

**1. Application**

The application represents the user-written JavaScript code (e.g., server logic, API endpoints).

It communicates with the underlying Node.js engine to execute operations.

**2. V8 JavaScript Engine**

V8 is the engine used by Node.js (developed by Google for Chrome) to execute JavaScript code.

It converts JavaScript into machine code for high performance.

**3. Node.js Bindings (Node API)**

These are the interfaces that allow the JavaScript code in the application to interact with lower-level operations (e.g., file system or network) through Node.js core modules.

Examples: fs, http, net.

**4. Libuv (Asynchronous I/O)**

Libuv is a C library that enables Node.js to handle asynchronous operations.

It provides the following features:

Non-blocking I/O: Handles tasks like file system operations and network requests.

Thread Pool: Delegates blocking operations to a set of worker threads to prevent blocking the event loop.

**5. Event Queue**

The event queue is where incoming requests (events) are stored until they are processed by the event loop.

Examples of events include incoming HTTP requests or file I/O operations.

**6. Event Loop**

The event loop is the core mechanism of Node.js.

It continuously checks the event queue for new tasks and delegates them for processing.

Non-blocking operations are executed immediately, while blocking operations are sent to worker threads.

**7. Blocking Operation**

Operations that take time to complete (e.g., database queries, file reading) are classified as blocking.

Instead of blocking the main thread, these tasks are offloaded to worker threads by libuv.

**8. Worker Threads**

A pool of threads managed by libuv to handle heavy or blocking operations like:

File system access

Network requests

CPU-intensive tasks

Once completed, the results are passed back to the event loop via callbacks.

**9. Executive Callback**

After a worker thread finishes a task, the result is processed in the event loop through a callback function.

This allows the application to respond to requests asynchronously.