

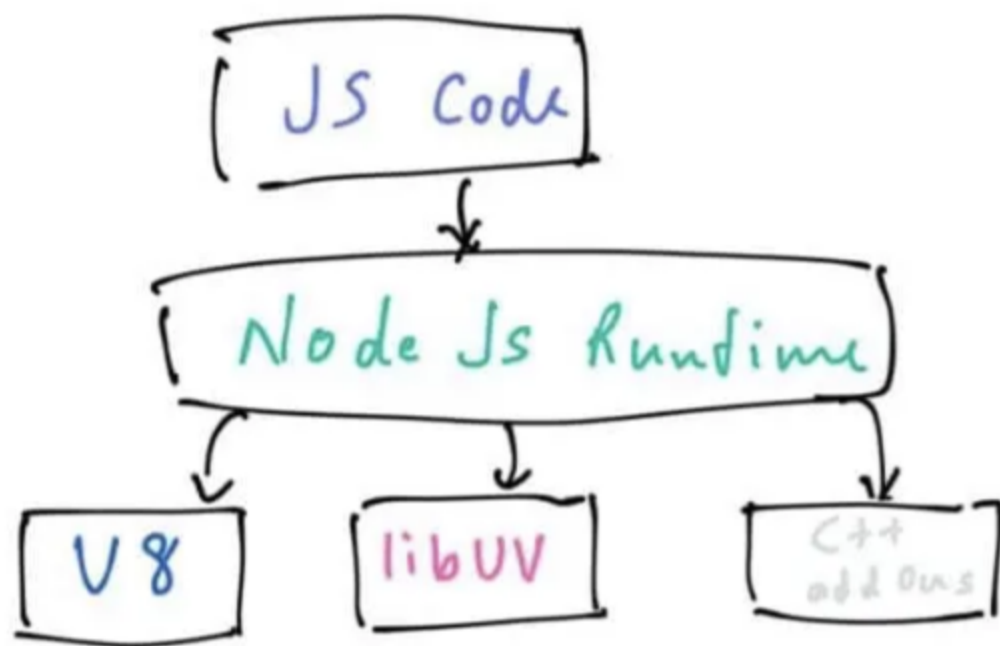
# Memory leak

Heap out of memory или утечки памяти в Node.js

## План доклада

- Краткий экскурс в архитектуру Node.js
- Утечки памяти - определение, кейсы и дебаггинг

# Архитектура Node.js - 1й вариант



**V8**





# V8

- Разработан датчанами из Google для повышения производительности и масштабируемости
- Написан на C++
- Благодаря нему можно использовать Node.js для написания обычных приложений (VScode, Postman)



## А что было до V8

- JS-код был только клиентским и мог воспроизводиться только в браузере

## Поэтому, он полезен ещё тем, что -

- Предоставляет функции, связанные с запуском JS-файлов (компилирует и выполняет)
- Обработывает стек вызовов\*
- Управляет выделением памяти для объектов JS
- Сбор мусора

**libuv**

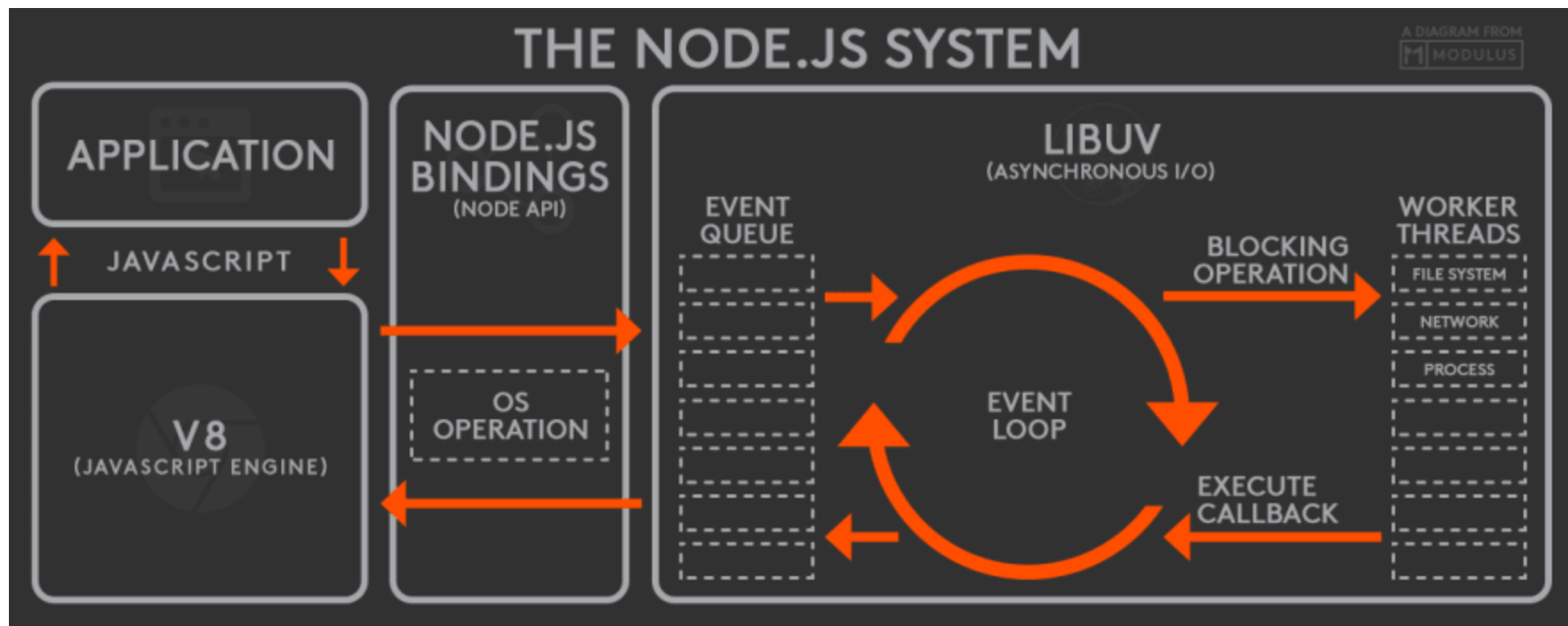


**libuv**

# libuv

- Сторонняя библиотека
- Написана на C
- Предоставляет доступ к операционной системе
- Предлагает функции, самые важные из которых - Event loop, Streaming, FileAsync, Pipes, ChildProcesses

**Как это работает всё вместе?**



## **Доп.вопрос - чем эта модель отличается от асинхронности в других ЯП?**

- Неблокирующий ввод-вывод, обратный вызов
- Go - параллелизм с доступом ко всем ядрам ЦП и общением через каналы



## V8 и память

- V8 - движок
- В его ЖЦ есть операции выделения памяти под собственные объекты и структуры данных
- Но нас интересуют `stack` и `heap` (стек и куча)

## V8 и память. Стек

- Это структура данных
- Список элементов, которые обрабатываются и хранятся по принципу LIFO
- В стеке JS хранит примитивные типы данных  
(string/number/boolean/null/undefined) и ссылки на всё остальное
- Стек - статическое\* выделение памяти

## V8 и память. Куча

- Это дерево
- Хранятся элементы, размер которых неизвестен заранее
- V8 выделяет на них память по мере необходимости
- Куча - динамическое\* выделение памяти

**Доп.вопрос - чем этот подход в работе с памятью отличается от других ЯП?**

## V8 и мусор

- Объекты-сироты из кучи, на которые никто не ссылается из стека (прямо или косвенно) являются мусором
- Сборщик мусора освобождает неиспользуемую память для того, чтобы повторно её использовать
- В V8 СМ'ы поколенческие, используются два этапа, но оба stop the world

## **Доп.вопрос - как справляются языки без СМ?**

- Ручное управление памятью
- Rust - области видимости

## Утечки памяти