**Nodejs – Event driven architecture**

Architecture

Built on googles V8 browser engine, Event loops, Thread pools, Event queues

Event loops:

🡪 Single thread but uses event loop for saving and triggering callbacks based on kernel responses --> seems like multi-threading

🡪 Node starts - initiates event loop - runs script in REPL (Read-Eval-Print-Loop) - schedule timers - stores callbacks - process.nexttick()

* --> timers: this phase executes callbacks scheduled by setTimeout() and setInterval().
* pending callbacks: executes I/O callbacks deferred to the next loop iteration.
* idle, prepare: only used internally.
* poll: retrieve new I/O events; execute I/O related callbacks (almost all with the exception of close callbacks, the ones scheduled by timers, and setImmediate()); node will block here when appropriate.
* check: setImmediate() callbacks are invoked here.
* close callbacks: some close callbacks, e.g. socket.on('close', ...).

Blocking and Non blocking I/O

🡪 Js execution must wait until a particular line of code gets executed – Blocking

🡪 synchronous and asynchronous calls. Concurrency and throughput

🡪Dangers (unexpected behavior should not be triggered first as synchronous before asynchronous process)

Do not block event loop

Timers

🡪Set time out, interval, immediate

🡪Clear timeout, interval, immediate

🡪timeout unref and ref, leaving behind timeouts