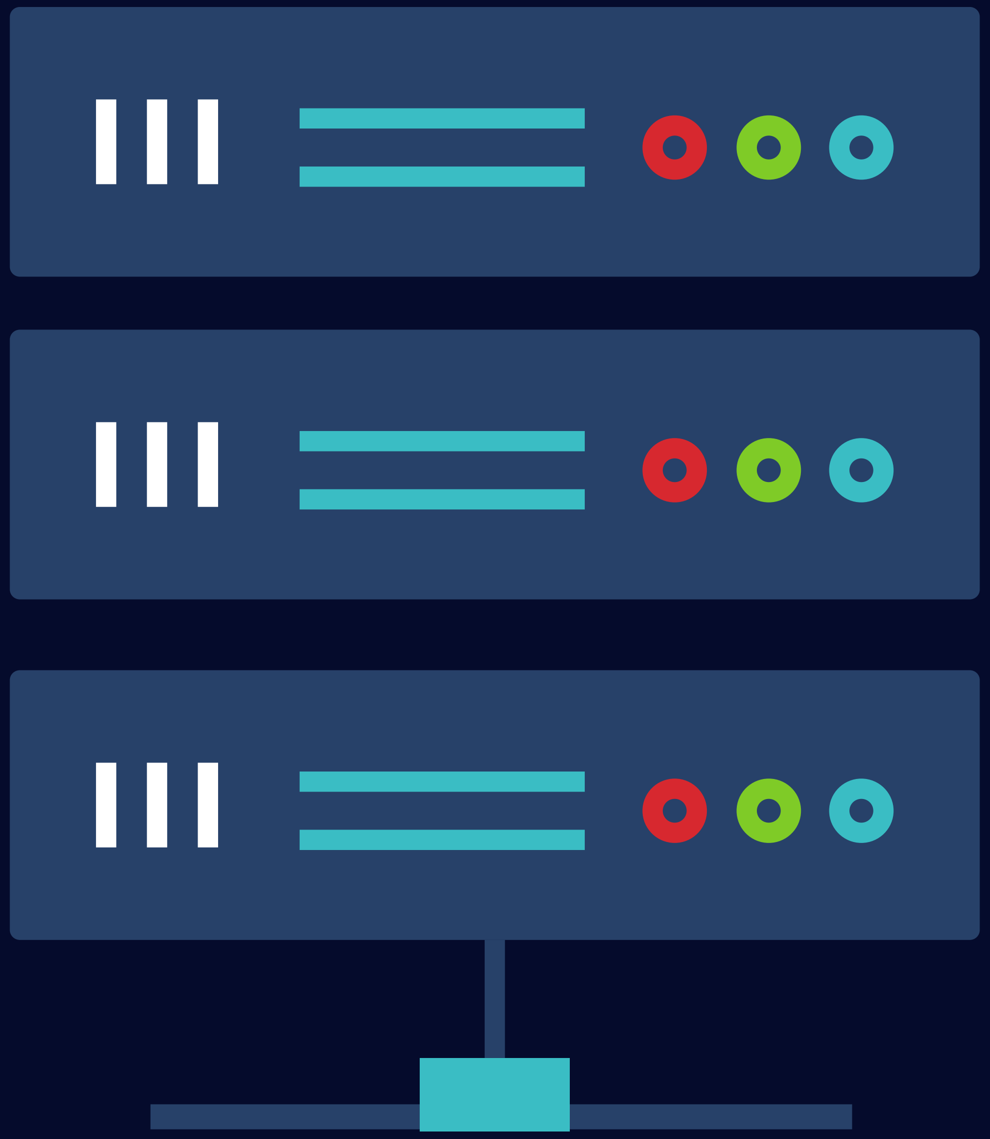
NODE JS

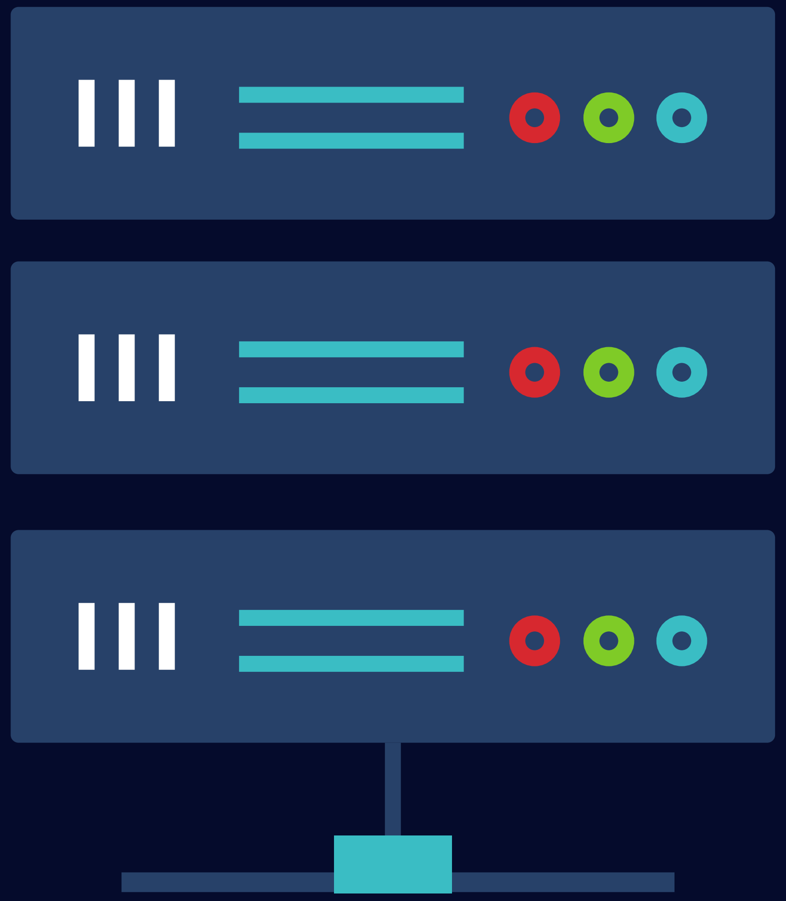
what is backend Development?

Components of a web application

What is a server?



What is a server



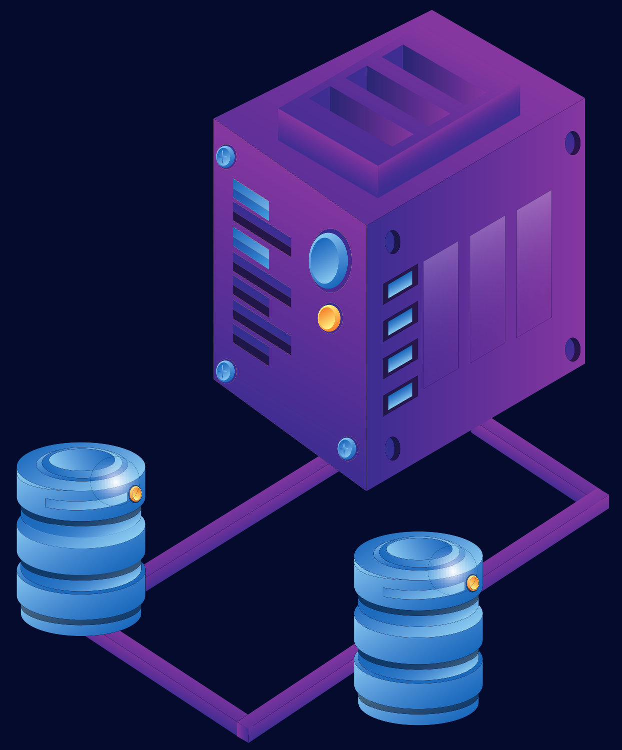
It is a program that accepts requests from clients and responds to those requests

NODE JS

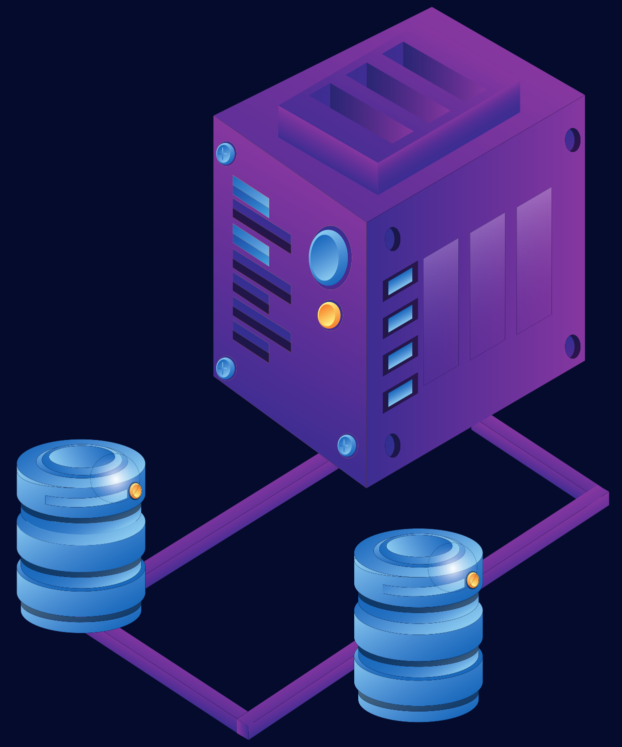
NODE JS

NODE JS

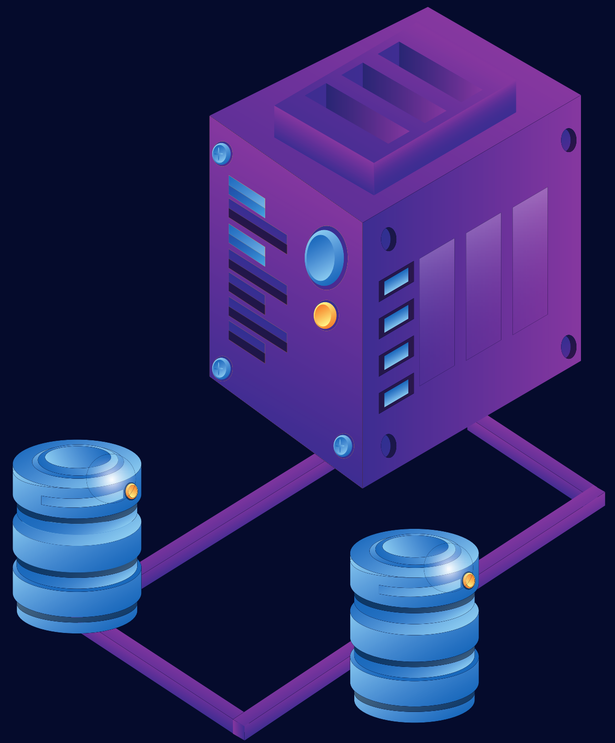
NODE JS

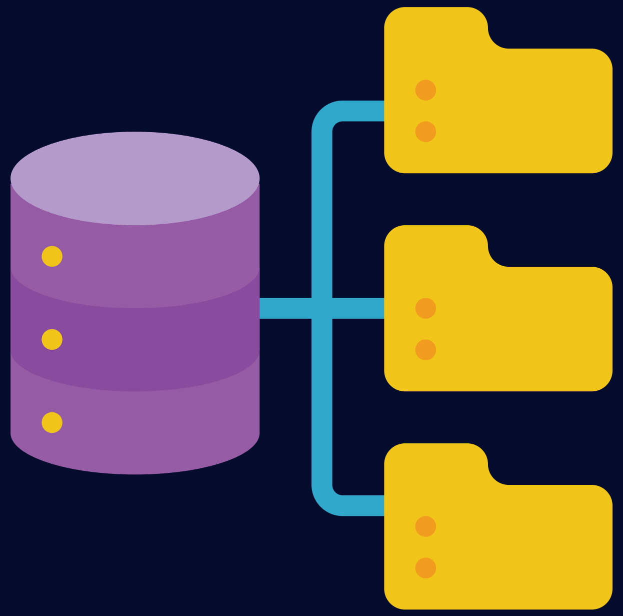


NODE JS



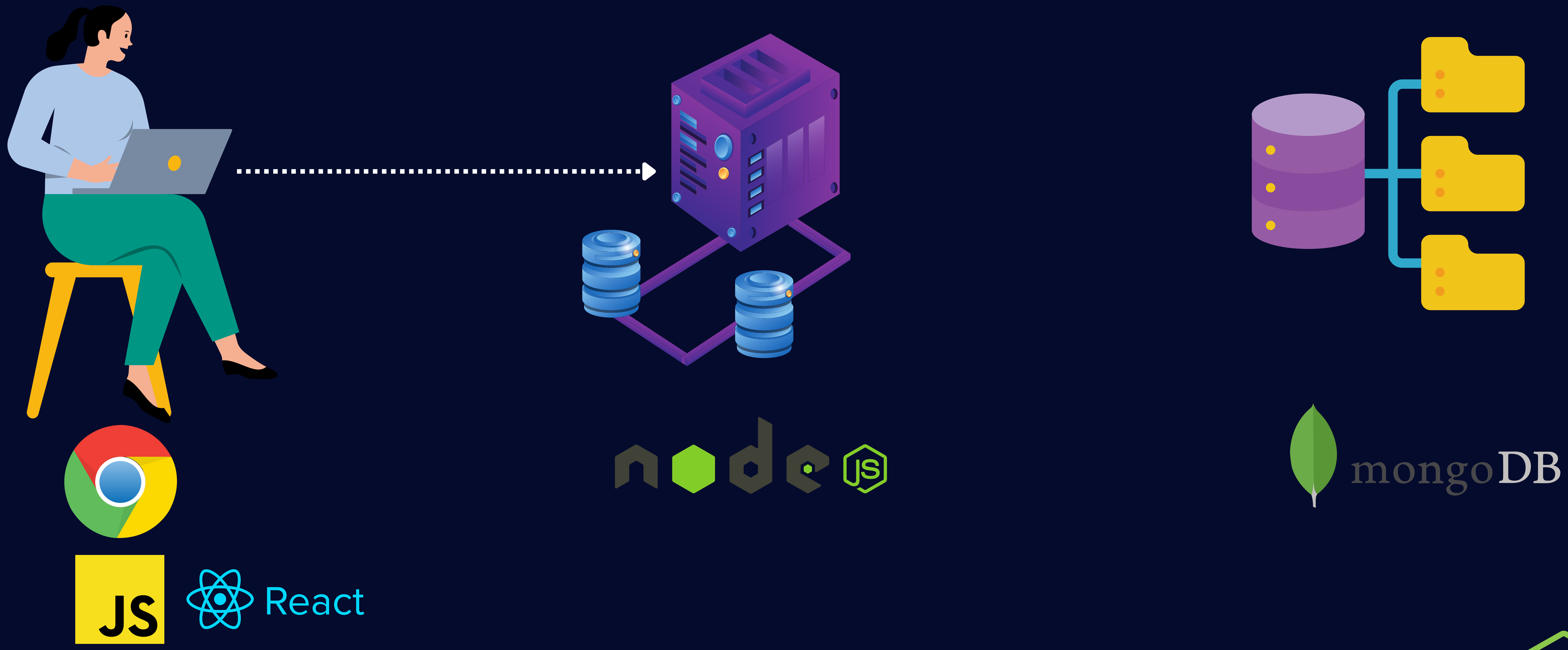
NODE JS



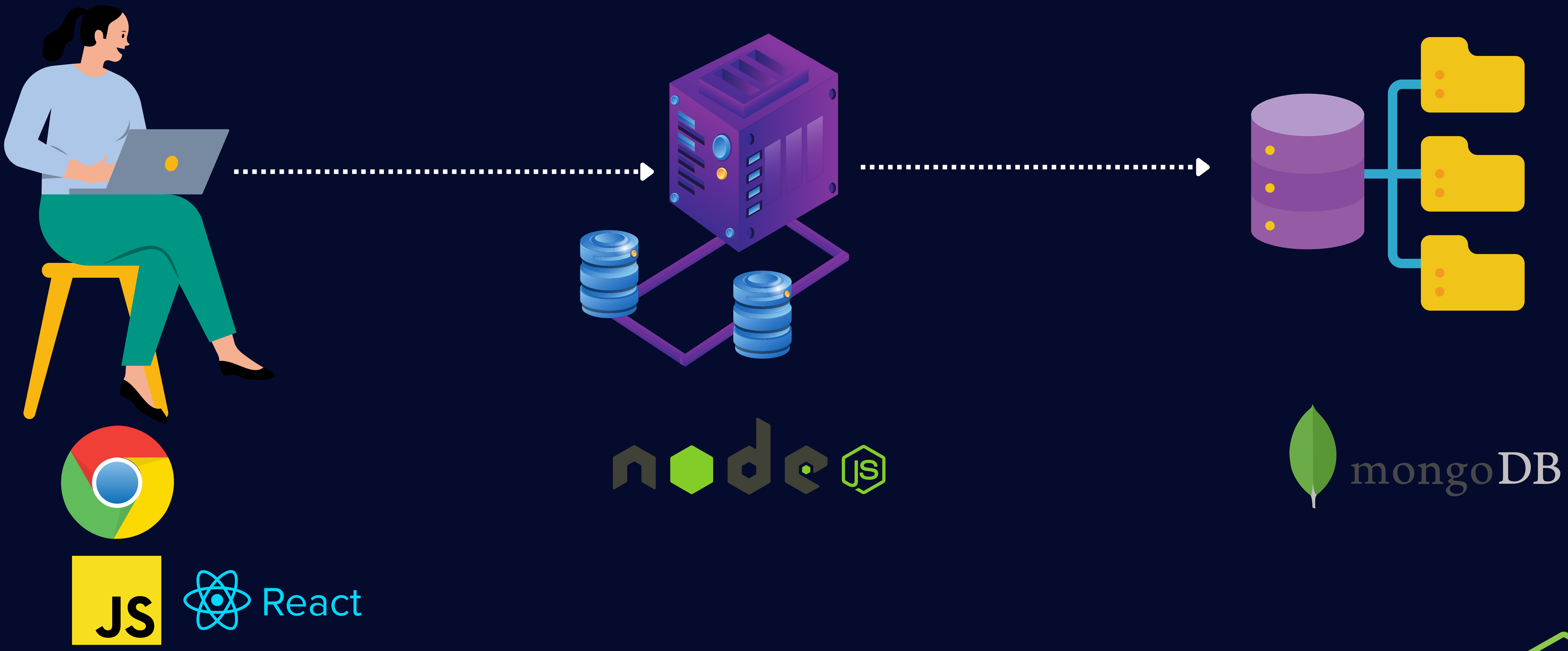


NODE JS

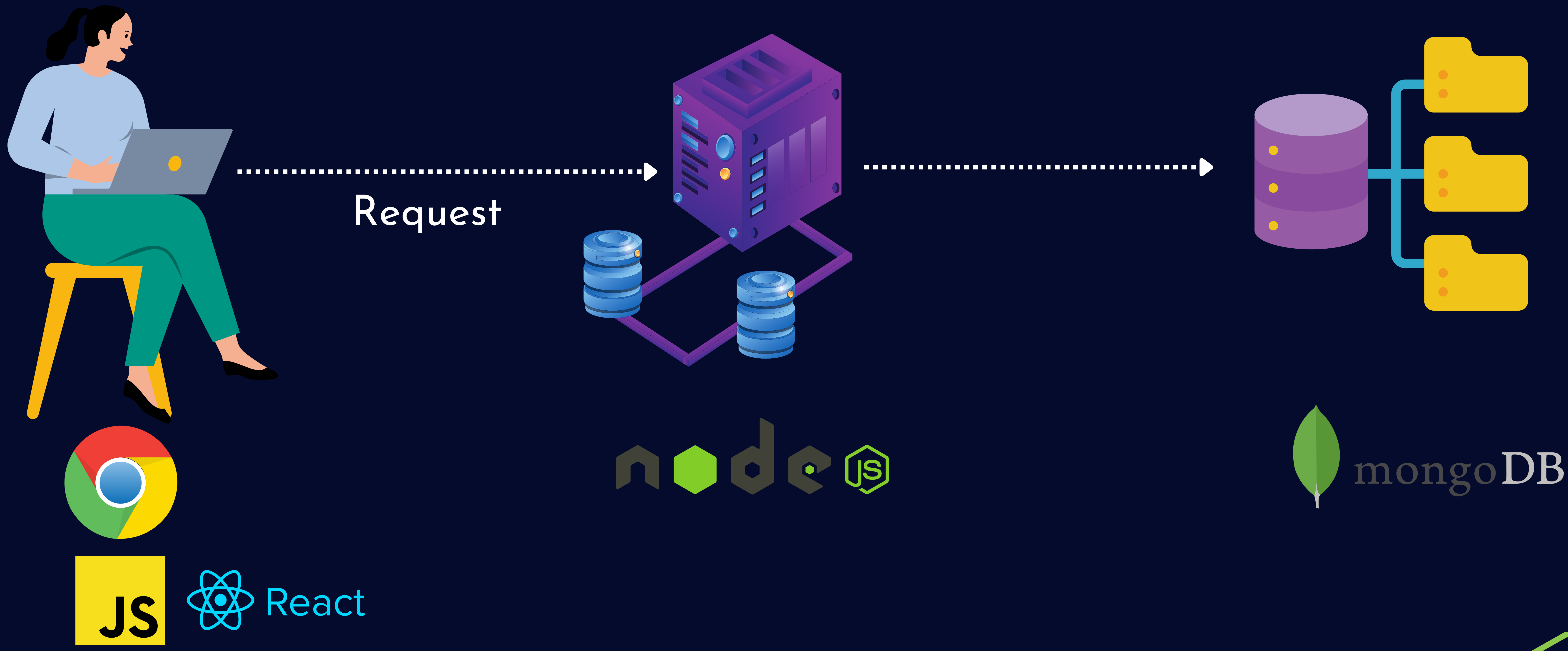
NODE JS



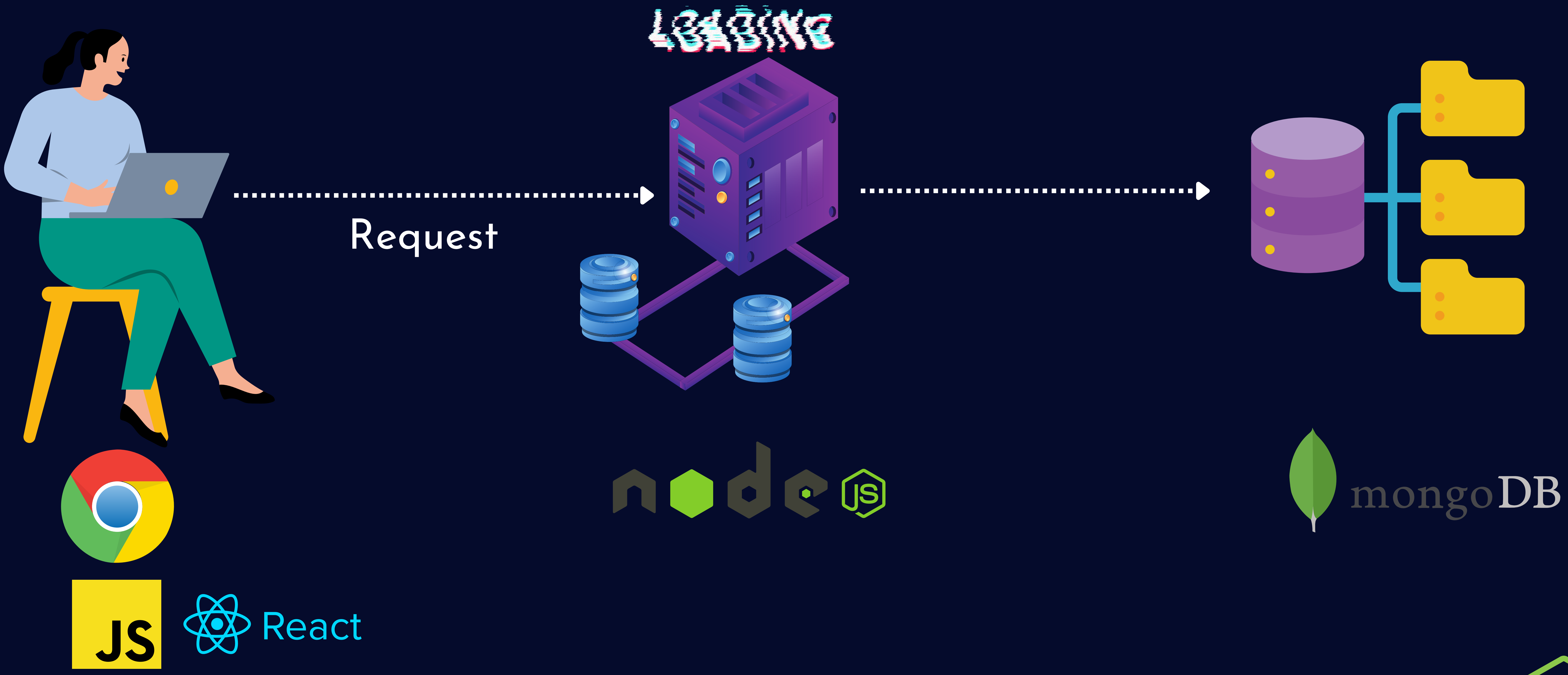
NODE JS



NODE JS

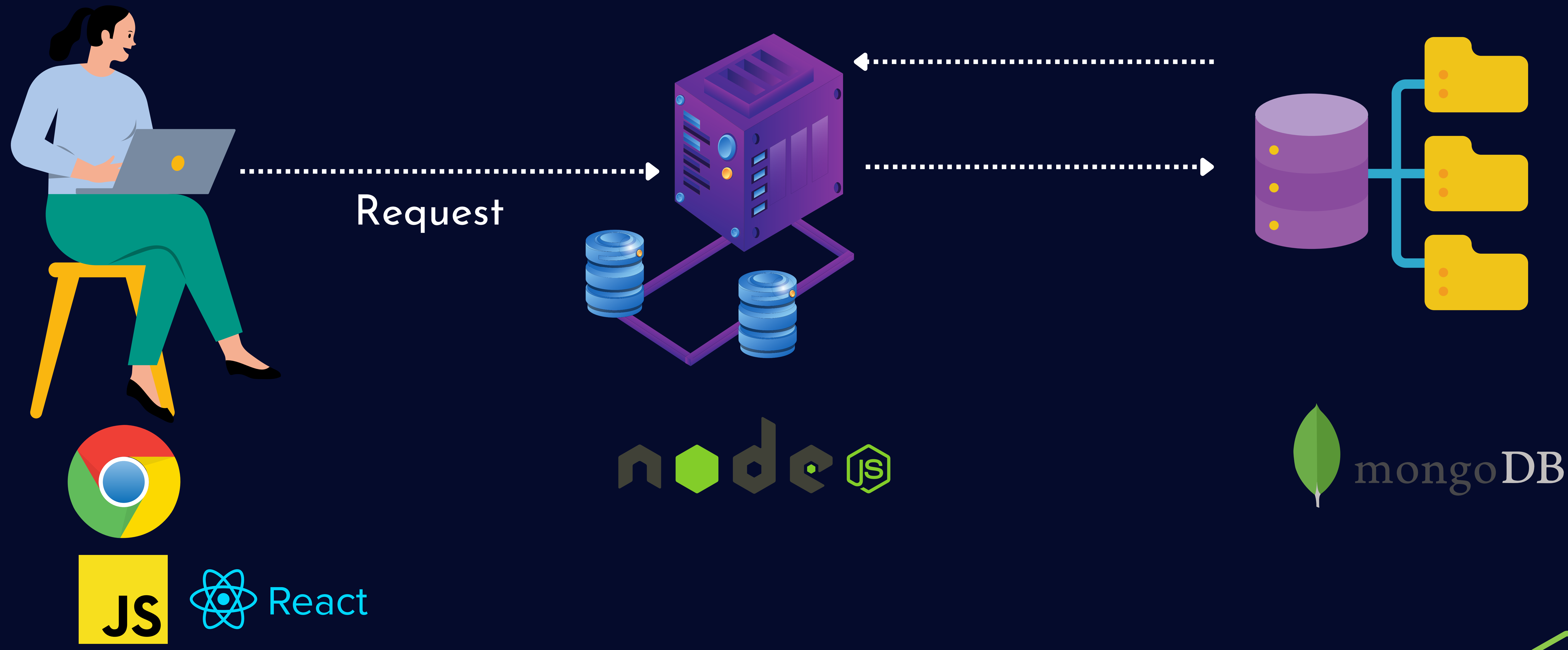


NODE JS

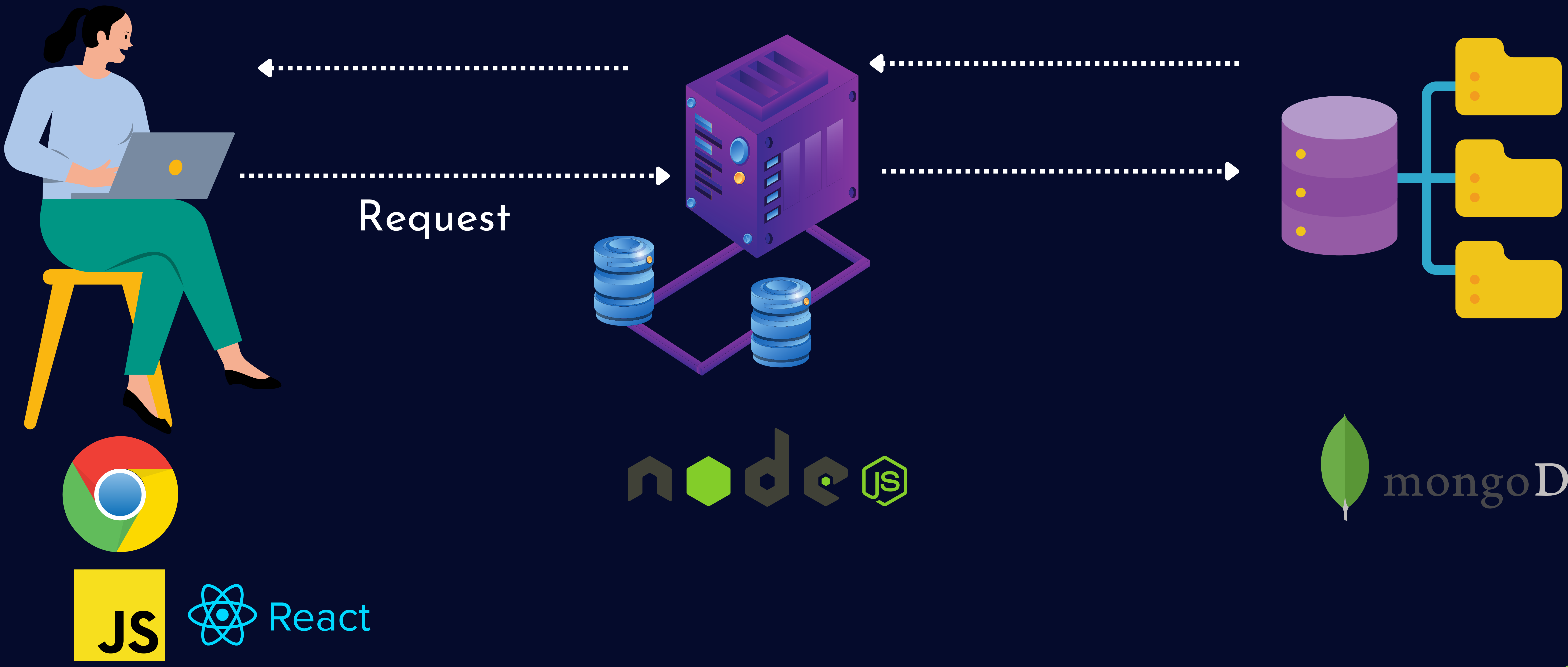


Request

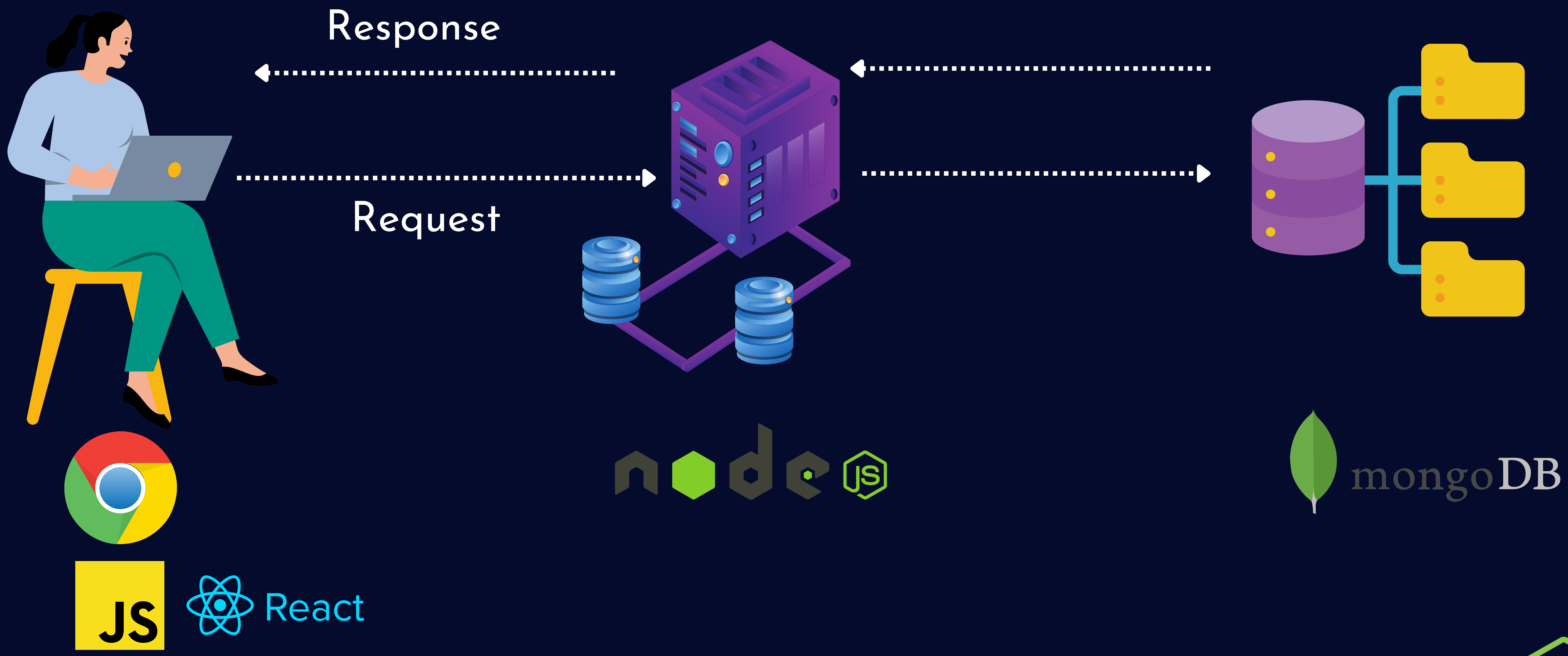
NODE JS



NODE JS



NODE JS



NODE JS

NODE JS

NODE JS

backend lopment

NODE JS

what is backend Development?

NODE JS

backend lopment

A backend developer is a software engineer who specializes in the development of server-side applications

backend Developer Roles

NODE JS

backend Developer Roles

- Implement business logic of an application

NODE JS

backend Developer Roles

- Implement business logic of an application
- Act as the server-side counterpart to front-end web developers

NODE JS

backend Developer Roles

- Implement business logic of an application
- Act as the server-side counterpart to front-end web developers
- Write, test and debug server-side code

NODE JS

backend Developer Roles

- Implement business logic of an application
- Act as the server-side counterpart to front-end web developers
- Write, test and debug server-side code
- Design, develop and implement web-based applications

NODE JS

backend Developer Roles

- Implement business logic of an application
- Act as the server-side counterpart to front-end web developers
- Write, test and debug server-side code
- Design, develop and implement web-based applications
- Develop and maintain server-side APIs

NODE JS

backend Developer Roles

- Implement business logic of an application
- Act as the server-side counterpart to front-end web developers
- Write, test and debug server-side code
- Design, develop and implement web-based applications
- Develop and maintain server-side APIs
- Manage server-side code libraries

NODE JS

backend Developer Roles

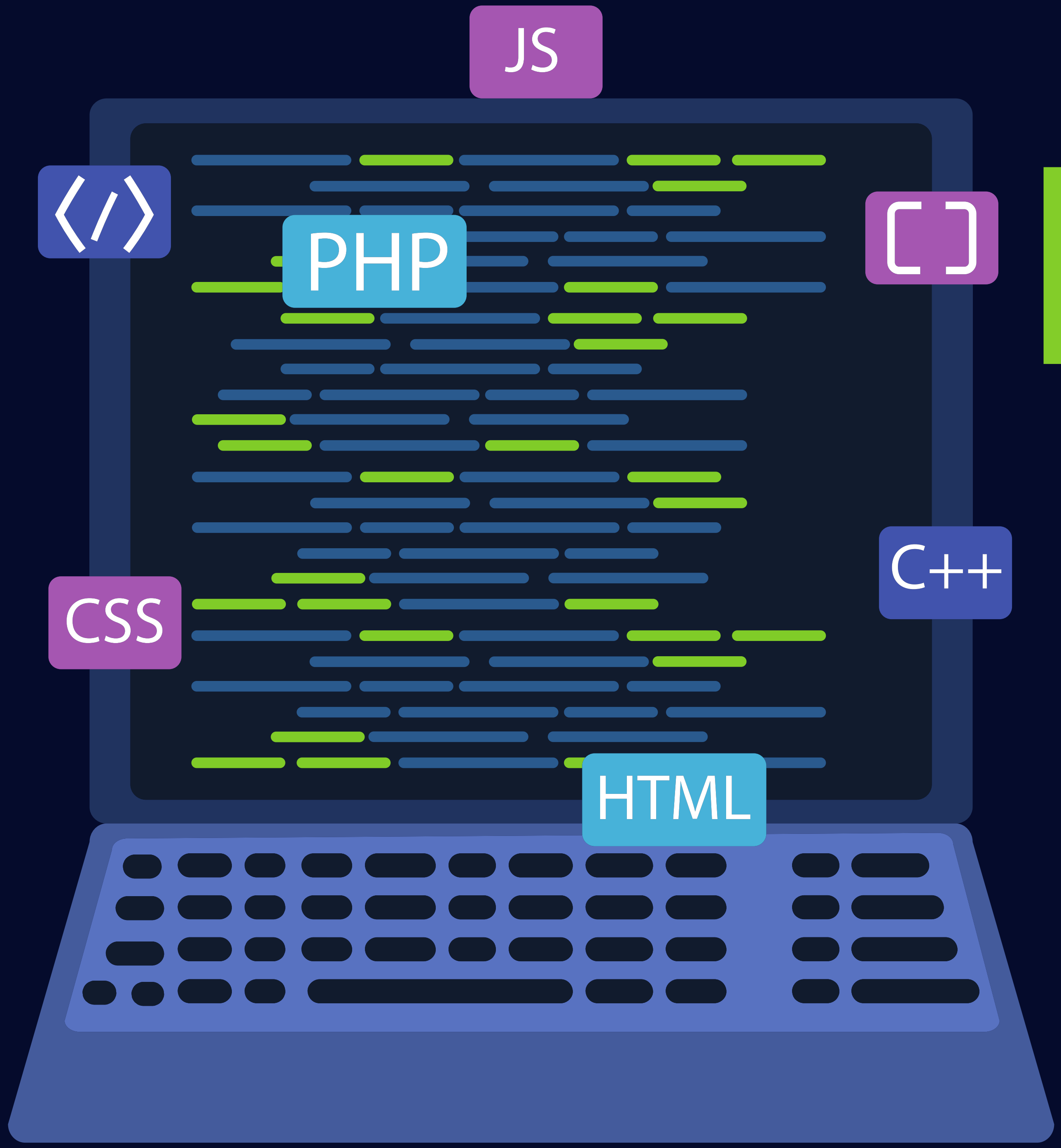
- Implement business logic of an application
- Act as the server-side counterpart to front-end web developers
- Write, test and debug server-side code
- Design, develop and implement web-based applications
- Develop and maintain server-side APIs
- Manage server-side code libraries
- Optimize server-side code for performance

NODE JS

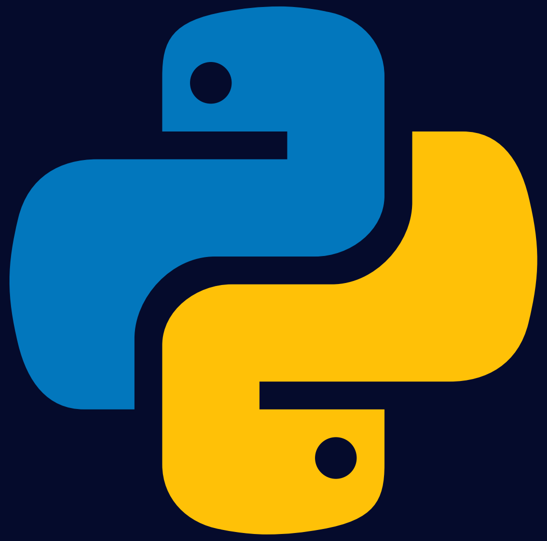
backend Developer Roles

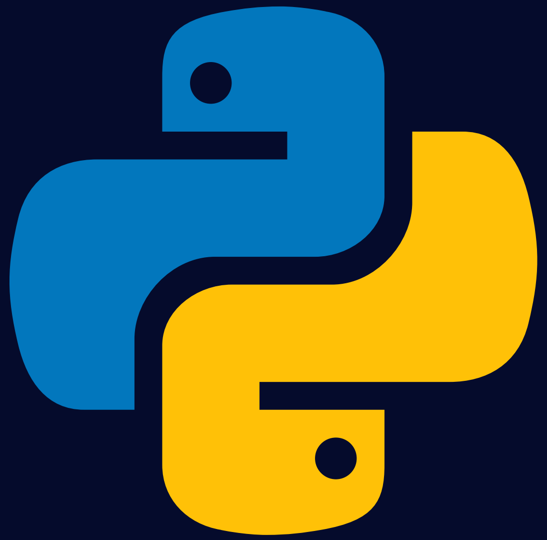
- Implement business logic of an application
- Act as the server-side counterpart to front-end web developers
- Write, test and debug server-side code
- Design, develop and implement web-based applications
- Develop and maintain server-side APIs
- Manage server-side code libraries
- Optimize server-side code for performance
- Handle server security issues

www.inovotekacademy.com i-novotek academy inovotekacademy



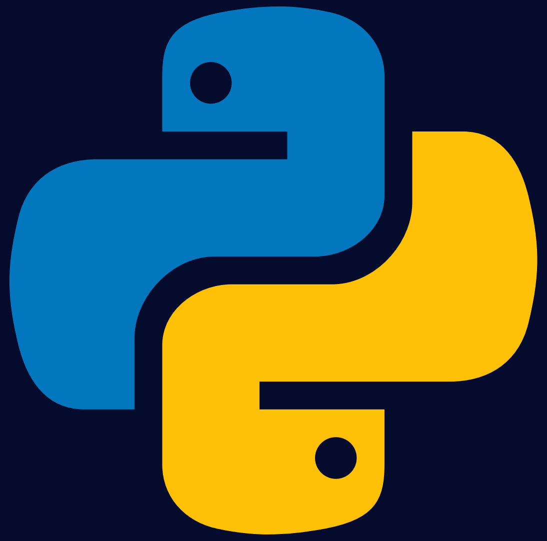
Programming Languages



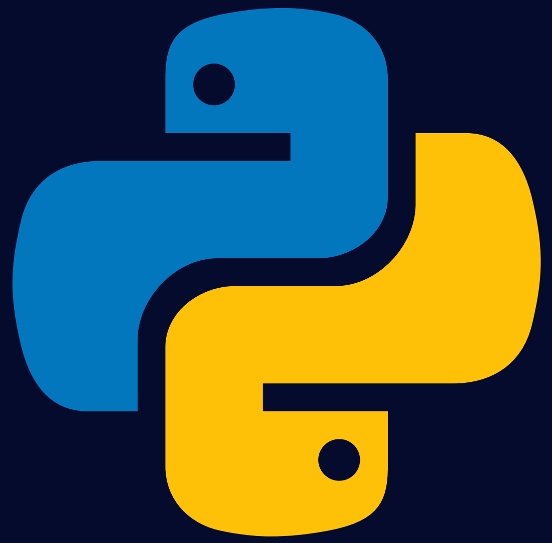


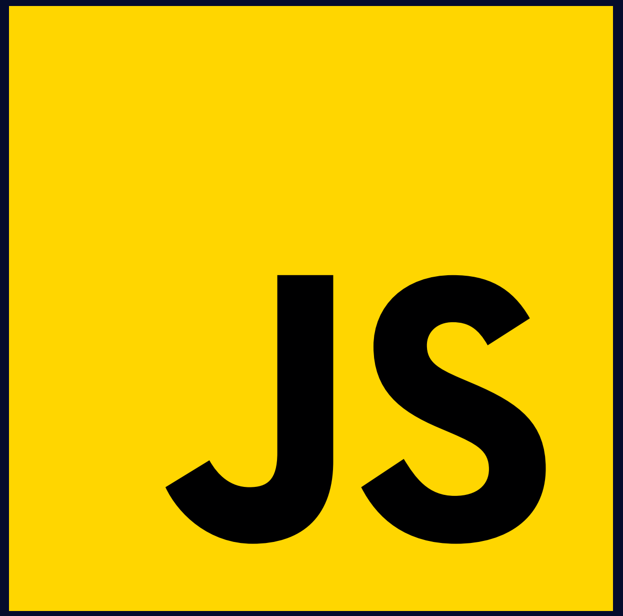
C++

NODE JS

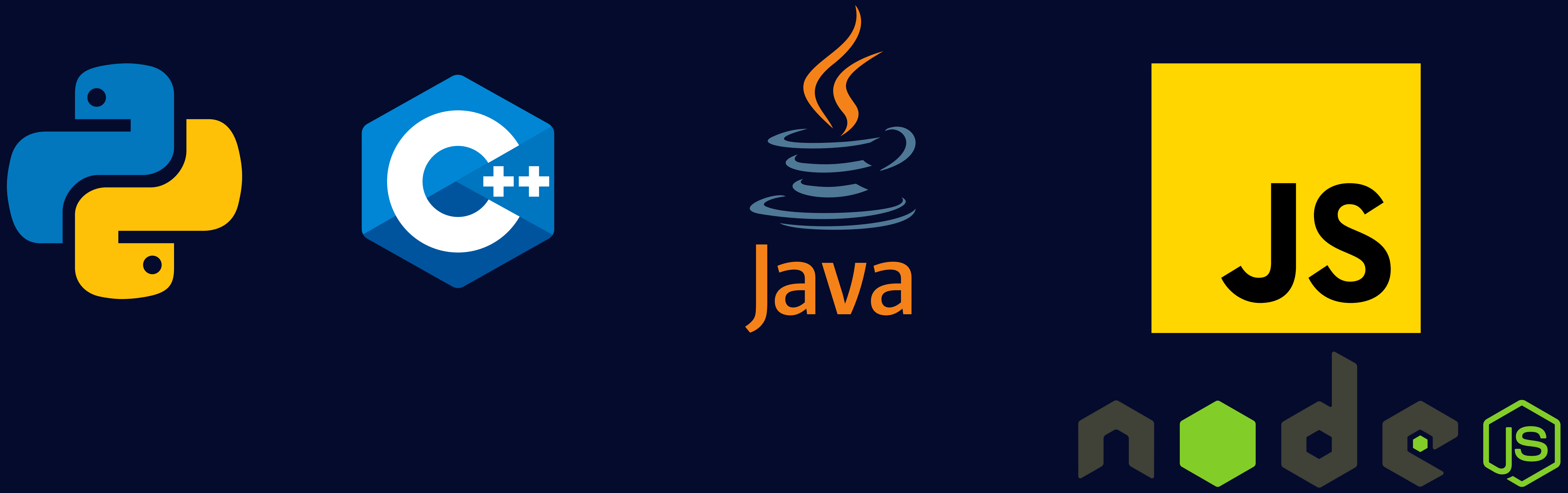


NODE JS





NODE JS



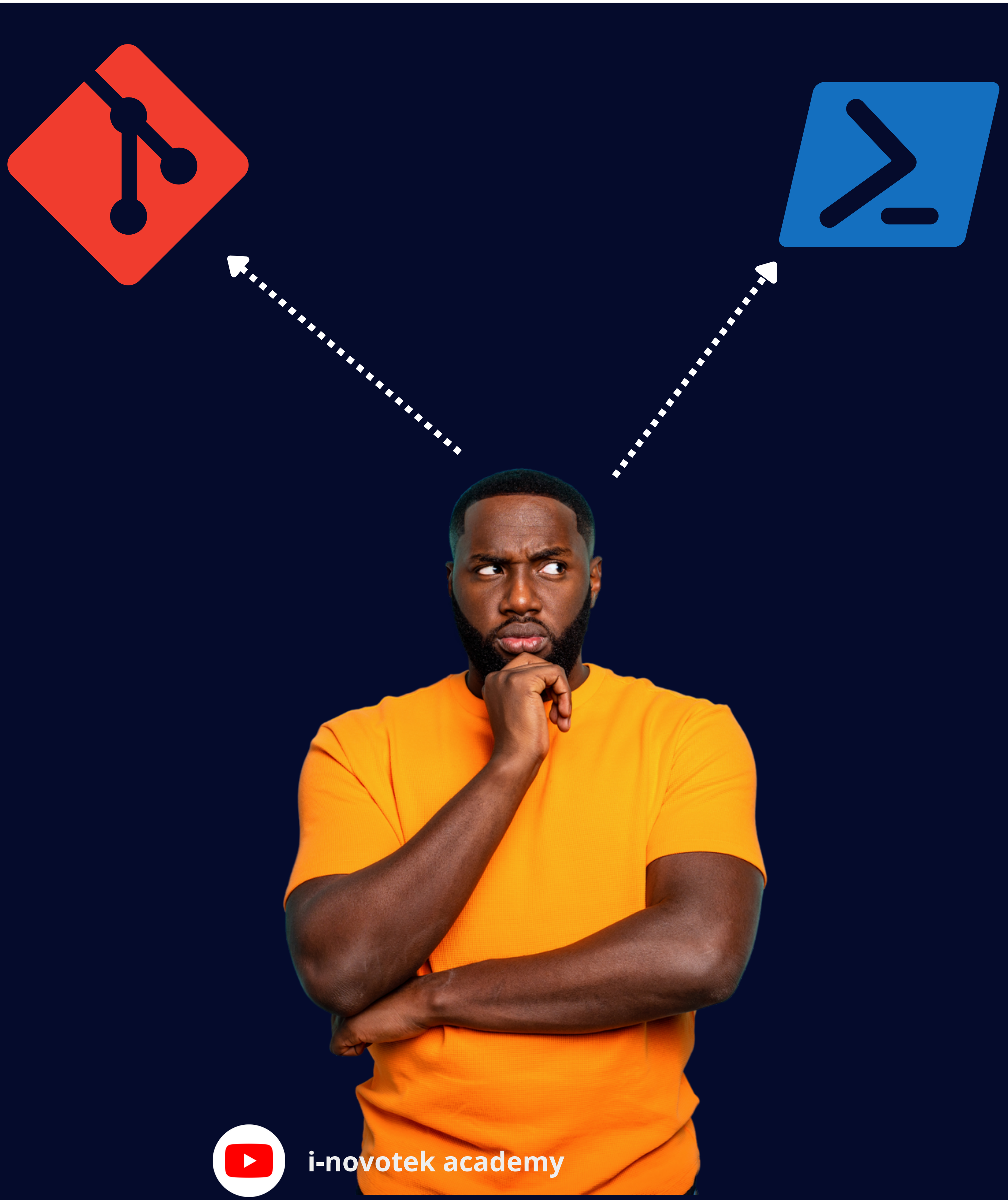
NODE JS

What You need to know

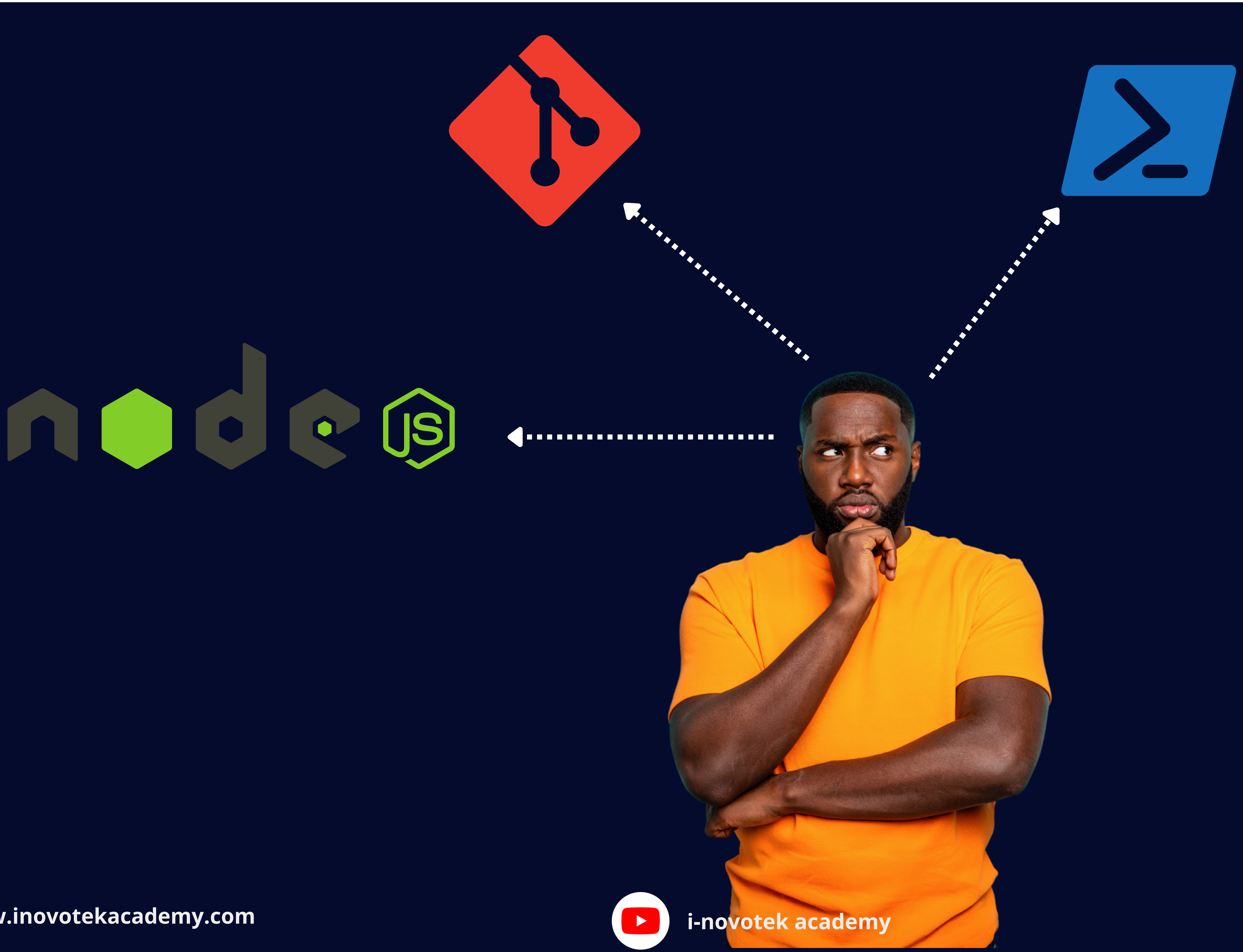
NODE JS

NODE JS

NODE JS



NODE JS



NODE JS

Command Line

Confusing Terminologies

Confusing Terminologies

Confusing Terminologies

👉 Terminal

Confusing Terminologies

👉 Terminal

👉 Shell

Confusing Terminologies

👉 Terminal

👉 Shell

👉 Command Line

Confusing Terminologies

Confusing Terminologies

Terminal

Confusing Terminologies

Terminal

The terminal is a device that gives you access to the console of your computer.

Confusing Terminologies

Terminal

The terminal is a device that gives you access to the console of your computer.

Shell

Confusing Terminologies

Terminal

The terminal is a device that gives you access to the console of your computer.

Shell

The shell is the program that runs in the terminal, giving users a command prompt. Popular shells include sh, bash, zsh, fish, ash, and csh. The shell processes commands and returns output, like Bash in Linux.

Confusing Terminologies

Terminal

The terminal is a device that gives you access to the console of your computer.

Shell

The shell is the program that runs in the terminal, giving users a command prompt. Popular shells include sh, bash, zsh, fish, ash, and csh. The shell processes commands and returns output, like Bash in Linux.

Command Line

Confusing Terminologies

Terminal

The terminal is a device that gives you access to the console of your computer.

Shell

The shell is the program that runs in the terminal, giving users a command prompt. Popular shells include sh, bash, zsh, fish, ash, and csh. The shell processes commands and returns output, like Bash in Linux.

Command Line

A command-line interface is any kind of interface that is used by entering (textual) commands. The terminal is one of these, but some programs have their own command-line interfaces.

GIT

GIT (Version Control System)

GIT (Version Control System)

A GIT is a software that helps developers keep track of the changes they make to their code. This is helpful because it makes it easy to go back and see what changes were made and when.

GIT (Version Control System)

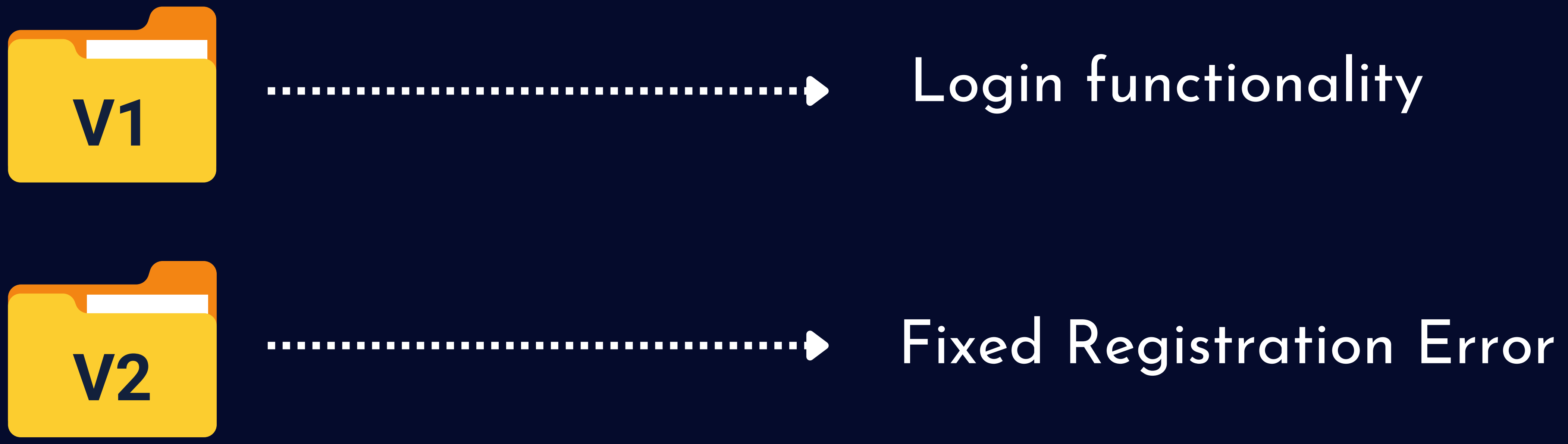
A GIT is a software that helps developers keep track of the changes they make to their code. This is helpful because it makes it easy to go back and see what changes were made and when.

GIT (Version Control System)

GIT (Version Control System)

V1
Login functionality

GIT (Version Control System)

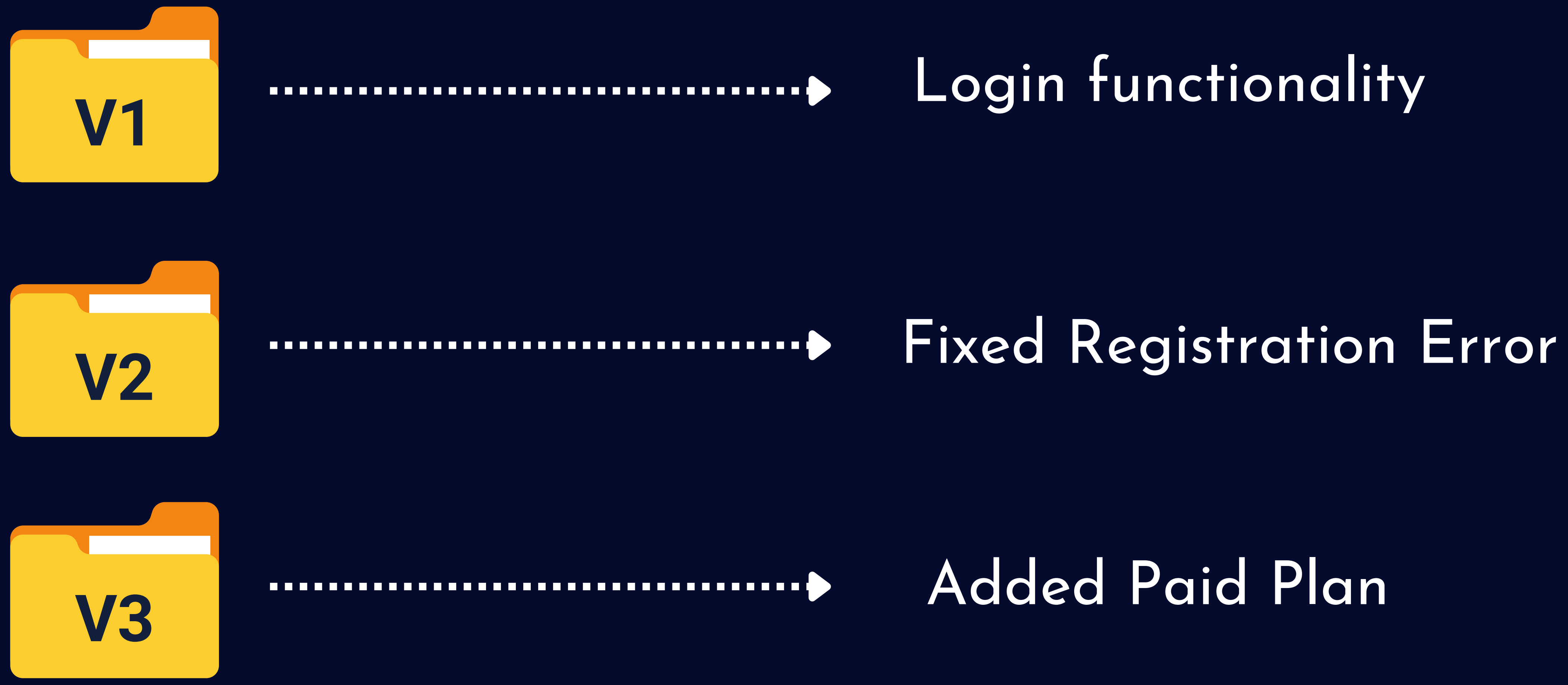


Login functionality

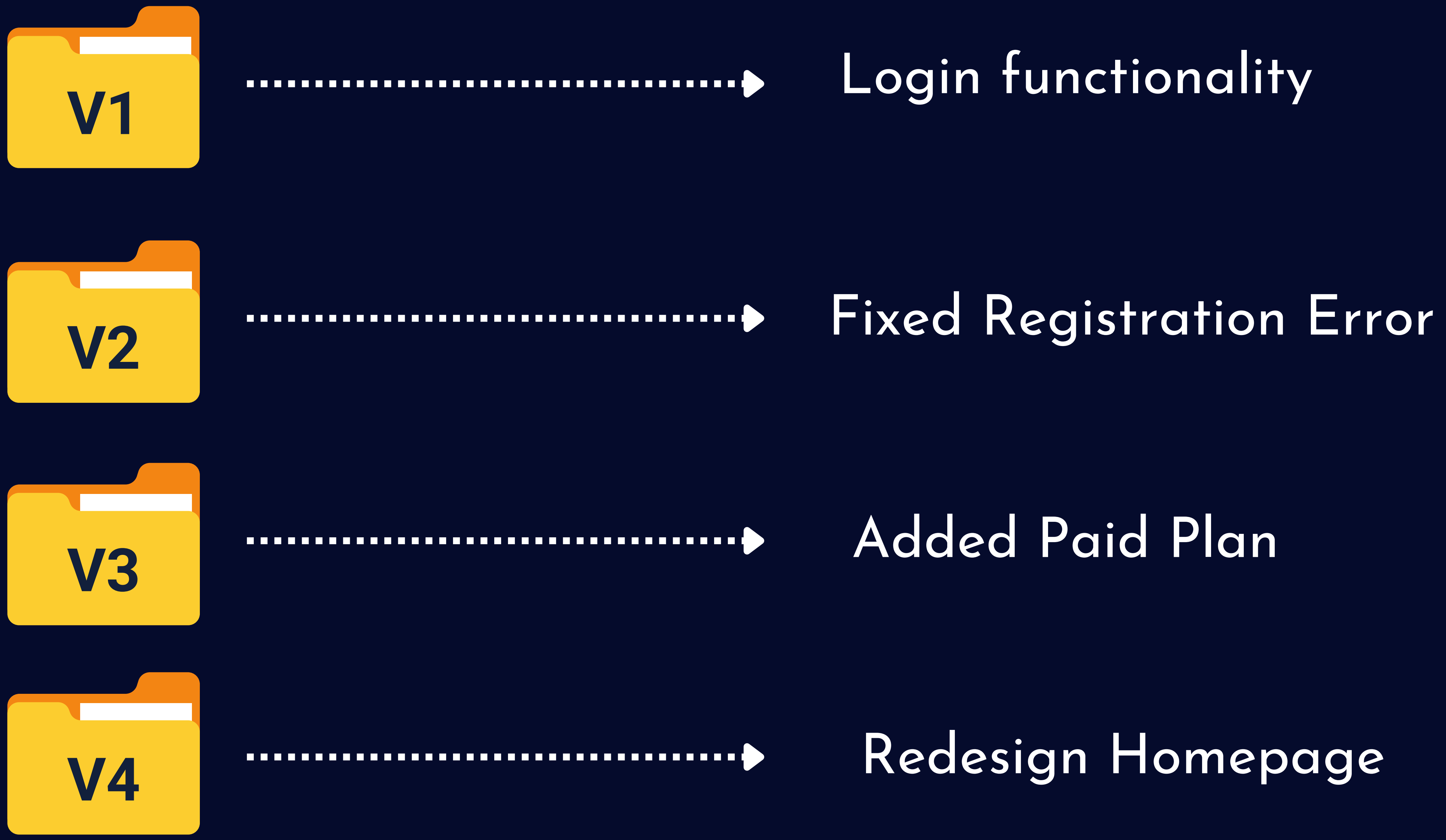
Fixed Registration Error

GIT

GIT (Version Control System)



GIT (Version Control System)



COMMON COMMANDLINE COMMANDS

COMMON COMMANDS

COMMON COMMANDS

- **pwd** → Print Working Directory

COMMON COMMANDS

- **pwd** → Print Working Directory
- **mkdir** → Makes a new directory

COMMON COMMANDS

- pwd ⟶ Print Working Directory
- mkdir ⟶ Makes a new directory
- touch ⟶ Create a file

COMMON COMMANDS

- pwd → Print Working Directory
- mkdir → Makes a new directory
- touch → Create a file
- ls → Lists all files and folders in the current directory

COMMON COMMANDS

- pwd ⟶ Print Working Directory
- mkdir ⟶ Makes a new directory
- touch ⟶ Create a file
- ls ⟶ Lists all files and folders in the current directory
- cd ⟶ Change Directory

COMMON COMMANDS

pwd	→	Print Working Directory
mkdir	→	Makes a new directory
touch	→	Create a file
ls	→	Lists all files and folders in the current directory
cd	→	Change Directory
rm	→	Remove a file

COMMON COMMANDS

pwd	→	Print Working Directory
mkdir	→	Makes a new directory
touch	→	Create a file
ls	→	Lists all files and folders in the current directory
cd	→	Change Directory
rm	→	Remove a file
rmdir	→	Removes an empty directory

COMMON COMMANDS

- **pwd** → Print Working Directory
- **mkdir** → Makes a new directory
- **touch** → Create a file
- **ls** → Lists all files and folders in the current directory
- **cd** → Change Directory
- **rm** → Remove a file
- **rmdir** → Removes an empty directory
- **rm -rf** → Removes a folder and all of its contents. BE CAREFUL! This command cannot be undone!

Getting Started With Node JS

www.inovotekacademy.com

i-novotek academy

inovotekacademy

Node JS

What is Node JS ?

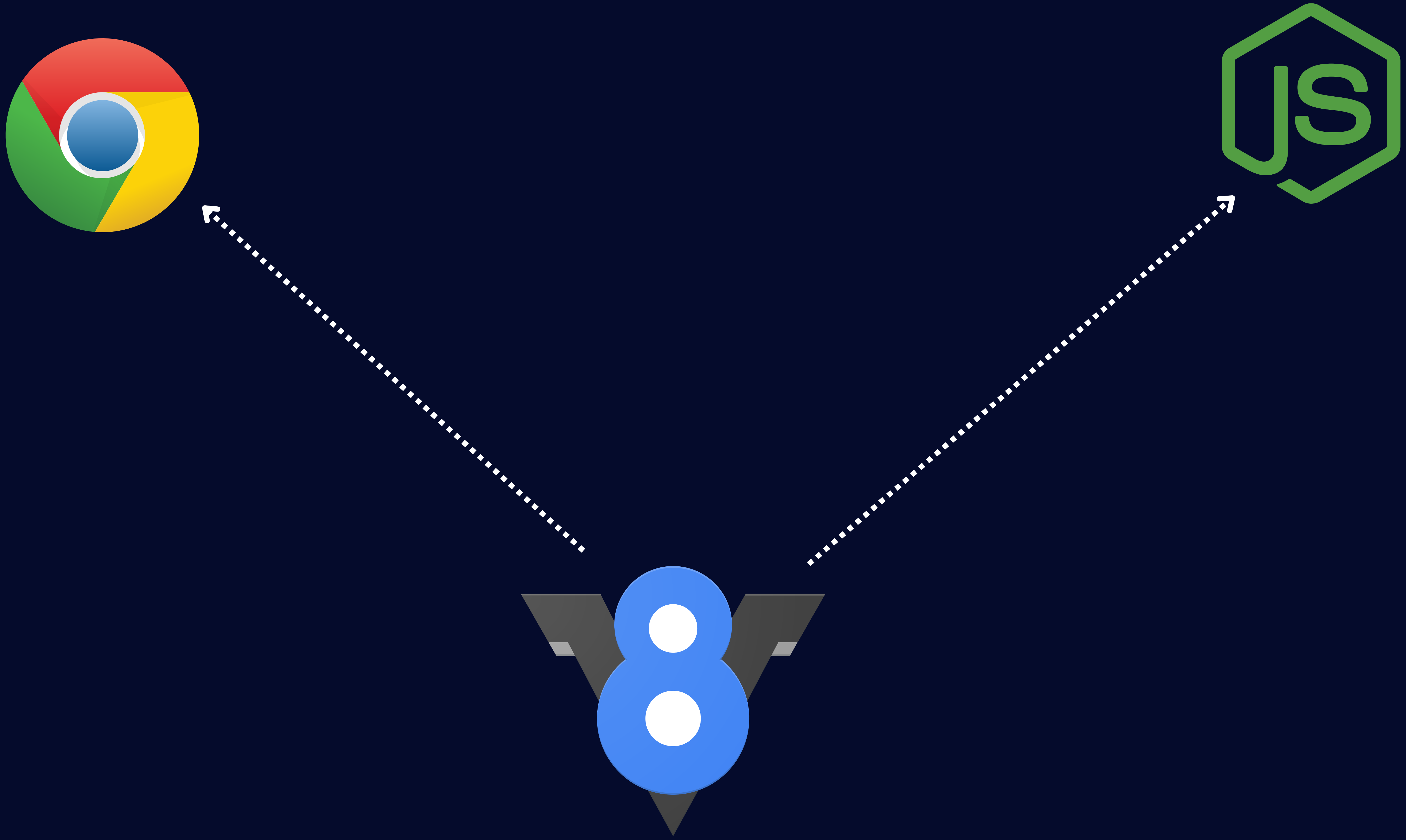
Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O. (input and output) model that makes it lightweight and efficient

JS Environments

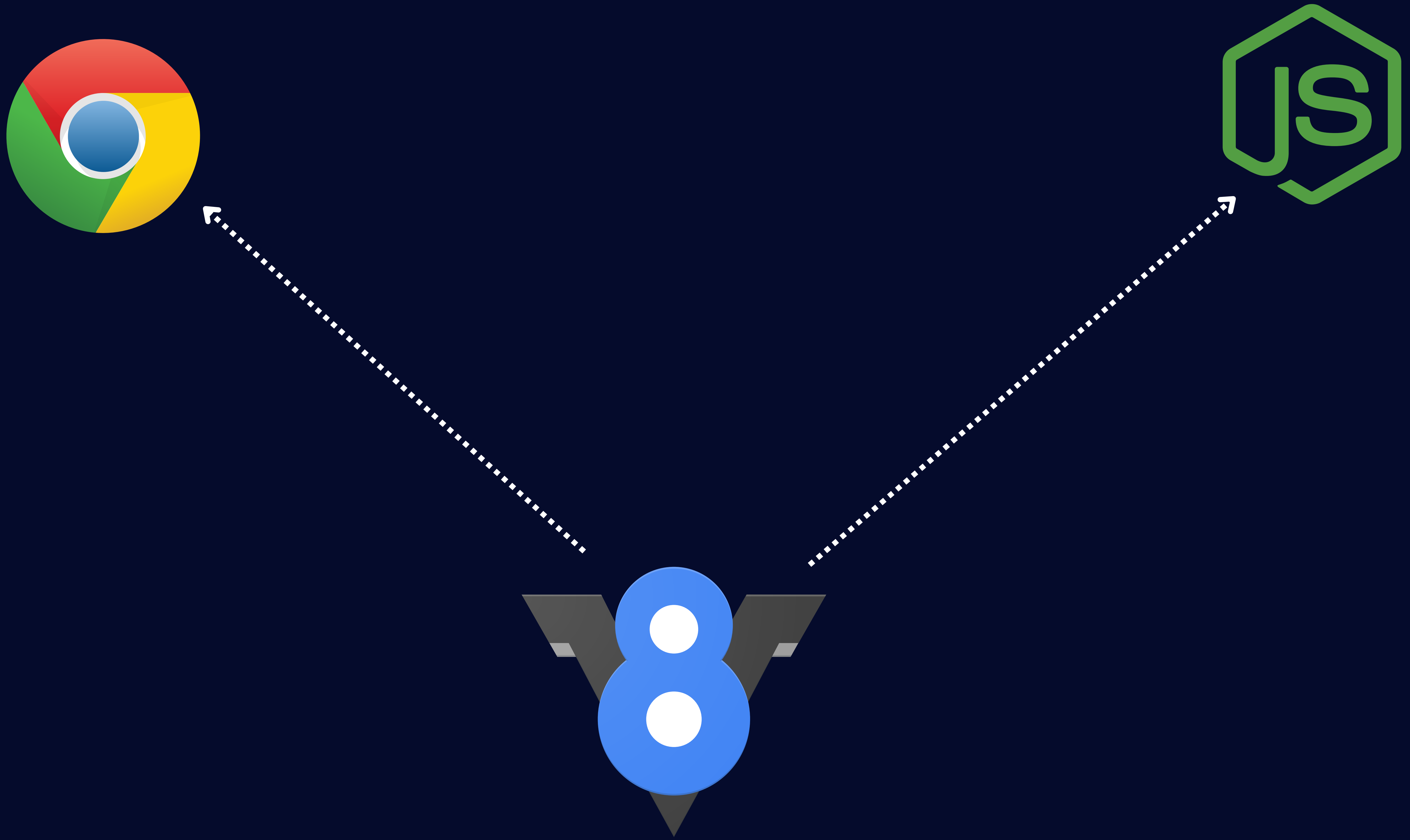
JS Environments

JS Environments

JS Environments



JS Environments



Node js was built on top of chrome v8 engine

NODE JS

Node js was built on top of chrome v8 engine

Why use Node JS?

Why use Node JS?

Why use Node JS?

- It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient

Why use Node JS?

- It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient
- It has a large and active open-source community

Why use Node JS?

- It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient
- It has a large and active open-source community
- It runs on various platforms, including Windows, Linux, and macOS.

Why use Node JS?

- It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient
- It has a large and active open-source community
- It runs on various platforms, including Windows, Linux, and macOS.
- It supports real-time data applications with features such as WebSockets and WebRTC

Uses of Node JS

Uses of Node JS

Uses of Node JS

- Developing web applications

Uses of Node JS

- Developing web applications
- Building chatbots

Uses of Node JS

- Developing web applications
- Building chatbots
- Creating real-time applications

Uses of Node JS

- Developing web applications
- Building chatbots
- Creating real-time applications
- Streaming video and audio

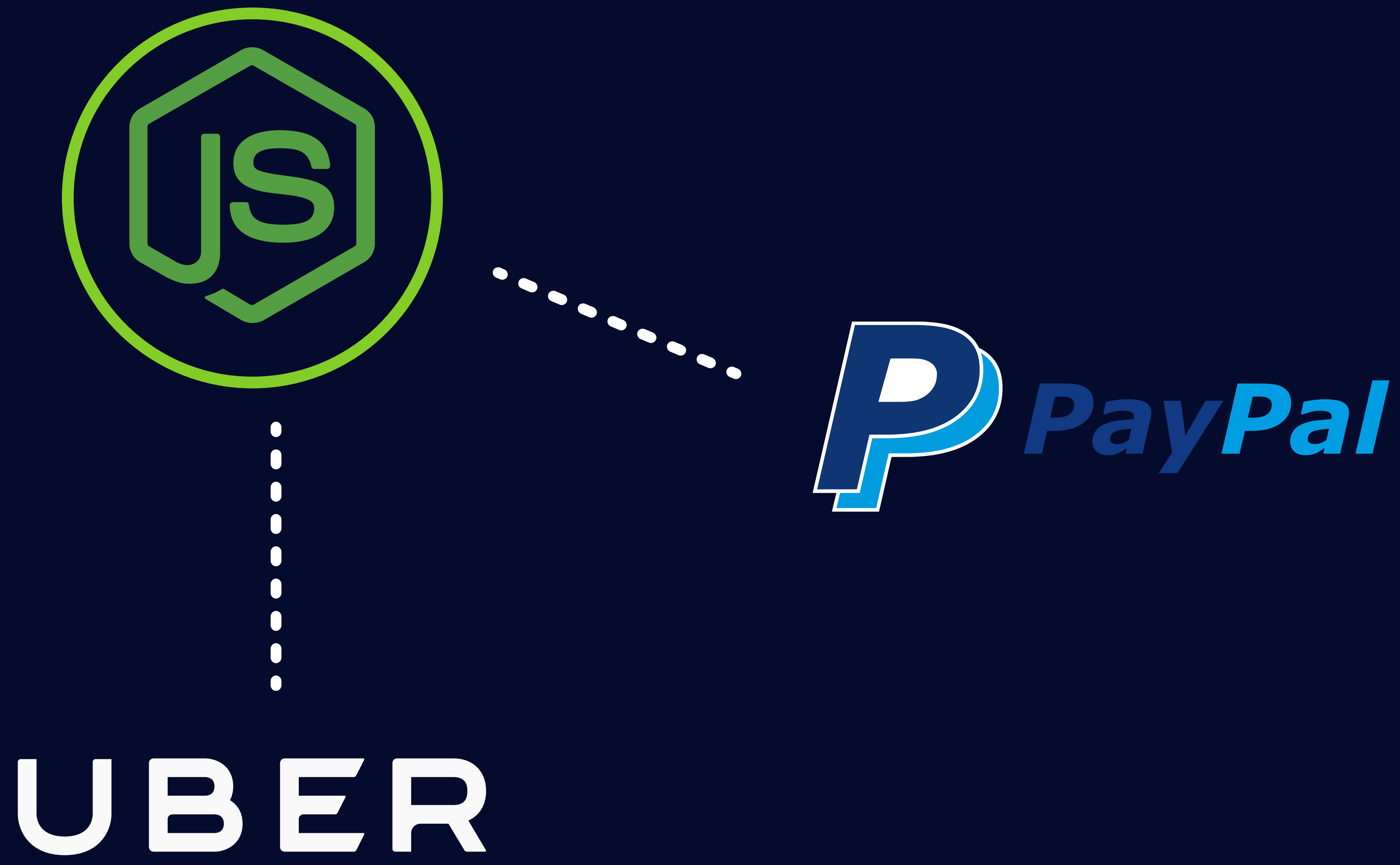
Uses of Node JS

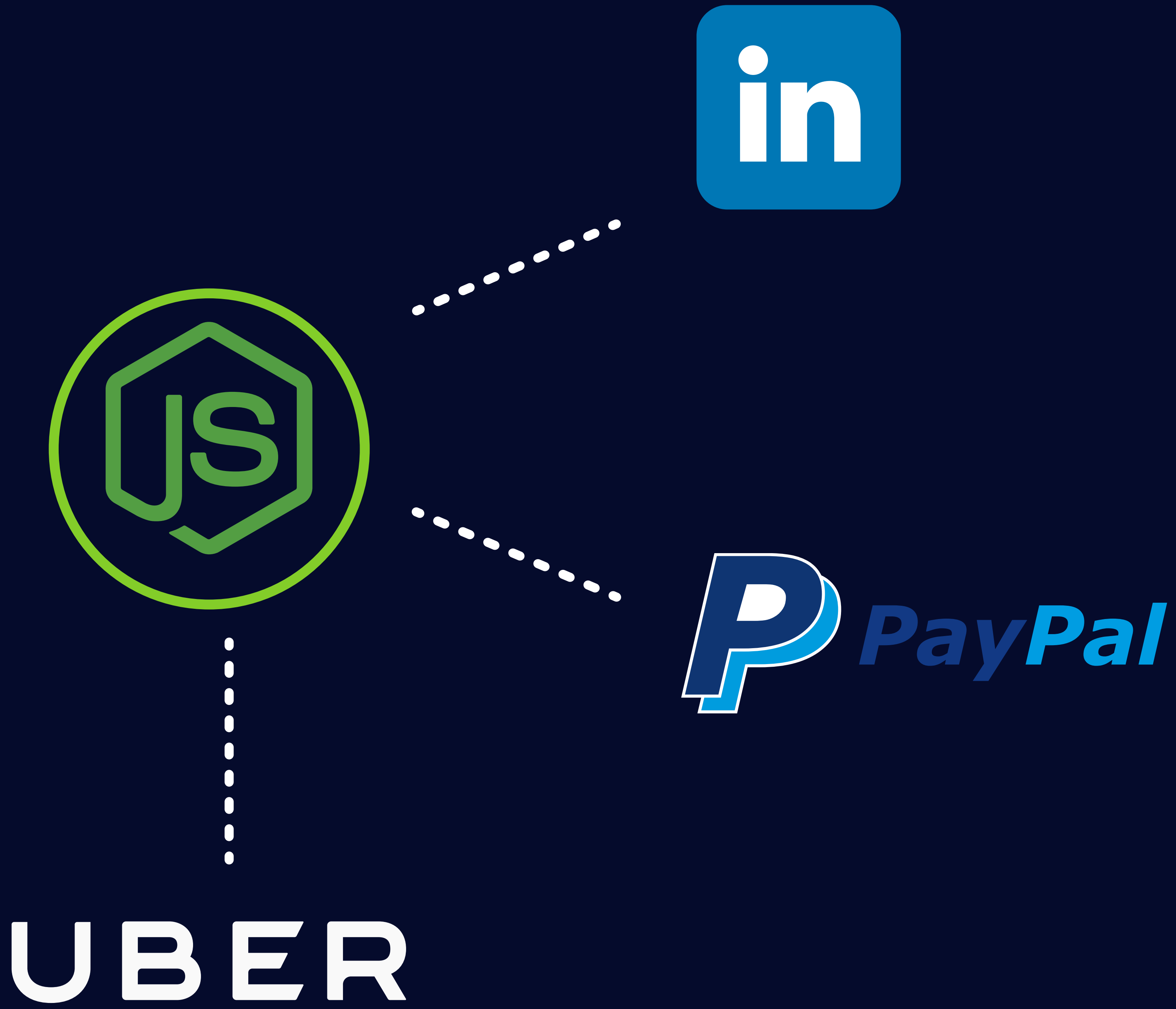
- Developing web applications
- Building chatbots
- Creating real-time applications
- Streaming video and audio
- Building microservices

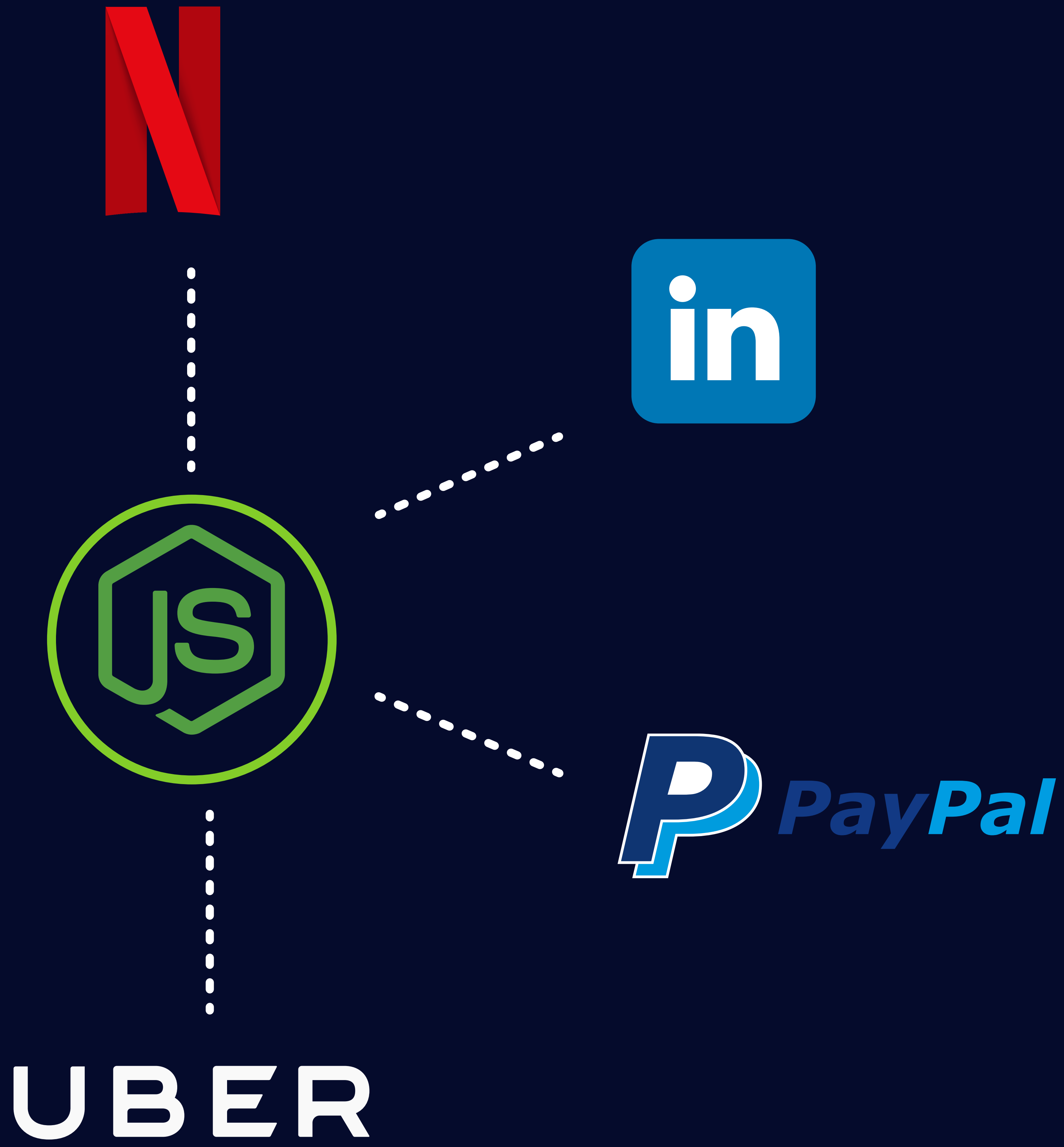
Who uses Node JS?

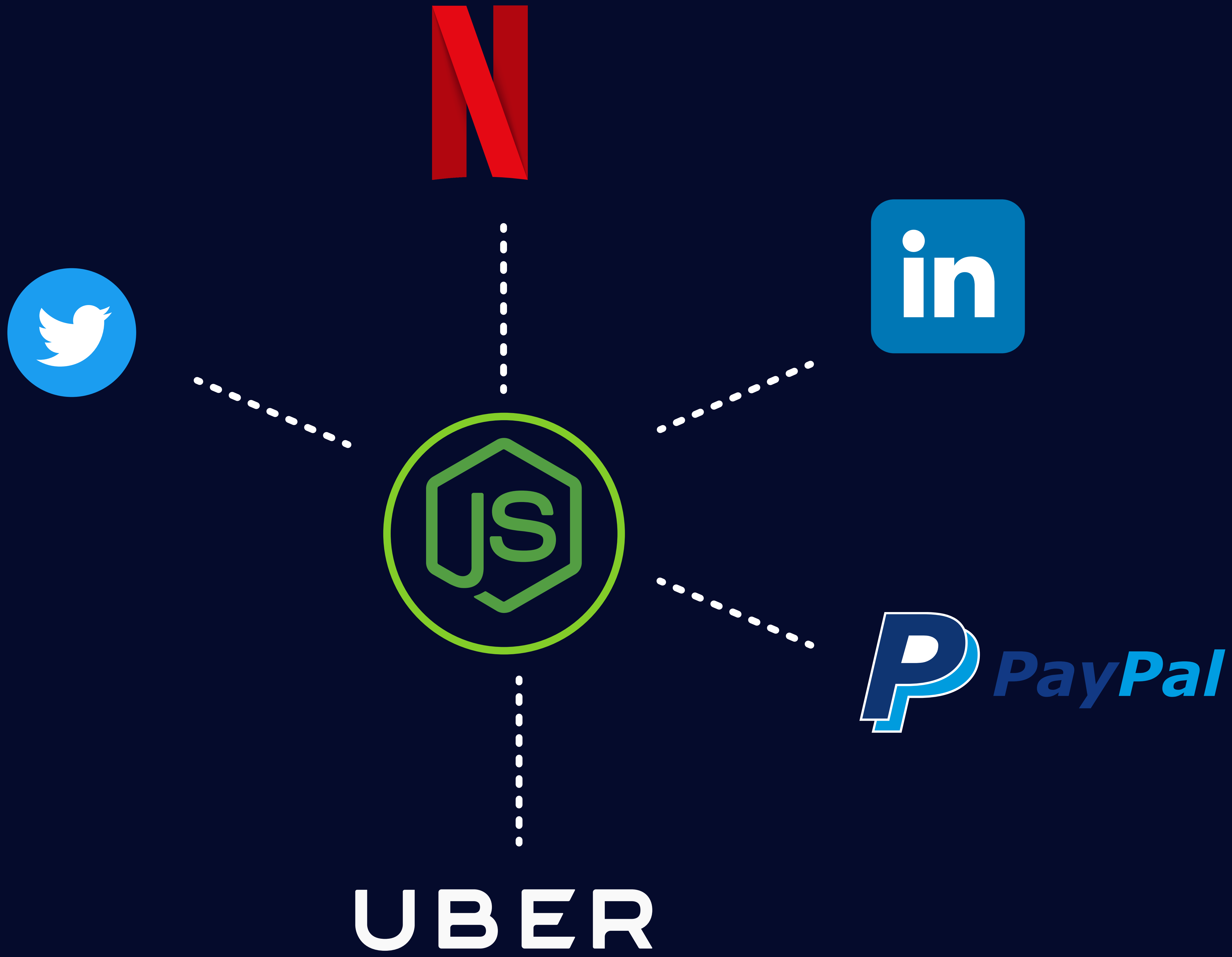
JS

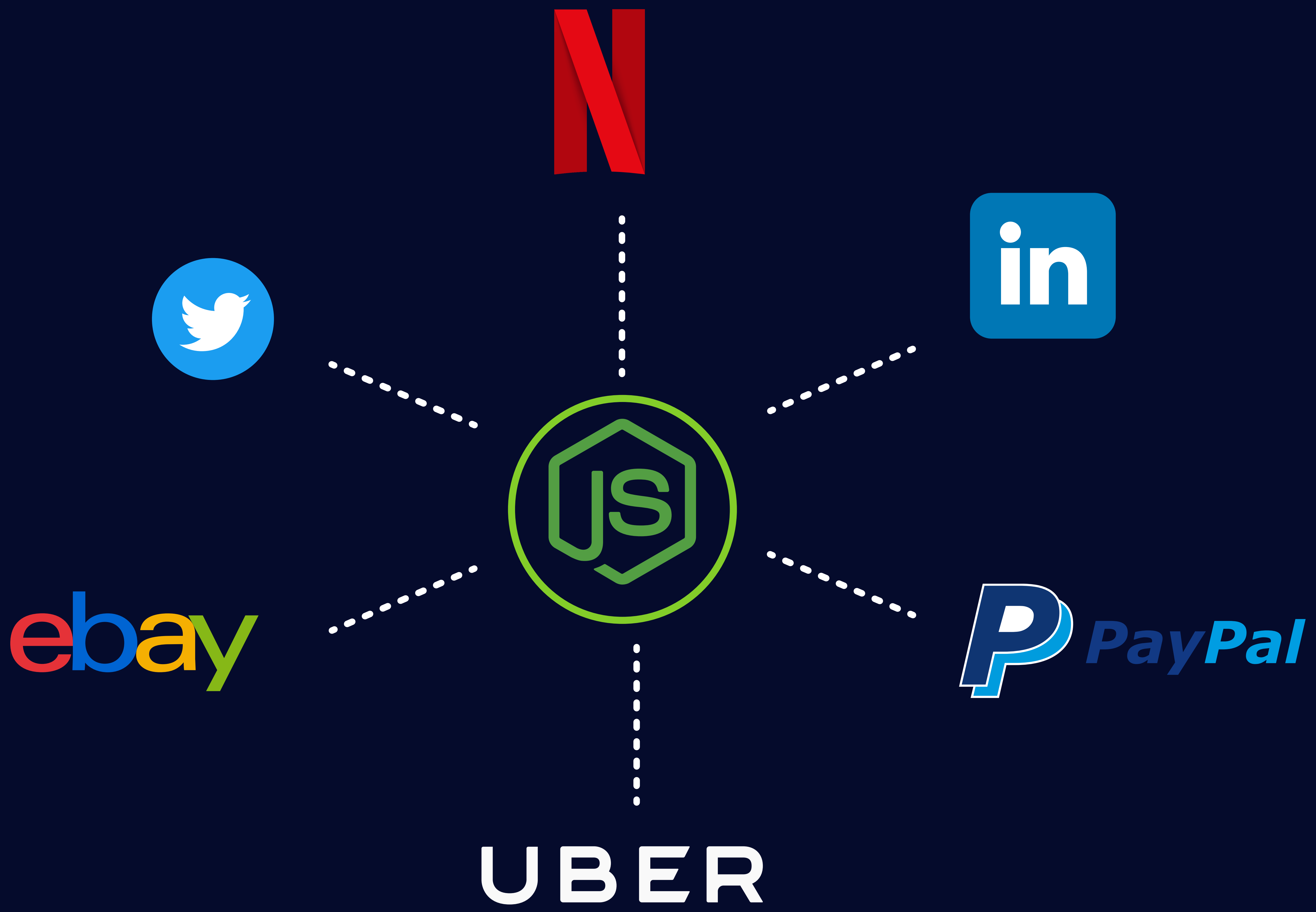
UBER


JS
PayPal
UBER


JS
in
PayPal
UBER


JS
in
PayPal
UBER


in
JS
PayPal
UBER


JS
in
ebay
PayPal
UBER

Install Node JS

REPL

R=Read

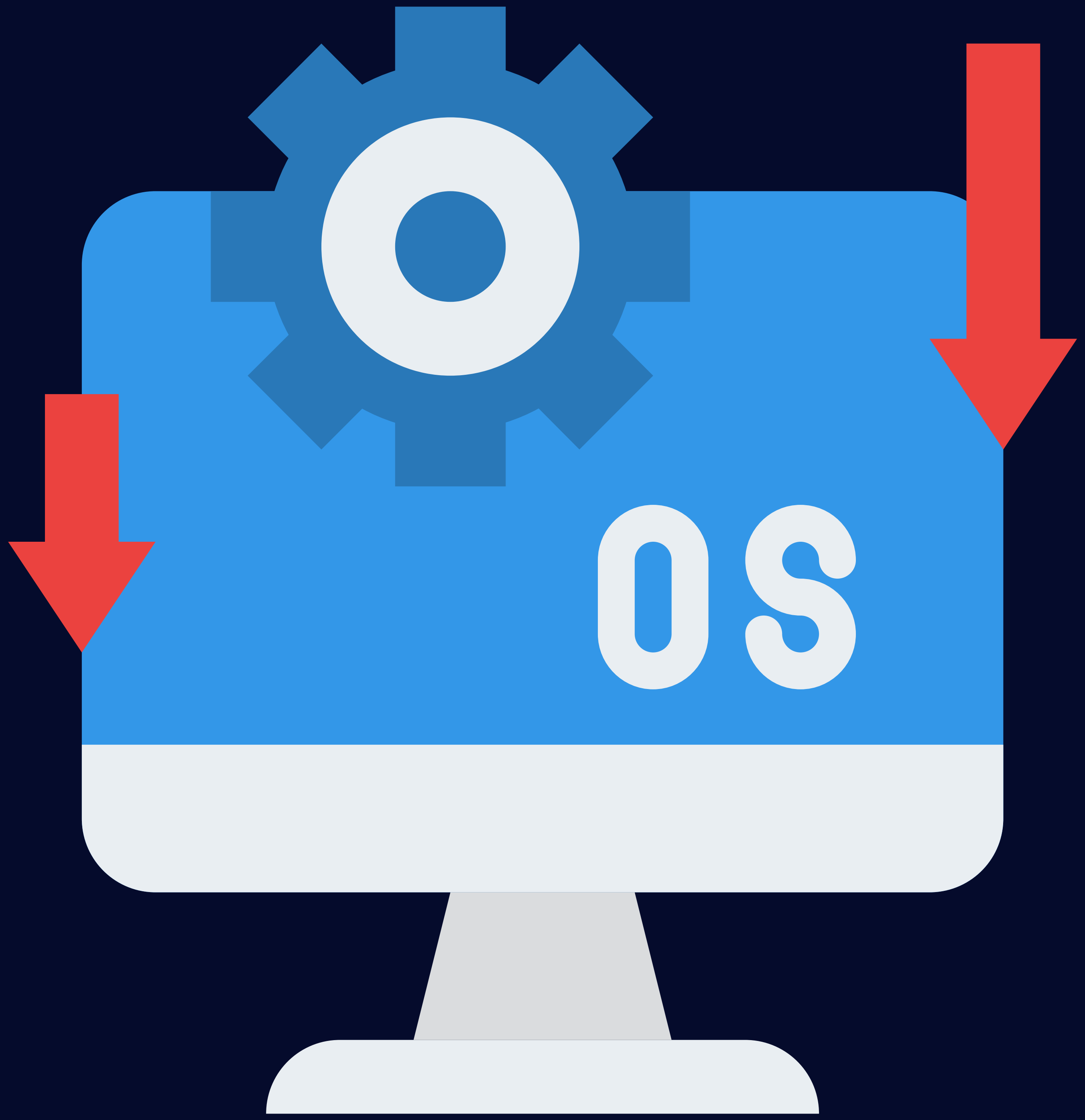
E=Evaluate

P=Print

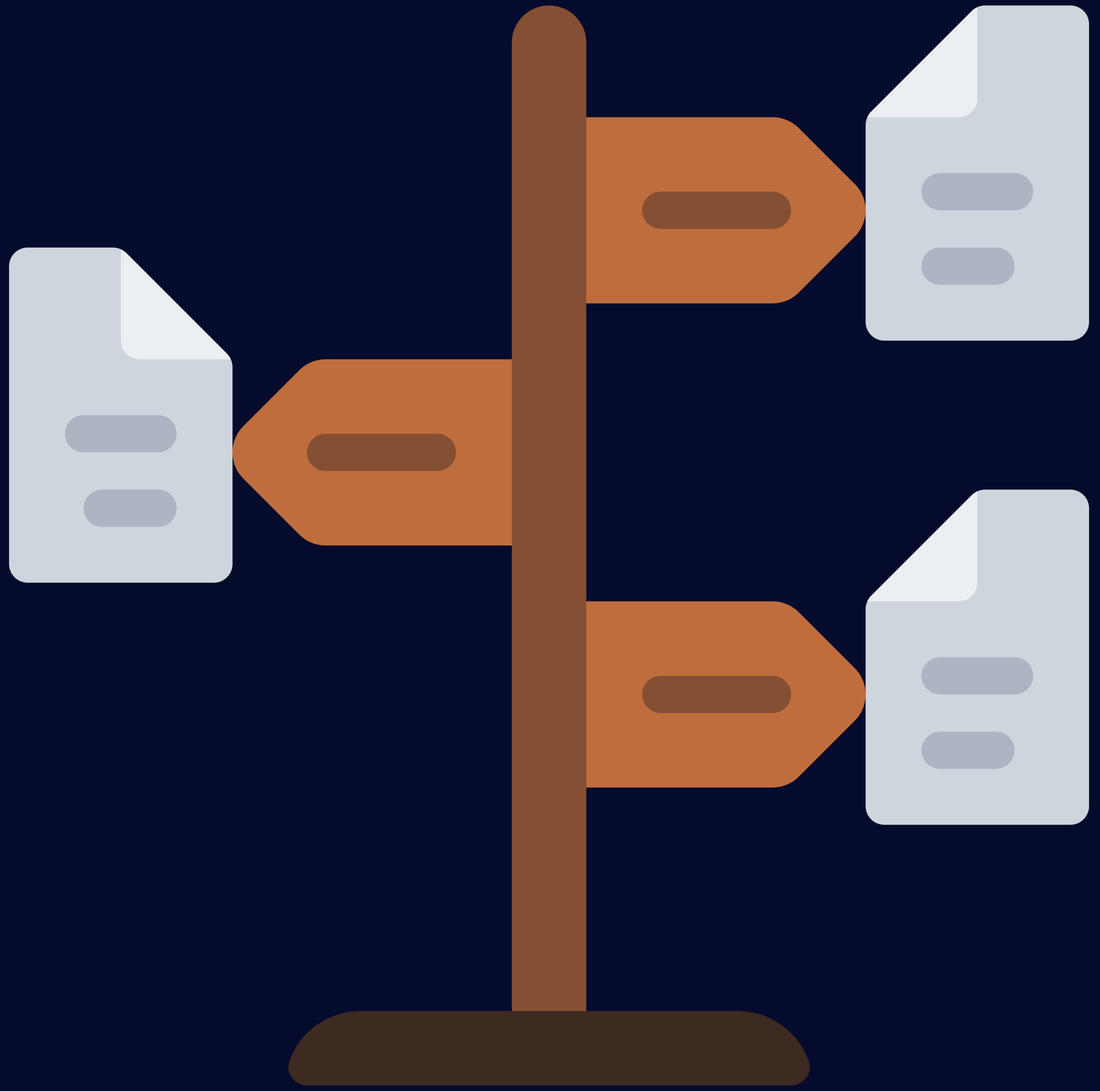
L=Loop

NODE JS MODULES

fs-File System



OS - Operating System



Path - Module