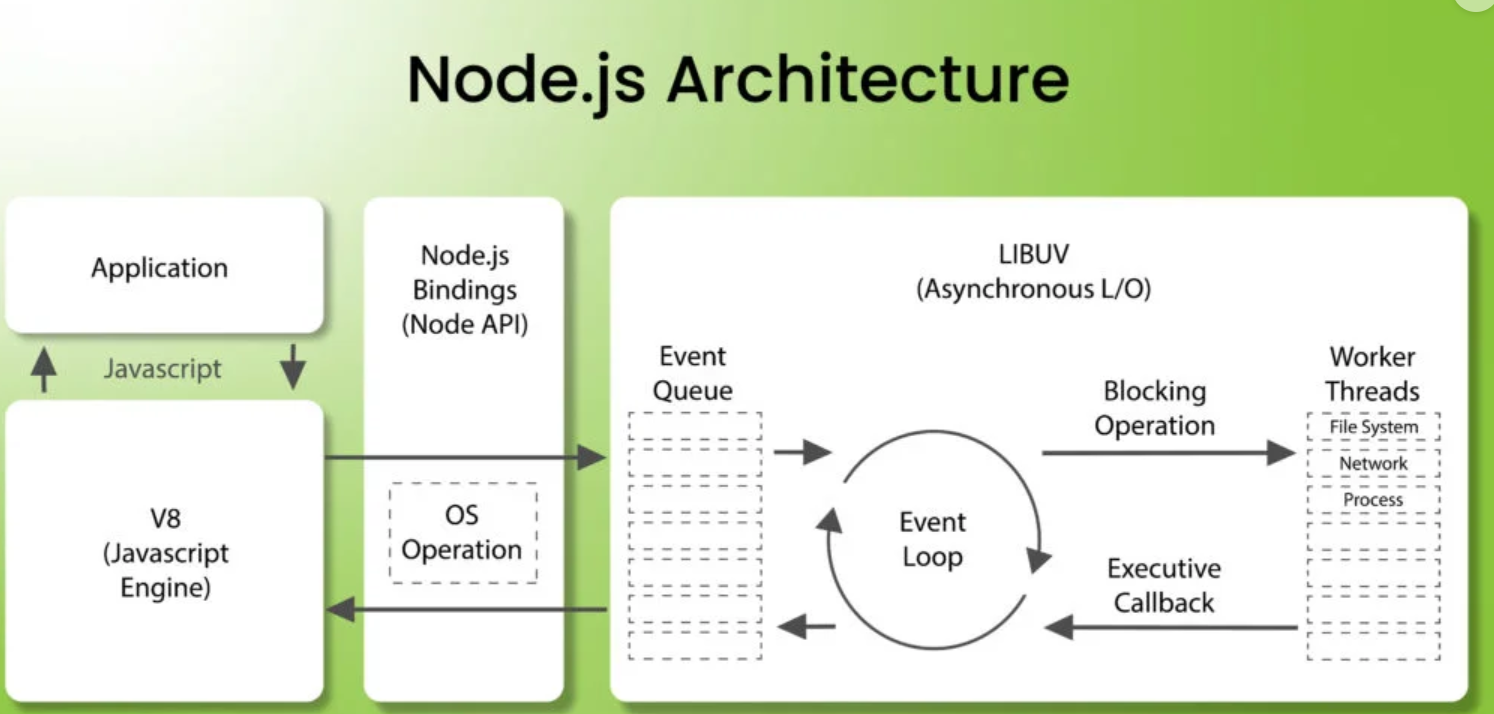
**Java Architecture**



The Architecture consists of V8 Engine, OS operations, Event Queue, Worker Threads, LIBUV.

* Node.js supports “Single Threaded Event Loop” architecture.

**V8 Engine:**

* Node.js uses V8 engine to interpret the JavaScript code. This makes it fast in execution because V8 gives high performance.

**Event:** Some actions to be performed.

**Event Queue:**

In event queue, it stores and executes the events which are non-blocking (i.e for statements which delay time is not mentioned).

It will execute non- blocking statements continuously without any interruption.

**Single-Threaded Event Loop:**

Node.js uses single Thread to handle all incoming requests. It doesn’t block the thread while waiting for operations. Instead, it sends those blocking requests to Workers Threads and executes after that in meanwhile it will handle the other statements.

**LIBUV:**

Libuv is a library that provides Asynchronous, Non-blocking I/O. It allows node.js to handle multiple operations concurrently, even though it runs on single thread.

**Worker Threads:**

Worker threads allow you to run multiple threads for heavy computations without blocking the main event loop.

**Flow of Execution**:

* When you run the Application, the JavaScript code is executed by V8 engine.
* Then those statements are sent to Event Queue.
* Event Queue stores and executes the non-blocking statements continuously.
* For Blocking Statements, it sends those statements into Worker Threads and remaining statements are executed as it is.
* After the delay then the blocking statements are moved to the event queue and then they get executed.
* Once the task is carried out completely, the response is sent to the Event Loop that in turn sends that response back as output.

**Advantages:**

* Handling multiple concurrent client requests is fast and easy
* Requires fewer resources and memory
* No need for creating multiple threads