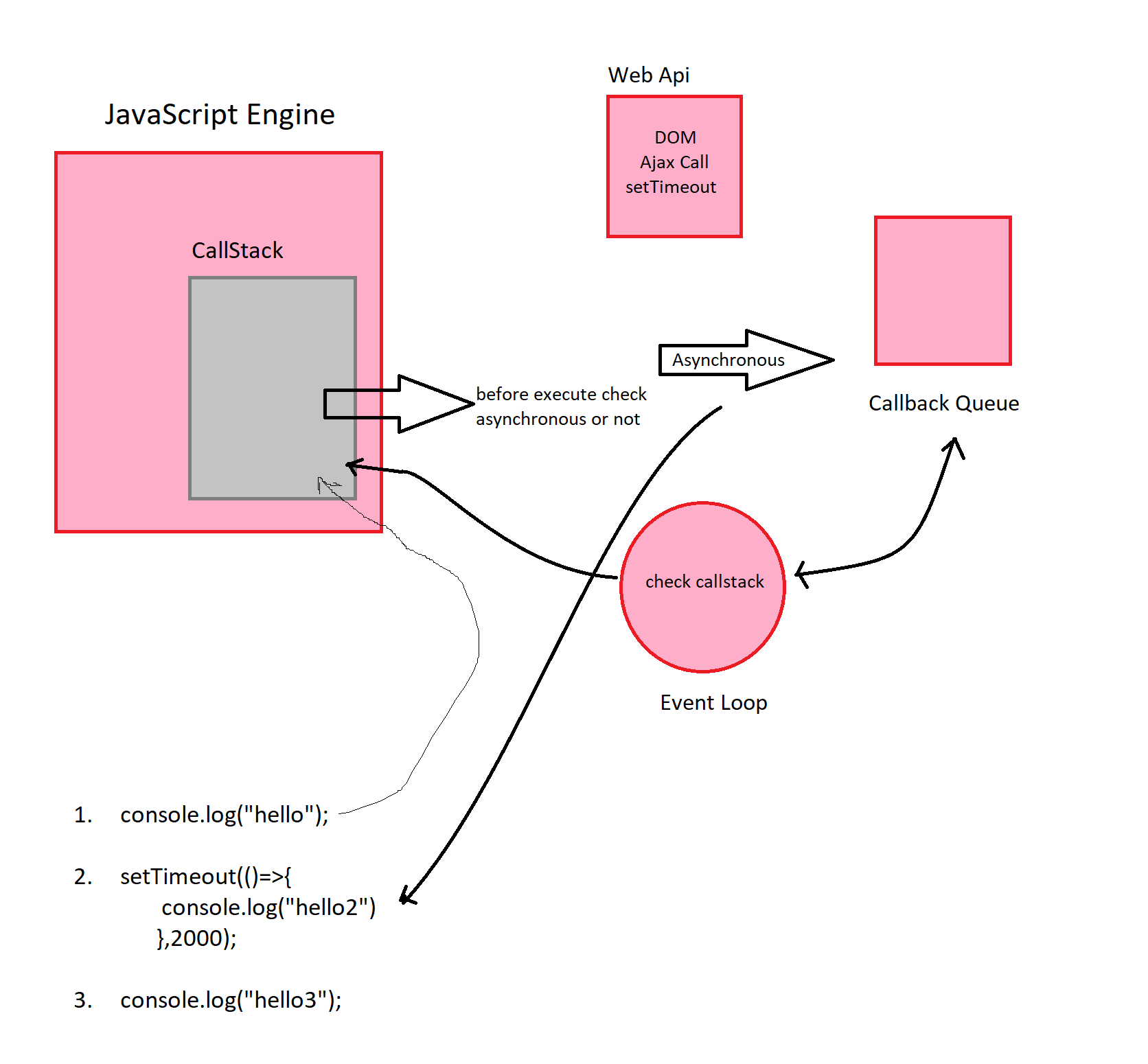
**Asynchronous**

* In Asynchronous, Second task do not wait to finish First Task.

Event Loop and Non Blocking :-



**Single Threaded**

Add(4,3) 5ms

Subtract(7,5) 5ms One after Another

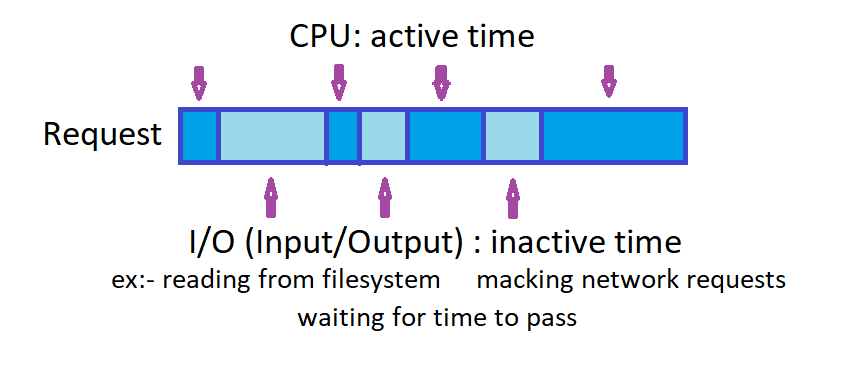
Divide(10,2) 5ms

Total 15ms

Saves from deadlock

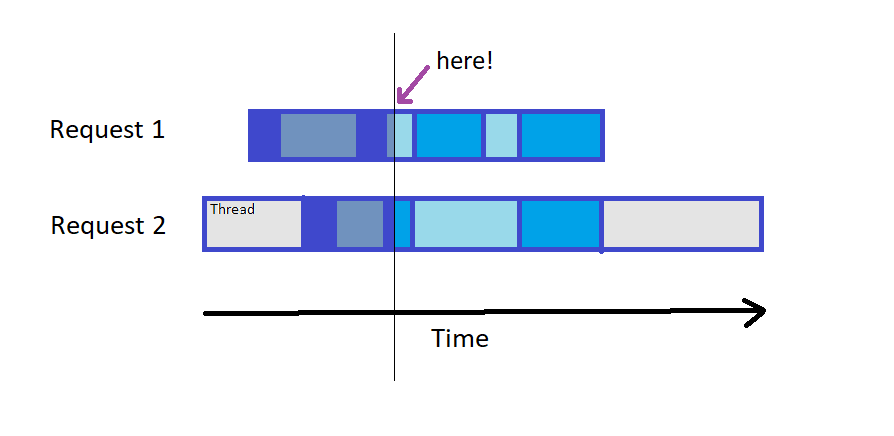
**I/O**

* A Request can be divided into CPU work and I/O work.
* Servers handle requests using threads, which each independently handle requests.
* Nonblocking I/O allows a thread to suspend a request while it’s performing I/O work on a different requests.
* Non-blocking I/O allows Node.js to efficiently handle IO-heavy workload with only one thread.

****

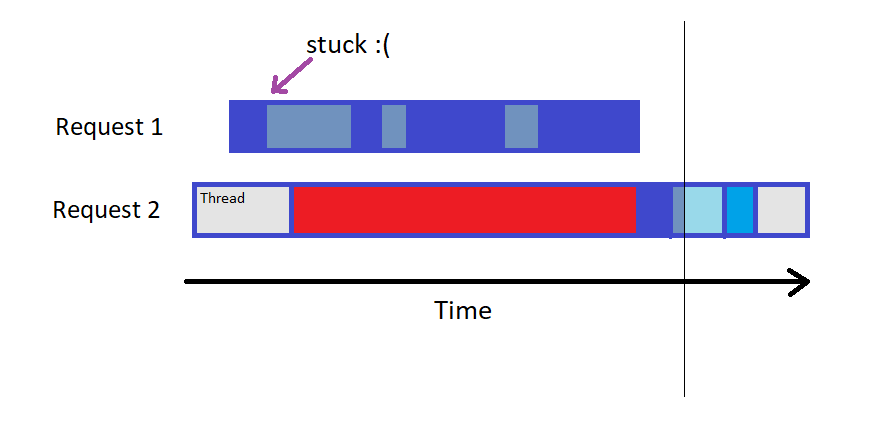
**Non blocking I/O**

While a request is doing I/O, the thread is not stuck waiting for the request to finish.



**Blocking I/O**

Thread is stuck, or blocked, waiting for I/O to finish.



**V8 Engine**

