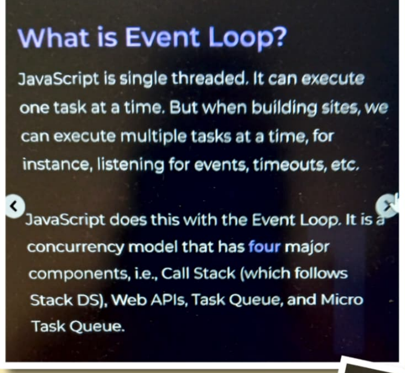
JS

Event Loop in JavaScript

JS

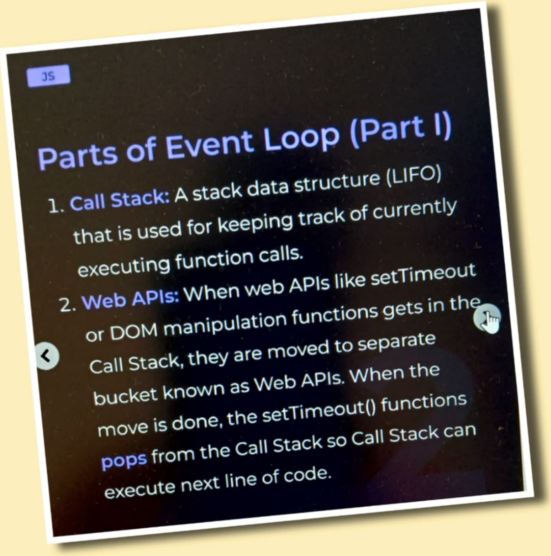


JS

What is Event Loop?

JavaScript is single threaded. It can execute one task at a time. But when building sites, we can execute multiple tasks at a time, for instance, listening for events, timeouts, etc.

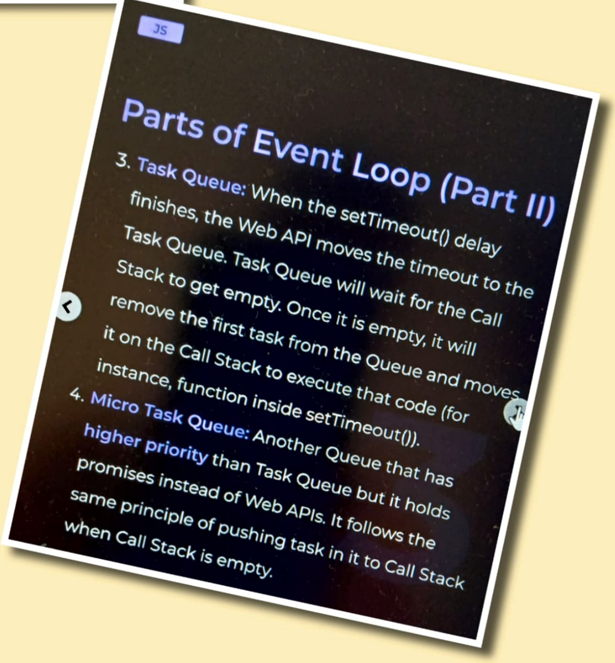
JavaScript does this with the Event Loop. It is a concurrency model that has **four** major components, i.e., Call Stack (which follows Stack DS), Web APIs, Task Queue, and Micro Task Queue.



JS

Parts of Event Loop (Part I)

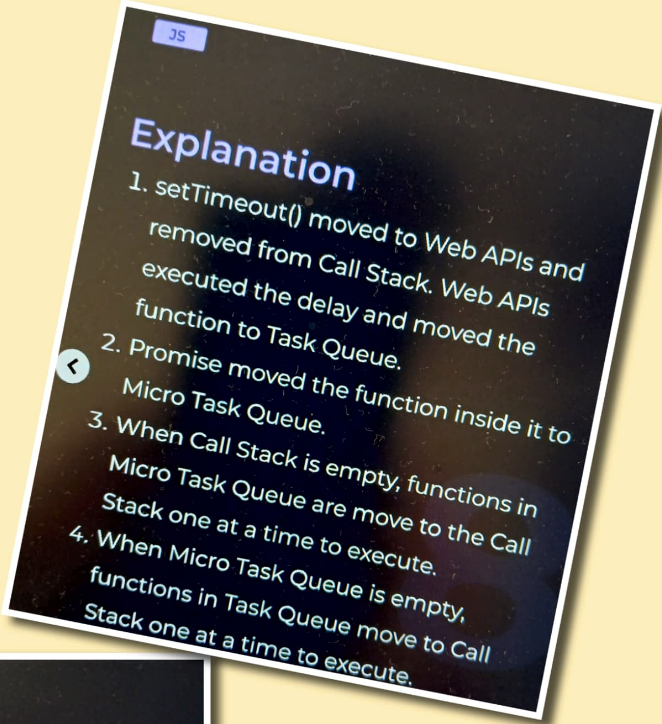
1. **Call Stack:** A stack data structure (LIFO) that is used for keeping track of currently executing function calls.
2. **Web APIs:** When web APIs like setTimeout or DOM manipulation functions gets in the Call Stack, they are moved to separate bucket known as Web APIs. When the move is done, the setTimeout() functions **pops** from the Call Stack so Call Stack can execute next line of code.



JS

Parts of Event Loop (Part II)

3. **Task Queue:** When the setTimeout() delay finishes, the Web API moves the timeout to the Task Queue. Task Queue will wait for the Call Stack to get empty. Once it is empty, it will remove the first task from the Queue and moves it on the Call Stack to execute that code (for instance, function inside setTimeout()).
4. **Micro Task Queue:** Another Queue that has **higher priority** than Task Queue but it holds promises instead of Web APIs. It follows the same principle of pushing task in it to Call Stack when Call Stack is empty.

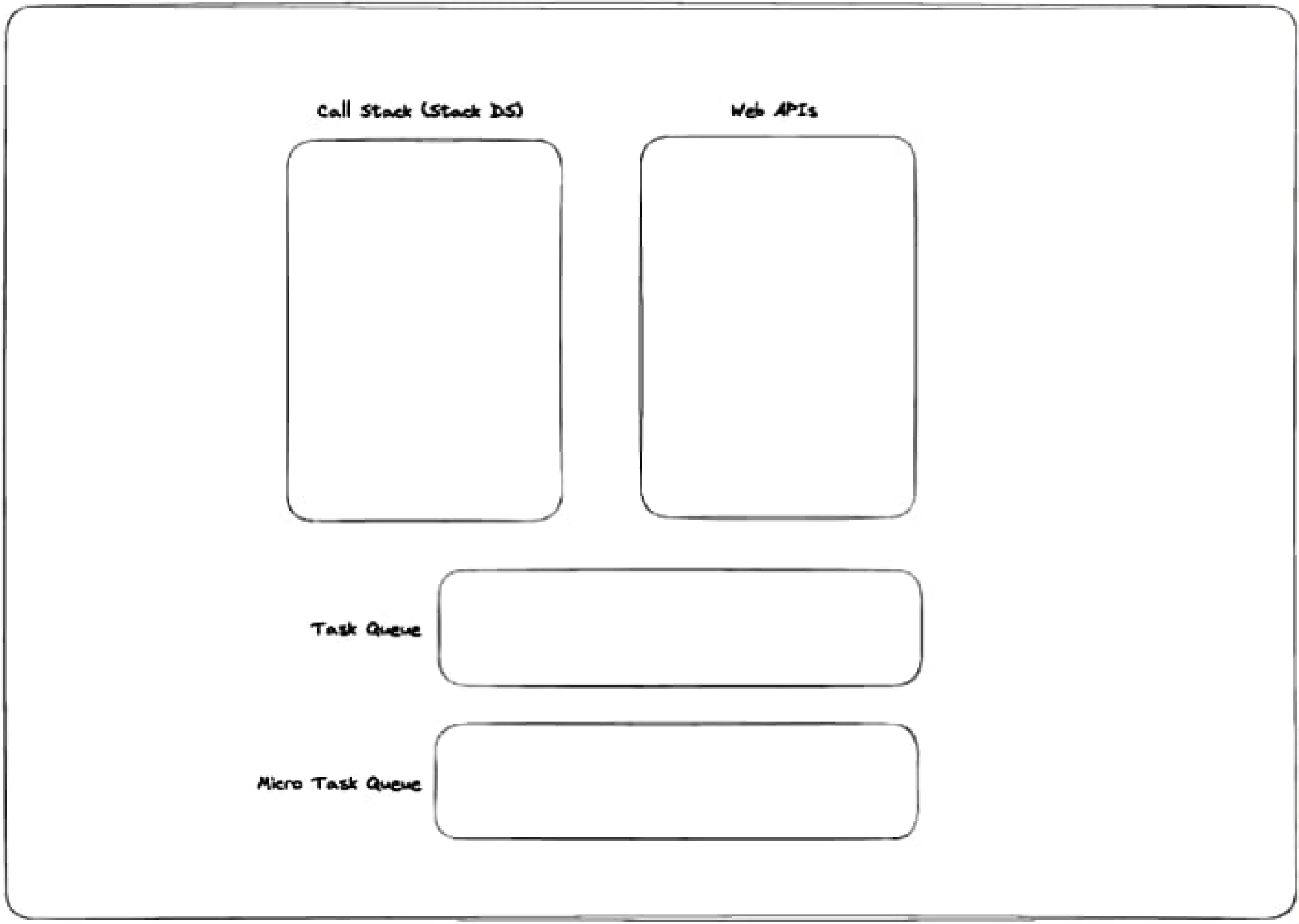


JS

Explanation

1. setTimeout() moved to Web APIs and removed from Call Stack. Web APIs executed the delay and moved the function to Task Queue.
2. Promise moved the function inside it to Micro Task Queue.
3. When Call Stack is empty, functions in Micro Task Queue are move to the Call Stack one at a time to execute.
4. When Micro Task Queue is empty, functions in Task Queue move to Call Stack one at a time to execute.

Event Loop in JavaScript



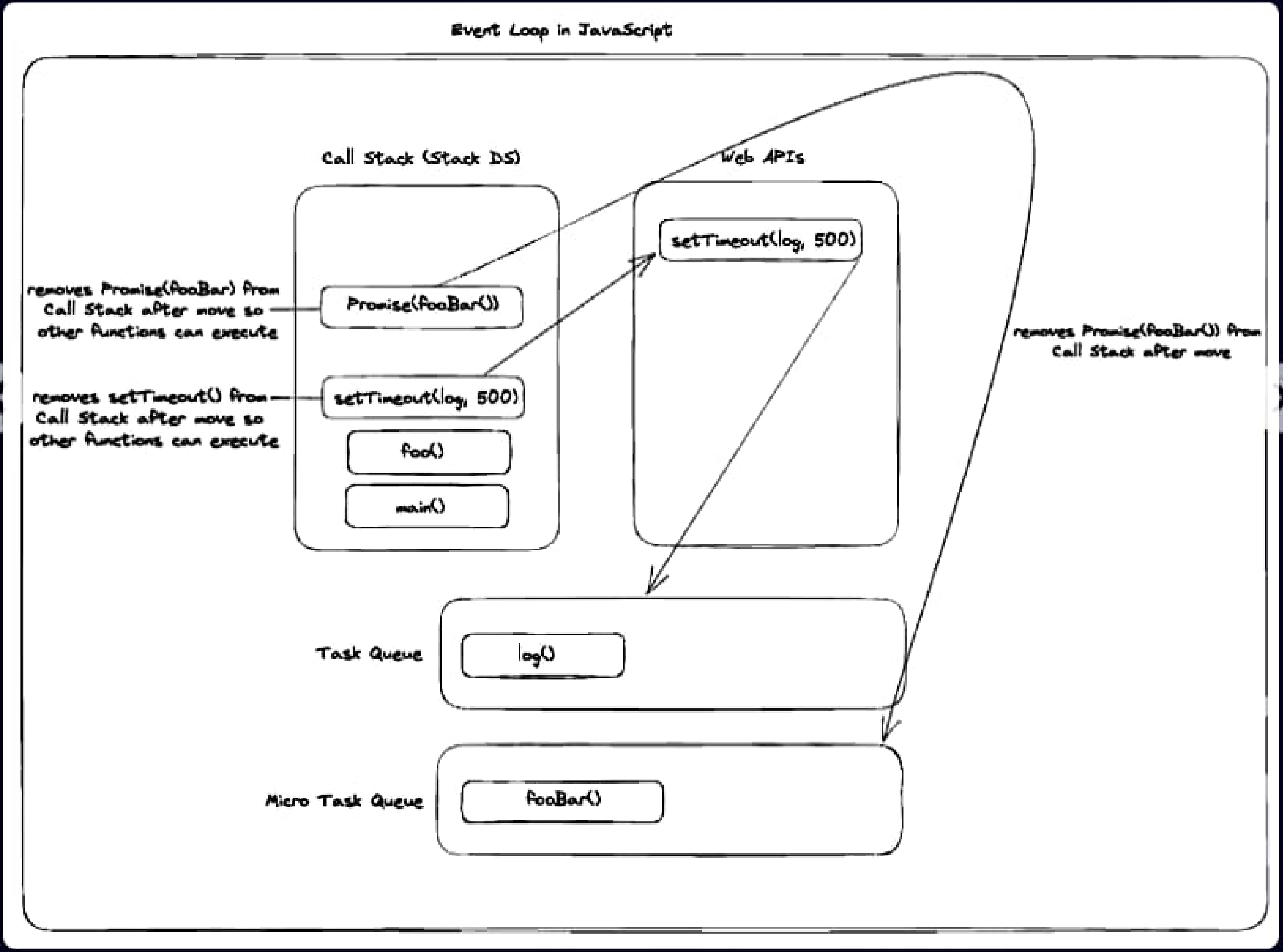
Call Stack (Stack DS)

Web APIs

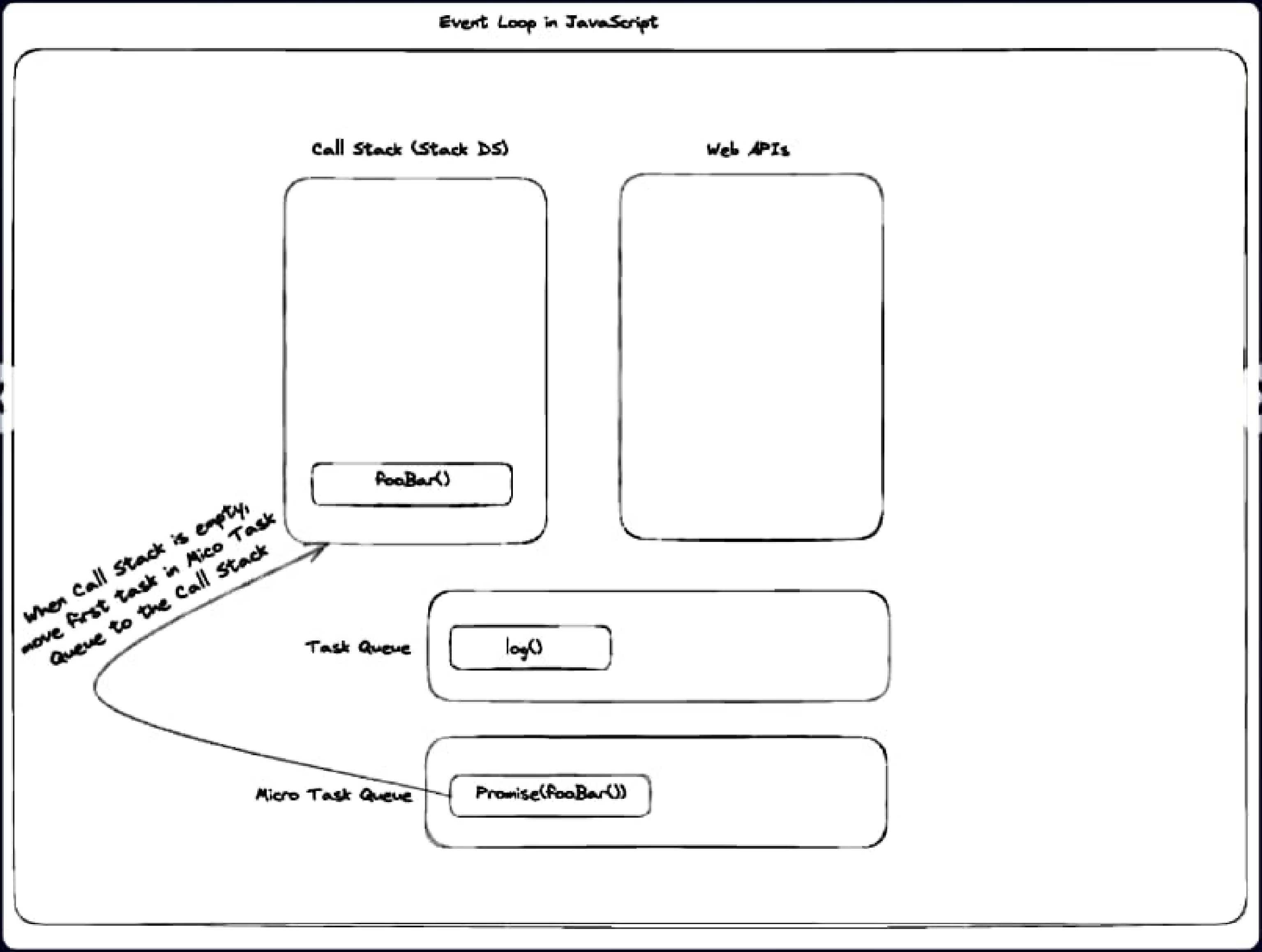
Task Queue

Micro Task Queue

Event Loop Example (Part 1)

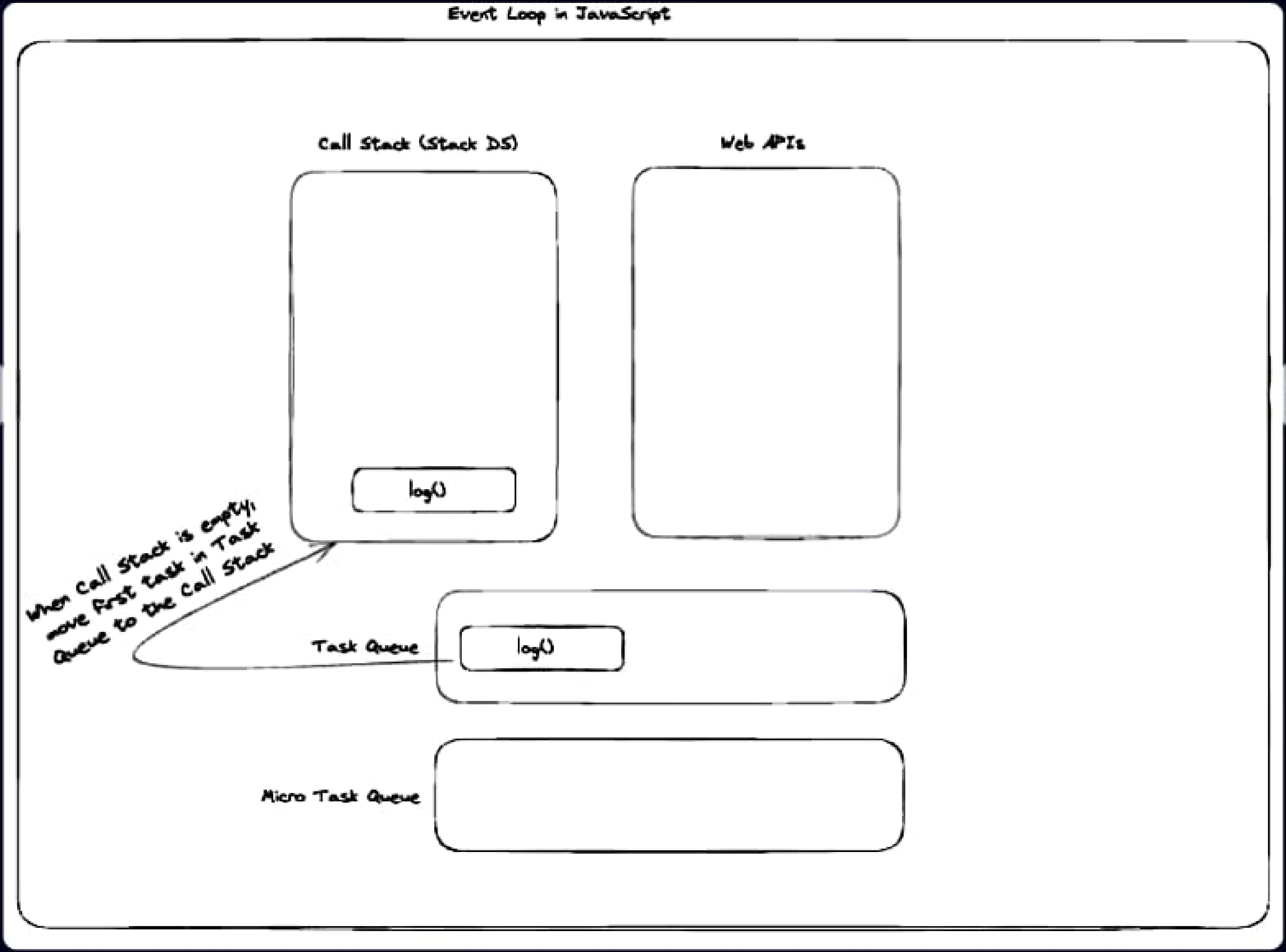

Event Loop in JavaScript
Call Stack (Stack DS)
Web APIs
removes Promise(fooBar) from Call Stack after move so other functions can execute
Promise(fooBar())
removes setTimeout() from Call Stack after move so other functions can execute
setTimeout(log, 500)
foo()
main()
setTimeout(log, 500)
removes Promise(fooBar()) from Call Stack after move
Task Queue
log()
Micro Task Queue
fooBar()

Event Loop Example (Part 2)



When Call Stack is empty, move first task in Mico Task Queue to the Call Stack

Event Loop Example (Part 3)


Event Loop in JavaScript
Call Stack (Stack DS)
Web APIs
log()
When Call Stack is empty, move first task in Task Queue to the Call Stack
Task Queue
log()
Micro Task Queue