HALMSTAD UNIVERSITY

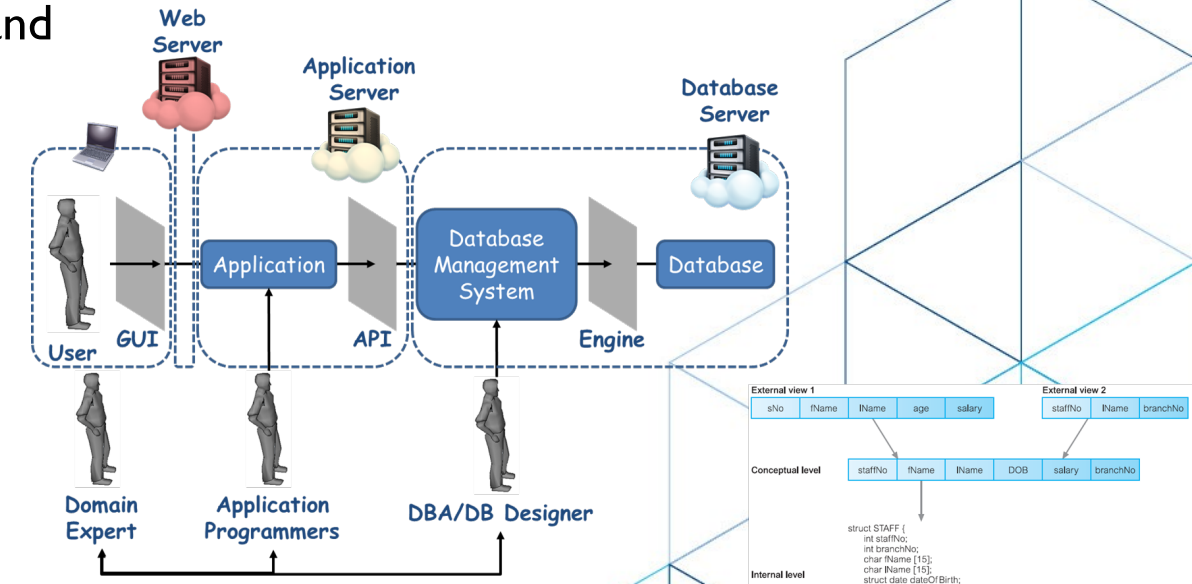
Web Systems Fundamentals and Databases (Grundläggande webbsystem och databaser) DI4020 - 11hp

Lecture 9

Wagner Ourique de Morais

Previously in the course

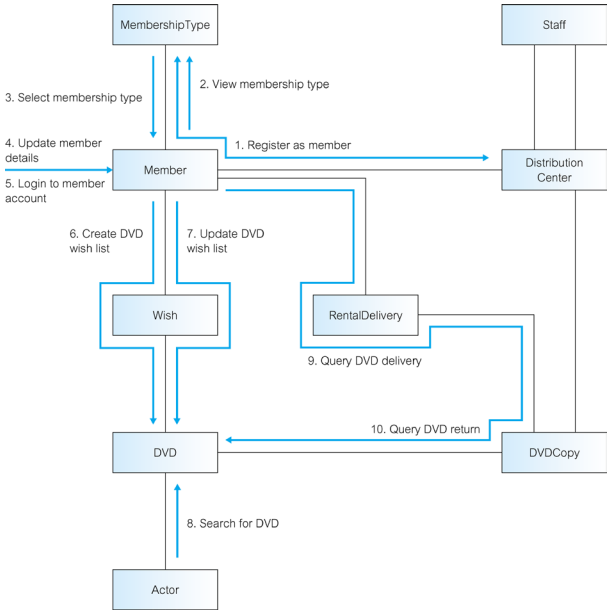
- Course Introduction
  - Course description and plan
- Lectures 1 and 2
  - Introduction to the field of field of database systems and database design.
    - Common uses of database systems.
    - Discuss the problems with file-based approach.
      - Data dependency
    - Terminology in the field.
    - Purpose, components and functions of a DBMS.
      - Advantages and disadvantages.
      - Types of languages that are used by DBMSs.
    - Data models and conceptual modelling.
      - Data independence
    - Client-server architecture and multi-tier architectures.

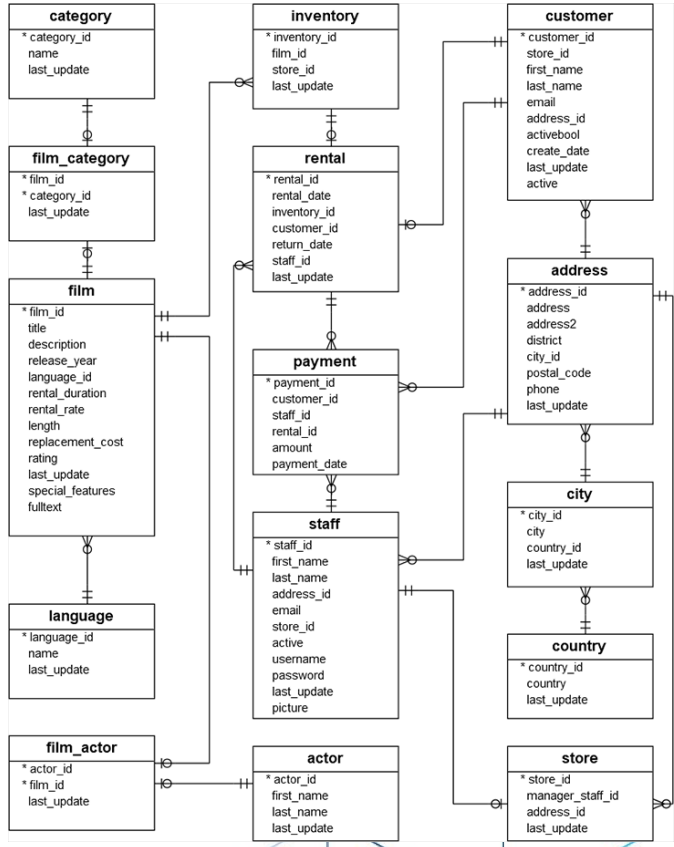


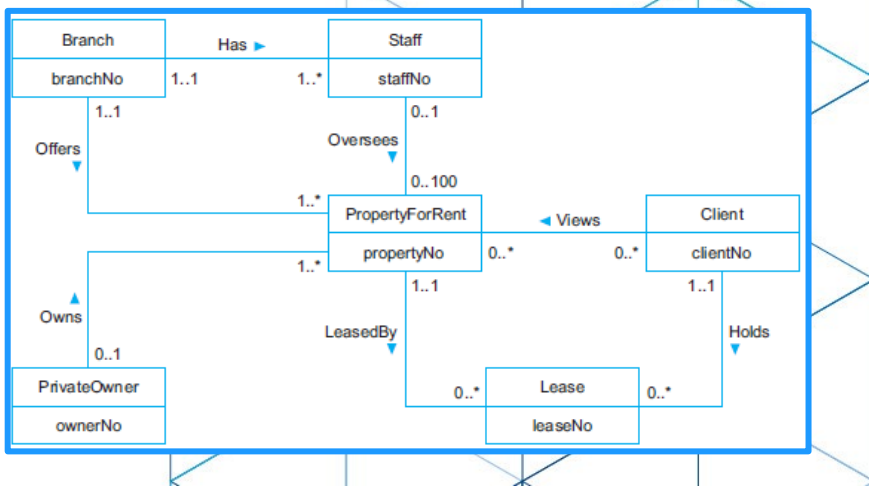
HALMSTAD UNIVERSITY

Previously in the course

- Lectures 3 and 4
  - Database design
    - Conceptual, logical and physical design.
  - Types of database design
    - Database design from for new systems
    - Database design from existing data
    - Database design from database redesign
  - Data modeling/database design process
  - Conceptual design
    - Build a model of the data used in an enterprise, independent of all physical considerations.
  - Entity-relationship (ER) data model and ER diagrams
    - Entities, attributes, and relationships





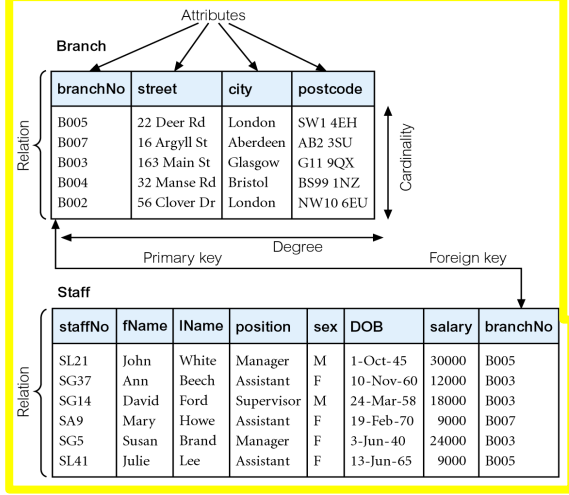


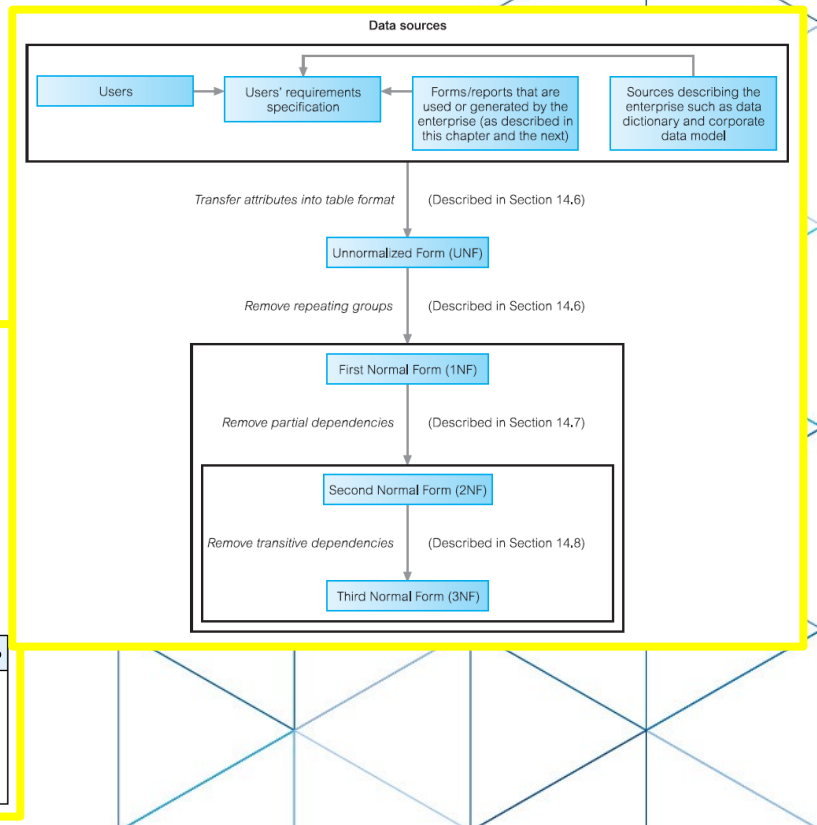
HALMSTAD UNIVERSITY

Previously in the course

- Lecture 5
  - Logical database design for the relational model
    - Build logical data model based on a specific data model, but independent of a particular DBMS and other physical considerations.
  - Relational model and relational database design.
  - Terminology of the relational model.
  - Primary, candidate, and composite keys.
  - Normalization and the normalization process.

Formal terms	Alternative 1	Alternative 2
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field





HALMSTAD UNIVERSITY

Previously in the course

- Lectures 6 to 8
  - Physical database design
    - Translate logical data model for target DBMS.
  - Structured Query Language (SQL)
    - Purpose and importance of SQL
    - SQL syntax and how to write SQL commands
    - Integrity constraints
    - DDL and DML

- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- CREATE INDEX - creates an index (search key)
- DROP TABLE - deletes a table
- DROP INDEX - deletes an index
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- SELECT - extracts data from a database
- UPDATE - updates data in a database

DATA DEFINITION
- CREATE tables, views, database …
- ADD or ALTER columns
- DROP tables, views, …

cols

ID	Fname	Lname	email	
1	Wagner	Morais		
2	Jesper	Hakerod		

rows

DATA MANIPUPLATION
- SELECT
- INSERT
- UPDATE
- DELETE

```
SELECT t1.column1, t2.column2, ...
FROM table_name1 t1
WHERE condition1 AND columnN LIKE pattern OR columnN IN (value1, ...)
JOIN table_name2 t2 ON t1.column_name = t2.column_name;
ORDER BY t1.column1, t2.column2, ... ASC|DESC;

INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);

UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;

DELETE FROM table_name
WHERE condition;
```

Week	Where	TLA	TLA	Content
w4	Tue, Jan 21 at 11:15	R4318	Course introduction	ILO, TLAs, Literature, Ats, ...
w4	Wed, Jan 22 at 15:15	R4318	Lecture 1	Intro to DB and DBMS area
w4	Fri, Jan 24 at 10:15	R4318	Lecture 2	Intro to DB and DBMS area
w5	Mon, Jan 27 at 15:15	R4318	Lecture 3	DB Design
w5	Wed, Jan 29 at 08:15	R4318	Lecture 4	DB Design
w5	Wed, Jan 29 at 10:15	R3140	Lab 1	Intro to DB design (ER and Rel Model)
w6	Tue, Feb 04 at 13:15	R4318	Lecture 5	Rel DB Design
w6	Wed, Feb 05 at 08:15	R3140	Lecture 6	Rel DB programming (SQL)
w6	Thu, Feb 06 at 08:15	R4318	Lab 2	Rel diagram
w7	Mon, Feb 10 at 13:15	R4318	Lecture 7	Rel DB programming (SQL)
w7	Tue, Feb 11 at 13:15	R4129	Lecture 8	Rel DB programming (SQL)
w7	Wed, Feb 12 at 08:15	R3140	Lab 3	DB programming
w8	Mon, Feb 17 at 10:15	R4129	Lecture 9	Intro Web
w8	Mon, Feb 17 at 13:15	R4318	Lecture 10	HTML
w8	Tue, Feb 18 at 13:15	R3140	Lab 4	DB programming
w9	Mon, Feb 24 at 10:15	R4318	Lecture 11	HTML
w9	Mon, Feb 24 at 13:15	R4318	Lecture 12	CSS
w9	Tue, Feb 25 at 13:15	R4318	Project Intro	
w9	Wed, Feb 26 at 10:15	Zoom	Lab 5	Static web - HTML
w10	Mon, Mar 03 at 13:15	R4318	Lecture 13	CSS
w10	Tue, Mar 04 at 11:15	R4318	Lecture 14	DB Review + JavaScript
w10	Wed, Mar 05 at 08:15	Zoom	Lab 6	Static web - HTML + CSS
w11	Mon, Mar 10 at 13:15	R4318	Lecture 15	JavaScript
w11	Tue, Mar 11 at 13:15	R4318	Lecture 16	Dynamic Web - PHP
w11	Wed, Mar 12 at 08:15	D308	Proj. Sup. 1	
w12	Fri, Mar 21 at 09:00		Exam 2202 Mand. Reg. Prelim.	DB Exam 2202
w13	Mon, Mar 24 at 10:15	R4318	Lecture 17	Dynamic Web - PHP
w13	Tue, Mar 25 at 13:15	R4318	Lecture 18	Dynamic Web - PHP
w13	Wed, Mar 26 at 08:15	Zoom	Lab 7	Dynamic Web - PHP + MySQL
w14	Mon, Mar 31 at 10:15	R4129	Lecture 19	Dynamic Web - PHP + MySQL
w14	Tue, Apr 01 at 11:15	R4129	Lecture 20	PHP Validation, State, Security
w14	Wed, Apr 02 at 08:15	Zoom	Lab 8	JavaScript, jQuery and AJAX
w15	Mon, Apr 07 at 08:15	D308	Proj. Sup. 2	
w15	Wed, Apr 09 at 13:15	Zoom	Lab 9	Dynamic Web - PHP + PostgreSQL
w17	Wed, Apr 23 at 08:15	D308	Proj. Sup. 3	
w19	Thu, Maj 08 at 08:15	D308	Proj. Sup. 4	
w21	Mon, Maj 19 at 10:15	R4129	Lecture 21	Web Review
w21	Wed, Maj 21 at 08:15	D308	Prj Seminar	
w22	Fri, Maj 30 at 09:00		Exam 2104 Mand Reg. Prelim.	Web exam 2104
w33	Mon, Aug 11 at 09:00		Re Exam 2202 Mand Reg. Prelim.	Web exam 2202
w33	Fri, Aug 15 at 09:00		Re Exam 2104 Mand Reg. Prelim.	Web exam 2104

Databases

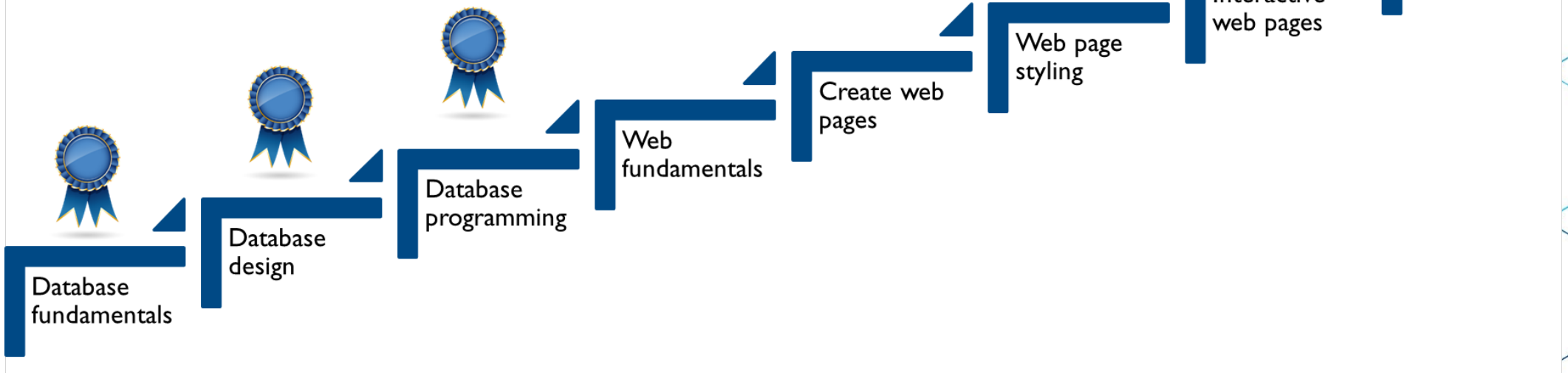
Web

Client-side development

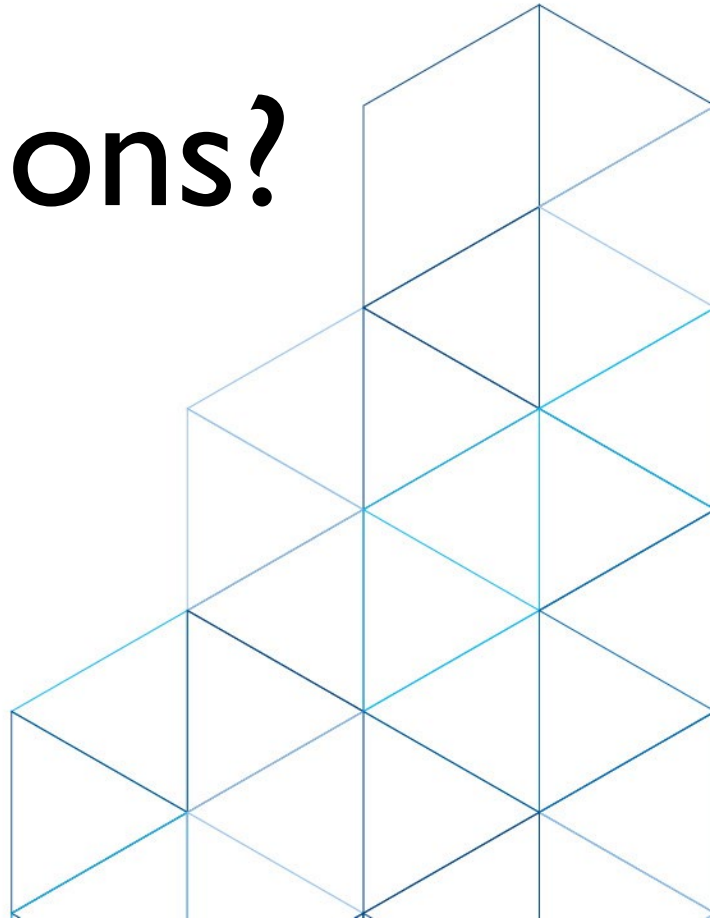
Server-side development

Course objectives

- The aim of this course is for students to **develop the knowledge and skills essential to modern web development**.
- Through the course, students will develop an **understanding of web system components and database concepts and practices** crucial for **database-driven web development**.
- Students will also acquire and practice **skills in developing and deploying a simple dynamic web system**.
- It is a **journey** from "beginner" to "comfortable" web developer.



Comments or Questions?



HALMSTAD UNIVERSITY

Objectives

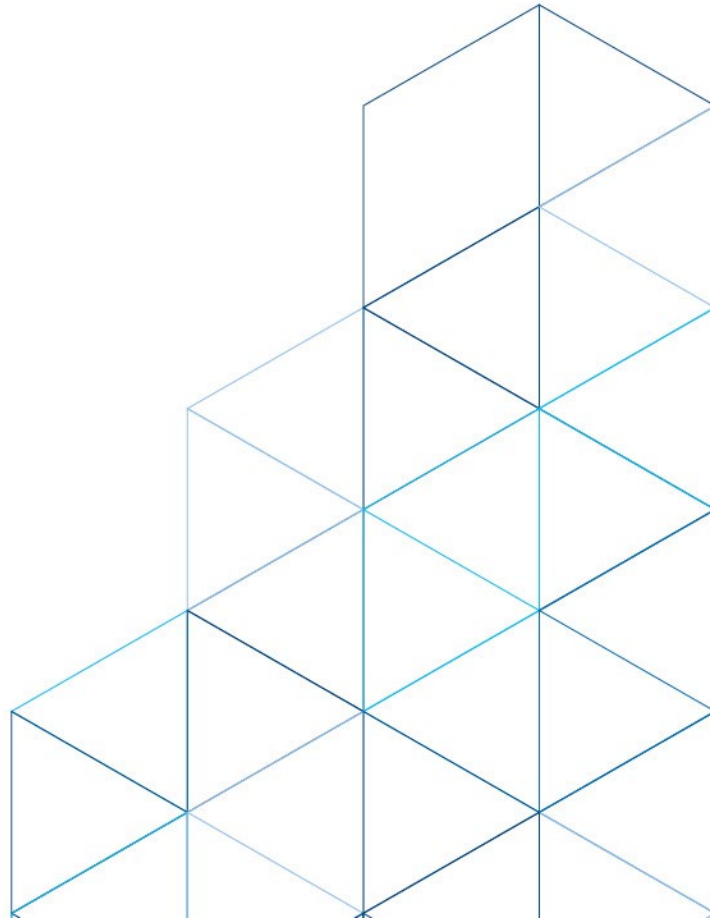
- Purpose
  - Discuss Web Development
- Describe the evolution of the Internet and the Web
- Discuss the Client-Server model
- Understand how the Web works
- Discuss fundamental protocols enabling the web
- Discuss the purpose of URL, HTTP, DNS, HTML
- Understand web browsers and servers

- Preparation
  - Chapter 1 gives an introduction to Web development.
  - Chapter 2 gives an overview about how the web works.
  - Although we use and follow the structure of the coursebook, in the Lectures, we cover parts of:
    - MDN - Learn Web Development
      - How the Web works (recommended reading)
      - Web mechanics (recommended reading)
      - MDN Web Docs Glossary

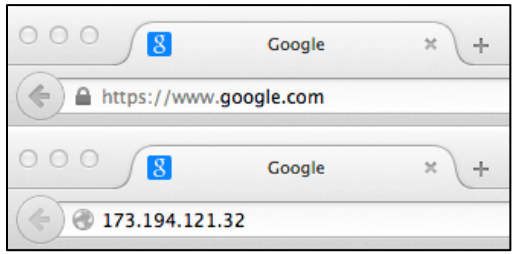
HALMSTAD UNIVERSITY

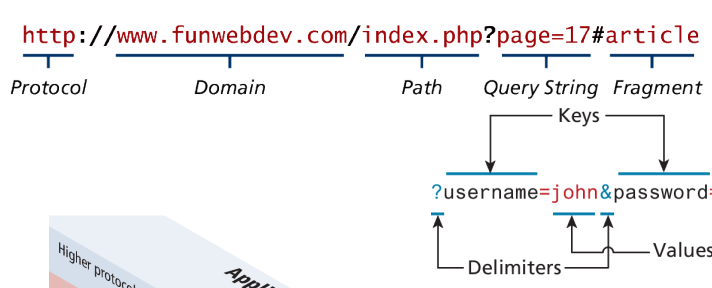
Terms

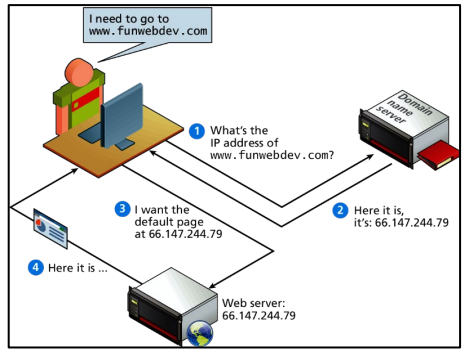
- URL
- HTTP
- Web server
- HTML
- Web browser
- Web page vs web site
- Search engine
- Web Applications vs Desktop Applications
- Static web pages vs dynamic web pages
- DNS

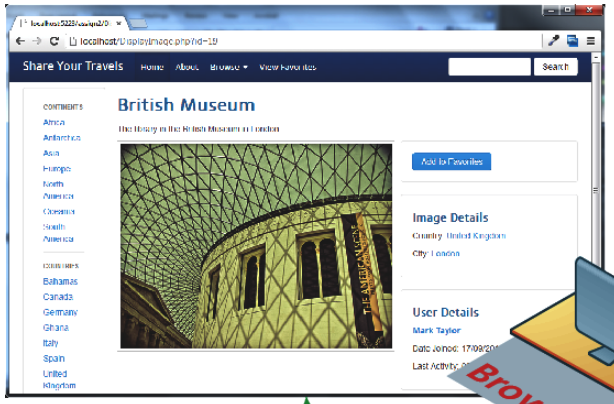


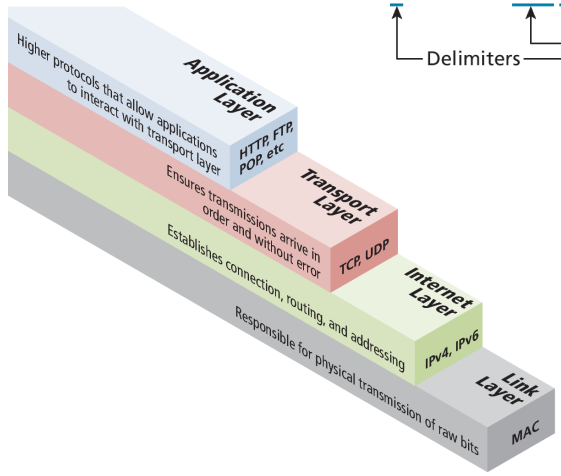
HALMSTAD UNIVERSITY

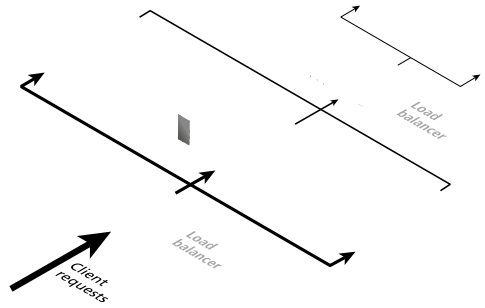

Google
https://www.google.com
Google
173.194.121.32

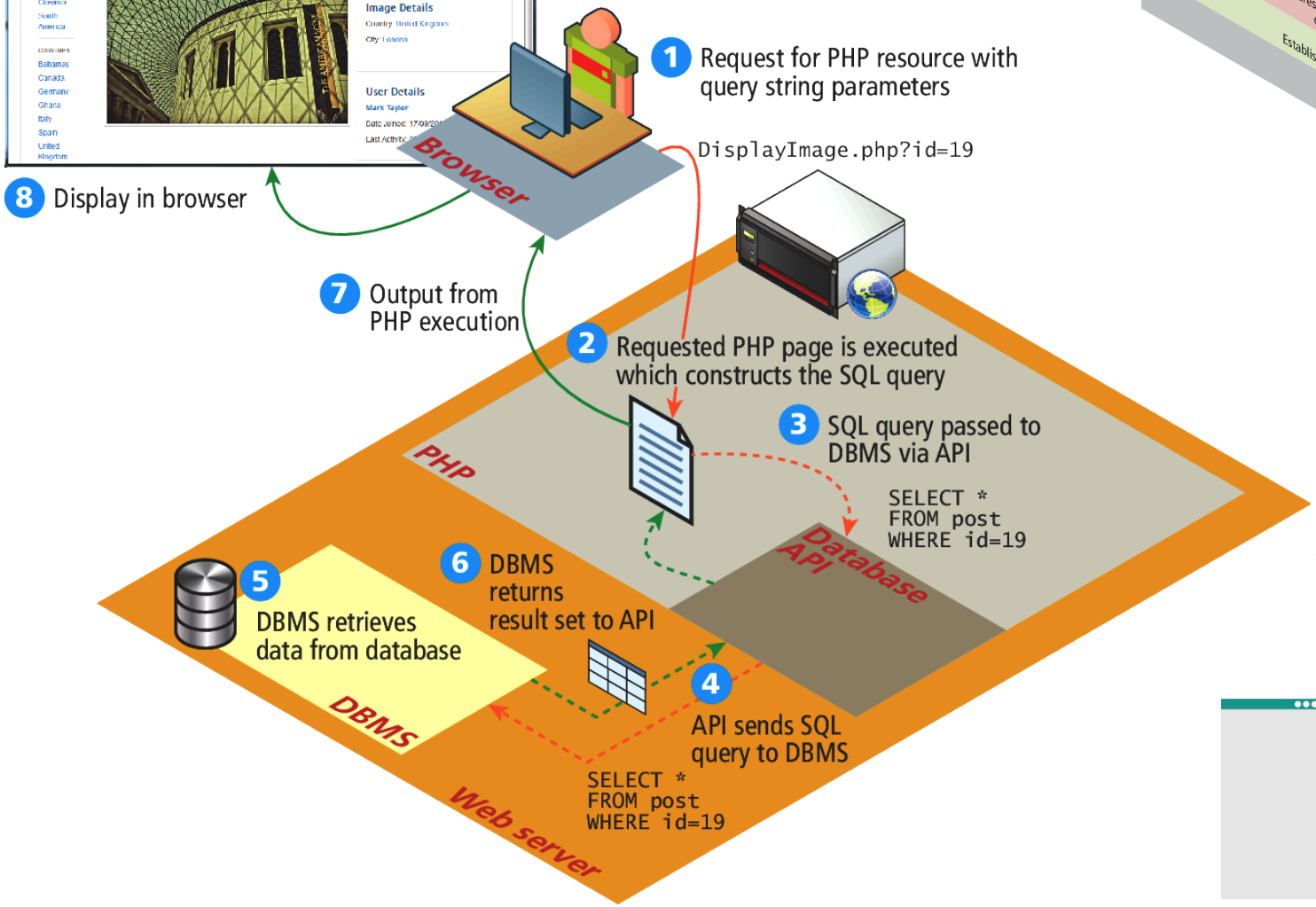

http://www.funwebdev.com/index.php?page=17#article
Protocol
Domain
Path
Query String
Fragment
Keys
?username=john&password=abcdefg
Delimiters
Values

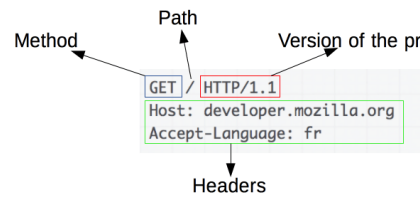

I need to go to www.funwebdev.com
What's the IP address of www.funwebdev.com?
Domain name server
I want the default page at 66.147.244.79
Here it is, it's: 66.147.244.79
Here it is ...
Web server: 66.147.244.79

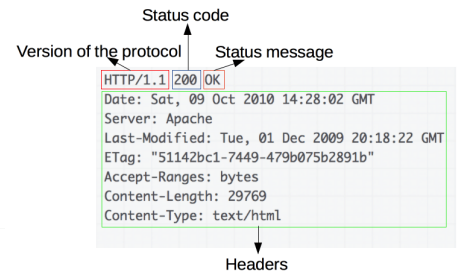

Share Your Travels
Search
British Museum
Image Details
User Details

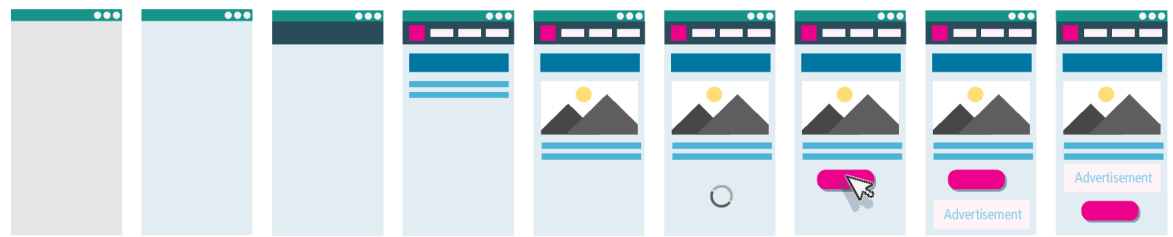

Application Layer
HTTP, FTP, POP, etc
Higher protocols that allow applications to interact with transport layer
Transport Layer
TCP, UDP
Ensures transmissions arrive in order and without error
Internet Layer
IPv4, IPv6
Establishes connection, routing, and addressing
Link Layer
MAC
Responsible for physical transmission of raw bits


Load balancer
Load balancer
Client requests


1 Request for PHP resource with query string parameters
DisplayImage.php?id=19
Browser
8 Display in browser
7 Output from PHP execution
2 Requested PHP page is executed which constructs the SQL query
PHP
3 SQL query passed to DBMS via API
SELECT *
FROM post
WHERE id=19
Database API
6 DBMS returns result set to API
5 DBMS retrieves data from database
DBMS
4 API sends SQL query to DBMS
SELECT *
FROM post
WHERE id=19
Web server

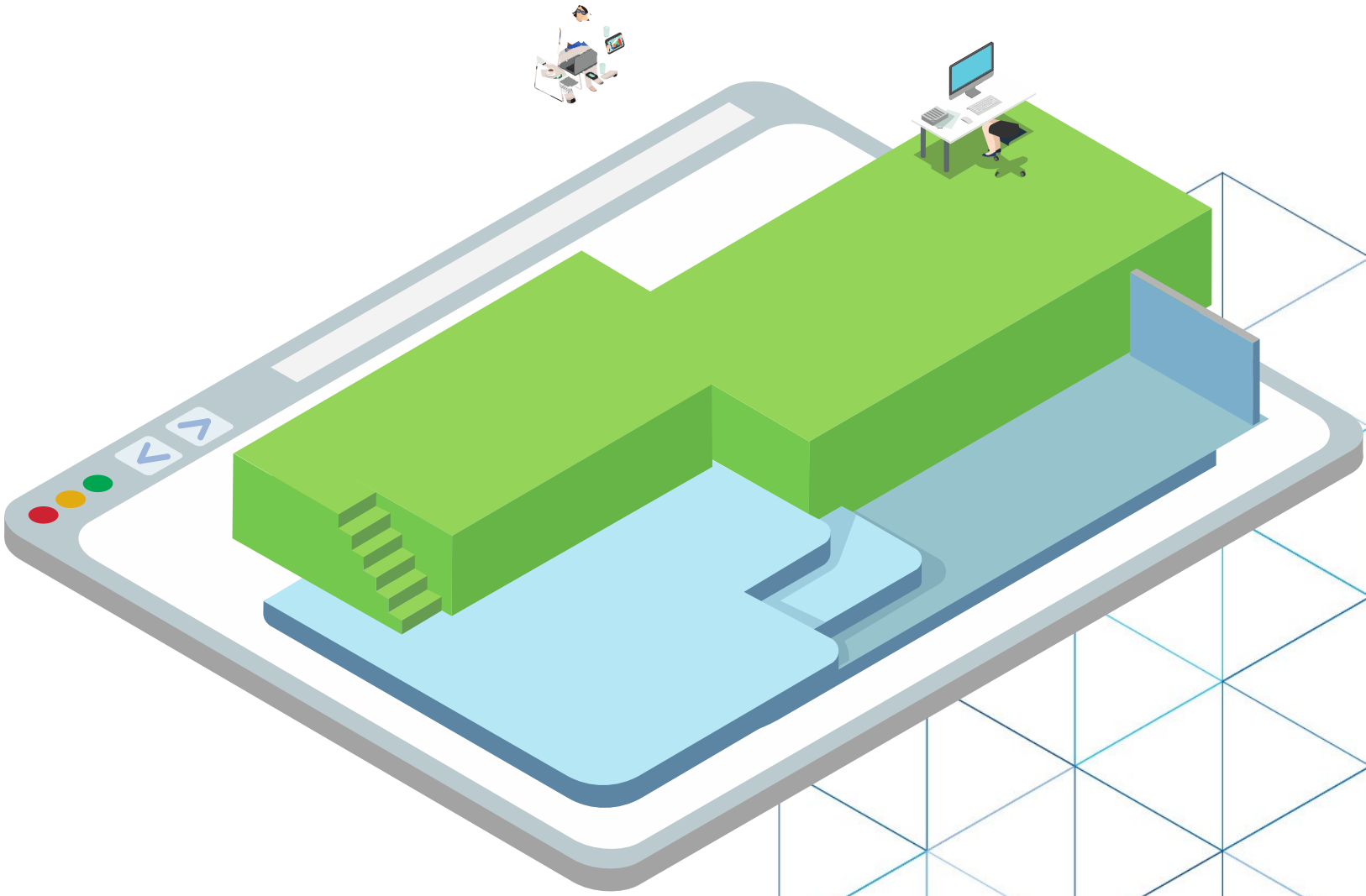

Method
Path
Version of the protocol
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
Headers


Status code
Version of the protocol
Status message
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html
Headers


Advertisement
Advertisement

Chapter 1 - Introduction to Web Development

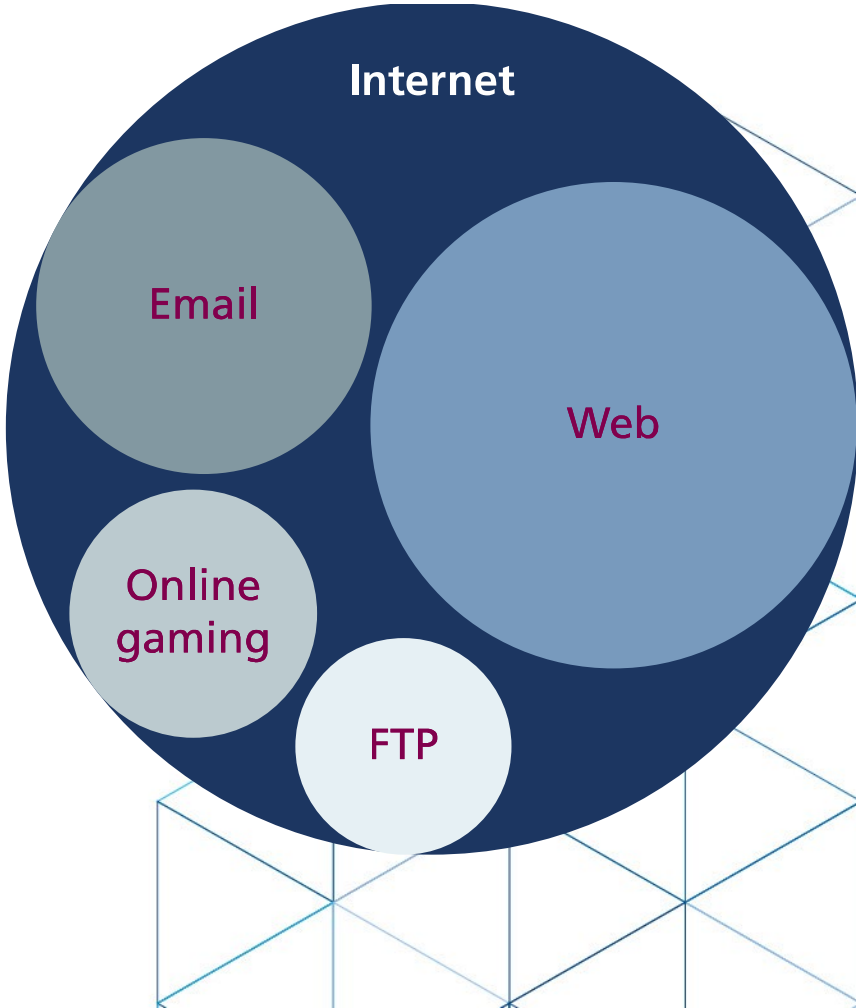
- Web development requires people with complementary but distinct expertise working together toward a single goal.
  - Graphic designer
    - Application of good graphic design strategies.
  - Database administrator
    - Simple interface to an underlying database.
  - Software engineers and programmers
    - Classic software development task with phases and deliverables.
  - Systems administrator
    - System that has to be secured from attackers.



HALMSTAD UNIVERSITY

Internet vs WWW?

- The Internet and WWW are different (but related) things
  - Circuit Switched Networks
    - Inefficient use of bandwidth and difficult to scale
  - Packet switching
    - Does not require a continuous connection
  - 1960s ARPANET
  - 1969 Internet - Global computer networks
  - 1981 TCP/IP was introduced to unify disparate networks
  - 1983 TCP/IP was adopted across all of ARPANET
  - 1992 Sr. Tim Berners-Lee publishes the main features of the web.



HALMSTAD UNIVERSITY

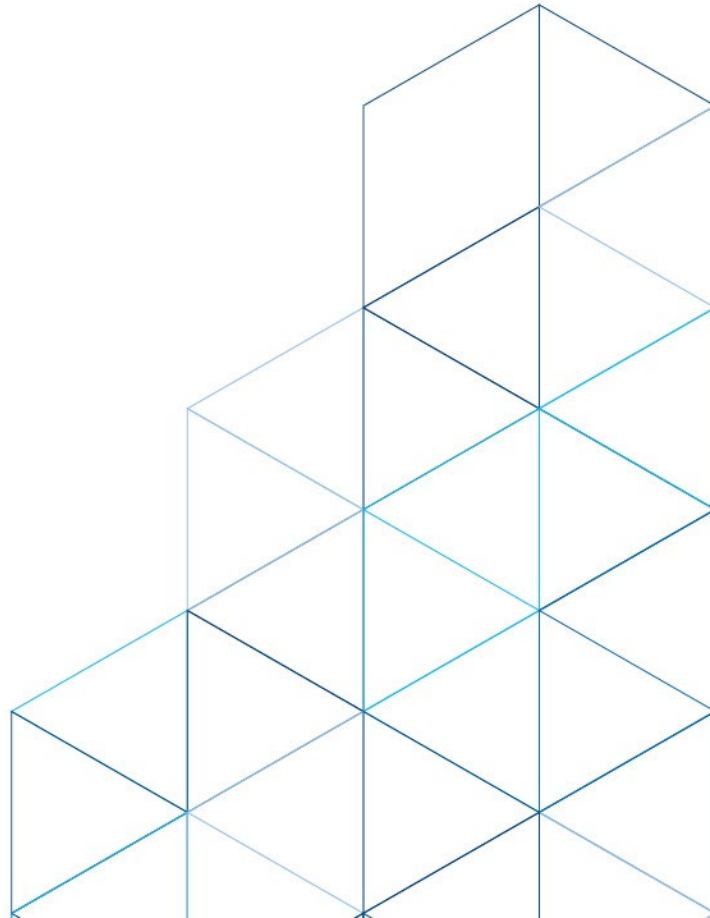
Web - Essential elements I

- A Uniform Resource Locator (**URL**) to uniquely identify a resource on the WWW.
- The Hypertext Transfer Protocol (**HTTP**) to describe how requests and responses operate
- A software program (**web server software**) that can respond to HTTP requests.
- Hypertext Markup Language (**HTML**) to publish documents.
- A program (a **browser**) that can make HTTP requests to URLs and that can display the HTML it receives.

- A **web page** is "a document which can be displayed in a web browser".
- A **website** is "a collection of web pages".
- A **search engine** is a web service that helps you find other web pages.

Web Applications vs Desktop Applications

- Advantages of web applications
- Disadvantages of web applications



HALMSTAD UNIVERSITY

Web Applications vs Desktop Applications

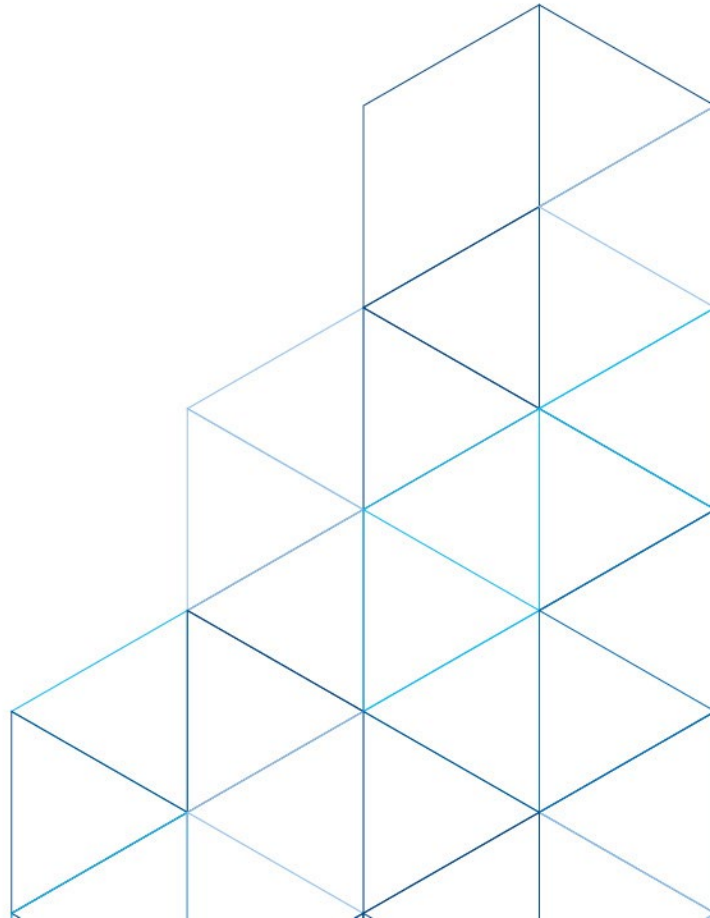
- Advantages of web applications
  - They can be accessed from any Internet-enabled computer.
  - They can be used with different operating systems and browser applications.
  - They are easier to roll out program updates since only software on the server needs to be updated as opposed to every computer in the organization using the software.
  - They have a centralized storage on the server, which means fewer security concerns about local storage (which is important for sensitive information such as health care data).

- Disadvantages of web applications
  - Requirement to have an active Internet connection
  - Security concerns about sensitive private data being transmitted over the Internet.
  - Concerns over the storage, licensing, and use of uploaded data.
  - Problems with certain websites not having an identical appearance across all browsers.
  - Restrictions on software from being installed and hardware from being accessed (like Adobe Flash on iOS).
  - Additional plugins might interfere with JavaScript, cookies, or advertisements.

Static web pages vs dynamic web pages

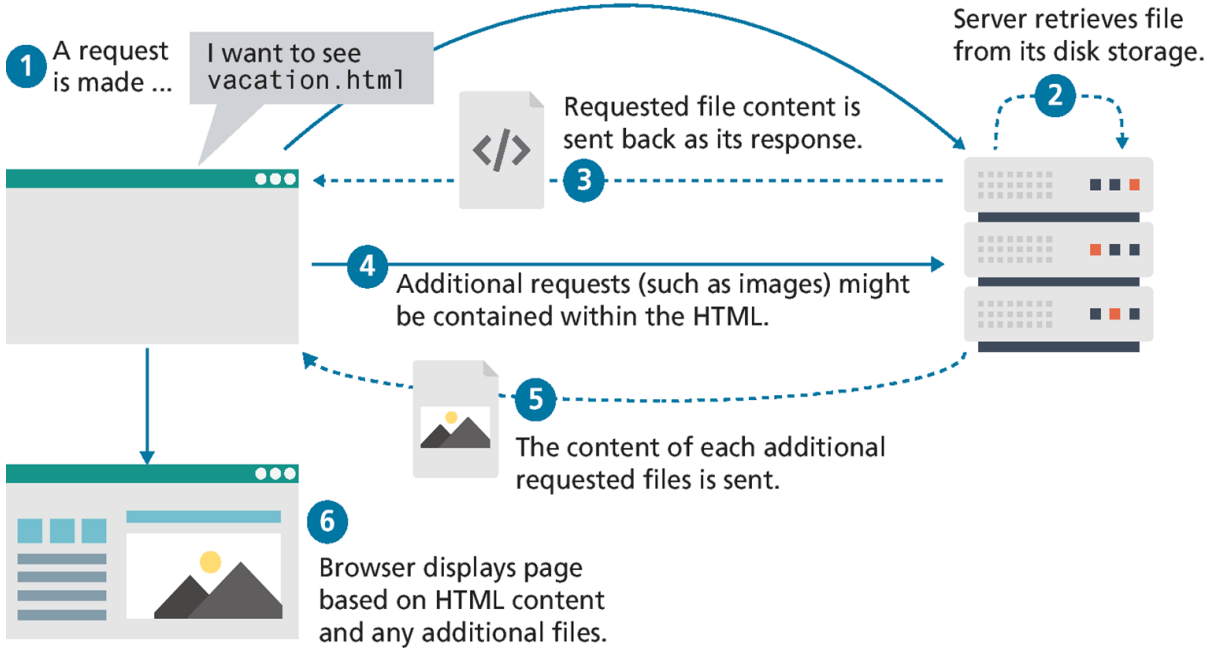
- Static web pages

- Dynamic web page

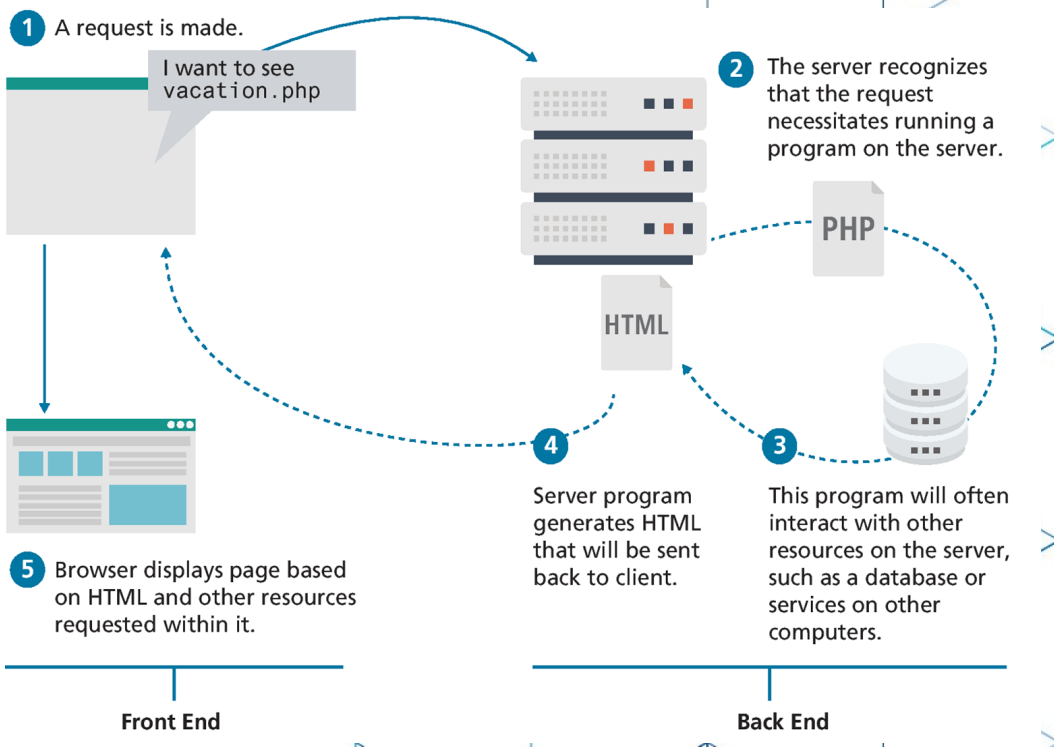


Static web pages vs dynamic web pages

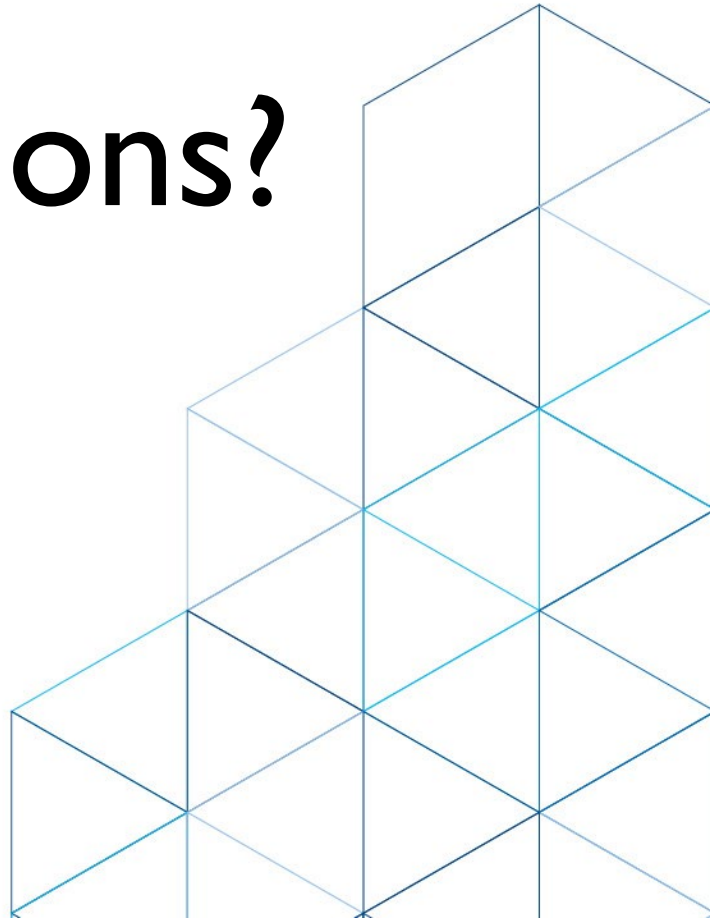
- Static web pages
  - Delivered to the user exactly as stored in the web server.
  - Web pages look identical for all users at all times.



- Dynamic web page
  - Dynamically constructed in the web server by a web application and, in the client, its is dynamically rendered in the web browser.



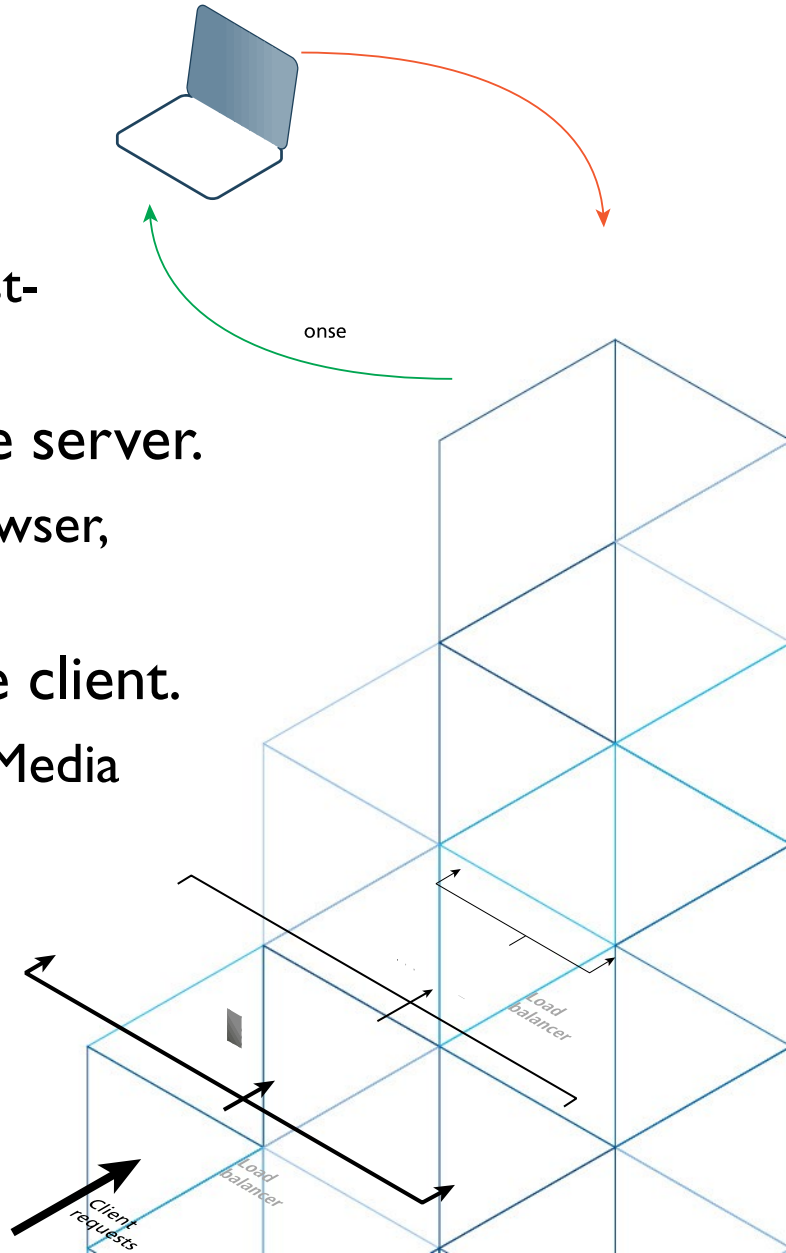
Comments or Questions?



HALMSTAD UNIVERSITY

Web - Essential elements II

- The Client-Server Model
  - Web applications use a Client/Server computing model in a request-response loop over a network.
- Client: requests some type of service (such as a file) from the server.
  - A web client connects to internet when needed, runs the web browser, requests and receives web documents.
- Server: responds the request and transmits the results to the client.
  - Web Servers, Application Servers, Database Servers, Mail Servers, Media Servers, Authentication Servers
  - A web server is continually connected to internet, runs the web server software, receives and responds to http requests.

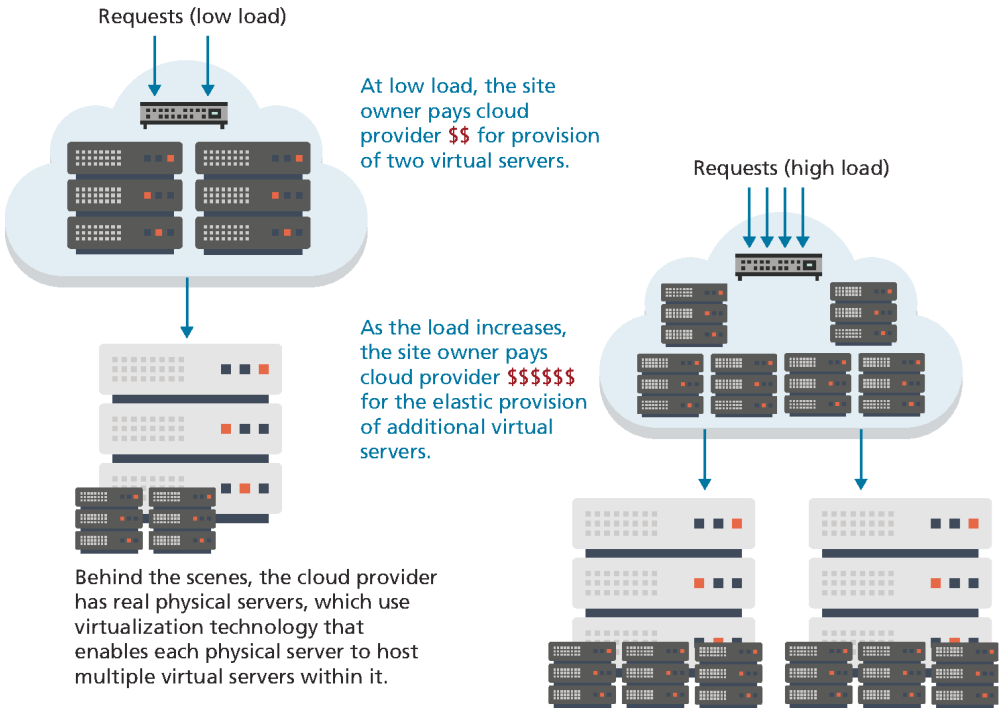


HALMSTAD UNIVERSITY

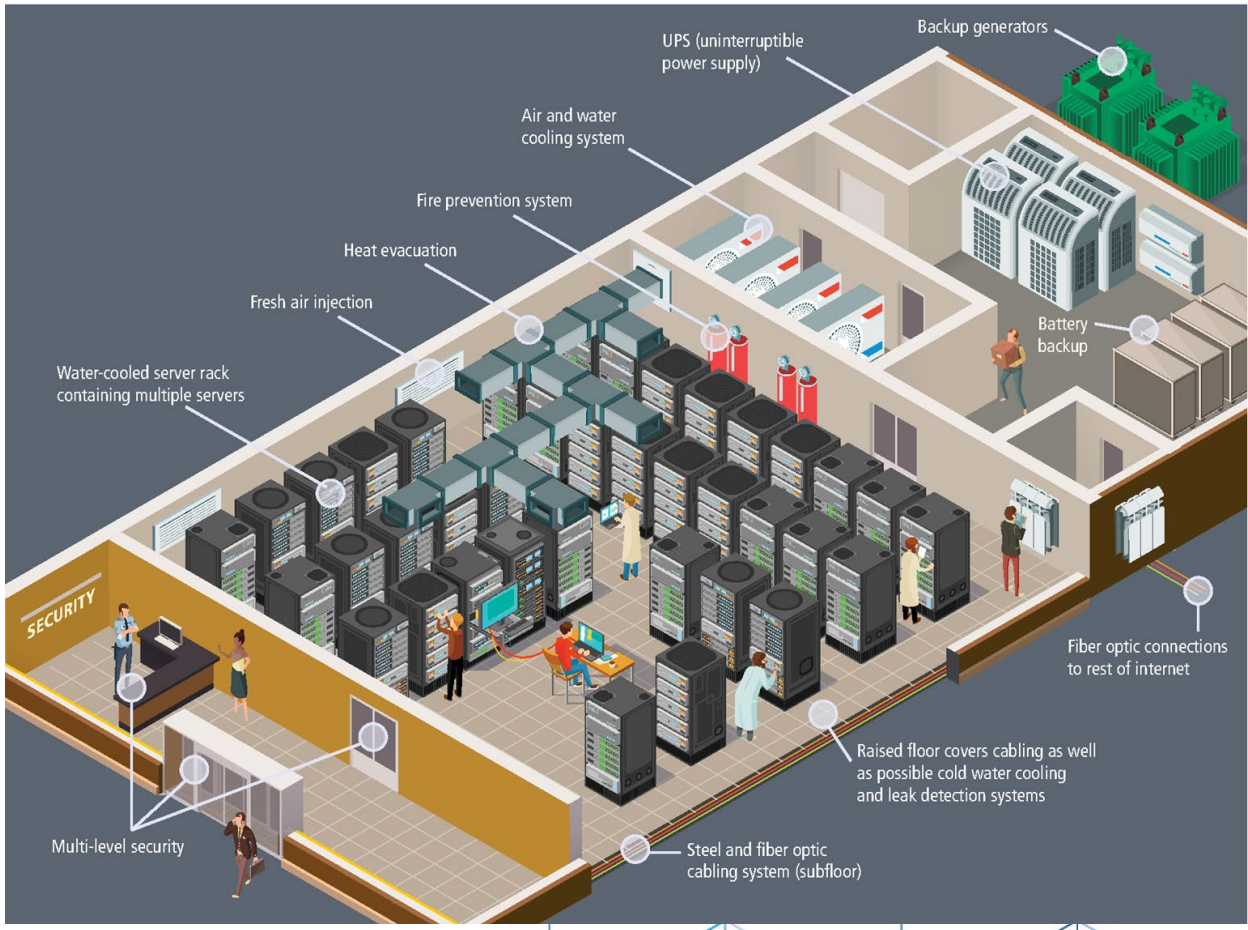
Real World Server Installations

(Not one server, but a cluster of multiple machines working together.)

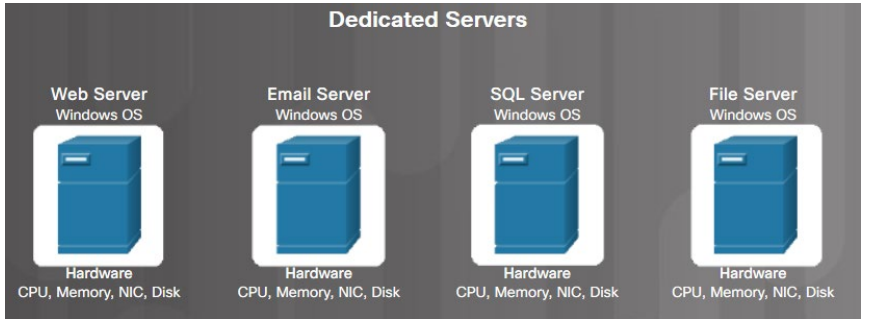
- Server Farm
- Load Balancers
- Failover Redundancy
- Server Racks
- Cloud Services



- Data Centers



Cloud Computing and Virtualization



- Traditionally, organizations delivered applications and services using powerful dedicated servers.
  - Dedicated servers are equipped with large amounts of RAM, powerful CPUs, and multiple large storage devices.
  - Disadvantages include: wasted resources, single-point of failure, and server sprawl.

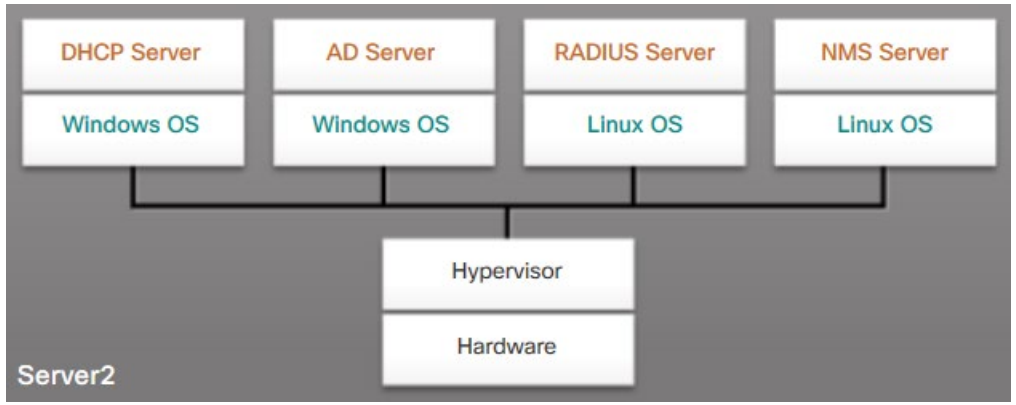
- Virtualization enables a single computer to host multiple independent virtual computers called virtual machines (VM) that share the host computer hardware.
- Virtualization software separates the actual physical hardware from the VM instances.
- An image of a VM can be saved as a file and then be re-started when required.
- Cloud computing separates the applications from the hardware.
- Service providers such as Amazon Web Services (AWS) own and manage the cloud infrastructure.

HALMSTAD UNIVERSITY

Virtualization is the foundation which supports cloud computing.

Server Virtualization

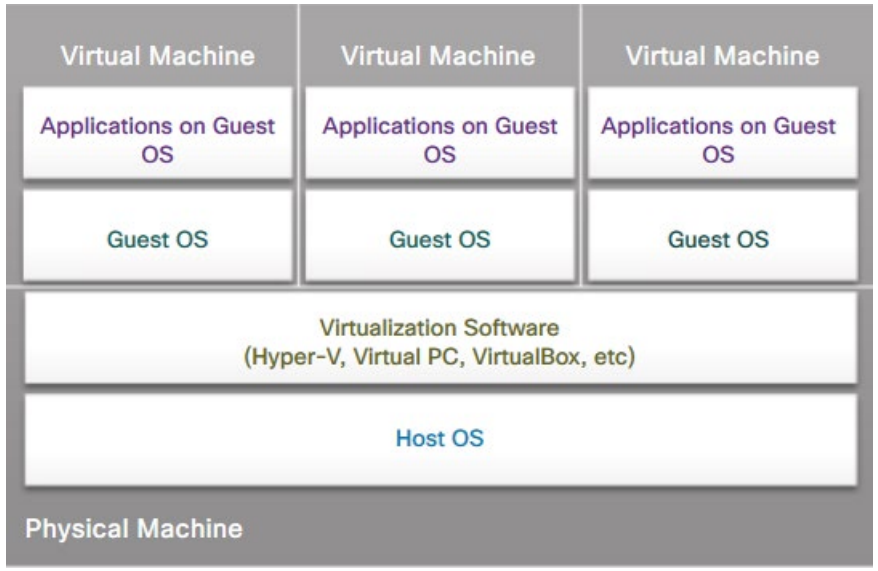
- Server virtualization takes advantage of idle resources to reduce the number of servers required.
- A program called the hypervisor is used to manage the computer resources and various VMs.
- It provides VMs access to the hardware in the physical machine such as CPUs, memory, disk controllers, and NICs.
- Each VM runs a complete and separate operating system.



- Advantages
  - Better use of resources
  - Less space required
  - Less energy consumed
  - Reduced cost
  - Faster server provisioning
  - Maximize server uptime
  - Improved disaster recovery
  - Support for legacy systems

Client-Side Virtualization

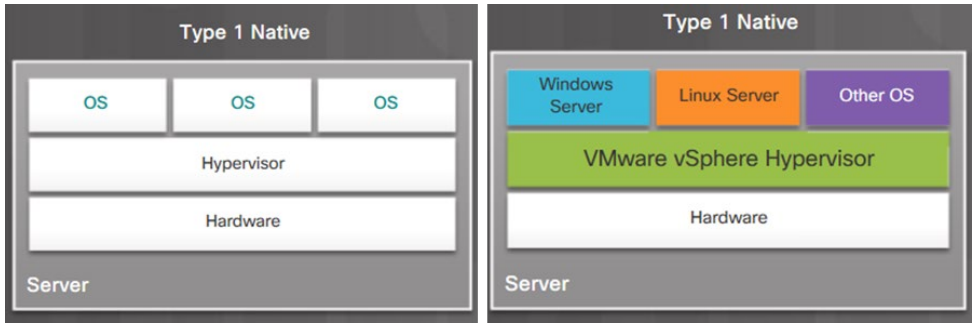
- Client-side virtualization enables users to run VMs on their local computer.
- It allows users to test new operating systems, software, or to run older software.
- Host computer – the physical computer controlled by a user.
- Host OS - the operating system of the host computer.
- Guest OS - the operating system that is running in the VM.



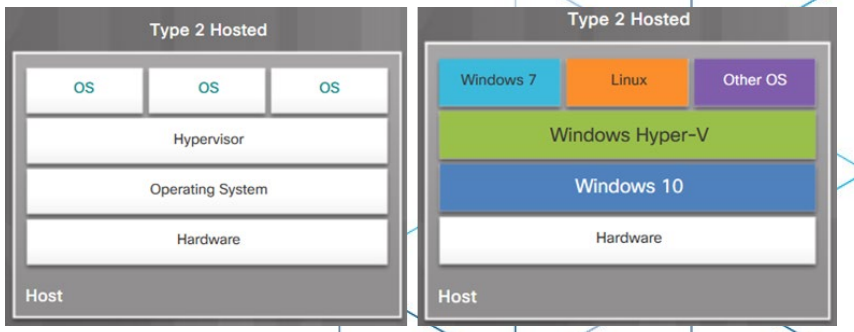
HALMSTAD UNIVERSITY

Type 1 and Type 2 Hypervisors

- Type 1 (native)
  - Type 1 hypervisor is typically used with server virtualization. For example, they are used in data centers and cloud computing.
  - Type 1 hypervisors run directly on the hardware of a host, and manage the allocation of system resources to VMs.
  - Type 1 hypervisors include VMware vSphere / ESXi, Xen, and Oracle VM Server.



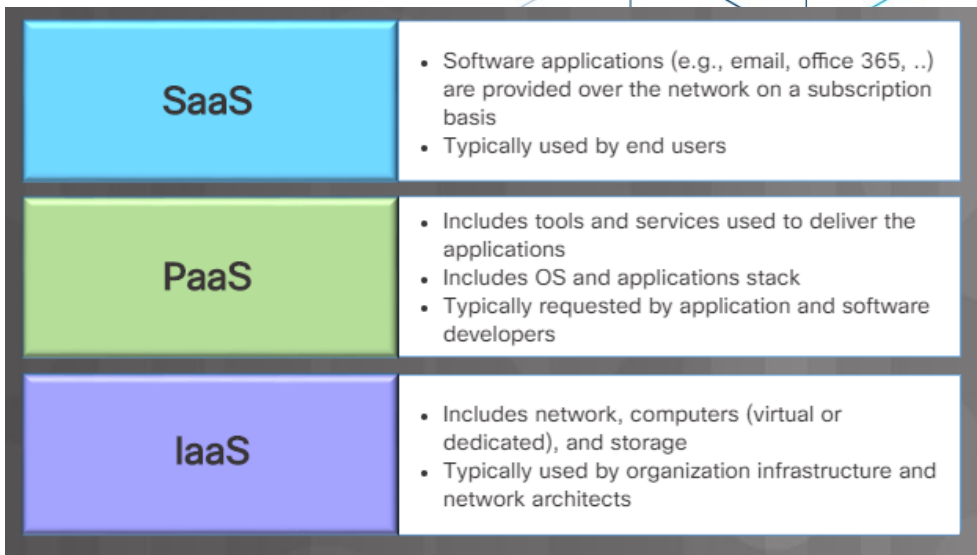
- Type 2 (hosted)
  - Type 2 hypervisors are commonly used with client-side virtualization.
  - Type 2 hypervisors work with the host computer to create and use multiple VMs.
  - Type 2 hypervisors include VMware Workstation, Windows Hyper-V, and Oracle VirtualBox.



HALMSTAD UNIVERSITY

Cloud Computing - Cloud Services

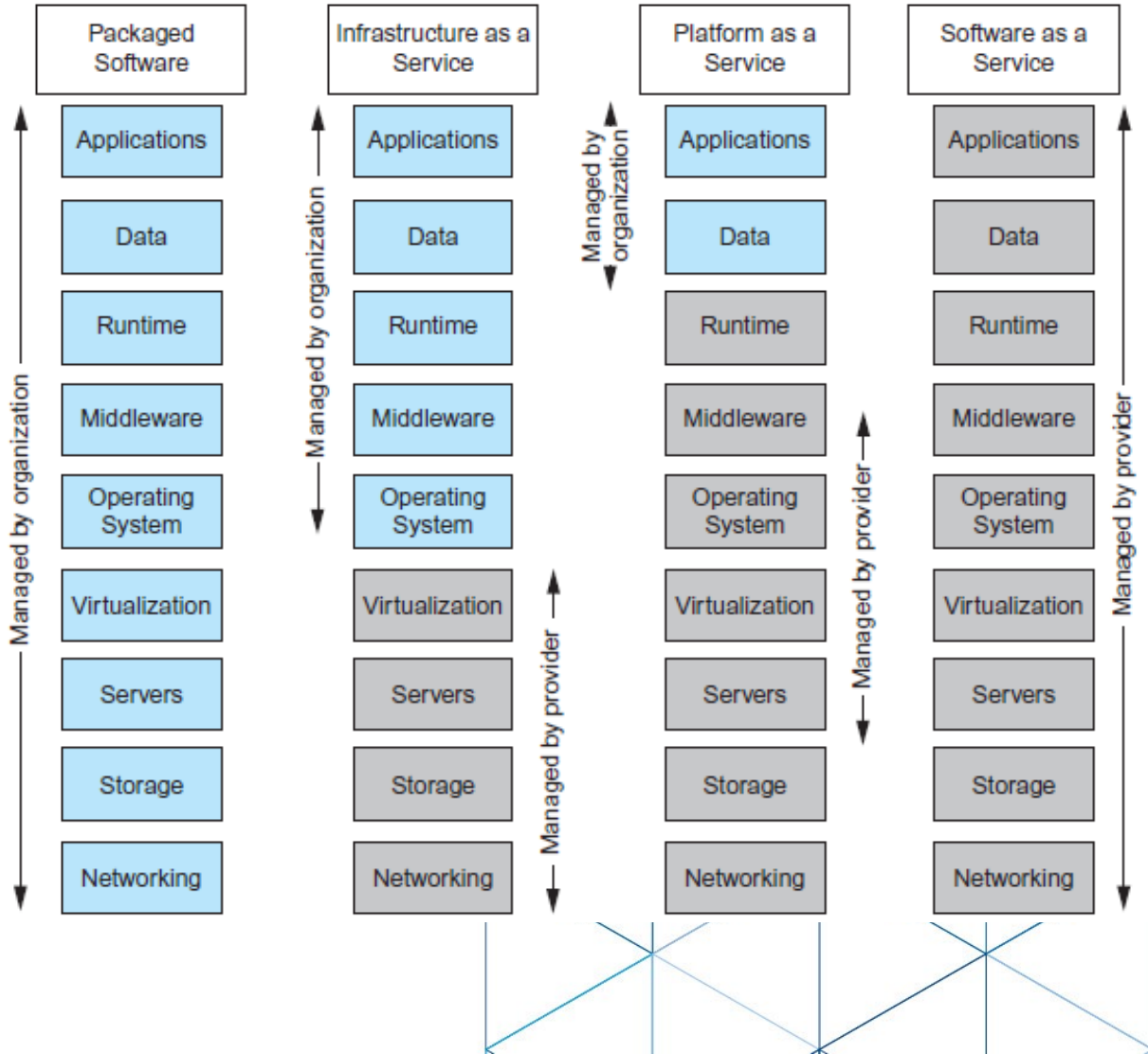
- "A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [National Institute of Standards and Technology (NIST)]
- Cloud service providers can provide various services tailored to meet customer requirements.
  - Software as a Service (SaaS)
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)
- Cloud service providers have extended the IaaS model to also provide IT as a service (ITaaS).



HALMSTAD UNIVERSITY

Comparison of Cloud Services Models

- Software as a Service (SaaS)
  - Software and data hosted on cloud.
  - Accessed via thin clients (e.g. web browser).
  - Example: Google's Gmail.
- Platform as a Service (PaaS)
  - Allows creation of web applications without buying/maintaining the software and underlying infrastructure.
  - Provider manages the infrastructure.
  - Customer controls deployment of applications and possibly configuration.
  - Ex: Google's App Engine and Microsoft's Azure.
- Infrastructure as a Service (IaaS)
  - Provider's offer servers, storage, network and operating systems – typically a platform virtualization environment – to consumers as an on-demand service, in a single bundle and billed according to usage.
  - Hosting websites.
  - Example: Amazon's Elastic Compute Cloud (EC2)



HALMSTAD UNIVERSITY

Cloud Computing Characteristics

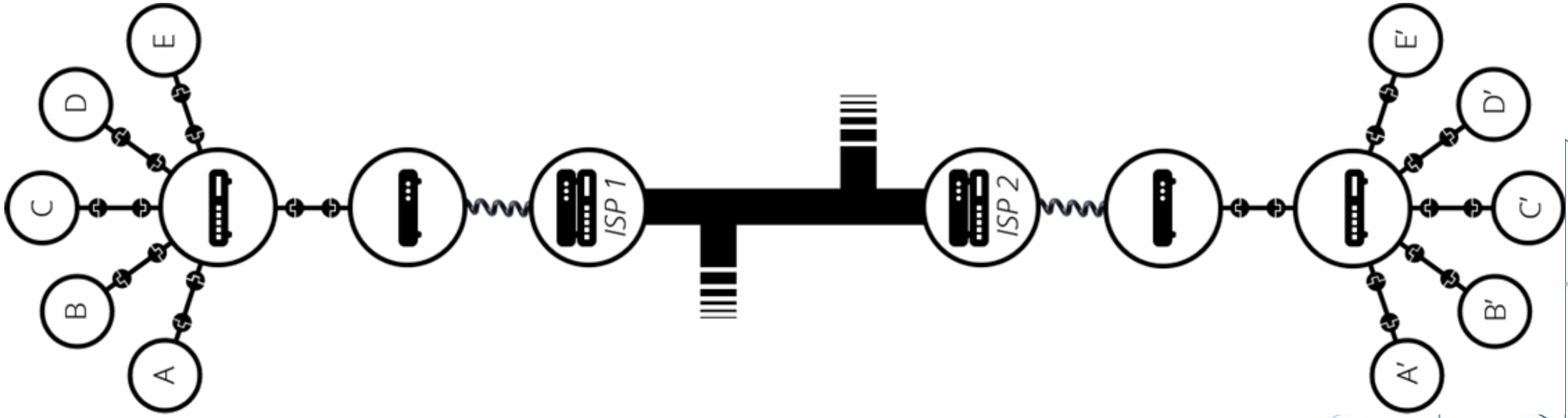
- On-demand self-service
  - Consumers can obtain, configure and deploy cloud services without help from provider.
- Broad network access
  - Accessible from anywhere, from any standardized platform (PC, smart phone)
- Resource pooling
  - Provider's computing resources are pooled to serve multiple consumers, with different physical and virtual resources (processing, memory, storage) dynamically assigned and reassigned according to consumer demand.
- Rapid elasticity
  - Provider's capacity caters for customer's spikes in demand and reduces risk of outages and service interruptions. Capacity can be automated to scale rapidly based on demand.
- Measured service
  - Provider uses a metering capability to measure usage of service (e.g. storage, processing, bandwidth, and active user accounts).

Cloud Computing - Benefits vs Risks

- Cost-Reduction: Avoid up-front capital expenditure.
- Scalability/Agility: resources on an as-needs basis.
- Providers can devote expertise & resources not affordable by customer, such as
  - Improved Security
  - Improved Reliability
  - Access to new technologies
- Faster development: Provider's platforms can provide many of the core services to accelerate development cycle.
- Large scale prototyping/load testing: Providers have the resources to enable this.
- More flexible working practices: Staff can access files using mobile devices.
- Increased competitiveness: Allows organizations to focus on their core competencies rather than their IT infrastructures.

- Network Dependency: Power outages, bandwidth issues and service interruptions.
- System Dependency: Customer's dependency on availability and reliability of provider's systems.
- Cloud Provider Dependency: Provider could become insolvent or acquired by competitor, resulting in the service suddenly terminating.
- Lack of control: Customers unable to deploy technical or organizational measures to safeguard the data. May result in reduced availability, integrity, confidentiality, intervenability and isolation.
- Lack of information on processing transparency

Chapter 2 - How the Web Works

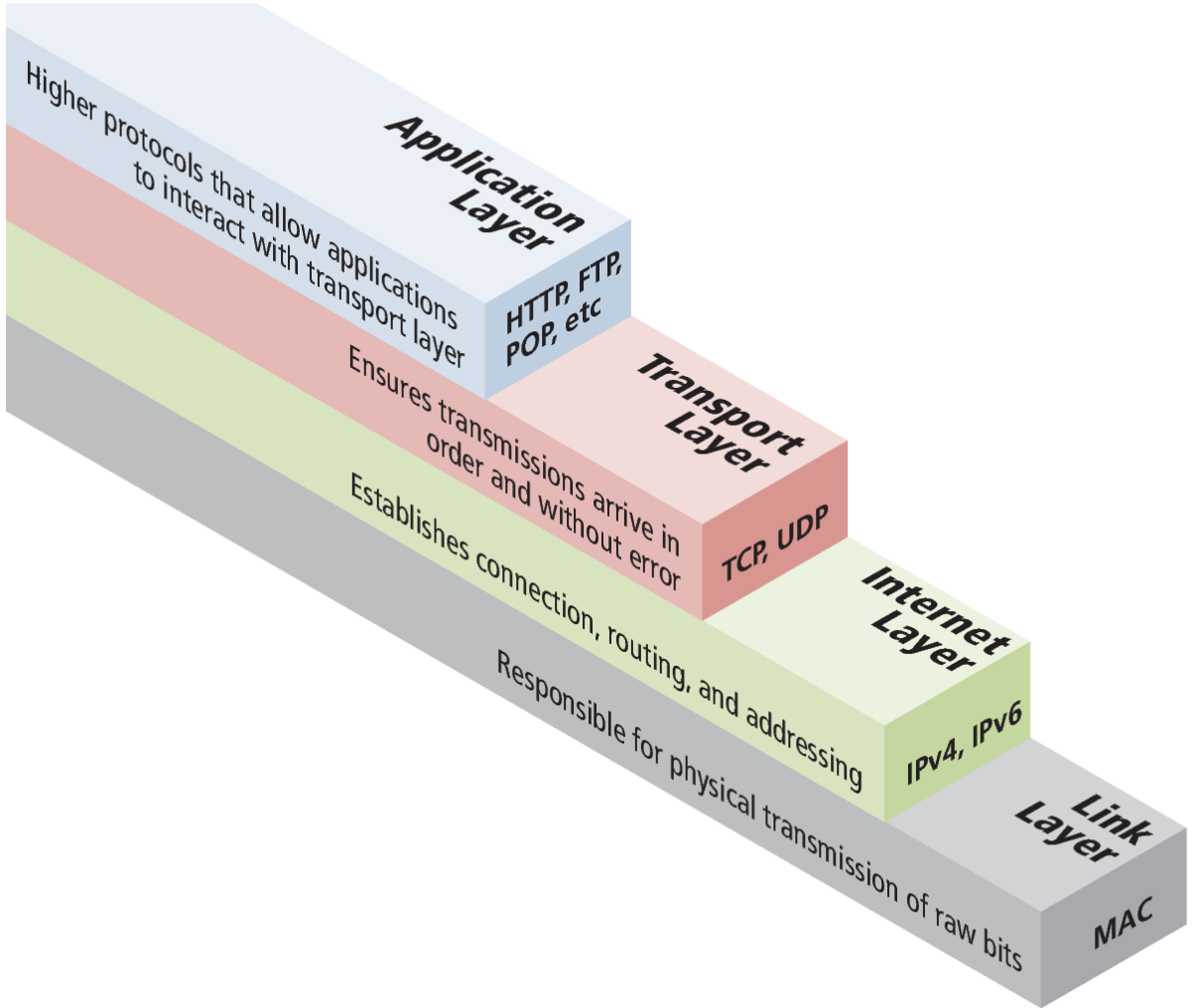


Source: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_does_the_Internet_work

HALMSTAD UNIVERSITY

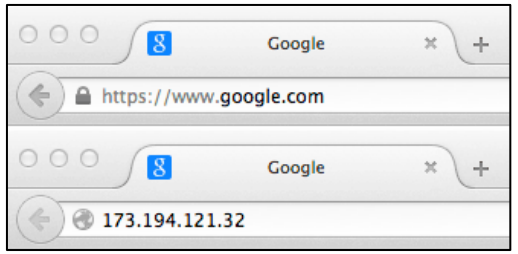
Layered Architecture and Internet Protocols

- The TCP/IP Internet protocols abstracted as a four-layer network model.



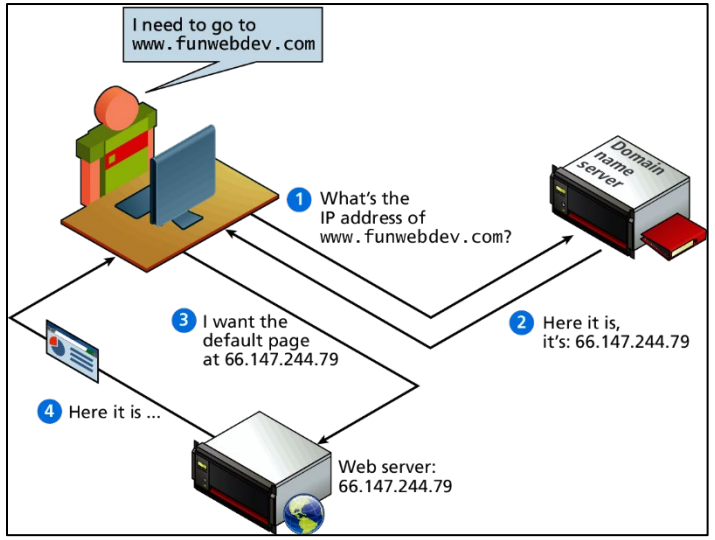
- Protocols
  - Rules that describe the methods used for clients and servers to communicate with each other over a network.
  - Handles issues like packet creation, transmission, reception, error detection, collisions, line sharing, and more.
- Application layer protocols
  - HTTP (80)
  - SSH (22)
  - FTP (21)
  - POP/IMAP/SMTP
  - DNS (53)

Domain Name System

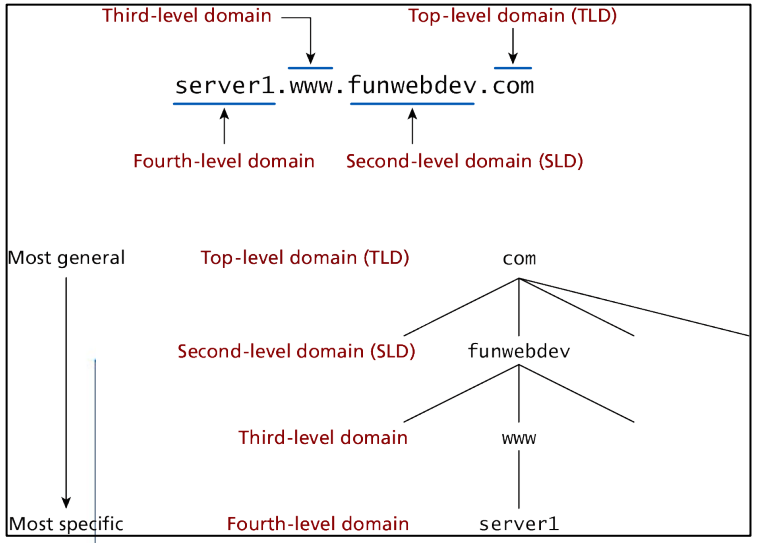


Source: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_does_the_Internet_work

- Instead of IP addresses, we use the Domain Name System (DNS).
- DNS translates a given URL to an IP address.
- The URL remains the same, but IP addresses can change.



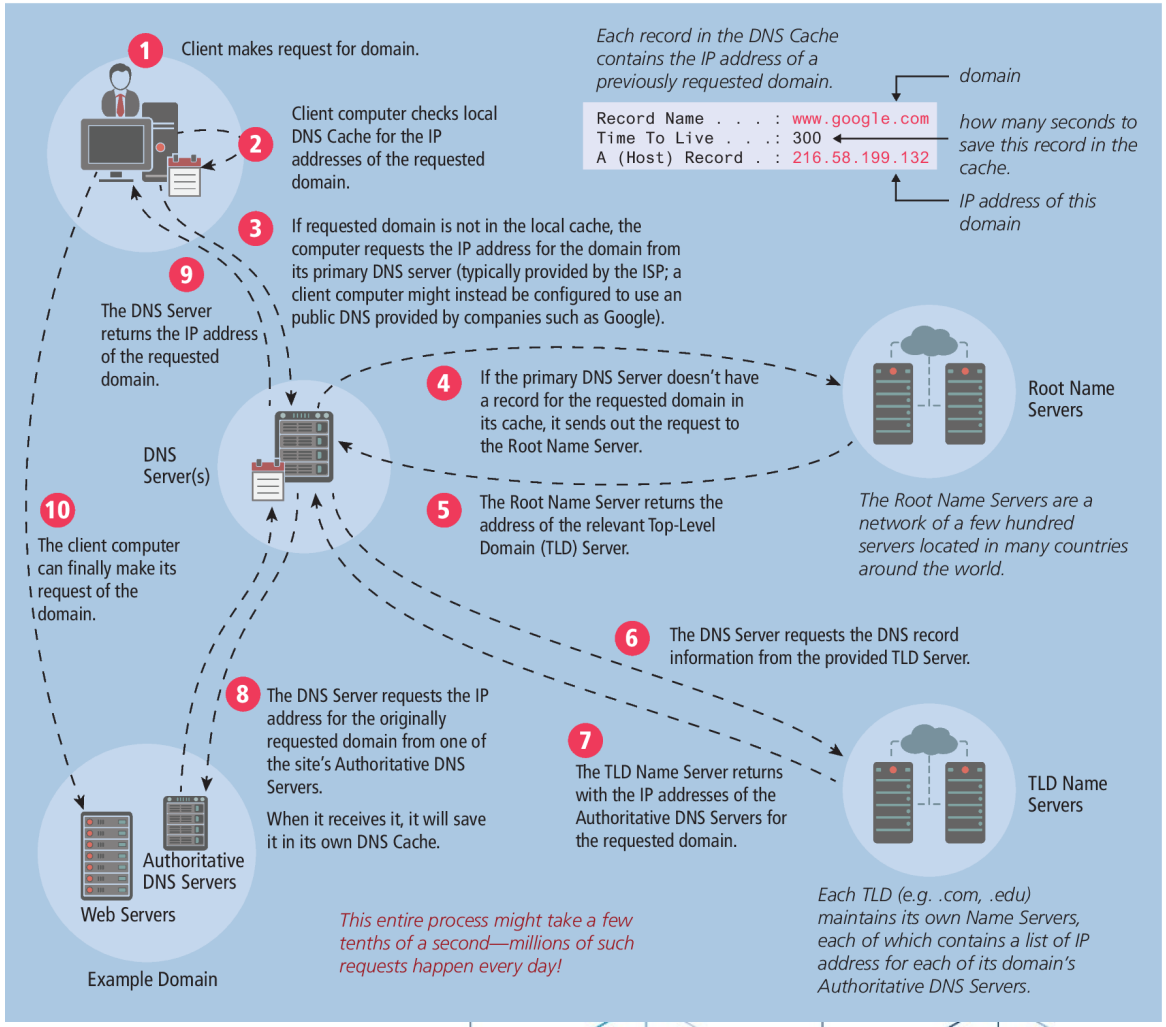
- A domain name can be broken down into several parts, which describe a hierarchy.
- All domain names have at least a top-level domain (TLD) name and a second-level domain (SLD) name



HALMSTAD UNIVERSITY

Address Resolution

1. Client makes request for domain
2. Client computer checks local DNS Cache
3. If requested domain is not in the local cache, the computer requests the IP address for the domain from its primary DNS server
4. If the primary DNS Server doesn't have a record for the requested domain in its cache, it sends out the request to the Root Name Server
5. The Root Name Server returns the address of the relevant Top-Level Domain (TLD) Server.
6. The DNS Server requests the DNS record information from the provided TLD Server.
7. The TLD Name Server returns with the IP addresses of the Authoritative DNS Servers for the requested domain.
8. The DNS Server requests the IP address for the originally requested domain from one of the site's Authoritative DNS Servers.
9. The DNS Server returns the IP address of the requested domain.
10. The client computer can finally make its request of the domain.

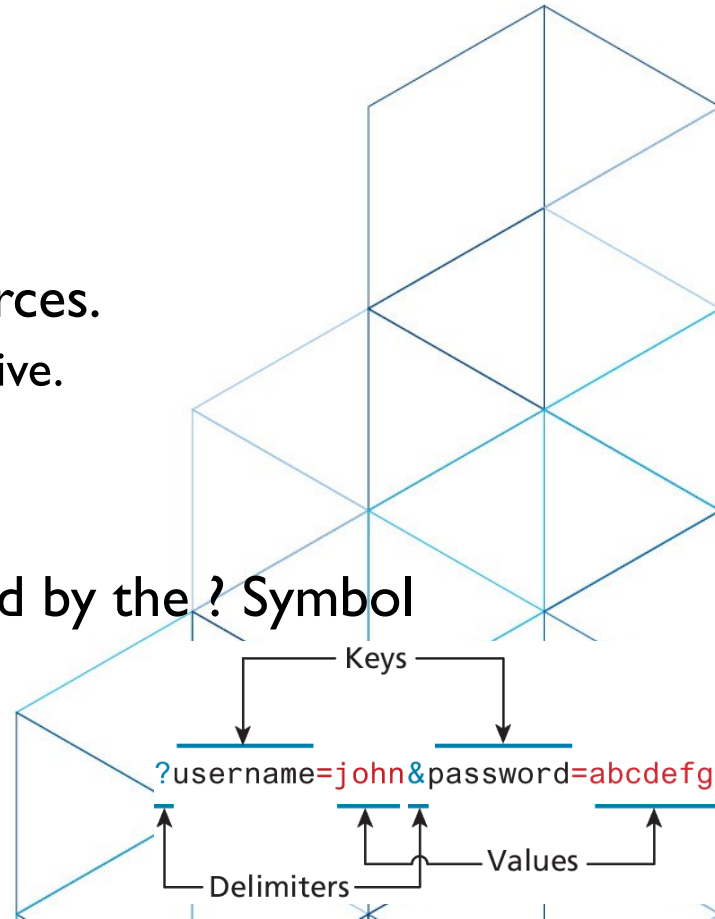


HALMSTAD UNIVERSITY

Uniform Resource Locators

- Represents the address of a resource on the Internet
- Normally in form of a **FQDN** (fully qualified domain name)

1. **Prototol**: FTP, SSH, HTTP, POP, IMAP, DNS, …
   - ftp://example.com/abc.txt - sends out an FTP request on port 21
   - http://example.com/abc.txt - transmits an HTTP request on port 80.
2. **Domain**: identifies the server from which we are requesting resources.
   - Since the DNS system is case insensitive, this part of the URL is case insensitive.
3. **Port**: connections to ports other than the defaults. domain:port
4. **Path**: somewhere on the server.
5. **Query string**: key-value pairs delimited by & symbols and preceded by the ? Symbol
6. **Fragment** identifier: location within the document



HALMSTAD UNIVERSITY

www, port number in URL and HTTP

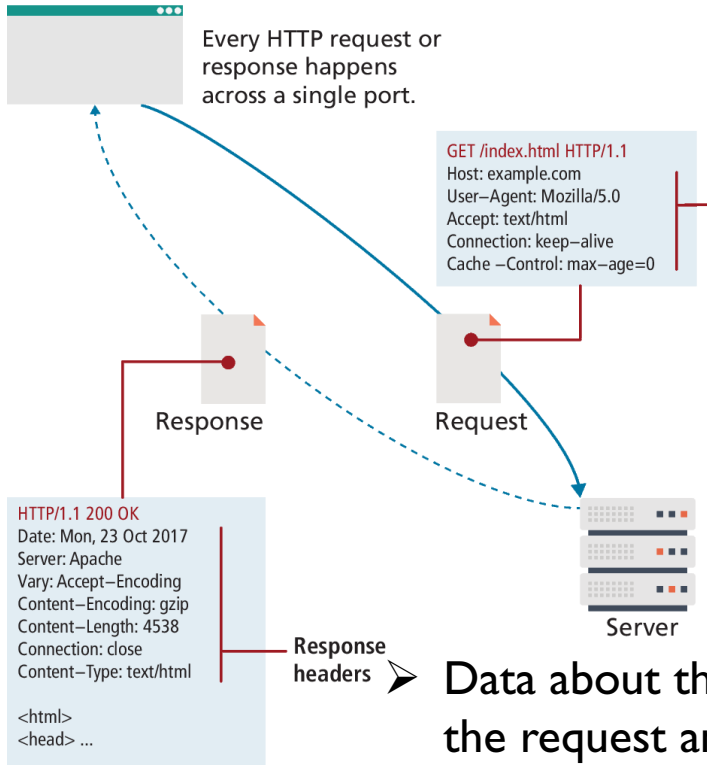
- "www"
  - Adding "www" before a URL is a convention that has been in use for many years.
    - Historically, the "www" subdomain was used to distinguish web servers from other types of servers (such as mail servers or FTP servers) that might also be hosted on the same domain.
  - Today, many websites no longer require the "www" prefix and are designed to work without it.

- Port number in URL
  - A port number is used in a URL to specify the specific network port that should be used to communicate with a particular web server.
  - Adding a port number to a URL is necessary when a web server is configured to listen on a non-standard port or when multiple web servers are running on the same machine and need to be accessed using different ports.
  - To include a port number in a URL, it is added after the domain name or IP address and separated by a colon. For example, if a web server is configured to listen on port 8080, the URL to access it would be:
    - http://example.com:8080
  - Similarly, if a web server is configured to listen on port 8443 for HTTPS traffic, the URL to access it would be:
    - https://ideweb2.hh.se:8443 or https://ideweb2.hh.se:8443/WebInterface/login.html

HALMSTAD UNIVERSITY

Hypertext Transfer Protocol (HTTP)

- Used to transfer files on the web
- Web browsers send HTTP requests for web pages and their associated files.
- Web servers send HTTP responses back to the web browsers.



➢ Data about the client machine.

➢ Data about the server answering the request and the data being sent.

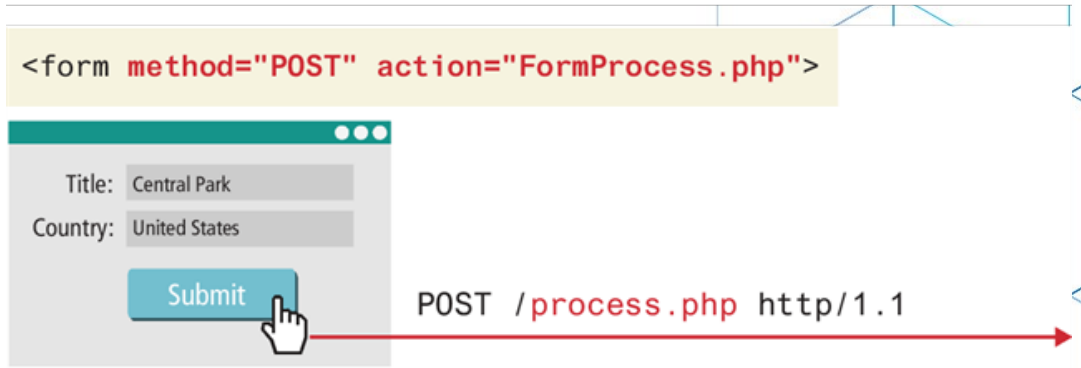
HALMSTAD UNIVERSITY

HTTP Request Methods

- The most common requests are the **GET** and **POST** request, along with the HEAD request
- Other HTTP verbs are PUT and DELETE. Less used HTTP verbs are CONNECT, TRACE, and OPTIONS.

  - Ask for a resource located at a specified URL to be retrieved.
  - Data can be transmitted through a GET request, with a query string.

  - Used to transmit data to the server using an HTML form.



HALMSTAD UNIVERSITY

HTTP flow

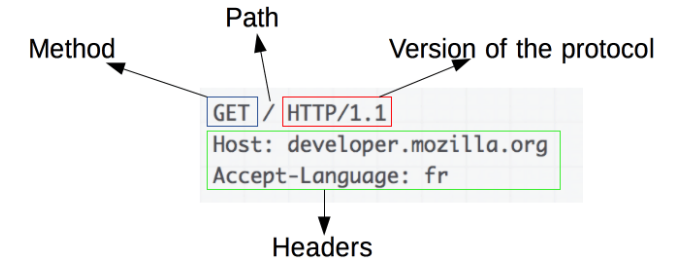
- When a client wants to communicate with a server, either the final server or an intermediate proxy, it performs the following steps:
  1. Open a TCP connection
  2. Send an HTTP message

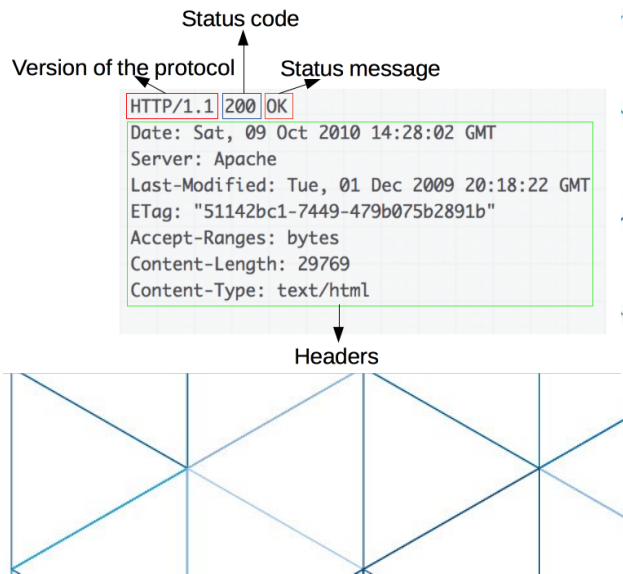
```
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```

  3. Read the response sent by the server, such as:

```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html
<!DOCTYPE html... (here comes the 29769 bytes of the requested web page)
```

  4. Close or reuse the connection for further requests.



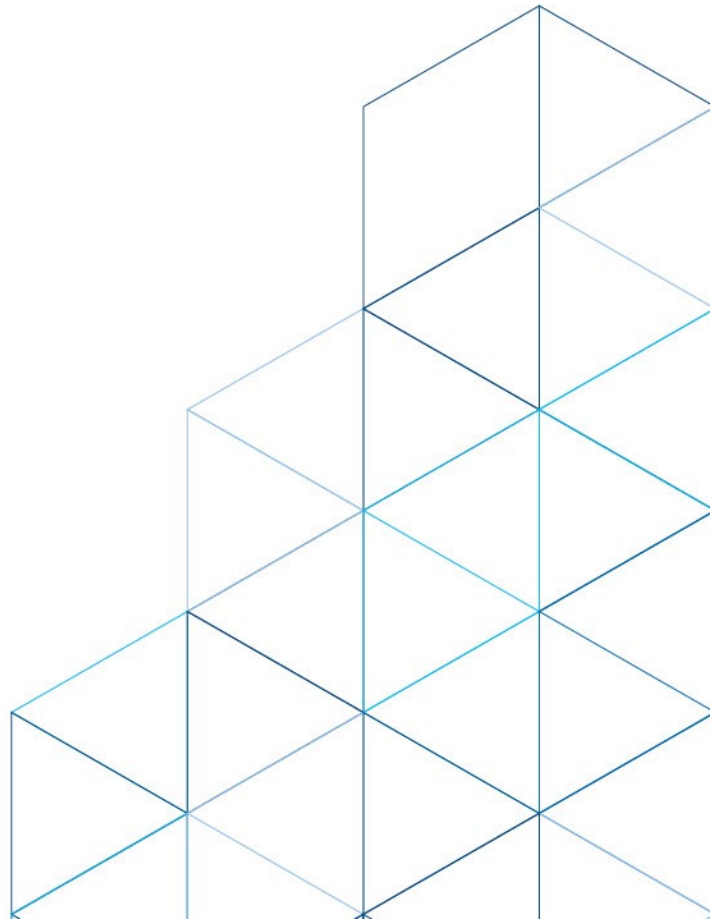


HALMSTAD UNIVERSITY

Response OR status codes

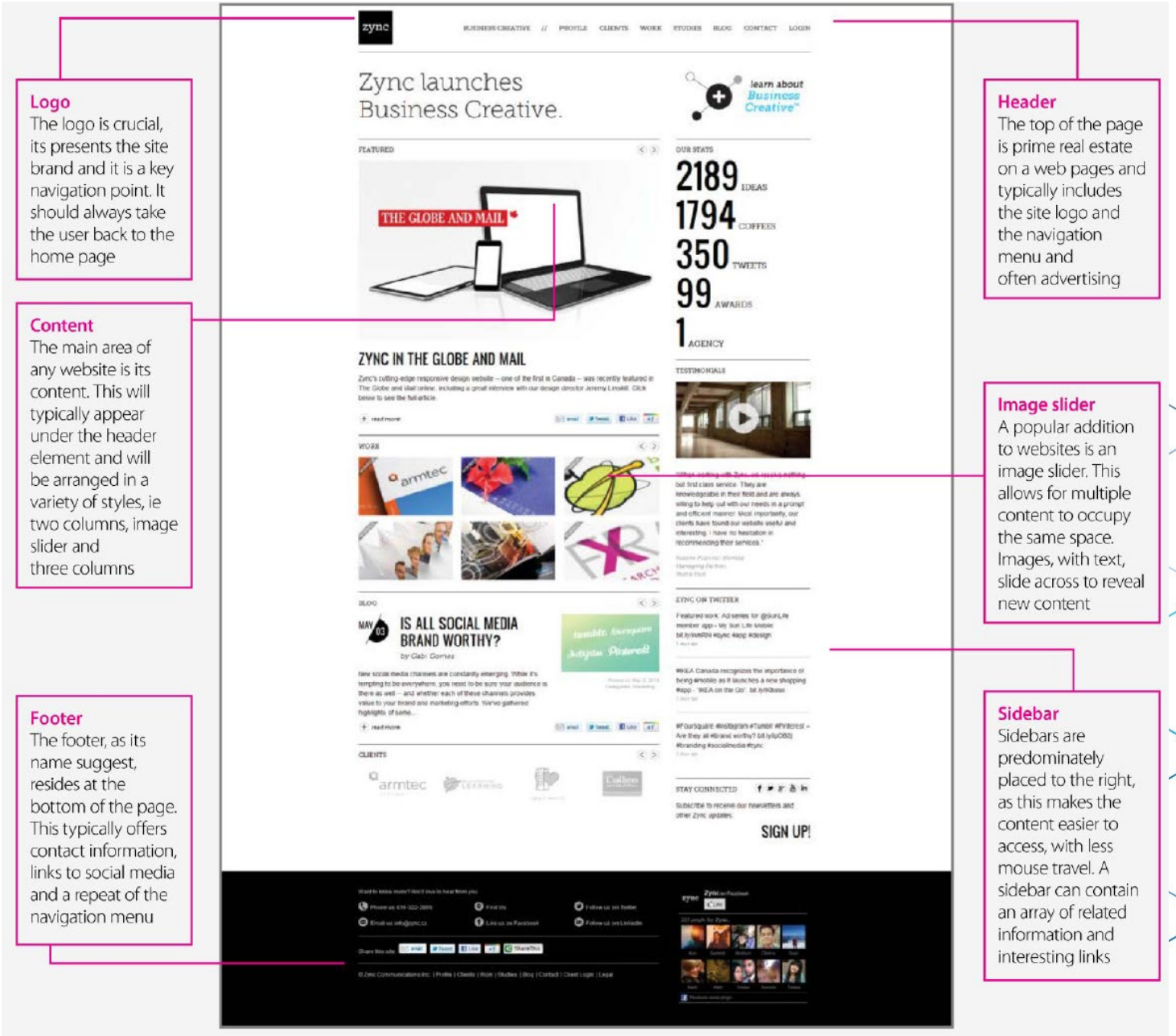
- Servers send HTTP status codes to describe the response to the client.
- HTTP response status codes indicate whether a specific HTTP request has been successfully completed.
- HTTP status codes influence caching, and handling of URIs on the client side.

- 1xx (Informational): Request received, continuing process.
- 2xx (Successful): The action was successfully received, understood, and accepted.
  - 200: OK
- 3xx (Redirection): Further action needs to be taken in order to complete the request.
  - 301: Moved Permanently
  - 304: Not Modified
  - 307: Temporary redirect
- 4xx (Client Error): The request contains bad syntax or cannot be fulfilled.
  - 400: Bad Request
  - 401: Unauthorized
  - 404: Not found
  - 414: Request URI too long
- 5xx (Server Error): The server failed to fulfil an apparently valid request.
  - 500: Internal server error



HALMSTAD UNIVERSITY

Anatomy of a Web Page

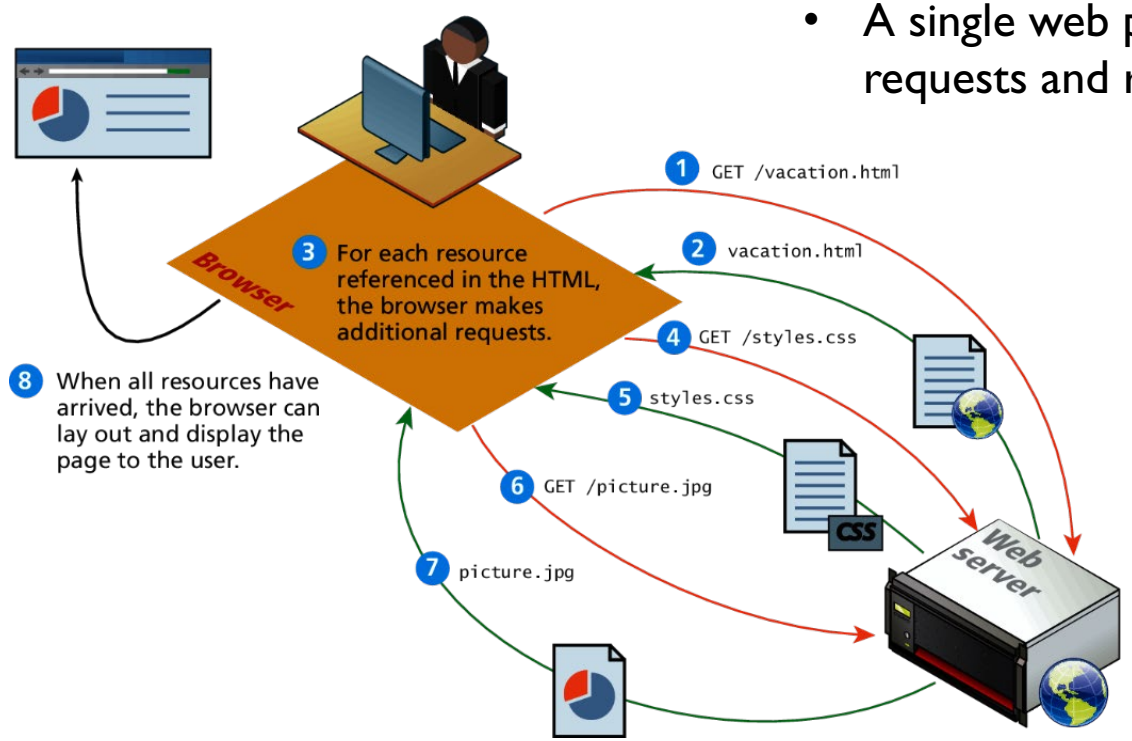


Source: 2015 - Web Design For Beginners 4th Revised Edition

HALMSTAD UNIVERSITY

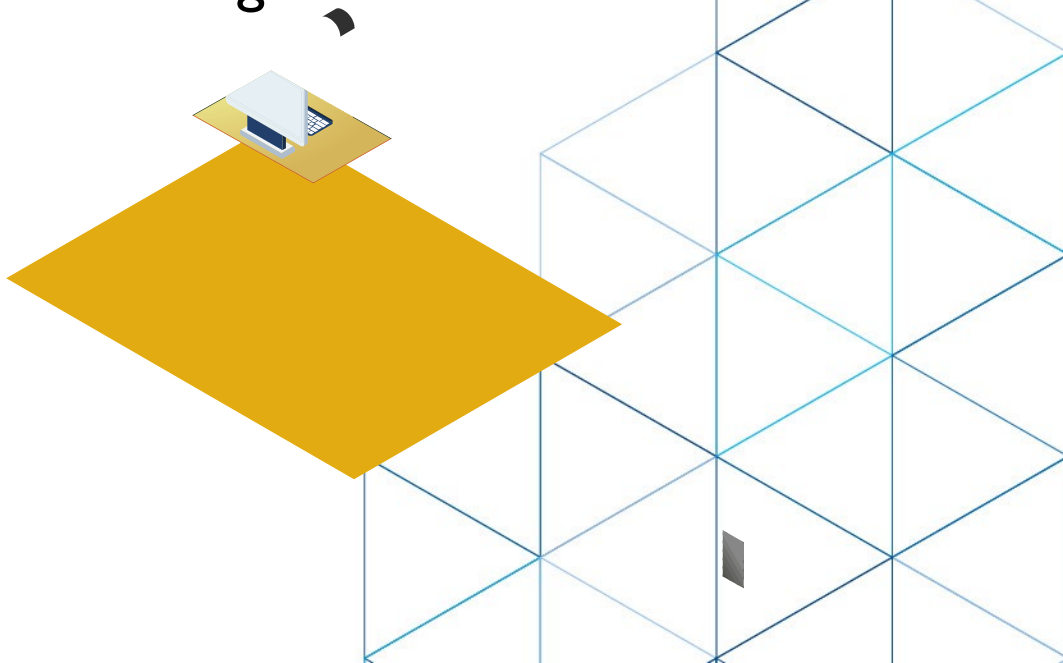
Web Browsers

- Fetching a Web Page



- Seeing a single web page is facilitated by the browser, which
  - requests the initial HTML page, then
  - parses the returned HTML to find all the resources referenced from within it (like images, style sheets, and scripts).
- Only when all the files have been retrieved is the page fully loaded for the user.
- A single web page can reference dozens of files and requires many HTTP requests and responses.

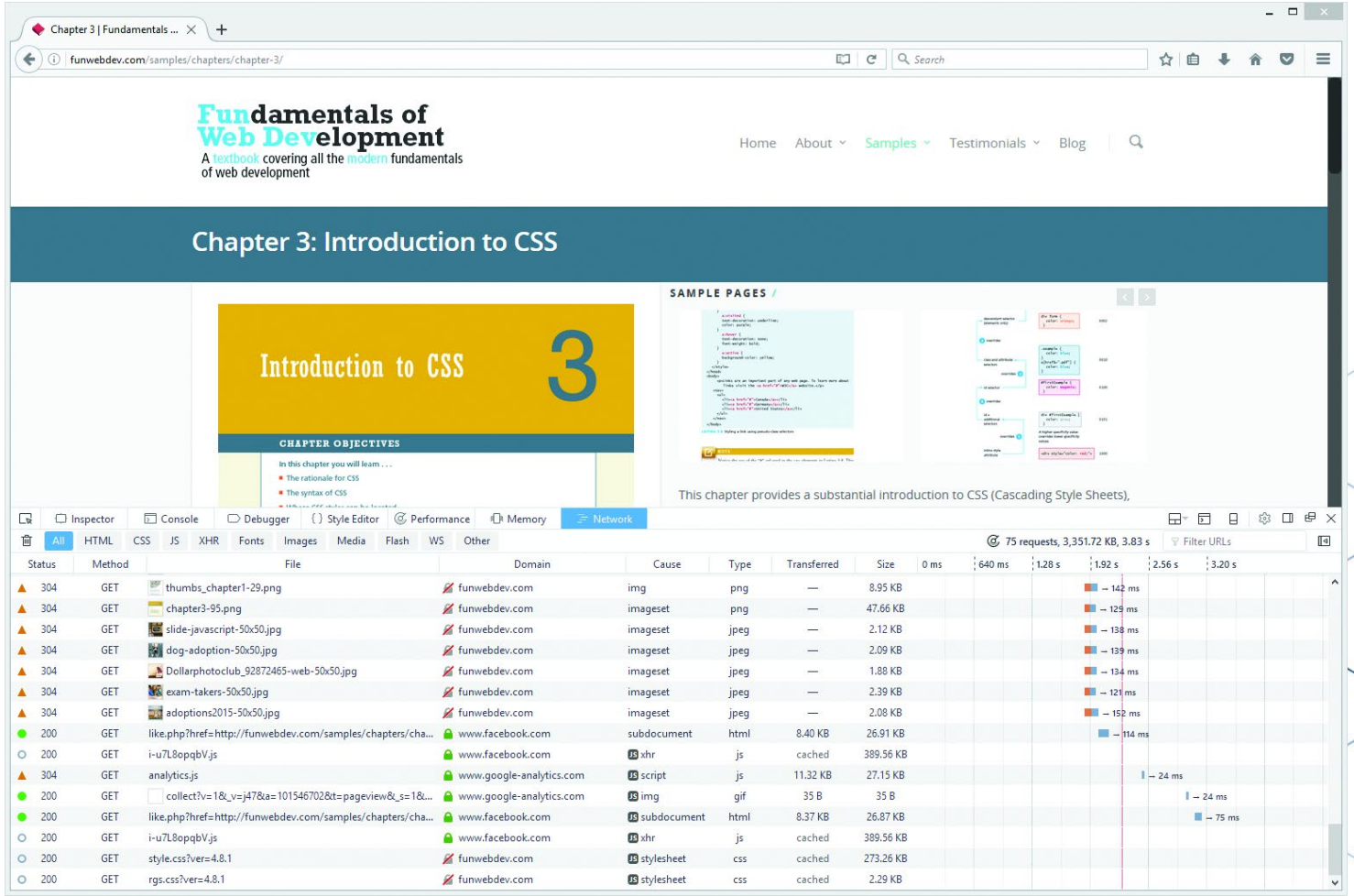
Browser Caching



HALMSTAD UNIVERSITY

Web Browsers

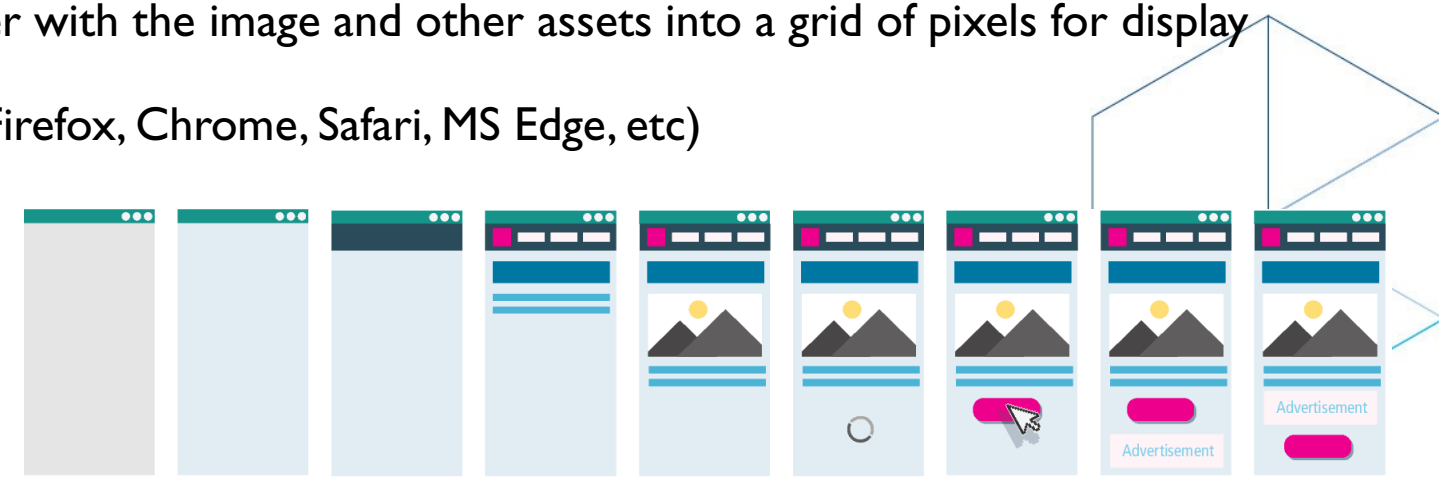
- Fetching a Web Page – Load Times and Cascades



HALMSTAD UNIVERSITY

Web Browsers

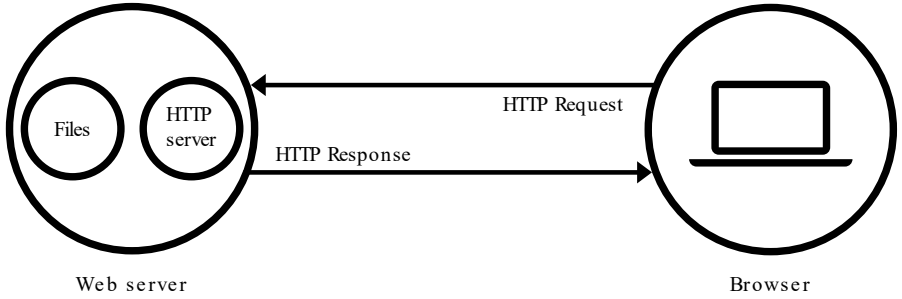
- Browser Rendering
  - The algorithms within browsers to download, parse, layout, fetch assets, and create the final interactive page for the user are commonly referred to collectively as the rendering of the page.
  - Interprets the entire HTML markup together with the image and other assets into a grid of pixels for display within the browser window.
  - Implemented differently for each browser (Firefox, Chrome, Safari, MS Edge, etc)
- Browser Features
  - Search engine integration,
  - URL and Form autocompletion,
  - Cloud caching of user history/bookmarks,
  - Phishing website detection,
  - Secure connection visualization,
- …
- Browser Extensions
  - Tools and features to customize Web Browsers



HALMSTAD UNIVERSITY

Web Servers

- A computer that responds to HTTP requests.



Source: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server

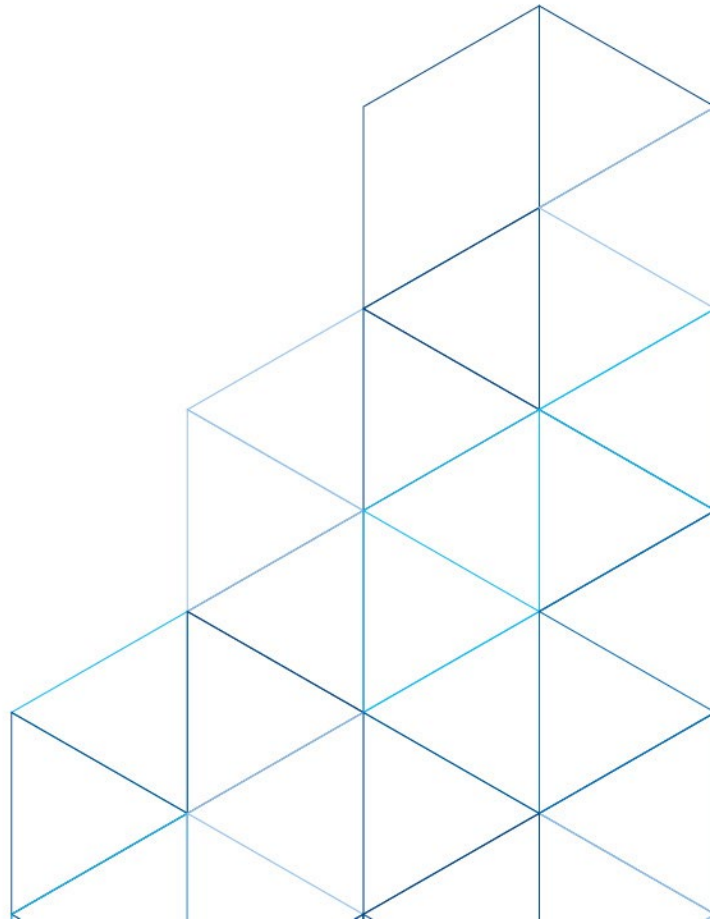
- Solution or application stack:
  - Operating system (Windows, Linux, Mac),
  - Web server software (Apache, IIS)
  - Database management system (PostgreSQL, MySQL, SQL Server)
  - Scripting language for dynamic requests (PHP, ASP.Net, Python)

  - Examples: LAMP, WAMP, WISA, …

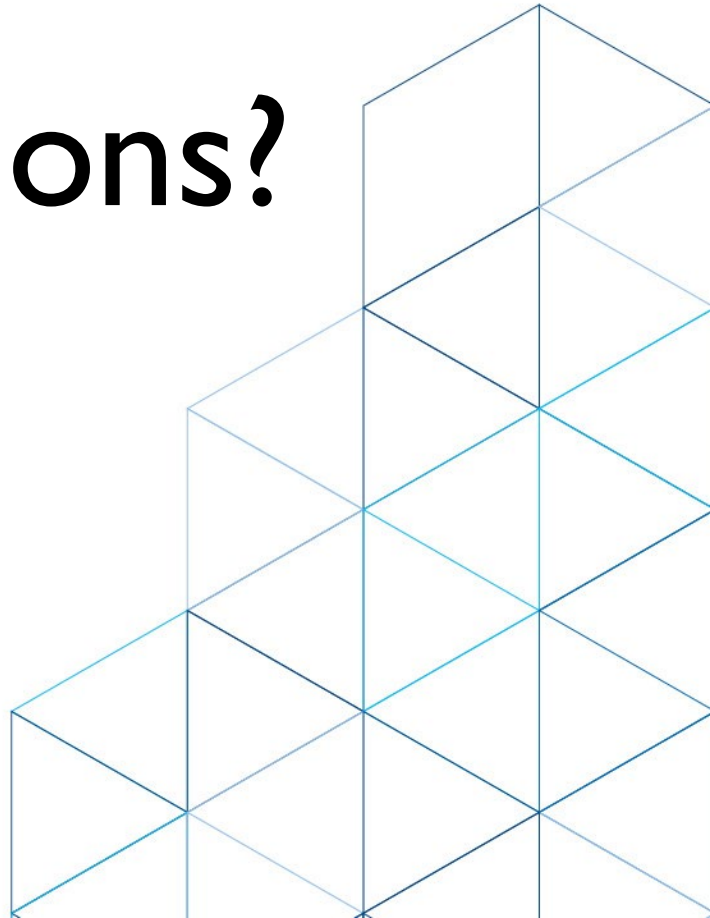
HALMSTAD UNIVERSITY

Text Editors

- It is likely that I will write code in different text editors.
  - If you are not used to coding, you might want to use the following text editors:
    - Notepad ++
    - VS Code
    - Sublime
  - There are online options too:
    - jsfiddle
    - jsbin



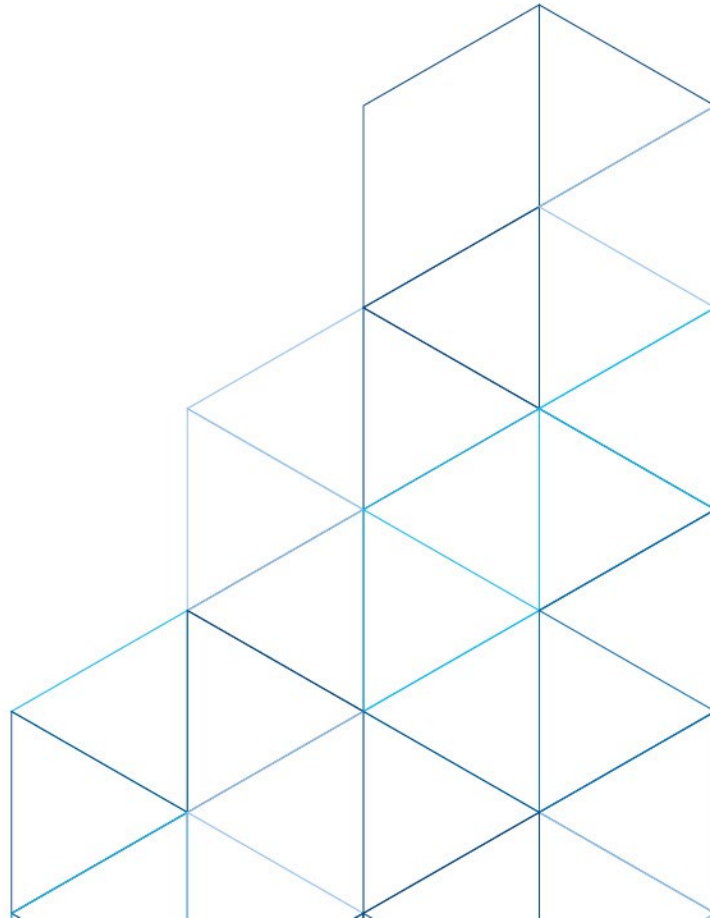
Comments or Questions?



HALMSTAD UNIVERSITY

Next

- HTML
  - Introduce HTML
    - HTML, XHTML, and HTML5
  - Introduce the syntax of HTML
  - Discuss semantic markup
  - Understand the structure of a web page
    - Head and body
  - Introduce the main elements in HTML
  - Introduce HTML tables and forms
  - Introduce Web media
    - Color models, image formats, HTML5 canvas
  - How to create and deploy a web page



HALMSTAD UNIVERSITY