INTRO TO NODE.JS & BACKEND DEVELOPMENT

SECTION 1

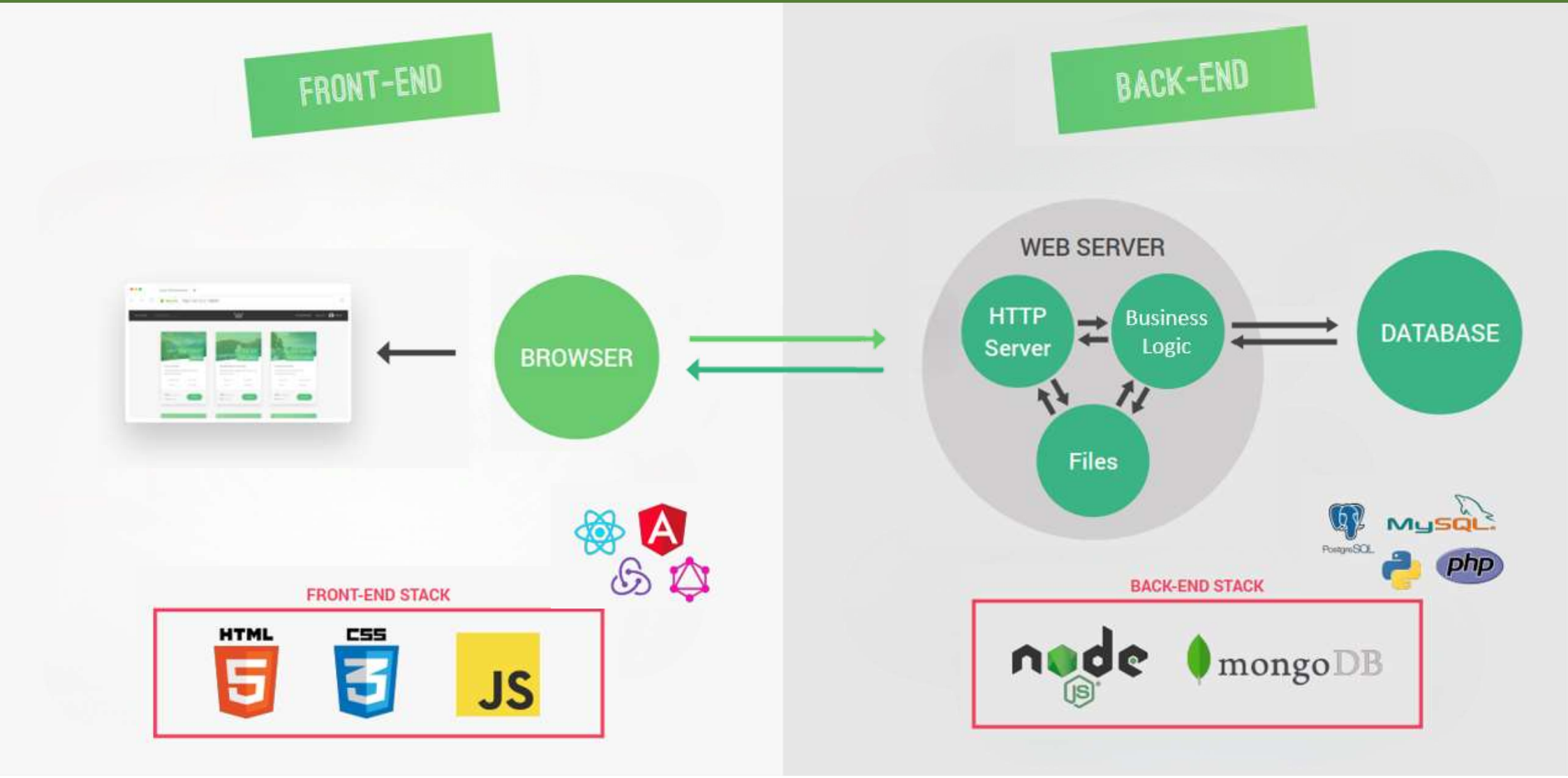
WHAT IS THE BACK-END

WHAT IS THE BACK-END?

- The Back-end is the part of a web application that runs on the server and is responsible for processing data, performing computations, and communicating with other servers.

- It typically consists of a database, a web server, and an application server that interacts with the database and performs the business logic of the application.

WHAT IS BACKEND?



THE IMPORTANCE OF BUSINESS LOGIC IN BACKEND

NOT JUST CRUD

- The main purpose of backend is to implement the business logic of a web application.
- The business logic is what makes an application unique and defines its value proposition.
- It involves complex algorithms, artificial intelligence, machine learning, big data processing, complex architecture, and scaling.

INTRO TO NODE.JS & BACKEND DEVELOPMENT

SECTION 2

HOW THE WEB WORKS

HOW THE WEB WORKS

👉 **Request-response model** or **Client-server architecture**

INTRO TO NODE.JS & BACKEND DEVELOPMENT

SECTION 3

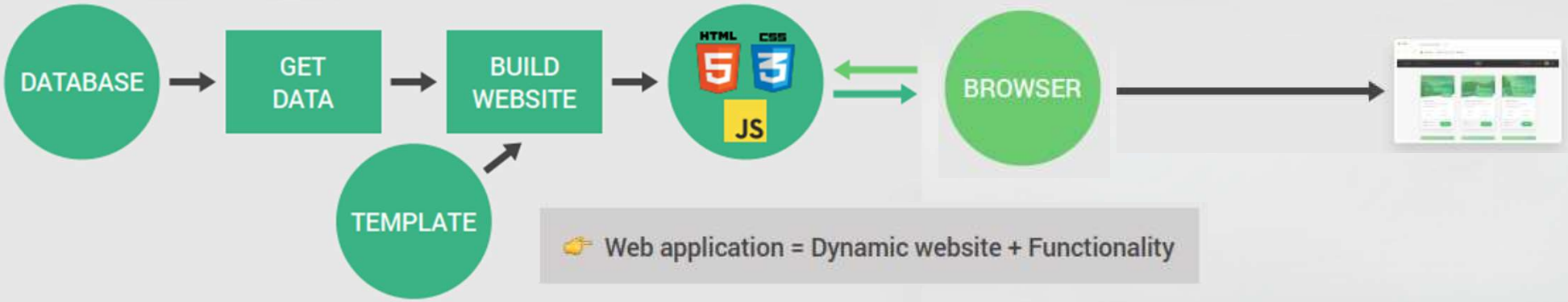
STATIC VS DYNAMIC WEBSITES

STATIC VS DYNAMIC WEBSITES

STATIC

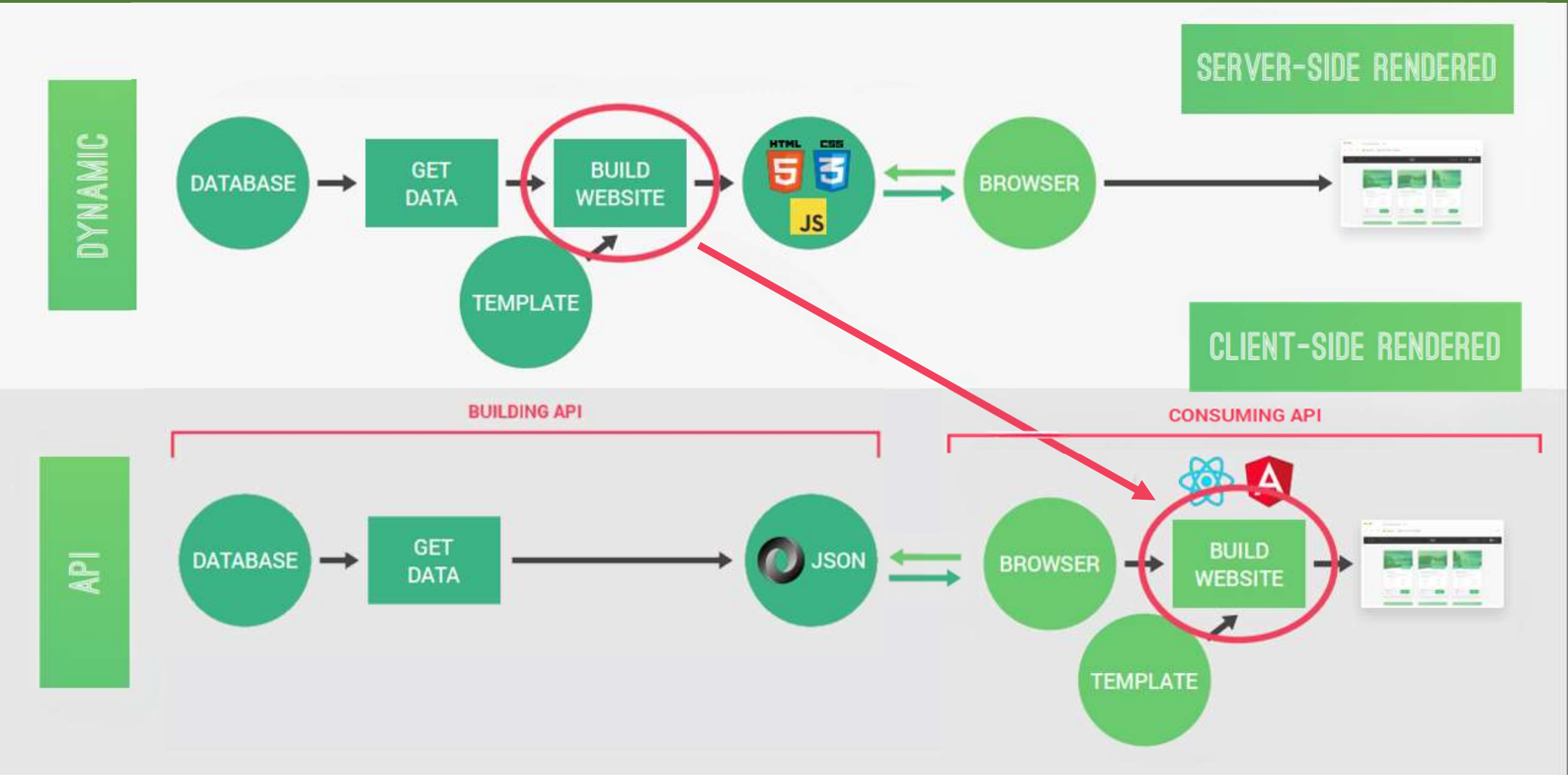
👉 JavaScript ≠ Dynamic

DYNAMIC



👉 Web application = Dynamic website + Functionality

DYNAMIC WEBSITES VS API-POWERED WEBSITES



INTRO TO NODE.JS & BACKEND DEVELOPMENT

SECTION 4

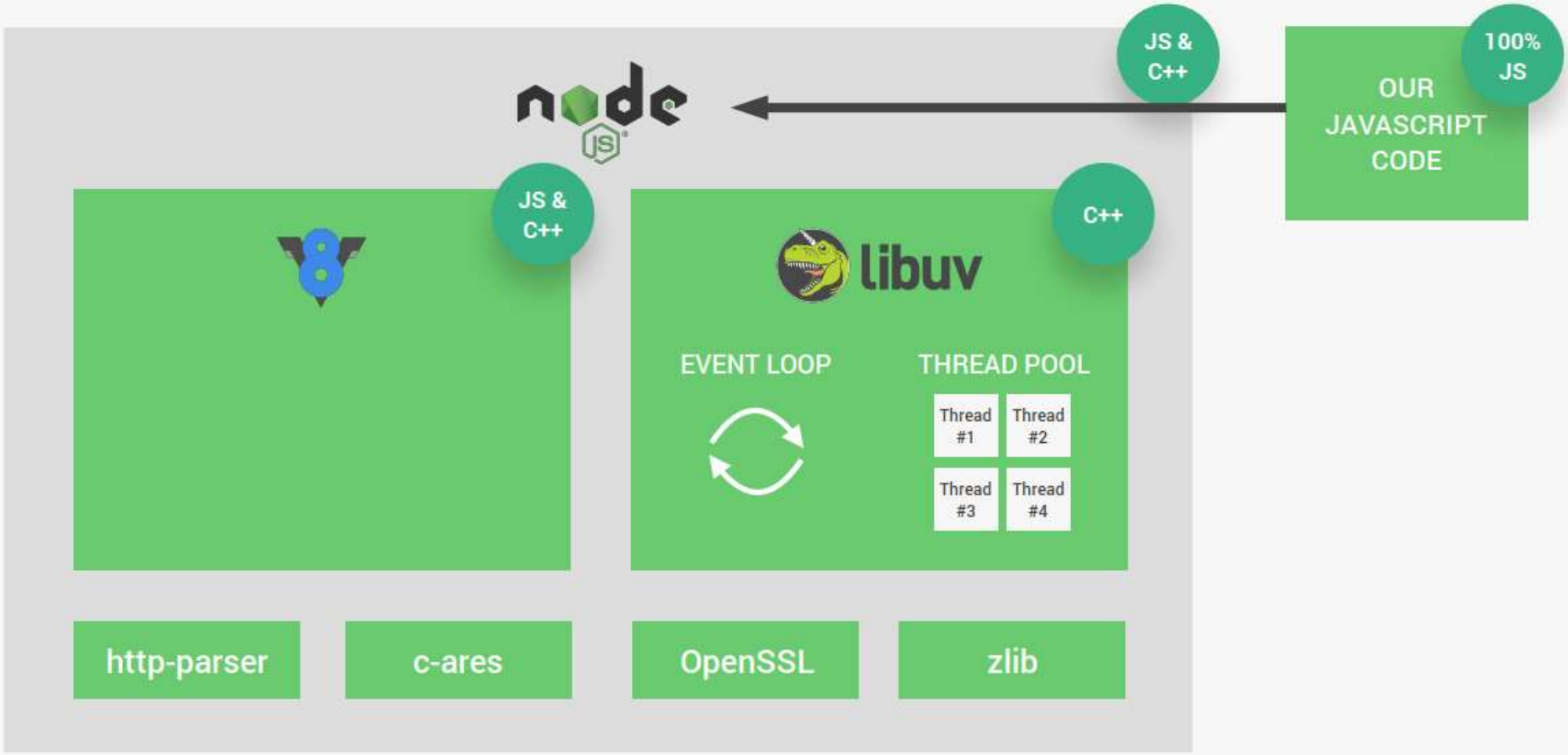
INTRODUCTION TO NODE.JS

WHAT IS NODE.JS

NODE.JS

NODE.JS IS A JAVASCRIPT RUNTIME BUILT ON GOOGLE'S OPEN-SOURCE V8 JAVASCRIPT ENGINE. 🤔

NODE.JS ARCHITECTURE



INTRO TO NODE.JS & BACKEND DEVELOPMENT

SECTION 4

SYNCHRONOUS VS. ASYNCHRONOUS COD

(BLOCKING VS. NON-BLOCKING)

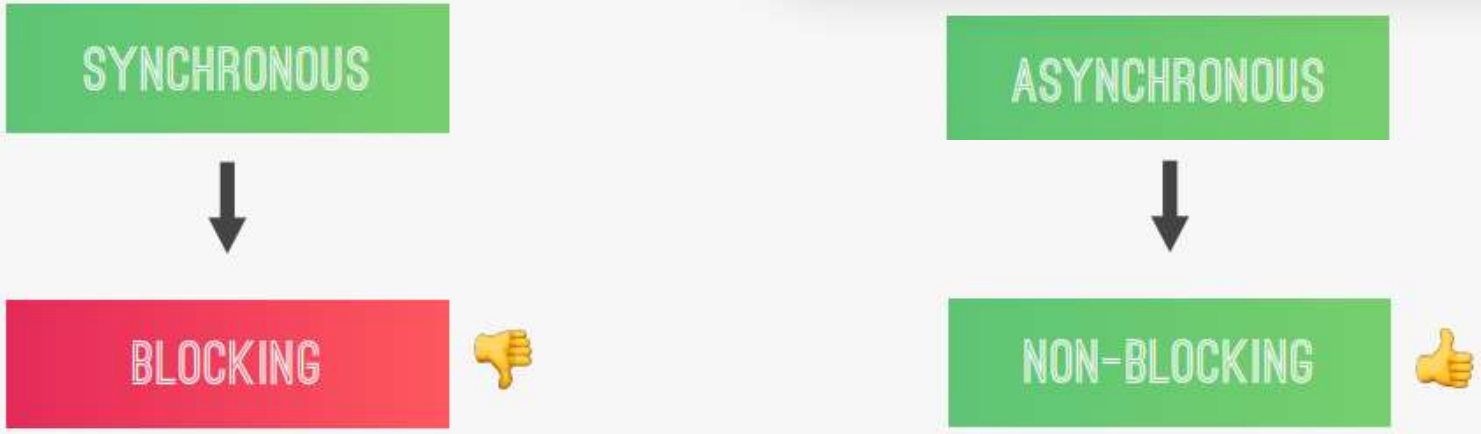
SYNCHRONOUS VS. ASYNCHRONOUS CODE (BLOCKING VS. NON-BLOCKING)

```
const fs = require('fs');

// Blocking code execution
const input = fs.readFileSync('input.txt', 'utf-8');
console.log(input);
```

```
const fs = require('fs');

// Non-blocking code execution
fs.readFile('input.txt', 'utf-8', (err, data) => {
    console.log(data);
});
console.log('Reading file...');
```



INTRO TO NODE.JS & BACKEND DEVELOPMENT

SECTION 5

WHY AND WHEN TO USE NODE..

WHY AND WHEN TO USE NODE.JS?

NODE.JS PROS

- Single-threaded, based on event driven, non-blocking I/O model 🤯😅
- Perfect for building **fast** and **scalable** data-intensive apps;
- Companies like NETFLIX UBER PayPal ebay have started using node in production;
- **JavaScript across the entire stack:** faster and more efficient development;
- **NPM:** huge library of open-source packages available for everyone for free;
- **Very active** developer community.

USE NODE.JS

- API with database behind it (preferably NoSQL);
- Data streaming (think YouTube);
- Real-time chat application;
- Server-side web application.

DON'T USE

- Applications with heavy server-side processing (CPU-intensive).

RAILS php

WHY AND WHEN TO USE NODE.JS?