

# ÔN LẠI: JAVASCRIPT RUNTIME

👉 **Concurrency model:** Cách Javascript xử lý nhiều tác vụ diễn ra cùng một lúc.

“Container” gồm tất cả các phần cần thiết để thực thi JavaScript code

“Trung tâm” của runtime

API cấp cho engine

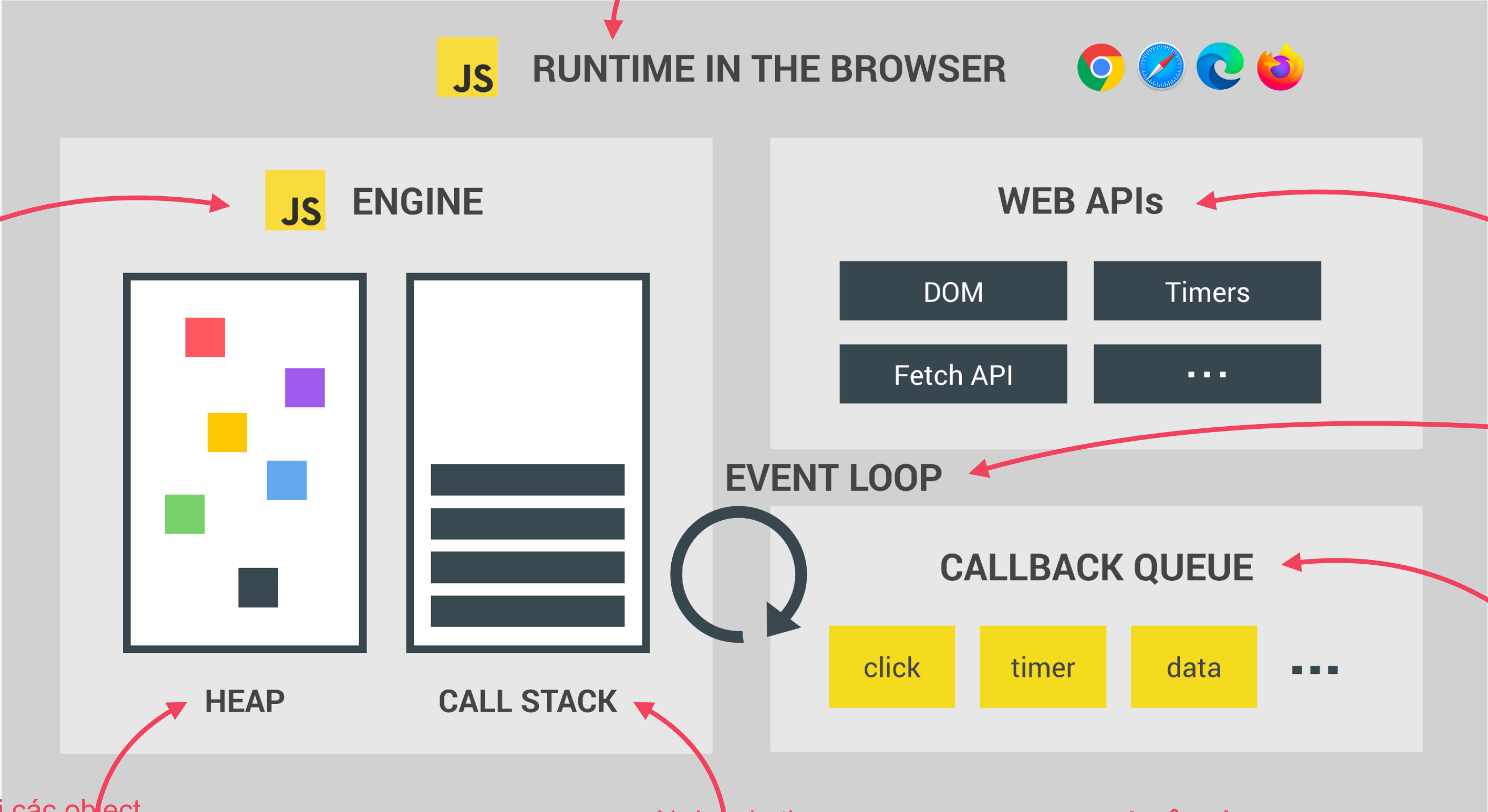
Gửi callbacks từ queue tới call stack

Các hàm callback sẵn sàng để thực thi (từ các event)

Nơi các object được lưu trữ trong bộ nhớ

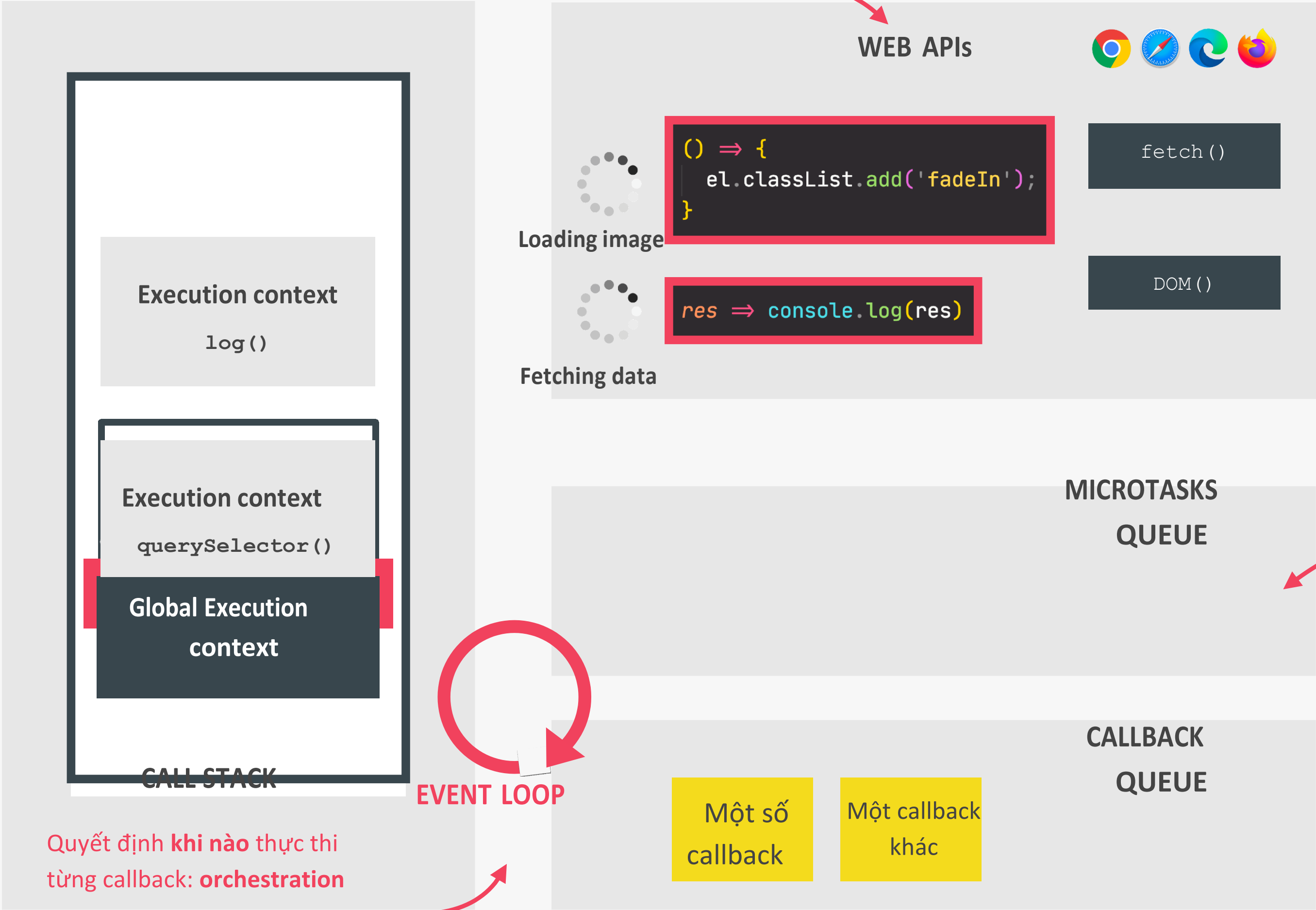
Nơi code thực sự được thực thi

Chỉ **MỘT** luồng thực thi. Không thực hiện đa tác vụ!



# CÁCH ASYNCHRONOUS JAVASCRIPT HOẠT ĐỘNG

Nơi chạy các tác vụ asynchronous



```
el = document.querySelector('img');
el.src = 'dog.jpg';
el.addEventListener('load', () => {
  el.classList.add('fadeIn');
});

fetch('https://someurl.com/api')
  .then(res => console.log(res));

// More code ...
```

Giống callback queue, nhưng các callback liên quan tới **promise**. Có **ưu tiên** so với callback queue!



Asynchronous code được thực thi như thế nào theo cách **non-blocking**, nếu **chỉ có một luồng** thực thi trong engine?

