# V8引擎源码分析

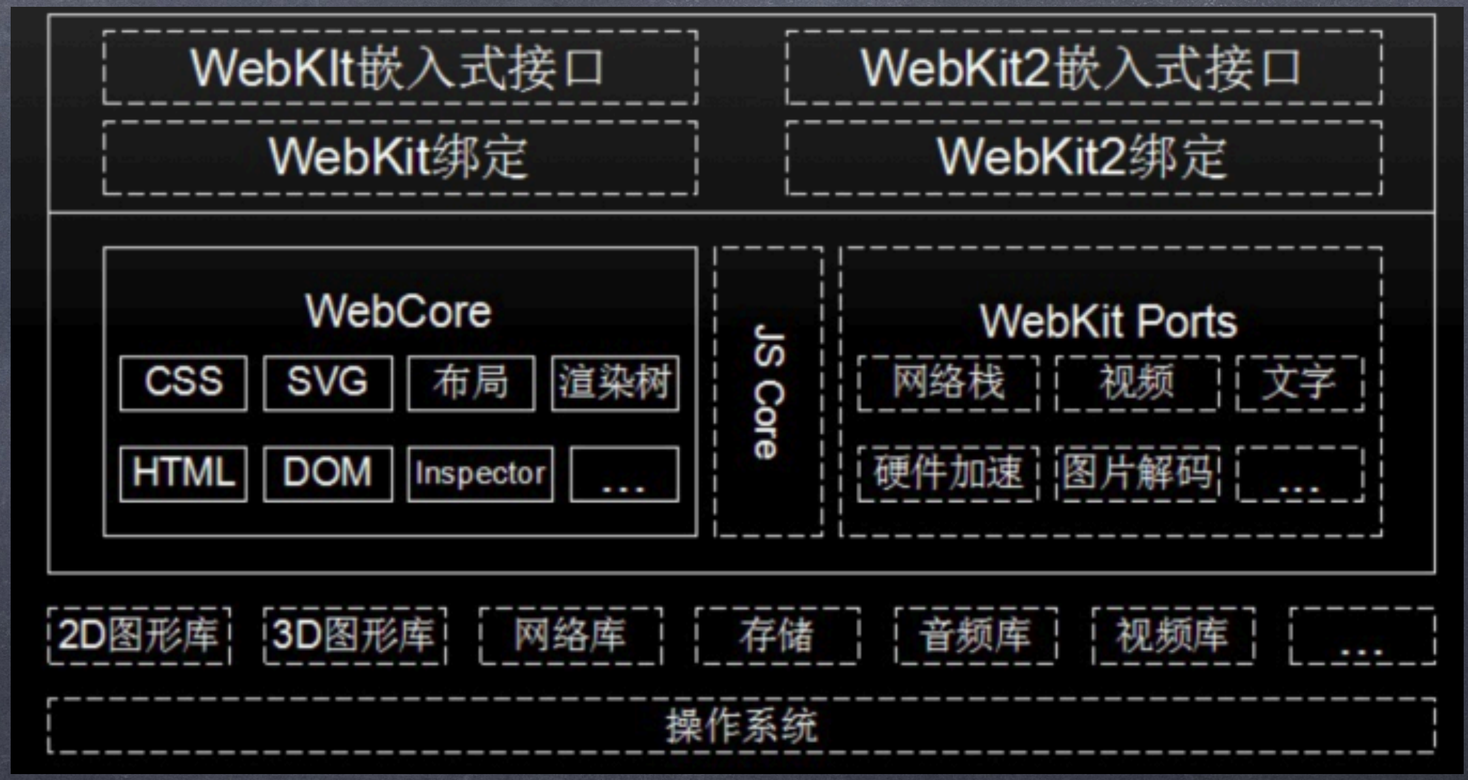
京程一灯
http://www.yidengxuetang.com

# V8引擎源码分析

- 渲染引擎及 Webkit 体系结构
- Node.js 中的 V8 引擎
- V8 源码总览
- V8 源码分析

# 渲染引擎及 WebKit 体系结构

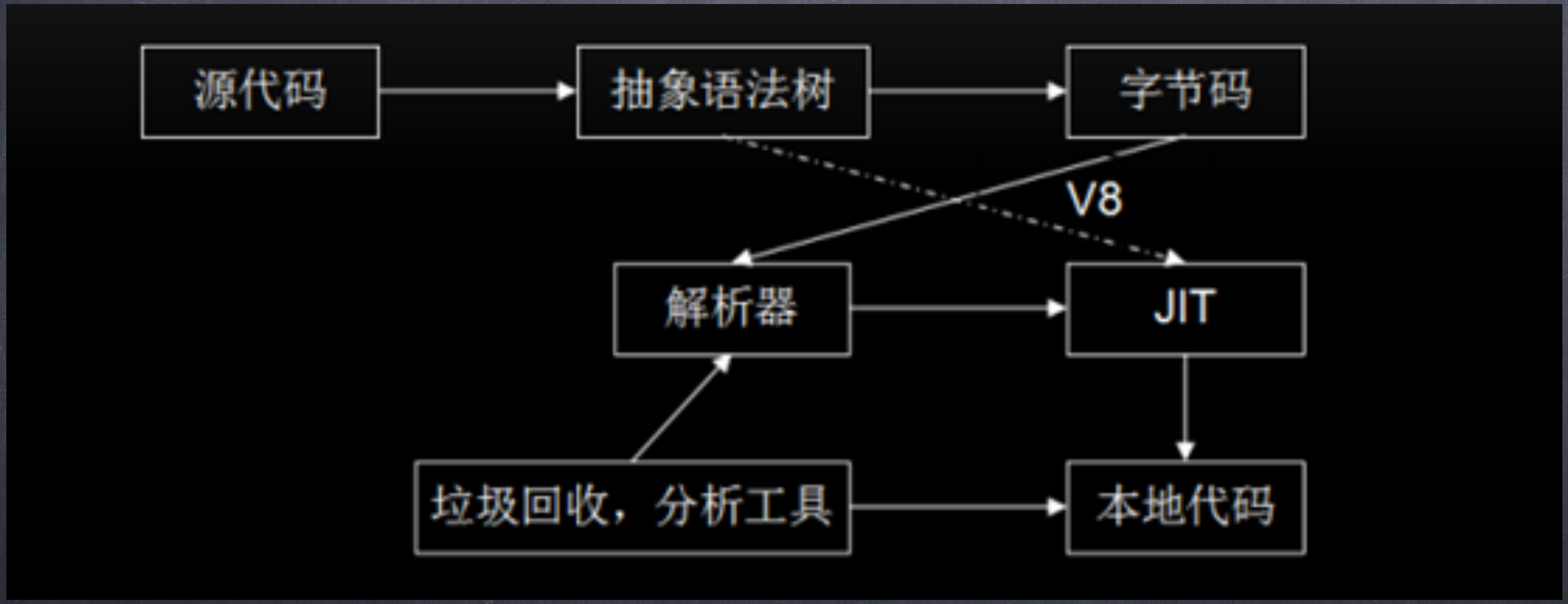
- 渲染引擎 - 能够能够将HTML/CSS/JavaScript文本及相应的资源文件转换成图像结果.
- 渲染引擎的种类
  - Tridend(IE)
  - Gecko(FF)
  - WebKit(Safari,Chrome,Andriod浏览器)等.处于独立的进程中



# JavaScript引擎与渲染引擎

- 渲染引擎使用JS引擎的接口来处理逻辑代码并获取结果。
- JS引擎通过桥接接口访问渲染引擎中的DOM及CSSOM

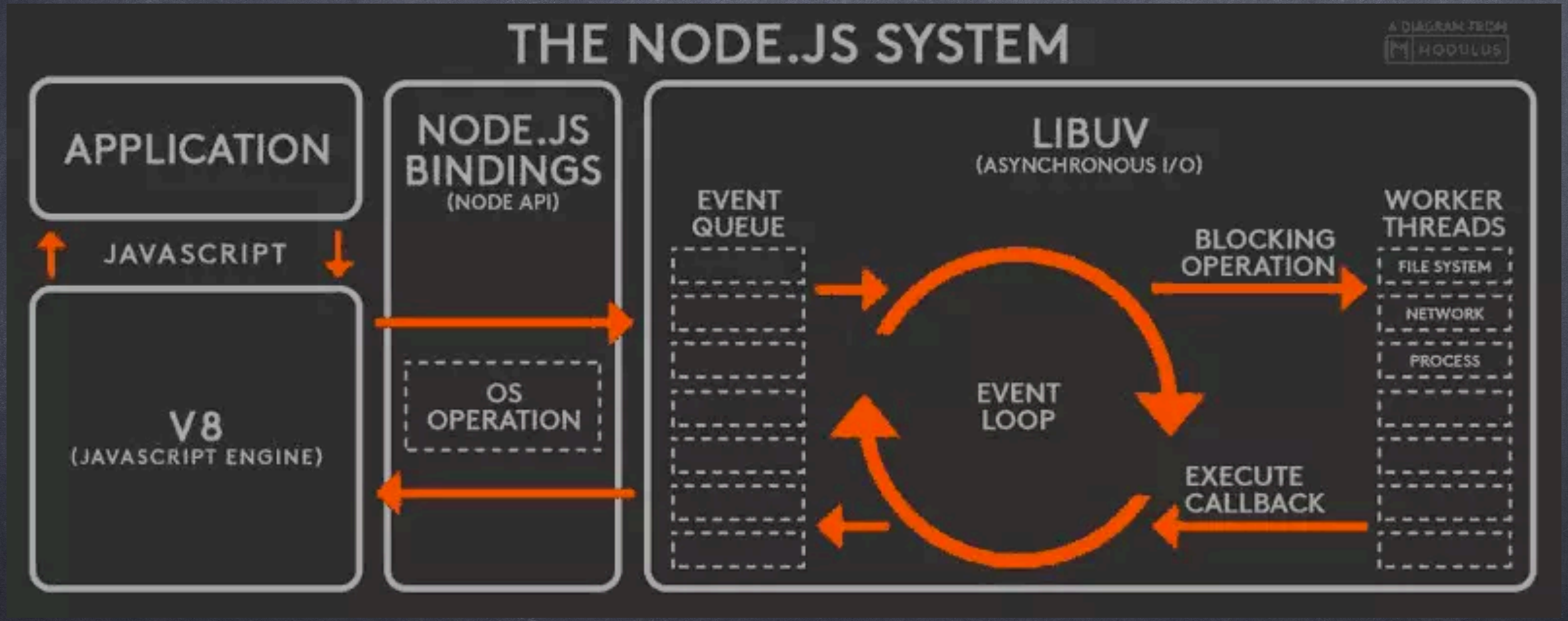
# JavaScript引擎工作流程



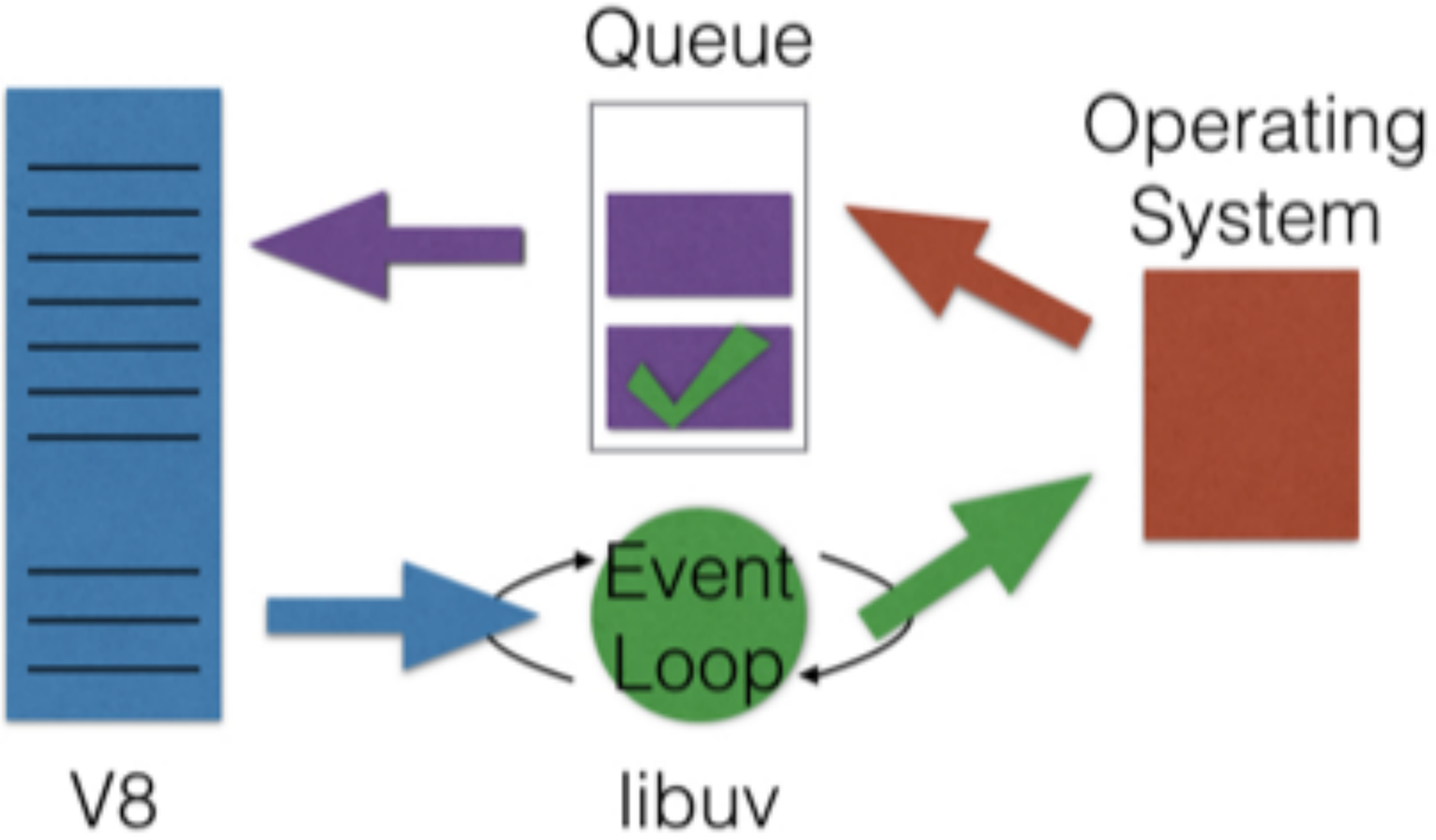
# V8 与 JavaScript Core

- JavaScript Core 引擎是WebKit中默认的JavaScript引擎，也是苹果开源的一个项目，应用较为广泛。最初，性能不是很好，从2008年开始了一系列的优化，重新实现了编译器和字节码解释器，使得引擎的性能有较大的提升。随后内嵌缓存、基于正则表达式的JIT、简单的JIT及字节码解释器等技术引入进来，JavaScriptCore引擎也在不断的迭代和发展。
- JavaScriptCore与V8有一些不同之处，其中最大的不同就是新增了字节码的中间表示，并加入了多层JIT编译器（如：简单JIT编译器、DFG JIT编译器、LLVM等）优化性能，不停的对本地代码进行优化。

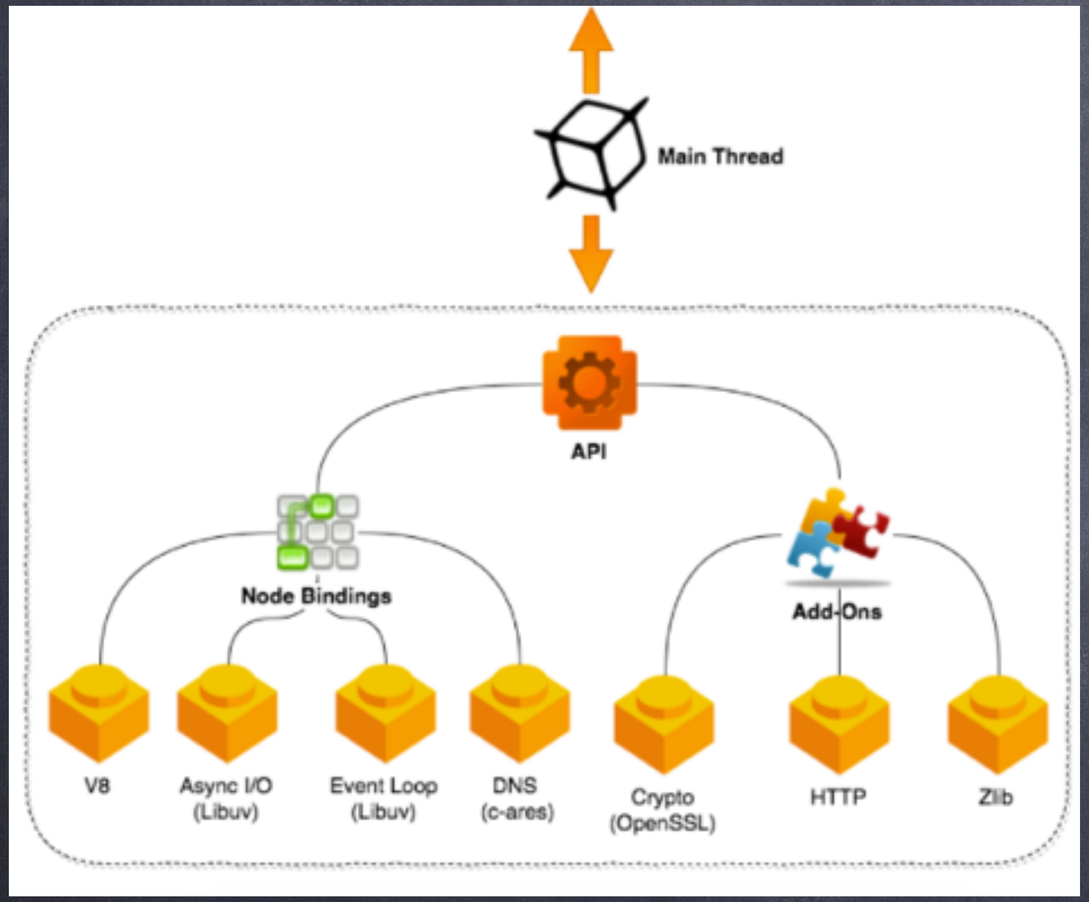
# Node.js 中的 V8


THE NODE.JS SYSTEM
A DIAGRAM FROM
MODULUS
APPLICATION
JAVASCRIPT
V8
(JAVASCRIPT ENGINE)
NODE.JS
BINDINGS
(NODE API)
OS
OPERATION
LIBUV
(ASYNCHRONOUS I/O)
EVENT
QUEUE
EVENT
LOOP
BLOCKING
OPERATION
EXECUTE
CALLBACK
WORKER
THREADS
FILE SYSTEM
NETWORK
PROCESS

# Node.js
# Non-Blocking Event Driven I/O



# Node.js 中的 V8




JavaScript

C/C++

| Node Standard Library | | |
|---|---|---|
| **Node Bindings**<br>(socket, http, file system, etc.) | | |
| **Chrome V8**<br>(JS engine) | **Async I/O**<br>(libuv) | **Event Loop**<br>(libuv) |

# V8 源码一览

- 文档：https://v8.dev/docs
- 源码：https://cs.chromium.org/chromium/src/v8/
- 通过源码可以学到的东西
  - 增强对JavaScript的理解
  - 前端算法
  - 内存管理与CG算法
  - 编译原理、操作系统等知识
  - 面试装逼的高级方式

# V8 引擎源码都看什么

- 工作过程
- 数据表示
- 类型
- 内存管理
- 绑定机制与扩展机制
- 字节码与JIT