Front-end Development

Lecturer: Ung Văn Giàu
Email: giau.ung@eiu.edu.vn

Web Technologies Basics

Concepts

Contents

01 **Web Page, Web Site and Web Application**

02 **Web Browser and Layout Engine**

03 **Server**

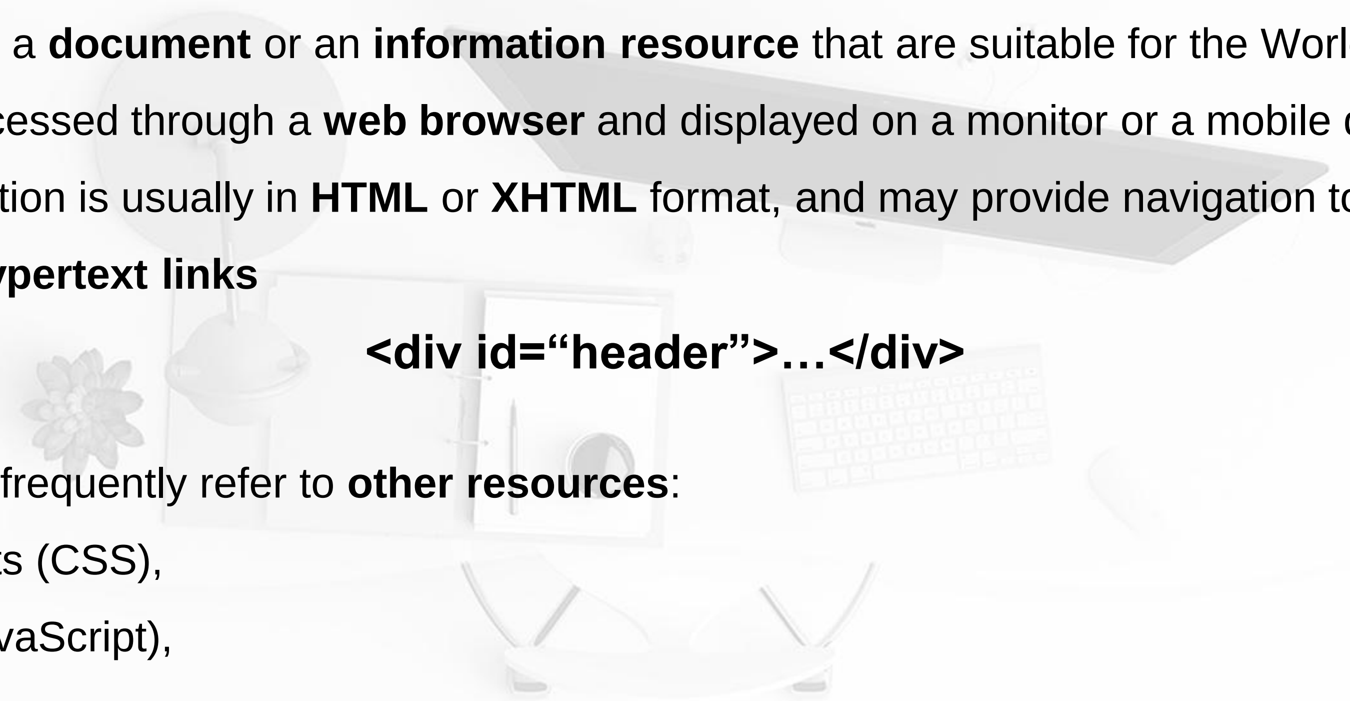
04 **HTTP and HTTPS**

05 **Software Architecture**

06 **Cloud**

1. Web Page, Web Site and Web Application

Web Page

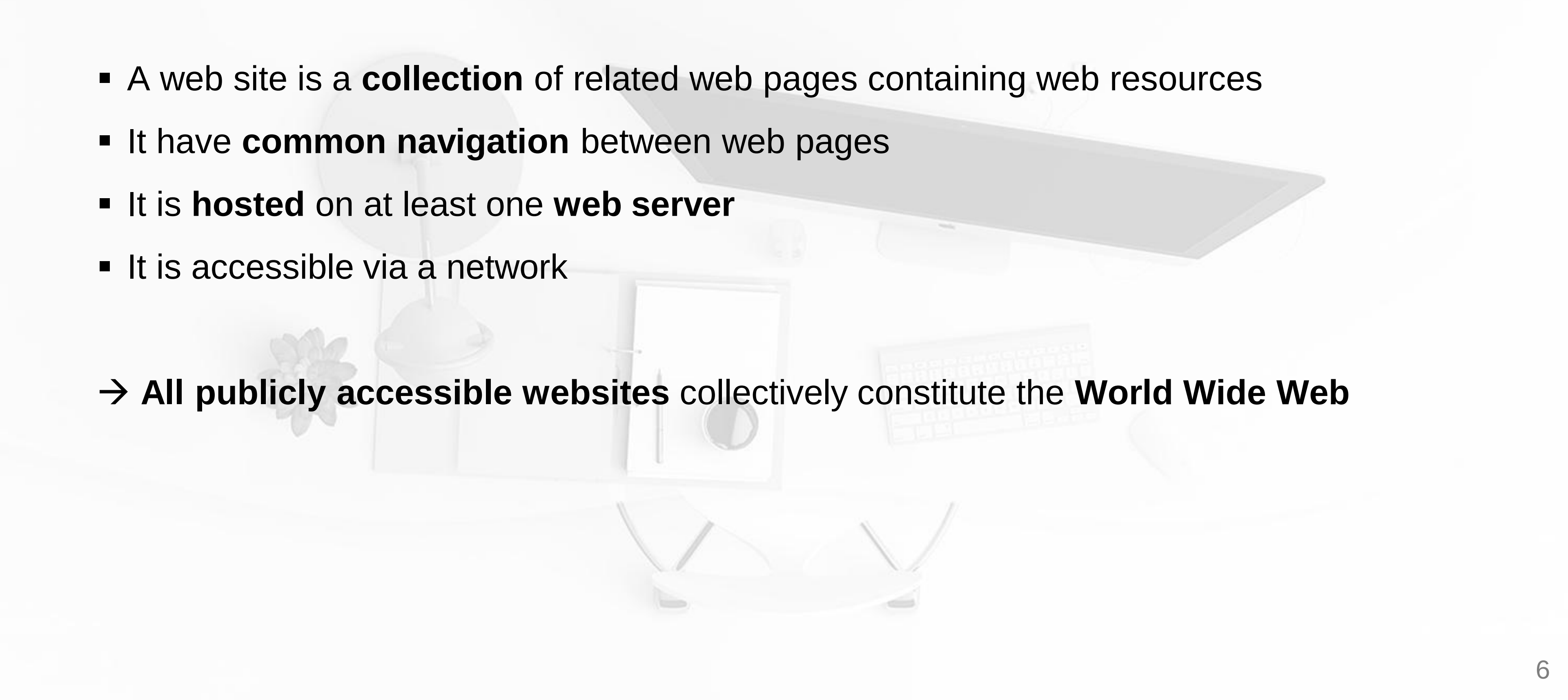


- Web page is a **document** or an **information resource** that are suitable for the World Wide Web
- It can be accessed through a **web browser** and displayed on a monitor or a mobile device
- This information is usually in **HTML** or **XHTML** format, and may provide navigation to other web pages via **hypertext links**

<div id=“header”>…</div>

- A web page frequently refer to **other resources**:
  - style sheets (CSS),
  - scripts (JavaScript),
  - and images

Web Site



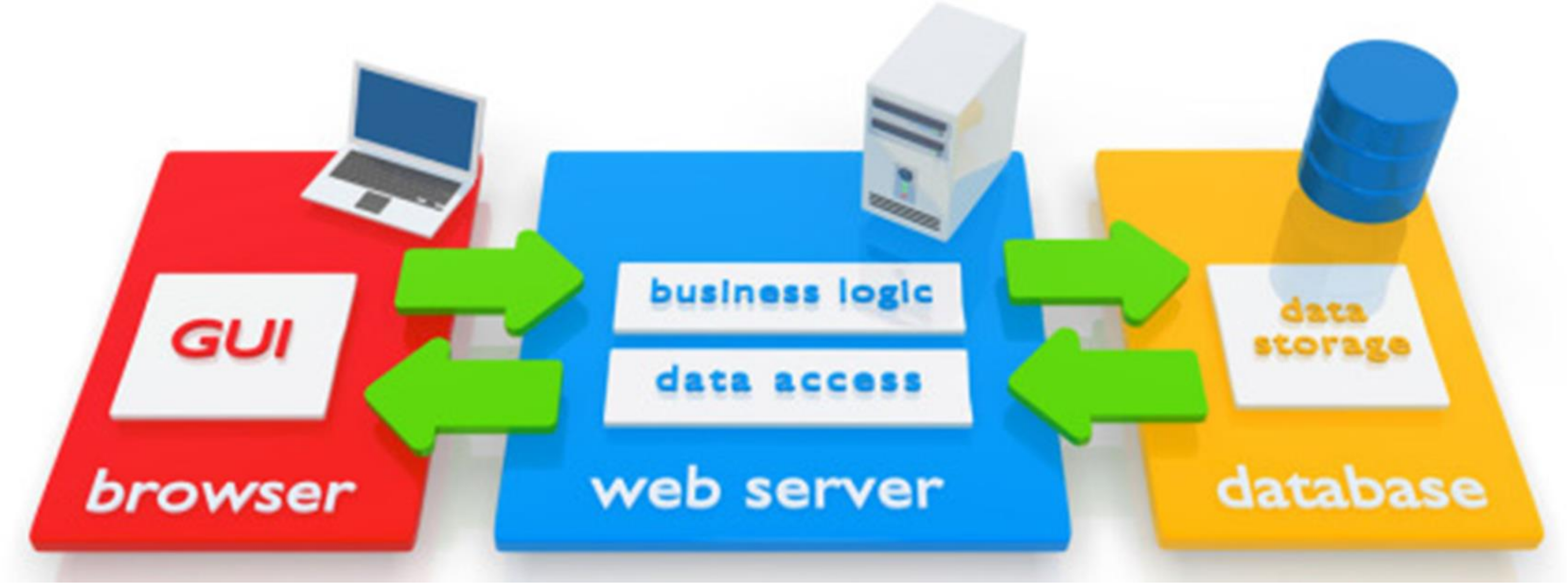
- A web site is a **collection** of related web pages containing web resources
- It have **common navigation** between web pages
- It is **hosted** on at least one **web server**
- It is accessible via a network

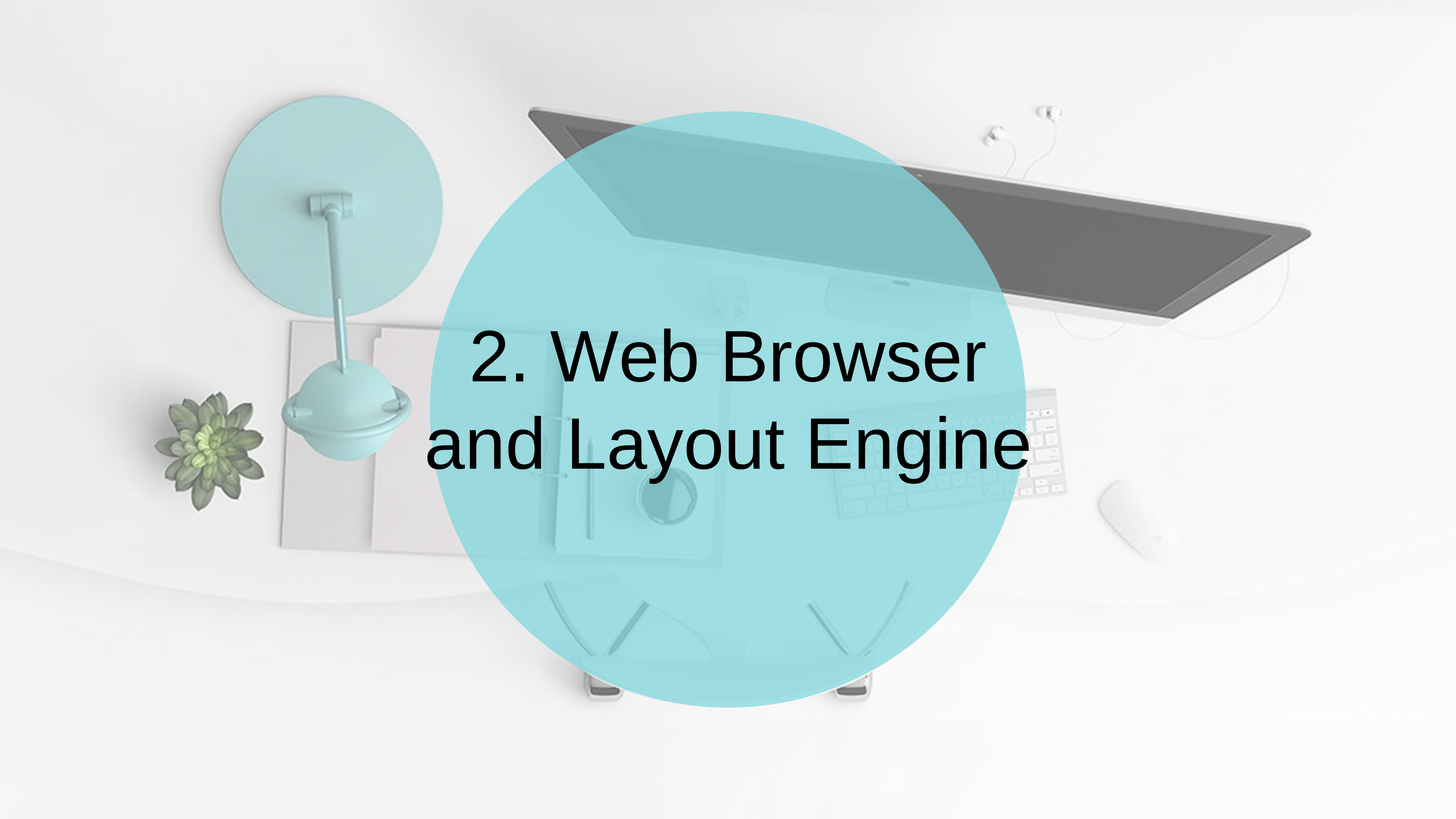
→ **All publicly accessible websites** collectively constitute the **World Wide Web**

Web Application

- Next level web sites
- High interactivity
- High accessibility (Cloud)
- Desktop-like application in the web browser

Operating Principle





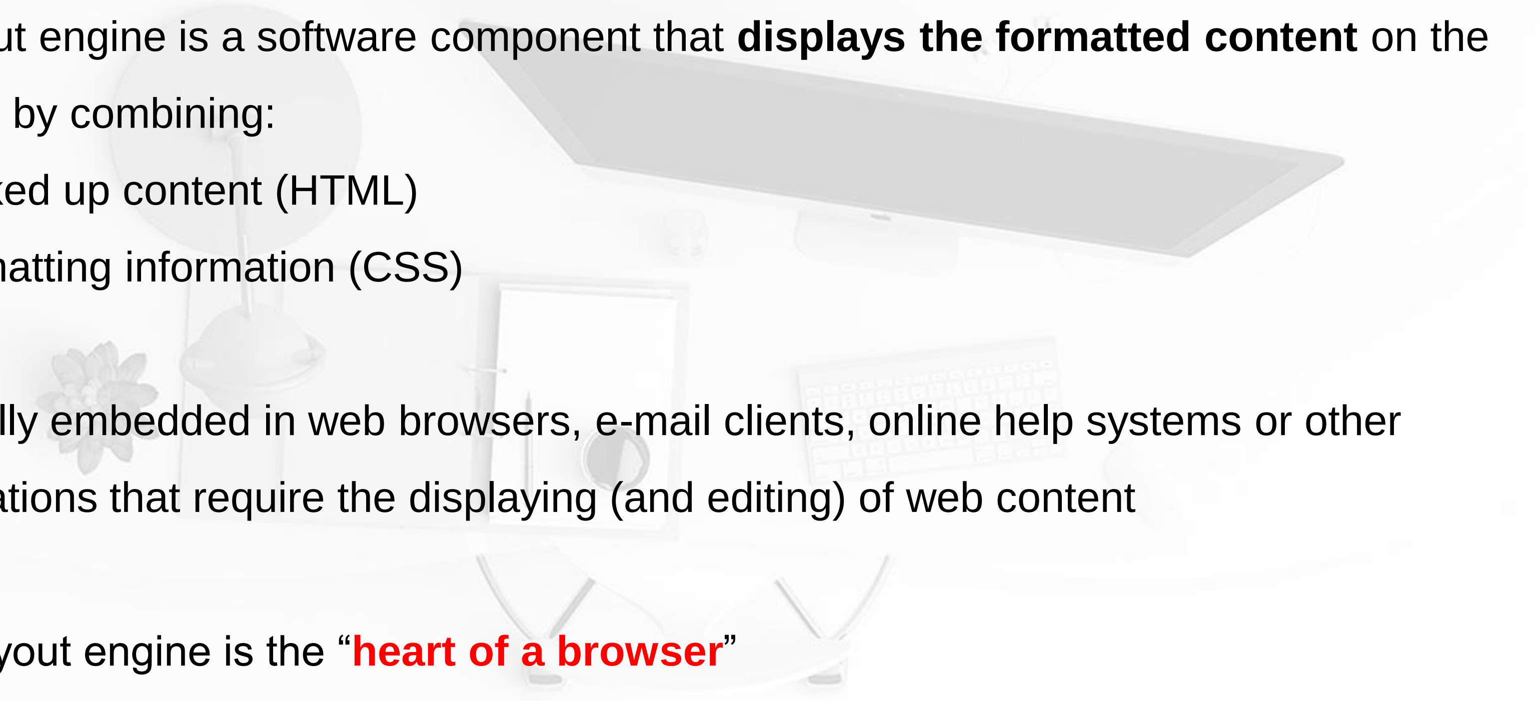
2. Web Browser and Layout Engine

Web Browser

- A Web browser is a program designed to enable users to **access, retrieve** and **view** documents and other resources from the Web

- Main responsibilities:
  - **Bring** information resources to the user
  - **Present** web content (render HTML, CSS, JS)
  - **Capable of executing** applications within the same context as the document on view (Flash)

Layout Engine



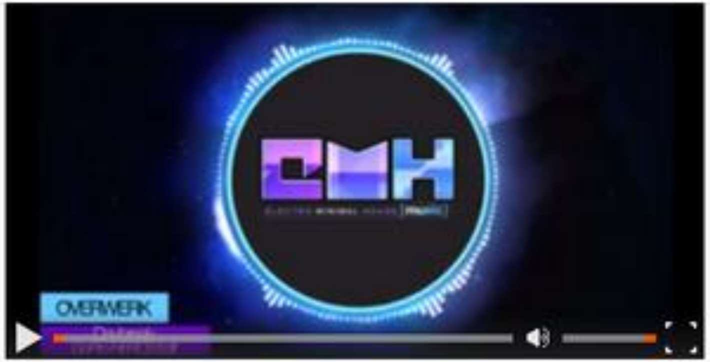
- A layout engine is a software component that **displays the formatted content** on the screen by combining:
  - Marked up content (HTML)
  - Formatting information (CSS)

- Typically embedded in web browsers, e-mail clients, online help systems or other applications that require the displaying (and editing) of web content

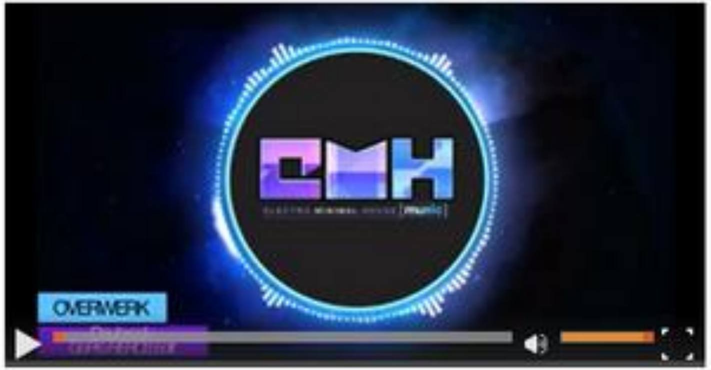
- The layout engine is the “**heart of a browser**”

Why are some elements displayed differently on different browsers?

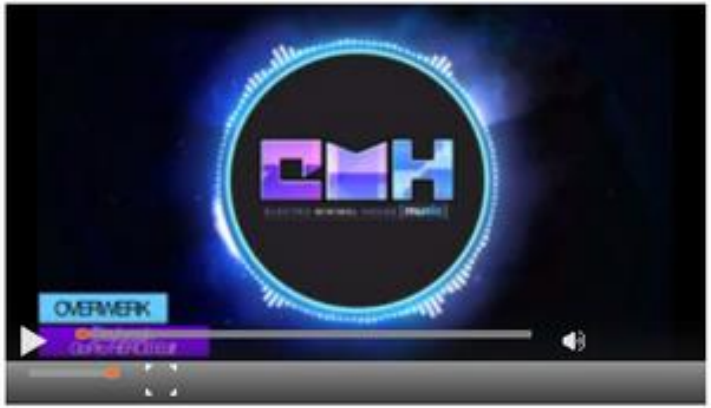
chrome



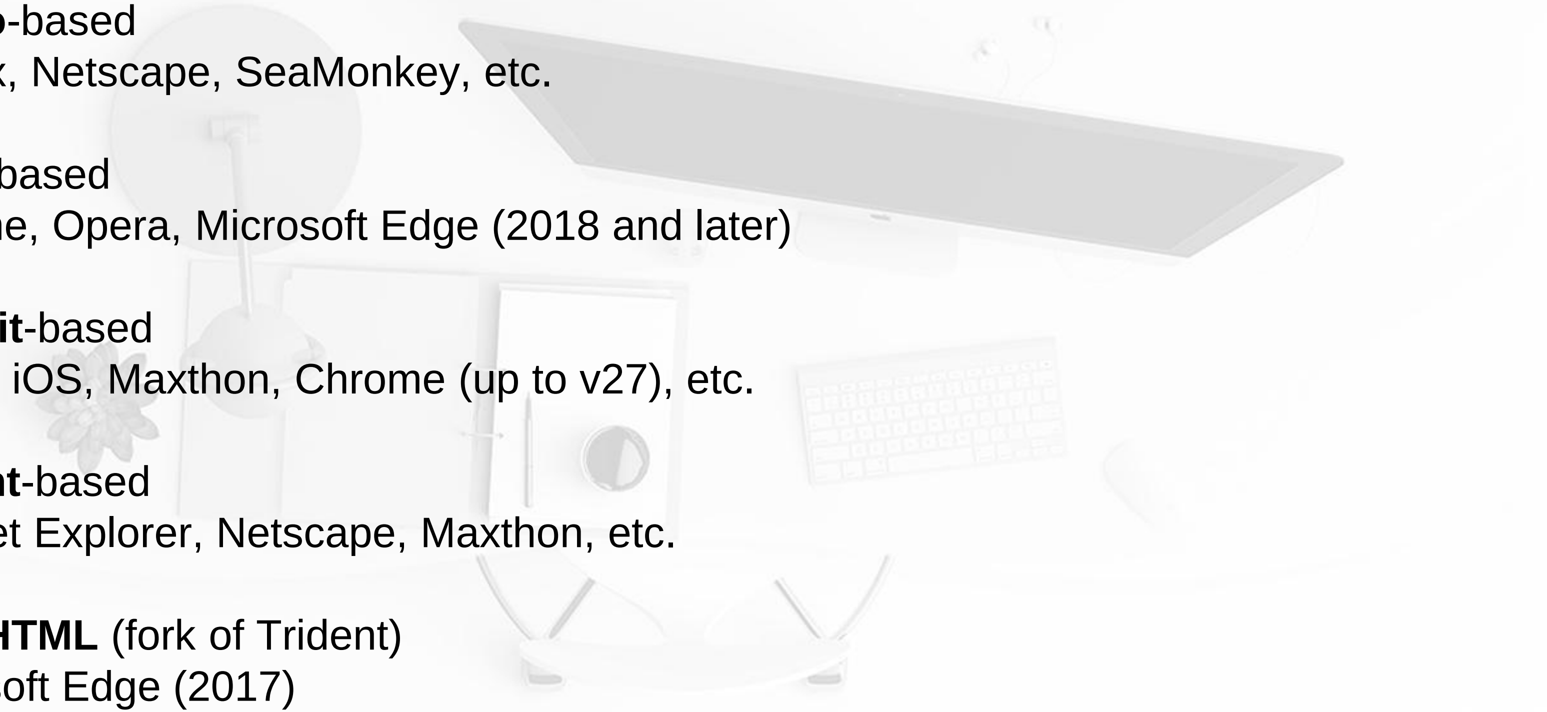
EDGE



Firefox



Layout Engines and Web Browsers



- **Gecko**-based
  Firefox, Netscape, SeaMonkey, etc.

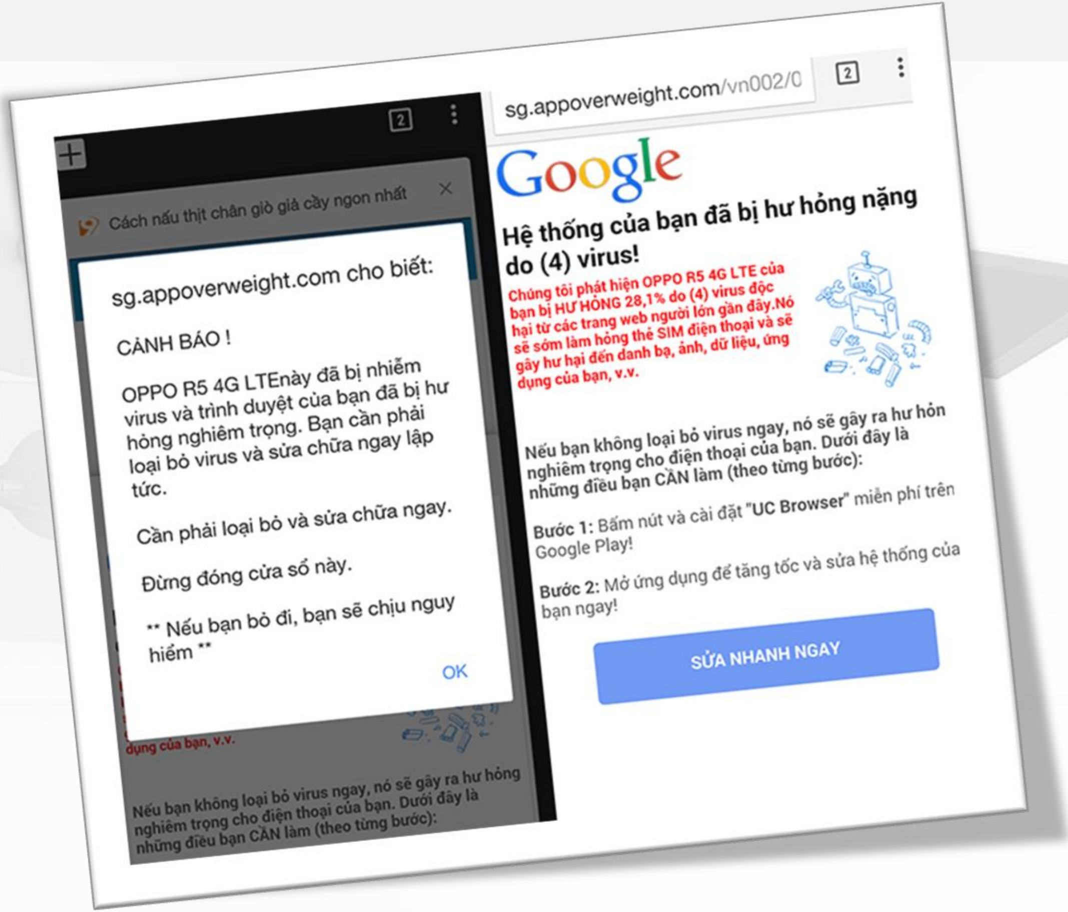
- **Blink**-based
  Chrome, Opera, Microsoft Edge (2018 and later)

- **WebKit**-based
  Safari, iOS, Maxthon, Chrome (up to v27), etc.

- **Trident**-based
  Internet Explorer, Netscape, Maxthon, etc.

- **EdgeHTML** (fork of Trident)
  Microsoft Edge (2017)

How do websites know what device we are on?



sg.appoverweight.com cho biết:

CẢNH BÁO !

OPPO R5 4G LTEnày đã bị nhiễm virus và trình duyệt của bạn đã bị hư hỏng nghiêm trọng. Bạn cần phải loại bỏ virus và sửa chữa ngay lập tức.

Cần phải loại bỏ và sửa chữa ngay.

Đừng đóng cửa sổ này.

** Nếu bạn bỏ đi, bạn sẽ chịu nguy hiểm **

dụng của bạn, v.v.

Nếu bạn không loại bỏ virus ngay, nó sẽ gây ra hư hỏng nghiêm trọng cho điện thoại của bạn. Dưới đây là những điều bạn CẦN làm (theo từng bước):

Hệ thống của bạn đã bị hư hỏng nặng do (4) virus!

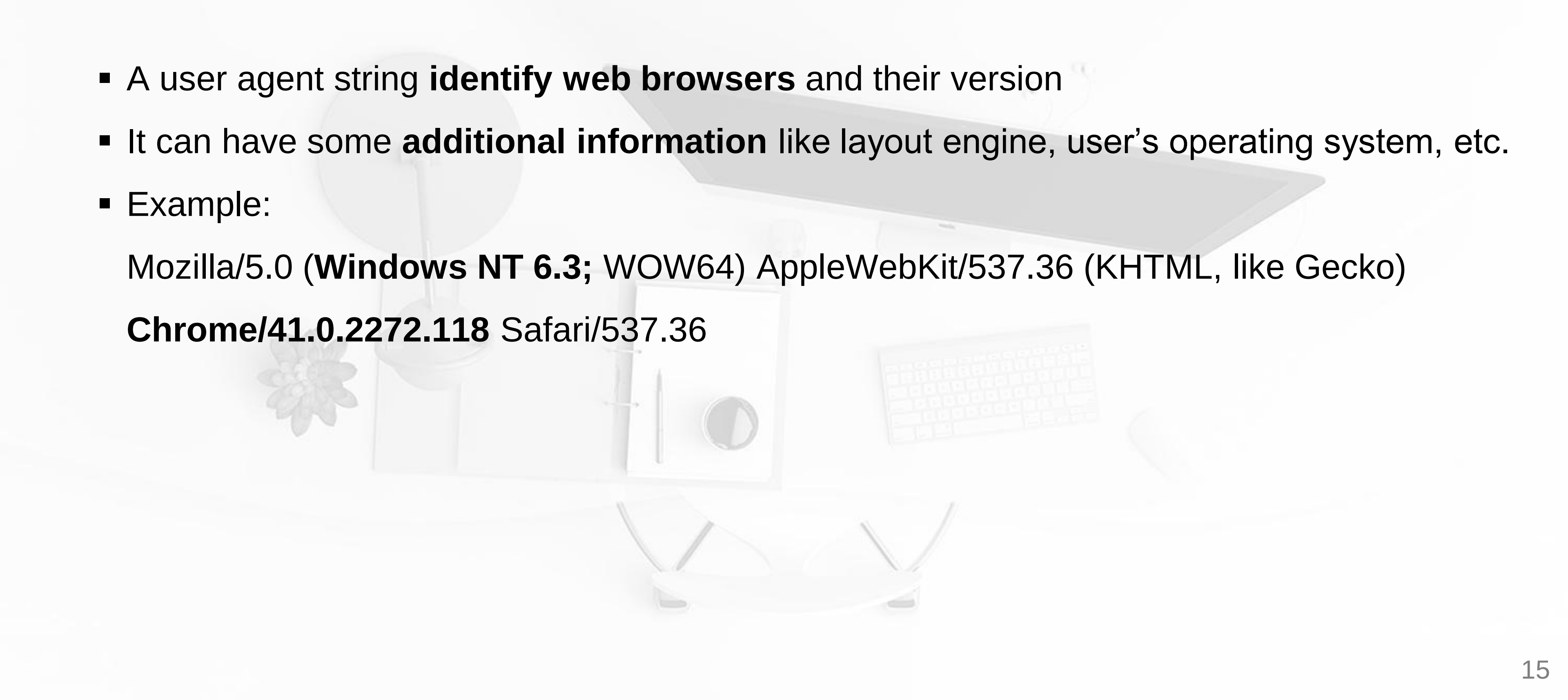
Chúng tôi phát hiện OPPO R5 4G LTE của bạn bị HƯ HỎNG 28,1% do (4) virus độc hại từ các trang web người lớn gần đây.Nó sẽ sớm làm hỏng thẻ SIM điện thoại và sẽ gây hư hại đến danh bạ, ảnh, dữ liệu, ứng dụng của bạn, v.v.

Nếu bạn không loại bỏ virus ngay, nó sẽ gây ra hư hỏn nghiêm trọng cho điện thoại của bạn. Dưới đây là những điều bạn CẦN làm (theo từng bước):

Bước 1: Bấm nút và cài đặt **"UC Browser"** miễn phí trên Google Play!

Bước 2: Mở ứng dụng để tăng tốc và sửa hệ thống của bạn ngay!

User Agent String



- A user agent string **identify web browsers** and their version
- It can have some **additional information** like layout engine, user’s operating system, etc.
- Example:

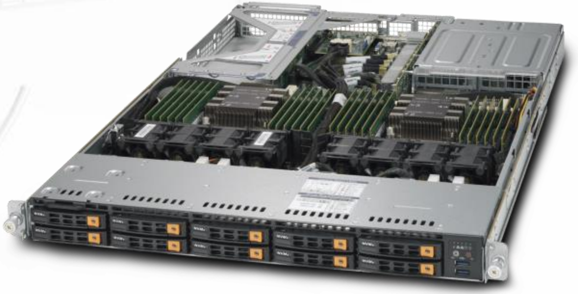
  Mozilla/5.0 (**Windows NT 6.3;** WOW64) AppleWebKit/537.36 (KHTML, like Gecko)

  Chrome/41.0.2272.118 Safari/537.36

3. Hardware Server

Hardware Server

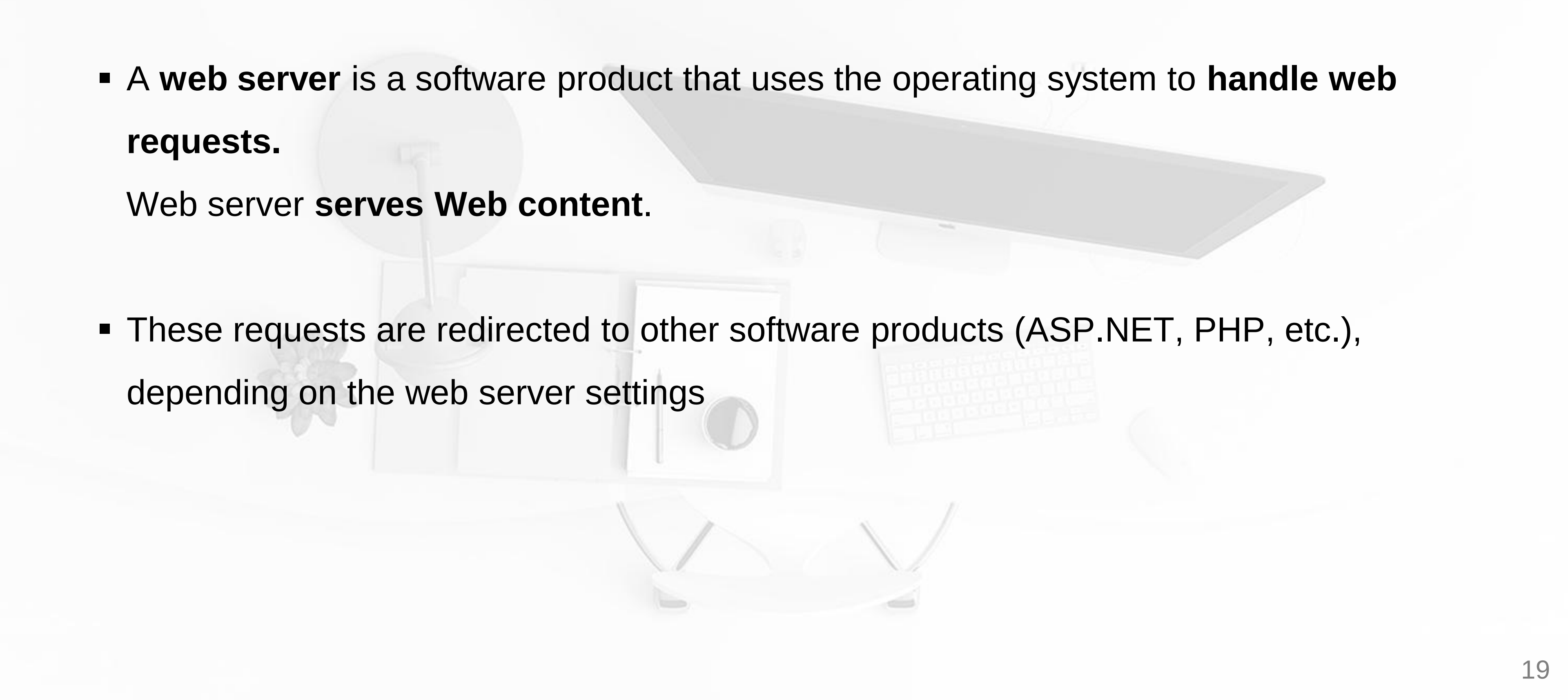
- A hardware server is a **physical computer** dedicated to running one or more such services
- Servers are placed in colocation centers
- The server may be:
  - Database server
  - File server
  - Mail server
  - Print server
  - VPS server



4. Web Server

Apache, IIS, nginx, lighttpd,...

What do the Web Servers Do?



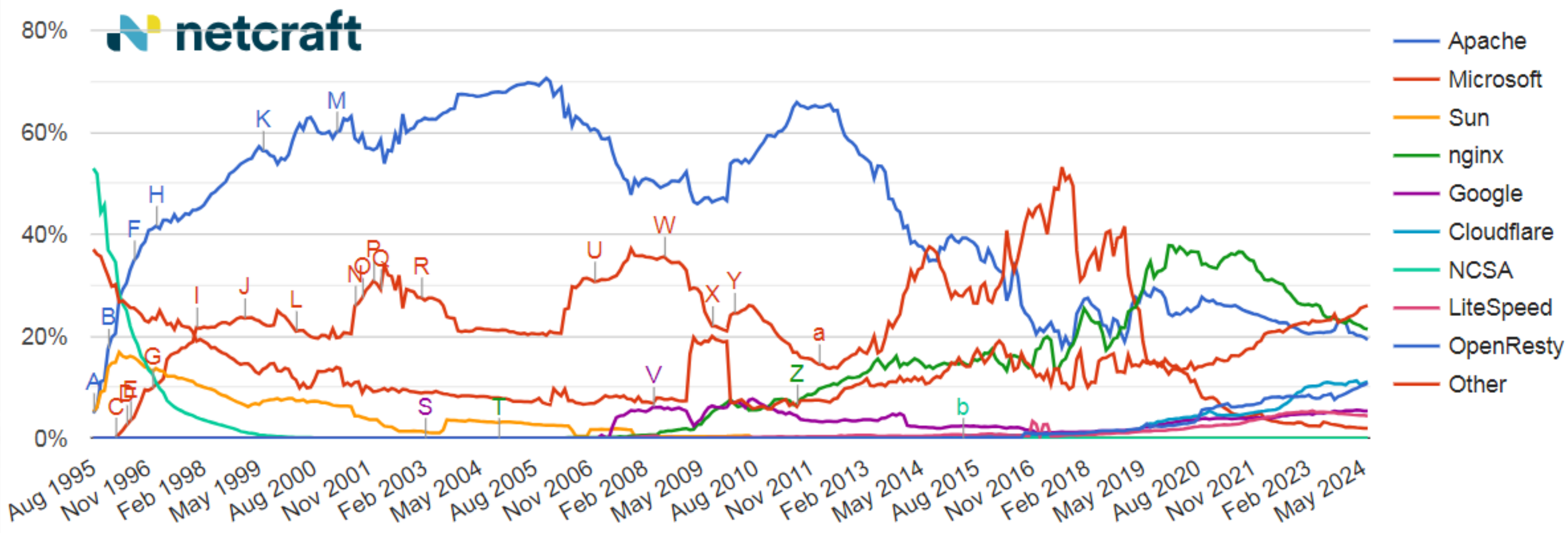
- A **web server** is a software product that uses the operating system to **handle web requests.**

  Web server **serves Web content**.

- These requests are redirected to other software products (ASP.NET, PHP, etc.), depending on the web server settings

Web Servers Market Share

May 2024



- Nginx 21.53%
- Apache 19.80%
- Microsoft 2%
- Cloudflare 10.8%

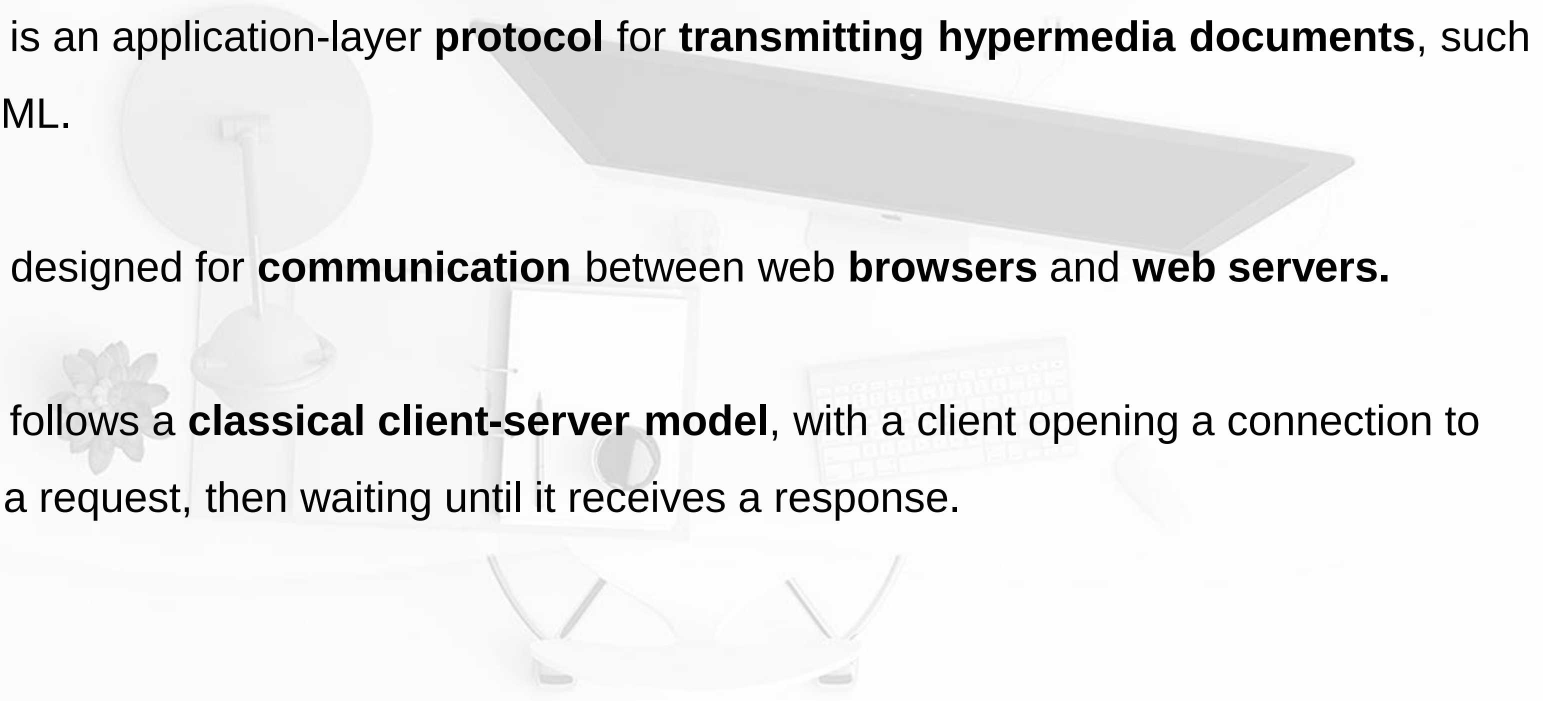
Source: https://news.netcraft.com/

5. HTTP and HTTPS

Hypertext Transfer Protocol

Hypertext Transfer Protocol

HTTP



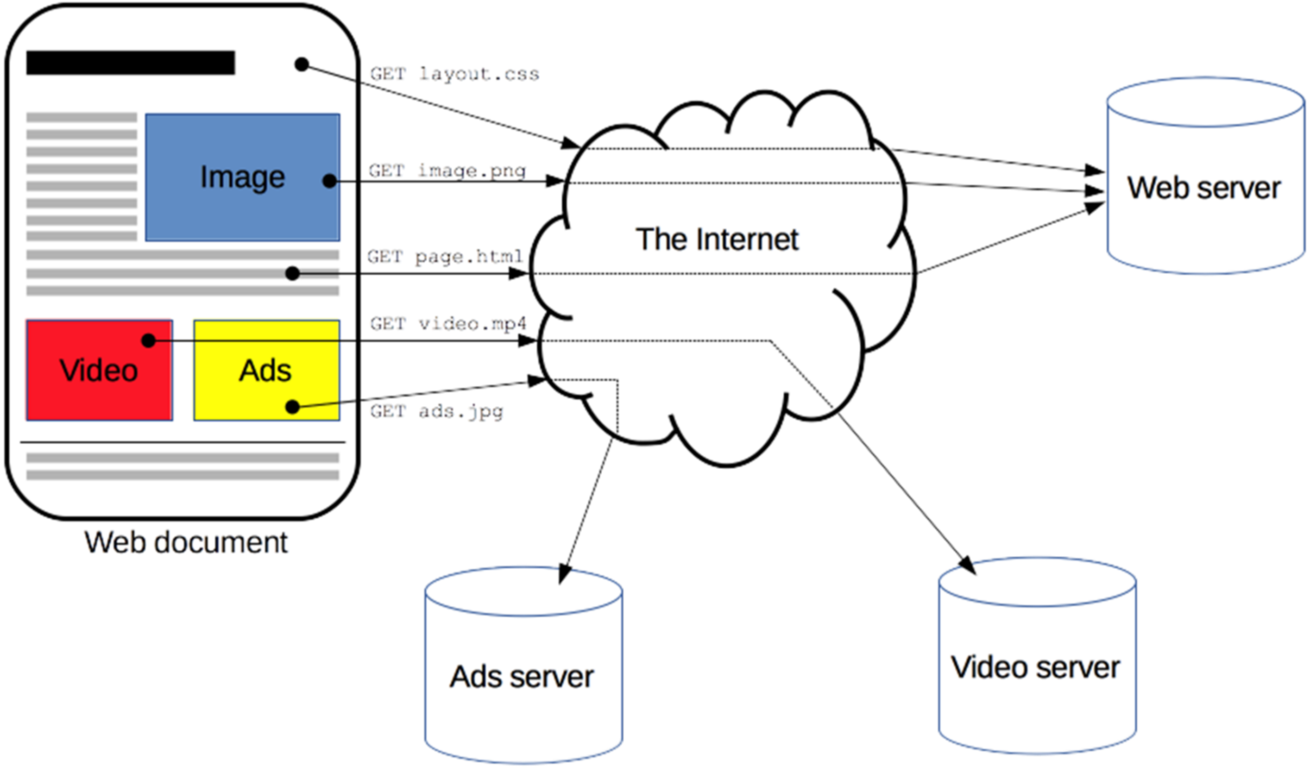
- HTTP is an application-layer **protocol** for **transmitting hypermedia documents**, such as HTML.

- It was designed for **communication** between web **browsers** and **web servers.**

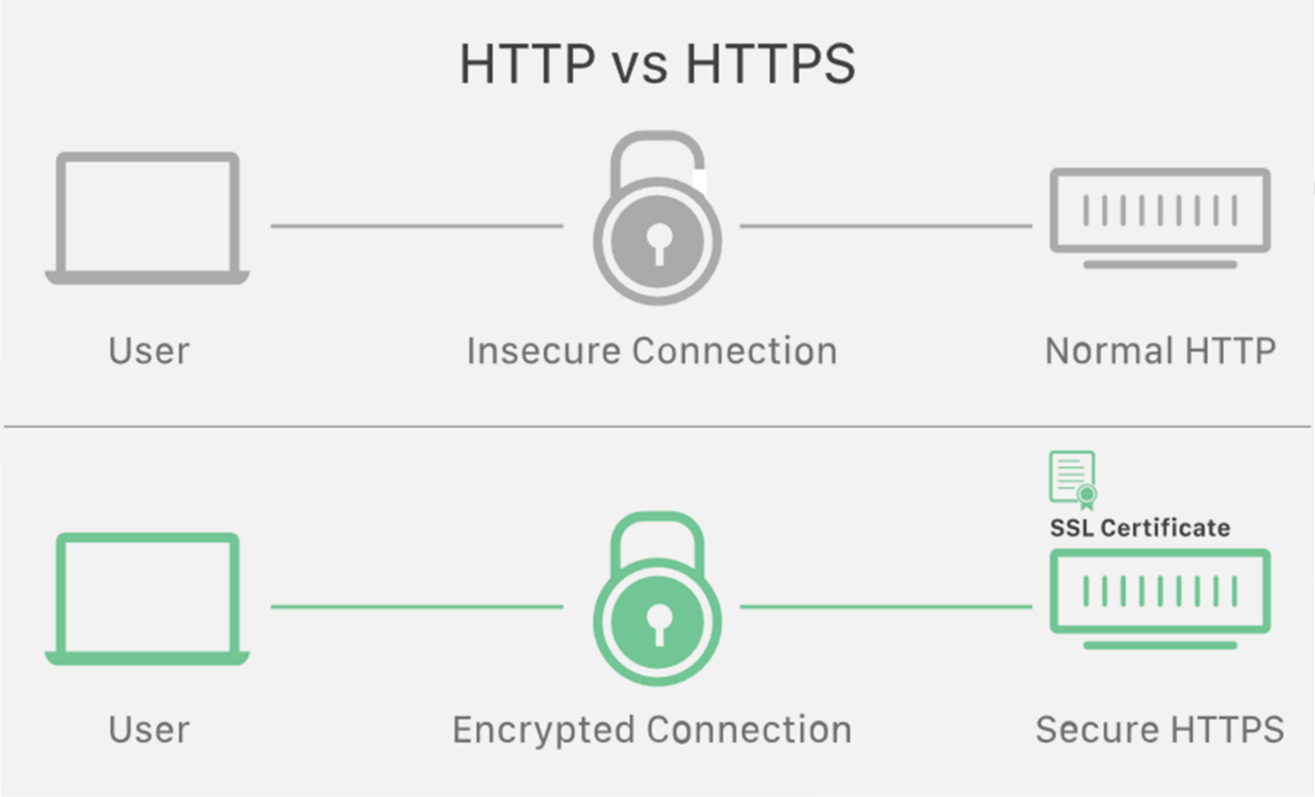
- HTTP follows a **classical client-server model**, with a client opening a connection to make a request, then waiting until it receives a response.

Hypertext Transfer Protocol

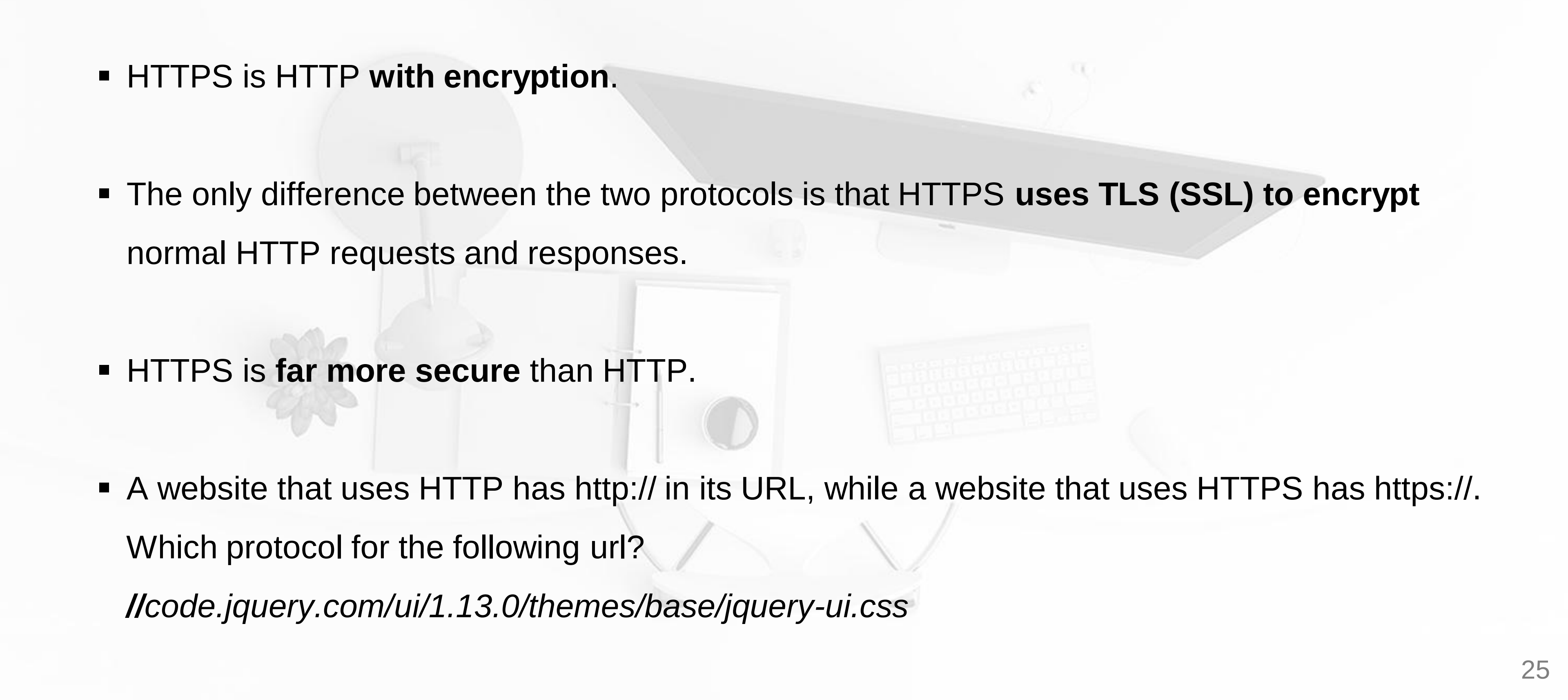
HTTP



HTTPS

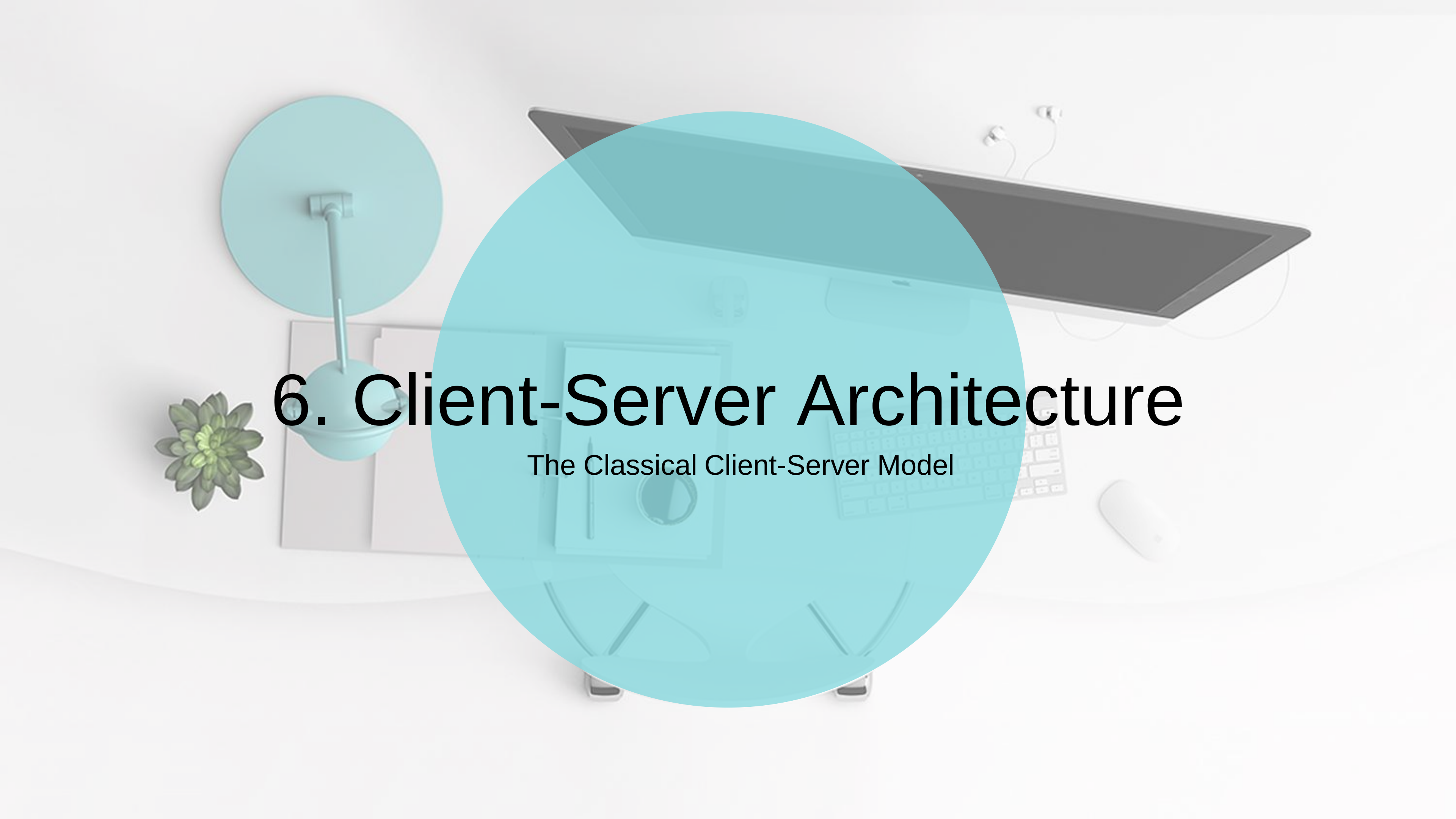

HTTP vs HTTPS
User
Insecure Connection
Normal HTTP
SSL Certificate
User
Encrypted Connection
Secure HTTPS

HTTPS



- HTTPS is HTTP **with encryption**.
- The only difference between the two protocols is that HTTPS **uses TLS (SSL) to encrypt** normal HTTP requests and responses.
- HTTPS is **far more secure** than HTTP.
- A website that uses HTTP has http:// in its URL, while a website that uses HTTPS has https://. Which protocol for the following url?

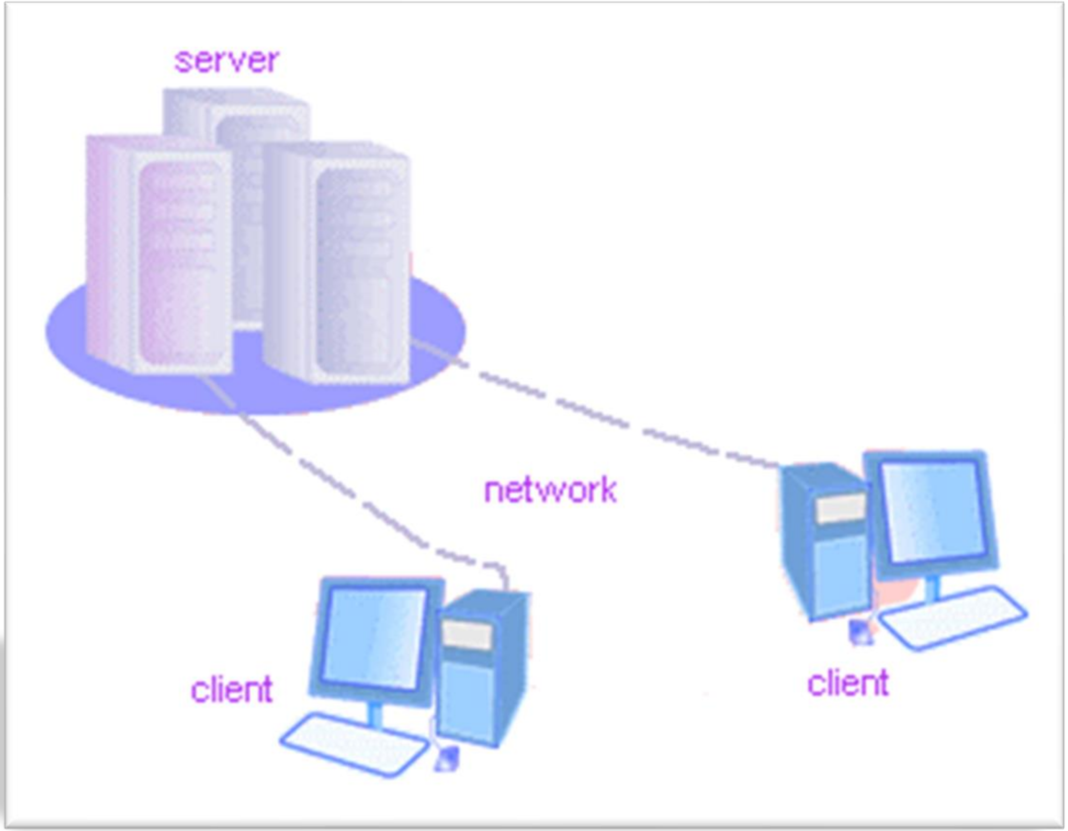
  //code.jquery.com/ui/1.13.0/themes/base/jquery-ui.css



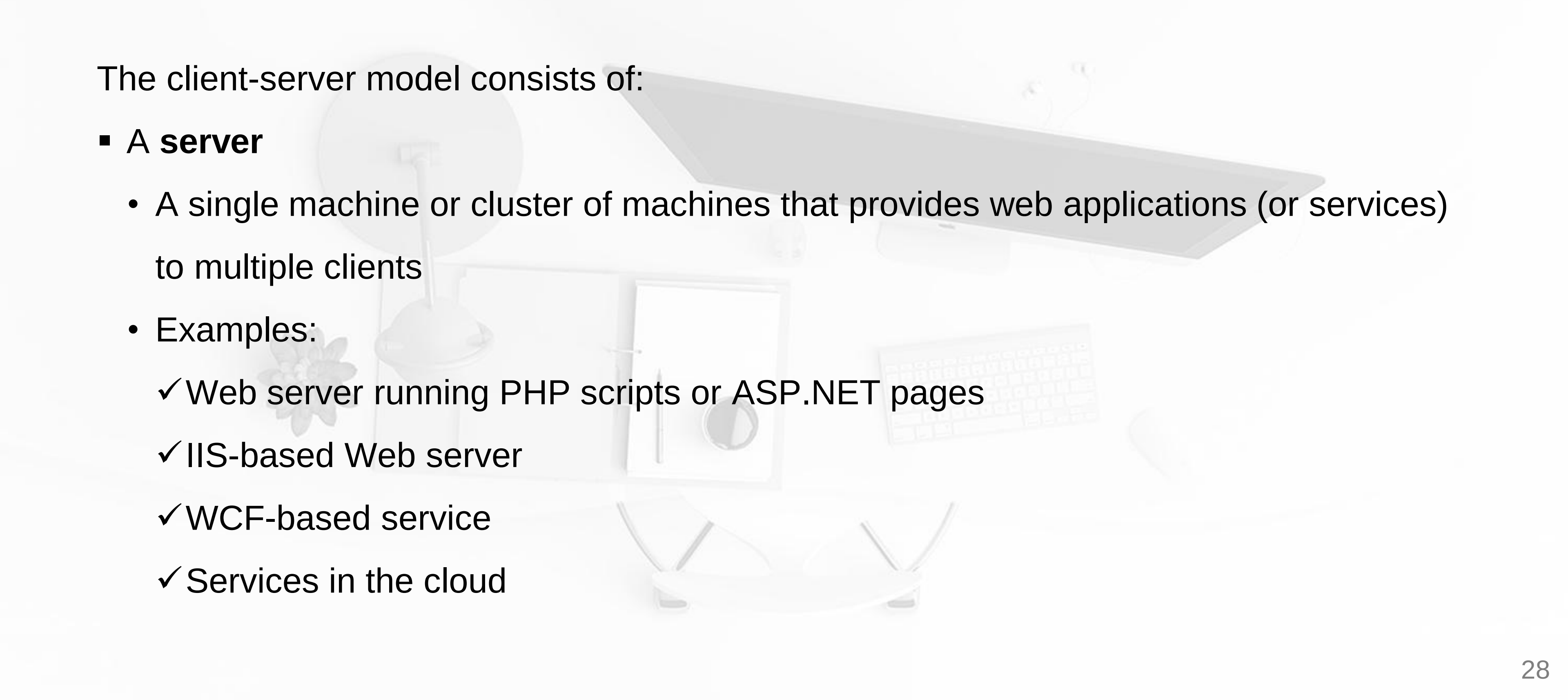
6. Client-Server Architecture

The Classical Client-Server Model

Client-Server Architecture



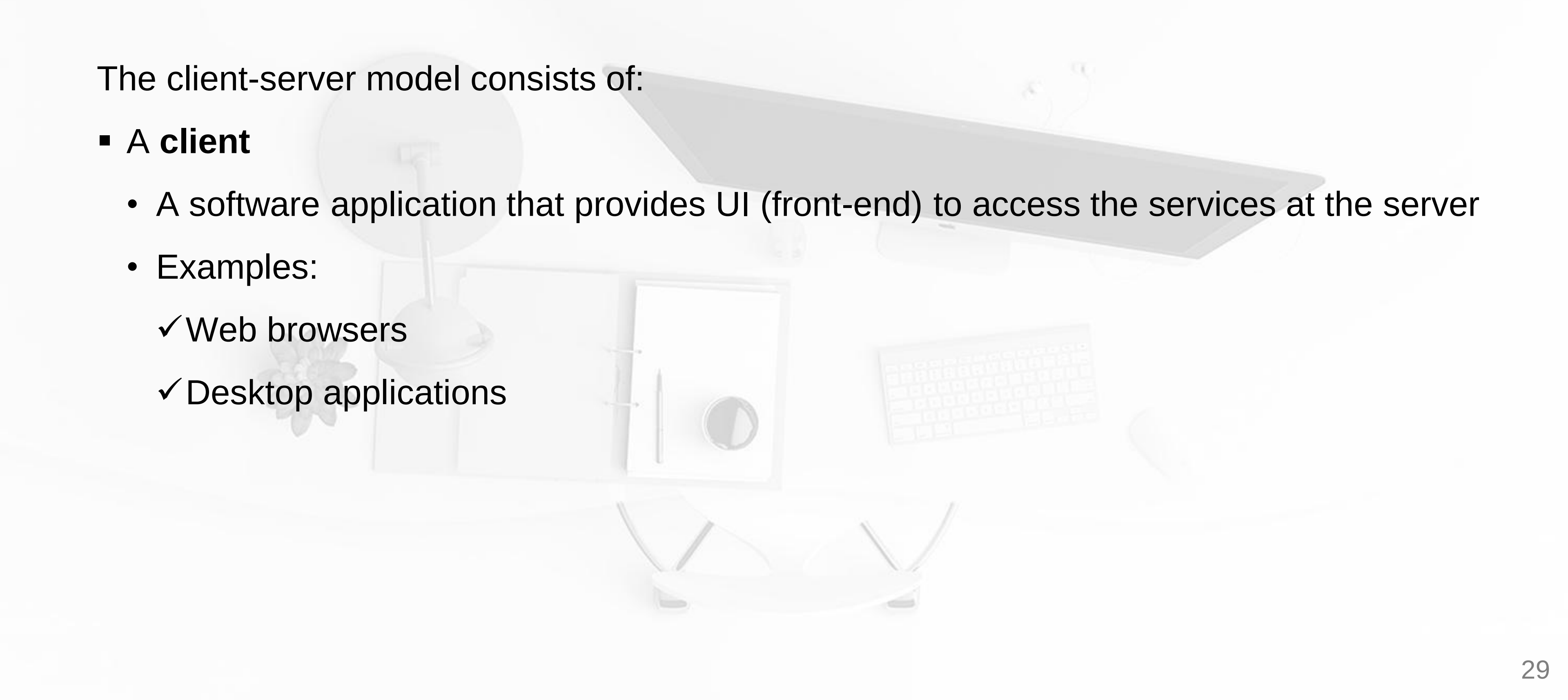
Client-Server Architecture



The client-server model consists of:

- A **server**
  - A single machine or cluster of machines that provides web applications (or services) to multiple clients
  - Examples:
    - ✓Web server running PHP scripts or ASP.NET pages
    - ✓IIS-based Web server
    - ✓WCF-based service
    - ✓Services in the cloud

Client-Server Architecture

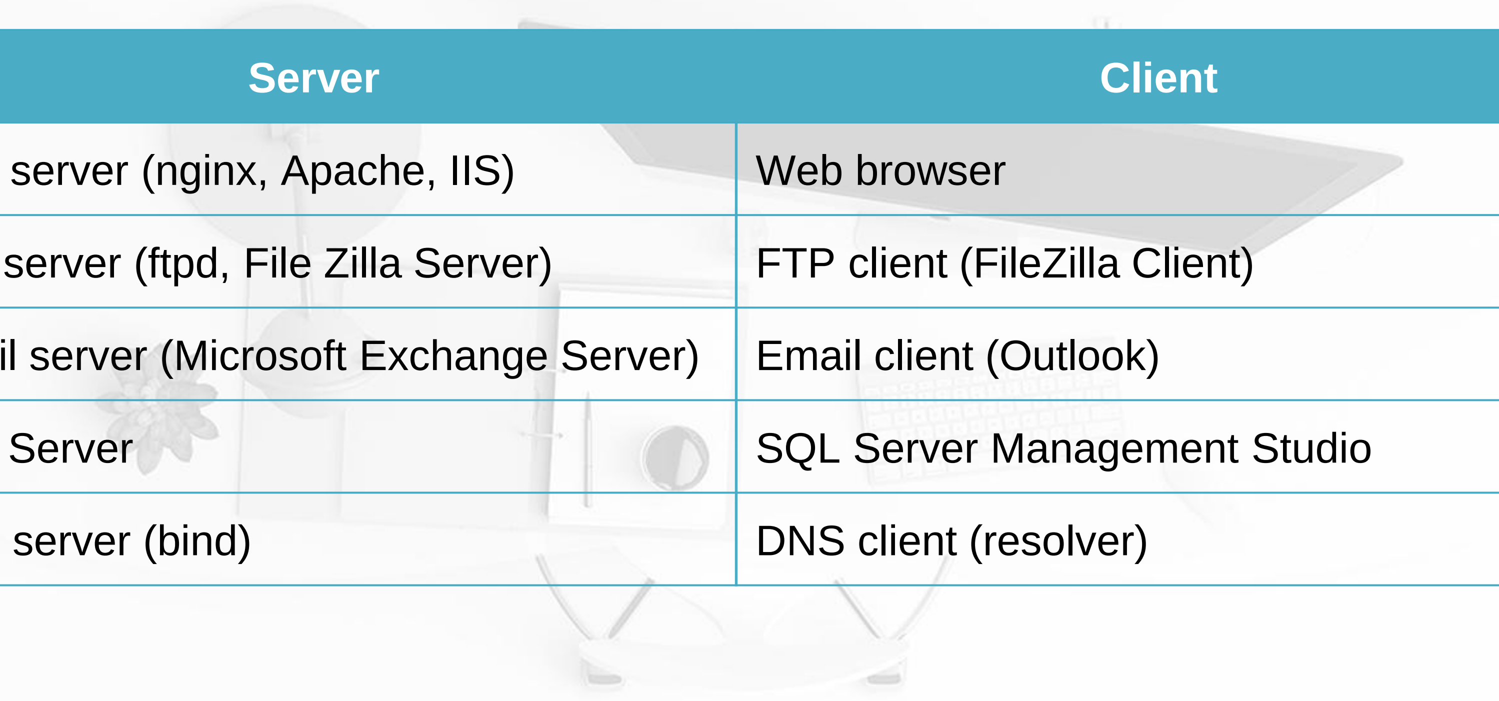


The client-server model consists of:

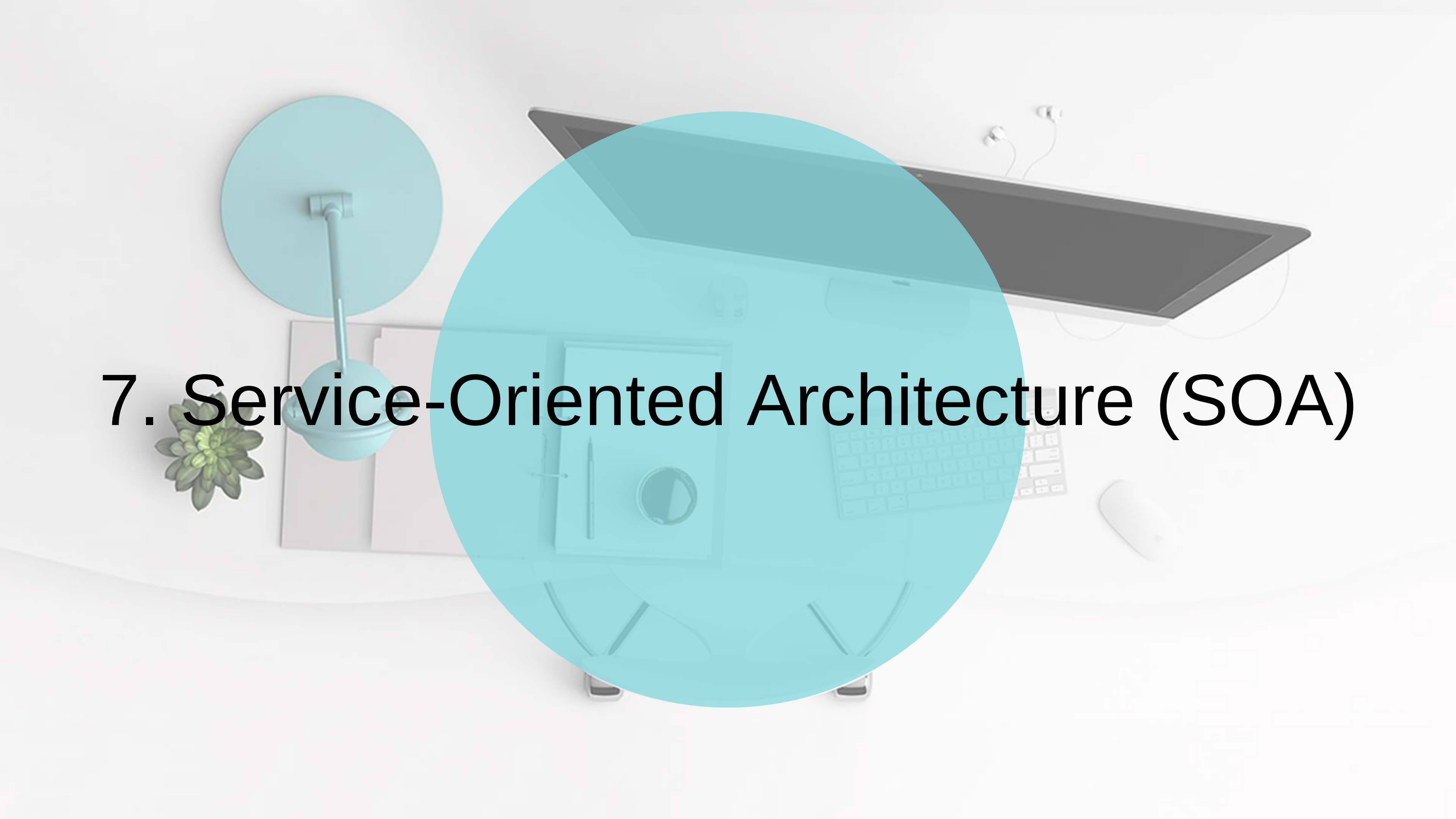
- A **client**
  - A software application that provides UI (front-end) to access the services at the server
  - Examples:
    - ✓Web browsers
    - ✓Desktop applications

Client-Server Model

Examples

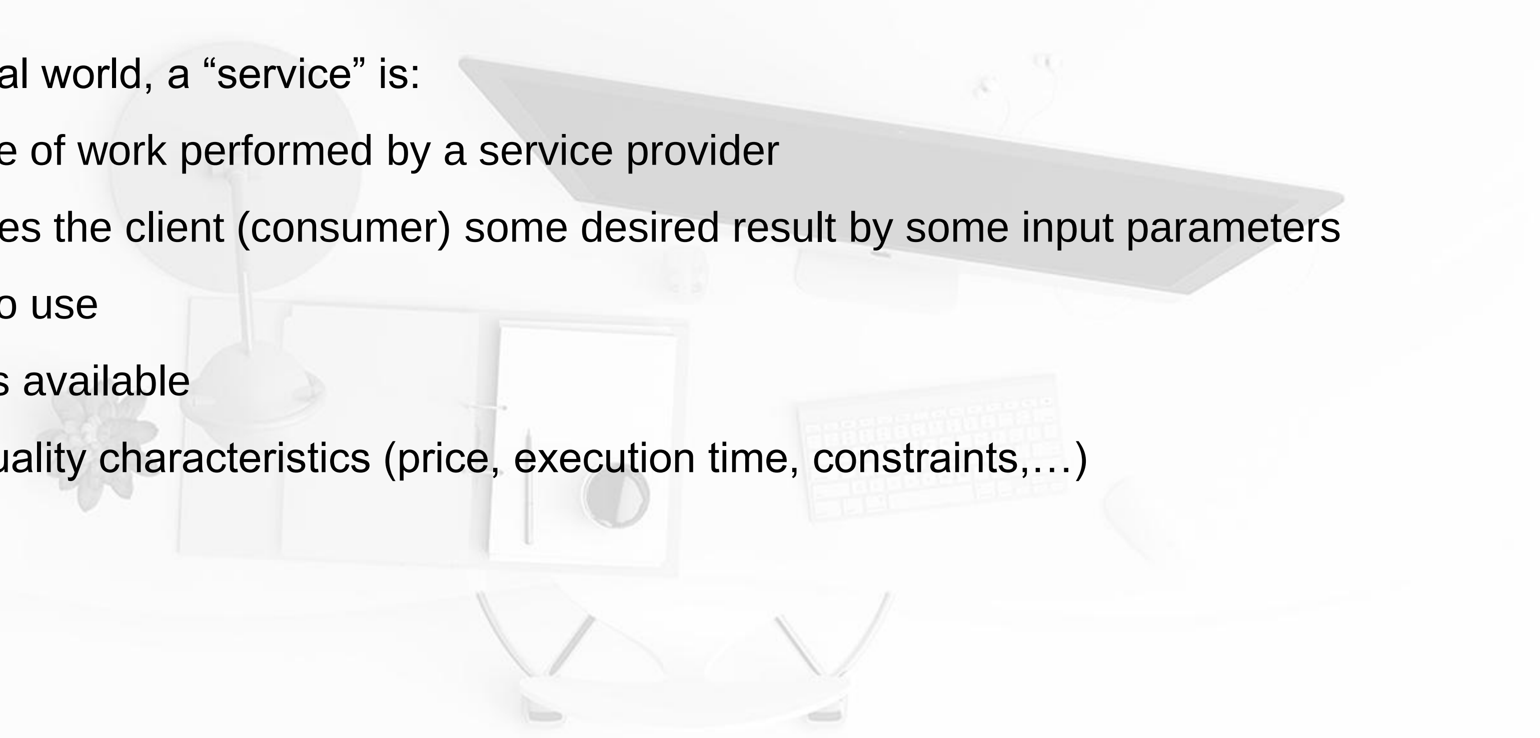


Server	Client
Web server (nginx, Apache, IIS)	Web browser
FTP server (ftpd, File Zilla Server)	FTP client (FileZilla Client)
EMail server (Microsoft Exchange Server)	Email client (Outlook)
SQL Server	SQL Server Management Studio
DNS server (bind)	DNS client (resolver)



7. Service-Oriented Architecture (SOA)

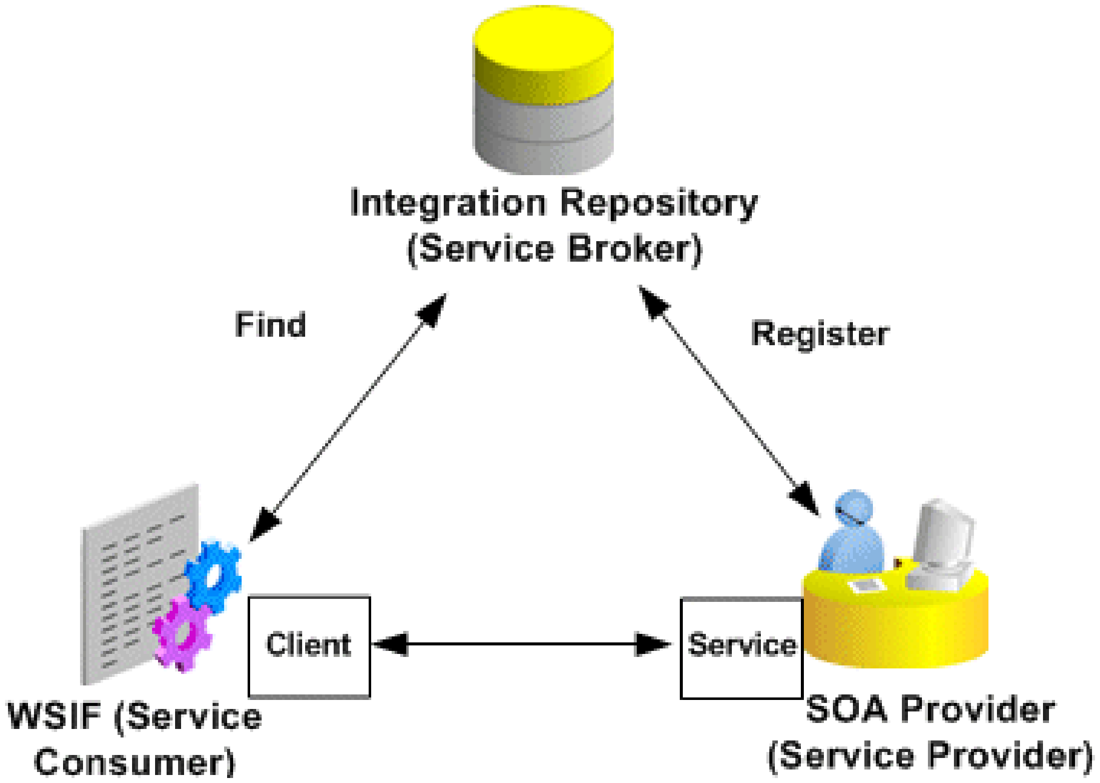
What is a Service?



In the real world, a “service” is:

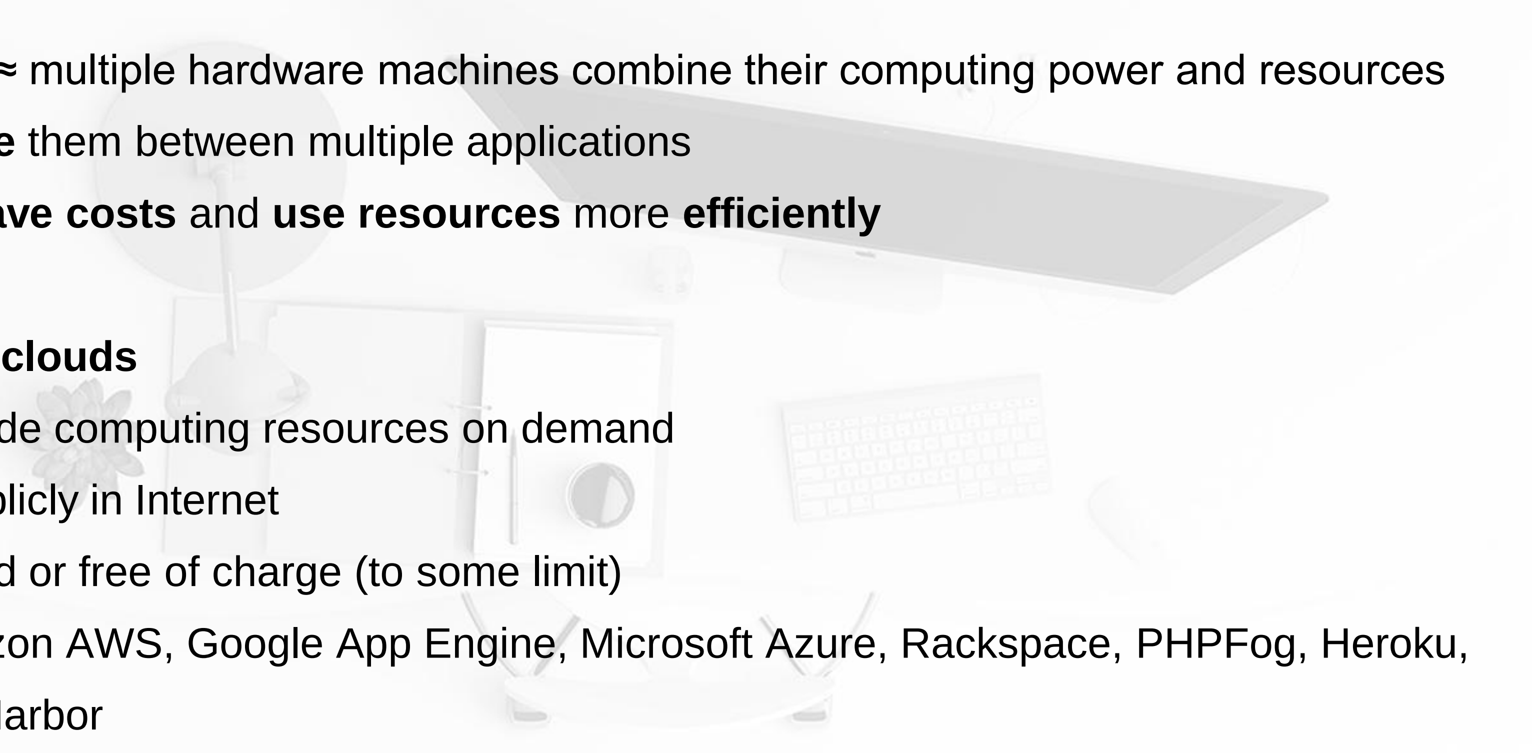
- A piece of work performed by a service provider
- Provides the client (consumer) some desired result by some input parameters
- Easy to use
- Always available
- Has quality characteristics (price, execution time, constraints,…)

Service-Oriented Architecture



8. What is “Cloud”?

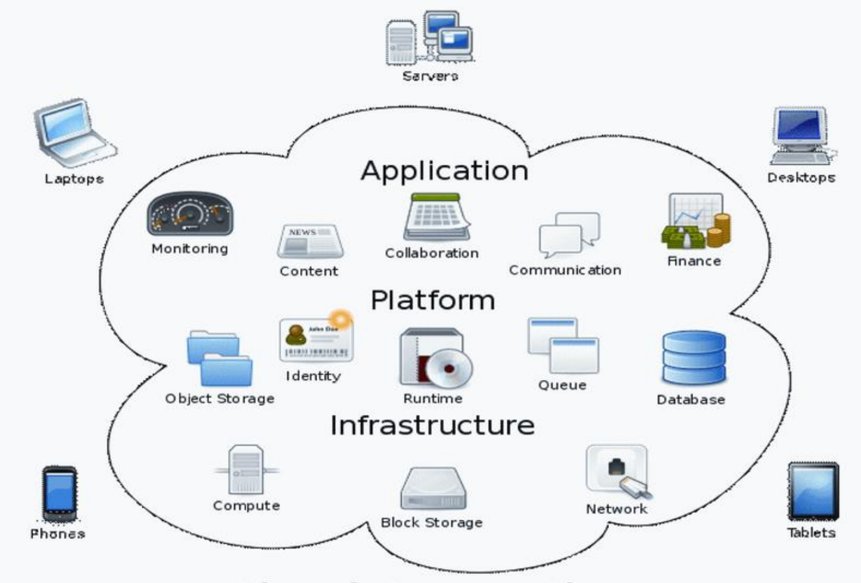
What is Cloud?

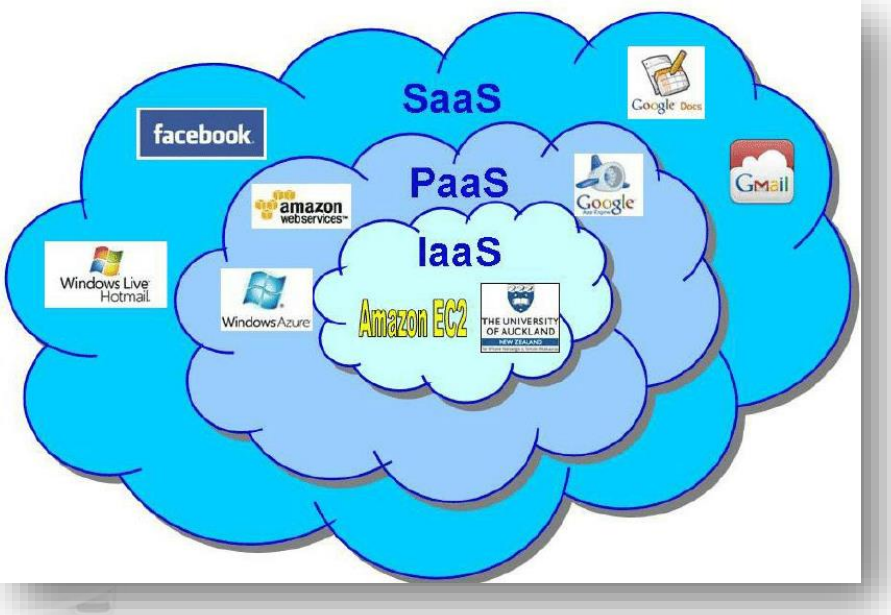


- **Cloud ≈** multiple hardware machines combine their computing power and resources
  - **Share** them between multiple applications
  - To **save costs** and **use resources** more **efficiently**

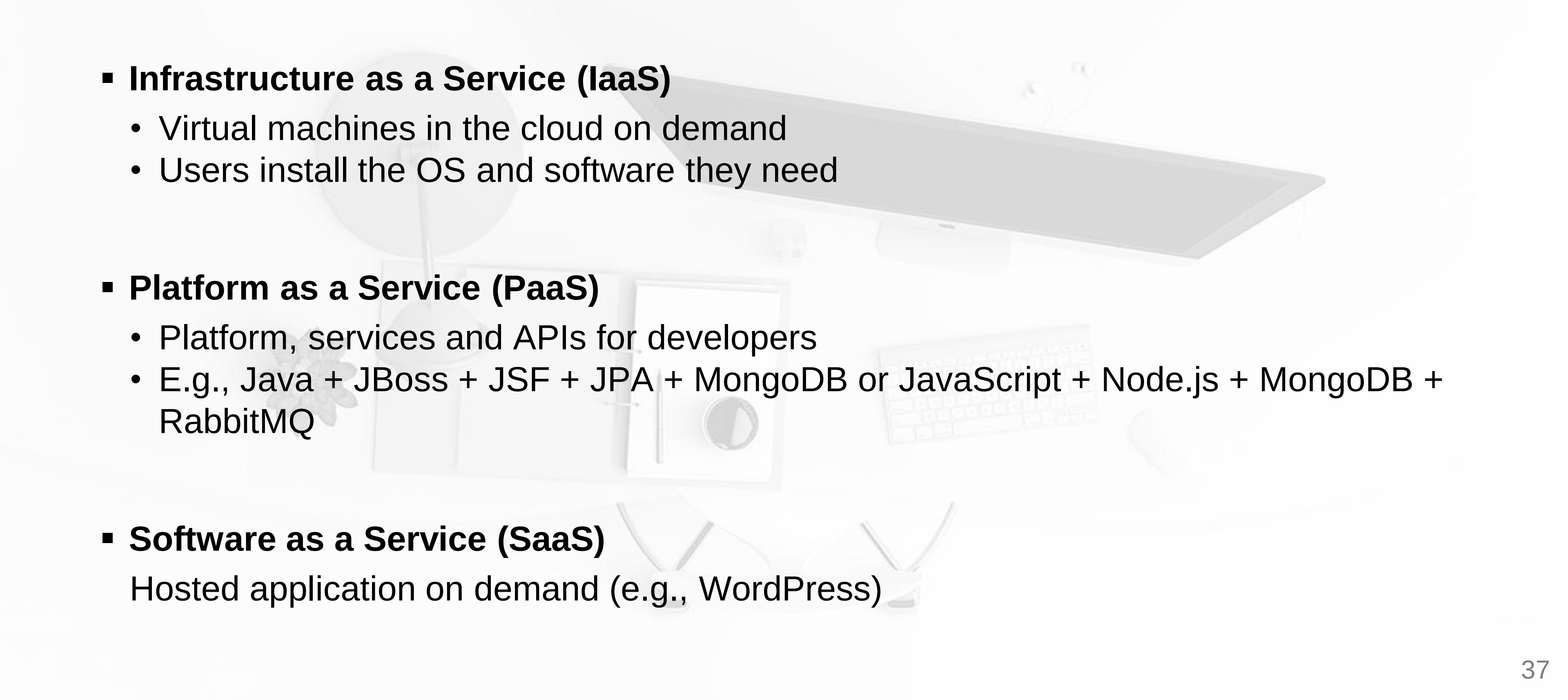
- **Public clouds**
  - Provide computing resources on demand
    - ✓Publicly in Internet
    - ✓Paid or free of charge (to some limit)
  - Amazon AWS, Google App Engine, Microsoft Azure, Rackspace, PHPFog, Heroku, AppHarbor

Cloud Computing Models





Cloud Computing Models



- **Infrastructure as a Service (IaaS)**
  - Virtual machines in the cloud on demand
  - Users install the OS and software they need
- **Platform as a Service (PaaS)**
  - Platform, services and APIs for developers
  - E.g., Java + JBoss + JSF + JPA + MongoDB or JavaScript + Node.js + MongoDB + RabbitMQ
- **Software as a Service (SaaS)**

  Hosted application on demand (e.g., WordPress)

Q&A