

CMPE 220

Week 11

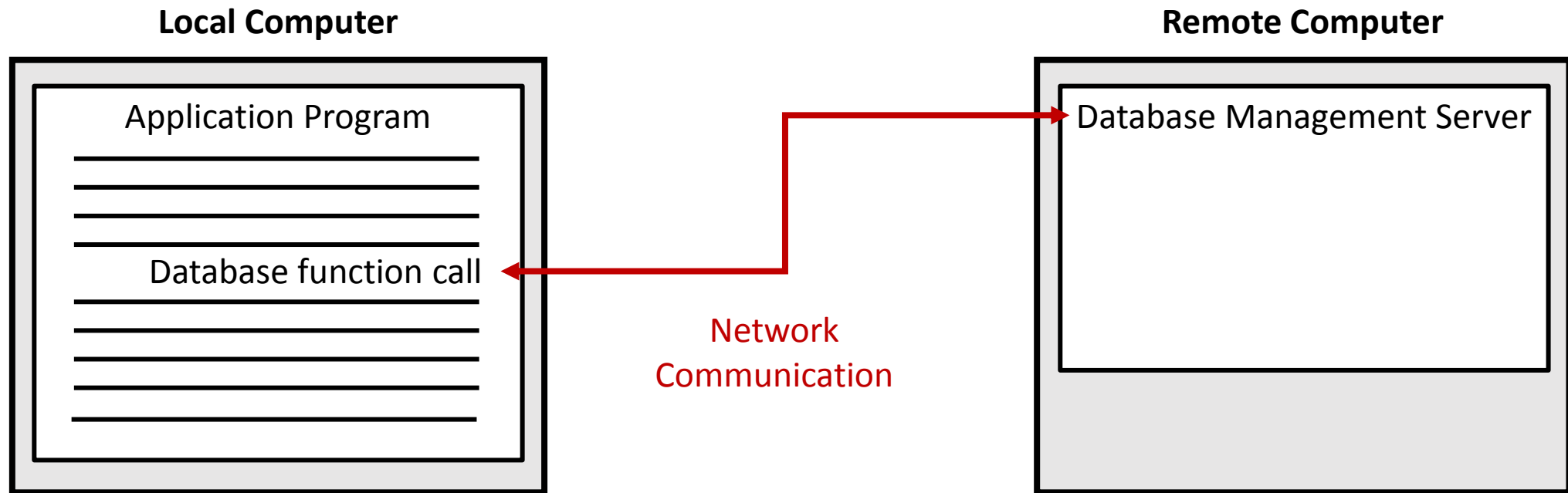
Servers (and client/server applications)

The Client / Server Model

- A *server* is a program that runs on a computer, providing a specific service to other program(s), called *clients*
- In the POSIX world, a server is often called a *daemon* (demon)
 - a long-running background process that answers requests for services
- *The server program may launch multiple processes, as needed*
- The software that makes up a server may - or may not - have system dependencies
- Examples:
 - Database Management System
 - FTP Server
 - Mail Server
 - Web Server
 - Windowing System

Local or Remote

- The *client* and the *server* can run on the same machine, or they may run on separate machines and communicate over a *network*.
- Network communication may be hidden by a function call library.



Accessing Network Services

- Client / Server applications require a *protocol* – a language and a set of rules to allow the client to communicate with the server
 - Languages do not need to be human readable (i.e. could be binary)
- Early protocols were ad hoc – made up by the application developer
 - FTP: File Transfer Protocol
 - Developed in the early 1970s by Abhay Bhushan, a student at MIT
- Protocols Encapsulate:
 - Authentication
 - Requests
 - Responses
 - State & Session Management

Standardizing Protocols: OSI

- **Open Systems Interconnection model (OSI model):** a conceptual model that characterizes and standardizes the communication functions
- Adopted as an ISO standard in 1980
- Seven Layers

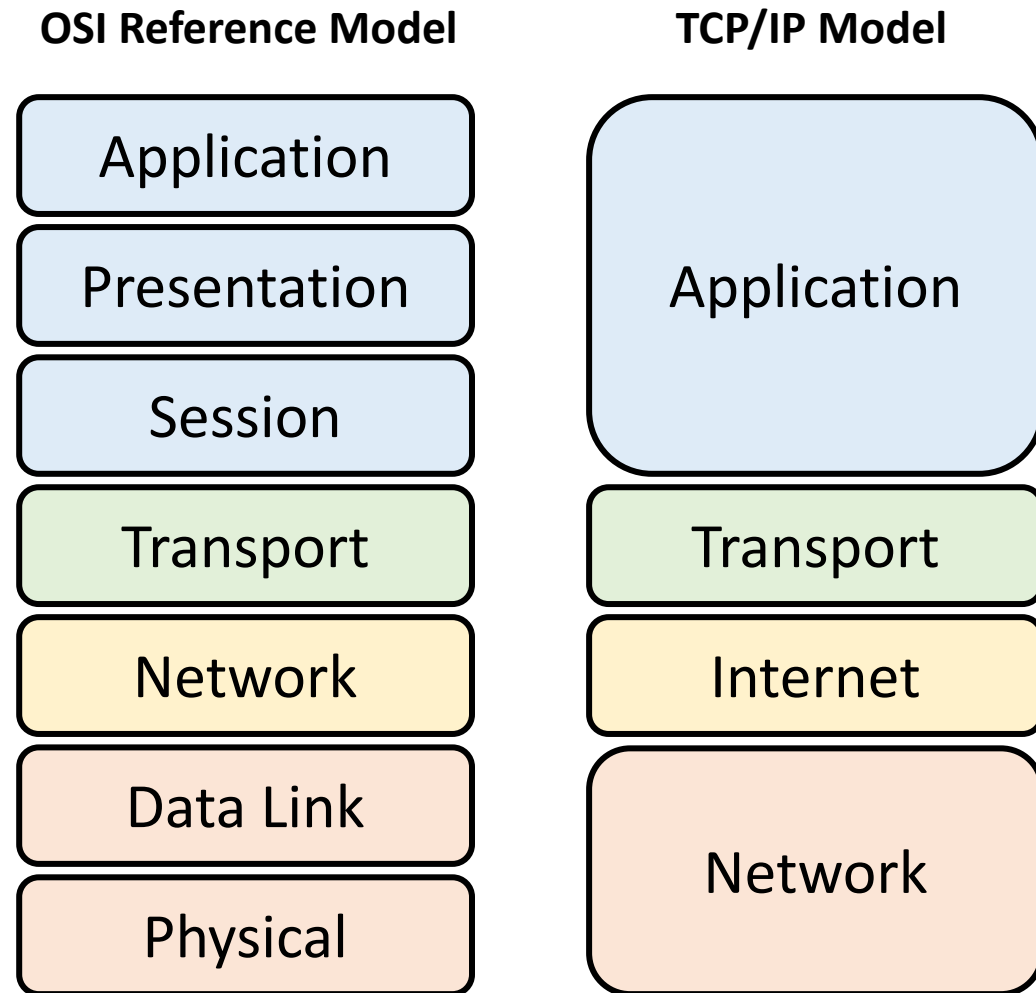
Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

TCP/IP

- Transmission Control Protocol / Internet Protocol
- Based on research done by the Army Research Projects Agency (ARPA) in the 1960s and 1970s
- Standardized by the US Department of Defense in 1982
- Administered by the Internet Engineering Task Force (IETF) since 1989
- Four Layers:

Application Layer
Transport Layer
Internet Layer
Network Interface Layer

OSI Versus TCP/IP Models



- OSI model provides a clear distinction between application, presentation, and session services.
 - TCP/IP groups these as a single *Application* layer
-
- In the OSI model, the data link layer and physical are separate layers.
 - TCP/IP groups these as a single *Network* layer

Application-Specific Protocols

- Each type of server (web, database, ftp, etc) has its own protocol specific to that application.
 - A web server uses the HyperText Transfer Protocol to request web pages
- Service-specific protocols are the top layer of the TCP/IP network model

Application Layer	
Transport Layer	Transmission Control Protocol (TCP)
Internet Layer	Internet Protocol (IP)
Network Interface Layer	

TCP/IP Enabled the Internet

- Ubiquitous: supported by virtually every operating system
- Essentially makes every system today a *Network Operating System*
- Every device on the Internet has a unique 32-bit *IP address*, consisting of four 8-bit numbers (4.2 billion addresses):
 - 67.169.41.253
- IPv6: an extended protocol which supports 128-bit addresses, made up of eight 16-bit numbers, expressed in hexadecimal
 - FE80:CD00:0000:0CDE:1257:0000:211E:729C
 - Drafted by the Internet Engineering Task Force (IETF) in 1998
 - Finalized in 2017 – currently being deployed worldwide

TCP/IP Service Requests

- In addition to IP addresses, the TCP/IP protocol uses the notion of a standard “port” to map requests to specific services
- For the **Transmission Control Protocol** and the **User Datagram Protocol**, a port number is a 16-bit integer in the header of a message
- Reserved (“well-known”) port numbers – *by convention*:
 - FTP: port 20/21
 - Telnet: port 23
 - DNS: port 53
 - HTTP: port 80
 - HTTPS: port 443
 - POP3 (email): port 110
 - MySQL: port 3306
- Servers use system calls to open read operations on network ports to receive incoming client requests

What Does a (modern) Operating System Do?

1. Process Management
 - Interprocess Communications
2. Input / Output (I/O) Management
3. Memory Management
4. File System Management
5. System Functions and Kernel Mode
6. User Interaction – (maybe)
7. Network Services

File Transfer Protocol (FTP) - Abhay Bhushan

- One of the earliest client/server applications (early 1970s by Abhay Bhushan)
- A graduate of the first class (1960–65) from the Indian Institute of Technology Kanpur
- Masters in EE from MIT
- Drafted RFC 114 – FTP
- Contributed to the development of the ARPAnet and email protocols

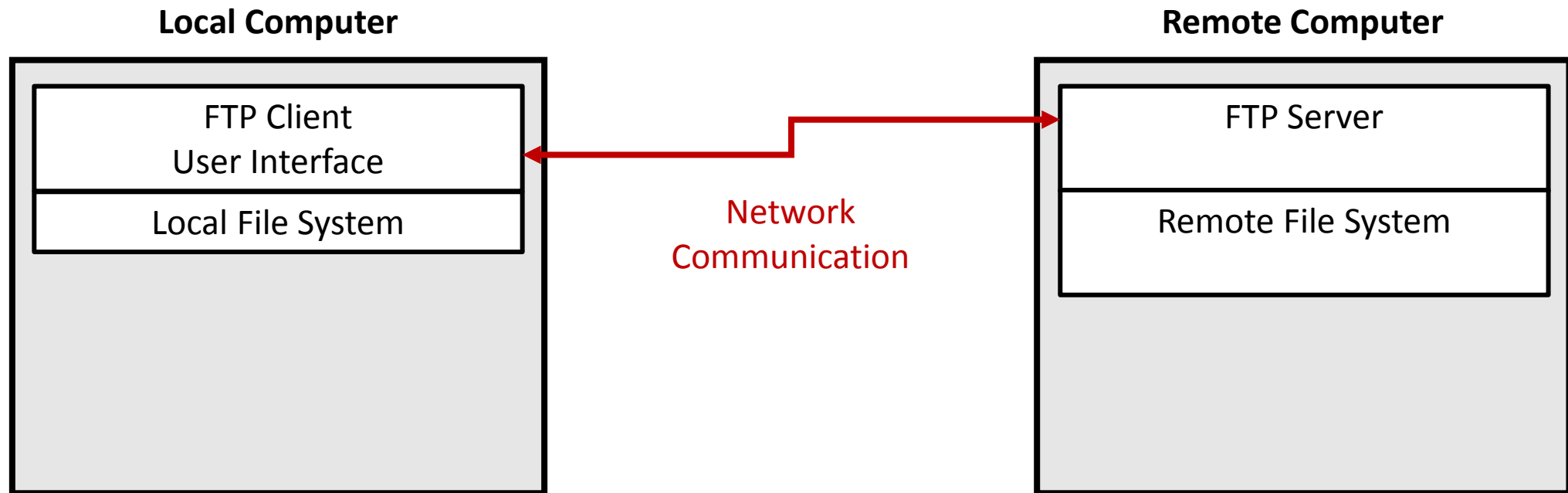


File Transfer Protocol (FTP)

- Originally operated over dialup phone lines
- An FTP client connects to an FTP server
- Via the client, a user can:
 - Authenticate (login)
 - List files
 - Request (get) files or upload (put) files
 - Change directories
 - Rename files
 - Delete files
- FTP Clients
 - FileZilla
 - Cyberduck
 - Transmit
 - WinSCP
 - *many others*

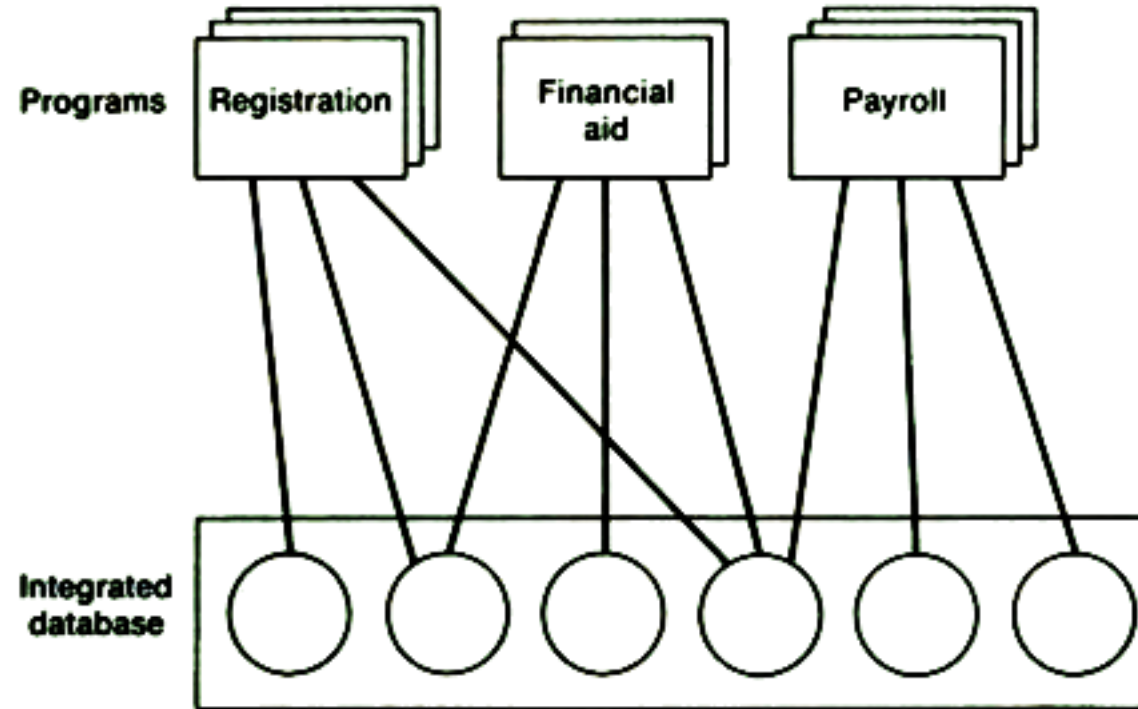
FTP

- The FTP client program is started by the user only when needed
- The FTP server program is started by the system and continues to run



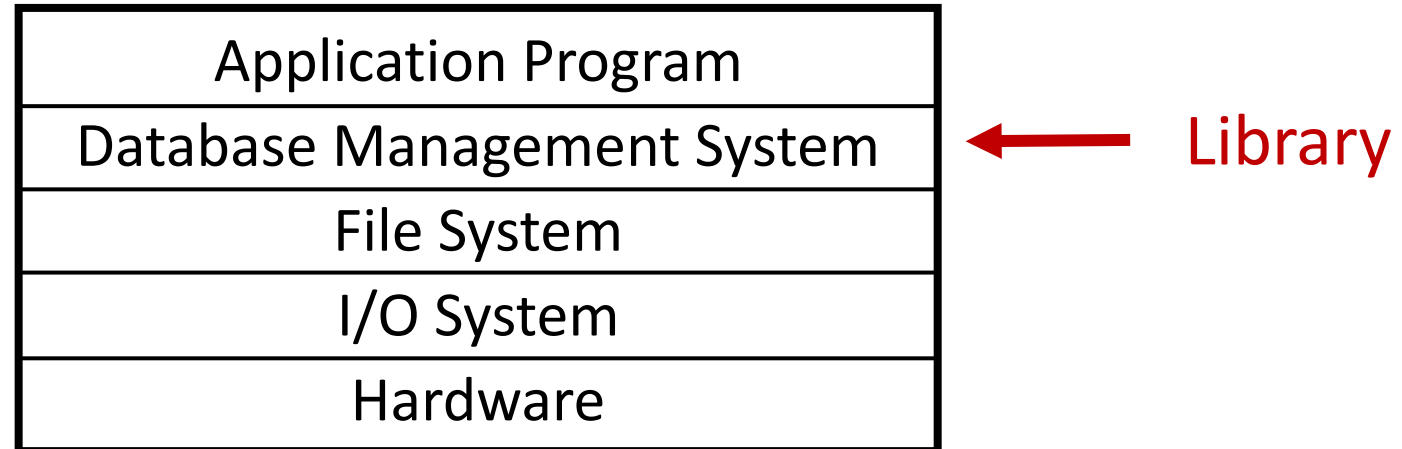
Database Management Systems

- A Database Management System (DBMS) allows information to be organized in a structured way, and used by one or more application programs.



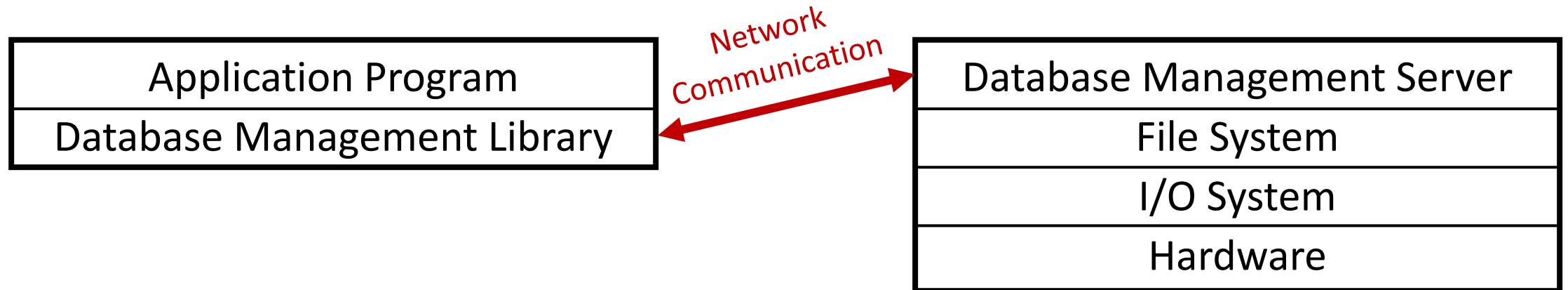
A Single Computer DBMS

- The DBMS is simply a library



A Networked DBMS

- The DBMS consists of a library, and a *server*



- The application and the database server may actually run on the same computer, but still use a network protocol to communicate (localhost)

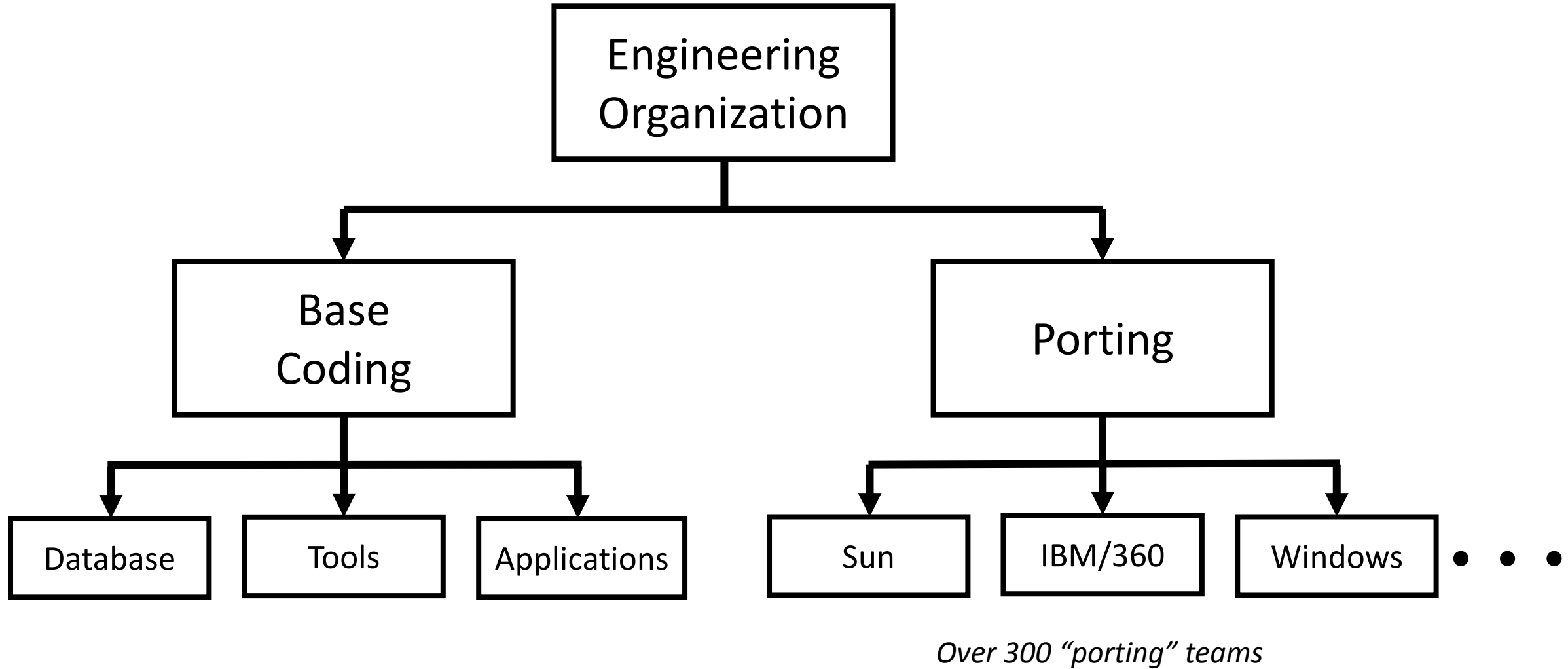
Database Clients

- Any application can be a client of a DBMS server
- Database systems usually include a front-end interface that allows programmers to examine and update database contents
 - This “front end” is simply another client

Is a DBMS System Software?

- NO:
 - Sits on top of the file system
 - Uses standard communication capabilities
- YES:
 - The DBMS is optimized by taking advantage of specific knowledge of the I/O system and the hardware architecture
 - Database vendors work closely with system vendors

How Oracle Builds Software



According to Larry Ellison, founder of Oracle

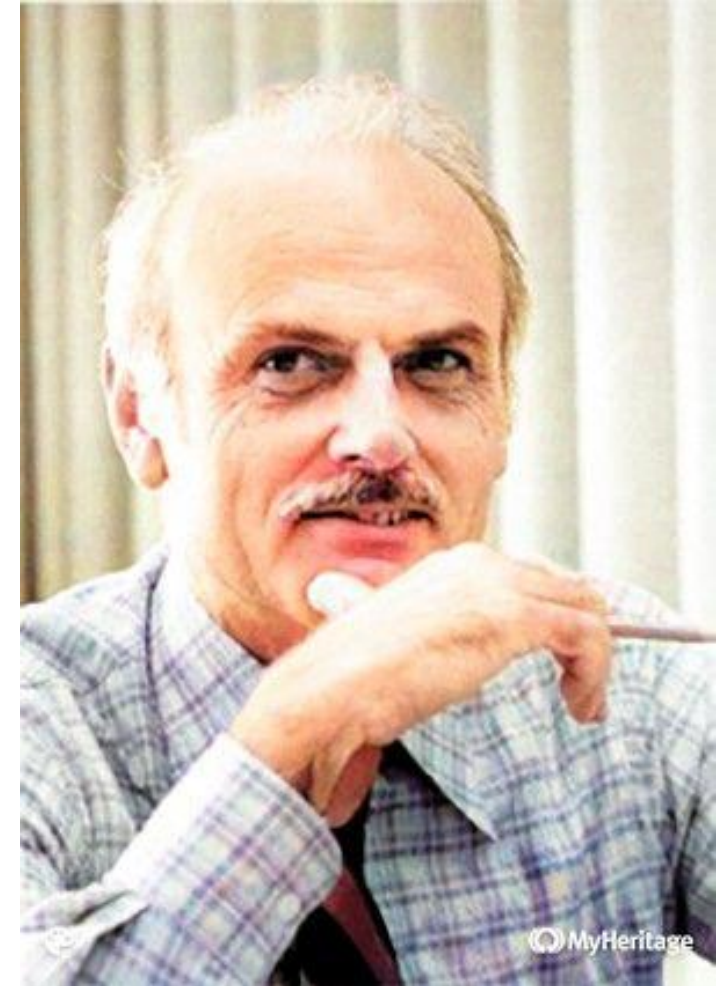
- “Oracle is not a database company. We are a portable software solutions company.”
- Just as we saw with high level programming languages, database management systems allow *portable* applications, by providing a standard interface – a database language – which is tuned to run on a wide range of systems.

Relational Databases

- The most widely used “type” of database
 - Oracle
 - MySQL
 - Accessed using Structured Query Language (SQL)
 - Rigidly defines the data structure
-
- NoSQL or non-relational databases
 - Allow storage, indexing, and retrieval of unstructured or loosely structured data
 - A number of types for different purposes

Relational Databases - History

- The term "relational database" was invented by Dr. Edgar (Ted) Codd at IBM in 1970.
- Codd introduced the term in his research paper "A Relational Model of Data for Large Shared Data Banks". In this paper and later papers, he defined what he meant by "relational".



Relational DB: Entities and Attributes

- Entity

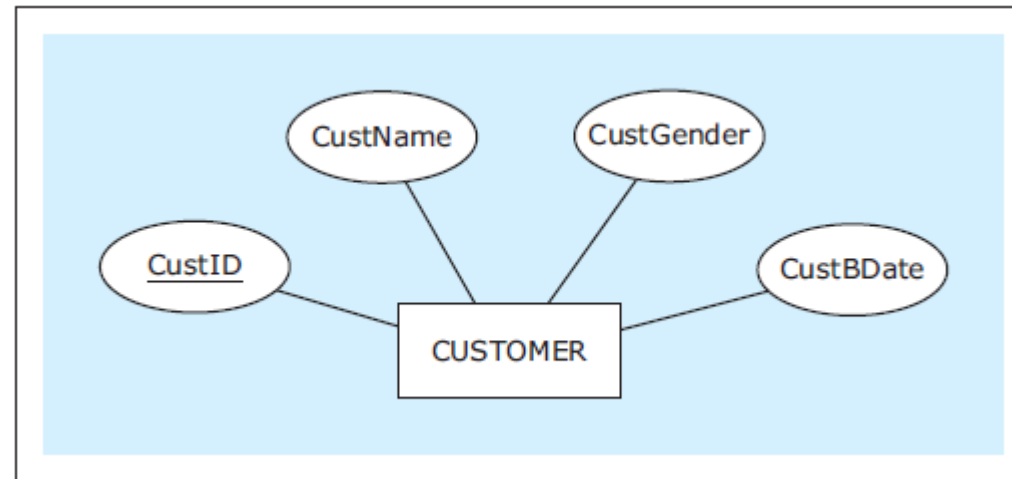
- Typically represents a real-world concept.
 - Examples: customer, product, store, event, etc.
- Data that the database stores.

- Attribute

- Characteristic of an entity that the database stores.
 - Examples (for a customer): name, address, id, etc.
- A *unique attribute* of an entity has a value that is different for each entity instance (such as a customer number)

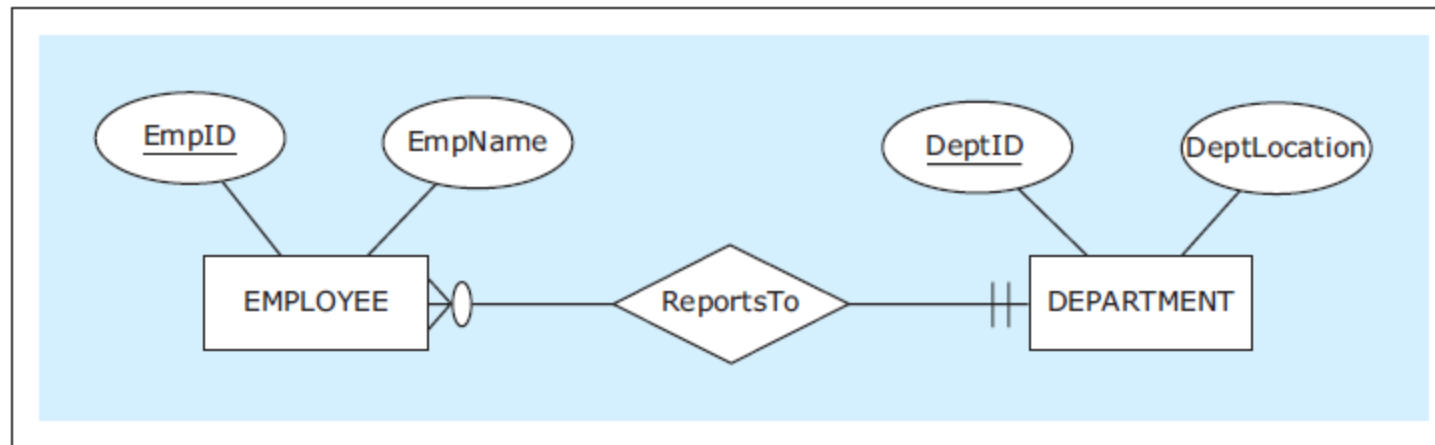
Diagramming Entities and Attributes

- An *entity relationship diagram*, or *ER diagram*, shows an entity with a rectangle and its attributes with ovals.
 - Underline the unique attribute.



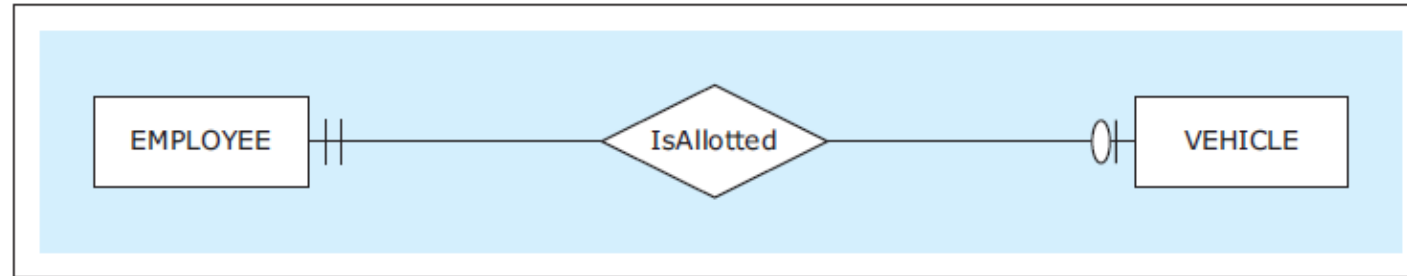
Relationships

- Each entity in an ER diagram **must be related** to at least one other entity.
- Show a relationship with a diamond and connect the diamond to the entities that are part of the relationship.



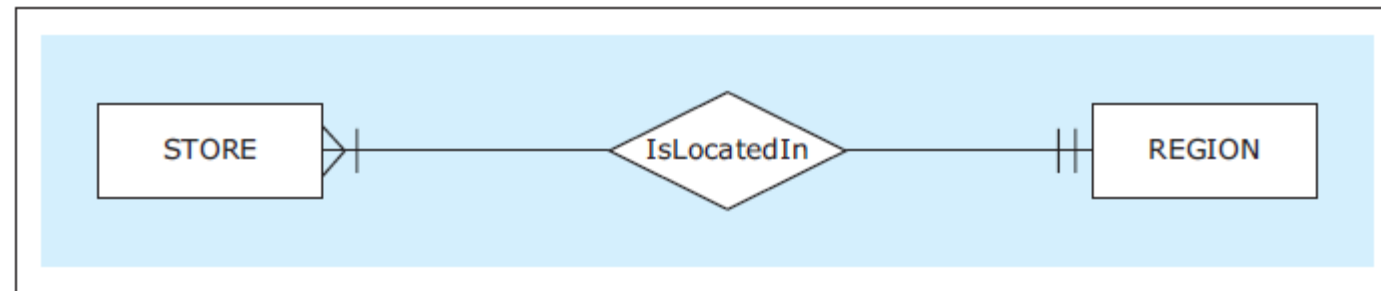
Types of Relationships

- One-to-one (1:1)



Each employee is allotted at most one (zero or one) vehicle.
Each vehicle is allotted to exactly one employee.

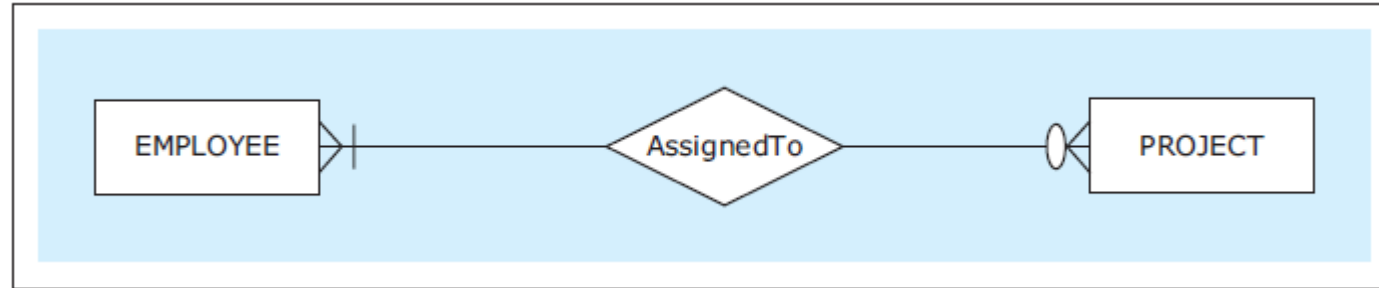
- One-to-many (1:M)



Each region has located in it at least one (i.e., many) stores.

Types of Relationships - continued

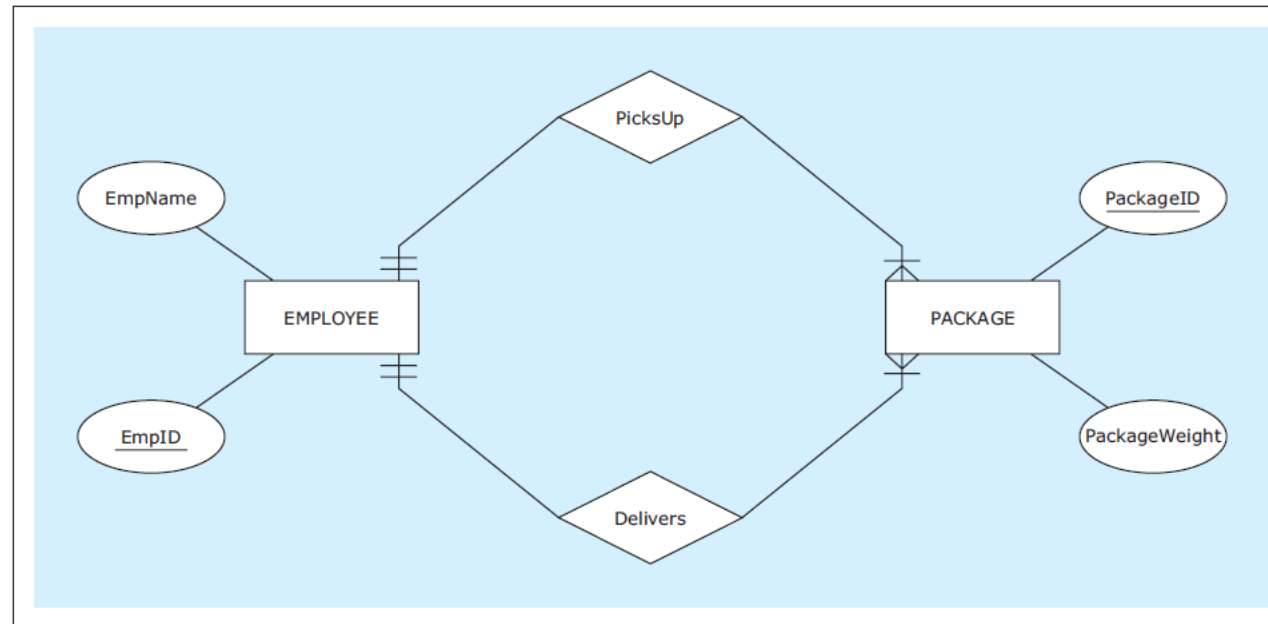
- Many-to-many (M:N)



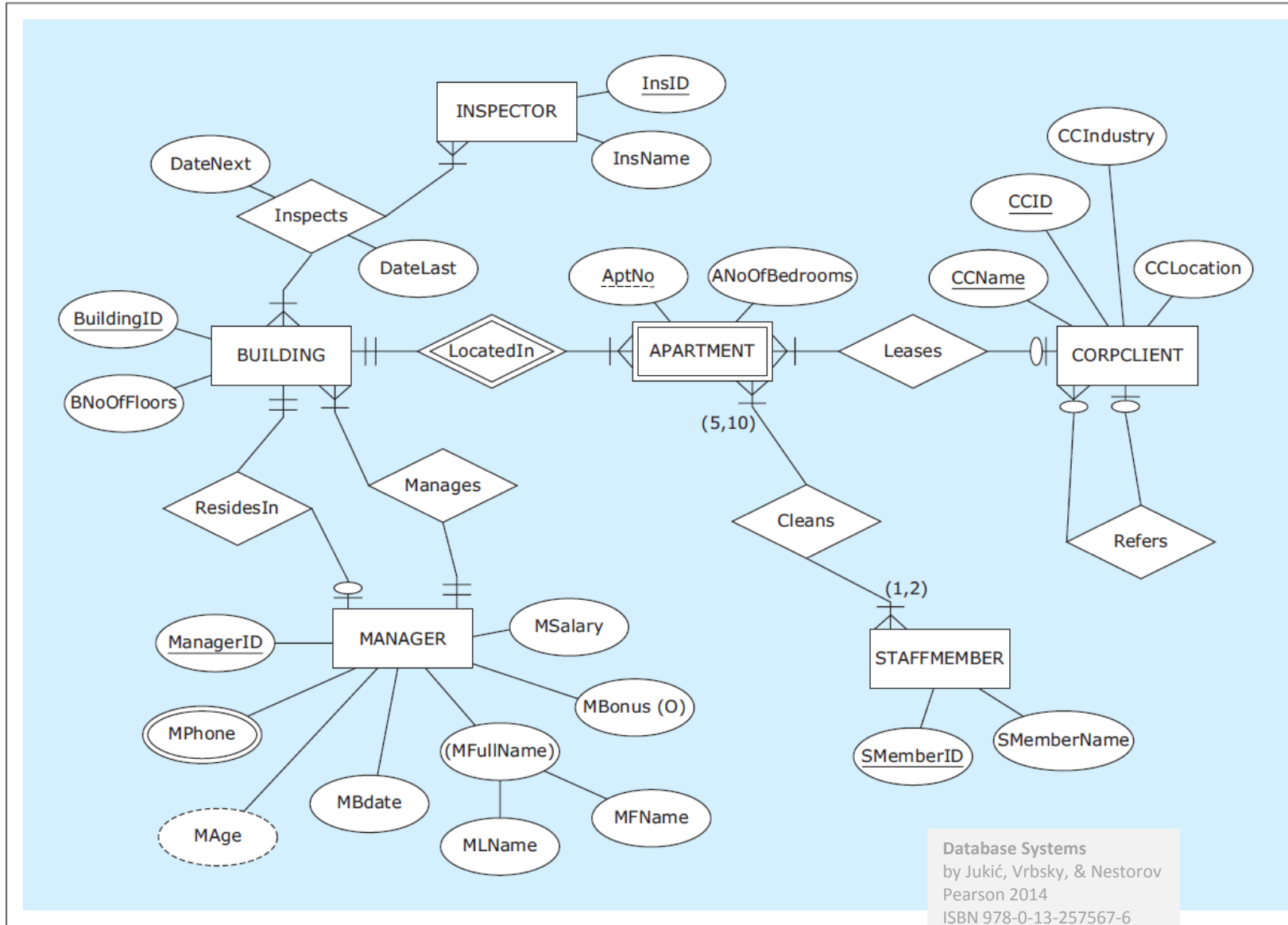
Each employee is assigned to zero or several (i.e., many) projects.
Each project has assigned to it at least one (i.e., one or many) employee.

Multiple Relationships

- Two entities can have multiple relationships with each other.



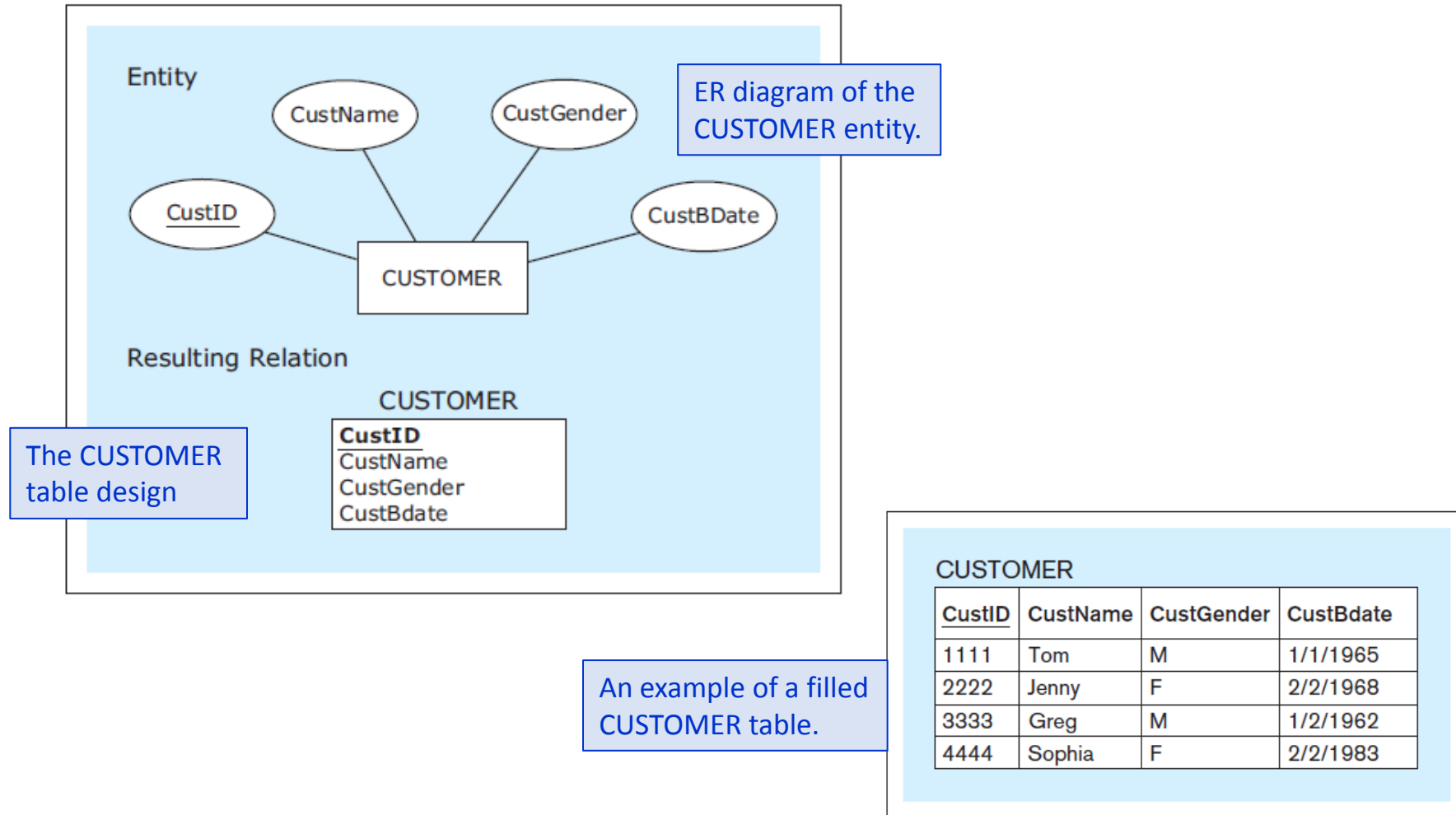
ER Diagram Example



Database Design

- The ER diagram is the design of our database.
- Map the diagram to a set of database tables.
 - Each table contains related data.
 - Example: An employee table.
 - A table is also called a *relation*.
- Each row of a table contains the data of a single record.
 - Example: One row per employee record.
- The table columns are the attributes.
 - Examples: Employee ID, name, gender, phone number, birthdate, etc.

Mapping Entities



Primary Key

- Each table must have a *primary key*.
 - A column or set of columns whose value uniquely identifies each row.
 - Underline the primary key in each database table.

EMPLOYEE				
<u>EmpID</u>	EmpName	EmpGender	EmpPhone	EmpBdate
0001	Joe	M	x234	1/11/1985
0002	Sue	F	x345	2/7/1983
0003	Amy	F	x456	8/4/1990
0004	Pat	F	x567	3/8/1971
0005	Mike	M	x678	5/5/1965
0010	Mike	M	x666	8/1/1974
0007	Barbara	F	x777	4/5/1980
0011	Ivan	M	x777	3/4/1981
0009	Amy	F	x777	1/11/1985

Database Systems
by Jukić, Vrbsky, & Nestorov
Pearson 2014
ISBN 978-0-13-257567-6

Structured Query Language (SQL) Statements

- SQL (“Sequel”) is a language for accessing and manipulating relational databases
- Statements can be entered on the command line of a database front-end (client)
- Statements can be sent by application programs to the database server

SQL – History

- SQL was initially developed at IBM by Donald D. Chamberlin and Raymond F. Boyce after learning about the relational model from Edgar F. Codd in the early 1970s.
- In June 1979, Relational Software (now Oracle) introduced the first commercially available implementation of SQL for VAX computers.
- Approved as an ISO standard in 1987.

SQL Query Examples – single table

- What is the class code of the Java programming class?

Class

Code	Teacher_id	Subject	Room
908	7008	Data structures	114
926	7003	Java programming	101
931	7051	Compilers	222
951	7012	Software engineering	210
974	7012	Operating systems	109

Desired fields

Source tables

Selection criteria

```
SELECT code
FROM class
WHERE subject = 'Java programming'
```

```
+-----+
| code |
+-----+
|  926 |
+-----+
```

SQL Query Examples – multiple tables (joins)

- Who is teaching Java programming?

Teacher

Id	Last	First
7003	Rogers	Tom
7008	Thompson	Art
7012	Lane	John
7051	Flynn	Mabel

Class

Code	Teacher_id	Subject	Room
908	7008	Data structures	114
926	7003	Java programming	101
931	7051	Compilers	222
951	7012	Software engineering	210
974	7012	Operating systems	109

```
SELECT first, last
FROM teacher, class
WHERE id = teacher_id
AND subject = 'Java programming'
```

```
+-----+-----+
| first | last  |
+-----+-----+
| Tom   | Rogers |
+-----+-----+
```

Selecting from multiple tables
is called a *join*.

SQL Query Examples - multiple tables

- What subjects does John Lane teach?

Teacher

Id	Last	First
7003	Rogers	Tom
7008	Thompson	Art
7012	Lane	John
7051	Flynn	Mabel

Class

Code	Teacher_id	Subject	Room
908	7008	Data structures	114
926	7003	Java programming	101
931	7051	Compilers	222
951	7012	Software engineering	210
974	7012	Operating systems	109

```
SELECT code, subject
FROM teacher, class
WHERE last = 'Lane' AND first = 'John'
AND id = teacher_id
```

```
+-----+-----+
| code | subject          |
+-----+-----+
|  951 | Software engineering |
|  974 | Operating systems   |
+-----+-----+
```

SQL Query Examples - multiple tables

- Who is taking Java programming?

```
SELECT id, last, first
FROM student, class, takes
WHERE subject = 'Java programming'
AND code = class_code AND id = student_id
```

```
+-----+-----+-----+
| id    | last  | first |
+-----+-----+-----+
| 1001  | Doe   | John  |
| 1021  | Smith | Kim   |
+-----+-----+-----+
```

Class

Code	Teacher_id	Subject	Room
908	7008	Data structures	114
926	7003	Java programming	101
931	7051	Compilers	222
951	7012	Software engineering	210
974	7012	Operating systems	109

Takes

Student_id	Class_code
1001	926
1001	951
1001	908
1005	974
1005	908
1014	931
1021	926
1021	974
1021	931

Student

Id	Last	First
1001	Doe	John
1005	Novak	Tim
1009	Klein	Leslie
1014	Jane	Mary
1021	Smith	Kim

SQL Update Statement - example

- Fixing a typo (last row of table)

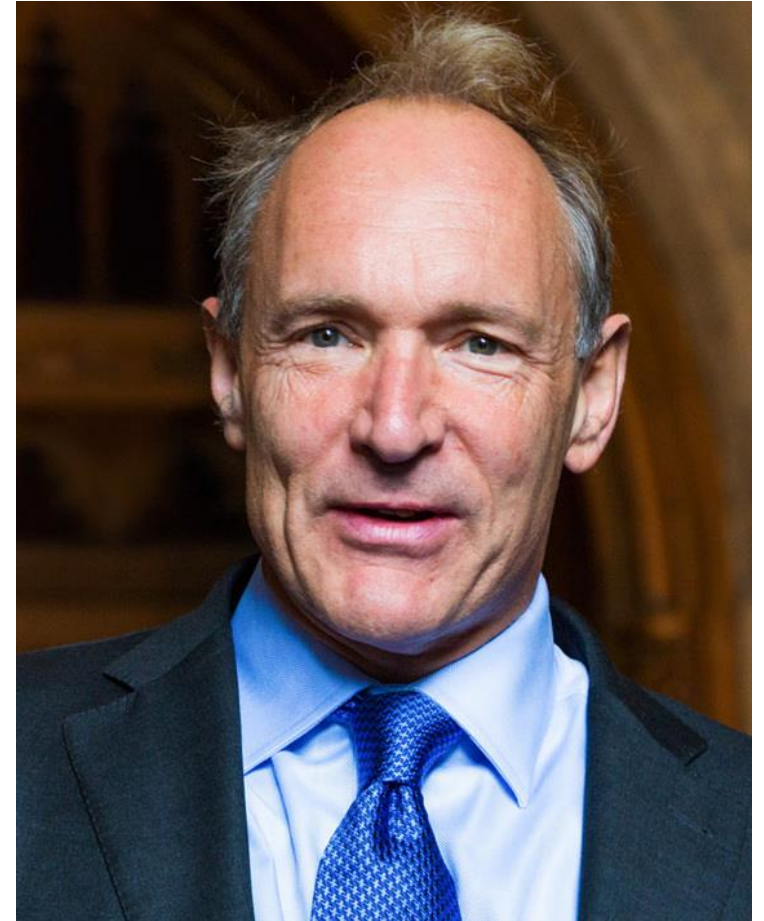
Class

Code	Teacher_id	Subject	Room
908	7008	Data structures	114
926	7003	Java programming	101
931	7051	Compilers	222
951	7012	Software engineering	210
974	7012	Operating sytsems	109

```
UPDATE class
SET subject = 'Operating systems'
WHERE code = '974'
```


Web Servers

- The first web browser and server were created by Sir Tim Berners-Lee in 1990
- Berners-Lee was knighted by Queen Elizabeth II in 2004
- **Browser:** a web client
 - Google Chrome
 - Firefox
 - Safari
 - Internet Explorer
 - Edge
- **Web Servers**
 - Apache
 - Microsoft Internet Information Service (IIS)



Basic Web Protocols

Web Browsers use **two** basic services

- Domain Name Server (DNS Server)
 - Looks up a domain name to get an IP address
 - User Datagram Protocol (UDP) – a low latency protocol - on port 53
- Web Server
 - Responds to web page requests
 - HyperText Transfer Protocol (HTTP) – over TCP - on port 80
 - HyperText Transfer Protocol - Secure (HTTPS) – over TCP - on port 443

Layered Protocols (TCP/IP Model)

Network Model Layers	Domain Name Lookup	Web Page Fetch
Application Layer	DNS request	HTTP or HTTPS
Transport Layer	UDP – User Datagram Protocol	TCP - Transmission Control Protocol
Internet Layer	IP – Internet Protocol	IP – Internet Protocol
Network Layer	Ethernet, etc	Ethernet, etc

- UDP is a connectionless, stateless protocol designed for low latency
- TCP is a connection based, stateful protocol designed for reliability
 - Requires each transmission to be acknowledge by the peer

HTTPS (Secure HyperText Transfer Protocol)

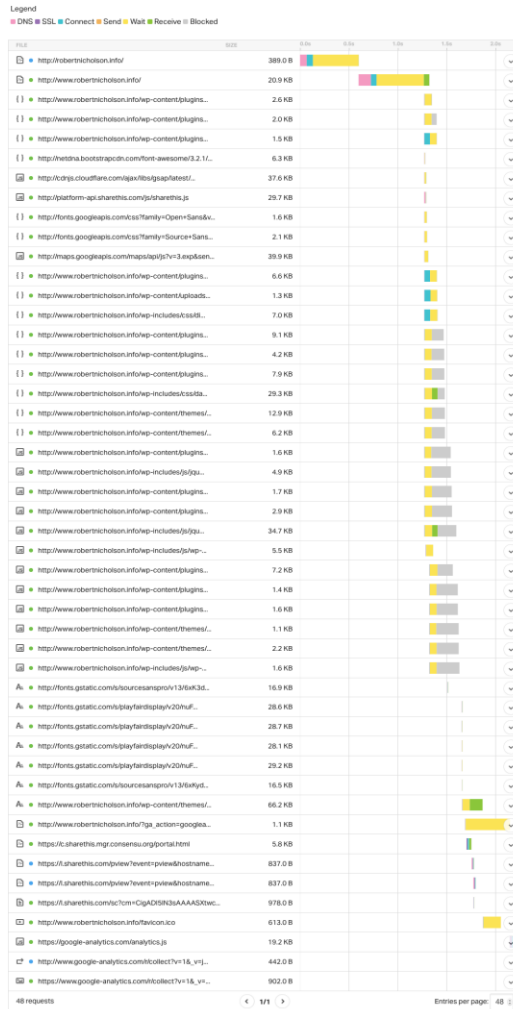
HTTPS has two functions

- Encrypts traffic between the browser and the web server
 - Allows transmission of sensitive information, such as bank account numbers
- Verifies the identity of the website by downloading a *certificate* issued by a recognized Certificate Authority (CA)

Additional Browser Functions

- “Build” web pages
 - Parse web pages looking for embedded content, such as images
 - Send additional HTTP requests to fetch embedded content
 - One web page may require *many* HTTP requests (hundreds)
- Process Cascading Style Sheets (CSS) to format pages
- Run embedded Javascript code
- Maintain a data model – the Document Object Model (DOM) – which allows Javascript to access and manipulate page elements
- Store local data associated with a particular website (*cookies*) – used for managing user *sessions*

Fetching and Building a Page



- The home page on my own website requires 48 separate fetches
<https://tools.pingdom.com/>
- Complex websites may require *hundreds* of fetches for a single page
- May require fetches from several domains
 - Ads
 - Shared font files
 - Services such as Google Analytics
- Handshakes may take a significant part of the overall load time

What Are Cookies?

- A cookie is a named data element, stored by the browser, and associated with a website
- Application code on a website can set a cookie, such as SESSION_ID
- On subsequent web page requests, the web application code can request the value of cookies the website set
 - This is used to allow logins and sessions
- Cookies can only be retrieved by the site that set them
 - ...but there is an important loophole!

Embedded Content

Click to go forward, hold to see history

California Mission Background and History

MISSION HOME MISSION OVERVIEW THE 21 MISSIONS RELATED HISTORY TEACHING MISSION NEWS CONTACT US



The California Missions

The California Missions represented the final expansion of the Spanish Empire. From 1769 to 1823, Spanish soldiers and monks built a total of 21 *Missions* and 5 *Presidios* (or military forts), stretching North from Mexico, along the Pacific coast, through the territory that was then known as Alta California. Over a short period – little more than 50 years – the Spanish brought a new culture to California, spreading European religion, agricultural practices, and eventually forms of government. The settlements around the missions became the seeds of modern California's major cities. The trail connecting the missions, *El Camino Real*, became California's first "highway," and its route is closely followed by modern Highway 101. The designs of the missions still influence California architecture. In a very real sense, California as we know it today would not exist without the foundation of the

missions.



History

Mission Guide

Mission Map

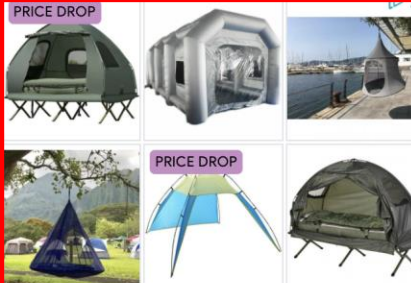
The Presidios

Fun Facts

Get the Book!

Embedded Ad (Google)

PRICE DROP



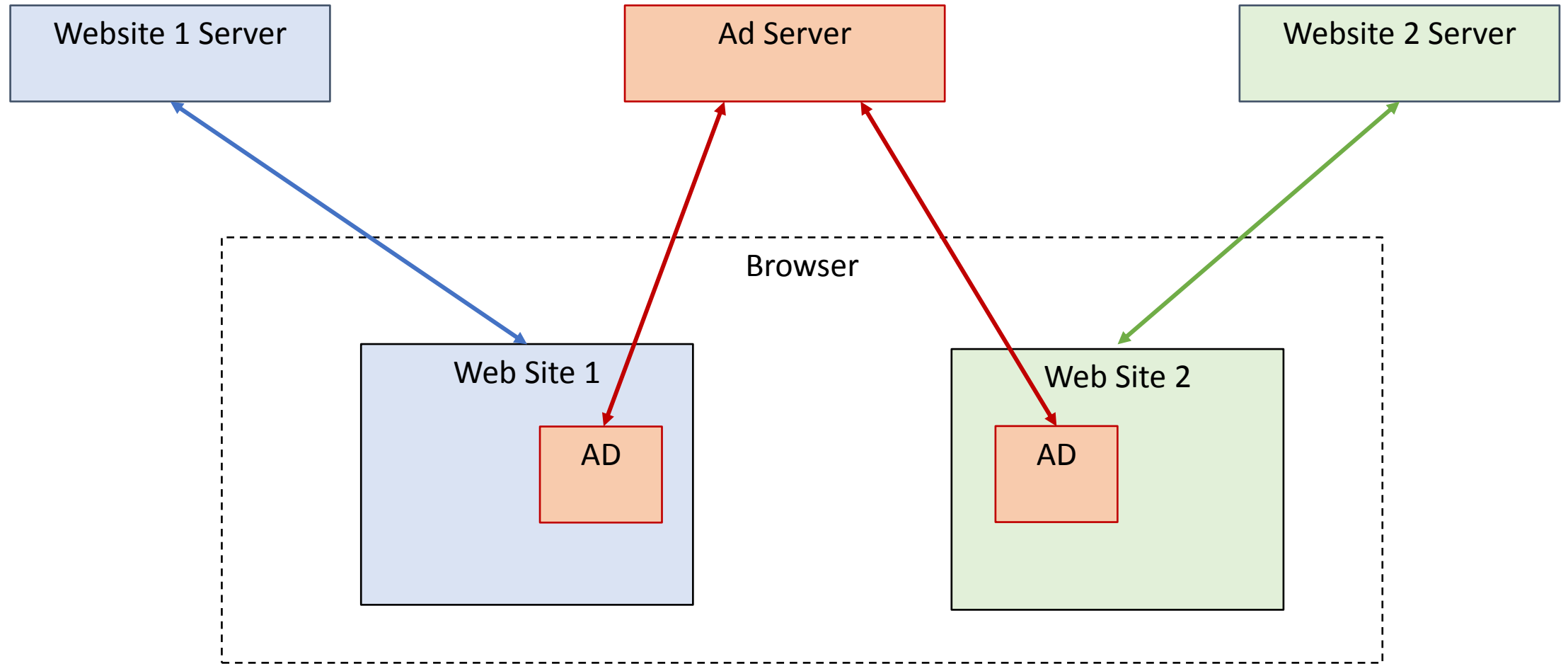
More unique pieces for less.

Wayfair

How Cookies Are Used for Tracking

- Remember that a single webpage may be built from many web requests
- If a website includes ads from an ad agency, such as Google, applications on that web server can set cookies
- If pages *on other websites* also include ads from the same agency, that web server can request the cookie value
- In this way, an ad agency can *track* the sites that you visit, and display similar ads

How Cookies Are Used for Tracking



Additional Web Server Functions

- Run back-end programs that dynamically build pages and send them to the browser
 - php
 - Java
 - Microsoft Active Server Pages (ASP)
 - Binary executable programs - Common Gateway Interface (CGI)
- Respond to additional protocols
- Cache commonly accessed pages to improve speed

Additional Web Protocols

- **HTTP get & HTTP post**
 - Two ways for a browser to submit *form data* to a program on the web server, which may then access databases and return appropriate page contents
- **JavaScript Object Notation (JSON):** allows downloading data from a web server, to be processed by Javascript programs in the browser
- **Simple Object Access Protocol (SOAP):** a protocol for exchanging structured data between a web client (browser) and a server
- **Asynchronous JavaScript And XML (AJAX):** a protocol for asynchronously retrieving web content to dynamically update web pages

Web Forms (get & post)

- Designated with an HTML form tag
<form action="/process-contact-form.php" method="get">
- *Action* code processes request and returns a web page
 - E.g. "Thank you for contacting us"

Get method

- Form fields are attached to the URL:
</process-contact-form.php&name=Robert&email=Robert@sjsu.edu>

Post Method

- Form fields are embedded in the HTTP request

Web Services

- A piece of software that provides a function or service over the Internet, using a standard (usually XML-based) interface
- Client applications make requests from the service, and get back results
- A web server is a type of API, but not all APIs are web services

Traditional Web Services (ca. 2007)

- Traditional web services are described by a service contract written in the Web Services Description Language (WSDL)
 - The WSDL document is an XML document that provides a machine-readable description of how the service can be called
 - The WSDL document and the request and response messages are transmitted over http or https
 - <https://www.w3.org/TR/2001/NOTE-wsdl-20010315>
 - W3 or W3C = World Wide Web Consortium

Traditional Web Services

- Messages use the Service Oriented Architecture Protocol (SOAP)
 - SOAP is also an XML-based format
 - https://www.w3schools.com/xml/xml_soap.asp
 - <https://www.w3.org/TR/soap/>

RESTful Web Services

- A simpler web services protocol is REST
 - Representational State Transfer
- A software *architecture style* consisting of guidelines and best practices for creating scalable web services
- REST is not an interface standard
 - <https://www.w3.org/2001/sw/wiki/REST>
 - <https://www.codecademy.com/article/what-is-rest>

The REST Architecture

- REST systems are *stateless*, meaning that the server does not need to know anything about what state the client is in and vice versa
- Client and server implementations are independent; the only connect is via messages
- In the REST architecture, clients send requests to retrieve or modify resources, and servers send responses to these requests

REST Requests

- REST requires that a client make a request to the server in order to retrieve or modify data on the server. A request generally consists of:
 - an HTTP verb, which defines what kind of operation to perform (GET, POST, PUT, DELETE)
 - a header, which allows the client to pass along information about the request
 - a path to a resource
 - an optional message body or *payload* containing data

REST Responses

- A REST Response consists of a status code and an option *payload*

Status code	Meaning
200 (OK)	This is the standard response for successful HTTP requests.
201 (CREATED)	This is the standard response for an HTTP request that resulted in an item being successfully created.
204 (NO CONTENT)	This is the standard response for successful HTTP requests, where nothing is being returned in the response body.
400 (BAD REQUEST)	The request cannot be processed because of bad request syntax, excessive size, or another client error.
403 (FORBIDDEN)	The client does not have permission to access this resource.
	The resource could not be found at this time. It is

Browser-Based Web Service Clients

- You can invoke many web services today from a web browser using AJAX
- You download JavaScript code from the web service provider that makes the AJAX calls.
 - Therefore, you don't worry about what protocol the web service provider uses
- Popular web service providers include Google, Facebook, Amazon, etc.

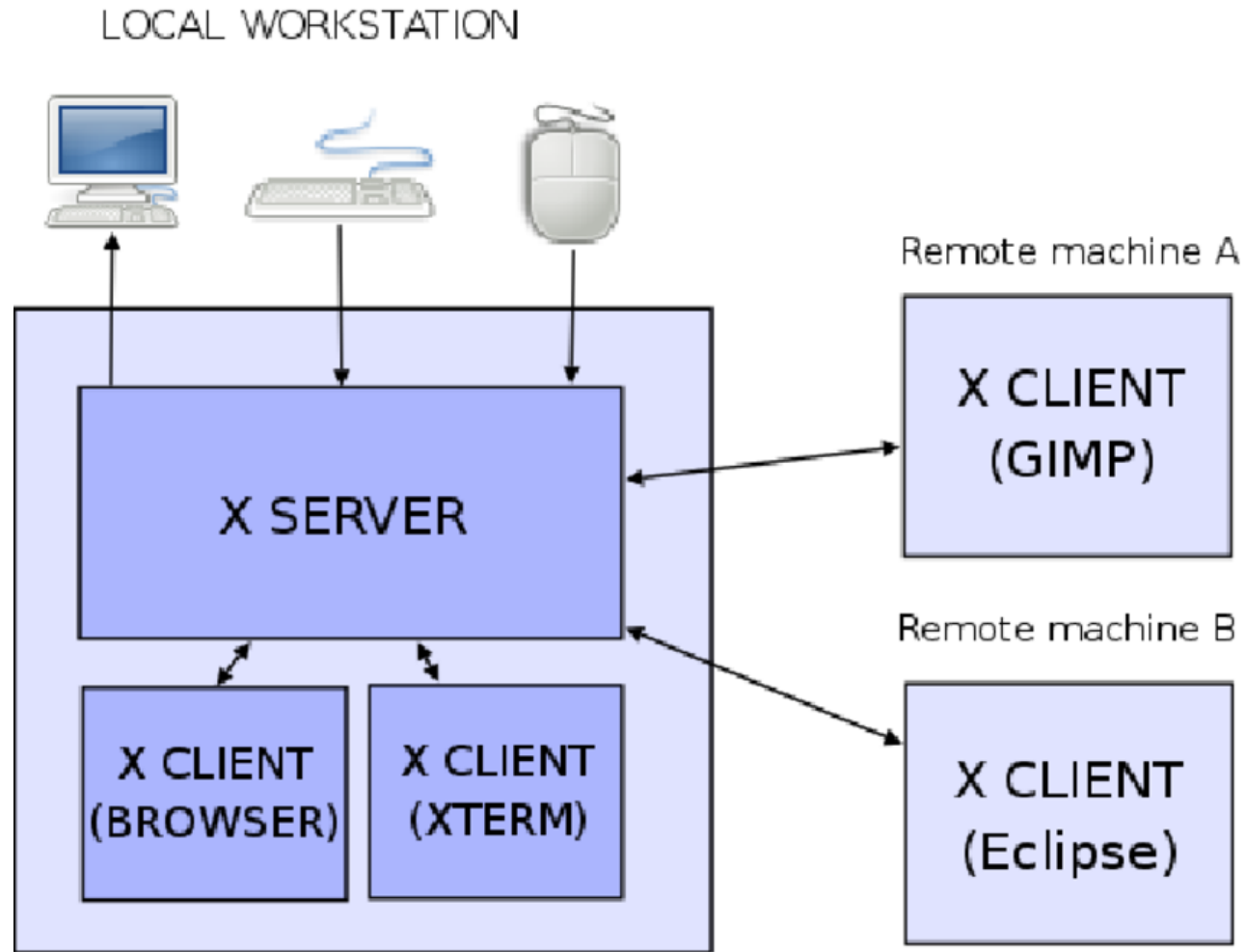
Web APIs

- APIs (Application Program Interfaces) include Javascript function calls as well as web service interfaces
- Function calls may hide the details of web services
- A web service is an API, but not all API are web services
- Additional differences
 - Web Services are network based (by definition)
 - APIs are protocol agnostic

Windowing Systems

- The window system (Graphic User Interface) on a desktop computer displays content from many programs
- The window system is a *server* – sometimes called a *display server*
- Application programs that want to display content or interact with the user are *clients*.
- GUI management and display features are typically implemented as function calls, which use inter-process communication to make requests from the windows server
- *Programs on another computer* can access the GUI and devices

Windowing Systems



Windowing Systems

- Microsoft Windows was originally a distinct component built on top of Microsoft DOS (Disk Operating System)
- In POSIX systems, Graphic User Interfaces are based on the X-
Windows server, developed at MIT in 1984
 - “The X Window System (X11) is an open source, cross platform, client-server computer software system that provides a GUI in a distributed network environment.”
- The Macintosh windowing system was based on Darwin, a derivative of X-Windows.

System Software

- System Software is not necessarily part of the operating system, but it includes specific knowledge of the underlying operating system and hardware, including:
 - Instruction set
 - Memory architecture and management
 - Process management
 - Network architecture
- System software enables program portability, by isolating programmers and users from underlying system details

System Software

- System software has grown in complexity
- Successfully writing system software requires:
 - A knowledge of system specifics
 - A knowledge of *algorithms* and *data structures* that have been developed over decades

For Next Class

- Log in to Canvas and complete Assignment 10