Introduction to Web Services And API

OBJECTIVES

You are going to learn:

- 👉 What Is API
- 👉 What is Web Service
- 👉 Rest
- 👉 Asp .NET Core Web API

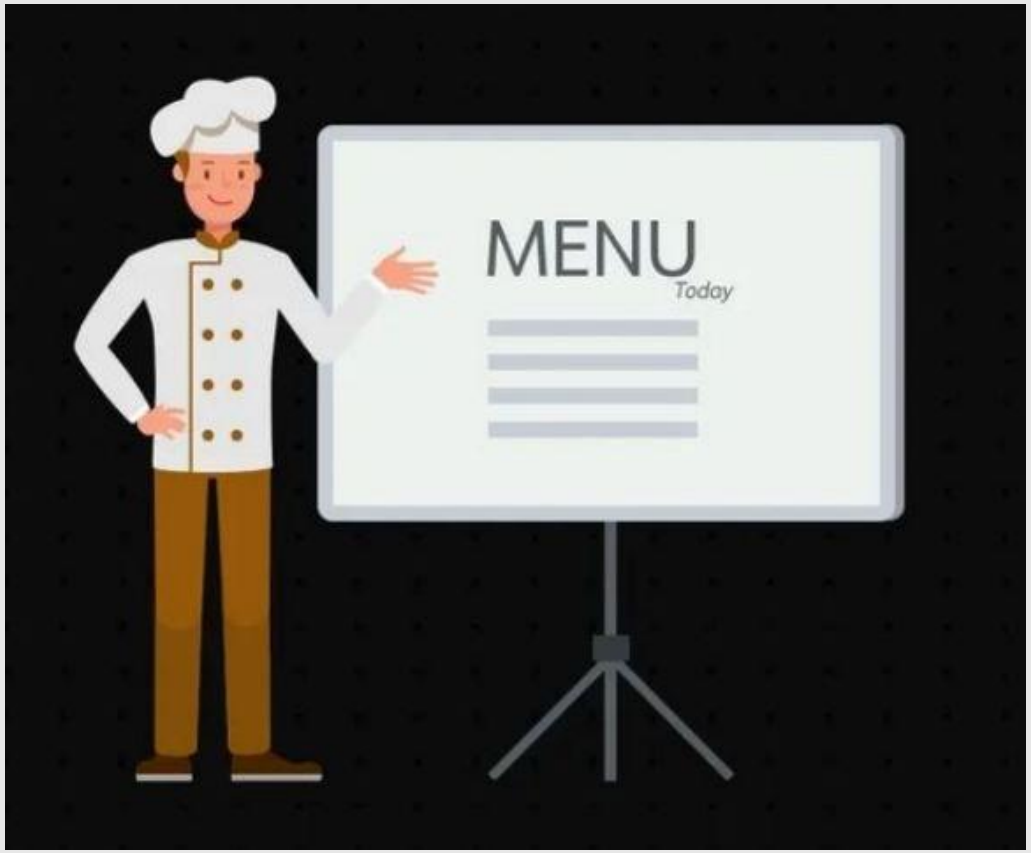
WHAT IS AN API?

Letter	Word	Meaning
A	Application	Software that does task
P	Programming	Program (P) that does the task in the Application (A)
I	Interface	Place (I) to tell the program (P) to run

So What is an API ?

- An API exists where you can tell (I) a computer program to run (P) to run in Application (A)
- At What Makes API great
  - Just use the program don't write it
  - Platform Independent
  - Upgrade Safe

WHAT IS AN API?



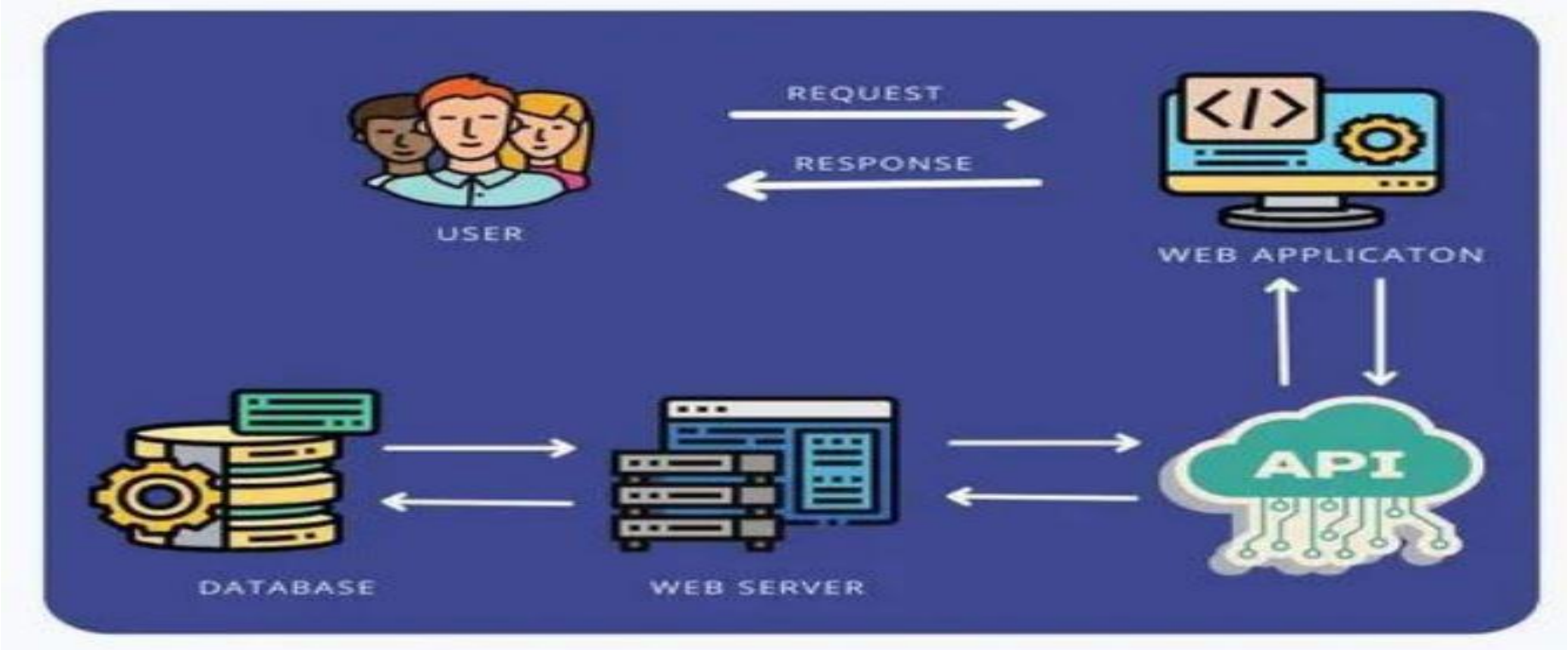
WHAT IS AN API?

WHAT IS AN API?

WHAT IS AN API?

API

Application **P**rogramming **I**nterface abbreviated as API which is a software intermediary that allows applications talk to each other.



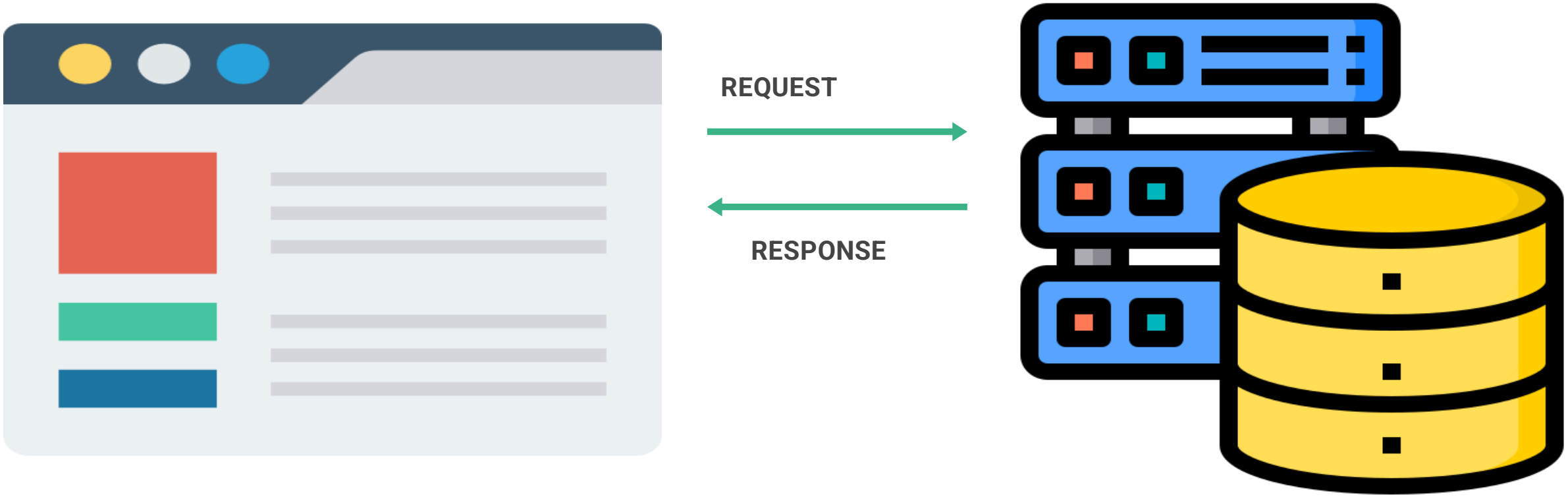
WHY WE NEED WEB API?

An API receives a request from the application. It then interreacts with a data source to collect and process a response, which is then returned to the application

Delivery app act as a waiter between you and restaurant

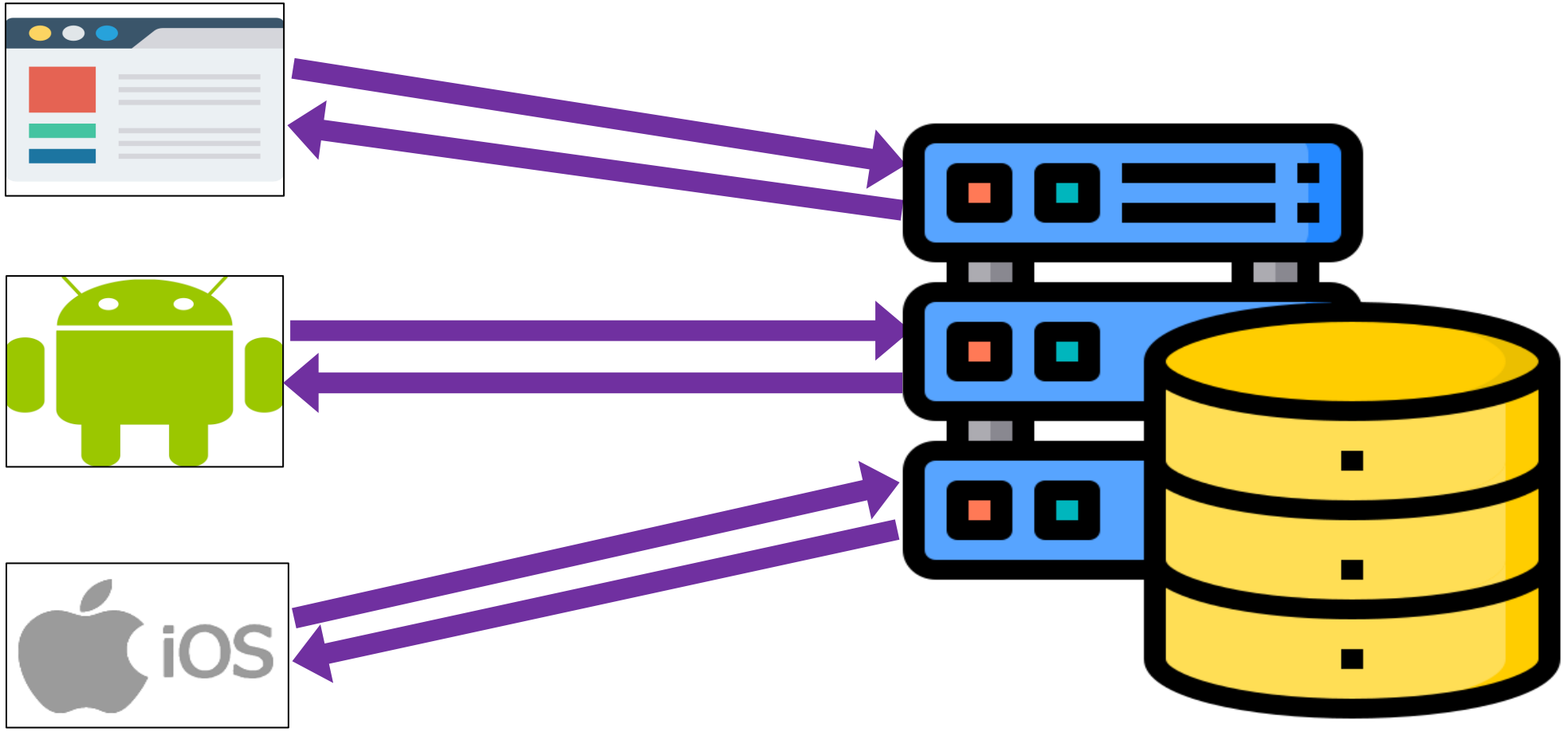
WHAT IS AN API?

Developing a web Application for users where they can manage, print electronic voucher system



WHAT IS AN API?

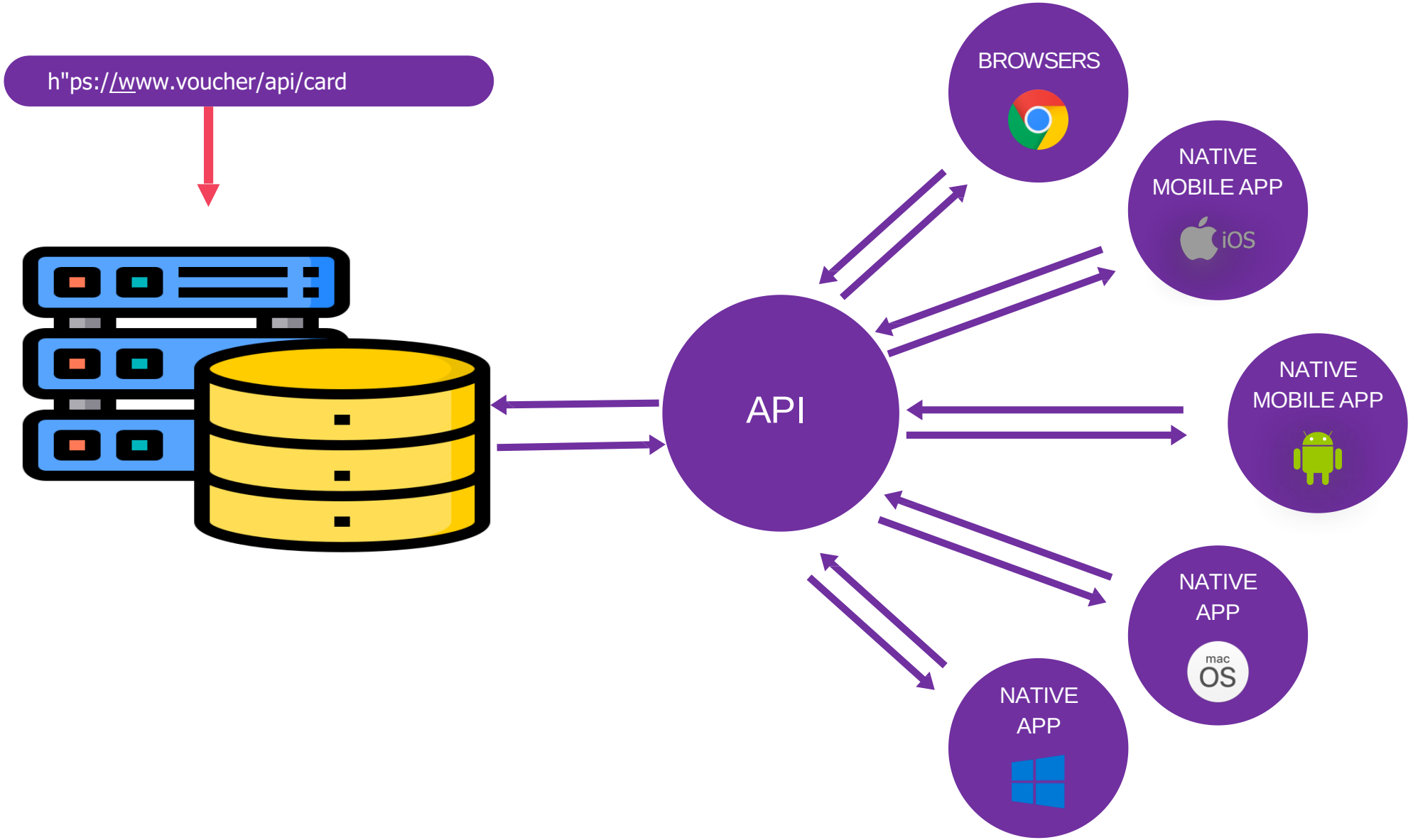
Developing a web Application for users where they can manage, print electronic voucher system



PROBLEMS

- Duplicate logic for each application
- Error prone code (Business Logic is written in each application)
- Some front end frameworks can not communicate with the database directly
- Hard to maintain

ONE API, MANY CONSUMERS



Types of Web API

TYPES OF WEB API

Types of API

Without DI

Email Sender

Dependency Graph

TYPES OF WEB API

Types of API

Public

Partner

Private

Public

Partner

Private or Internal

TYPES OF WEB API

Types of API

Public

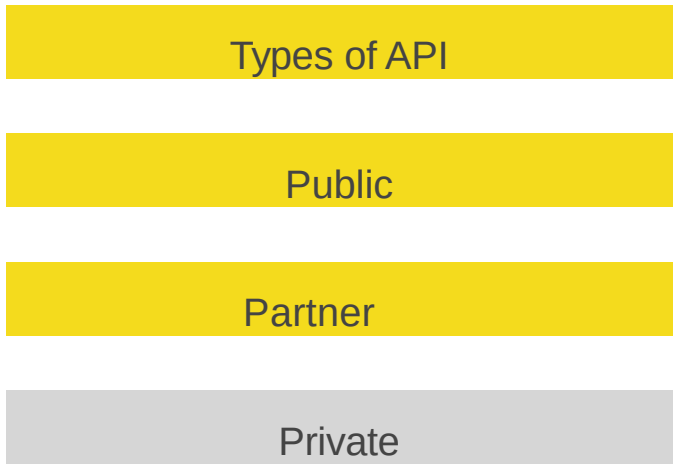
Partner

Private

- 👉 Available to any third party developers

- 👉 API’s are publicly available to developers and others users with minimal restriction. They may require registration or maybe completely open

TYPES OF WEB API



- 👉 Partners APIs are API exposed by/to the strategic business partners
- 👉 They are not available publicly and need specific entitlement to access them

TYPES OF WEB API

Types of API

Public

Partner

Private

- 👉 Private APIs, are hidden from external users and only exposed by internal systems
- 👉 They Internal APIs are not mean for consumption outside of the company but rather for use across different internal development

Web Service

WHAT IS A WEB SERVICE?

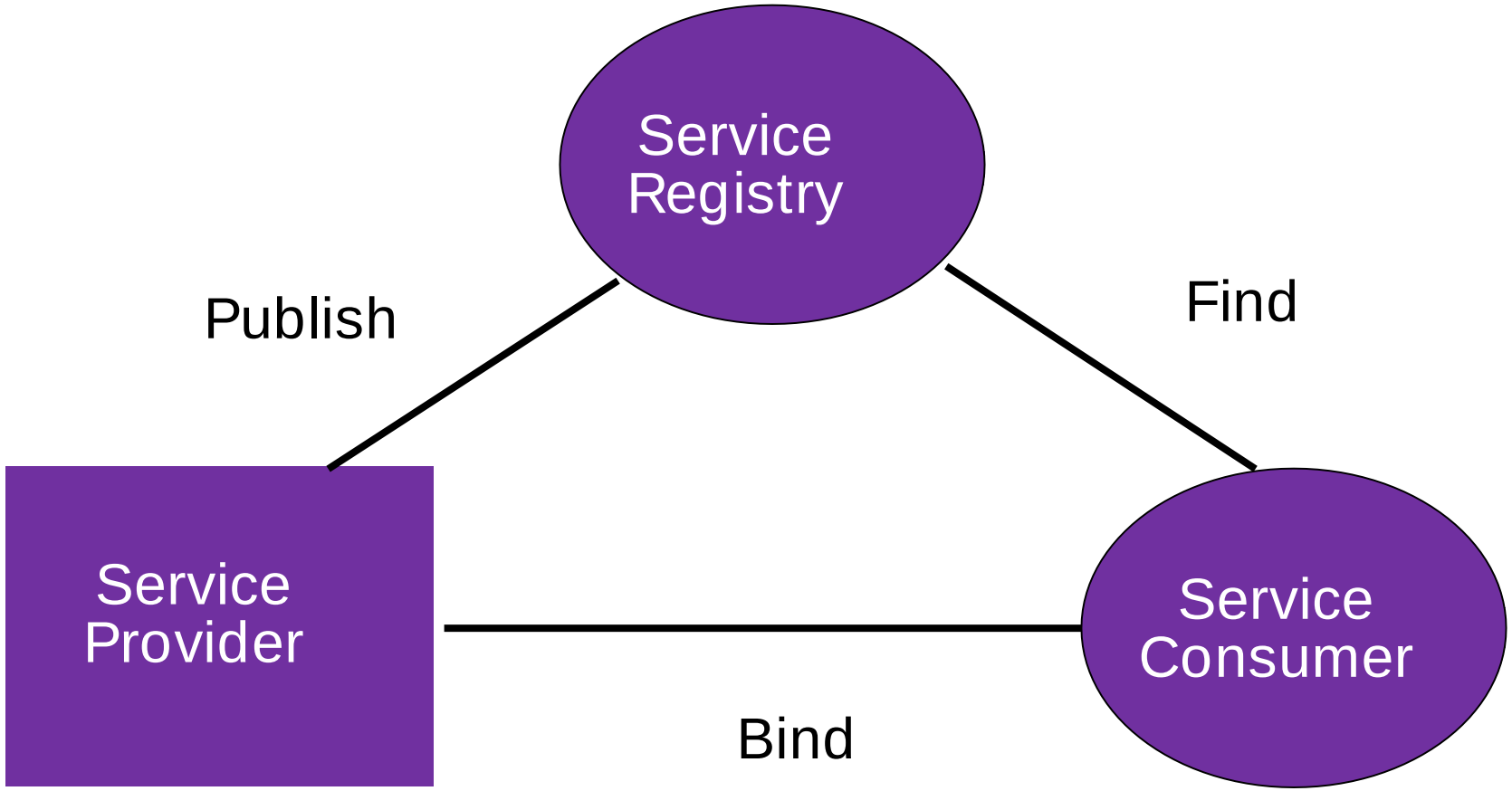
- Web = internet
- Service = API
- Web Service = API that uses the internet
- So, all web services is an API.
- But not all APIs are web services. Not all APIs use the internet

WHAT IS A WEB SERVICE?

- Web service is a means by which computers talk to each other over the web using HTTP and other universally supported protocols.

- A Web service is an application that:
  - Runs on a Web server
  - Exposes Web methods to interested callers
  - Listens for HTTP requests to invoke Web methods
  - Executes Web methods and returns the results

WEB SERVICE ARCHITECTURE



WEB SERVICE BASED ON

- HTTP (Hypertext Transport Protocol)
- SOAP (Simple Object Access Protocol)
- XML or JSON format to send over the Internet.
- UDDI (Universal Description, Discovery and Integration)
- WS-POLICY (Web Services Policy)

Soap And Rest

What is Soap and Rest ?

- Ways of doing web services that its API go over the internet
- Ways to form HTTP Request and Response

REST & RESTful WEB SERVICES

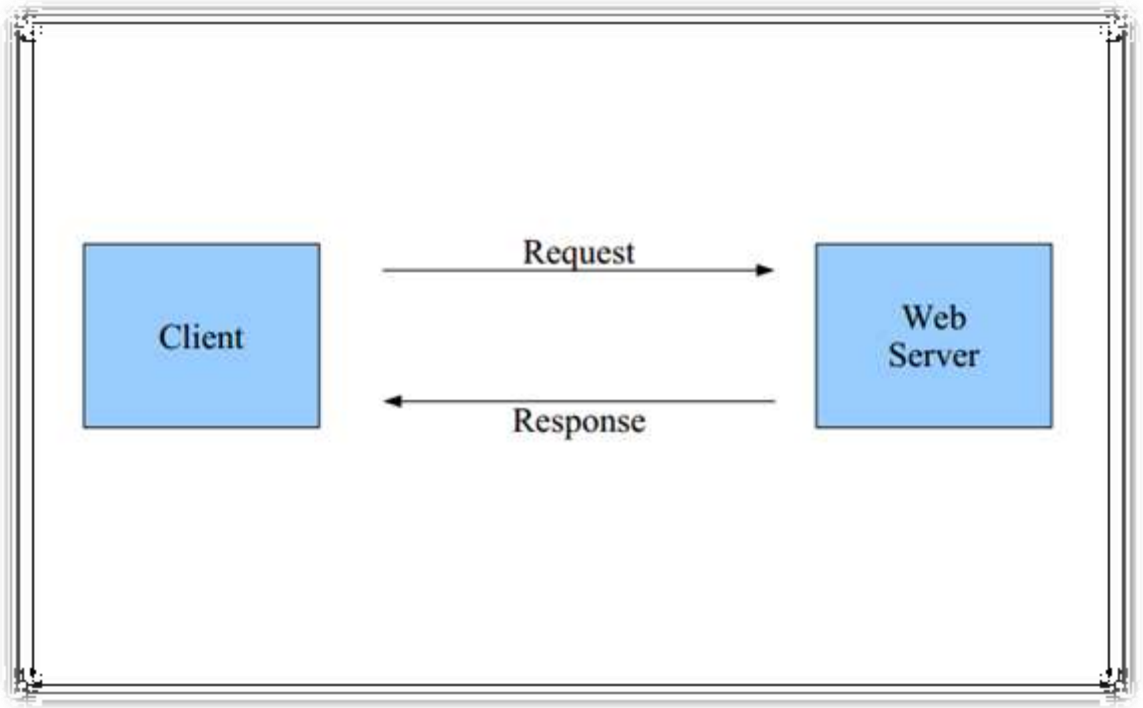
Rest

- **Re**presentational **S**tate **T**ransfer
- Architectural style (technically not a standard)
- REST is an architecture all about the Client-Server communication.
- REST is about resources and how to represent resources in different ways.
- REST is about client-server communication.
- REST is about how to manipulate resources.
- REST offers a simple, interoperable and flexible way of writing web services that can be very different from other techniques.

Architectural Style

- REST is the architecture of the Web as it works today and, so it is already used in the **web**!
- It is an software architectural model which is used to describe distributed systems like **WWW** (**W**orld **W**ide **W**eb).
- It has been developed in parallel with **HTTP** protocol.

THE WEB



Rest

- Client **request**s a specific **resource** from the server.
- The server **respond**s to that request by delivering the requested resource.
- Server does not have any information about any client.
- So, there is no difference between the two requests of the same client.
- A model which the representations of the resources are transferred between the client and the server.
- The Web as we know is already in this form!

Resources

- Resources are just consistent mappings from an identifier [such as a URL path] to some set of views on server-side state.

- Every resource must be uniquely addressable via a URI.

Requests & Responses

👉 REQUEST

```
GET /news/ HTTP/1.1
Host: example.org
Accept-Encoding: compress, gzip
User-Agent: Python-httplib2
```

Here is a **GET** request to «http://example.org/news/»

Method = **GET**

Requests & Responses

- And here is the response…
- RESPONSE

HTTP/1.1 200 Ok

Date: Thu, 21 Apr 2021 15:06:24 GMT

Server: Apache

ETag: "85a1b765e8c01dbf872651d7a5"

Content-Type: text/html

Cache-Control: max-age=3600

<!DOCTYPE HTML>

…

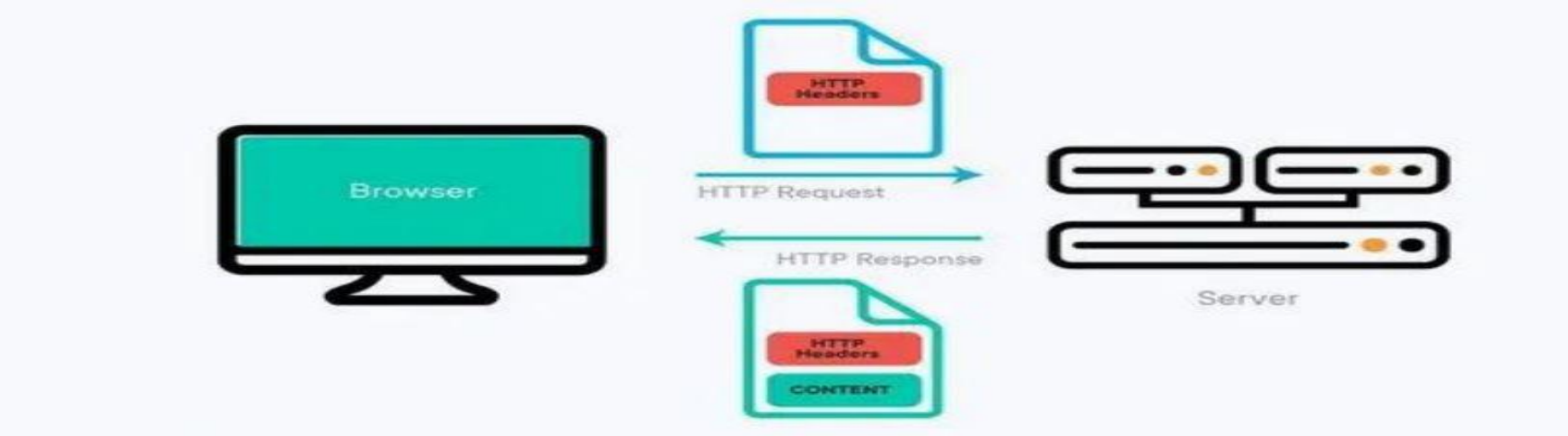
Requests & Responses

- The request is to a resource identified by a **URI** (**URI** = **U**nified **R**esource **I**dentifier).
- In this case, the resource is «http://example.org/news/»
- Resources, or addressability is very important.
- Every resource is URL-addressable.

Requests & Responses URI Examples

- http://localhost:9999/restapi/books
  - GET – get all books
  - POST – add a new book
- http://localhost:9999/restapi/books/{id}
  - GET – get the book whose id is provided
  - POST – update the book whose id is provided
  - DELETE – delete the book whose id is provided

HTTP Methods



- GET
- PUT
- POST
- DELETE

CRUD Operations Mapped to HTTP Methods in RESTful Web Services

OPERATION	HTTP METHOD
Create	POST
Read	GET
Update	PUT or POST
Delete	DELETE

RESTful Web Services

- RESTful web services are web services which are REST based.
- **Stateless** & **cacheable**.
- Uses **URL** & **HTTP** methods.
- Quiet light, extensible and simple services.
- The reason behind the **popularity of REST** is that the applications we use are **browser-based** nowadays and top it all, REST is built on **HTTP**.
- Main idea: Providing the communication between **client** and **server** over **HTTP** protocol rather than other complex architectures like SOAP and RPC etc.

RESTFUL Web Services

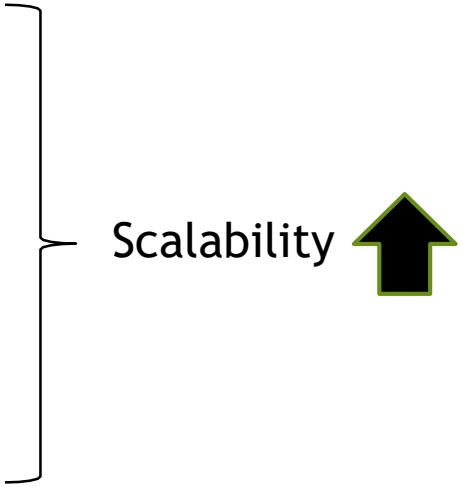
- You can do anything you already do with normal web services.
- No severe restrictions on how the architectural model will be and what properties it will have.
- Models like SOAP have severe rules, REST does not.
- There are lots of frameworks to develop RESTful web services on platforms like C# and Java, but you can write one easily using some standard libraries.

RESTFUL Web Services

- Platform independent.
- Language independent.
- Work on HTTP protocol.
- Flexible and easily extendible.
- They also have some constraints or principles.
  - *Client-Server*
  - *Stateless*
  - *Cacheable etc.*

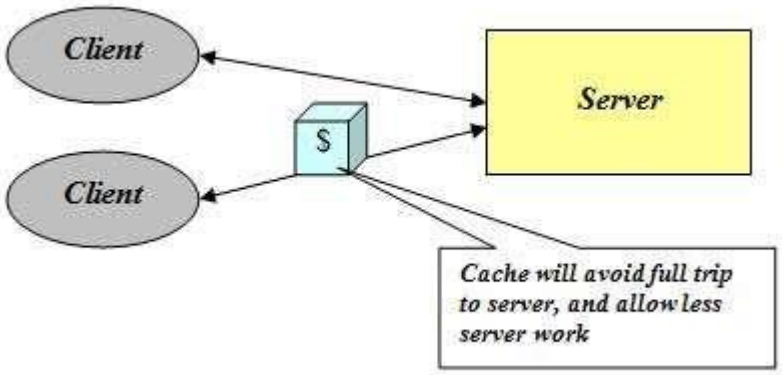
Stateless

- Each request is independent from other requests.
- No client session data or any context stored on the server.
- Every request from client stores the required information, so that the server can respond.
- If there are needs for session-specific data, it should be held and maintained by the client and transferred to the server with each request as needed.
- A service layer which doesn't have to maintain client sessions is much easier to scale.
- Of course there may be cumbersome situations:
  - The client must load the required information to every request. And this increases the network traffic.
  - Server might be loaded with heavy work of «validation» of requests.



Cacheable

- HTTP responses must be cacheable by the clients.
- Important for performance.
- If a new request for the resources comes within a while, then the cached response will be returned.



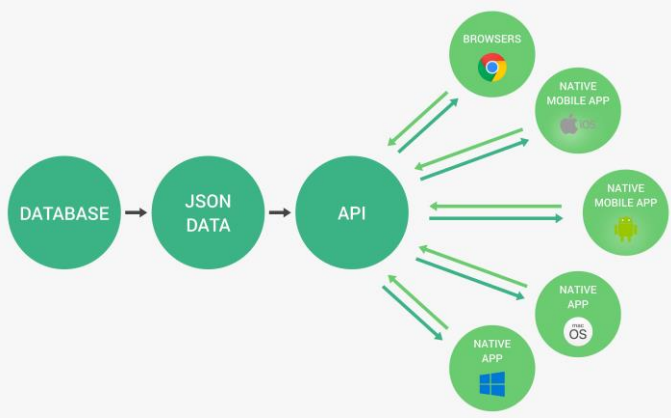
ASP.NET COREWEB API

ASP.NET CORE WEB API

1. ASP.NET Core Web API
2. Restful APIS
3. Use HTTP methods(verbs)
4.

☞**ASP.NET Core Web API:** A framework for building RESTful APIs using the HTTP protocol .

☞**ASP.NET Core Web API:** Is running over the .NET Framework



☞ services can then be consumed by a broad range of clients like:

- ☞ Node. Browsers
- ☞ Mobile applications;
- ☞ With Desktop applications;
- ☞ IOTs