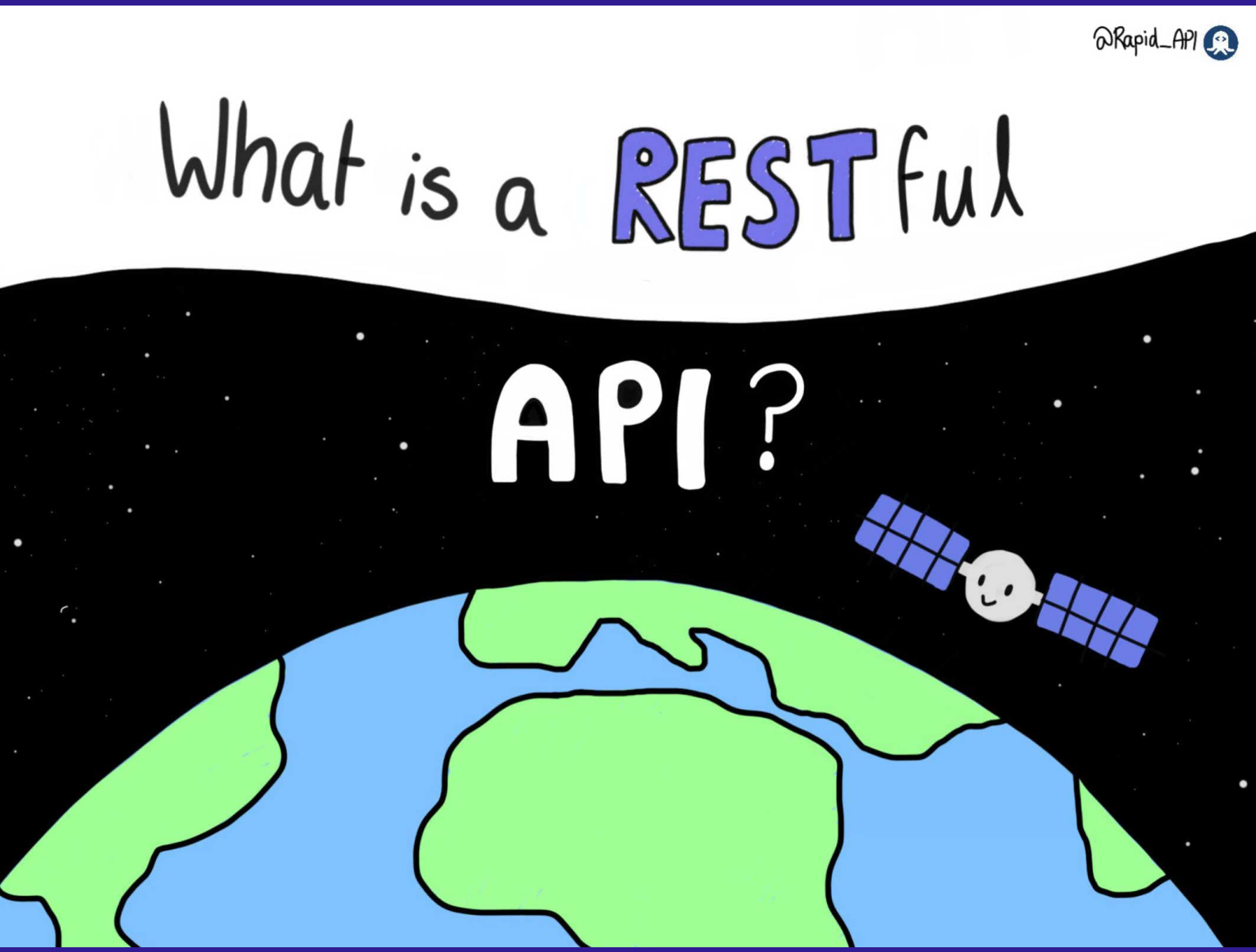

@Rapid_API
What is a RESTful API?

@Rapid_API

RESTful APIs follow REST architecture

↳ Representational State Transfer

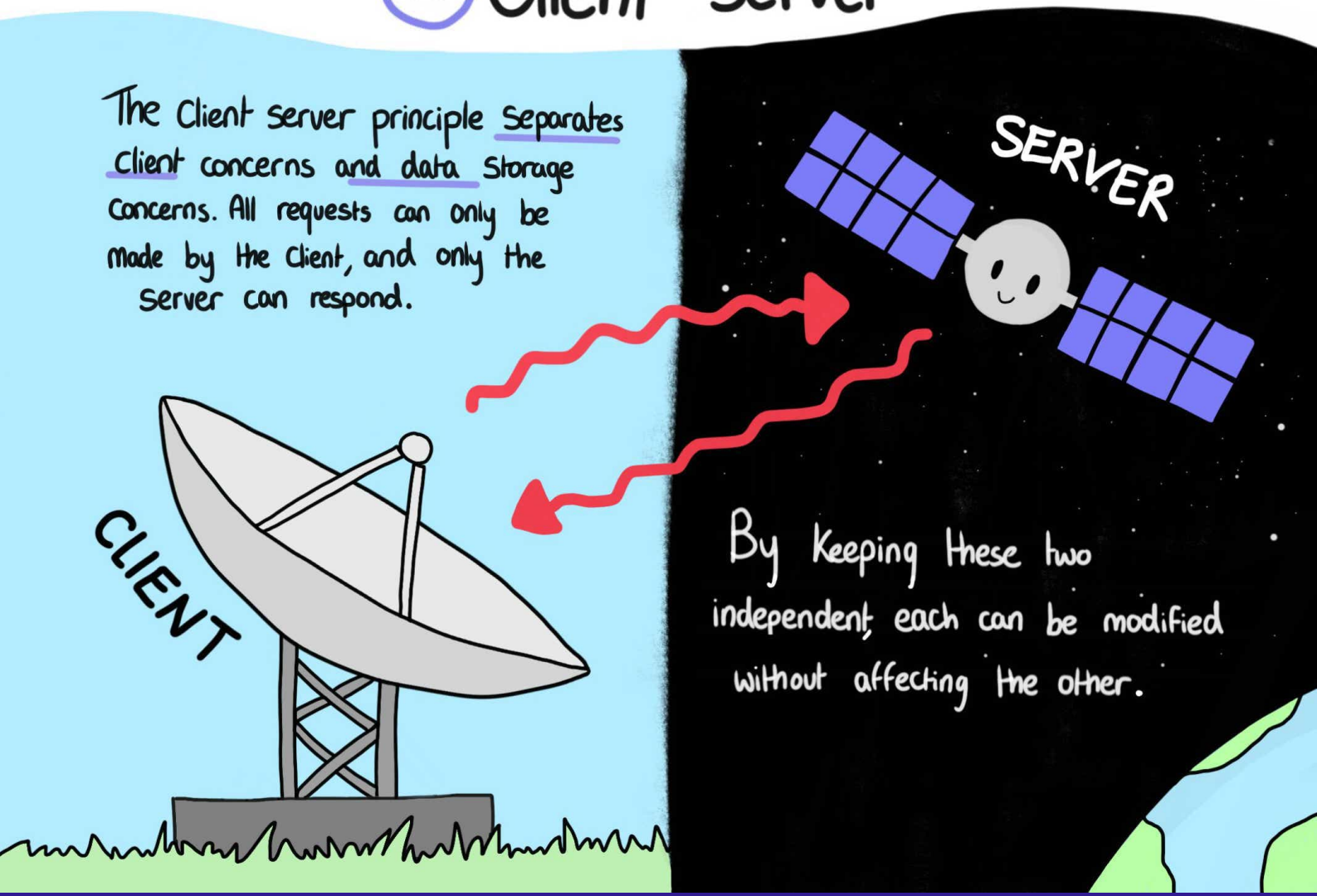
For an API to be RESTful, it must follow a number of principles, properties and constraints.

A deep dive into these principles ↓

1 Client-server

@Rapid_API

The Client server principle separates client concerns and data storage concerns. All requests can only be made by the client, and only the server can respond.



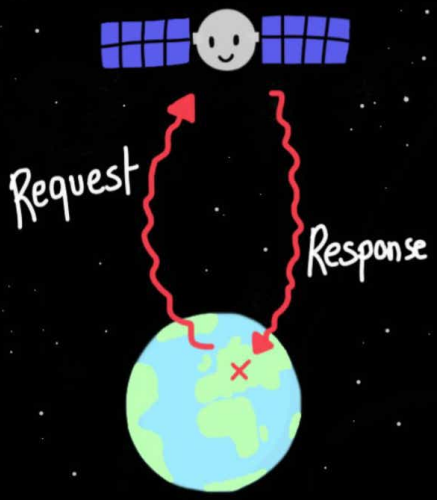
By keeping these two independent, each can be modified without affecting the other.

2 Uniform Interface

@Rapid_API

This principle requires that all responses follow the same format. Applications and servers can use different languages, so a uniform interface as an intermediary makes communication easier and simplified.

REST APIs use HTTP as their common Protocol.



Common HTTP Methods:

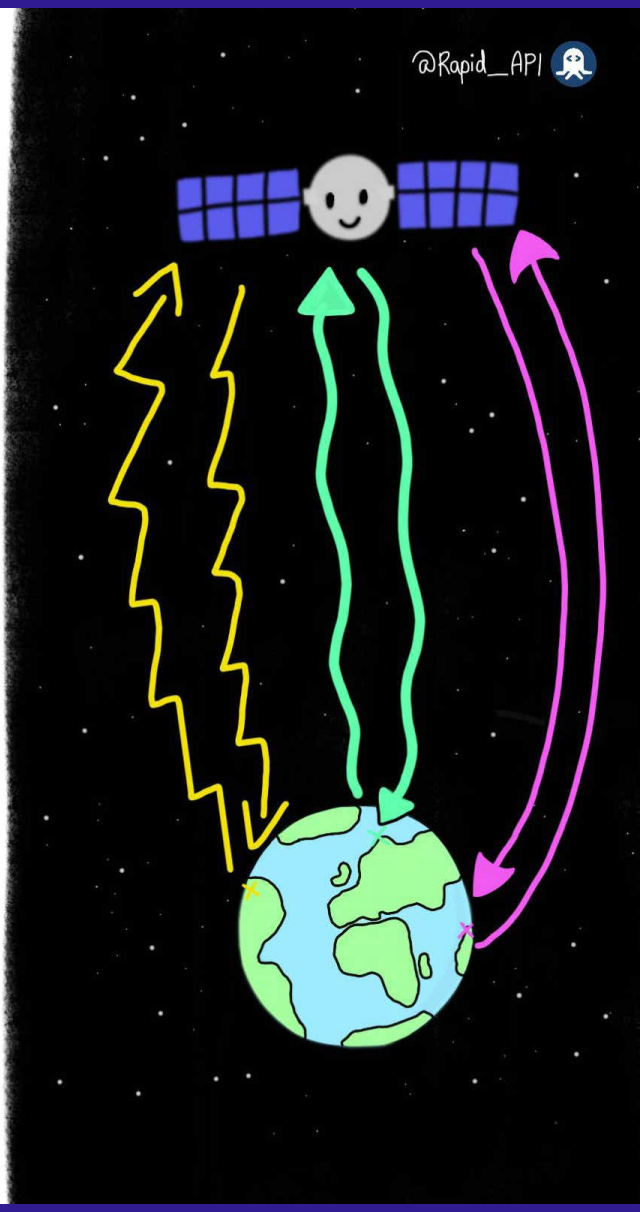
GET – Retrieves a resource.
POST – Creates a new resource.
PUT – Updates an existing resource.
DELETE – Deletes a resource.

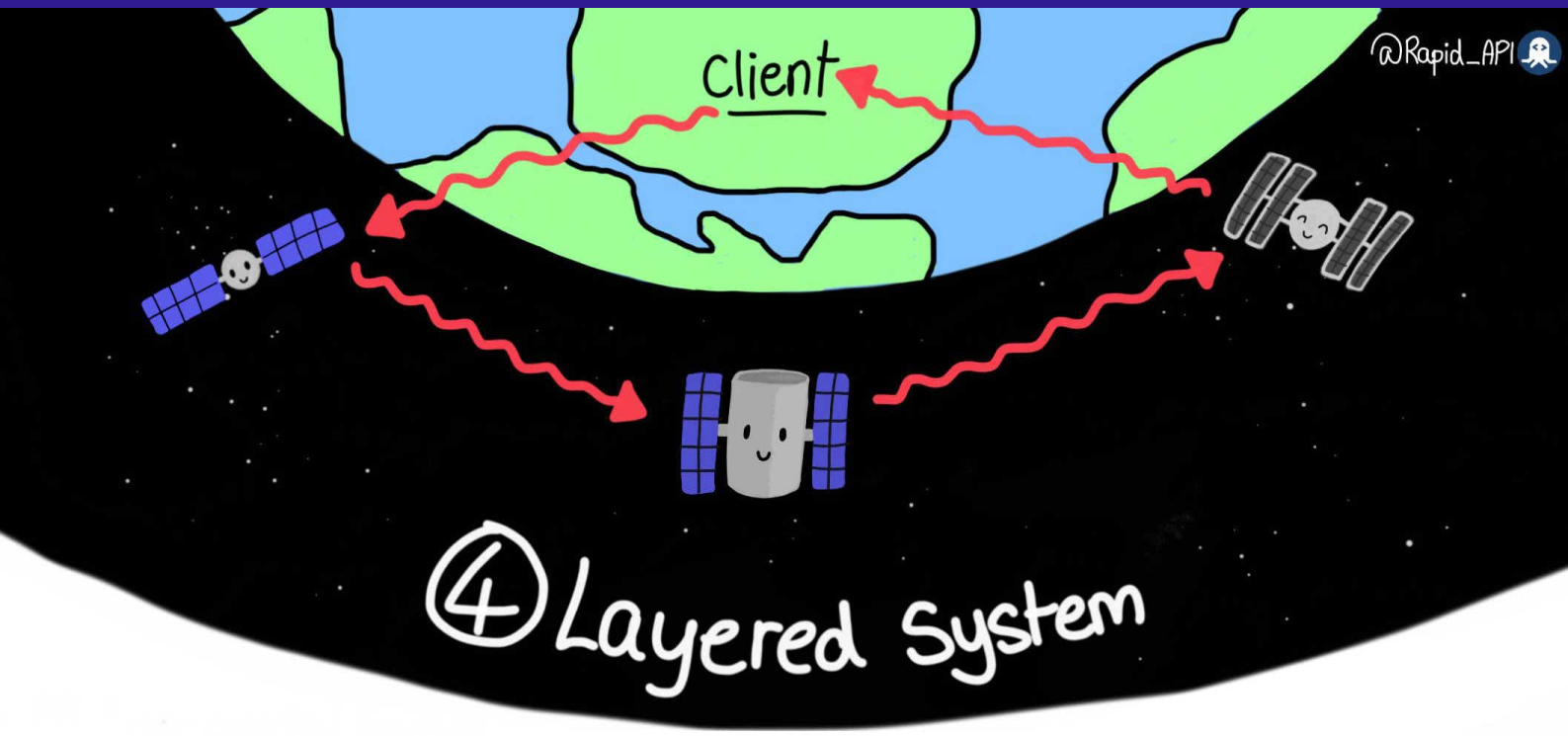
3 Stateless

Stateless means each server request is dealt with independently, regardless of previous requests.

Stateless transfers allow interactions to be scalable because less server memory is required, and theres no need to retrieve old data.

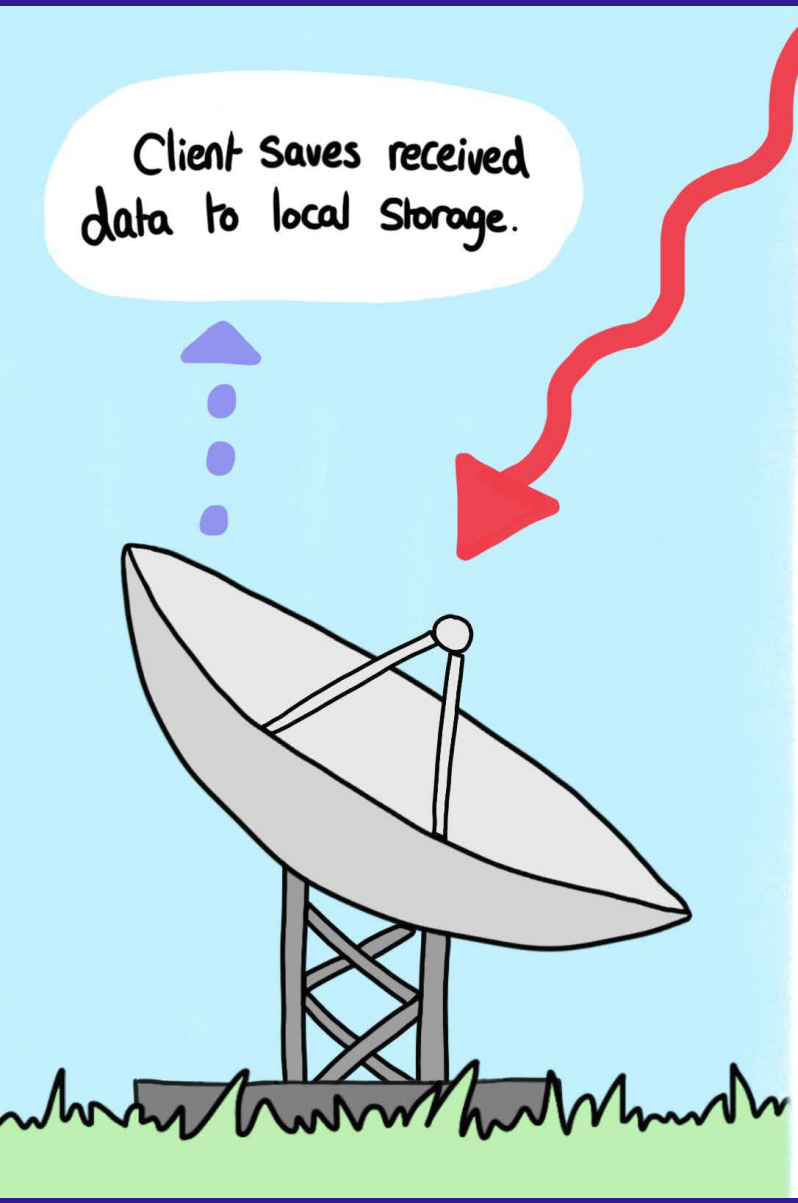
As software grows, using large amounts of memory isn't a concern.





Other servers (layers) between the client and server carry out other essential functions. The layered system principle requires data to be transferred in the same format.

This means servers can be modified or updated without affecting the API requests and responses.



@Rapid_API

5 Cacheable

Caching allows locally saved data to be loaded quickly when a user returns to a website.

REST APIs can indicate if a resource can be cached.

Caching reduces page load time and saves bandwidth.