

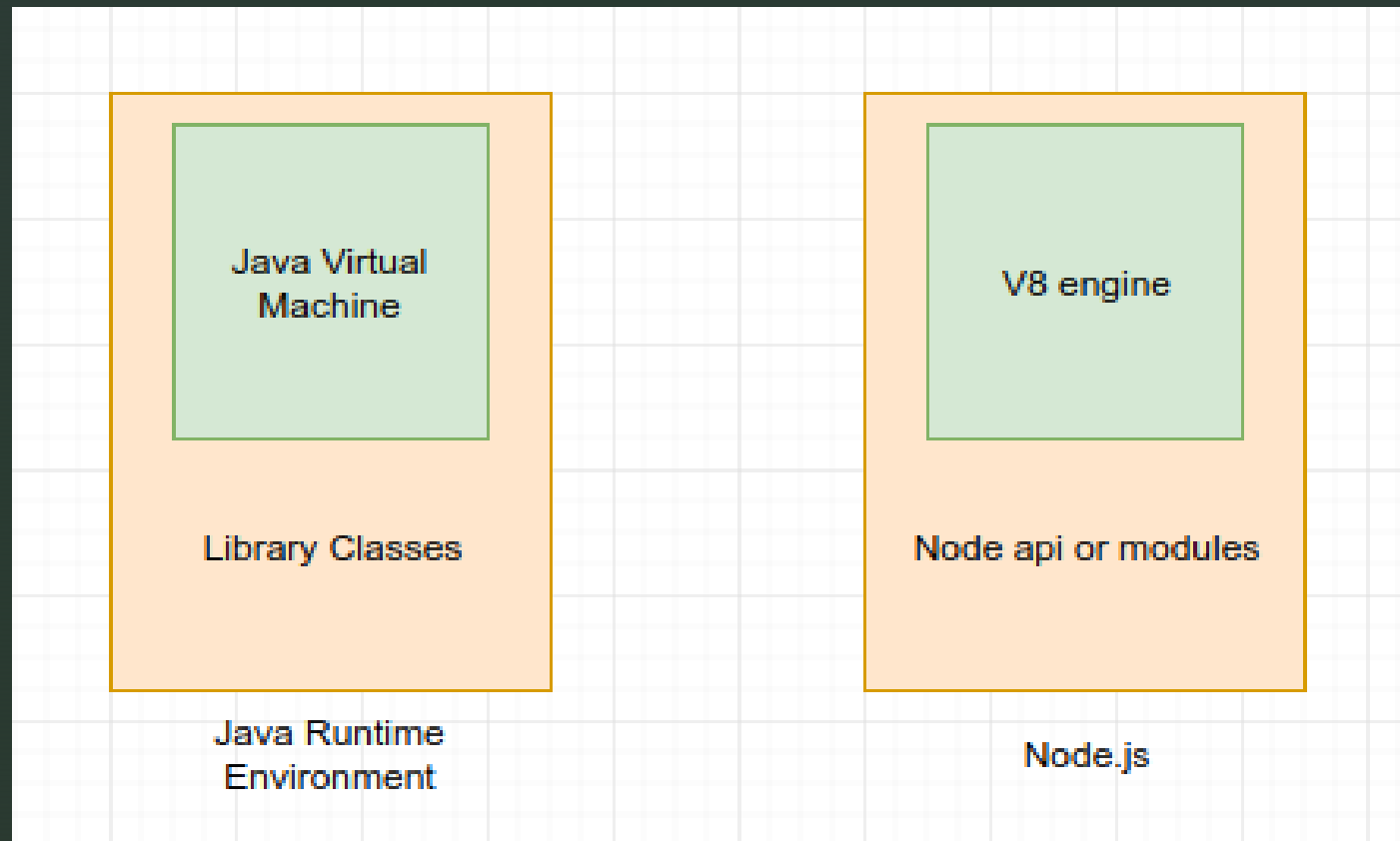


JavaScript Runtime  
On Server Side

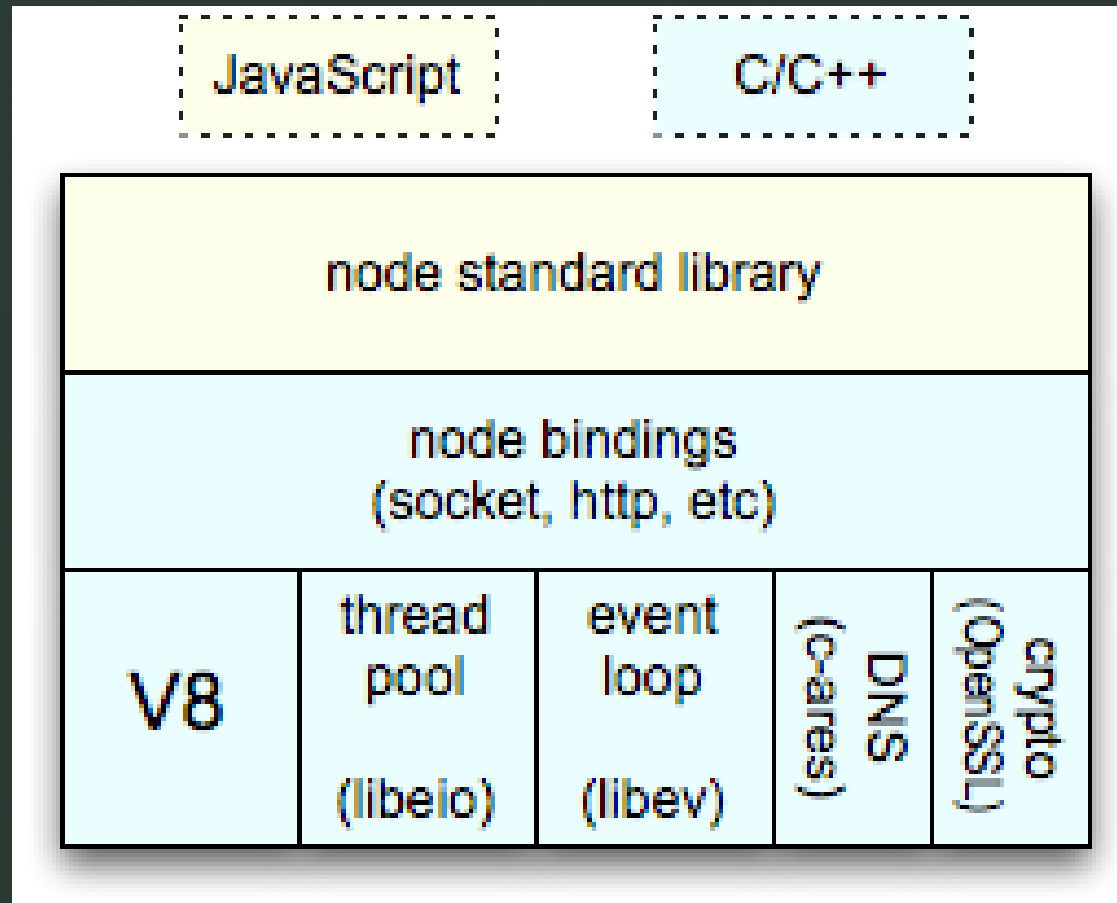
# Architecture



# Tổng quan



# Tổng quan



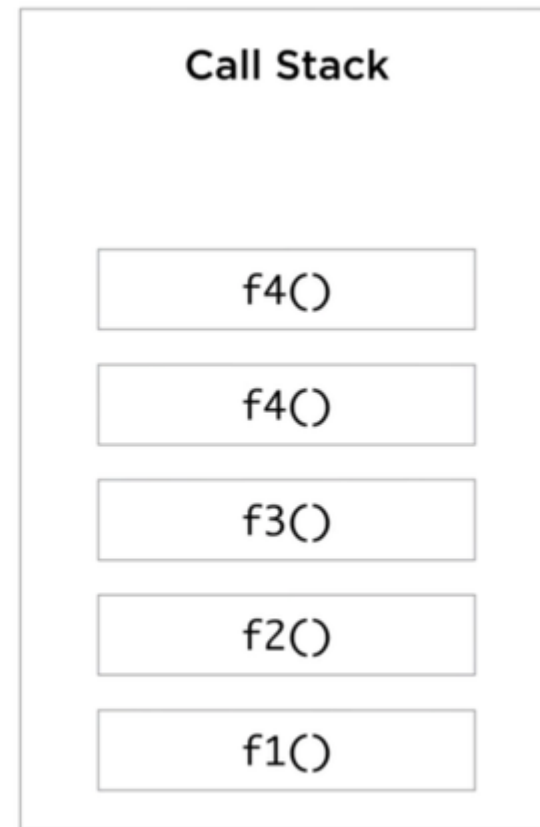
# Call Stack

- Call Stack là một phần của V8, được sử dụng để lưu vết những functions được gọi.
- Mỗi khi function được gọi thì các function được đưa vào Call Stack & chờ thực thi
- Chỉ có MỘT Call Stack trong Node process → Nếu call stack bị đầy (stuck)?



# Call Stack

```
const f1 = () => { f2(); };  
const f2 = () => { f3(); };  
const f3 = () => { f4(); };  
const f4 = () => { f4(); };
```



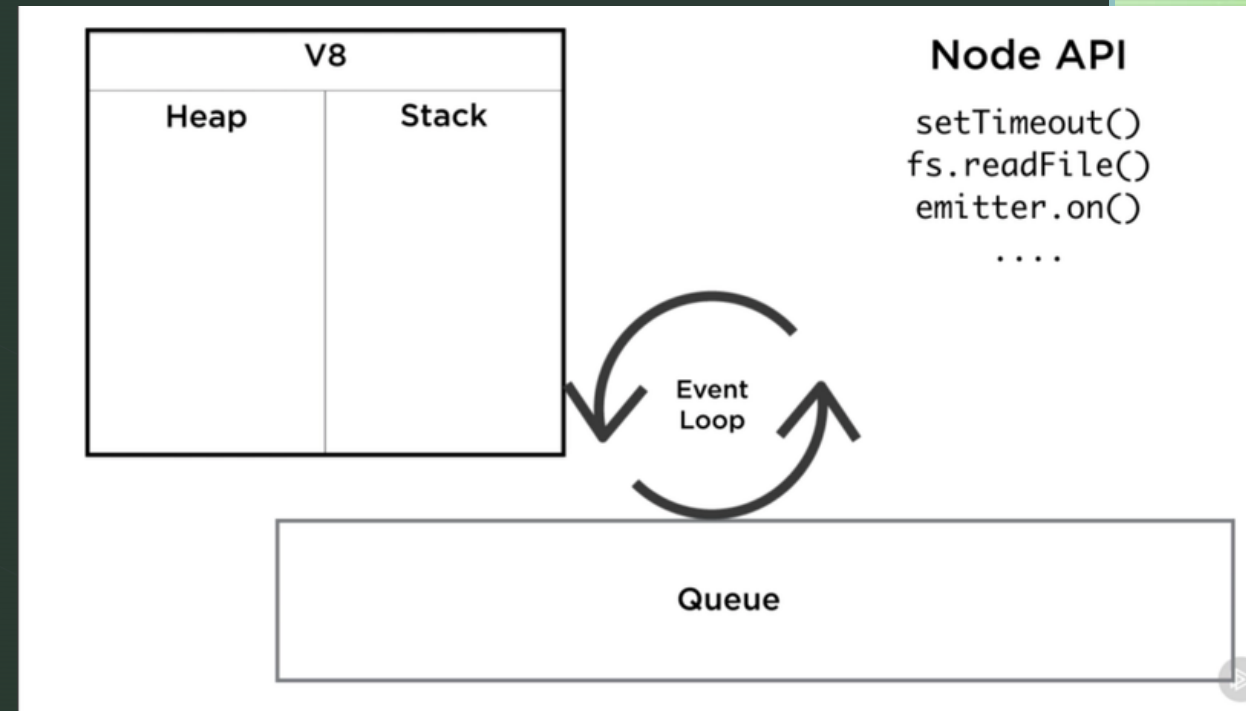
# Event Loop & Callback Queue

- Event loop được cung cấp bởi thư viện libuv, và nó không phải là một phần của V8
- Event loop là một entity xử lý các external event và chuyển chúng thành việc gọi callback
- Nó là một vòng lặp mà nhặt lấy cái event từ event queue, sau đó đẩy các callback của chúng vào Call stack



# Event Loop & Callback Queue

- Node API là những function kiểu như `setTimeout` hay `fs.readFile`
- Node API phải là hàm của Javascript, chúng chỉ là những hàm đc cung cấp bởi Node



Event loop đứng ở giữa và hoạt động kiểu như một người tổ chức





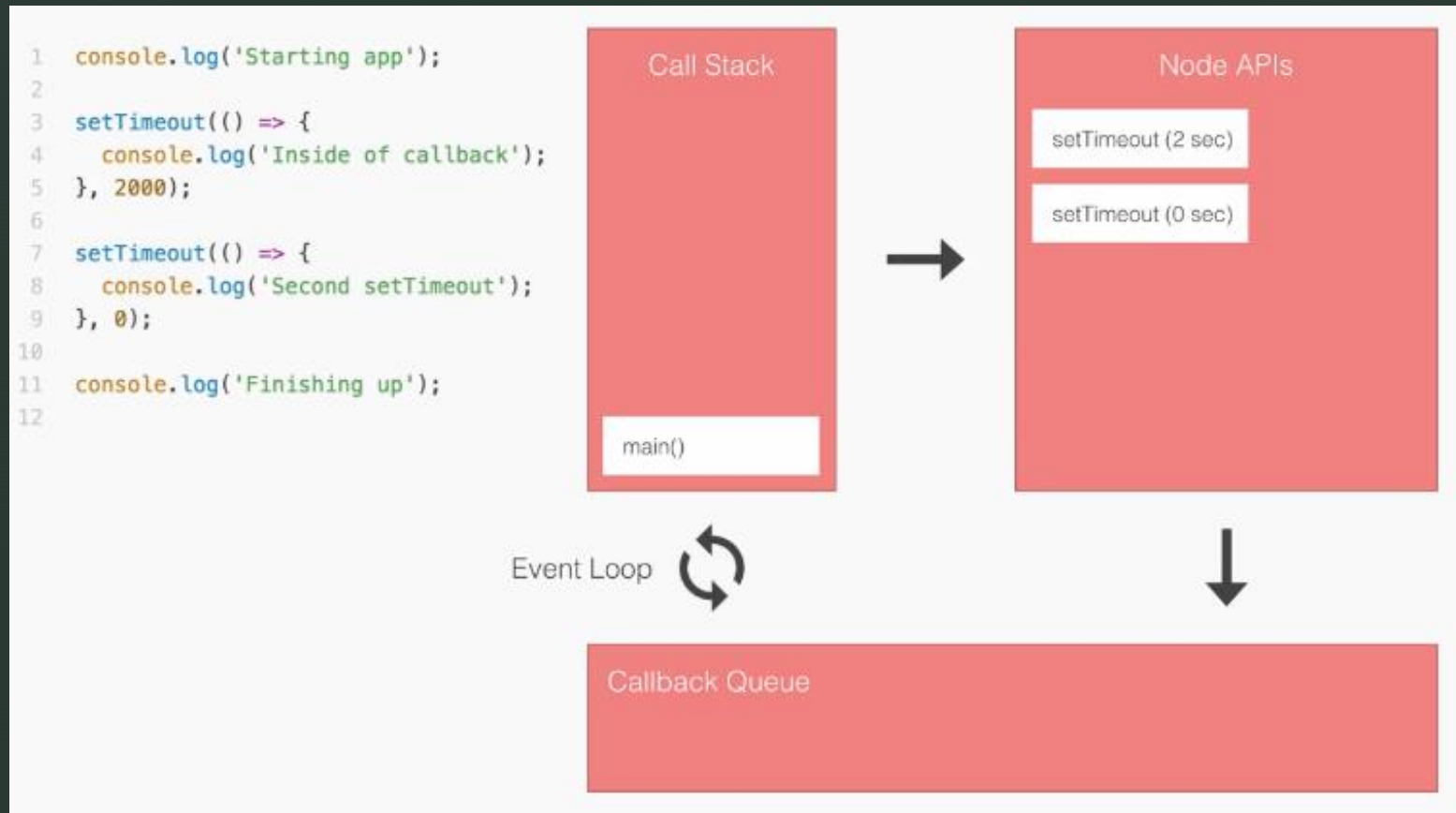
# NodeJS làm gì khi Call Stack trống?

- Đơn giản là Node sẽ exit
- Để giữ cho Node process chạy, ta cần phải đặt một cái gì đấy vào đâu đó ở event queue
- Ví dụ, khi chạy một timer hoặc một HTTP server, về căn bản, chúng ta đã bảo event loop tiếp tục chạy và kiểm tra những sự kiện đó





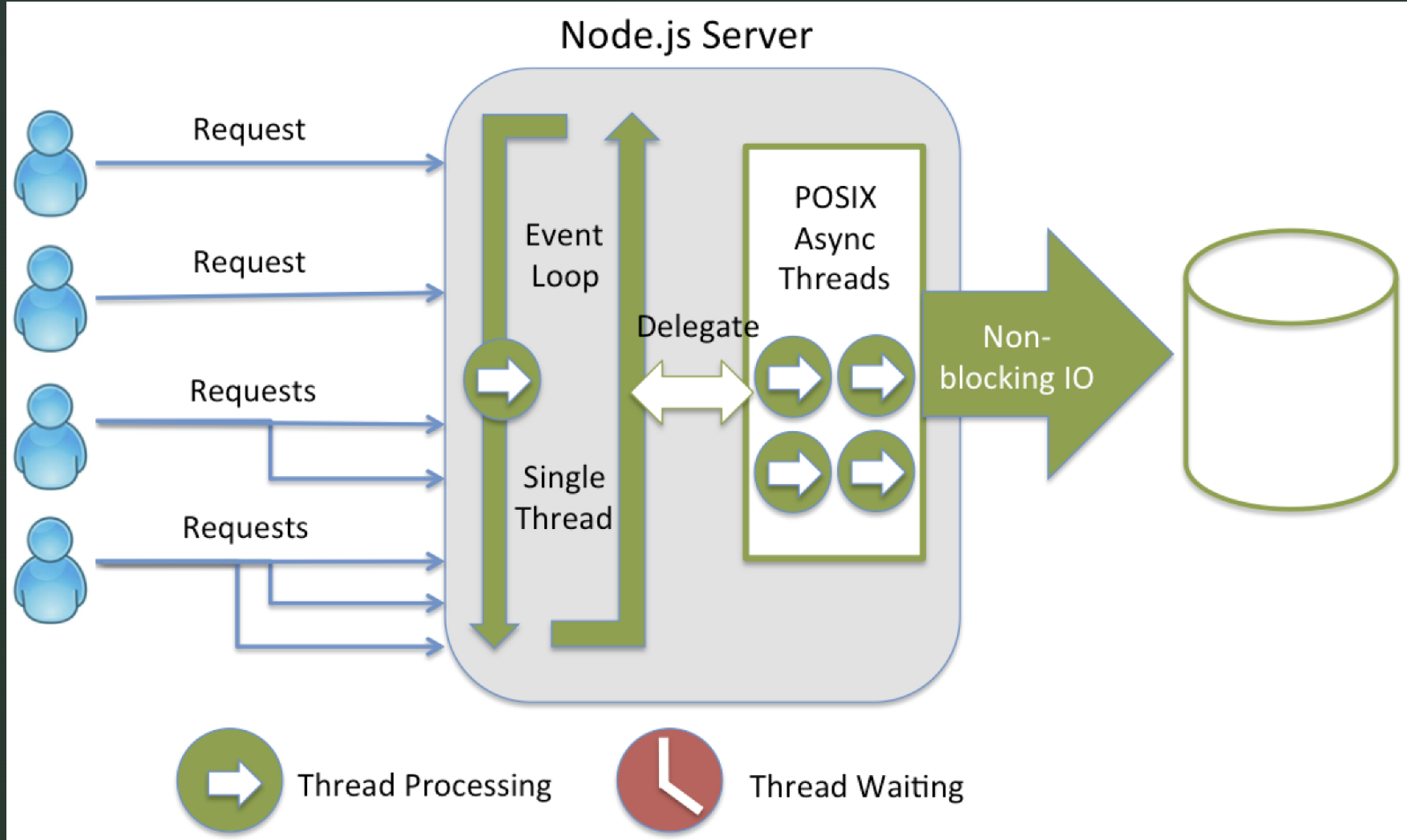
# Node Process Running + Demo



<http://latentflip.com/loupe/>

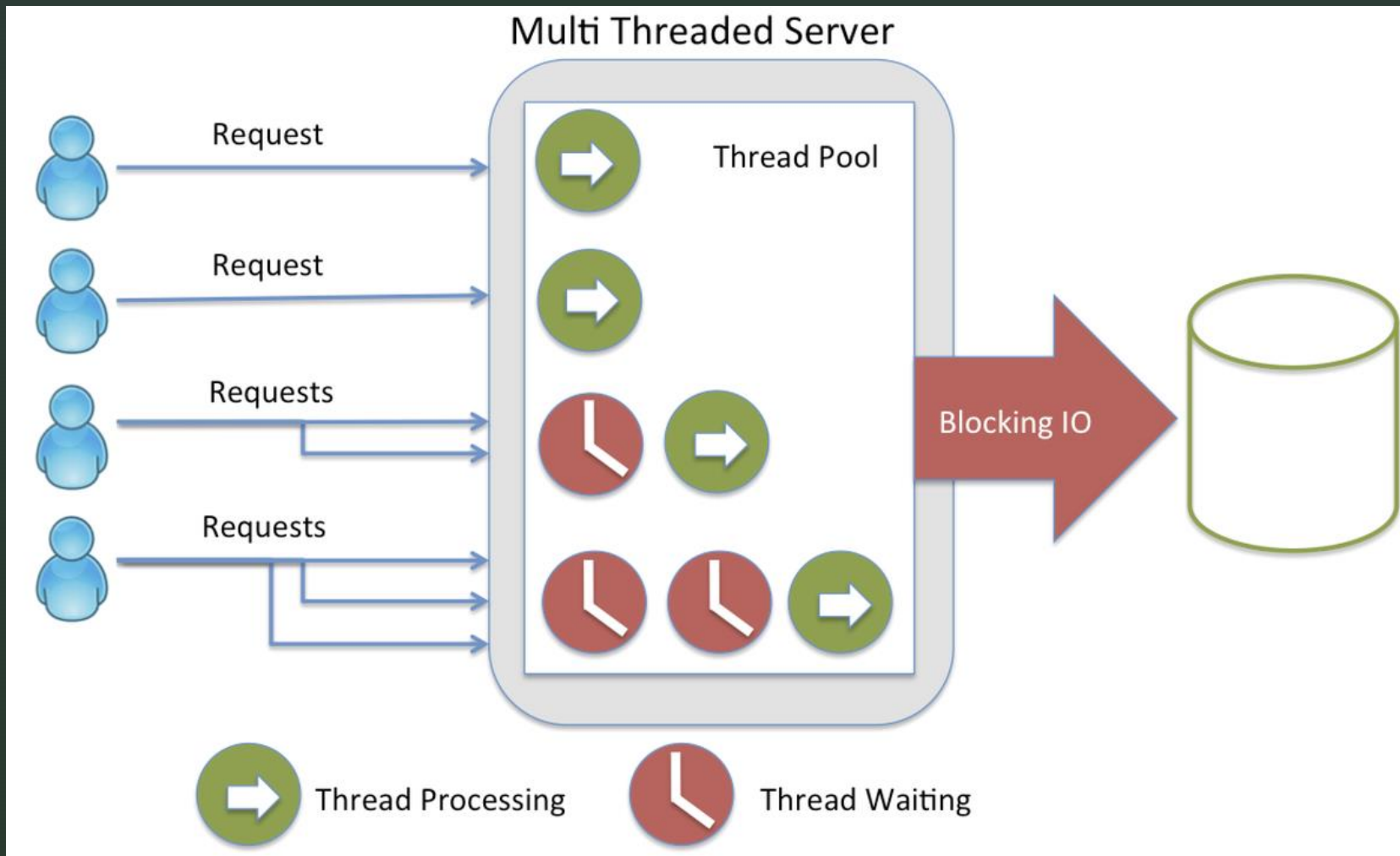


# Non-Blocking Architecture



# Non-Blocking Architecture

Node process nhận nhiều hơn size của Thread Pool?



# Có thể chạy Node mà không cần V8 ?

- Cần có 02 VM để chạy Node process
- Chakra là một VM giống với V8
- <https://github.com/nodejs/node-chakracore>



# References

- <https://nodejs.org/en/about/>
- <http://latentflip.com/loupe/>
- <https://github.com/nodejs/node-chakracore>

