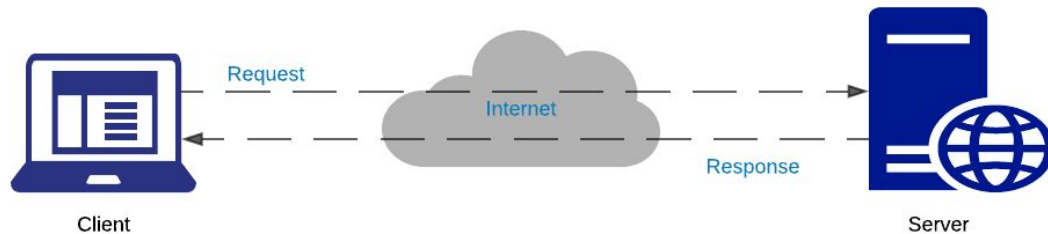


Client-Server Model

Ứng dụng web hoạt động theo mô hình chuẩn client-server. Ý nghĩa của mô hình này là client (máy khách) sẽ gửi request (yêu cầu) tới chủ (máy chủ), server sẽ nhận request để xử lý và trả về (Response) kết quả cho máy khách.



Client - Server giao tiếp với nhau bằng những thông điệp (message), chu kỳ hoạt động của client-server là khoản thời gian gửi thông điệp từ phía client và nhận chờ kết quả trả về từ server.

Để client và server giao tiếp được với nhau, chúng phải có cùng chung một chuẩn(phương thức) liên lạc chung như cách giao tiếp giữa người với người. Chuẩn này trong client-server gọi là giao thức (protocol).

Có nhiều chuẩn giao tiếp khác nhau giữa client và server

- **TCP/IP (Transmission Control Protocol / IP Address)**
Giao thức thiết lập cách kết nối, chia nhỏ các gói dữ liệu và định tuyến các gói dữ liệu đó trên internet để đảm bảo chúng tới được đích.
- **HTTP (HyperText Transfer Protocol)**
Cho phép trao đổi thông tin siêu văn bản (Text, Hình Ảnh, Media...) qua internet.
- **SMTP (Simple Mail Transfer Protocol)**
Cho phép gửi thông điệp điện tử (**Email**) qua internet.
- **POP3 (Post Office Protocol)**
Cho phép nhận các thông điệp điện tử (**Email**) qua internet.

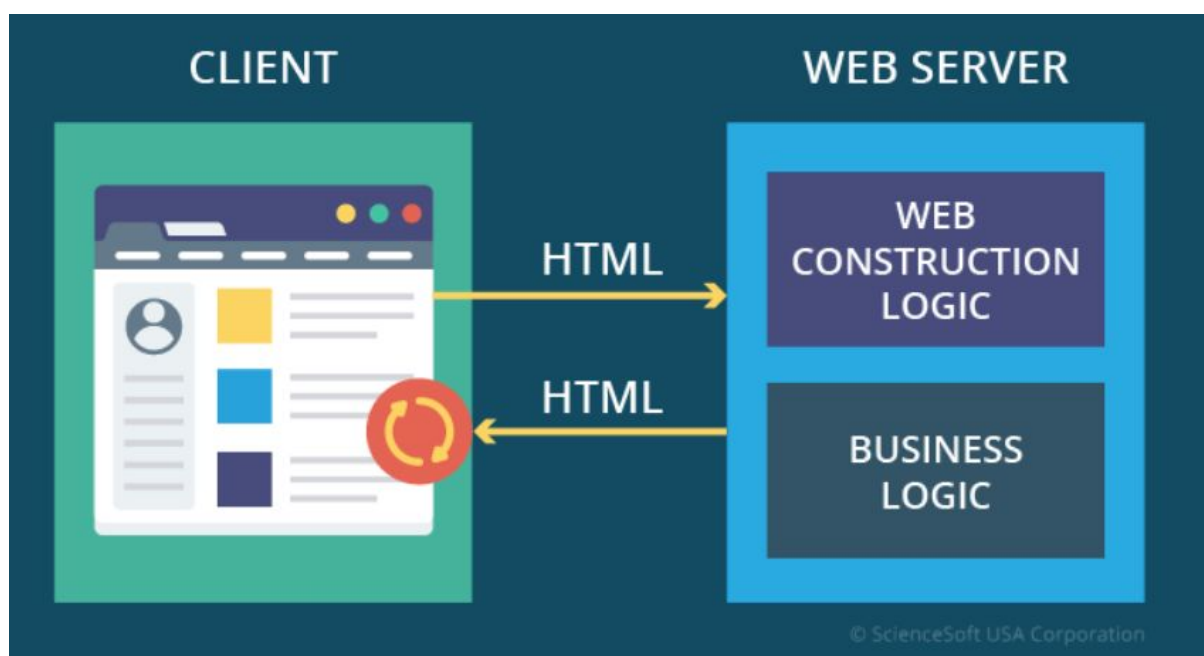
Web Application Architecture

Có nhiều kiến trúc ứng dụng web khác nhau, tùy thuộc vào mục đích và nhu cầu sử dụng mà ta lựa chọn kiến trúc phù hợp để xây dựng hệ thống web.

Kiến trúc ứng dụng web phụ thuộc vào mô hình giao tiếp mà nó sẽ sử dụng, như vậy kiến trúc web là một sơ đồ (Flow Diagram) mô tả dạng khái quát phương thức hoạt động của hệ thống web.

Dưới đây là một vài kiến trúc web được áp dụng.

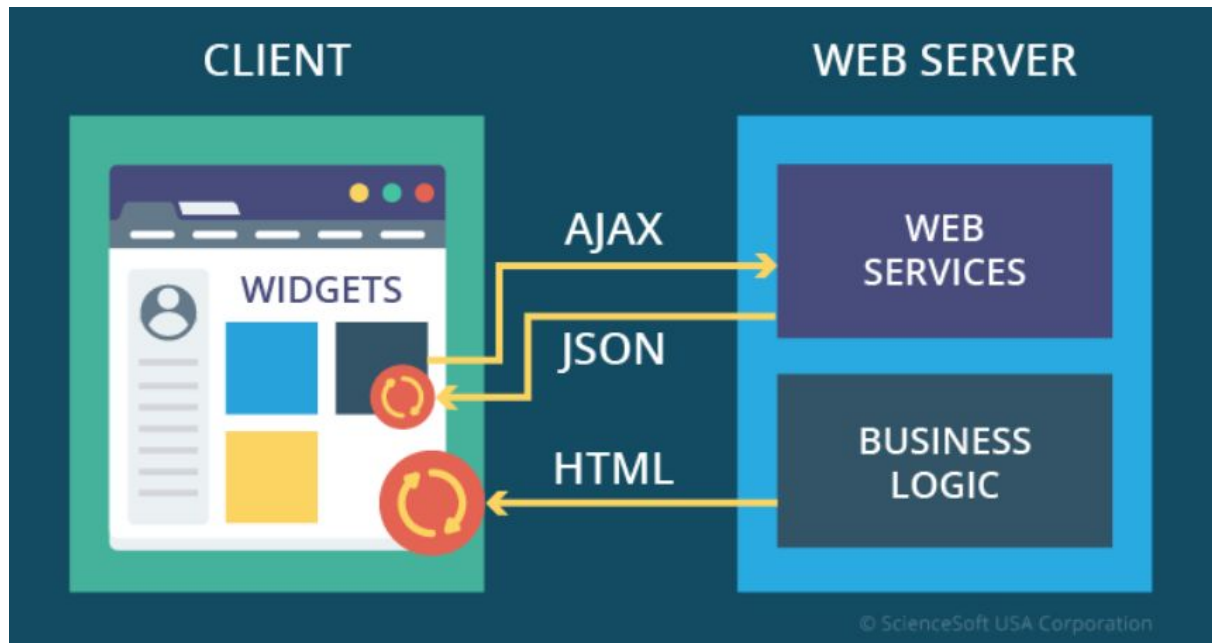
Kiến Trúc HTML Tĩnh



Nguồn: www.scnsoft.com

Theo kiến trúc này, tất cả dữ liệu của ứng dụng web sẽ được lưu trữ dưới dạng file HTML tĩnh trên server. Khi có một yêu cầu truy cập vào trang web, Browser (Client Site) sẽ gửi request truy cập lên web server (Nơi lưu trữ hệ thống web). Server nhận request và trả kết quả về browser dưới dạng HTML, Browser sẽ render kết quả lên màn hình.

Kiến Trúc Widget

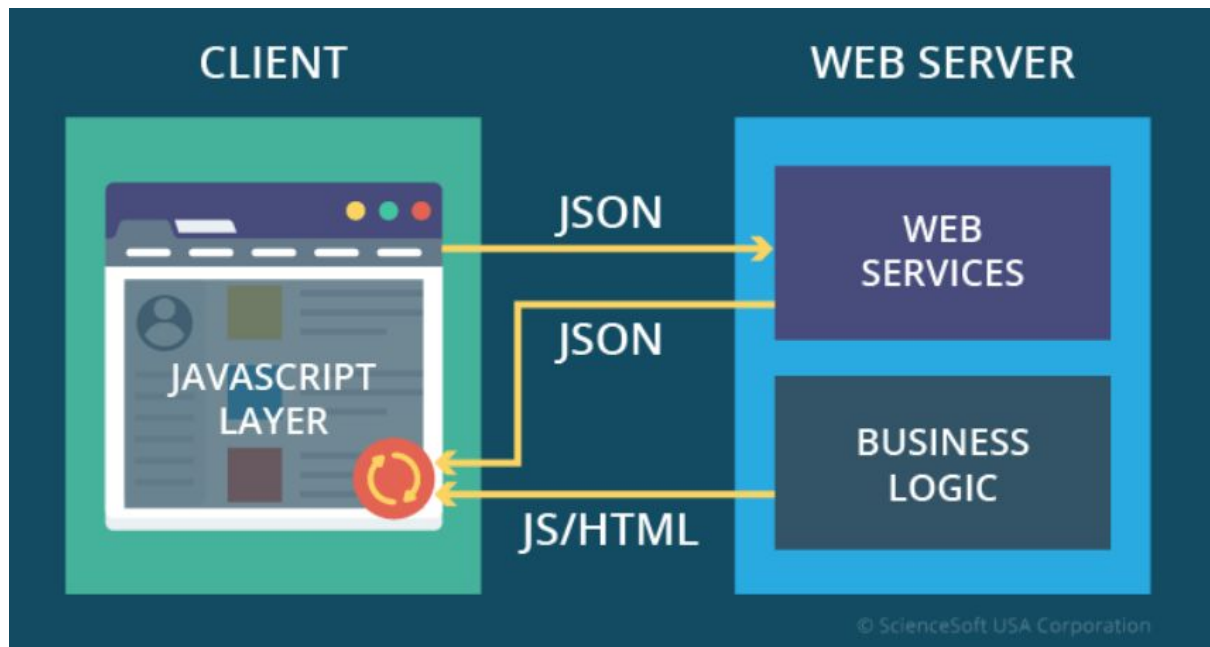


Nguồn: www.scnsoft.com

Đây là dạng kiến trúc tiện ích, Trang web sẽ được tổ chức thành nhiều phần nhỏ được gọi là Widget, Mỗi view sẽ tương tác với server thông qua đối tượng AJAX.

Khi có một yêu cầu truy cập vào trang web, server sẽ trả về kết quả cho Browser dạng HTML, trong HTML có những Widget tương tác với Web Server một cách độc lập.

Kiến Trúc Single-Page



Nguồn: www.scnsoft.com

Đây là dạng kiến trúc phổ biến hiện nay. Hệ thống web được tổ chức theo một trang HTML chính duy nhất, Nội dung trong HTML được làm mới thông qua những kỹ thuật render nội dung của mô hình MVC hay bằng kỹ thuật Widget với đối tượng AJAX.

Trong kiến trúc này, Browser (Client site) sẽ nhận kết quả trả về từ Server với nhiều kiểu dữ liệu khác nhau như Html, Text, JSON... Phụ thuộc vào cách mà ta tổ chức hiển thị mà những kiểu dữ liệu này được sử dụng theo những cách khác nhau.

Web Application Structure

Trong phần về ReactJS, chúng ta sẽ xây dựng một ứng dụng web client được tổ chức theo cấu trúc thư mục tạo bởi công cụ create-react-app.

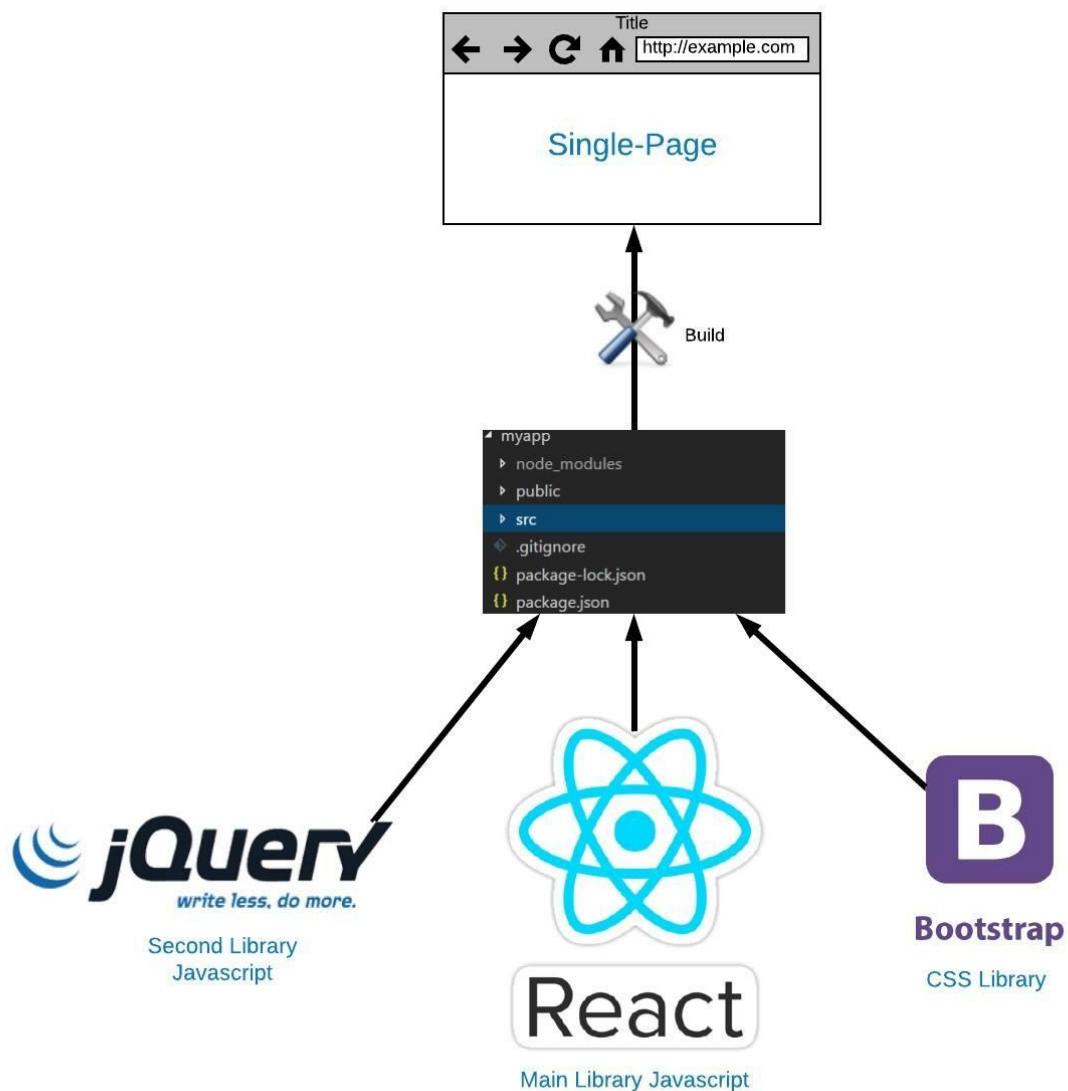
Thư viện Javascript chính được dùng để xây dựng ứng dụng là ReactJS. Có 2 thư viện phụ:

1. JQuery

Thư viện JQuery phụ trợ thư viện ReactJS tương tác trực tiếp với DOM HTML.

2. Bootstrap

Thư viện CSS HTML, dùng để style cho giao diện client của trang web.



References

1. https://vi.wikipedia.org/wiki/Giao_thức_truyền_thông
2. <https://www.scnsoft.com/blog/web-application-architecture>