



# NodeJS

# MERN Stack

---



MySQL



Express



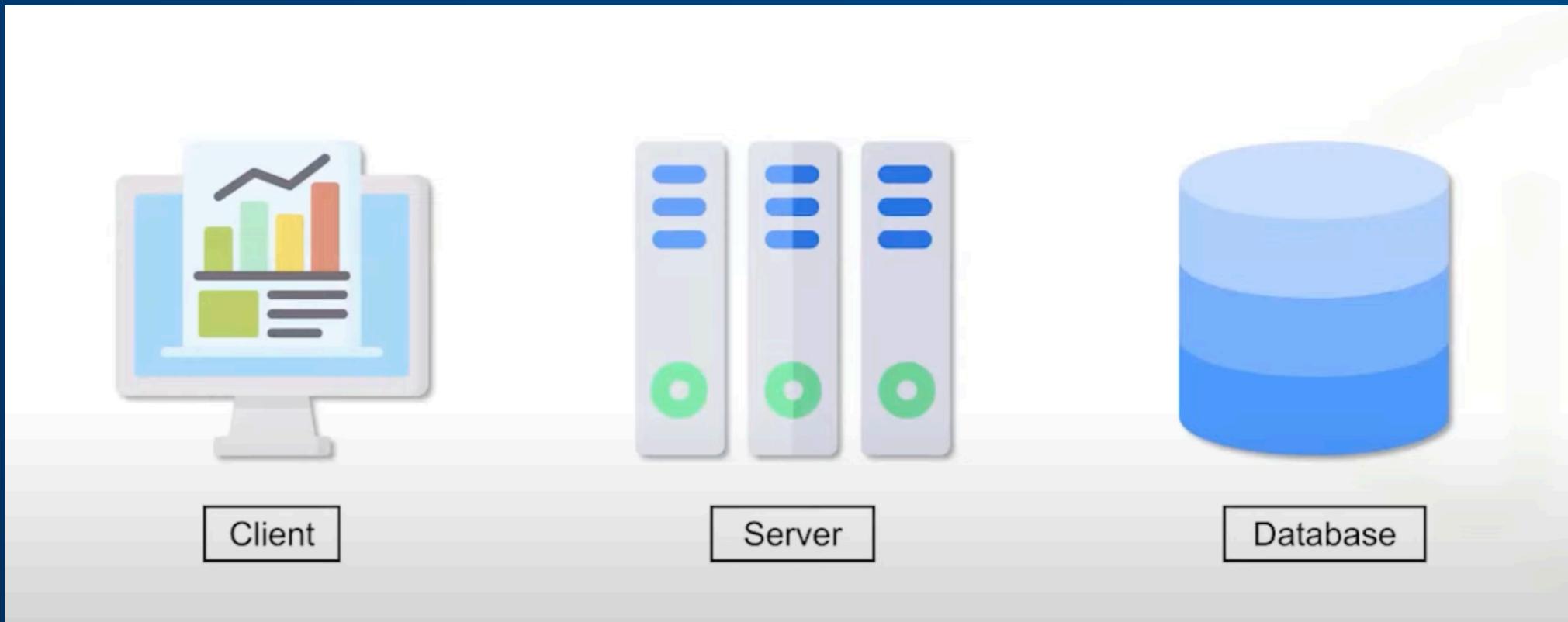
React



NodeJS

# Client Server Architecture

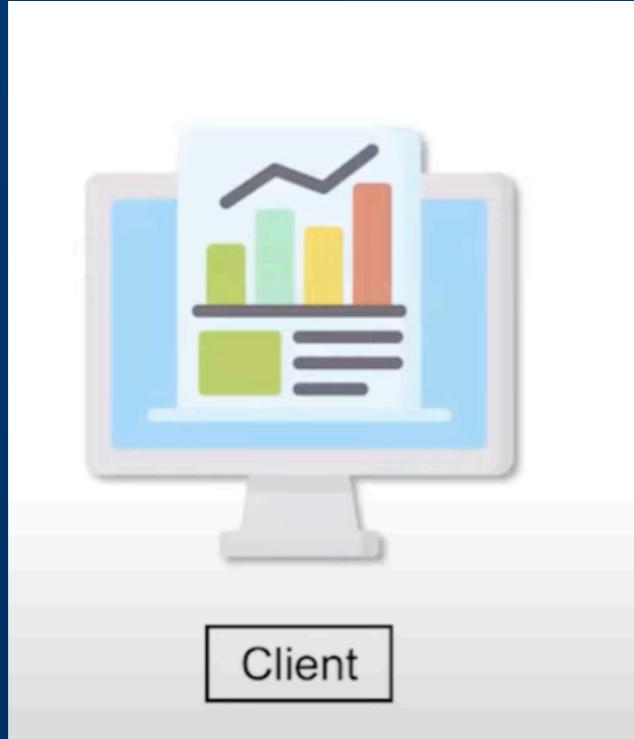
---



# Client

---

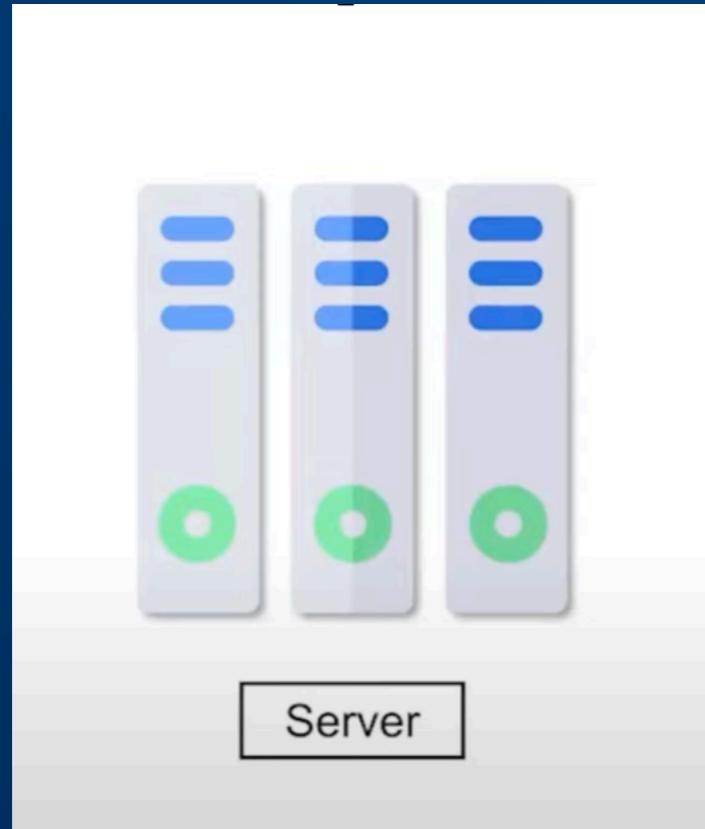
## Front end Frameworks



- ReactJS
- AngularJS
- Next.js
- Vue.js
- Bootstrap
- Jquery and more

# Server

---



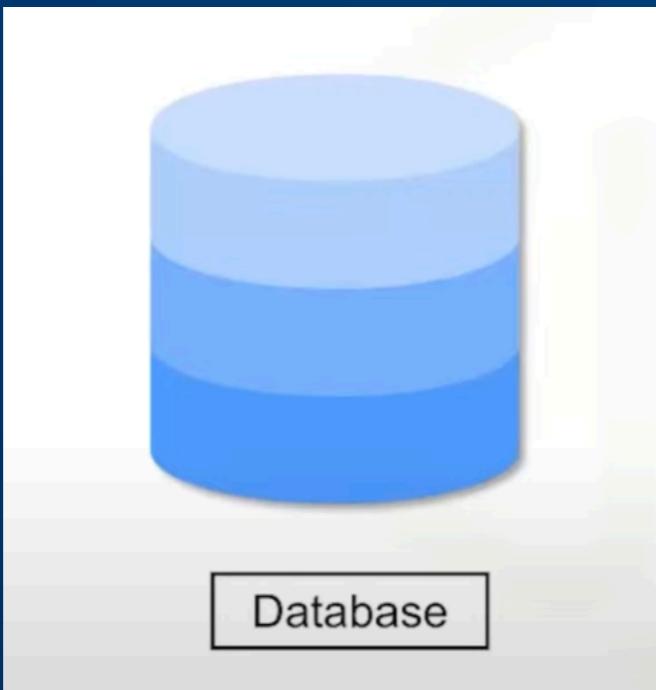
## Backend Servers

- NodeJS
- Python
- Java
- Ruby
- PHP and more

# Database

---

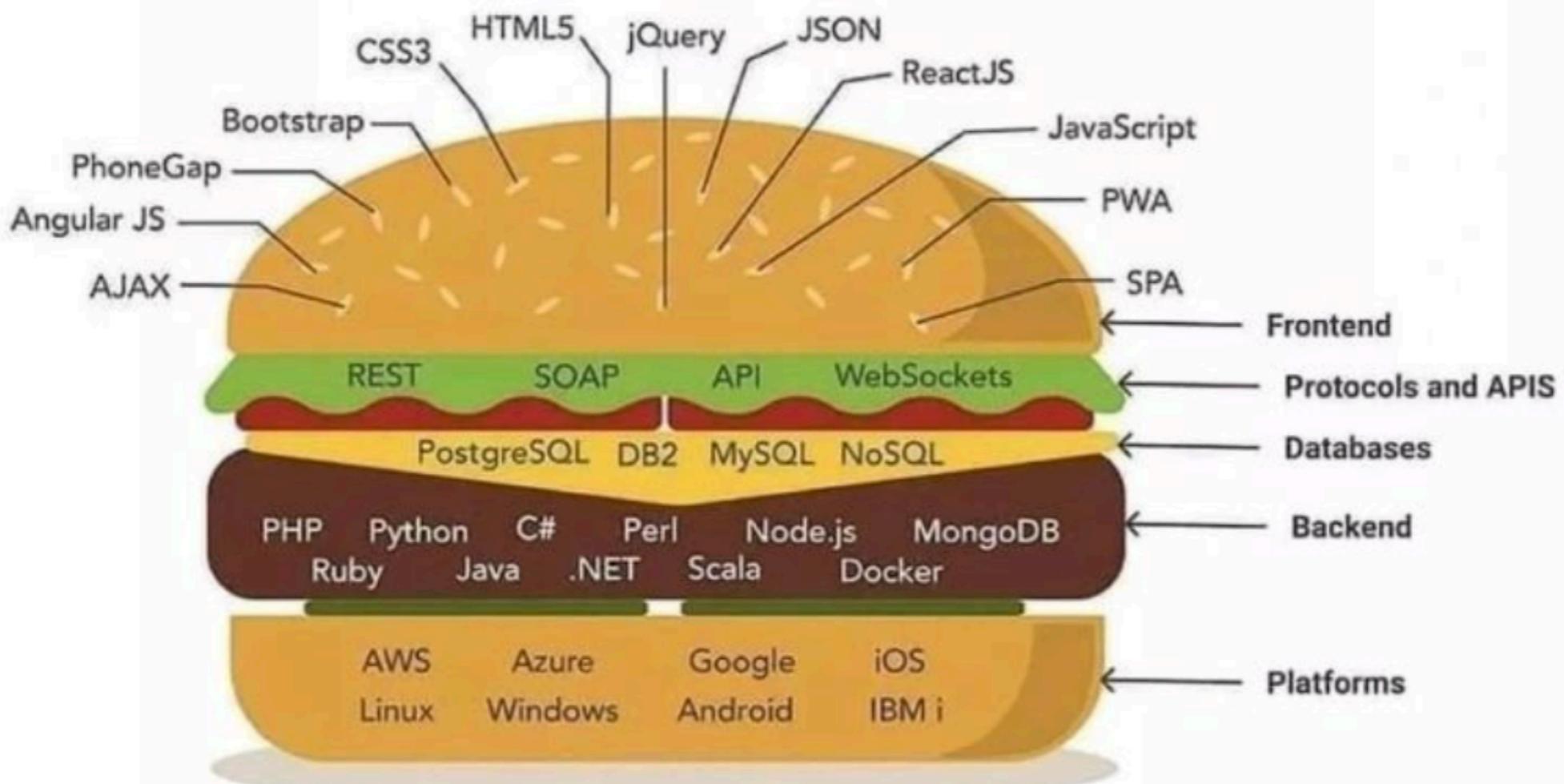
## Backend Databases



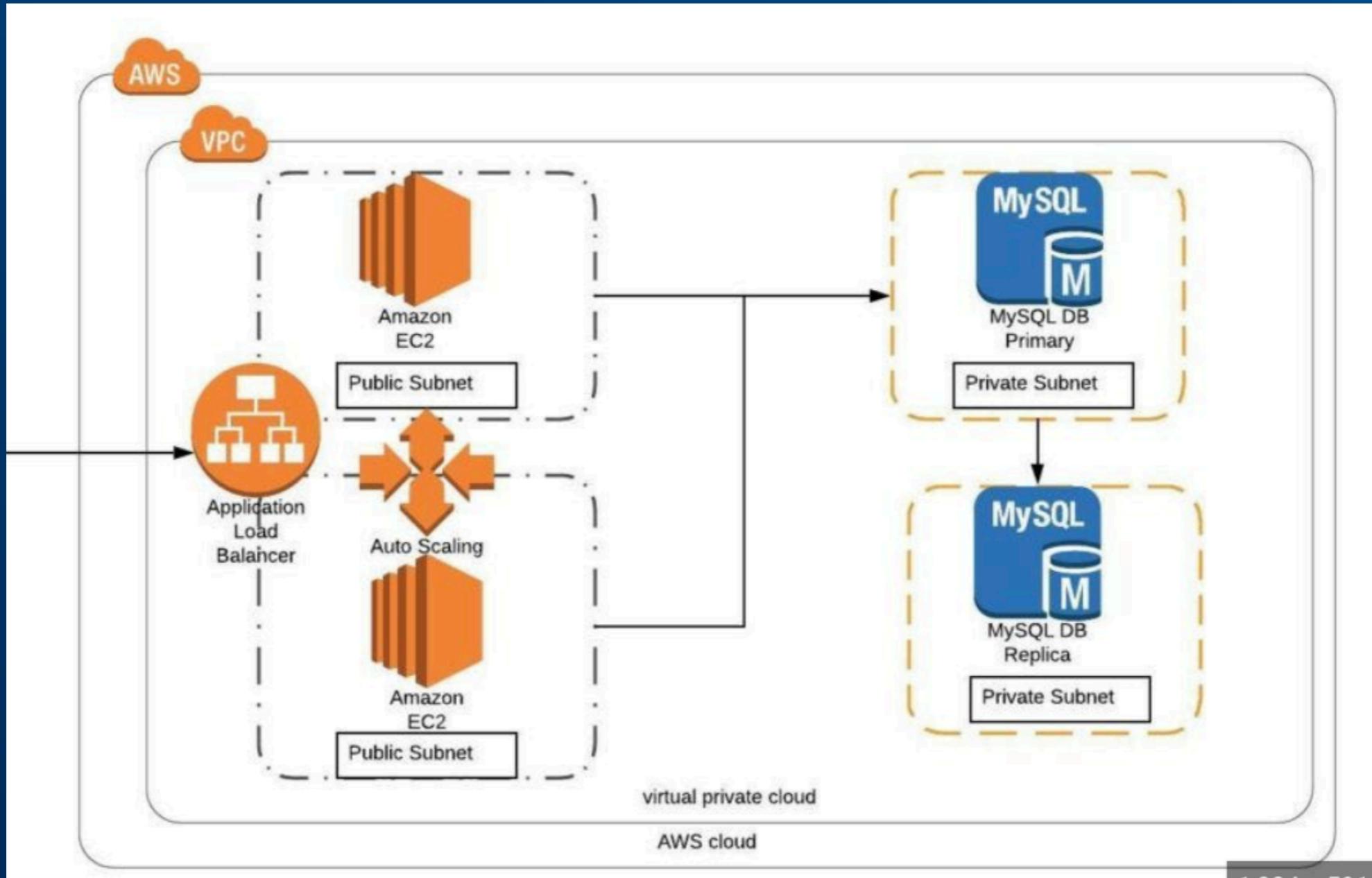
- MySQL
- MongoDB
- PostgreSQL
- DynamoDB
- more

# Client Server Architecture

## “Full Stack Developer”

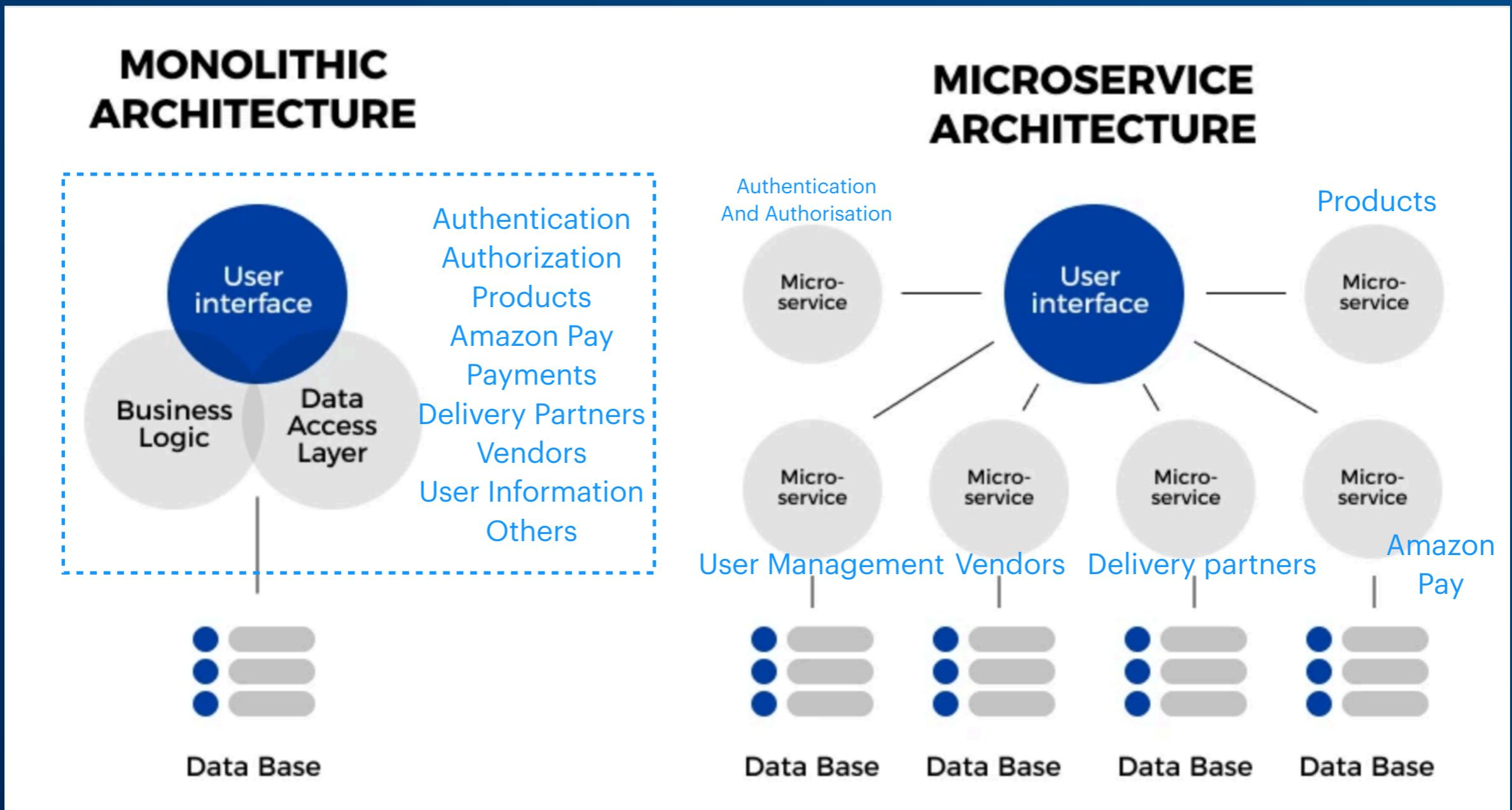


# AWS Hosting

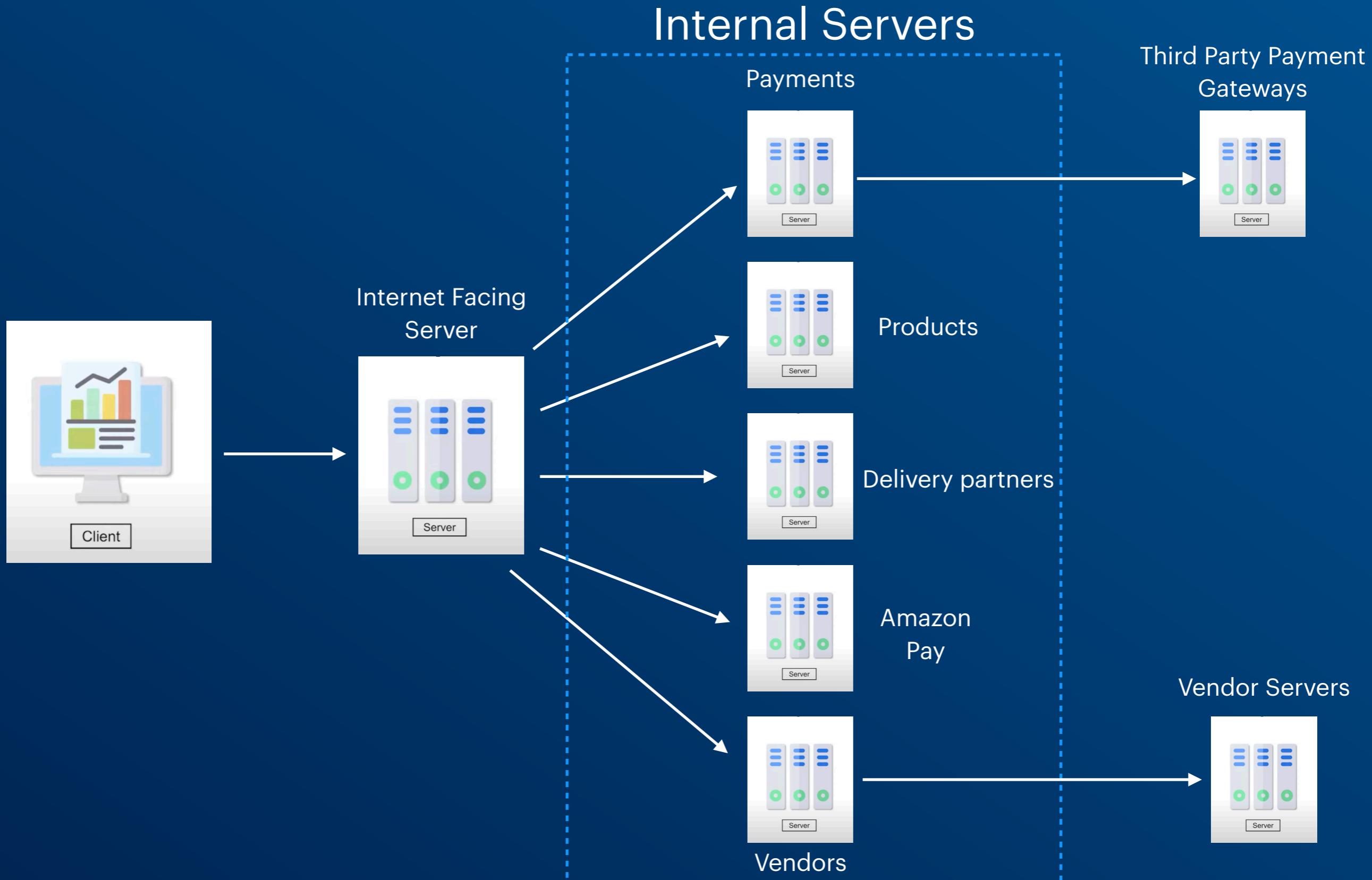


# Client Server Architecture

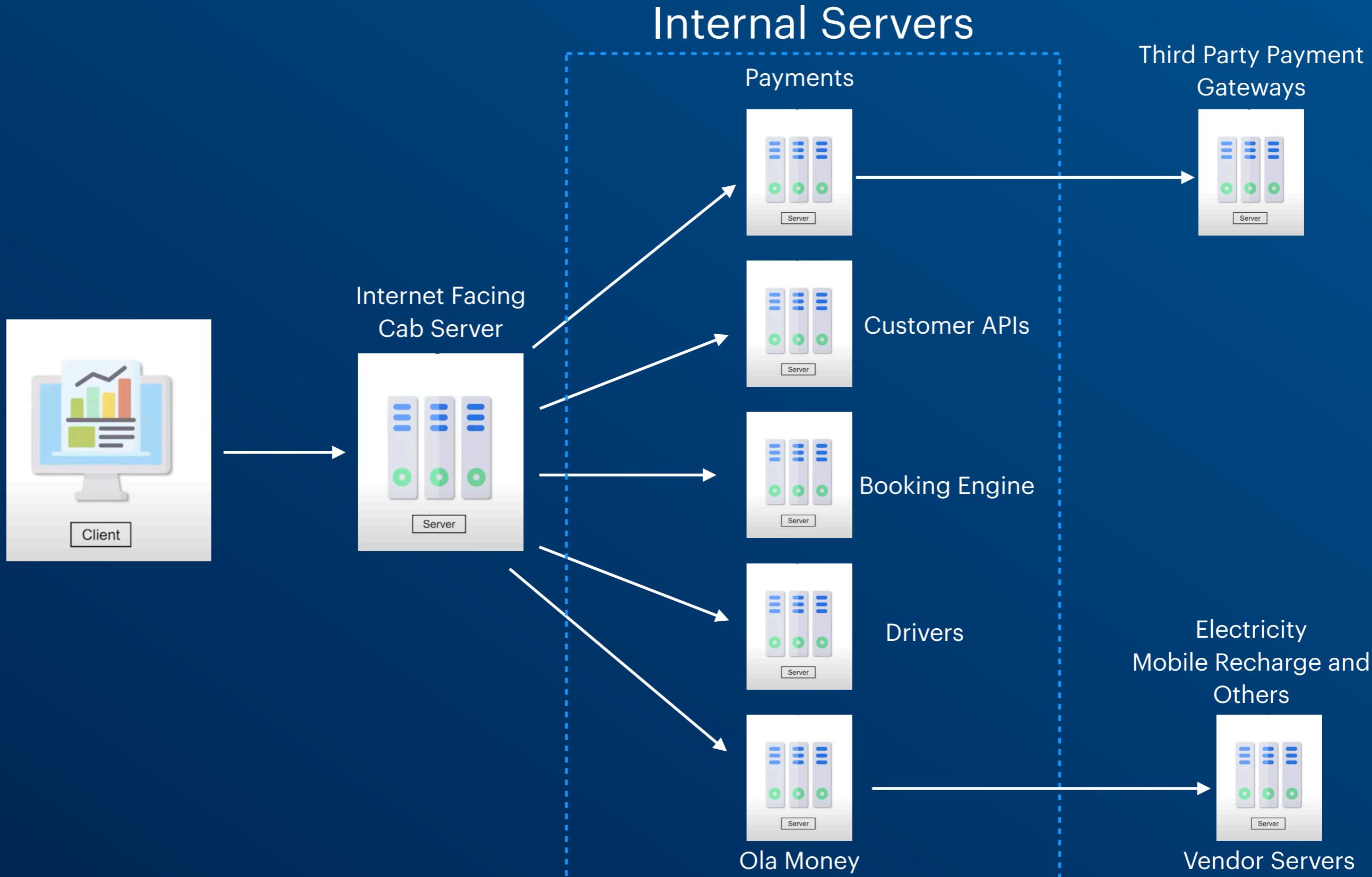
Ex: Amazon



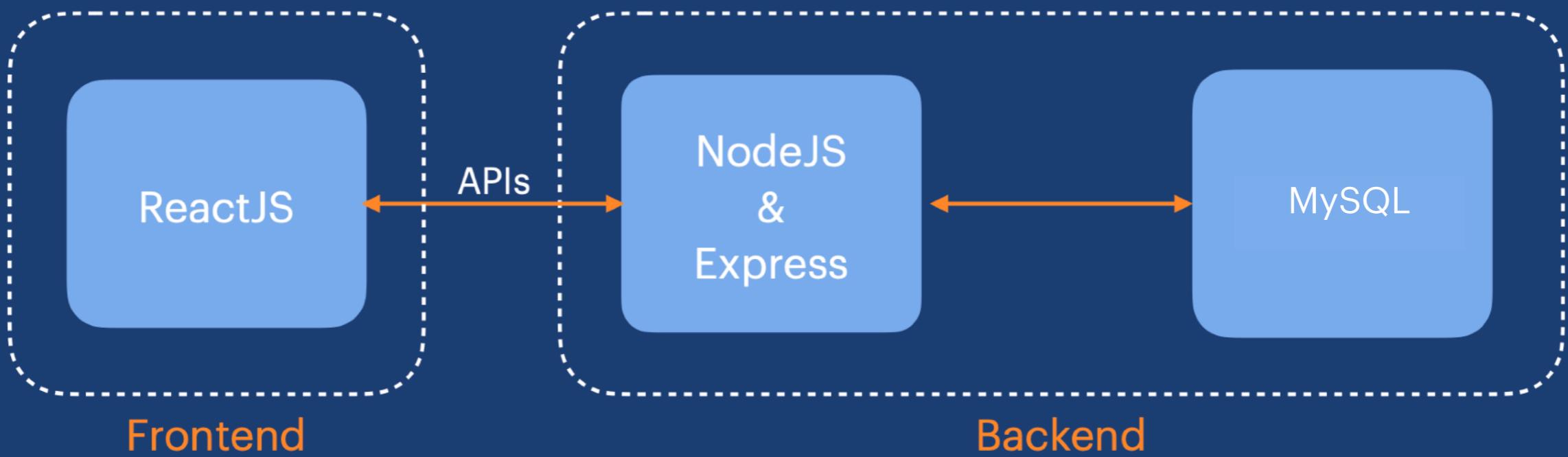
# Amazon - Client Server Architecture



# OLA - Client Server Architecture



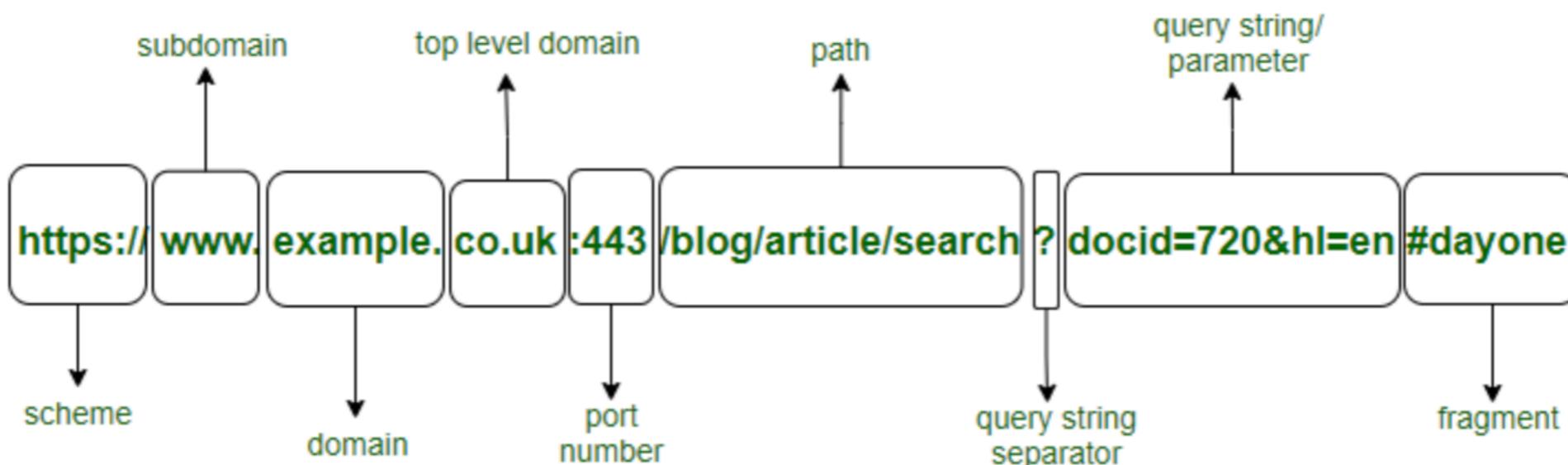
# Client Server Architecture



# URL

## Parts of a URL

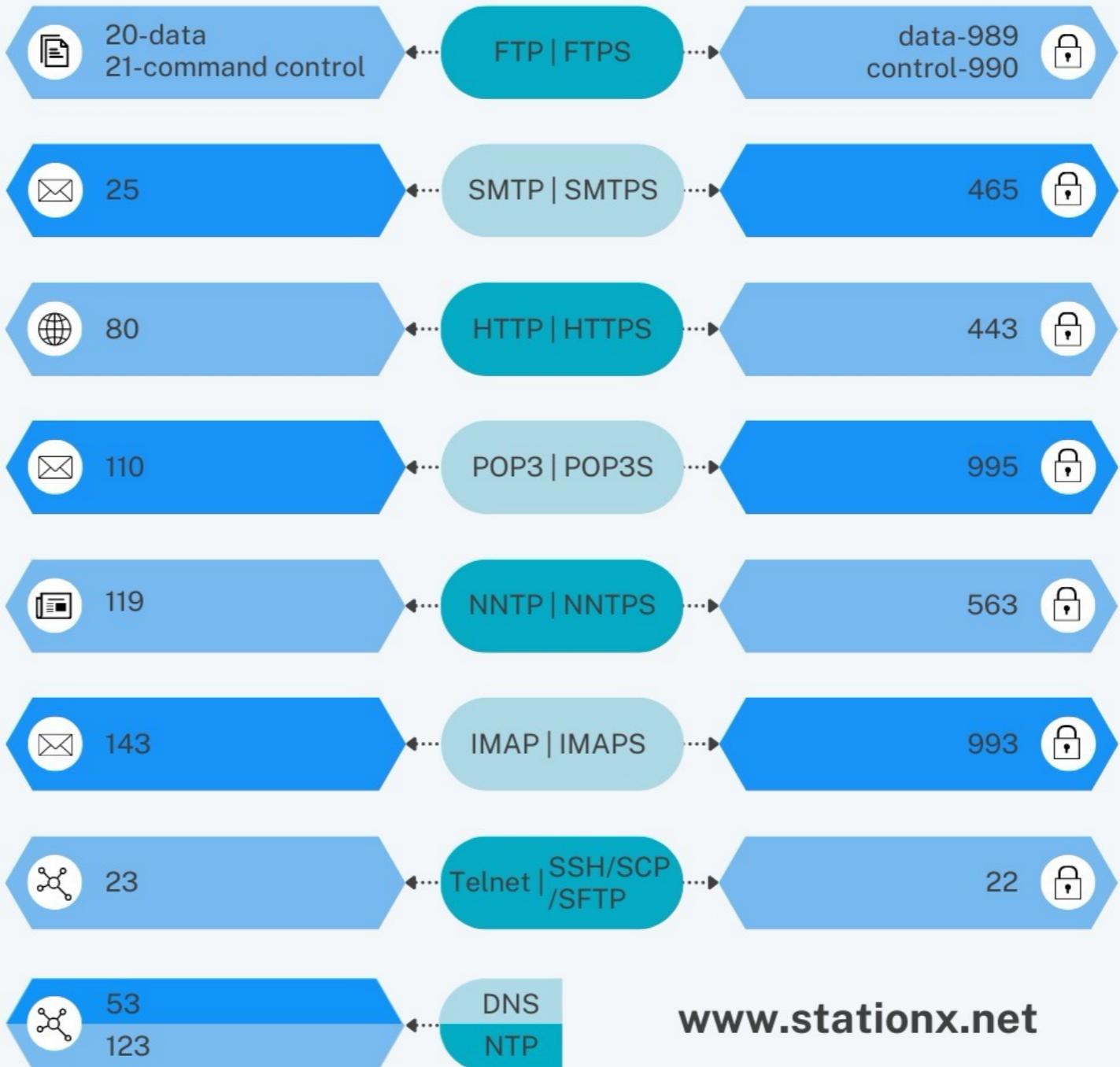
URL : `https://www.example.co.uk:443/blog/article/search?docid=720&hl=en#dayone`



# Ports

## Well-Known Ports: Unencrypted vs Encrypted

Must-know commonly used ports to memorize



# API

---

Two software components to communicate with each other using a set of definitions and protocols.



# API

---

- REST APIs
- WebSocket APIs
- ~~SOAP APIs~~

# API

---

- Private APIs
- Public APIs
- Partner APIs

# API Format

**Request**

Pretty Raw Hex ⌂ ⌂ ⌂ ⌂

```
1 POST /doLogin HTTP/1.1
2 Host: demo.testfire.net
3 Cookie: JSESSIONID=97D77081A17B397D489D5B7BE99D46CE
4 Content-Length: 37
5 Cache-Control: max-age=0
10 Origin: https://demo.testfire.net
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/124.0.6367.118 Safari/537.36
18 Referer: https://demo.testfire.net/login.jsp
22 Connection: close
23
24 uid=admin&passw=admin&btnSubmit=Login
```

# API/HTTP Methods

---

Get

To **read** the data from the server

POST

To **create** the data in the server

Put

To **update** the data in the server

Delete

To **Delete** the data in the server

# CRUD Operations

---

POST

To CREATE the data in the server

Get

To READ the data from the server

Put

To UPDATE the data in the server

Delete

To Delete the data in the server

# Example for API methods

---

Method	Sample
GET	GET /api/address/1
POST	POST /api/address/
PUT	PUT /api/address/1
DELETE	DELETE /api/address/1

# Example for API methods

---

Method	Sample
GET	GET /api/address?address_id=1
POST	POST /api/address
PUT	PUT /api/address
DELETE	DELETE /api/address

# HTTP Status Codes

<b>1xx: Information</b>	<b>3xx: Redirection</b>	<b>4xx: Client Errors</b>
<b>100 - Continue</b> Server accepts initial request; client should continue.	<b>301 - Moved Permanently</b> Resource has been moved permanently.	<b>400 - Bad Request</b> Request cannot be processed.
<b>101 - Switching protocols</b> Server acknowledges a switch in protocol.	<b>302 - Found</b> Resource has been moved temporarily.	<b>401 - Unauthorized</b> Authentication needed.
<b>102 - Processing</b> Request is processing (WebDAV).	<b>303 - See Other</b> Redirect to a different resource, typically after a POST operation.	<b>403 - Forbidden</b> Client is authenticated but lacks permissions.
<b>103 - Early Hints</b> Hints for preloading resources.	<b>304 - Not Modified</b> Resource hasn't changed since the last request used for caching purposes.	<b>404 - Not Found</b> Resource cannot be found.
<b>2xx: Success</b>	<b>5xx: Server Errors</b>	<b>405 - Method Not Allowed</b> The HTTP method used isn't allowed for the resource.
<b>200 - OK</b> Request succeeded.	<b>500 - Internal Server Error</b> Generic error message indicating a server-side issue.	<b>406 - Not Acceptable</b> Server can't produce a response that matches the client's Accept headers.
<b>201 - Created</b> Request fulfilled new resource created.	<b>502 - Bad Gateway</b> Upstream server sent an invalid response.	<b>409 - Conflict</b> Request conflicts with the current server state.
<b>202 - Accepted</b> Request accepted, but not yet completed.	<b>503 - Service Unavailable</b> Server temporarily unable to handle the request.	<b>413 - Payload Too Large</b> The request body is too large for the server to process.
<b>204 - No Content</b> Response is successful, but there's no content to send in the body.	<b>504 - Gateway Timeout</b> Upstream server did not respond in time.	<b>429 - Too Many Requests</b> Client has made too many requests in a given amount of time.
<b>206 - Partial Content</b> Partial resource sent (for byte-serving).		

# Request Headers

**Request**

Pretty   Raw   Hex

1 POST /api/login HTTP/1.1  
2 Host: 127.0.0.1:8000  
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36  
12 Content-Type: application/json  
16 Connection: close  
17 Content-Length: 60  
18  
19 {  
20 "email": "sunil@sunil.com",  
21 "password": "sunil"  
22 }

# Response Headers

```
Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Access-Control-Allow-Origin: *
4 Access-Control-Allow-Credentials: true
5 Set-Cookie: user_token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InN1bmLsQHN1bmLsLmNvbSIiZpcnN0X25hbWUi0iJzdW5pbCIsImxhc3RfbmFtZSI6Imt1bWFyIiwicGhvbmVfbnVtYmVyIjo5MDM2MDc2MTYzLCJ1c2VyX2lkIjoxMCviaWF0IjoxNzE2MDMwNzEzLCJleHAiOjE3MTYwMzQzMjN9.yP-6pC88iNWuxF6H0ZwE4-bk4UnuMpxM8rt5AXITQSg; Max-Age=3000; Domain=localhost; Path=/; Expires=Sat, 18 May 2024 12:01:53 GMT; HttpOnly; SameSite=Lax
6 Content-Type: application/json
7 Date: Sat, 18 May 2024 11:11:53 GMT
8 Connection: close
9 Content-Length: 316
10
11 {
    "message": "Login successful!",
    "user_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InN1bmLsQHN1bmLsLmNvbSIiZpcnN0X25hbWUi0iJzdW5pbCIsImxhc3RfbmFtZSI6Imt1bWFyIiwicGhvbmVfbnVtYmVyIjo5MDM2MDc2MTYzLCJ1c2VyX2lkIjoxMCviaWF0IjoxNzE2MDMwNzEzLCJleHAiOjE3MTYwMzQzMjN9.yP-6pC88iNWuxF6H0ZwE4-bk4UnuMpxM8rt5AXITQSg"
}
```

# JSON Data

---

```
{  
  "email": "sunil@sunil.com",  
  "password": "sunil"  
}
```

```
{  
  "address": "JP nagar",  
  "pincode": "2122112",  
  "city": "Bangalore",  
  "state": "Karnataka",  
  "country": "India"  
}
```

```
{  
  "address": {  
    "line1": "JP nagar",  
    "line2": "1st cross"  
  },  
  "pincode": "2122112",  
  "city": "Bangalore",  
  "state": "Karnataka",  
  "country": ["India", "Australia"]  
}
```

# application/www-x-form-urlencoded

**Request**

Pretty Raw Hex

1 POST /doLogin HTTP/1.1  
2 Host: demo.testfire.net  
3 Cookie: JSESSIONID=F4A367FBFEA427CAF30DEC217BD98506; AltoroAccounts=  
"ODAwMDAwfkNvcnBvcmF0ZX4xLjBFMjZ80DAwMDAxfkNoZWNrav5nf105Ljk50Tk40DE1NTMyNTU5NEUyNXw=";  
4 Content-Length: 37  
5 Cache-Control: max-age=0  
6 Sec-Ch-Ua: "Not-A.Brand";v="99", "Chromium";v="124"  
7 Sec-Ch-Ua-Mobile: ?0  
8 Sec-Ch-Ua-Platform: "macOS"  
9 Upgrade-Insecure-Requests: 1  
0 Origin: https://demo.testfire.net  
1 Content-Type: application/x-www-form-urlencoded  
2 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like  
Gecko) Chrome/124.0.6367.118 Safari/537.36  
3 Accept:  
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng  
,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7  
4 Sec-Fetch-Site: same-origin  
5 Sec-Fetch-Mode: navigate  
6 Sec-Fetch-User: ?1  
7 Sec-Fetch-Dest: document  
8 Referer: https://demo.testfire.net/login.jsp  
9 Accept-Encoding: gzip, deflate, br  
0 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8  
1 Priority: u=0, i  
2 Connection: close  
3  
4 uid=admin&passw=admin&btnSubmit=Login

---

Day 1

# NodeJs

---

## Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

### Open Source

Source code is publicly available for accessing and modifying

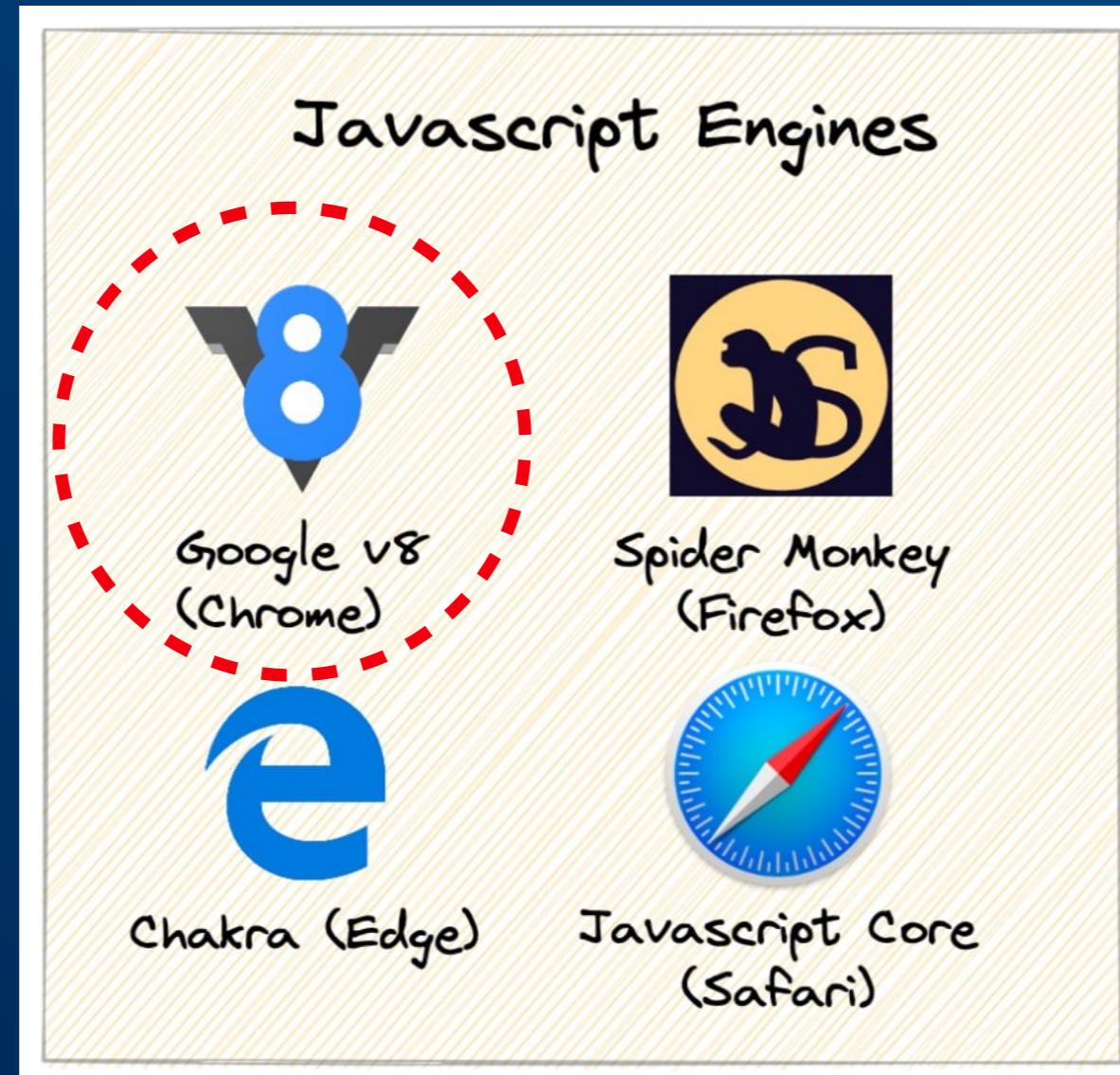
### Cross Platform

Supported for Mac, Linux and Windows

### JavaScript runtime environment

# JavaScript Engine

- JavaScript can't be understood by Computer
- JavaScript Engine - Conversion of JS code to machine readable code



# V8 JavaScript Engine

---

## V8 JavaScript Engine

V8 is Google's open source JavaScript engine.

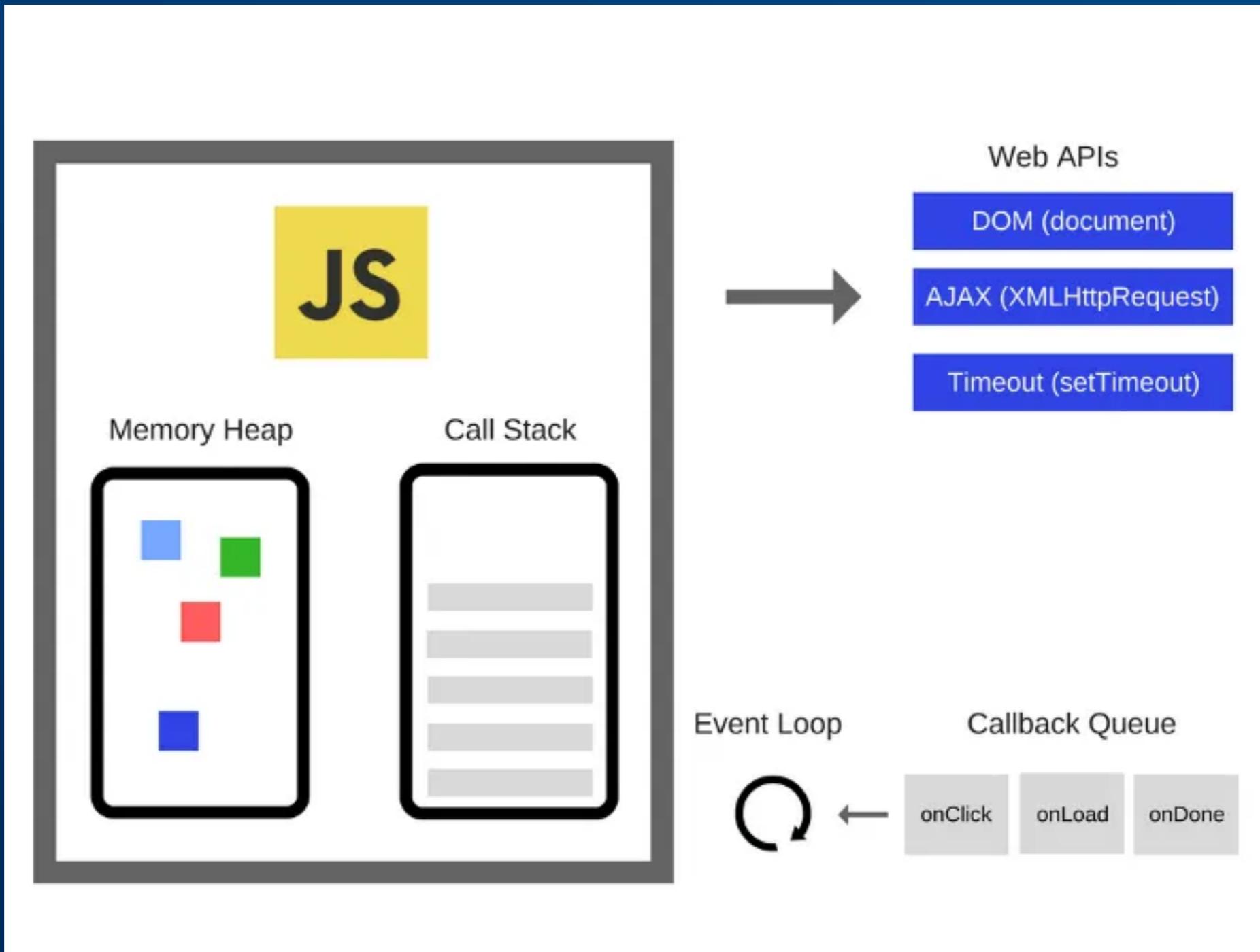
V8 implements ECMAScript as specified in ECMA-262.

V8 is written in C++ and is used in Google Chrome, the open source browser from Google.

V8 can run standalone, or can be embedded into any C++ application.

- C++ program can include V8 engine to execute JS code
- C++ is efficient in performing file operations/network calls/DB actions
- NodeJs integrates V8 engine to execute JavaScript code
- NodeJS is written in C++

# JavaScript RunTime



# NodeJs

---

## Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

### Open Source

Source code is publicly available for accessing and modifying

### Cross Platform

Supported for Mac, Linux and Windows

### JS runtime environment

# What can be done with NodeJs

---

- Web Application
- Backend Servers
- Implement API calls
- Real Time applications
- Streaming applications
- And more

<https://github.com/nodejs/node>

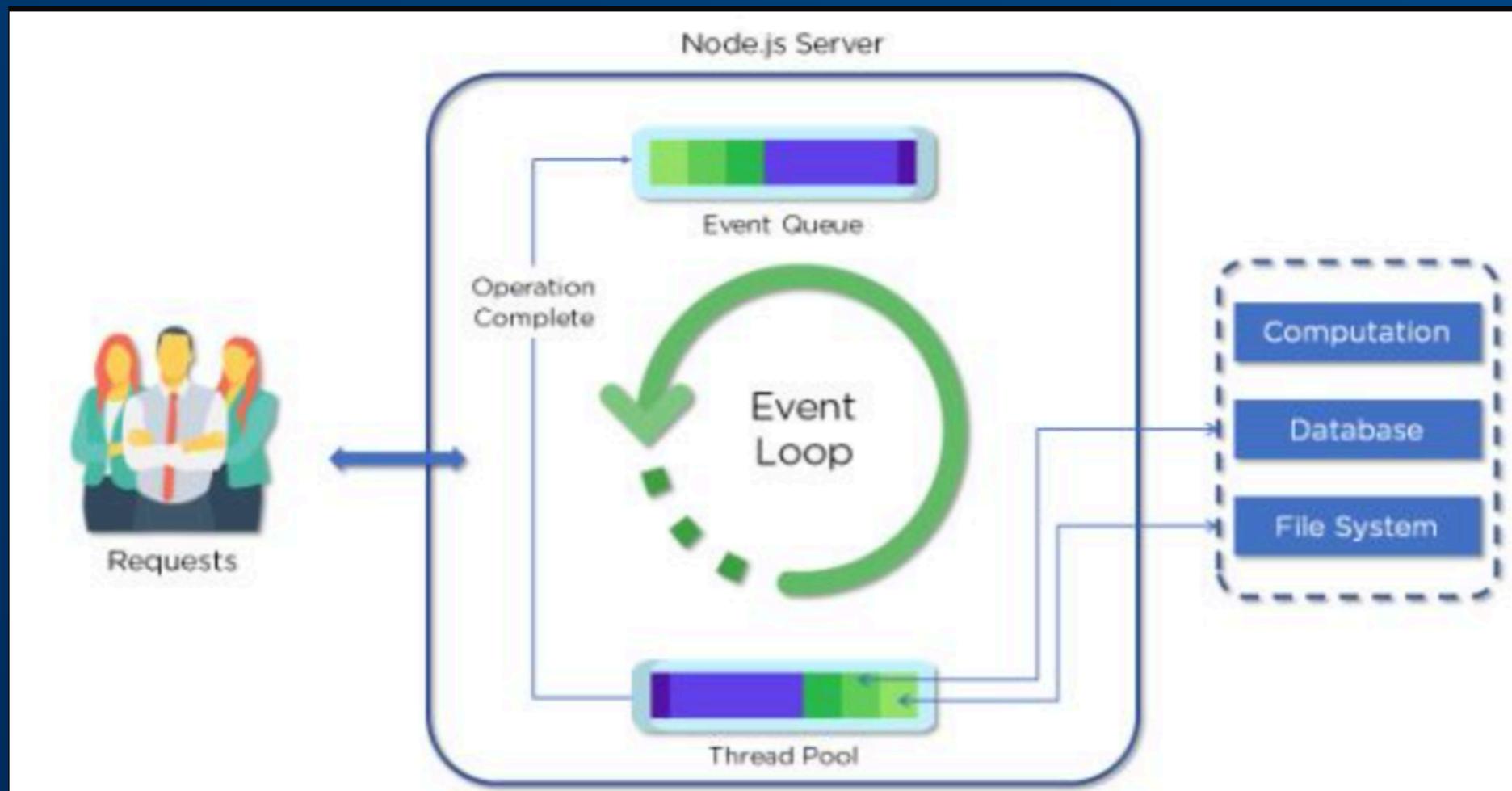
# NodeJS

---

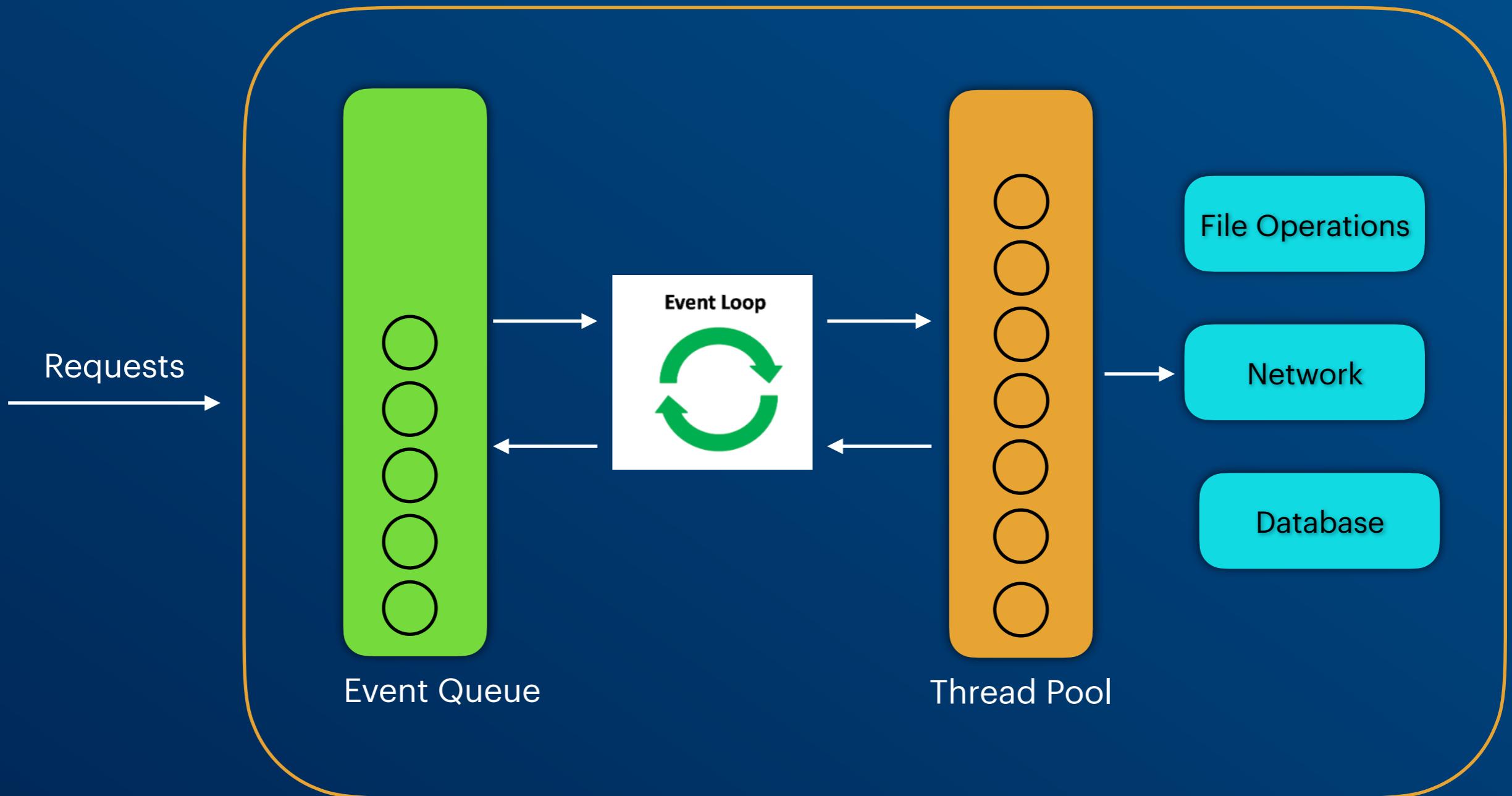


# NodeJS Architecture

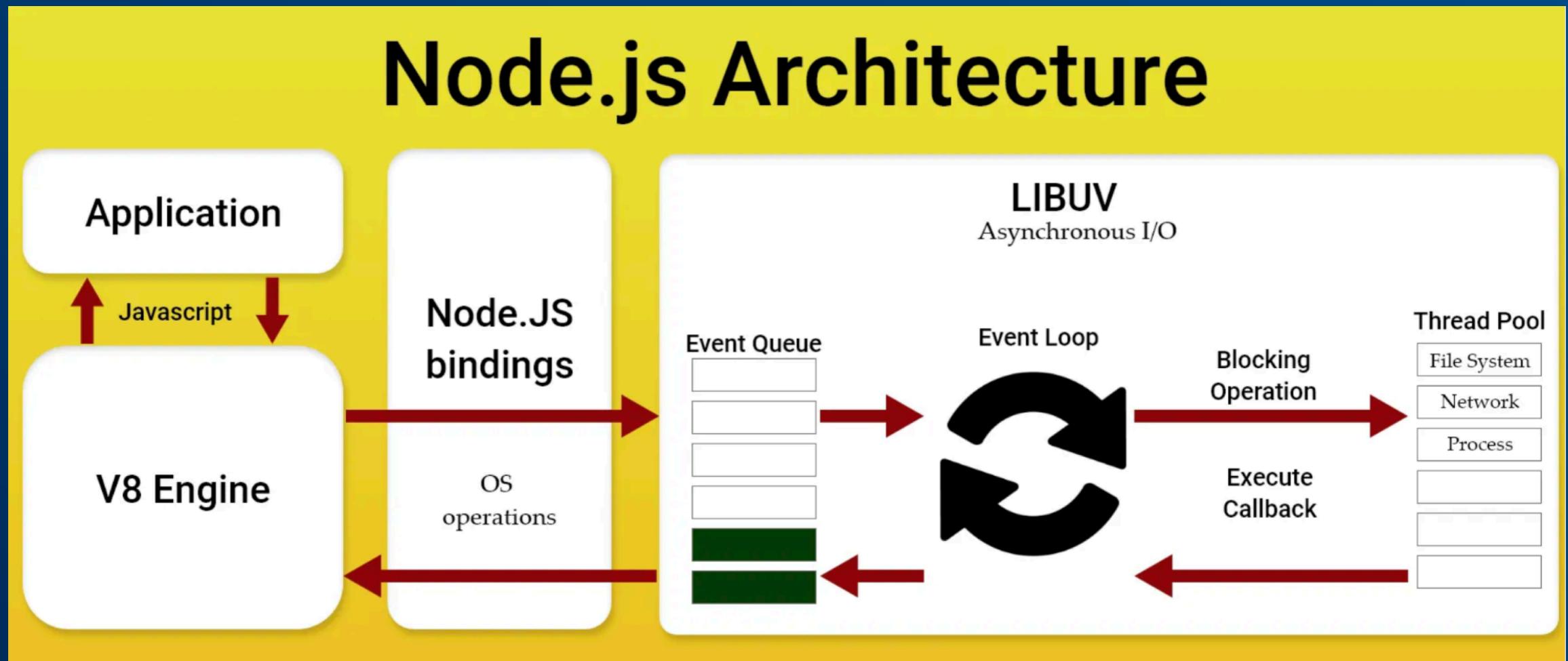
**Non Blocking, Asynchronous and Event Driven**



# NodeJS Architecture



# NodeJS Architecture

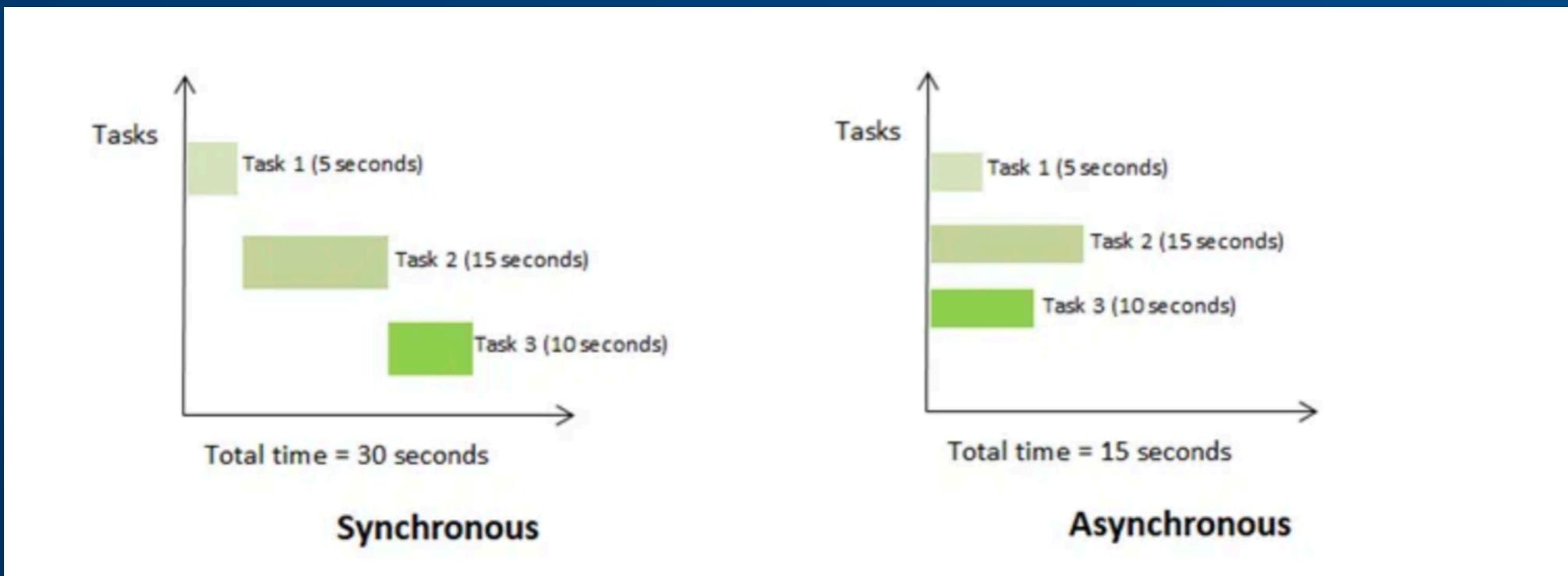


# Advantages Node JS

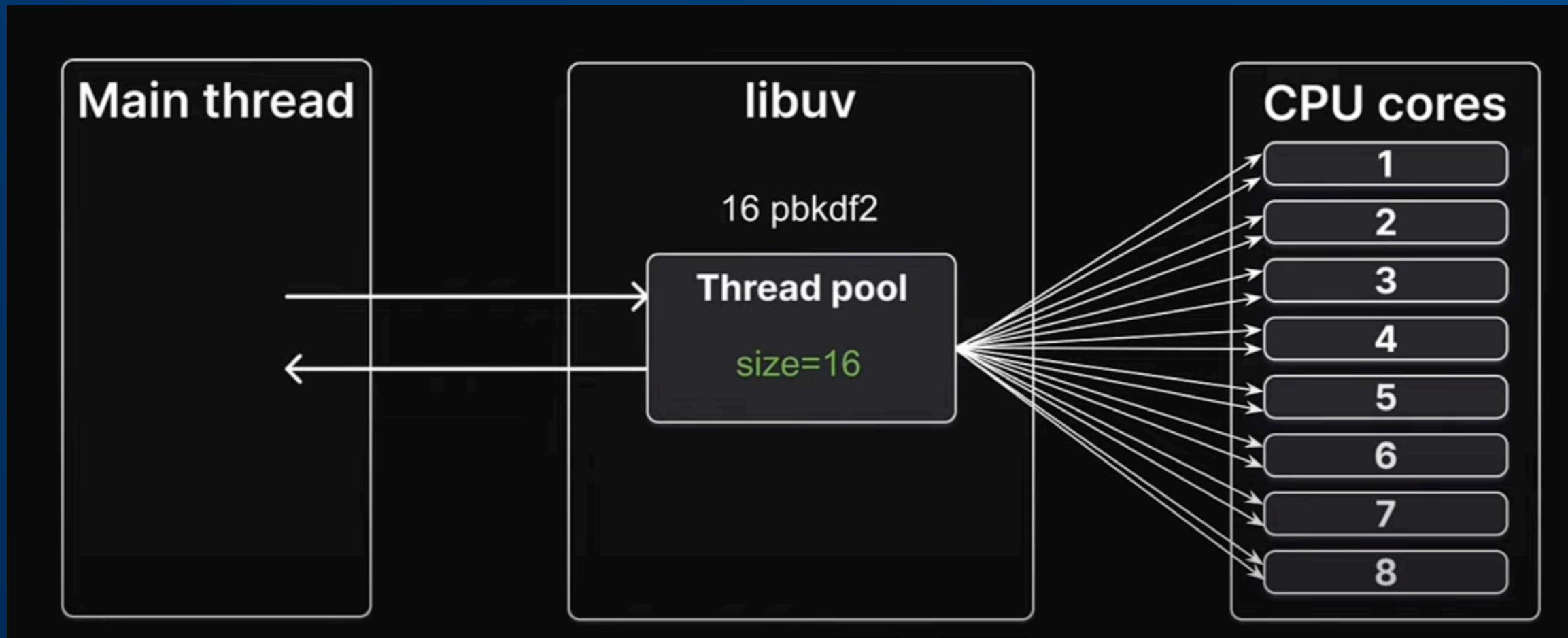
---

- Lightning-Fast Speed with Single thread and event driven
- Easy and fast database access
- Cross-Platform Compatibility
- Front end to Back end applications can be developed over Javascript
- Huge set of third party libraries are available and support
- Popular Major applications such as Netflix, LinkedIn, NASA and more

# Synchronous VS Asynchronous



# Thread Pool



---

Day 2