

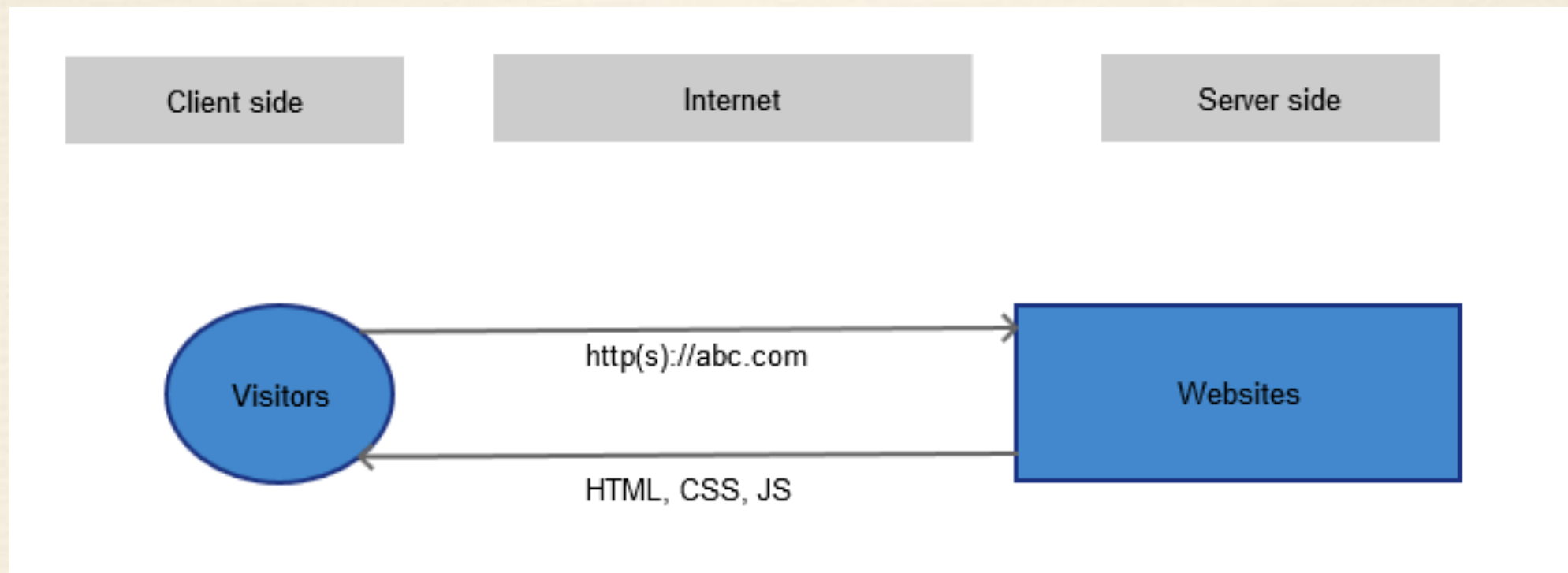
EasyWeb

A revolution framework for websites



Trịnh Ngọc Quốc Bảo
Founder

How a website works



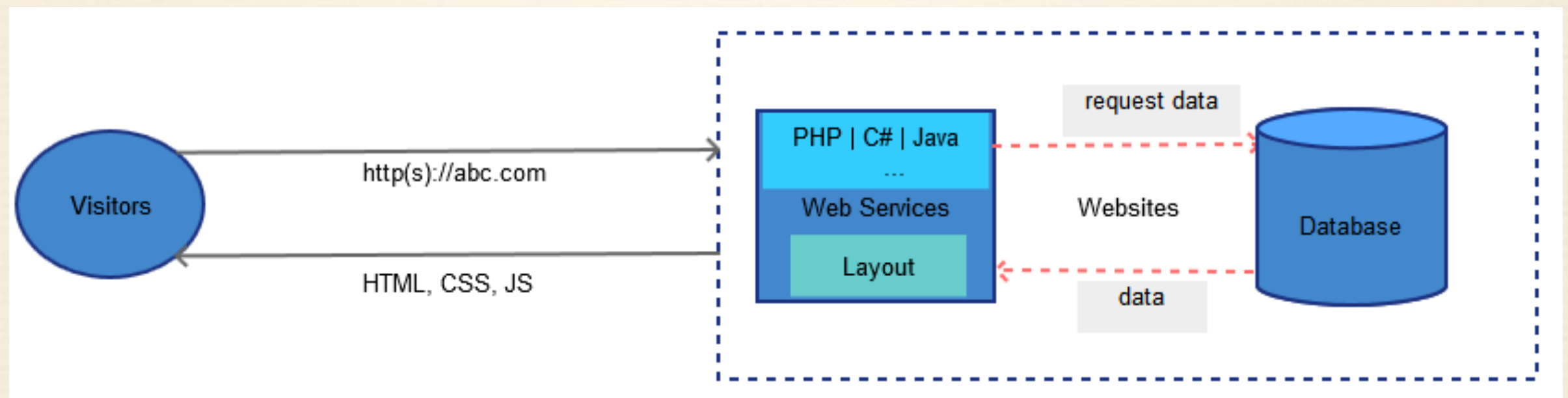
HTML “describes the structure of a web page semantically “

CSS: “describes the presentation of a document written in a markup language... “

HTML, CSS, JS “form a triad of cornerstone technologies for the World Wide Web”
theo Wikipedia

(*) ngoài ra còn hình ảnh, video, pdf, ...

Dynamic websites: > 90% websites world wide



2. Repeat the process for each view

1. Database

3. Server programming languages

1. Database

❖ Nguồn lực

- ❖ Nhiều nguồn lực khi xây dựng website: thiết kế database, lập trình và triển khai hệ thống, cũng như điều chỉnh, bổ sung tính năng mới sau này
- ❖ bảo đảm hệ thống database hoạt động ổn định, mở rộng khi website có nhiều người truy cập. Nhân sự liên tục theo dõi, bảo trì hệ thống, cũng như cung cấp tài nguyên servers để hoạt động. Những website lớn đòi hỏi database phải được chạy trên nhiều servers song song để bảo đảm 24/7 cũng như thường xuyên backup để phòng chống sự cố, hoặc tấn công từ bên ngoài
- ❖ Đòi hỏi nhân sự chuyên môn trong thiết kế và vận hành

❖ Ảnh hưởng

- ❖ Với các websites vừa và nhỏ, database gây trở ngại rất lớn trong việc đơn giản nhanh chóng khi triển khai mới.
- ❖ Và trong quá trình mở rộng, phát triển sau này, tốn rất nhiều nguồn lực cho việc nâng cấp, tối ưu và nhiều trường hợp không thể đáp ứng phải chuyển sang hệ thống database mới

2. Repeat the process for each view

- ❖ Vấn đề:

- ❖ **Tốc độ phản hồi:** với mỗi lượt truy cập, servers phải xử lý quá trình kết nối, lấy dữ liệu từ database, xử lý và trả ra HTML, CSS, JS cho end-users.
- ❖ **Khả năng xử lý:** trong 1s, servers phải xử lý hàng ngàn, thậm chí là hàng triệu truy cập với các websites lớn, nhưng kết quả HTML, CSS, JS trả ra hầu hết giống nhau.

- ❖ Giải pháp: caching HTML, CSS

- ❖ lưu lại kết quả HTML, CSS, JS trong lần xử lý đầu tiên và trả ra cho các lần truy cập sau đó.
- ❖ Hệ thống “caching” cũng phải có cơ chế xóa, “clear cache” và cập nhật lại kết quả khi có thay đổi trong database.

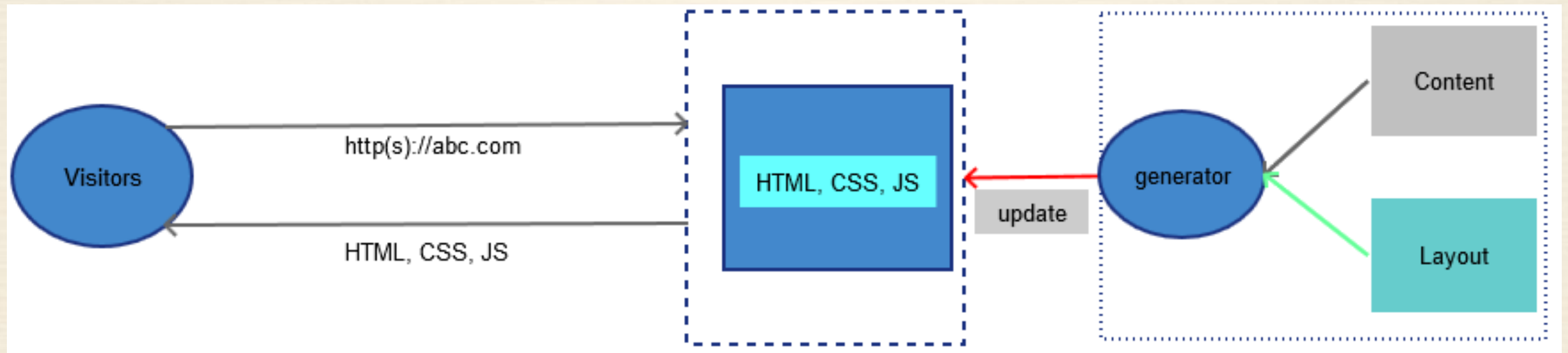
Caching => hệ thống phức tạp

3. Server programming languages

- ❖ Programming skills:
 - ❖ PHP, C#, Java, ...
 - ❖ Object oriented , MVC, ...
- ❖ Framework skills:
 - ❖ Application framework: zend fw, yii, asp.net mvc, spring, ...
- ❖ Open sources:
 - ❖ cần nhiều thời gian đọc tài liệu và vọc code để hiểu rõ, do tài liệu thường không đầy đủ, chính xác.
 - ❖ Để hiểu rõ, thành thạo, một cá nhân không đủ mà phải là 1 team
 - ❖ Rất khó khăn để điều chỉnh theo yêu cầu đặc thù

**Không thể với người không chuyên,
quá phức tạp với người mới học lập trình**

EasyWeb, static websites



Only accepts and returns, no processing

No database,
no server languages

Update when content changes

One update for thousands, millions views

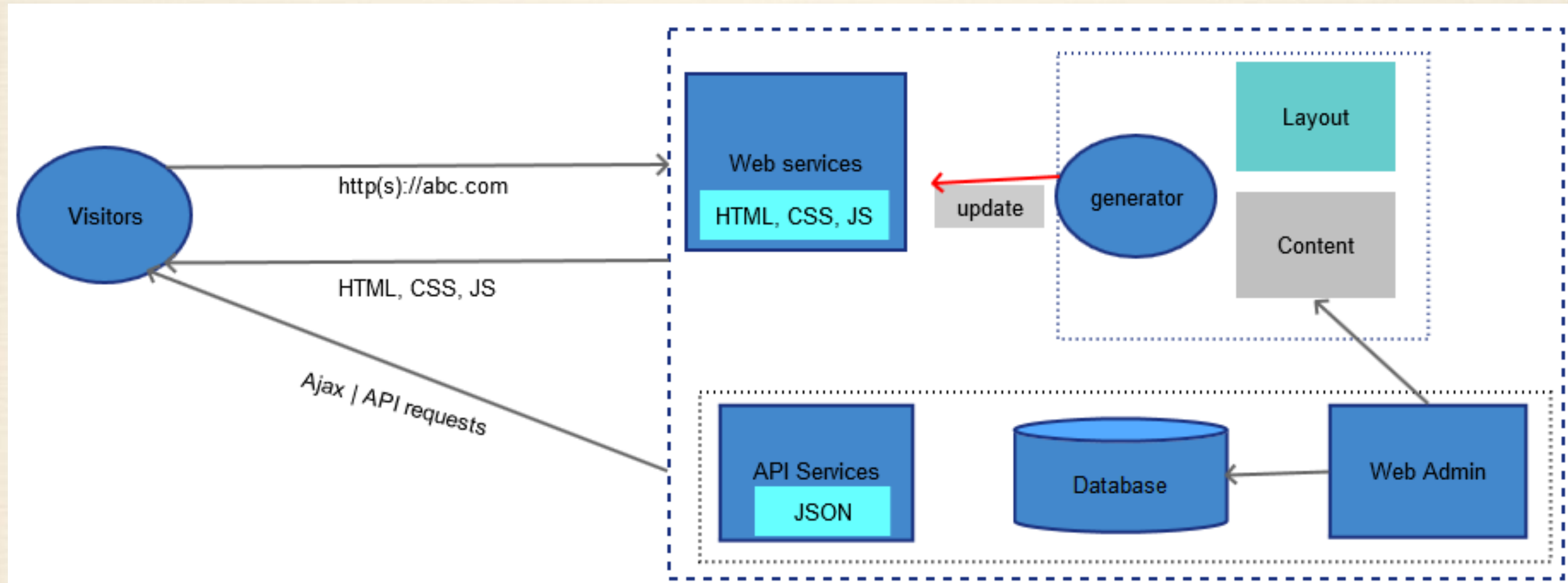
Sự tối ưu của static websites

- ❖ Loại bỏ các phần phức tạp => dễ dàng với những người không chuyên
 - ❖ Hệ thống Database của website
 - ❖ Không cần kĩ năng về server programming languages
 - ❖ Chỉ cần kĩ năng cơ bản về HTML, CSS (và JS)
- ❖ Tối ưu cho triển khai, vận hành, bảo mật
 - ❖ Nguồn lực khi xây dựng, nhất là mở rộng sau này
 - ❖ Chi phí vận hành:
 - ❖ Hạ tầng: không cần nhiều CPU, ưu tiên SSD (HDD) và RAM
 - ❖ Nhân sự: bảo trì, giám sát cũng như bảo mật hệ thống
 - ❖ Tốc độ của websites:
 - ❖ Bản thân websites là hệ thống caching tối ưu
 - ❖ đáp ứng traffic lớn với hạ tầng tối thiểu

EasyWeb, static websites

- ❖ Không có dữ liệu động (dynamic content)
 - ❖ Dữ liệu dành riêng theo đối tượng người dùng
 - ❖ Profile thông tin users
 - ❖ Thông tin được phân loại theo users
 - ❖ Không lưu được thông tin tương tác của người dùng trong quá trình vận hành: comments, submit thông tin, ...
 - ❖ Dữ liệu thay đổi liên tục, tự động bởi hệ thống
- ❖ Tối ưu cho các website đơn giản:
 - ❖ Giới thiệu thông tin: công ty, sản phẩm
 - ❖ Trang blog : cá nhân công ty
 - ❖ Trang tin tức, ...

EasyWeb with dynamic content



Dynamic content: API microservices

Static/Dynamic content requests > 90%

dynamic API services

- ❖ Sử dụng chuẩn Restful API theo mô hình Micro Services
 - ❖ json format cho API input và output
 - ❖ Micro services cho phép tách biệt theo module tính năng
- ❖ Tất cả back-end framework: php, .net, java,... đều dễ dàng chuyển đổi sang API services
- ❖ Dễ dàng mở rộng tính năng bằng cách tích hợp các API với nhau
 - ❖ API chuyển đổi từ hệ thống đang có hoặc viết mới theo yêu cầu
 - ❖ API cung cấp sẵn bởi EasyWeb hoặc các đối tác

Mọi websites đều dễ dàng chuyển sang mô hình EasyWeb

- ❖ EasyWeb cho phép xây dựng mọi loại website,
 - ❖ cơ bản như đã liệt kê phần statics
 - ❖ phức tạp với nhiều tương tác như: ecommerce, ứng dụng quản lý nền web, ...
- ❖ Dễ dàng chuyển đổi các websites đang có
 - ❖ Static: sử dụng Easyweb
 - ❖ Dynamic: dễ dàng chuyển đổi từ hệ thống đang có hoặc tích hợp, sử dụng các API do EasyWeb cung cấp

Lợi ích khi chuyển sang EasyWeb

1. Chuyển đổi dễ dàng, nhanh chóng (so với xây dựng nền tảng mới)
 - ❖ Giữ nguyên giao diện website hiện tại
 - ❖ Kế thừa toàn bộ các tính năng và dữ liệu đang có
2. Tối ưu chi phí hoạt động, nhất là khi website phát triển, traffic tăng
 - ❖ hạ tầng dành cho vận hành
 - ❖ nhân sự cho bảo trì, quản trị hệ thống
3. Dễ dàng mở rộng, phát triển tính năng sau này
 - ❖ Tối ưu cho phát triển mới: nguồn lực và thời gian
 - ❖ tích hợp các services có sẵn

EasyWeb technologies

- ❖ NoSQL: Content Model (replaced for database)
- ❖ Static page Generator
- ❖ Micro services: dynamic API services

(*) <https://github.com/easywebhub/easyapp/wiki/EasyWeb,-a--revolution-framework-of-future-websites>

Content Model, json schema

- ❖ Với Content Model, chúng ta dễ dàng, nhanh chóng thiết kế, quản lý thông tin của websites thay vì can thiệp vào hệ thống database như cách cũ.
- ❖ Tùy ý định nghĩa cấu trúc thông tin (json schema)
 - ❖ Thêm mới trường thông tin, kiểu hiển thị
 - ❖ Cấu trúc thông tin phức tạp cho bất kì loại websites nào
- ❖ Dễ dàng nhập dữ liệu cho từng trang
 - ❖ Sử dụng form nhập liệu đơn giản
 - ❖ Cho phép tùy chỉnh, thay đổi form hiển thị cho thân thiện với người dùng

EasyBuilder, static page Generator

- ❖ EasyBuilder ngoài vai trò là Static Page Generator còn có các vai trò
 - ❖ Content builder: định nghĩa Content Model
 - ❖ Content Management System (CMS) hoàn chỉnh: cập nhật nội dung
 - ❖ Local Preview: xem website hoạt động trên máy tính cá nhân
- ❖ EasyBuilder cho phép thay đổi bất kỳ nội dung hay giao diện của websites một cách trực quan nhanh chóng mà chỉ cần kiến thức về HTML, CSS

Dynamic API: micro-services

- ❖ EasyWeb cung cấp API cho hầu hết các tính năng phổ biến
 - ❖ Comments, live chat, private chat,...
 - ❖ ecommerce modules (shopping cart, email marketing, ...)
 - ❖ users account of websites
 - ❖ tích hợp FB, GA, ...
- ❖ Tính năng đặc thù theo yêu cầu
 - ❖ Phát triển bởi đội ngũ in-house
 - ❖ Phối hợp với đối tác của EasyWeb để thực hiện

Three big advantages

1. Convert millions old websites to EasyWeb
 - ❖ Don't care old framework: PHP, C#, Java,....
 - ❖ Possible and low cost of converting
 - ❖ Low cost for operating and mantanance
2. Prototype: an efficient approach to build
 - ❖ Similar the real website
 - ❖ Build fast with low cost
 - ❖ Reuse for building real website
3. Low cost for websites developing, operating and mantanances

Nổi bật của EasyWeb

- ❖ Chưa có nền tảng nào cho phép chuyển đổi như EasyWeb
 - ❖ Giữ nguyên giao diện website đang có
 - ❖ giữ nguyên tính năng hiện có bằng cách chuyển đổi (thay vì phát triển lại)
- ❖ Hệ thống đơn giản sau khi chuyển đổi rất nhiều
 - ❖ đơn giản tối ưu trong vận hành, quản trị
 - ❖ dễ dàng, nhanh chóng trong việc mở rộng, bổ sung sau này

Nâng tầm Prototype

- ❖ Với EasyWeb, Prototype đã trở thành 1 phần quan trọng trong quá trình xây dựng websites
 - ❖ Xây dựng Prototype dễ dàng, nhanh chóng chỉ với kiến thức về HTML, CSS
 - ❖ Prototype hoạt động như hệ thống thật và cho phép dễ dàng tùy chỉnh theo yêu cầu
 - ❖ Tái sử dụng hoàn toàn cho websites hoàn chỉnh
- ❖ Cho phép xây dựng Prototype cho các hệ thống có nhiều tương tác phức tạp
 - ❖ Phần quản trị AdminCP
 - ❖ Hệ thống ứng dụng trên nền web
 - ❖ Hỗ trợ responsive theo từng thiết bị

Easy Market

- ❖ Là mô hình hợp tác giữa EasyWeb và cộng đồng front-end developers hoặc designers
- ❖ Mô hình kinh doanh của Easy Market
 - ❖ Chứa danh sách websites mẫu cho mọi loại nhu cầu
 - ❖ Bán các API services dùng chung cho khách hàng
 - ❖ Cung cấp dịch vụ điều chỉnh, bổ sung tính năng theo yêu cầu trong suốt vòng đời của websites

EasyWeb cần bạn tham gia

- ❖ Người sử dụng: đáp ứng nhu cầu
- ❖ Đối tác: cùng hợp tác và kinh doanh
- ❖ Phát triển để EasyWeb ngày càng áp dụng rộng rãi và phổ biến.

Người sử dụng

1. Lựa chọn website mẫu ưng ý và triển khai theo nhu cầu của mình
 - ❖ Biết sử dụng EasyBuilder
2. Xây dựng websites bằng EasyWeb từ template có sẵn hoặc tự viết mới
 - ❖ Chỉ cần biết HTML, CSS, (JS càng tốt)
 - ❖ được training hoặc tự đọc tài liệu của EasyWeb

Đối tác của EasyWeb

- ❖ Xây dựng websites mẫu và dịch vụ mở rộng sau này (designers & front-end devs)
- ❖ API services tích hợp vào websites (back-end devs)
 - ❖ tính năng dùng cho web front & web admin
- ❖ Sử dụng công nghệ EasyWeb để phát triển websites cho nhu cầu riêng
 - ❖ phát triển các websites nội bộ
 - ❖ websites cho khách hàng riêng của đối tác
- ❖ Cùng tham gia dự án của EasyWeb

Phát triển EasyWeb

- ❖ Sử dụng và đóng góp ý kiến, đưa ra nhu cầu sử dụng
- ❖ Giới thiệu, quảng bá EasyWeb
 - ❖ bài viết giới thiệu, tài liệu hướng dẫn
 - ❖ quảng bá đến mọi người
- ❖ Tham gia phát triển EasyWeb (freelancer/full-time)
 - ❖ EasyBuilder: js, nodejs, handlebarJS, electron, ...
 - ❖ EasyAdminCP: Aurelia, Couchbase, .Net Restful API,...
 - ❖ Hệ thống: lập trình back-end, auto scripts trên linux, cho nginx, dockers, ...

Demo EasyWeb

- ❖ Giới thiệu Easy Builder
 - ❖ Free account, EasyMarket, Manage& Preview, Sync&Deploy
 - ❖ Xây dựng websites từ mẫu của Easy Market
- ❖ Điều chỉnh theo nhu cầu (mức cơ bản)
 - ❖ bổ sung trang mới từ layout html, css có sẵn
 - ❖ định nghĩa cấu trúc thông tin (Content Model)
 - ❖ binding data cơ bản
 - ❖ block dùng chung: partial layout & metadata