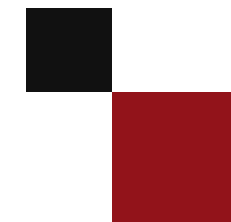


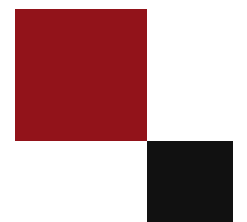
## Web API

Đỗ Thành Long  
*dtlong@opengis.vn*

<https://opengis.vn>



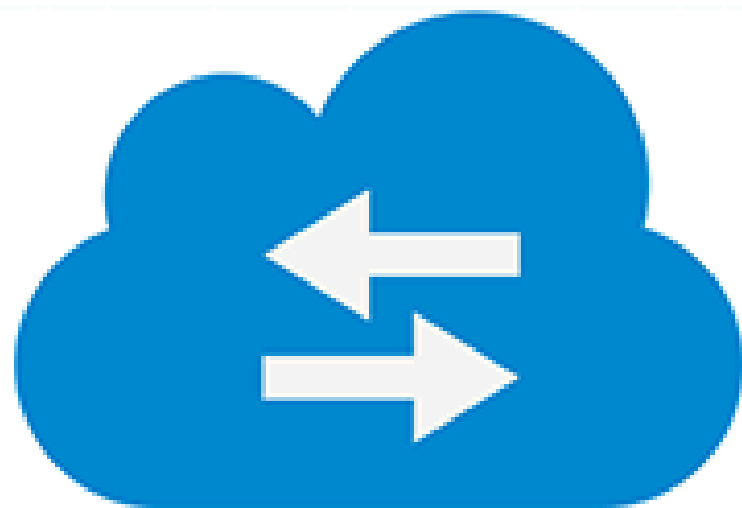
**OPENGIS**  
Discover the world, Learn with maps  
 <https://opengis.vn>



## Nội dung

- Giới thiệu về Web API
- Phương pháp tiếp cận
- Kiến trúc Web API
- Mô hình triển khai





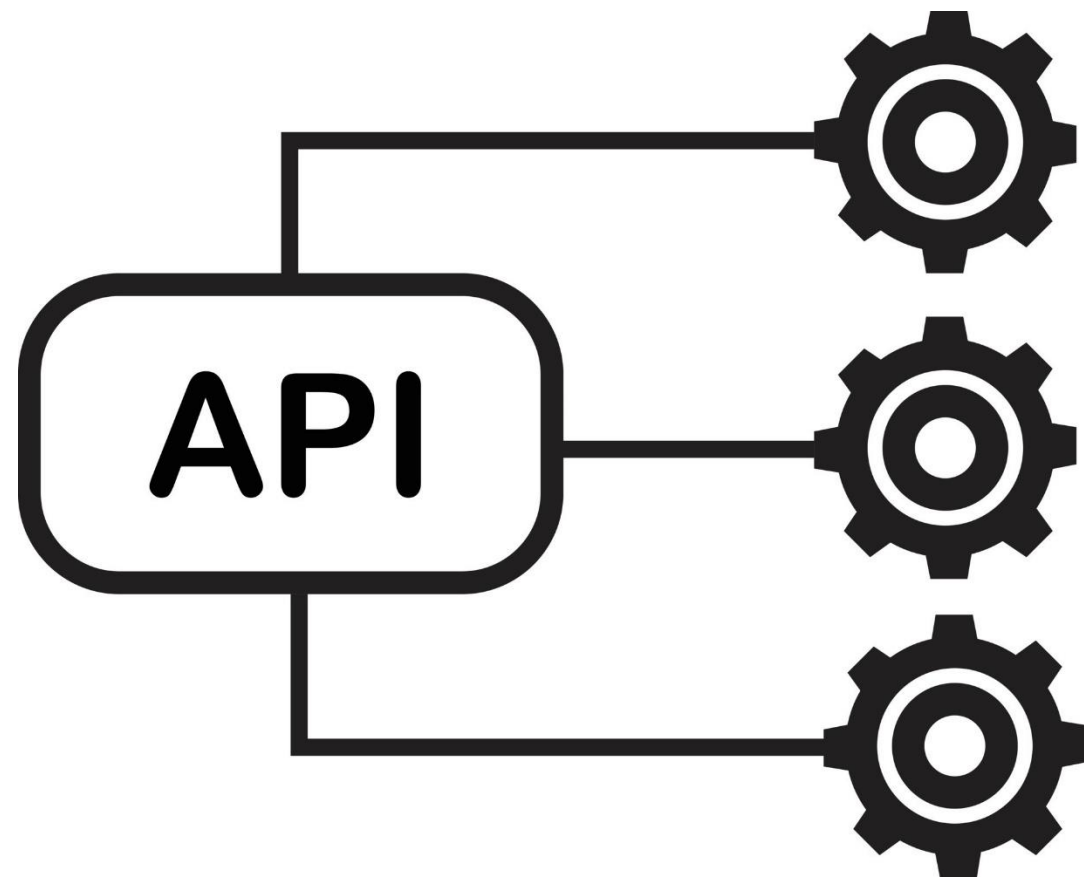
Web API

# Giới thiệu

Application Programming Interface - API



## Giới thiệu

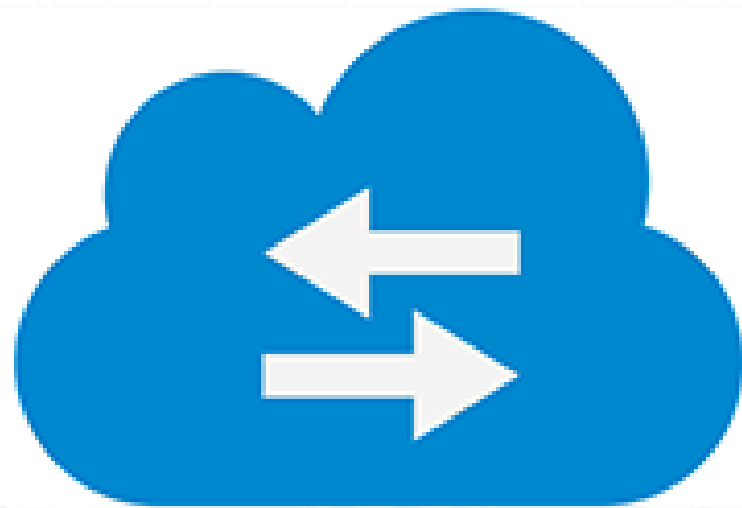


- Tập hợp các quy tắc, giao thức và công cụ cho phép các phần mềm khác tương tác và giao tiếp với nhau
- Cung cấp khả năng truy xuất đến một tập các hàm hay dùng
- Trao đổi dữ liệu giữa các ứng dụng

Application Programming Interface - API



## Phân loại



Web API

- **Web API**

- sử dụng trong các hệ thống website.
- Kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu.
- Các ứng dụng di động cũng sử dụng Web API.

Application Programming Interface - API



## Phân loại



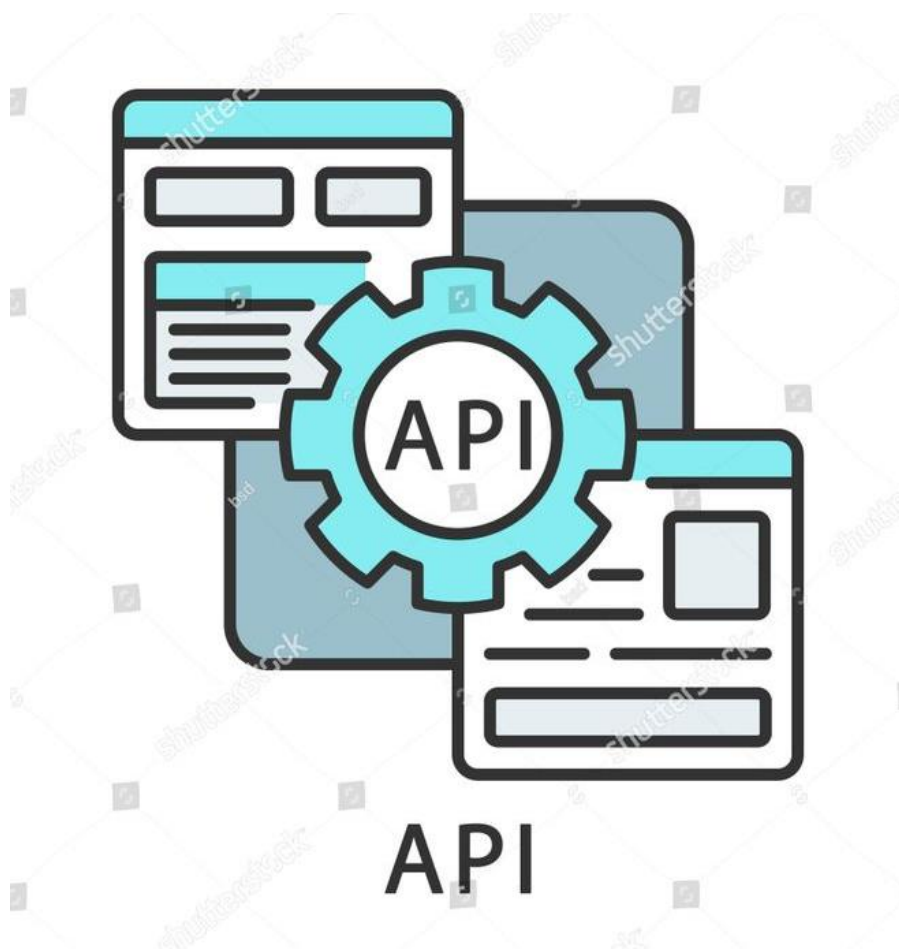
- **API trên hệ điều hành**

- Windows, Linux, Android, IOS có rất nhiều API
- Cung cấp các tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức kết nối với hệ điều hành.
- Lập trình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành.

Application Programming Interface - API



## Phân loại

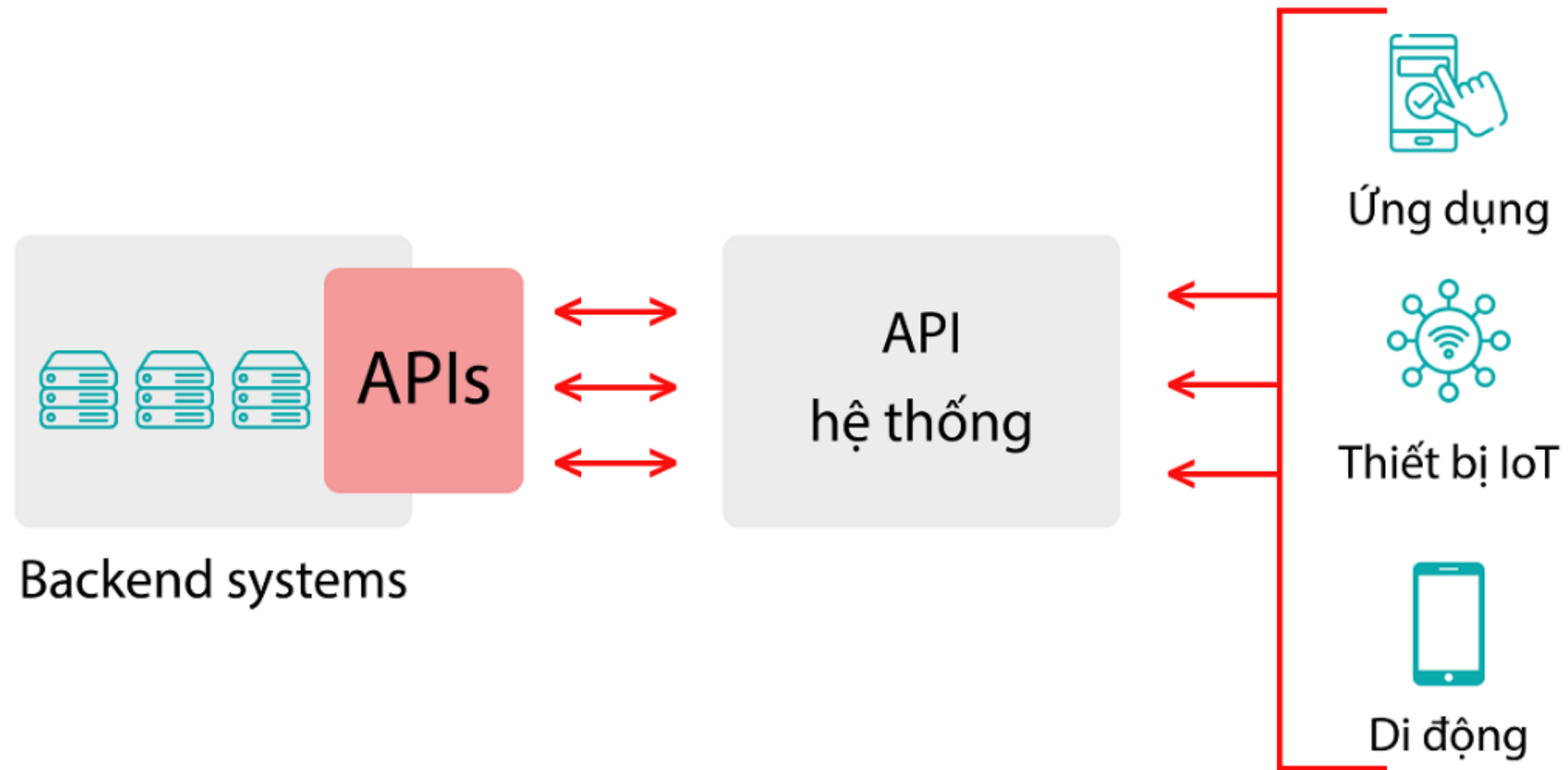


- **API của thư viện phần mềm hay framework:**
- Mô tả và quy định các hành động mong muốn mà các thư viện cung cấp.
- Giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng thư viện được viết bằng ngôn ngữ khác. Ví dụ bạn có thể dùng Php để yêu cầu một thư viện tạo file PDF được viết bằng C++.

Application Programming Interface - API



## Vị trí của API trong hệ thống

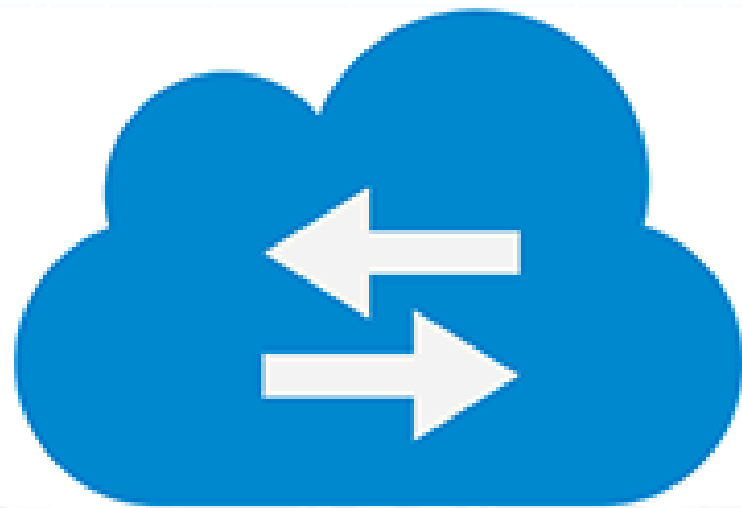


Application Programming Interface - API





## Đặc điểm của Web API



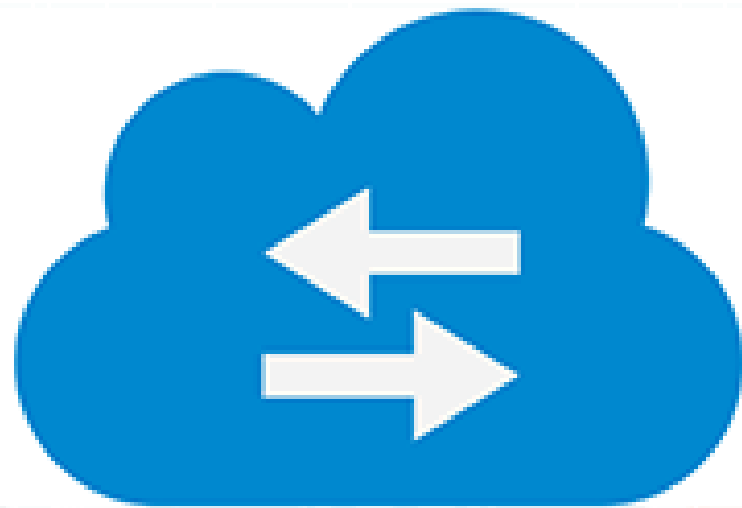
Web API

### • Web API

- Giao tiếp qua mạng, giao thức HTTP hoặc HTTPS
- Có tính độc lập: Nền tảng, ngôn ngữ, hệ điều hành, môi trường,...
- Định dạng dữ liệu: Json, XML,...
- Bảo mật: No Auth, Basic Auth, Token,...
- Mở rộng, tái sử dụng
- Endpoints



## Đặc điểm của Web API



Web API

- **Endpoints**

- URL đích mà các ứng dụng khách gọi yêu cầu HTTP đến
- Mỗi Endpoint đại diện cho 1 chức năng hay 1 tài nguyên cụ thể
- Được xác định bằng các URL (Uniform Resource Locator)
- Có thể có các tham số: `"/users"`, `"/products/{id}"`
- Gắn với các phương thức:
  - GET được sử dụng để lấy thông tin,
  - POST được sử dụng để tạo mới một tài nguyên,
  - PUT được sử dụng để cập nhật tài nguyên,
  - DELETE được sử dụng để xóa tài nguyên

## Đặc điểm của Web API

# API ENDPOINTS

### Fraud Detection:

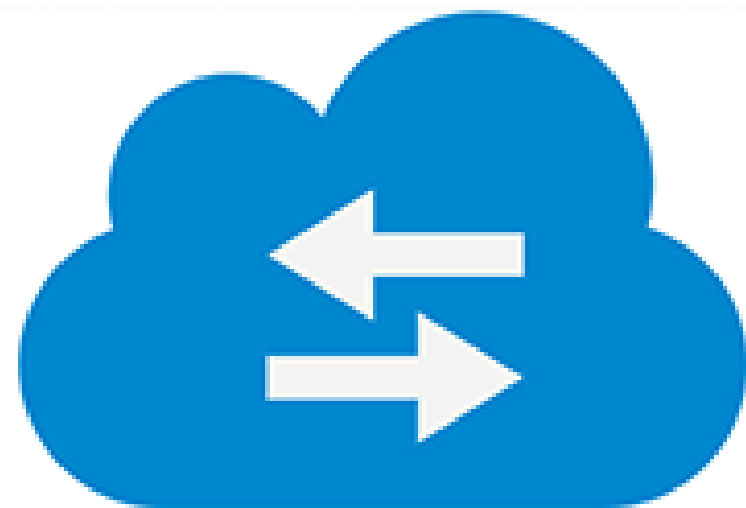
`http(s)://DSS_HOST:PORT/public/api/v*/  
service-name/predict_fraud/predict`

### Loan Default Prediction:

`http(s)://DSS_HOST:PORT/public/api/v*/  
service-name/loan_default/predict`



Banking API Service



Web API

# Tiếp cận

Application Programming Interface - API



## Các phương pháp tiếp cận



- **RESTful API**

- REST (Representational State Transfer) là một kiến trúc phần mềm dựa trên các nguyên tắc và giao thức web tiêu chuẩn.
- Phương thức HTTP: GET, POST, PUT và DELETE
- Định dạng dữ liệu như JSON hoặc XML
- Endpoint là duy nhất



## Các phương pháp tiếp cận

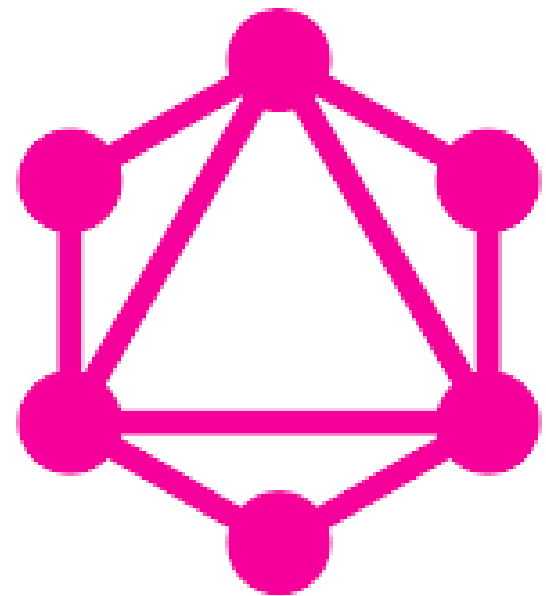
### • SOAP API



- SOAP (Simple Object Access Protocol) là một giao thức truyền tải thông tin giữa các ứng dụng.
- SOAP API sử dụng các thông điệp XML để trao đổi dữ liệu
- Giao thức HTTP hoặc các giao thức khác như SMTP hoặc JMS để truyền tải thông điệp.
- SOAP API có thể định nghĩa các phương thức và phản hồi chi tiết hơn, nhưng cũng có thể phức tạp hơn để triển khai và sử dụng.



## Các phương pháp tiếp cận



GraphQL

- **GraphQL**

- GraphQL là một ngôn ngữ truy vấn và kiến trúc dữ liệu linh hoạt.
- GraphQL cho phép người dùng định rõ các yêu cầu dữ liệu mà họ muốn từ một API và trả về dữ liệu chính xác theo yêu cầu đó (thay vì trả về toàn bộ dữ liệu được định nghĩa trước như REST)
- GraphQL là khả năng tự động tạo tài liệu API. Với GraphQL, bạn có thể tạo ra một tài liệu API tự động và chi tiết, giúp người dùng dễ dàng hiểu và sử dụng API của bạn.



## Các phương pháp tiếp cận

- **gRPC**

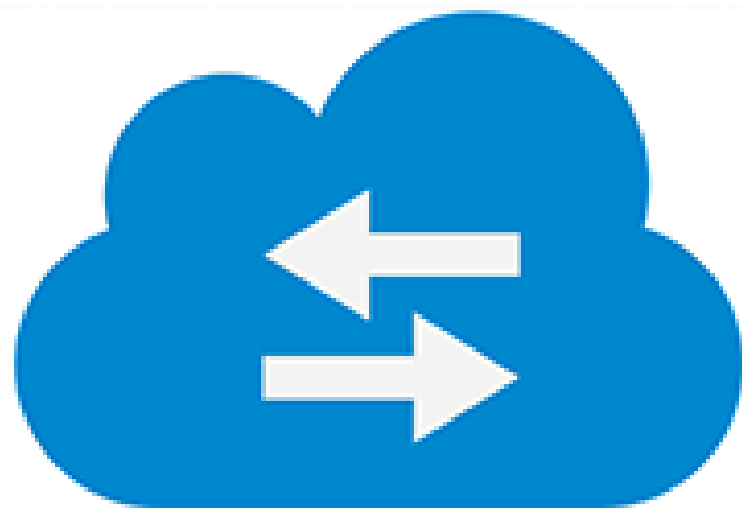
- gRPC là một framework mã nguồn mở do Google phát triển,
- Sử dụng giao thức RPC (Remote Procedure Call) để giao tiếp giữa các ứng dụng khác nhau.
- gRPC hỗ trợ nhiều ngôn ngữ lập trình và sử dụng Protocol Buffers, một định dạng dữ liệu nhị phân, để truyền tải dữ liệu.
- gRPC cung cấp hiệu suất cao và hỗ trợ streaming dữ liệu hai chiều



gRPC







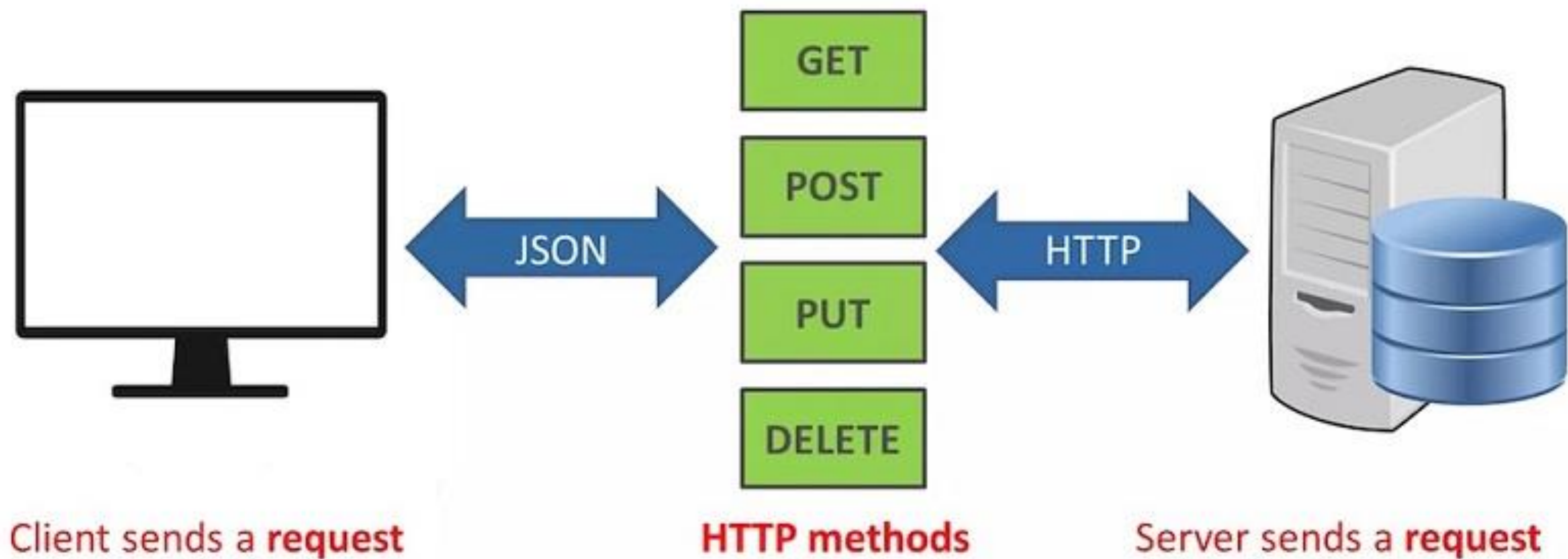
Web API

# Kiến trúc Web API

Application Programming Interface - API



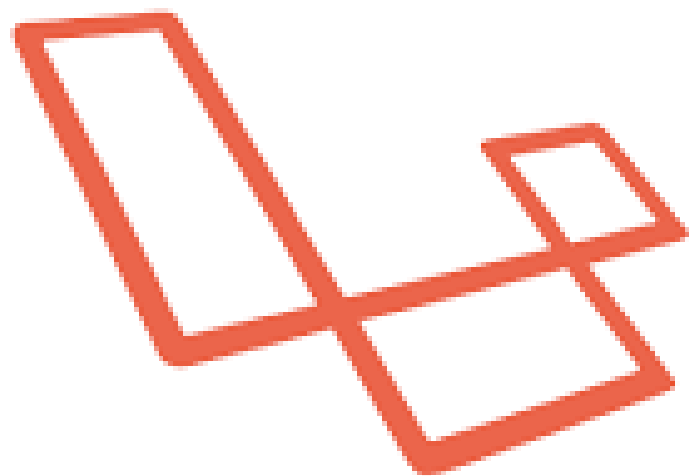
## Kiến trúc Web API



## Kiến trúc Web API

- **Resources**

- Tất cả các thành phần trong hệ thống là các tài nguyên
  - ví dụ: người dùng, sản phẩm, đơn hàng, v.v.
- Mỗi tài nguyên được đại diện bằng một URL riêng biệt



API Resources

{json:api}



## Kiến trúc Web API

### • HTTP Methods

- RESTful API sử dụng các phương thức HTTP để thực hiện các hoạt động trên tài nguyên, bao gồm:
  - GET: Lấy thông tin chi tiết về tài nguyên.
  - POST: Tạo mới tài nguyên.
  - PUT: Cập nhật thông tin của tài nguyên.
  - DELETE: Xóa tài nguyên.



## Kiến trúc Web API

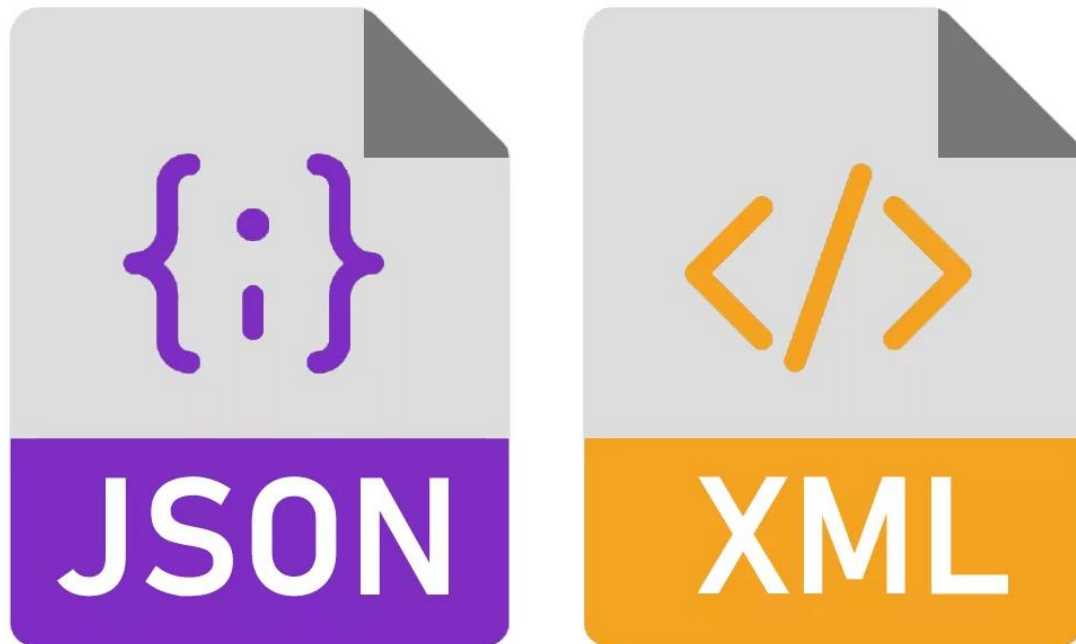
- **Data Formats**

- RESTful API sử dụng các định dạng dữ liệu để truyền tải dữ liệu giữa máy khách và máy chủ như:

- JSON (JavaScript Object Notation);
- XML (eXtensible Markup Language).

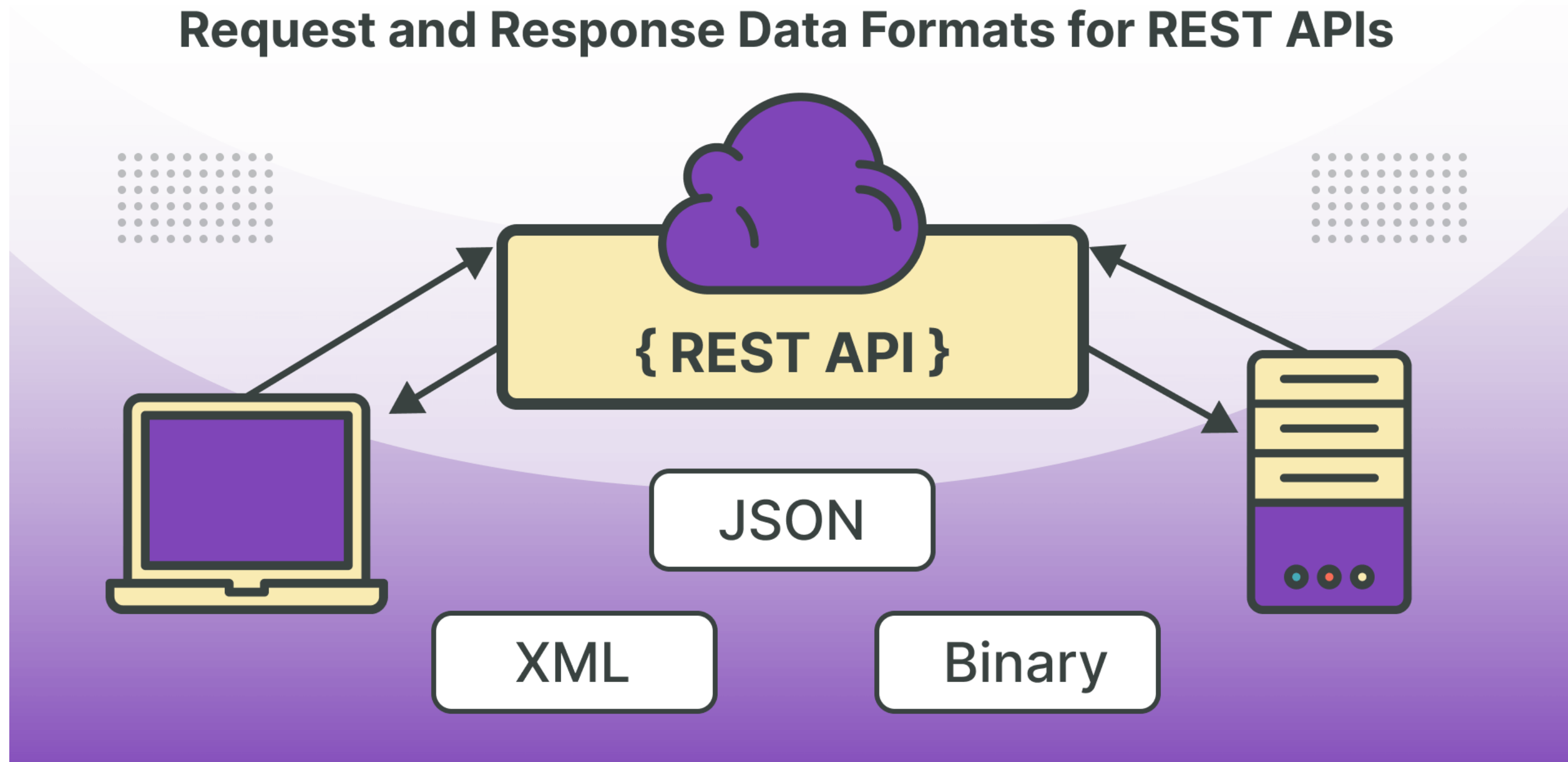
- Ngoài ra:

- HTML
- Hình ảnh
- File
- Âm thanh,...



## Kiến trúc Web API

### Request and Response Data Formats for REST APIs



## Kiến trúc Web API

- **State Management**

- RESTful API không lưu trạng thái (stateless),
- Có nghĩa là mỗi yêu cầu từ máy khách đến máy chủ là độc lập và không phụ thuộc vào các yêu cầu trước đó

# STATE

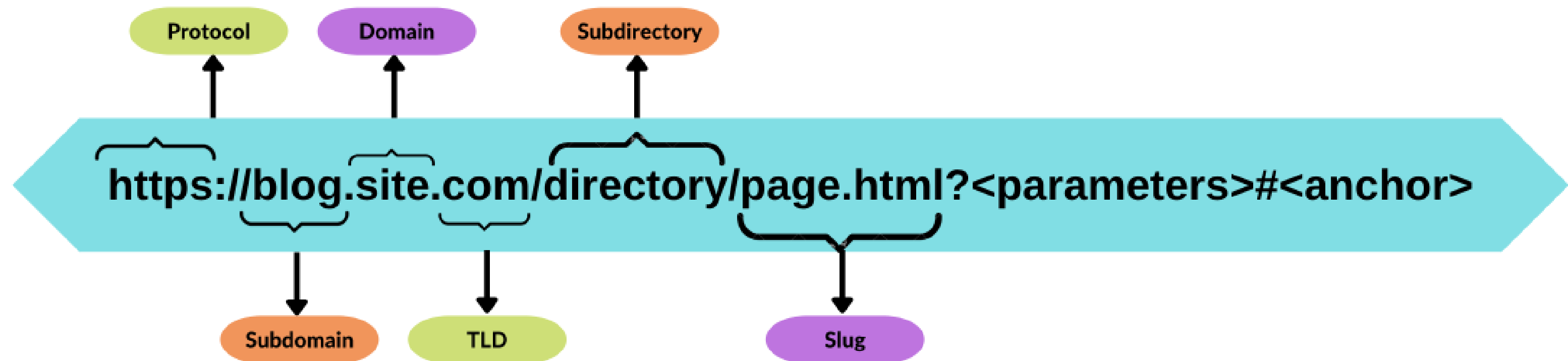


## Kiến trúc Web API

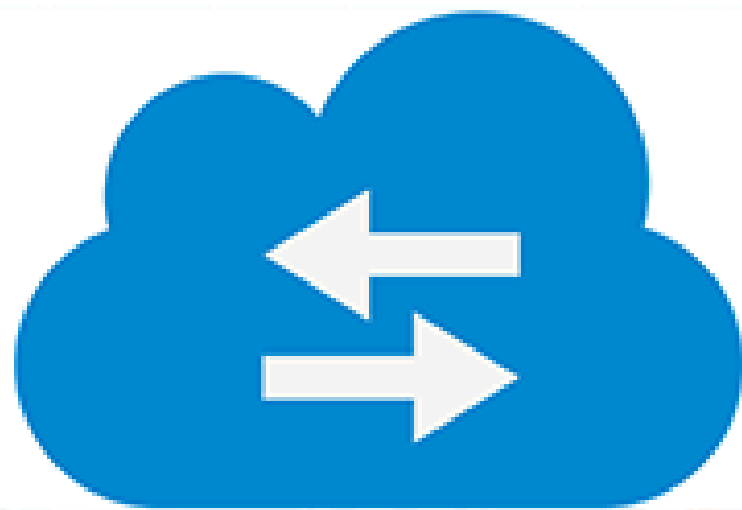


### • URL Structure

- RESTful API sử dụng cấu trúc đường dẫn để định vị tài nguyên và xác định các thao tác trên tài nguyên.







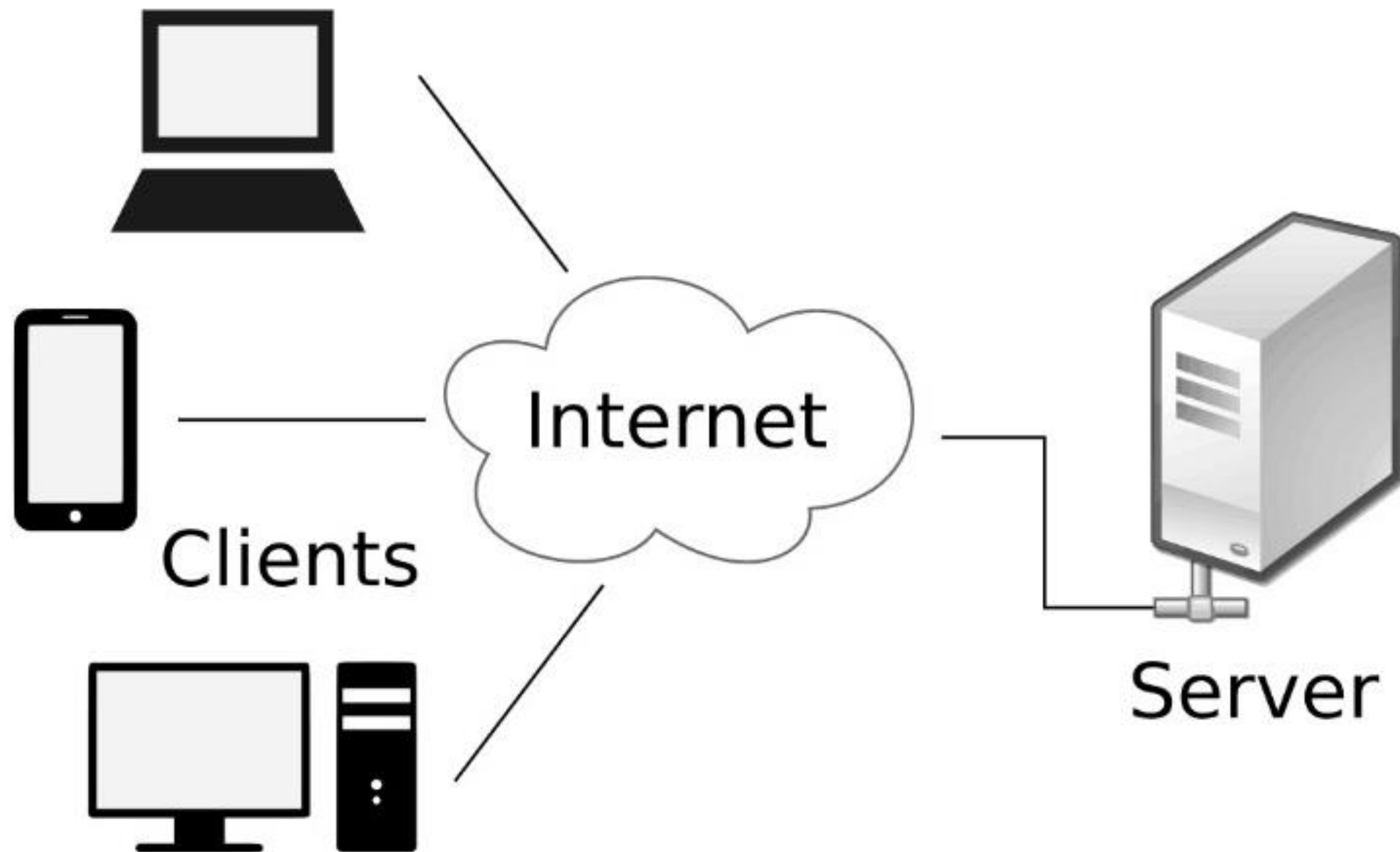
Web API

# Mô hình triển khai

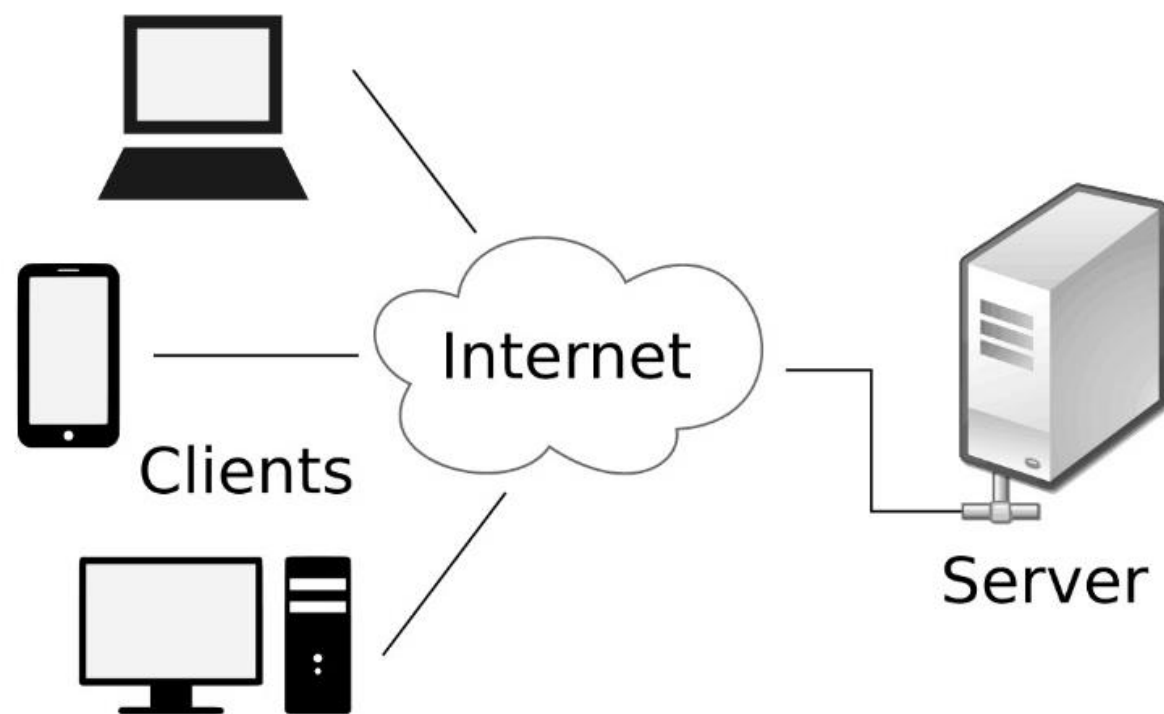
Application Programming Interface - API



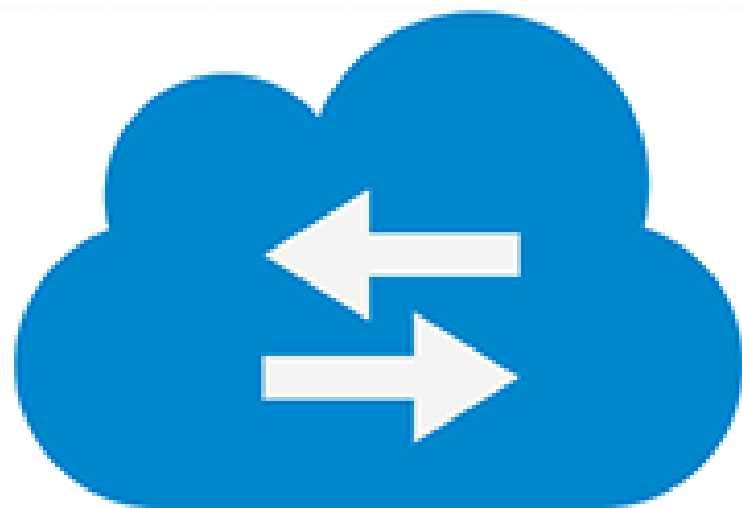
## Mô hình triển khai



## Mô hình triển khai



- **Client** gửi yêu cầu HTTP đến server, sử dụng các phương thức HTTP như GET, POST, PUT, DELETE.
- **Server** nhận và xử lý yêu cầu từ client, truy cập vào cơ sở dữ liệu hoặc các nguồn dữ liệu khác để lấy thông tin cần thiết.
- **Server** xử lý dữ liệu và trả về kết quả cho client, thường dưới dạng định dạng dữ liệu như JSON hoặc XML.
- **Client** nhận kết quả từ máy chủ và hiển thị dữ liệu cho người dùng hoặc tiếp tục thực hiện các thao tác khác.



Web API

# Xây dựng Web API

Application Programming Interface - API



## PHP – SQL / Kết nối

```
<?php
define("PG_DB", "ten_db");
define("PG_HOST", "localhost");
define("PG_USER", "postgres");
define("PG_PORT", "5432");
define("PG_PASS", "*****");
$dbcon=pg_connect("dbname=".PG_DB." user=".PG_USER."
password=".PG_PASS." host=".PG_HOST." port=".PG_PORT);
?>
```

## PHP – SQL / Kết nối

1. Tạo connection (\$dbcon)
2. Tạo lệnh truy vấn
3. Thực thi lệnh truy vấn (pg\_query(\$dbcon,\$truyvan))

## Các lệnh SQL cơ bản

```
SELECT cột  
FROM bảng;
```

```
INSERT INTO bảng ("cột1", "cột 2")  
VALUES ("giá trị cột 1", "giá trị cột 2");
```

## Các lệnh SQL cơ bản

```
//Select
$truyvan='select * from users';
$thuc thi=pg_query($dbcon,$truyvan);

while($kq=pg_fetch_assoc($thuc thi)){
    //var_dump($kq);
    //echo $kq['ten'];
    echo 'Nguoi dung ' . $kq['ten'] . ' co tuoi la
' . $kq['tuoi'];
    echo '<br>';
}
```



## Các lệnh SQL cơ bản

```
//Insert
```

```
$truyvan="insert into users(ten,tuoi)  
values('Khang',30)";
```

```
$thucthi=pg_query($dbcon,$truyvan);
```

## Các lệnh SQL cơ bản

```
UPDATE Bảng  
    SET "cột 1"=?, "cột 2"=?  
WHERE điều kiện;
```

```
DELETE FROM Bảng  
WHERE điều kiện;
```

## Các lệnh SQL cơ bản

```
//Update
```

```
$truyvan="update users set ten='An' where id=16";
```

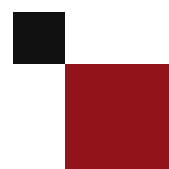
```
$thucthi=pg_query($dbcon,$truyvan);
```

## Các lệnh SQL cơ bản

```
//Delete  
//$truyvan="delete from users where id=15";  
//Xoa het Khang  
$truyvan="delete from users where ten ilike  
'%khang%'";  
$thucthi=pg_query($dbcon,$truyvan);
```



OPENGIS



THANK YOU

