



what is
backend
Development?

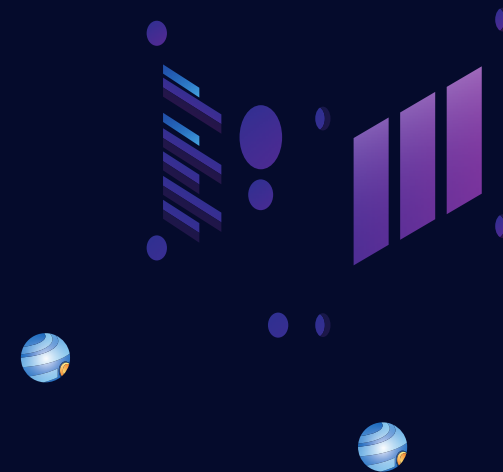


Components of a web application

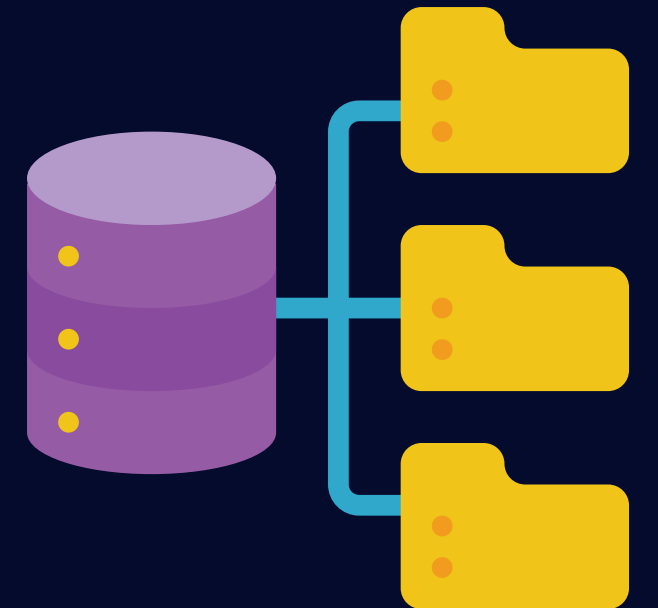
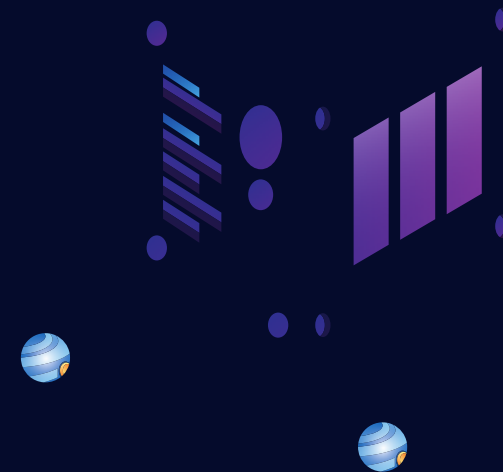


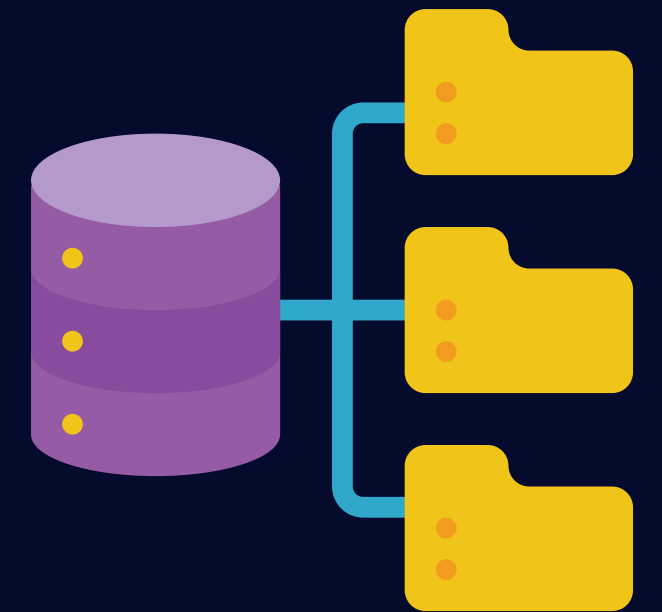


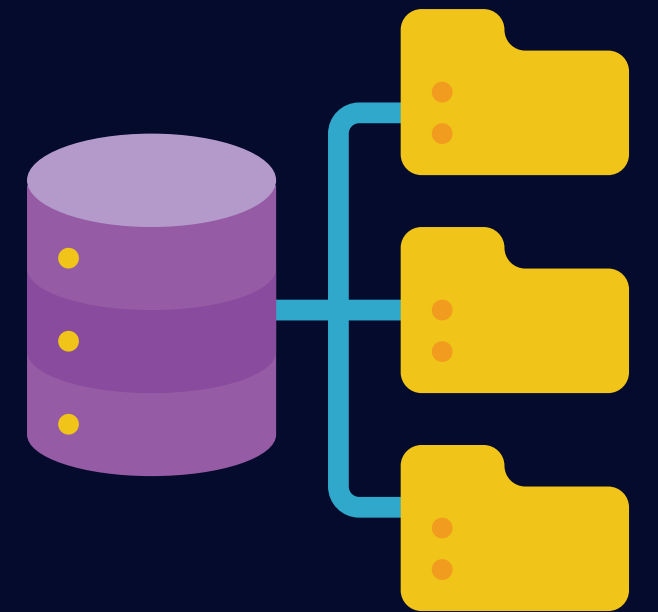


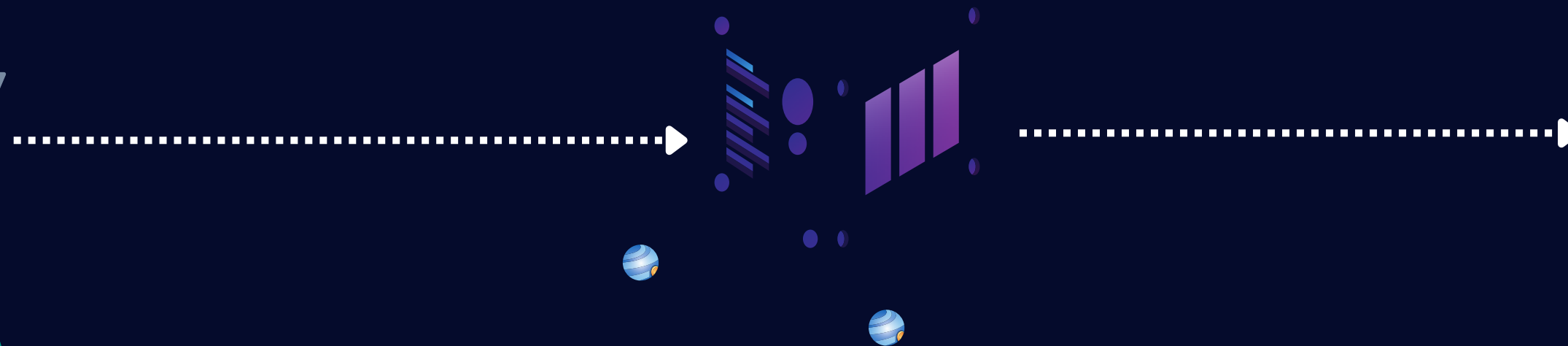






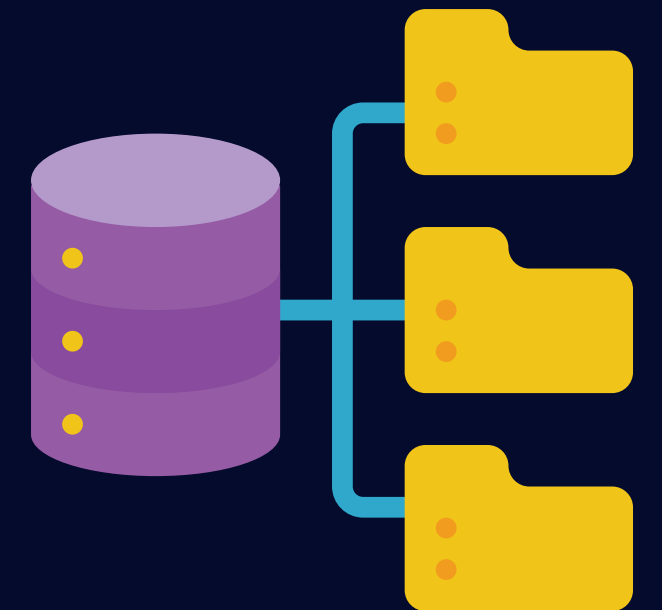
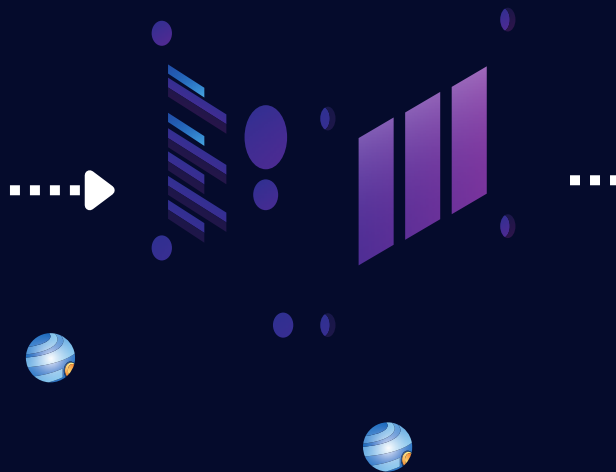


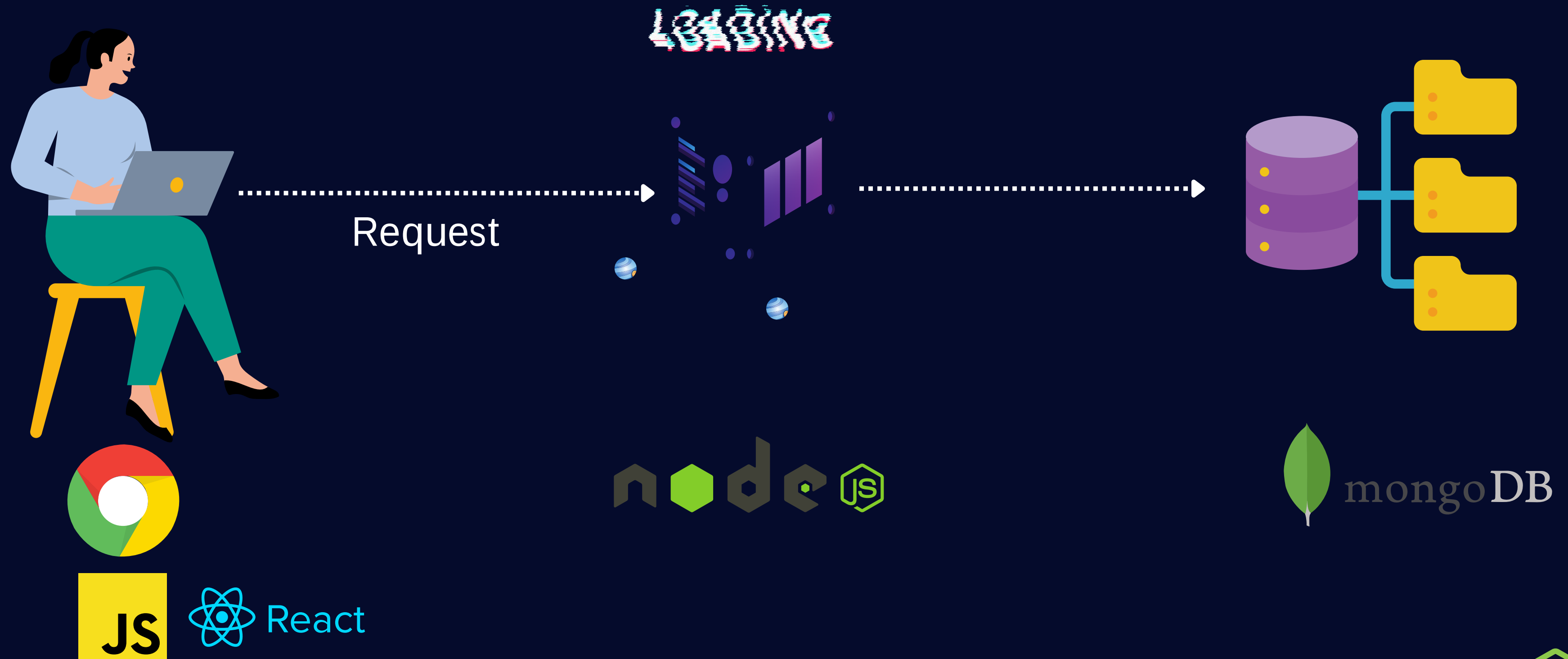


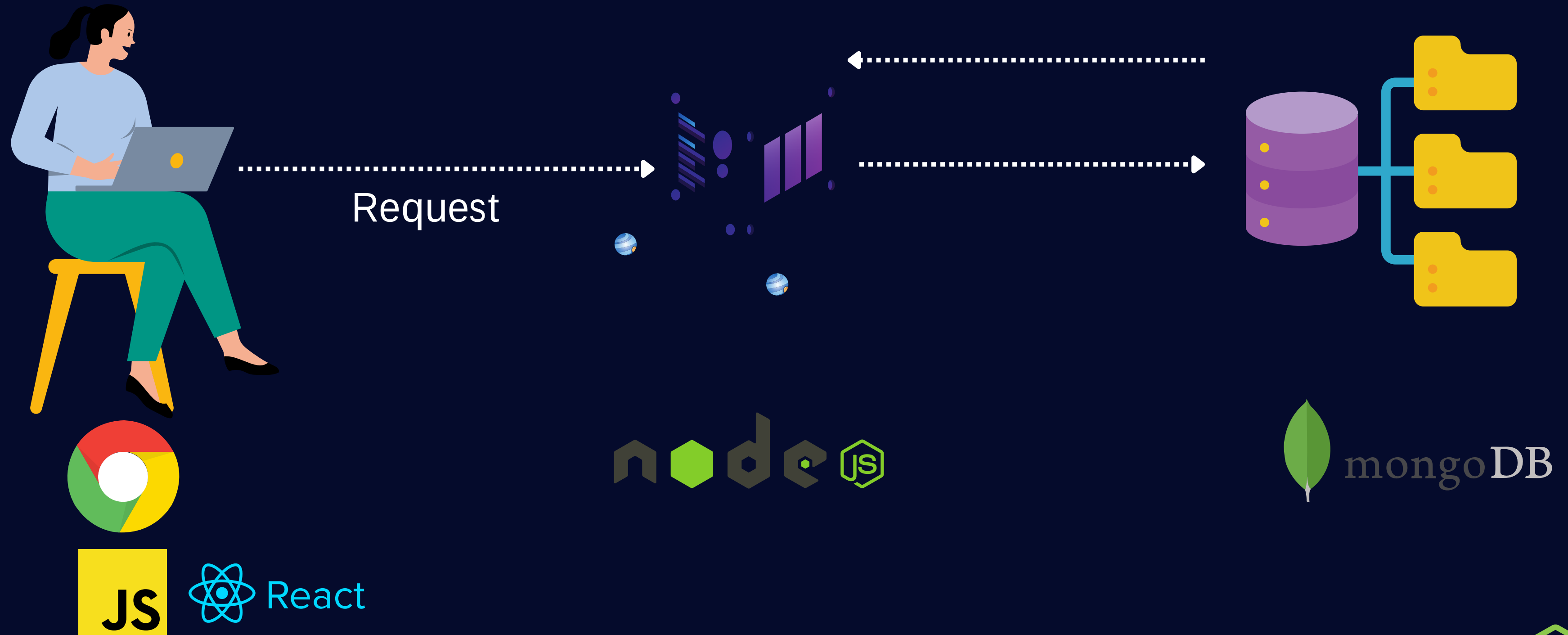


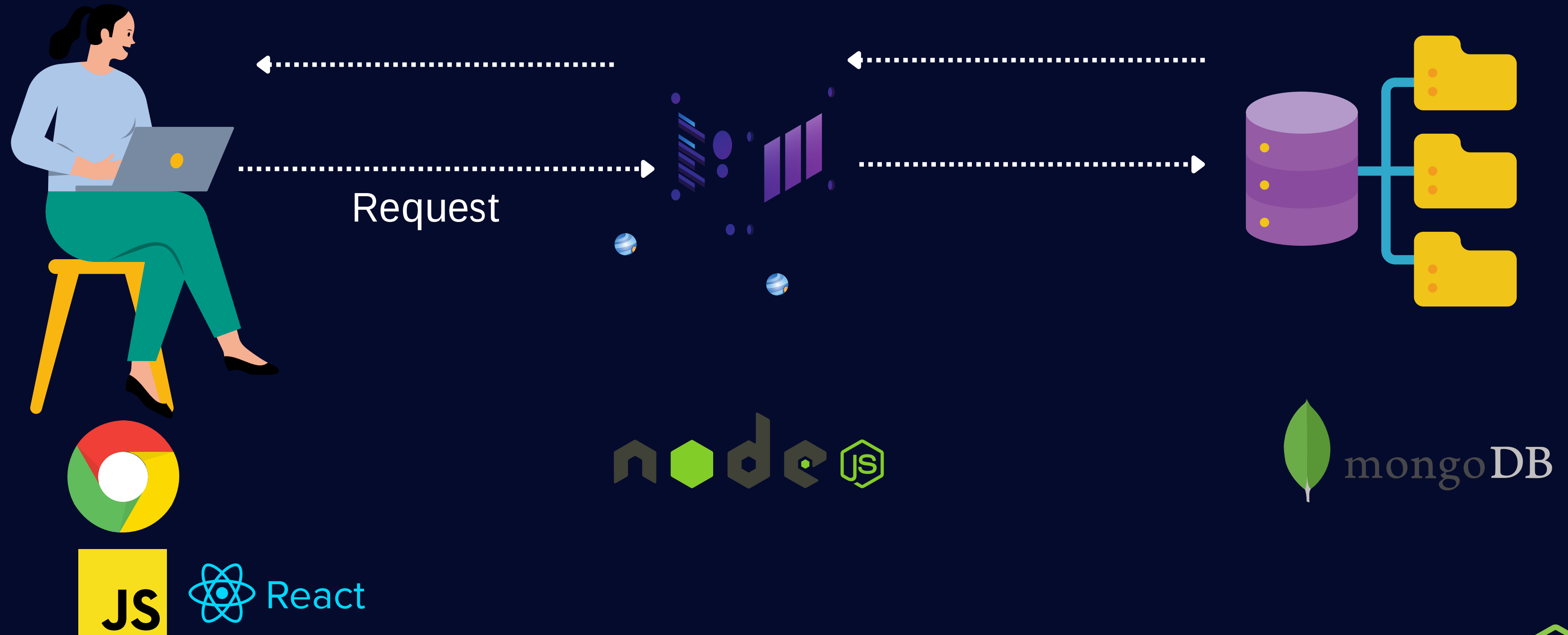


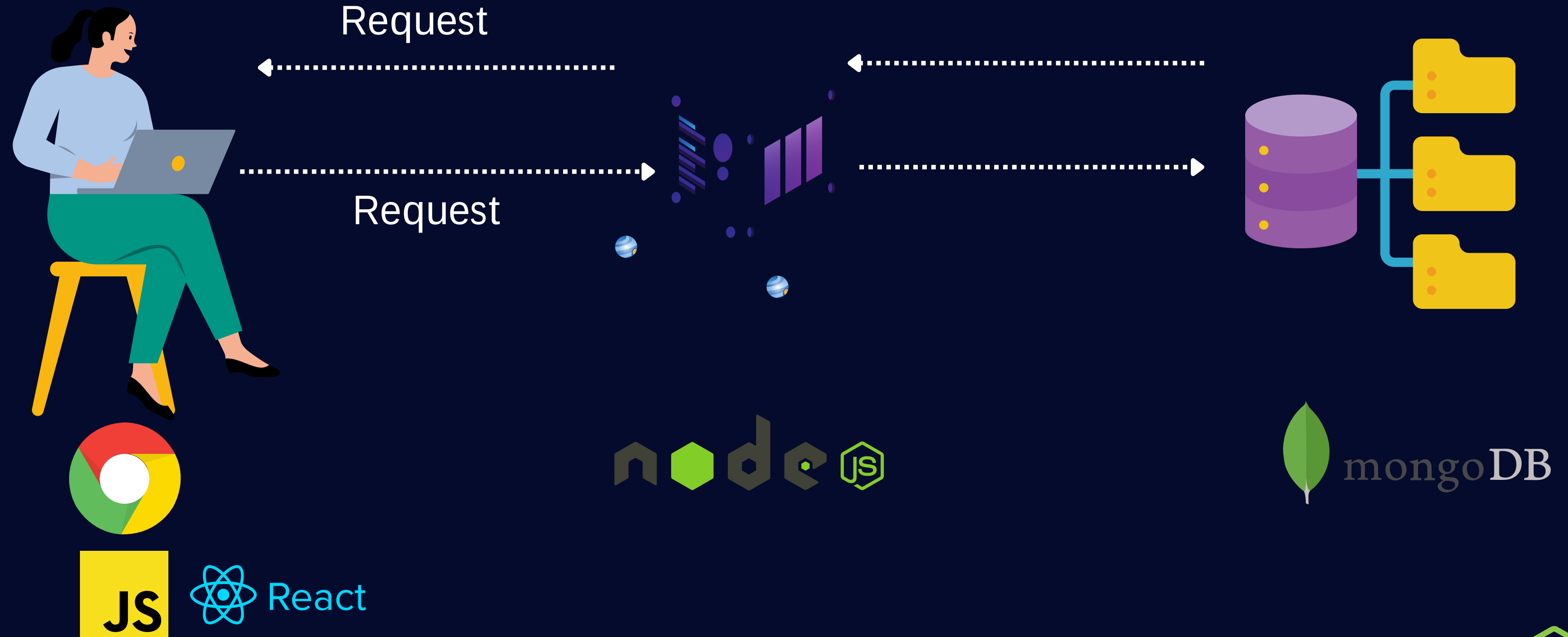
Request



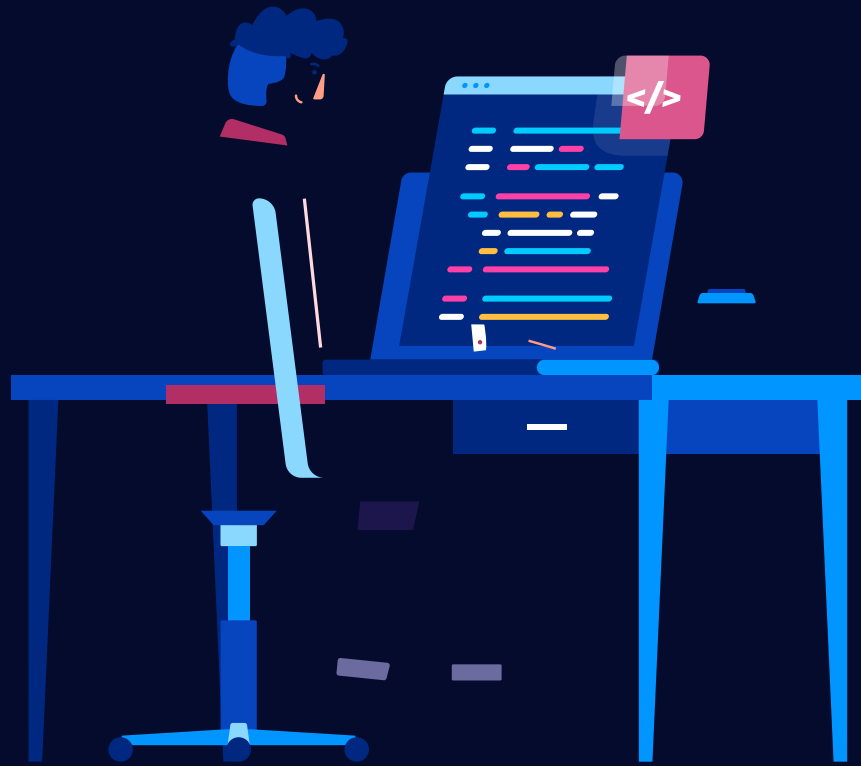








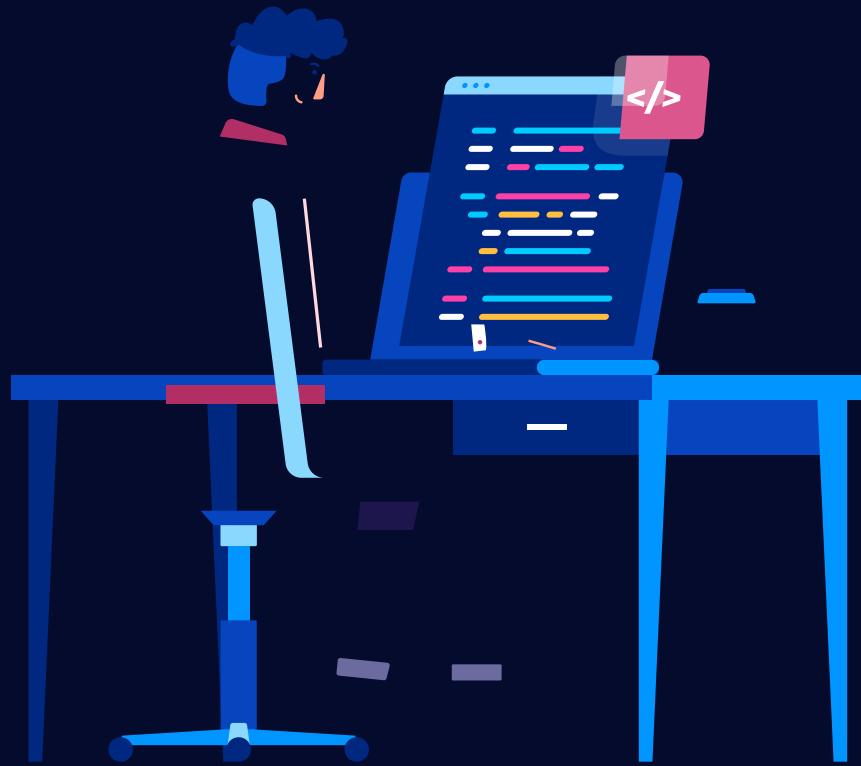




backend **lopment**



what is
backend
Development?

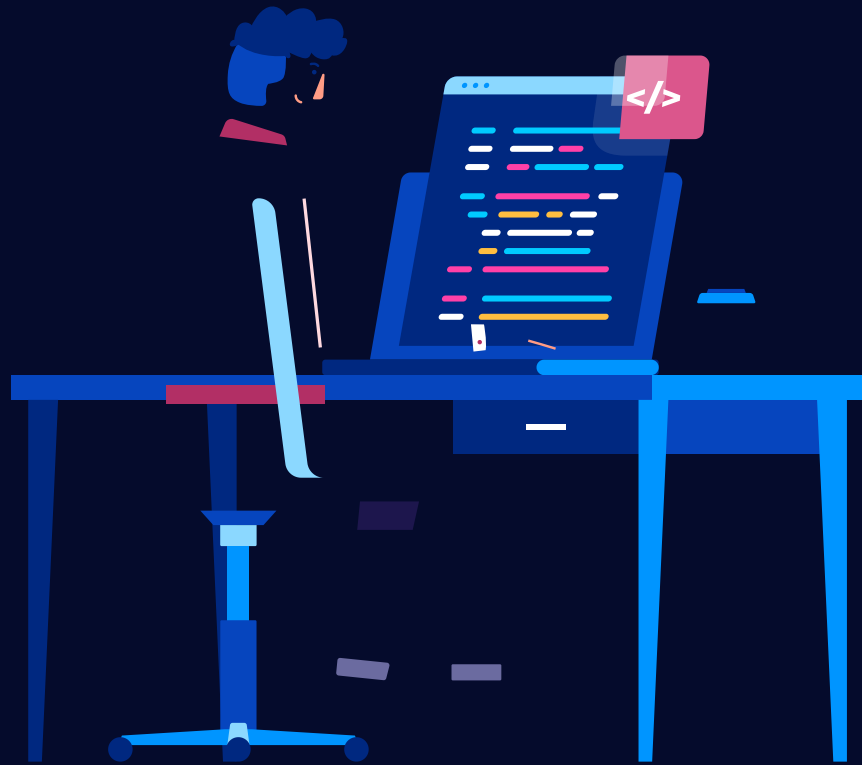


backend development

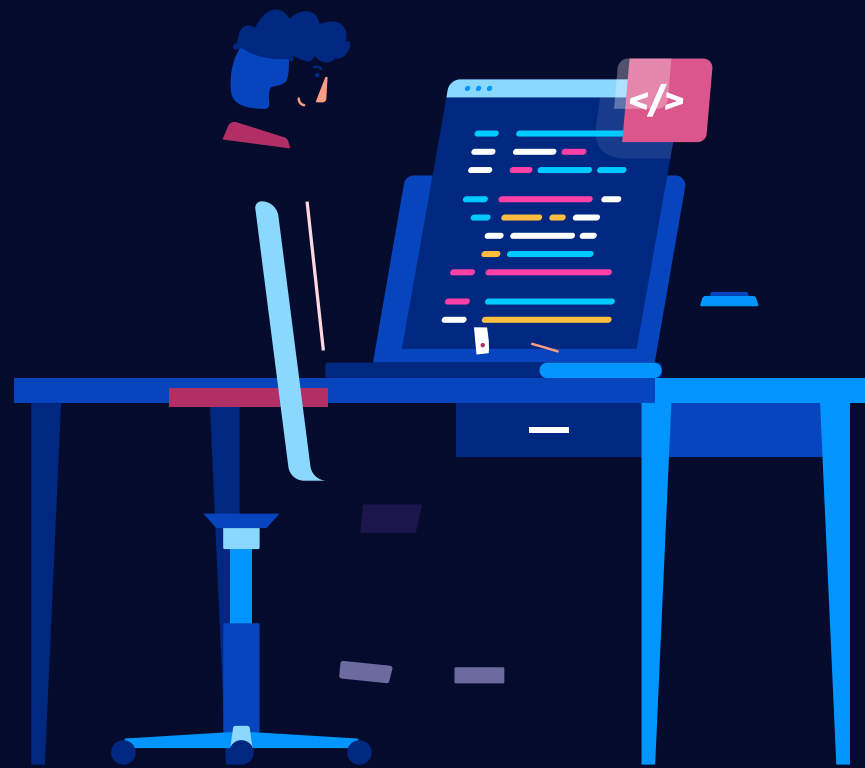


A backend developer is a software engineer who specializes in the development of server-side applications

backend Developer Roles

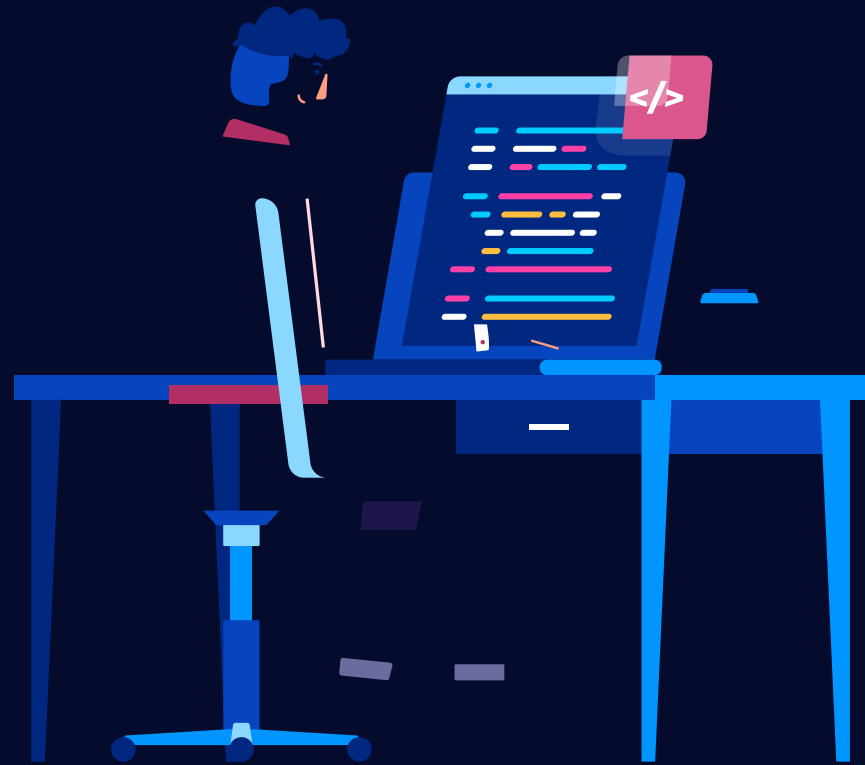


backend Developer Roles



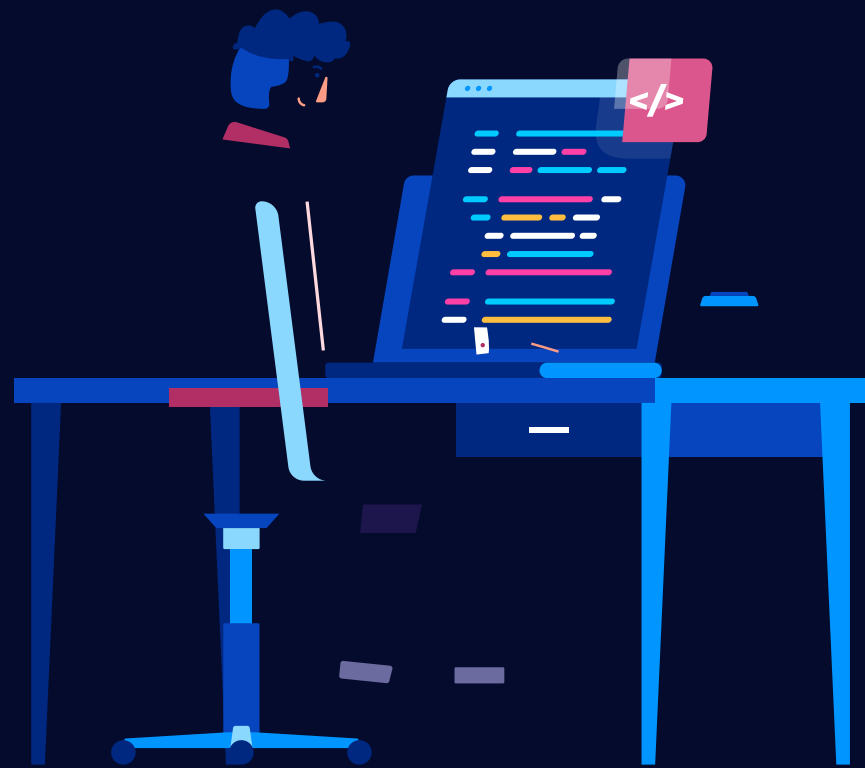
- Implement business logic of an application

backend Developer Roles



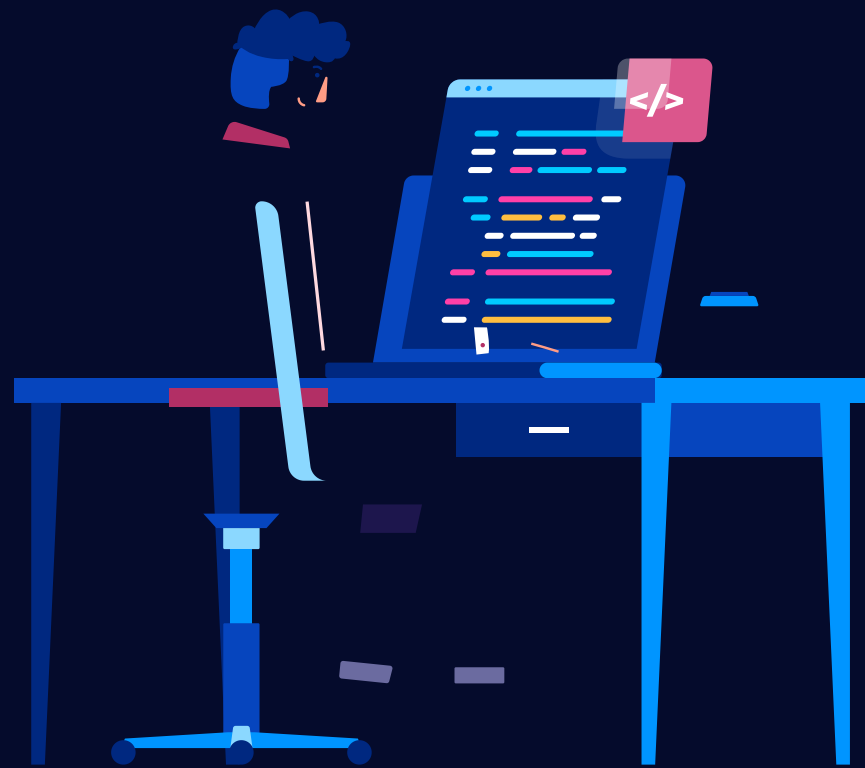
- ▶ Implement business logic of an application
- ▶ Act as the server-side counterpart to front-end web developers

backend Developer Roles



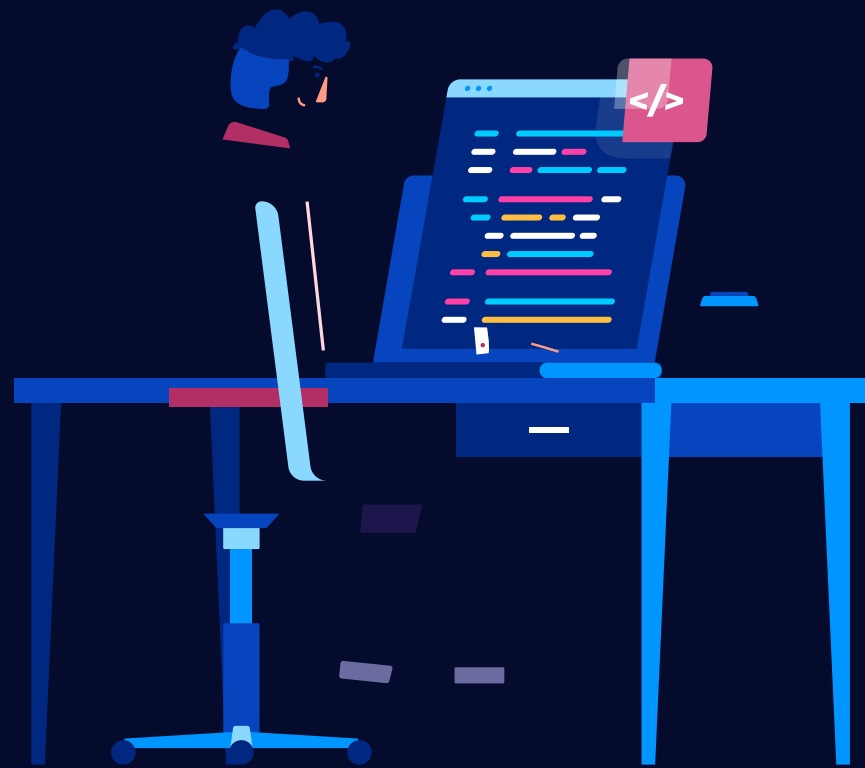
- ▶ Implement business logic of an application
- ▶ Act as the server-side counterpart to front-end web developers
- ▶ Write, test and debug server-side code

backend Developer Roles



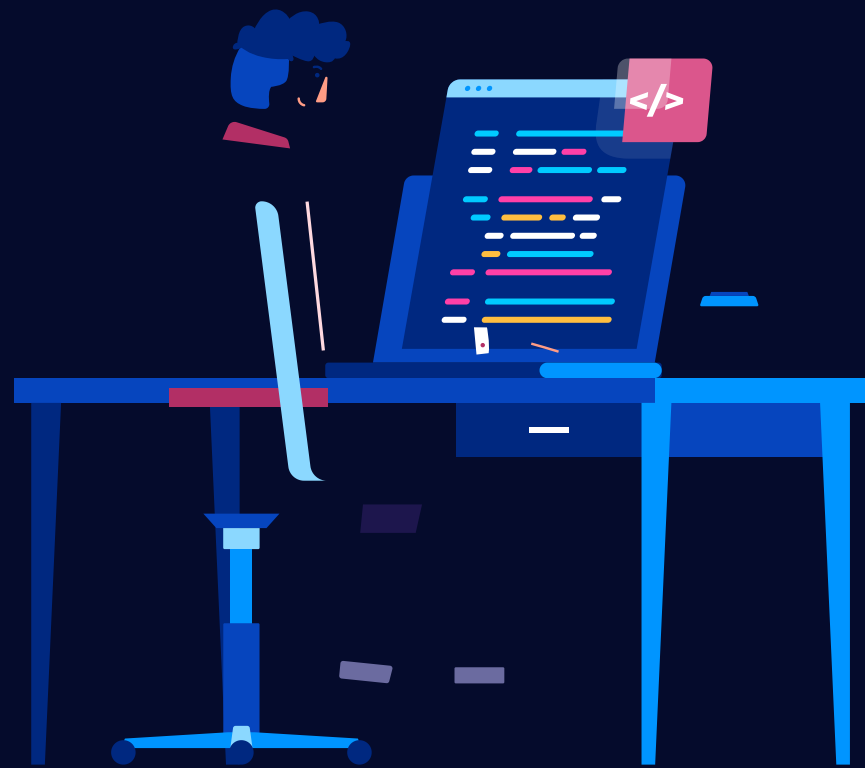
- ▶ Implement business logic of an application
- ▶ Act as the server-side counterpart to front-end web developers
- ▶ Write, test and debug server-side code
- ▶ Design, develop and implement web-based applications

backend Developer Roles



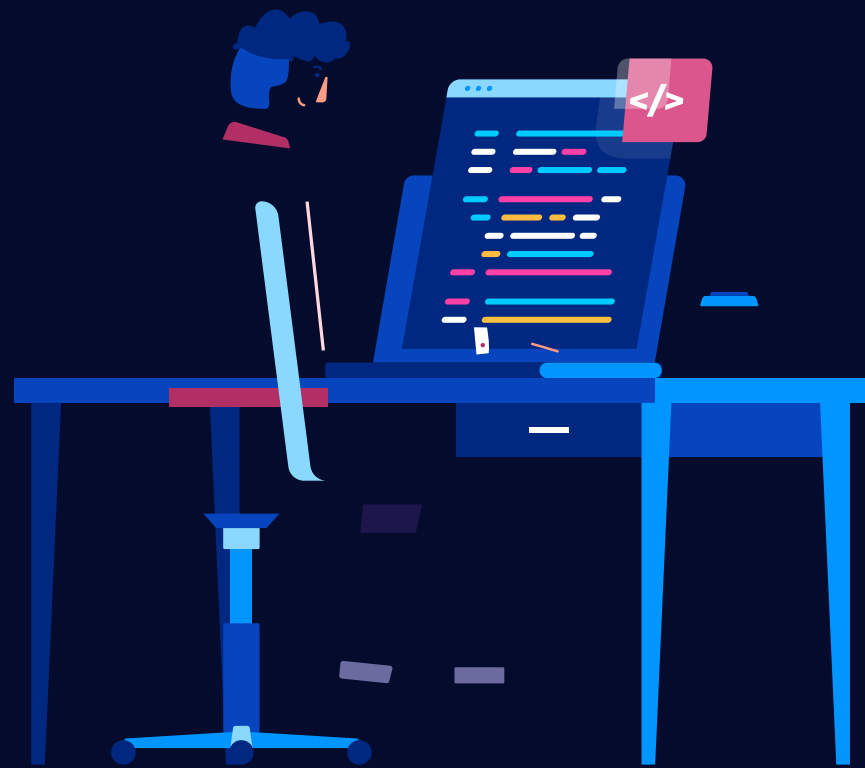
- ▶ Implement business logic of an application
- ▶ Act as the server-side counterpart to front-end web developers
- ▶ Write, test and debug server-side code
- ▶ Design, develop and implement web-based applications
- ▶ Develop and maintain server-side APIs

backend Developer Roles



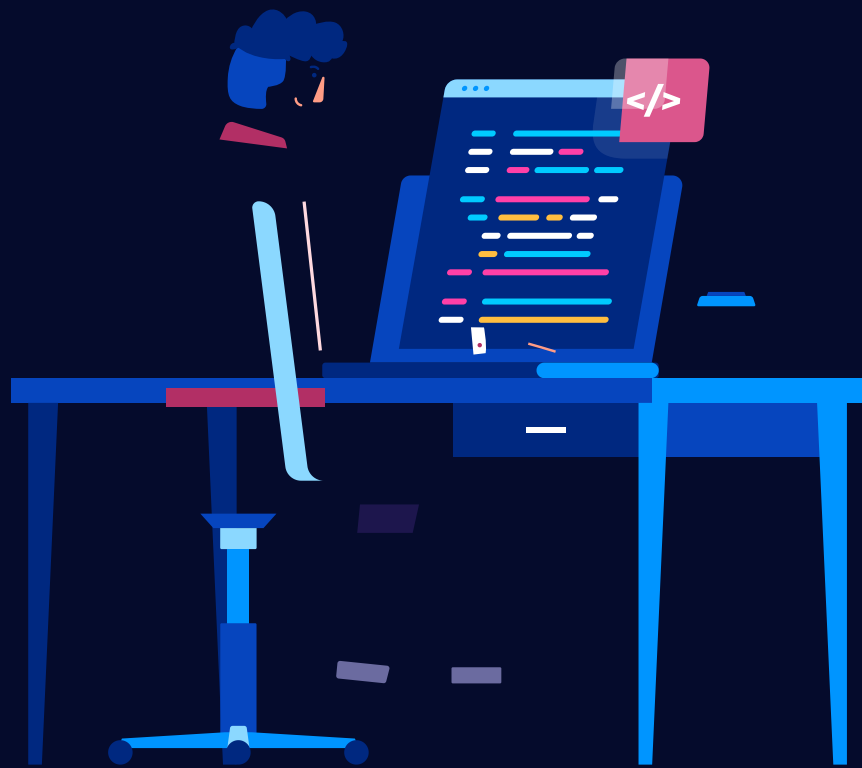
- ▶ Implement business logic of an application
- ▶ Act as the server-side counterpart to front-end web developers
- ▶ Write, test and debug server-side code
- ▶ Design, develop and implement web-based applications
- ▶ Develop and maintain server-side APIs
- ▶ Manage server-side code libraries

backend Developer Roles

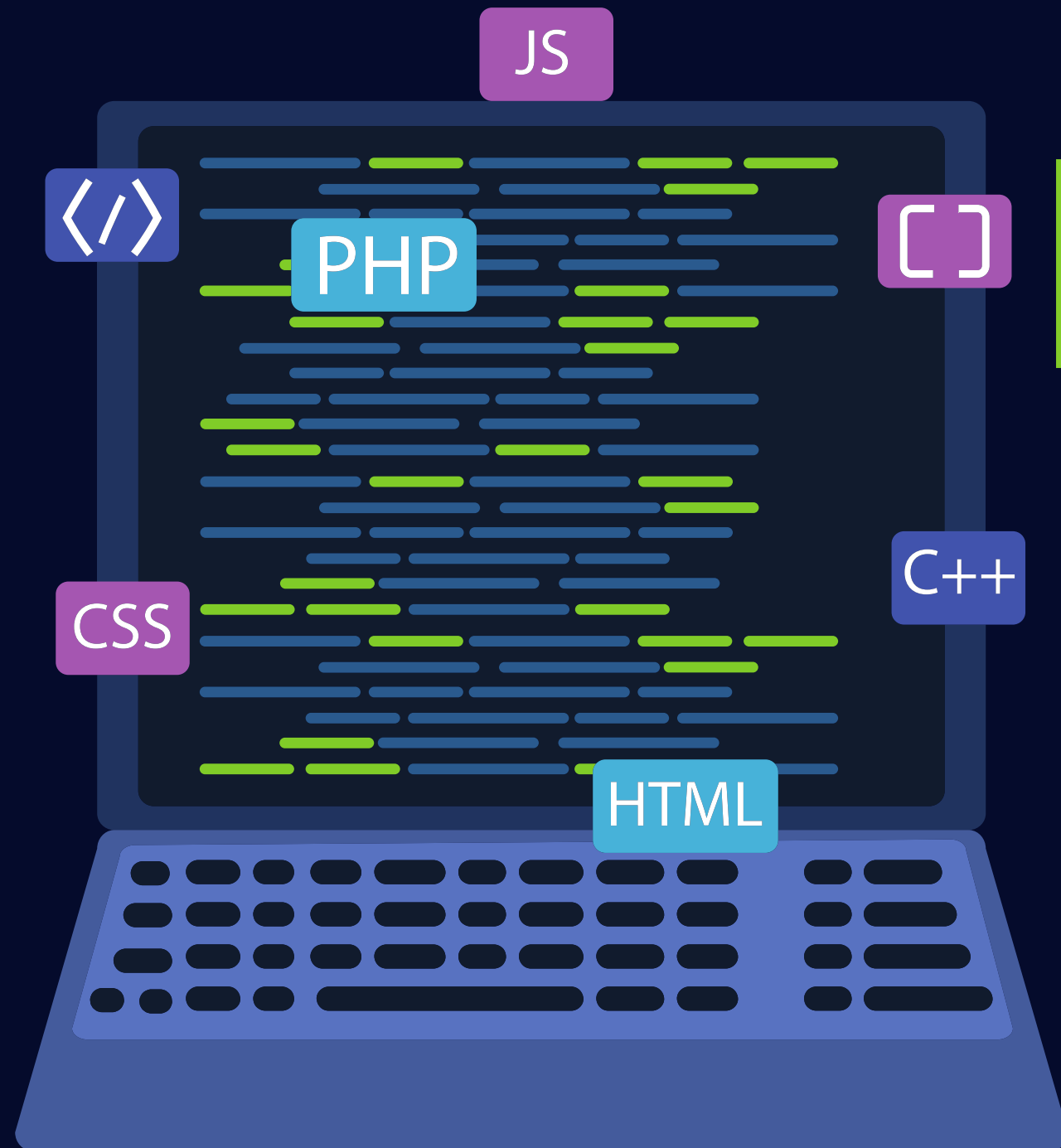


- ▶ Implement business logic of an application
- ▶ Act as the server-side counterpart to front-end web developers
- ▶ Write, test and debug server-side code
- ▶ Design, develop and implement web-based applications
- ▶ Develop and maintain server-side APIs
- ▶ Manage server-side code libraries
- ▶ Optimize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

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

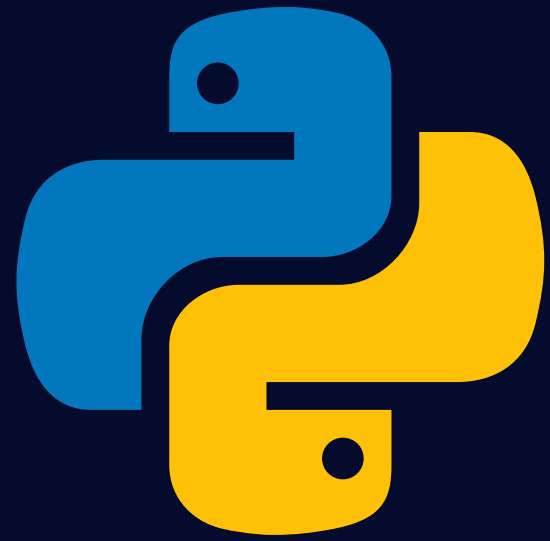


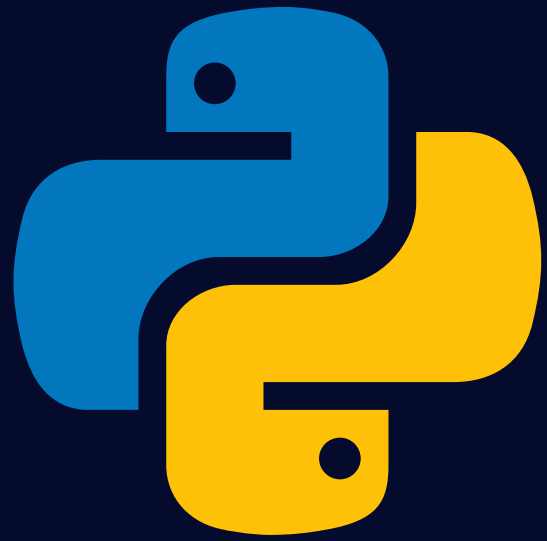
Programming Languages

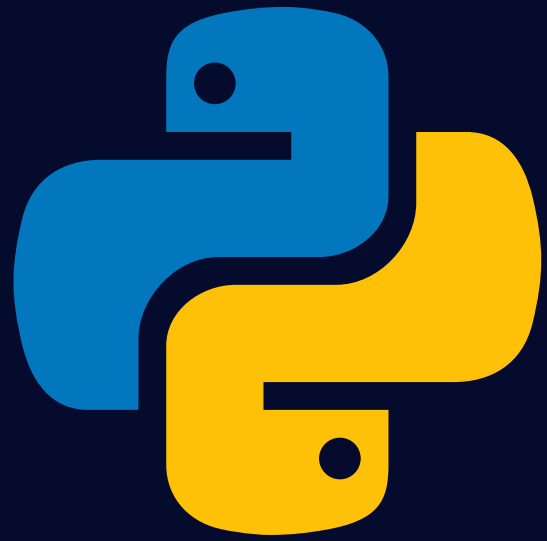


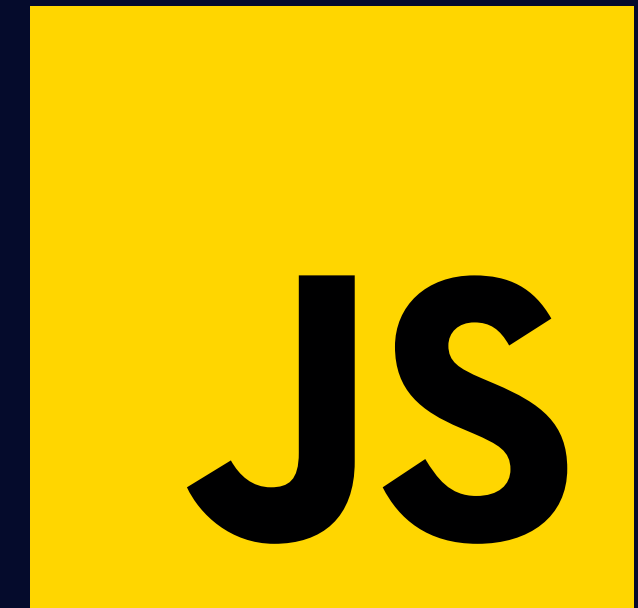
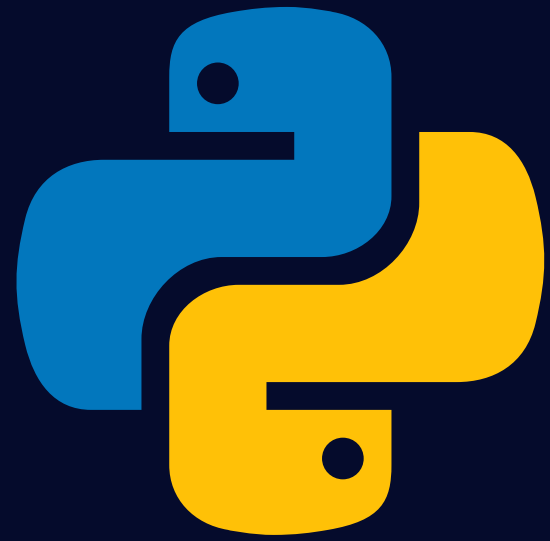
NODE JS

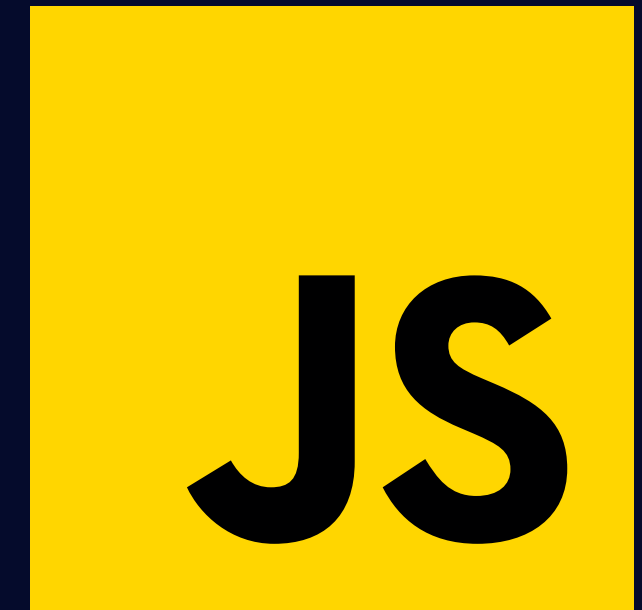
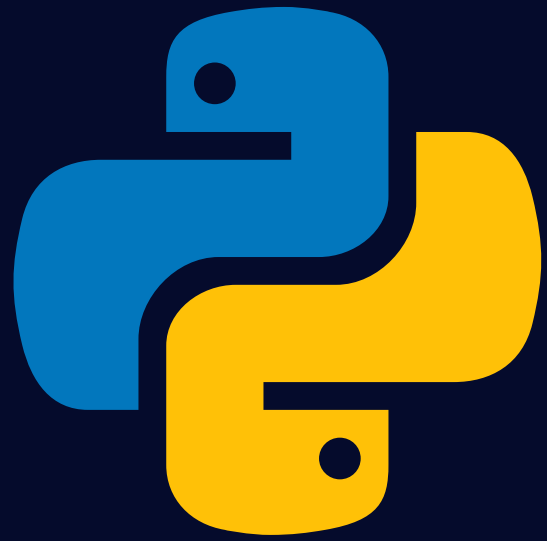














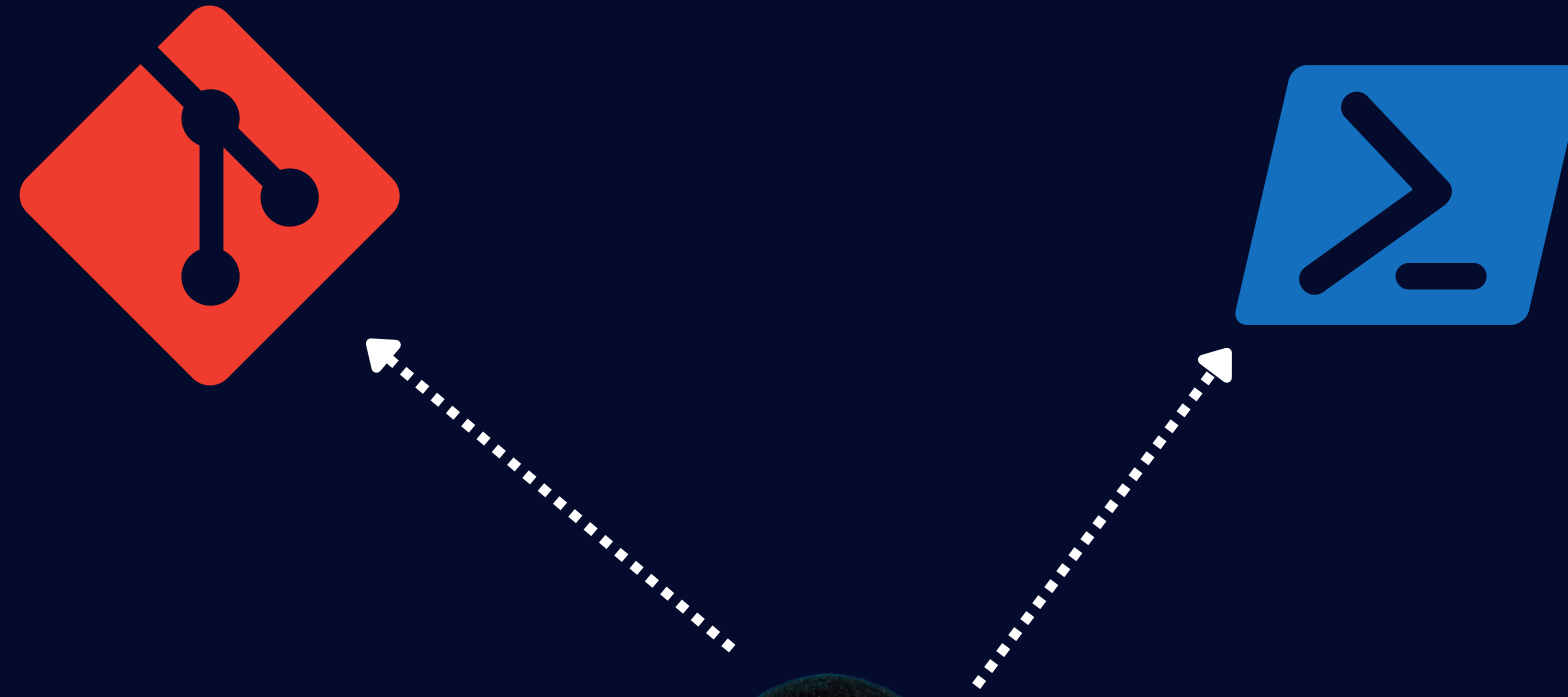
NODE JS

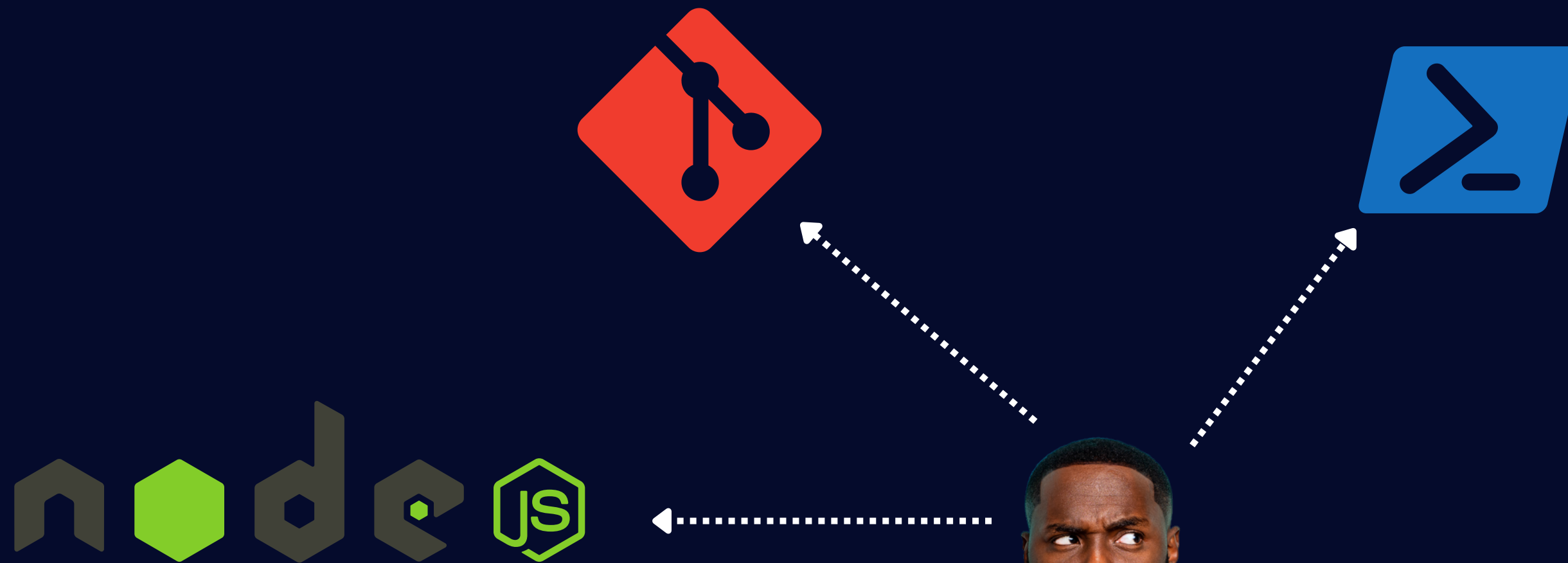


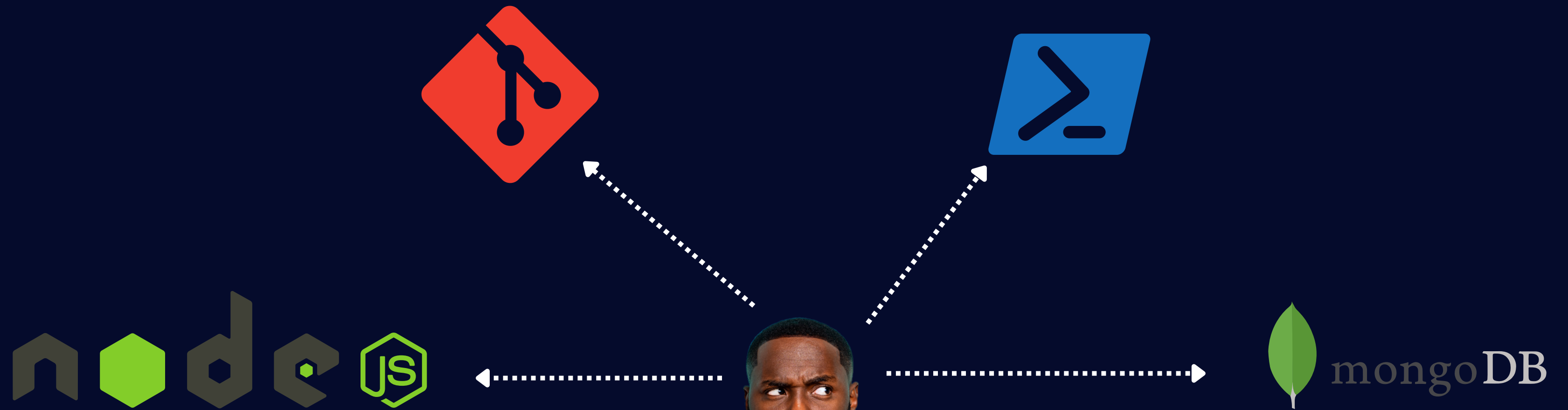
What You need to know















Confusing Terminologies





👉 Terminal



☞ Terminal

☞ Shell



☞ Terminal

☞ Shell

☞ Command Line





Terminal



Terminal

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



Terminal

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

Shell



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.



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



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)



Login functionality



GIT (Version Control System)



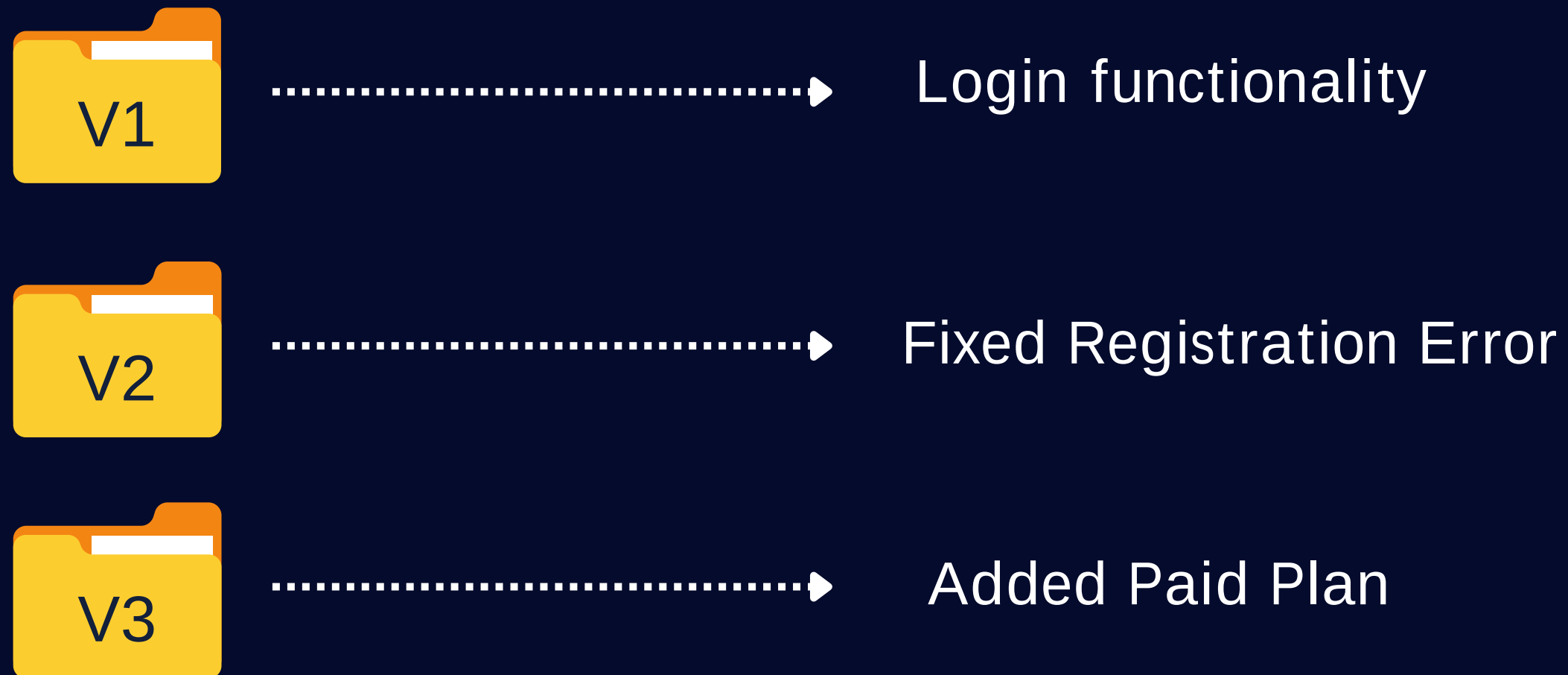
Login functionality



Fixed Registration Error

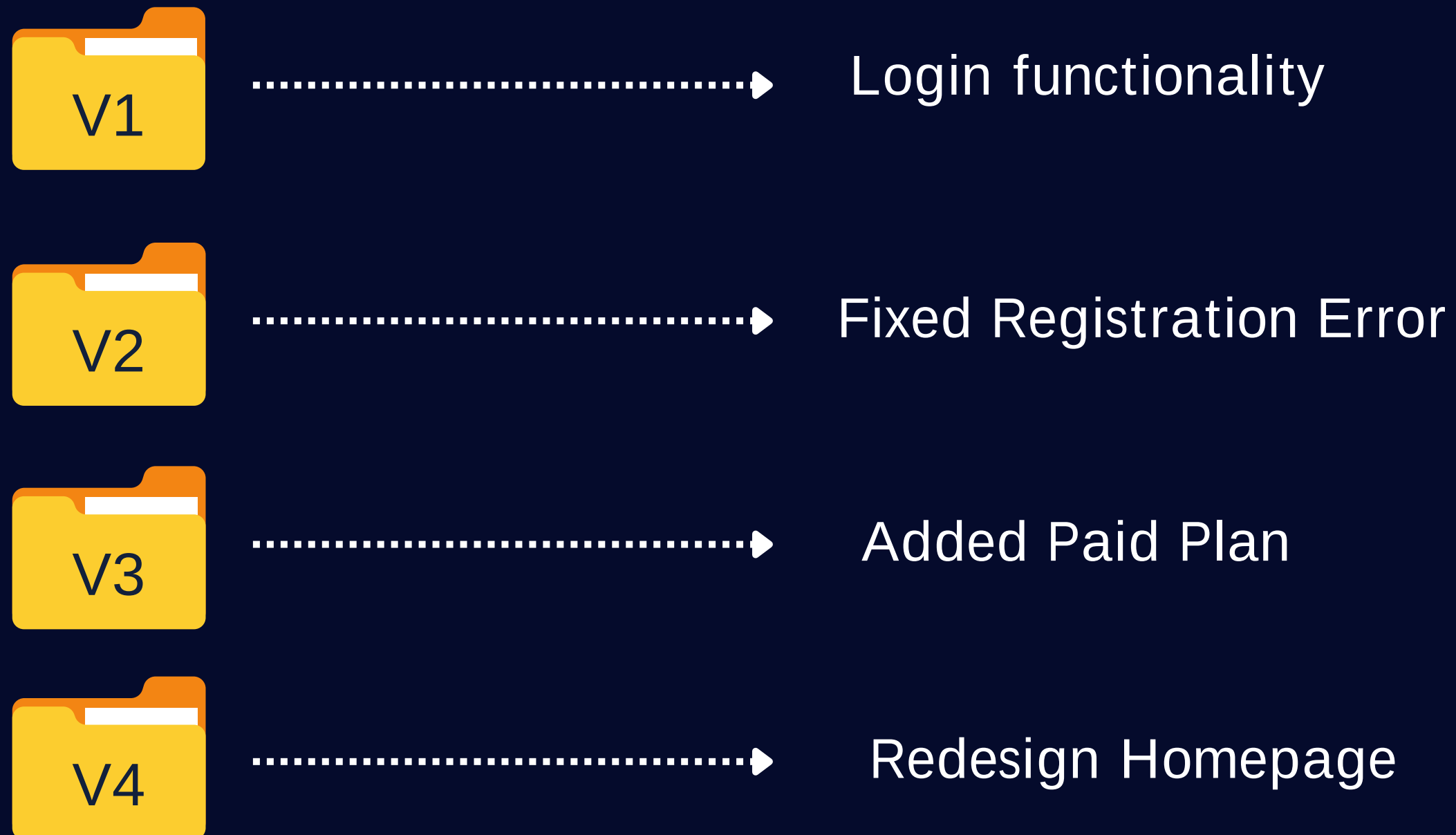


GIT (Version Control System)





GIT (Version Control System)





COMMON COMMANDLINE COMMANDS



► **pwd** → Print Working Directory



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





- ▶ **pwd** → Print Working Directory
- ▶ **mkdir** → Makes a new directory
- ▶ **touch** → Create a file



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



- ▶ **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



- ▶ **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



- ▶ **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



- ▶ **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





Node JS

What is Node JS



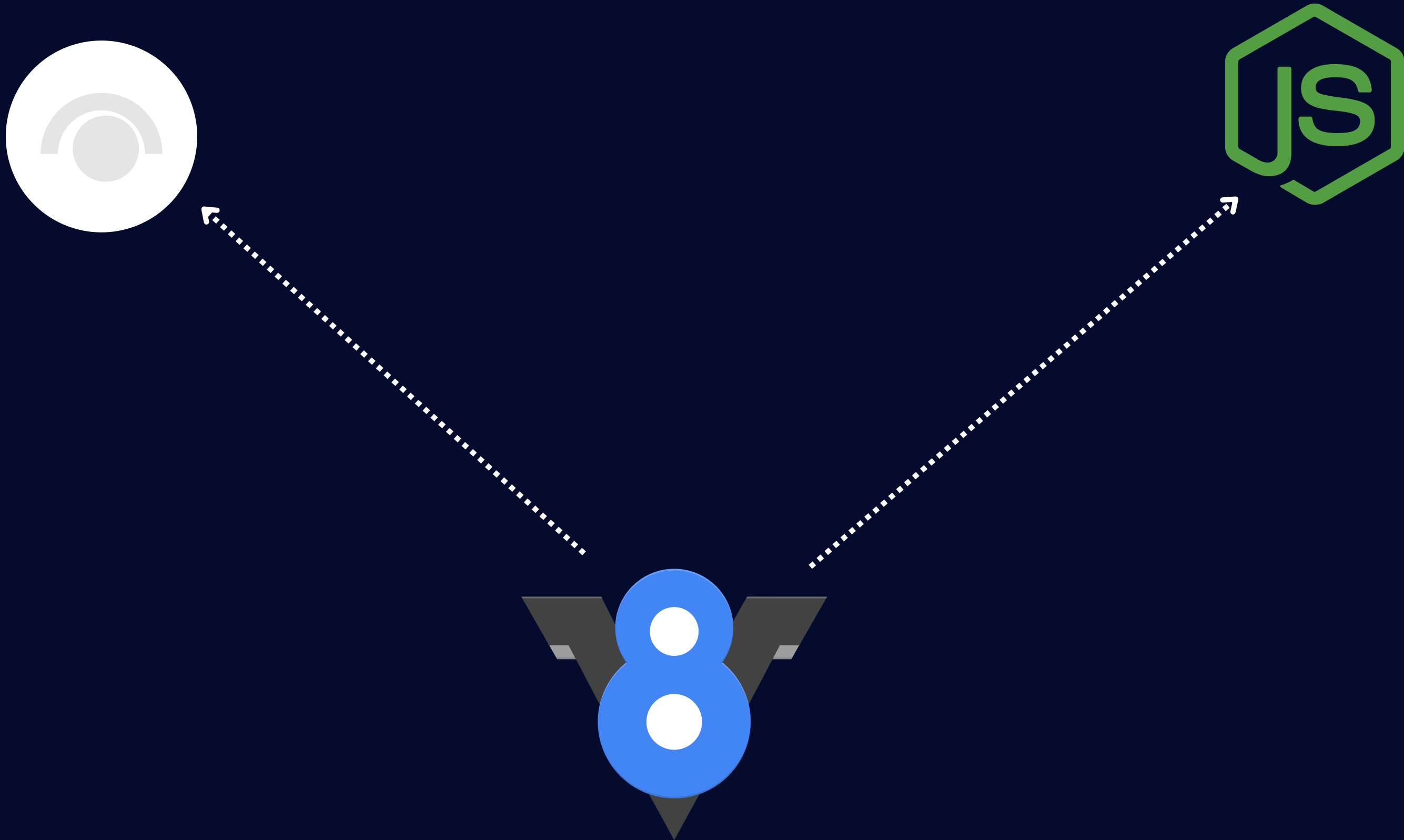


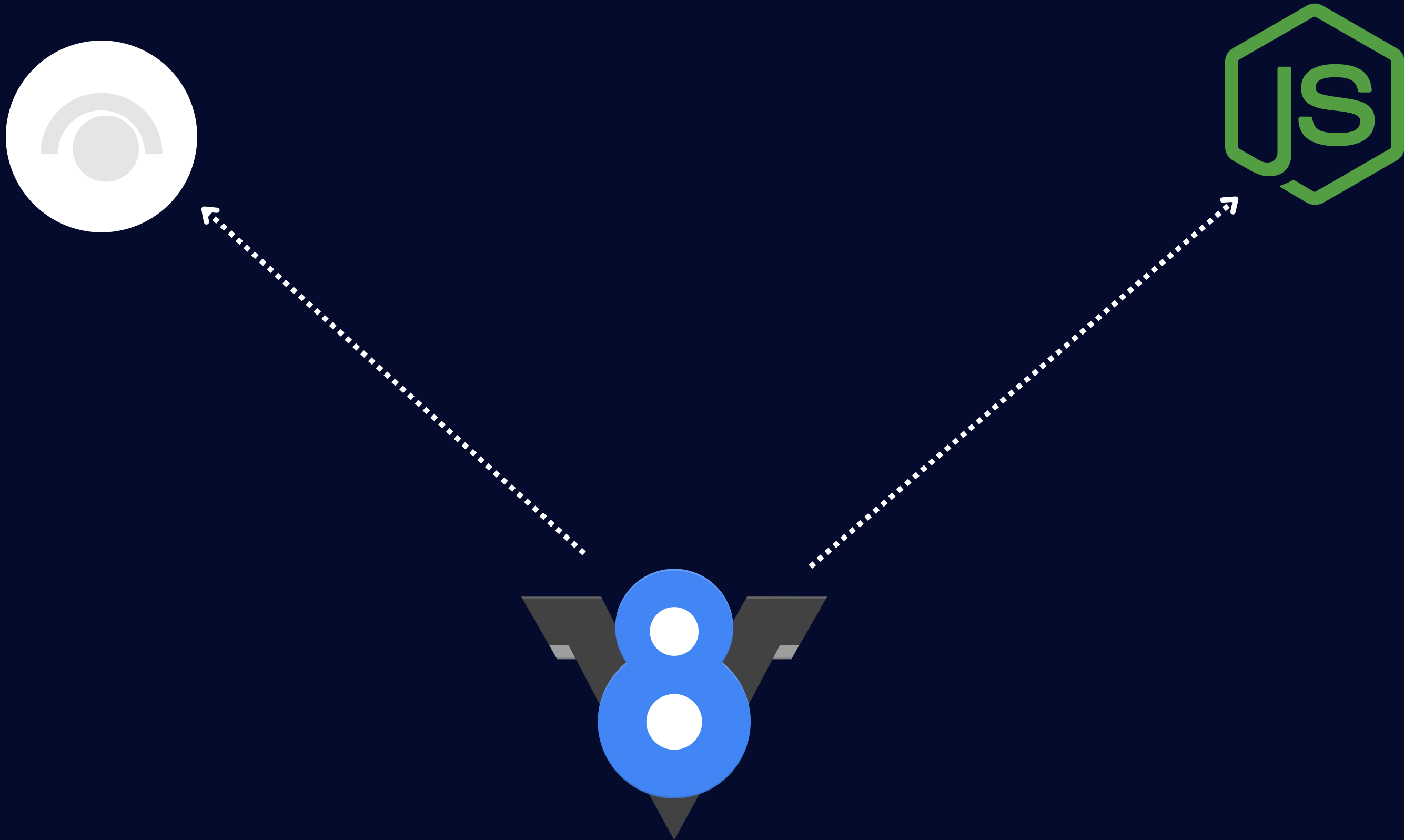
Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O. (input and output) model that makes it lightweight and efficient

JS Environments









Node js was built on
top of chrome v8
engine



Node js was built on
top of chrome v8
engine





Why use Node JS?



Why use Node JS?



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

- 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 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 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



* !

*



. * ! *



ZZZ LQRYRM#NDFDGHP \ FRP



LQRYRM#N DFDGHP \



LQRYRM#NDFDGHP \





&ö.ö<[3 † öé î [[+êî k<3d

▶ &ö.ö<[3 † öé î [[+êî k<3d

▶ p ÷ 3 ê î ké<kd

► &ö.ö<[3 † öé î [[+êî k<3d

► p † 3 ê î ké<kd

► ^öî k3 ^öî † 2 ö î [[+êî k<3d

► &ö.ö<[3 † öé î [[+êî k<3d

► p † 3 ê î ké<kd

► ^öî k3 ^öî † 2 ö î [[+êî k<3d

► † ^öî 2 3 ..đö< î 3đ î pđ <

► &ö.ö<[3 † öé î [[+êî k<3d

► p ð 3 ê î ké<kd

► ^öî k3 ^öî ð 2 ö î [[+êî k<3d

► —k^öî 2 3 ..đö< î 3đ î pđ <

► p ð 3 2 ê ^<dö^..êöd

Who uses Node JS?





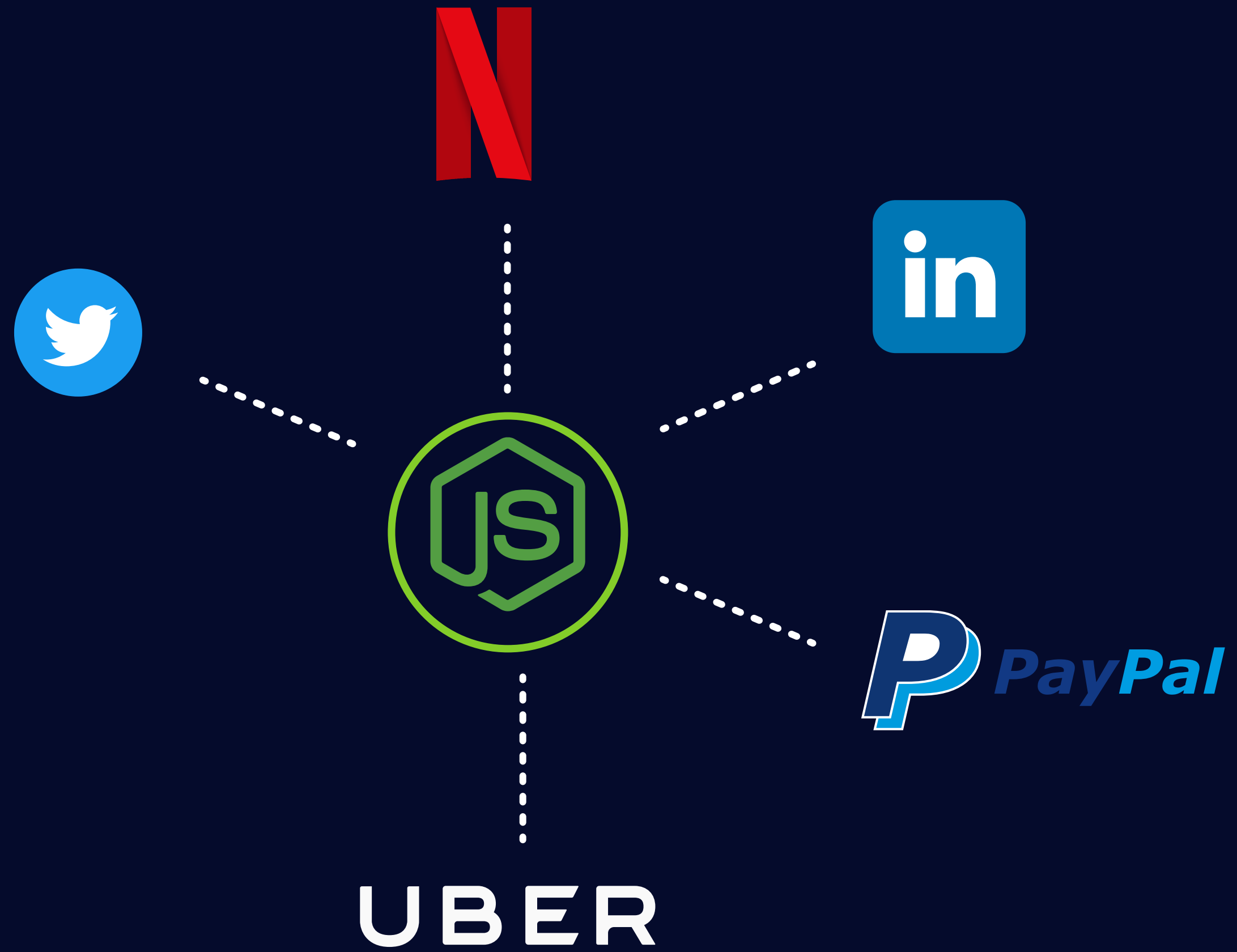


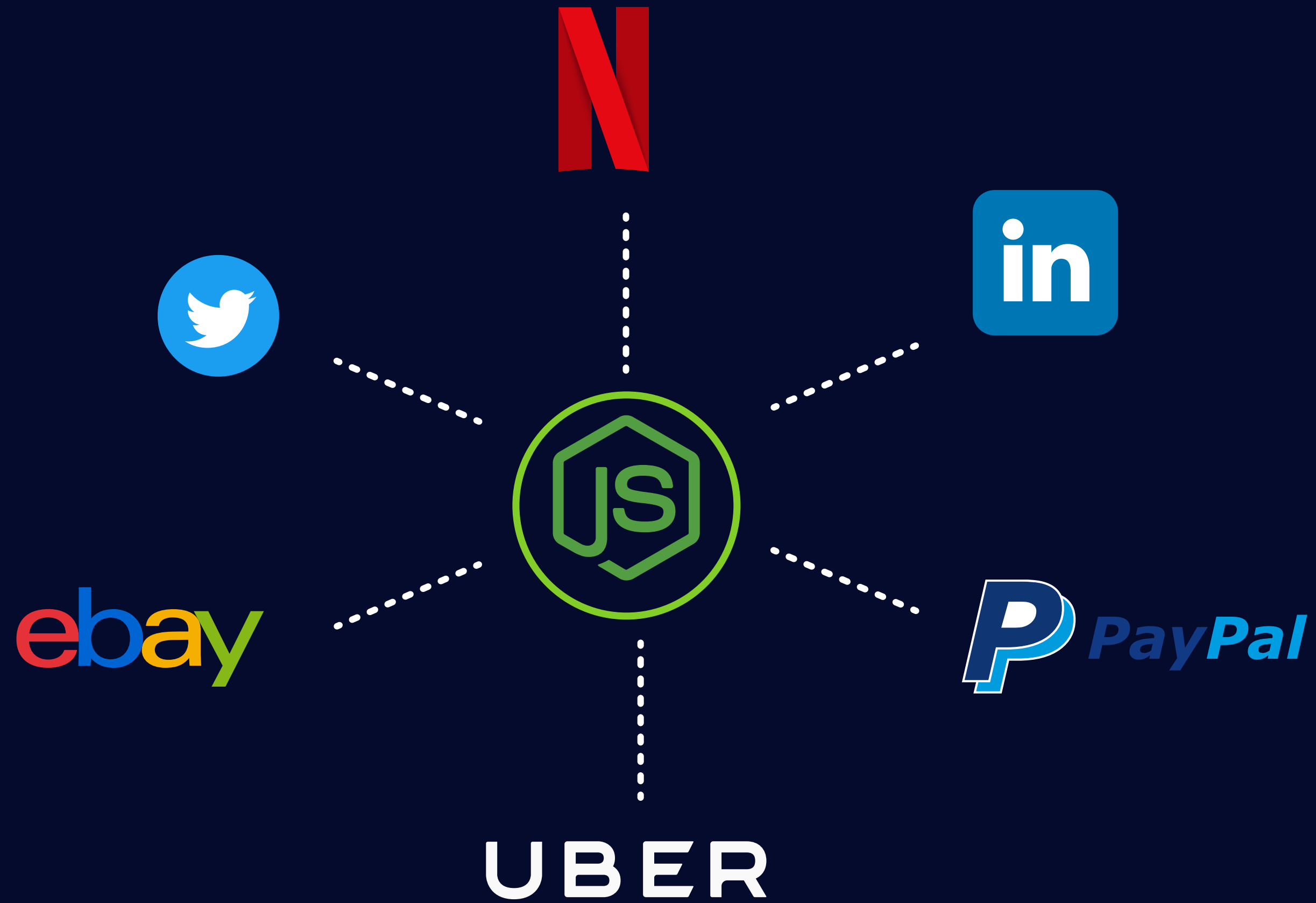
UBER













Install Node JS

R=Read

E=Evaluate

P=Print

L=Loop



NODE JS MODULES



fs-File System