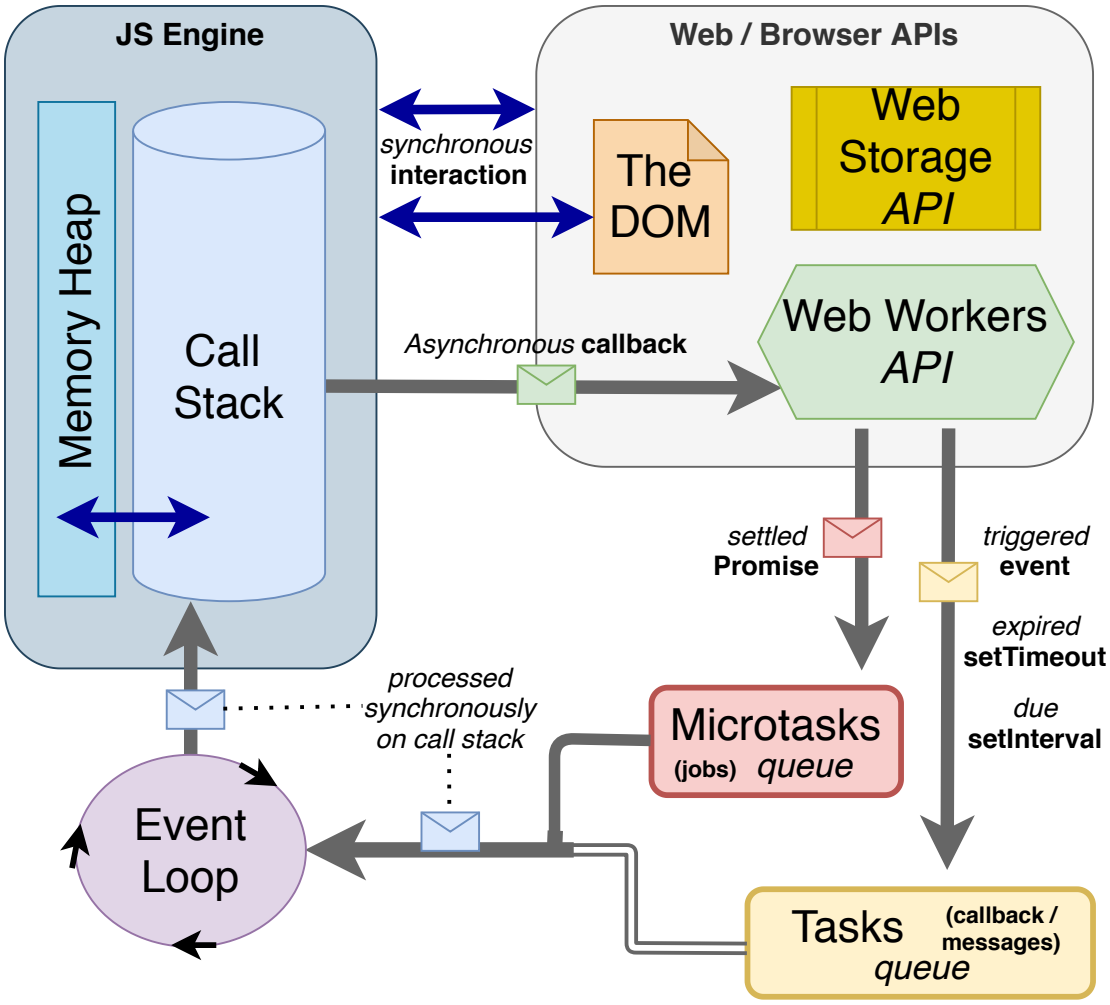
The JavaScript Runtime Environment v0.5



Source: Vahid Dejwakh, 2021

Explanation

CS — The *single-threaded* JS engine (**JSE**) pushes every new function execution context onto the **Call Stack**, then pops it off when returning from it.

D — The JSE may also synchronously interact with **the DOM**.

W — Every *asynchronous* callback (✉) created within an execution context is immediately sent to be handled by the **Web Workers API**, which processes them *outside* the JSE.

MQ / TQ — The Web Workers API sends *settled* **promises** to the back of the **Microtasks** queue, and any *triggered* **events** (e.g. clicks, etc) or *expired* **callbacks** to the back of the **Tasks** queue.

EL — The **Event Loop** continuously cycles. Whenever there's either: a **1)** break *in* or **2)** no more frames left *on* the call stack, the EL first completes items in the **Microtasks** queue, from front to back. The EL then completes items in the **Tasks** queue, also from front to back. Any new *microtask* created by a *microtask* will be executed before any *tasks*.