EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLOGICO NACIONAL DE MEXICO

REST API

Members:
Castelan Delgado Lakshme Radharane
Gasca Aguilar Jaaziel Alberto
Leon Quintero Saul Andres
Sanchez Tapia Erick Daniel

INDEX

- DEFINITION
- HOW DOES IT WORK
- FEATURES
- ADVANTAGES
- DISADVANTAGES
- APPLICATIONS

CONTEXT

API

Application programming interface, is a set of rules that define how applications or devices can connect to and communicate with each other.

REST

Representational State Transfer is a software architecture that imposes conditions on how an API should work.

CONTEXT

CLIENT

Is the person or program using the API. The client makes requests to the API in order to retrieve some information or change something within the application.

RESOURCE

Is any piece of information that the API can provide the client. Each resource has a unique name, called the resource identifier.

CONTEXT

SERVER

Is used by the application that receives client requests, and contains resources that the client wants. The server has an API to interact with clients without giving them direct access to content stored in its database.

1 — DEFINITION

DEFINITION

A REST API is an application programming interface that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services.

2 — HOW DOES IT WORK

HOW DOES IT WORK?

Works by fielding requests for a resource and returning all relevant information about the resource, translated into a format that clients can easily interpret. Clients can also modify items on the server and even add new items to the server through a REST API.

HOW DOES IT WORK?

All requests to a REST API contains two pieces of information:

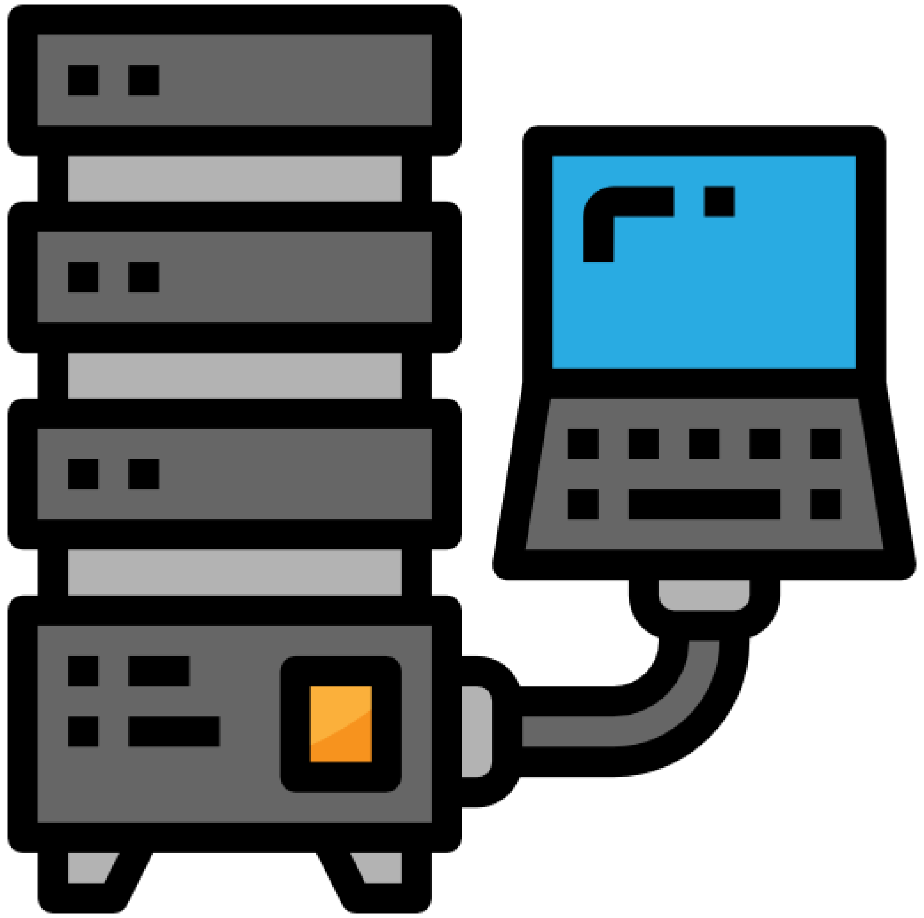
GET is the HTTP method: This specifies the action the client wants to make on the resource. The four basic HTTP requests a client can make are:
GET: To retrieve a resource.
POST: To create a new resource.
PUT: To edit or update an existing resource.
DELETE: To delete a resource.

https://... is the URL. The URL contains the uniform resource identifier, or URI, which specifies the target resource.

3 — FEATURES

Client-Server Separation

Under REST architecture, the client and server can only interact in one way: The client sends a request to the server, then the server sends a response back to the client.

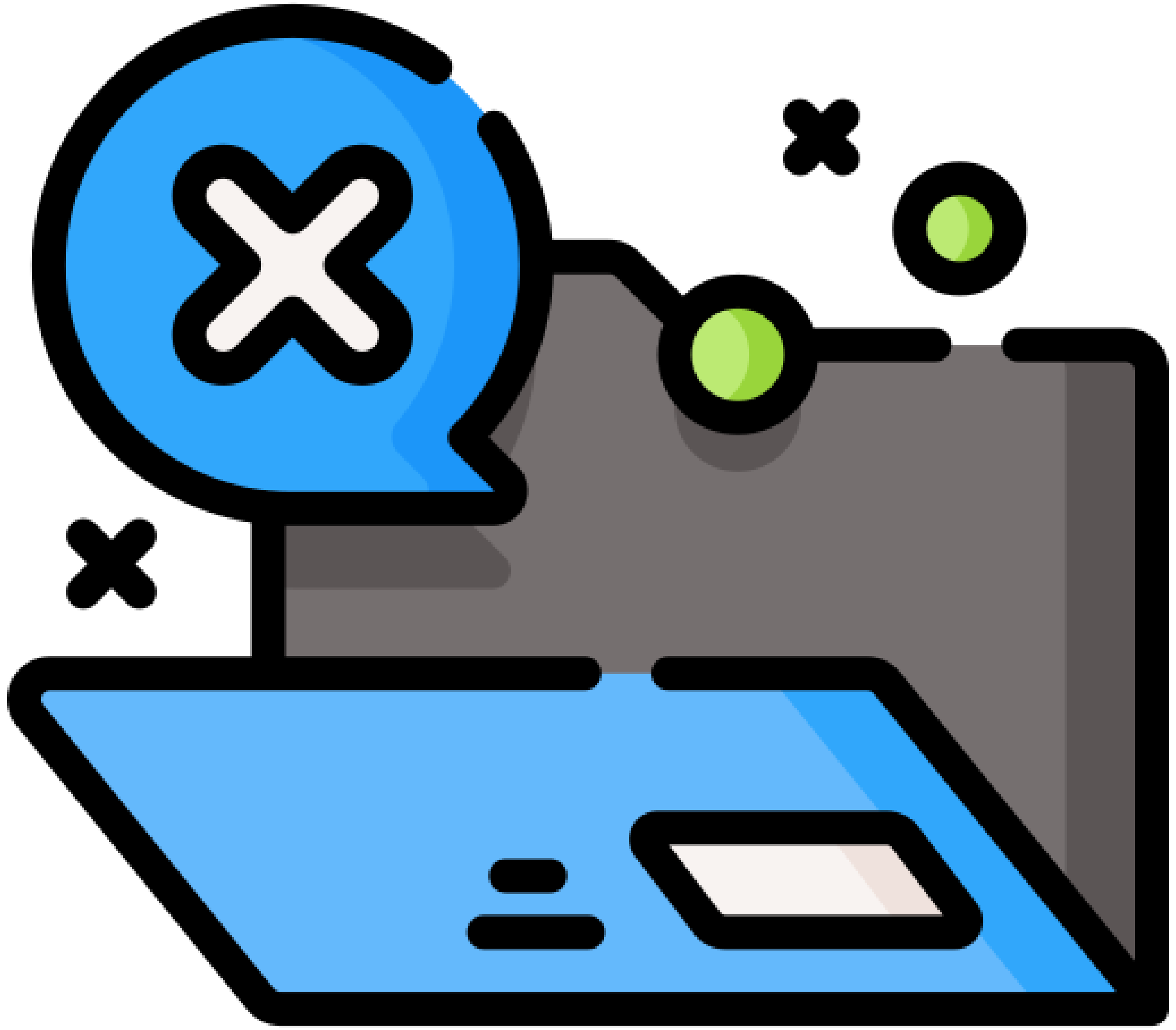


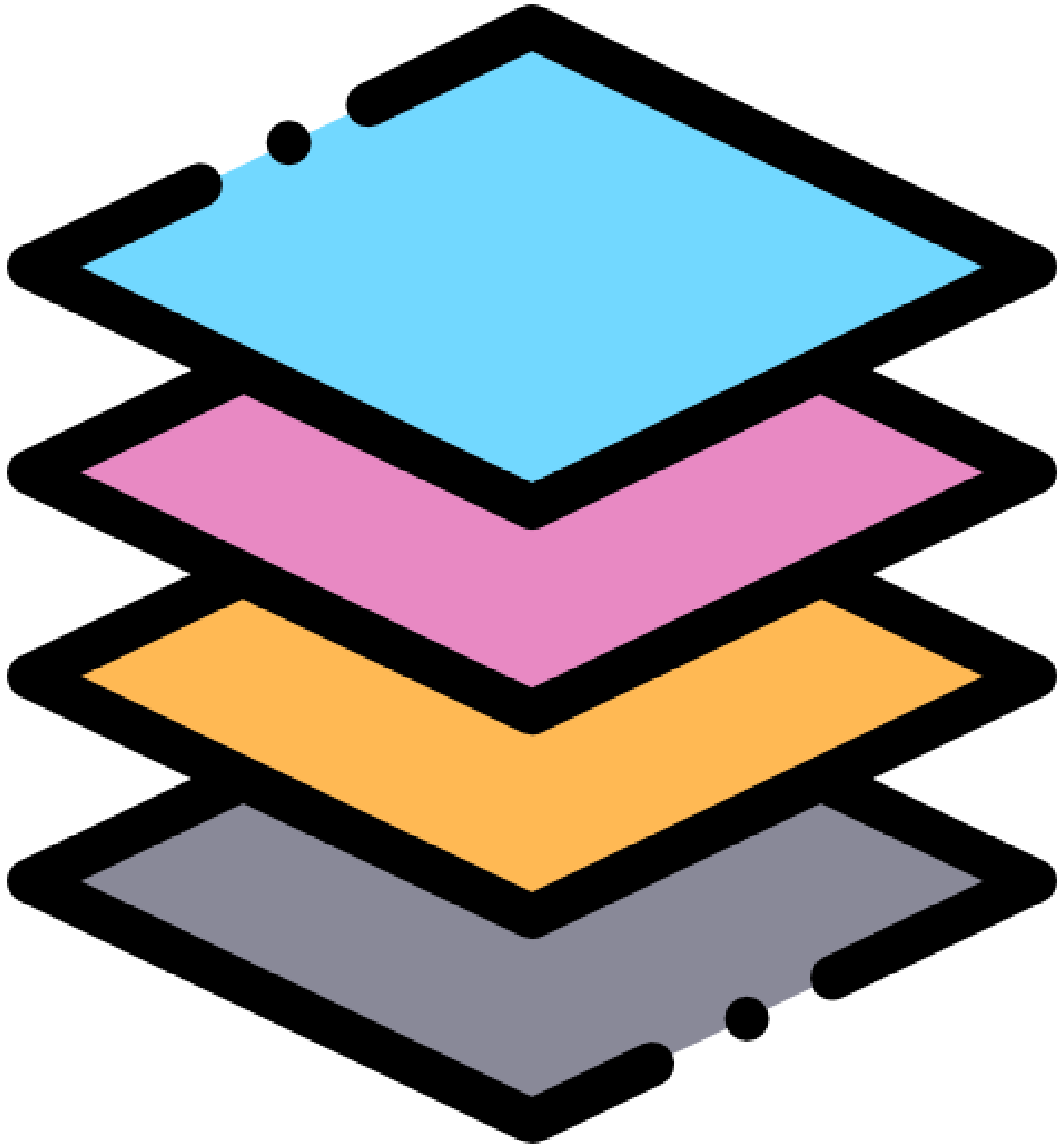
Uniform Interface

This guideline states that all requests and all responses must follow a common protocol, or a way of formatting their messages. Without standardized communication, translating requests and responses between software would be a total mess.

Stateless

This means that every interaction is independent, and each request and response provides all the information required to complete the interaction.

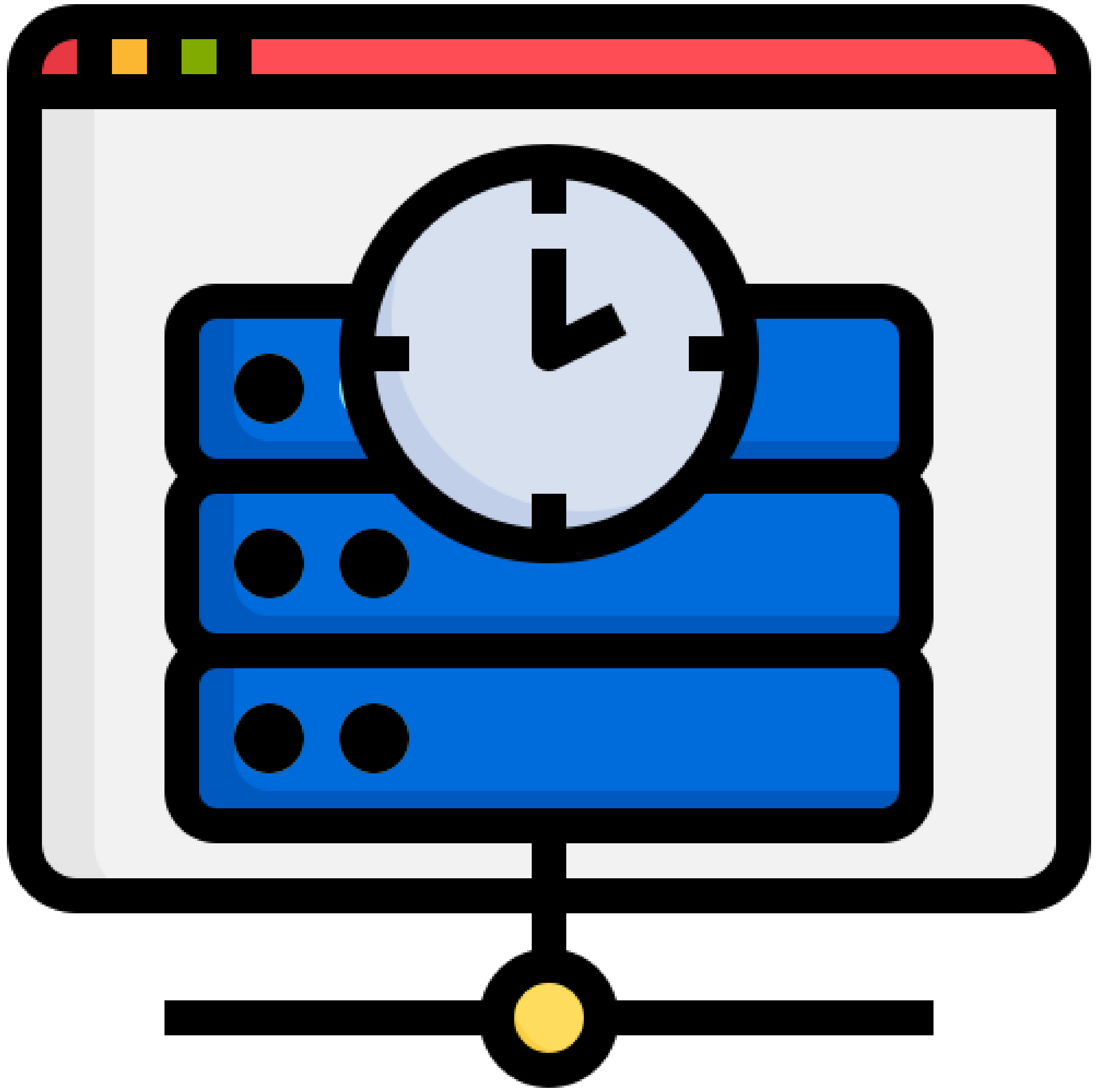




Layered System

This principle requires that messages between the client and target server should always be formatted and processed the same way, regardless of layers that sit between them.

Cacheable



Caching occurs when media is stored on a client’s device when visiting a website. When a client returns to that site, the cached data is loaded quickly from local storage instead of being fetched again from the server.

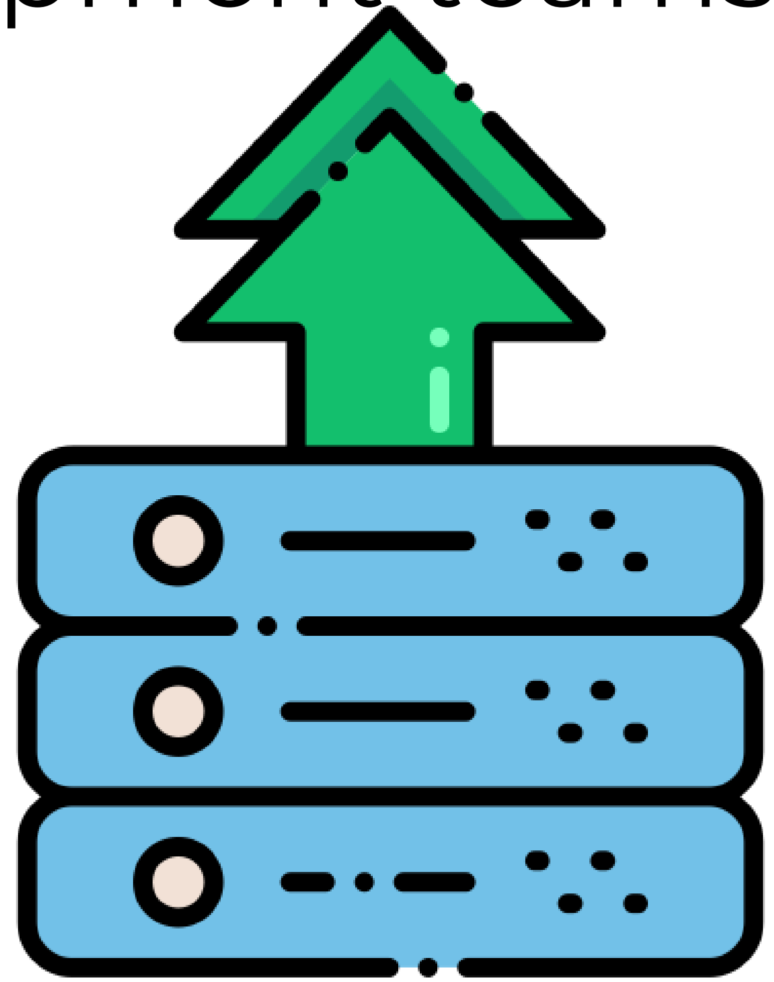
Code on Demand

If desired, an API can send computer code to clients in its response. This empowers the client to run the code in its own backend.

4 — ADVANTAGES

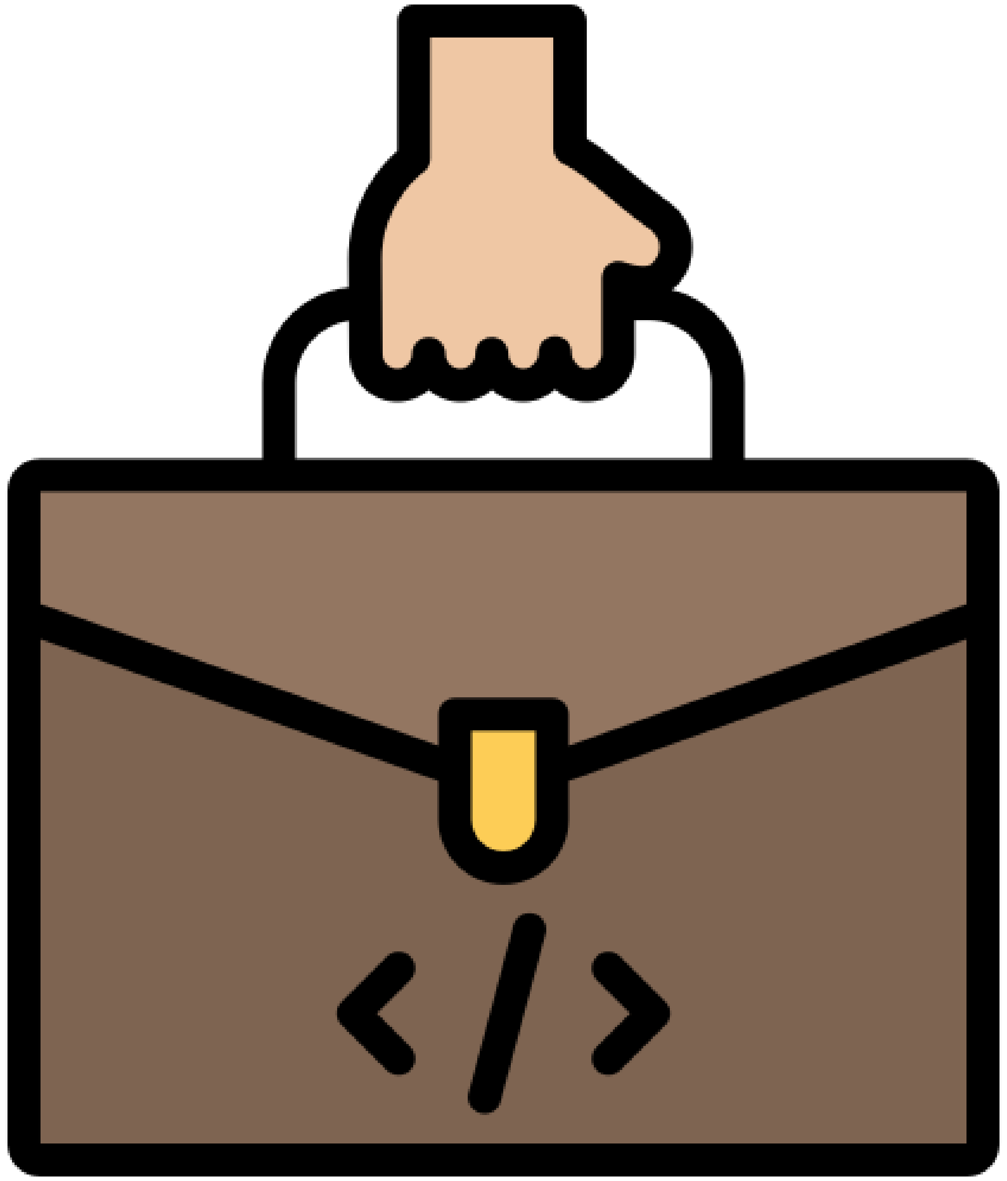
SCABILITY

Due to the separation between client and server, the product can be scaled by development teams without much difficulty.



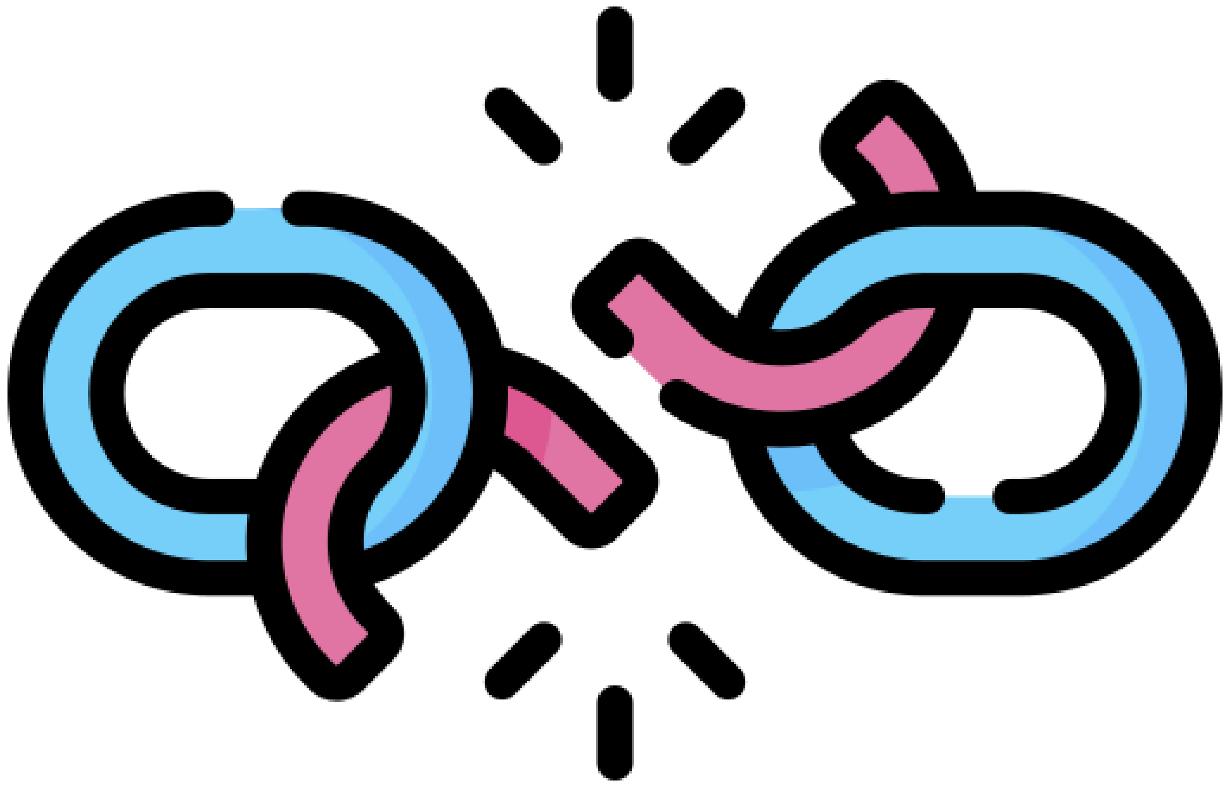
FLEXIBILITY AND PORTABILITY

Since REST-style APIs require data from one of the requests to be sent properly, it’s possible to perform a migration from one server to another. It’s also possible to carry out changes on the database at any time.



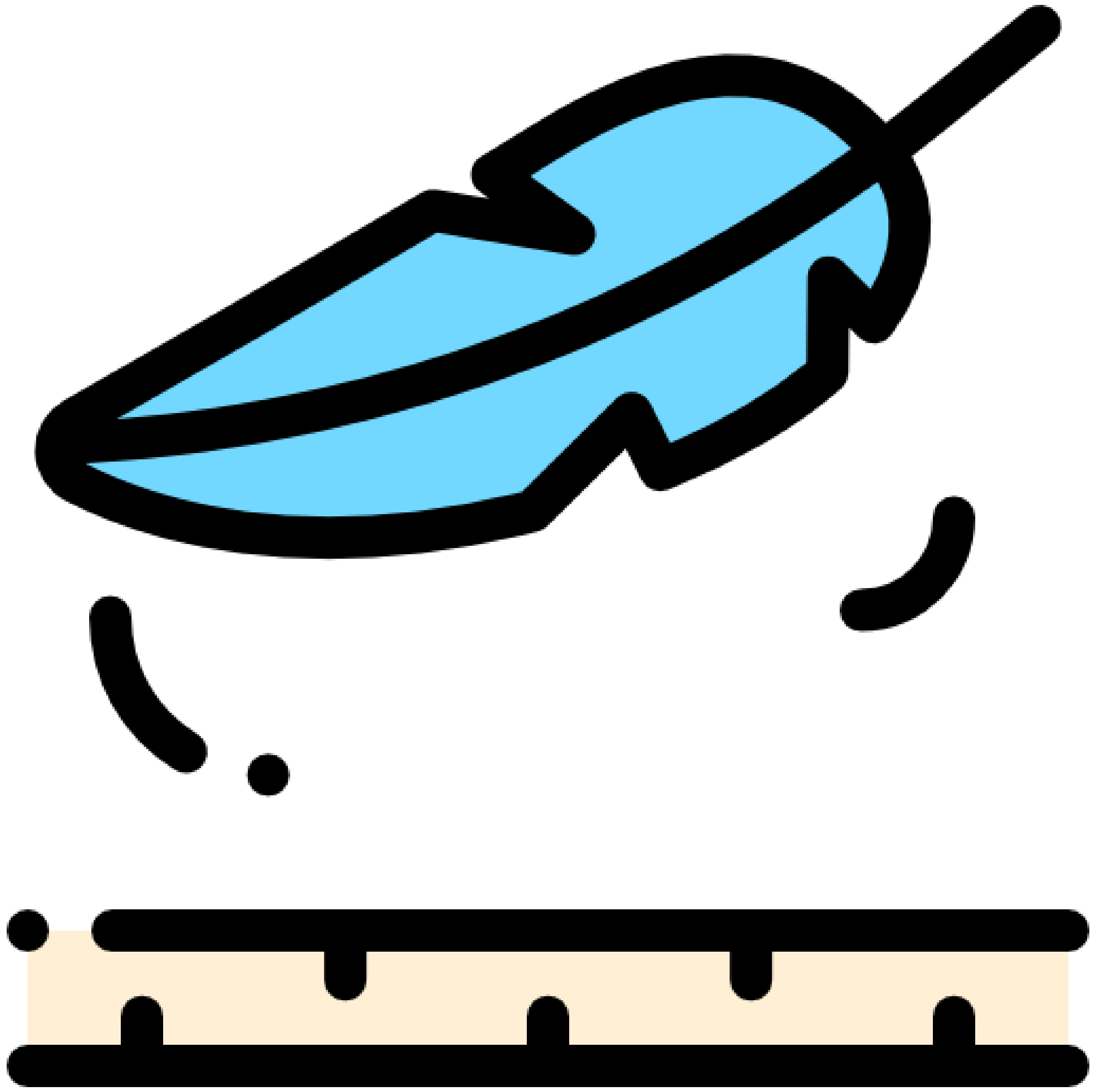
INDEPENDENCE

With the separation between client and server, the protocol makes it easier for developments across a project to take place independently. REST APIs are also adaptable to the working syntax and platform, which offers opportunities to test several environments at a time while developing.



LIGHTWEIGHT

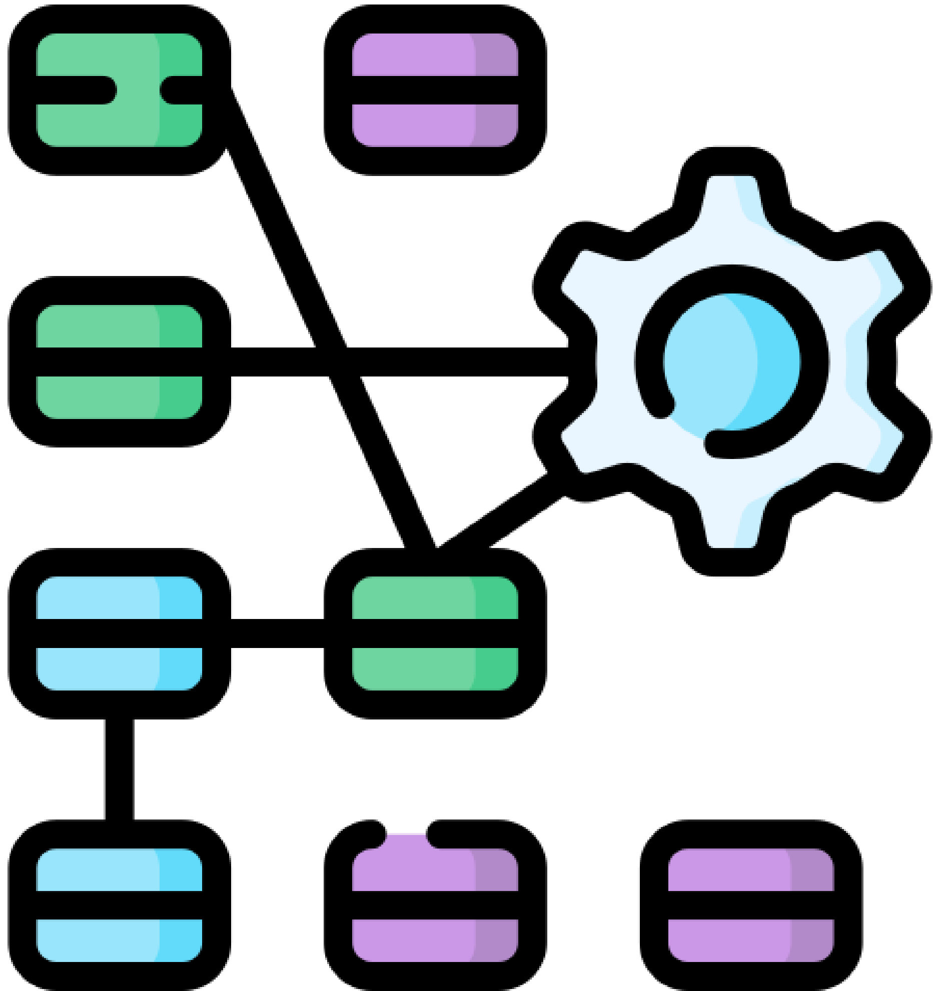
REST APIs are lightweight and fast, as they utilize the HTTP standard that supports multiple formats including JSON, XML, and HTML. This feature makes it ideal for mobile app projects, IoT devices, and much more.



5 — DISADVANTAGES

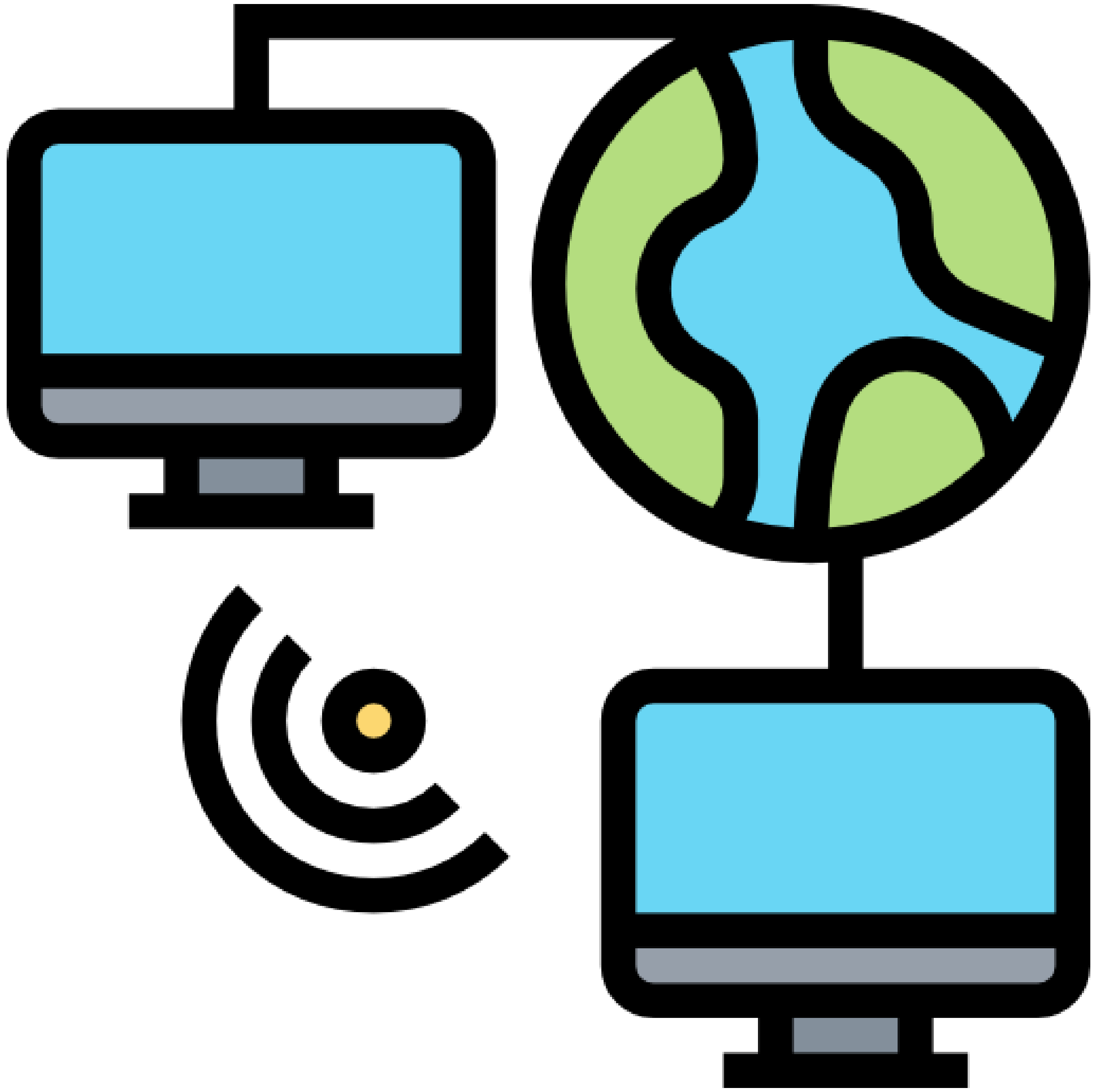
INCREASED DESIGN COMPLEXITY

The REST API uses architectural principles that you need to be familiar with, in order to develop your API.



WEB CONNECTION

A REST API requires an Internet connection to function, which means it may be less useful in offline environments.



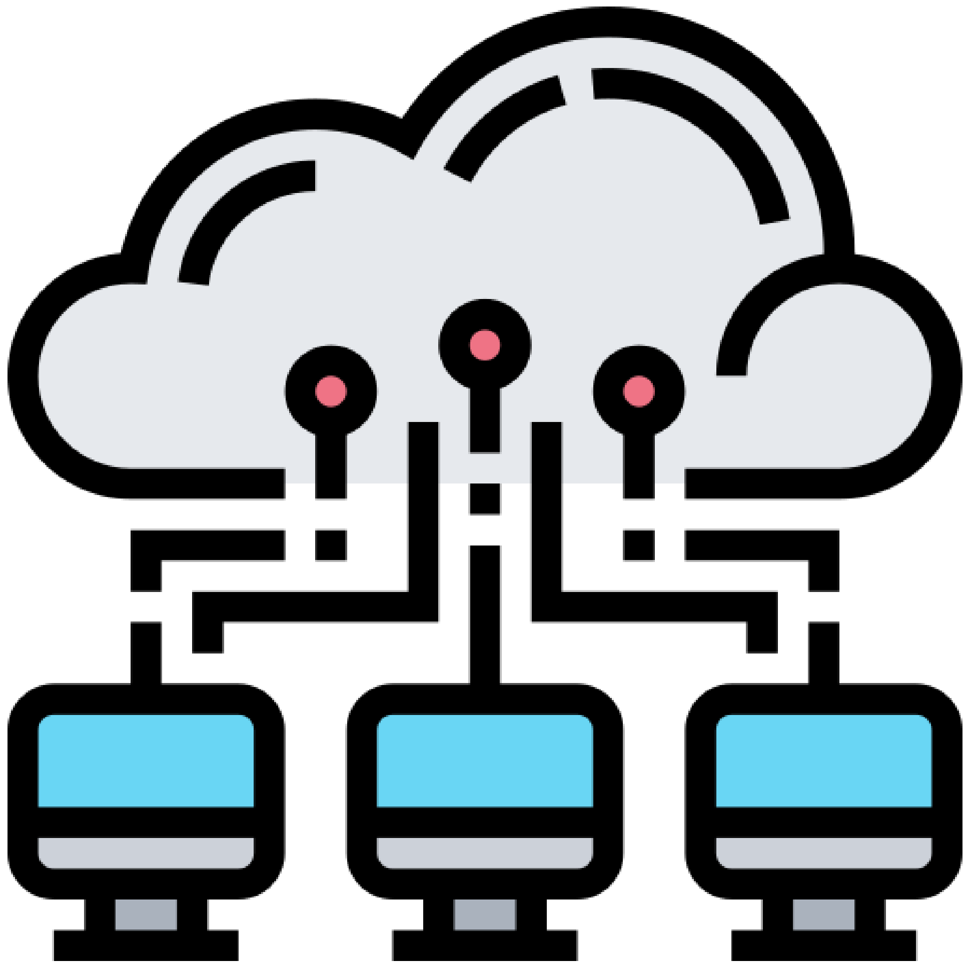
VARIABLE PERFORMANCE AND FLEXIBILITY

REST APIs can have slightly lower performance than other APIs, depending on the servers and their internet speed. Although its architecture is quite versatile and can be synchronized with other applications, the development of this architecture is less flexible.

6 — APPLICATIONS

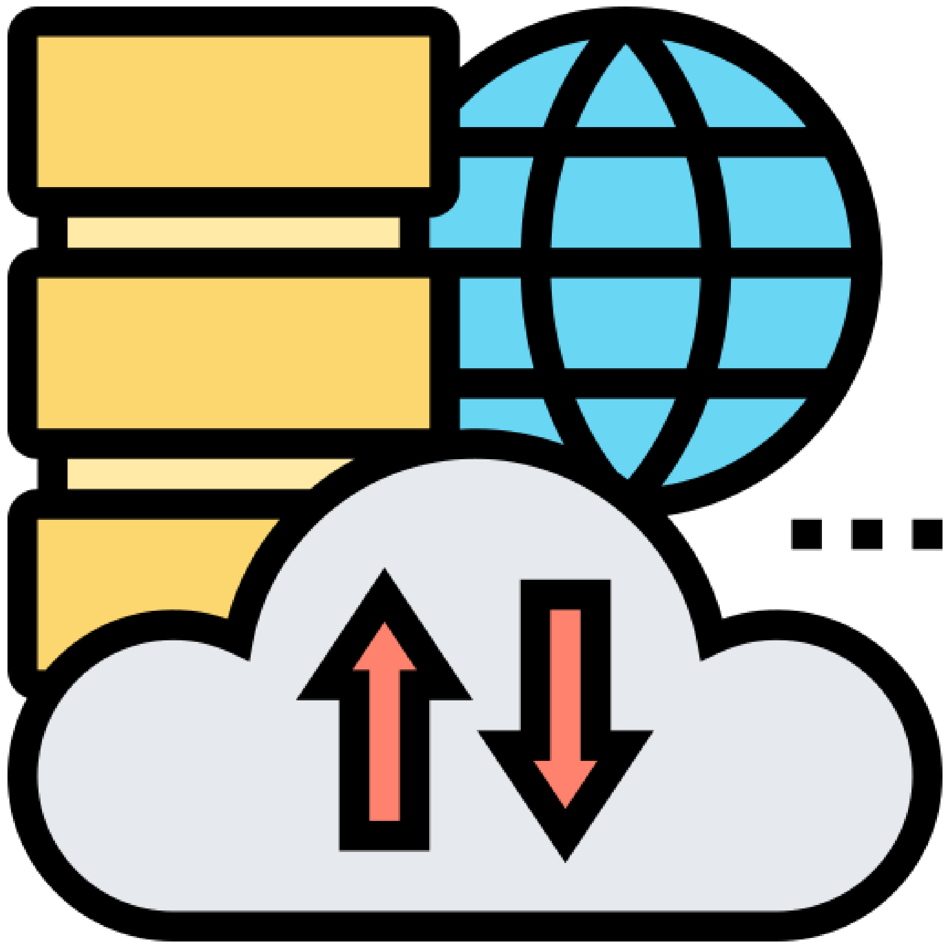
CLOUD-BASED APPLICATIONS

Due to the statelessness of its calls, REST APIs are advantageous in cloud applications. Stateless modules can easily reinstall and grow to meet the capacity requirement if something goes wrong.



CLOUD SERVICES

Programmers use RESTful web services to connect to cloud services using the internet.



TWITTER

Using the Twitter API, developers can streamline the registration process by leveraging Twitter's identification system. The API also enables the display of tweets to users based on criteria such as location or trending hashtags, as well as effective marketing using Twitter's data.

INSTAGRAM

The Instagram Basic Display API provides developers with access to profile data, images, and videos on the platform. This allows developers to build apps that integrate user data from Instagram into their own products.

AMAZON

AWS AI Services from Amazon allows developers to incorporate AI functionality into their applications for a more adaptive and intelligent interaction. This can also help to secure data exchange between systems by detecting potential security vulnerabilities.