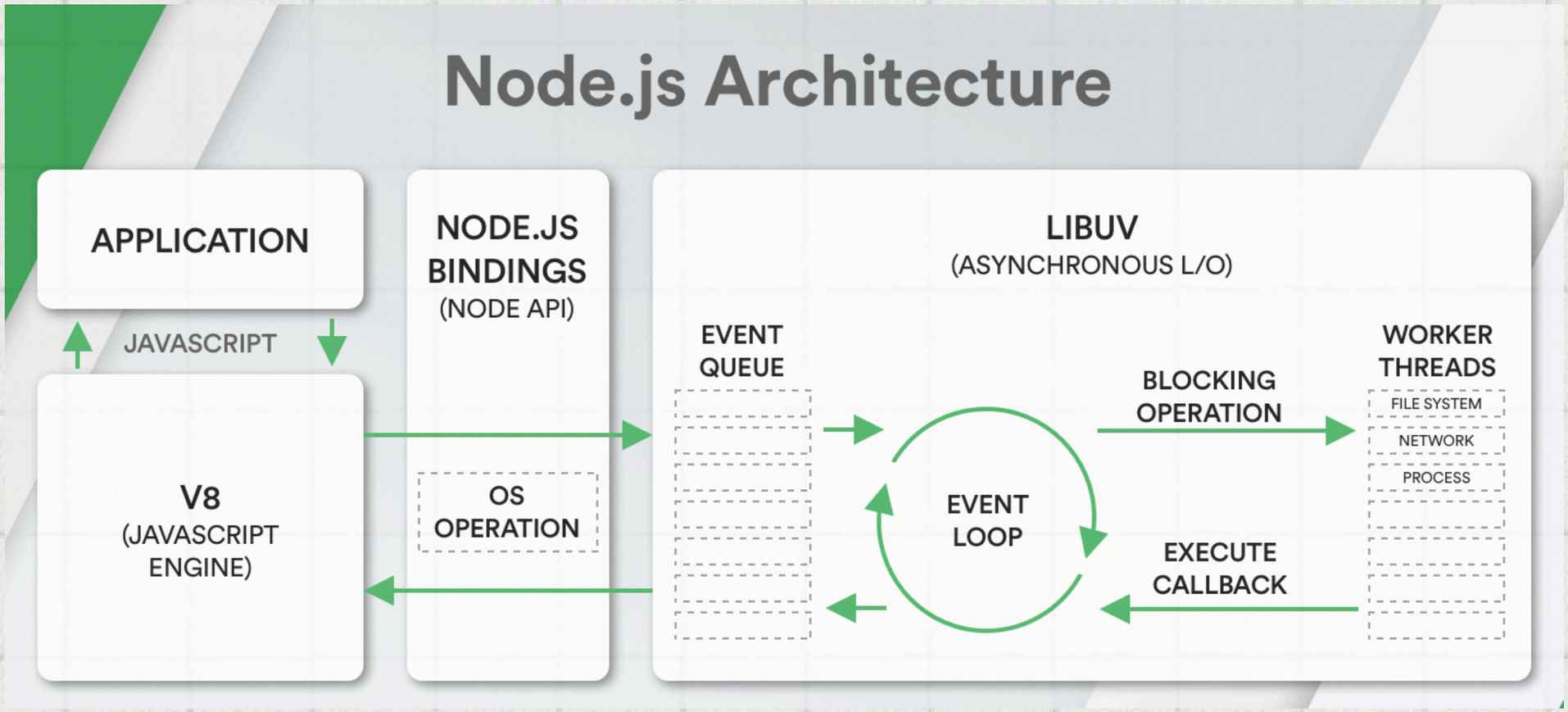


Single-threaded



Node.js operates on a single-threaded event loop, meaning there is only one main thread that handles all the operations. However, despite being single-threaded, Node.js can handle multiple operations concurrently thanks to its asynchronous and event-driven nature. The single thread doesn't get stuck waiting for long-running operations to complete. Instead, it offloads these tasks to the underlying system (like the OS for I/O operations) and moves on to handle other tasks.

Node.js Architecture



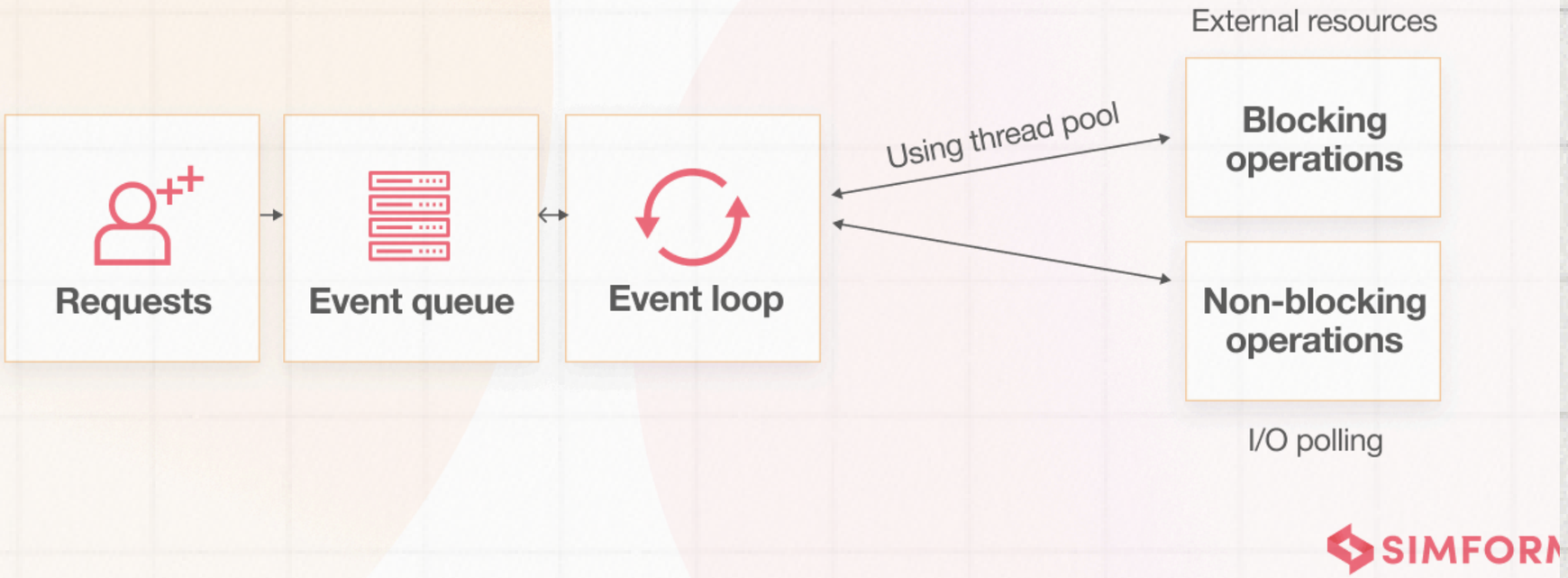
How These Concepts Work Together in Node.js

Single-threaded: **Node.js runs on a single thread, which means it can only execute one task at a time. However, due to its asynchronous nature, this single thread doesn't get blocked by long-running tasks like file I/O or network requests.**

***Asynchronous*: When an I/O operation is initiated, Node.js doesn't wait for the operation to complete. Instead, it registers a callback function and moves on to execute the next piece of code. When the I/O operation completes, the callback function is added to the event queue.**

***Event-driven*: The event loop continuously checks the event queue for pending tasks (callbacks). When an event is detected, the event loop picks up the corresponding callback and executes it. This process allows Node.js to handle multiple operations efficiently, even though it's single-threaded.**

Key elements of the node.js architecture



Thank you very much!