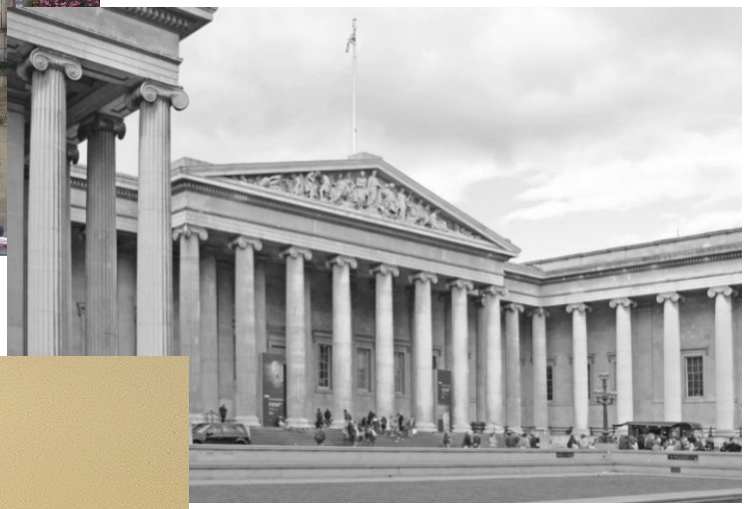
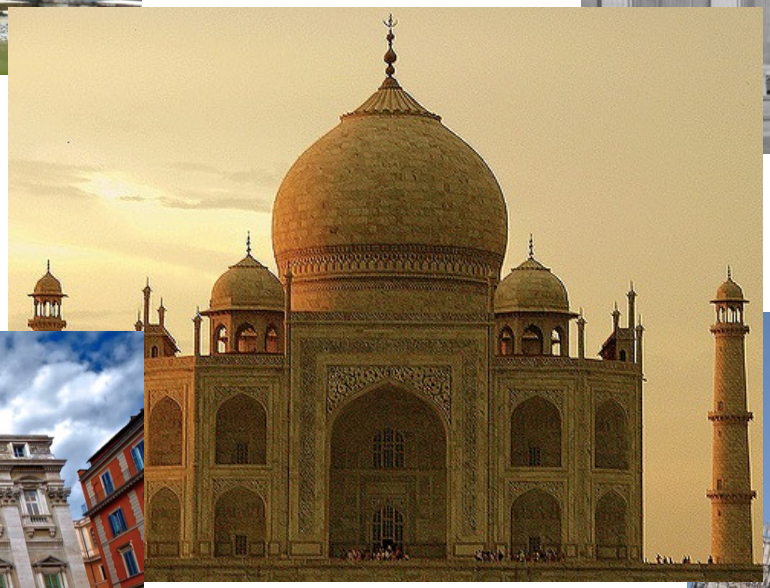
Can you get a better sense of location or area from architectures?





Images from Google images, March 4, 2019

Architecture Definitions

- "The complex or carefully designed structure of something"
  http://www.oxforddictionaries.com/definition/english/architecture November 13, 2015

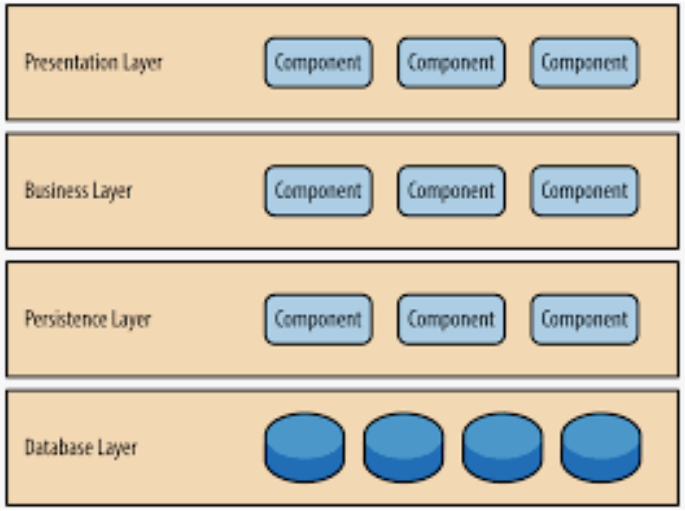
- "The conceptual structure and logical organization of a computer or computer-based system" http://www.oxforddictionaries.com/definition/english/architecture November 13, 2015

- In short, it is a common way that we have organized our software as a big design. Knowing the architecture tells you a bit about what to expect of the design and the behaviour of the program.

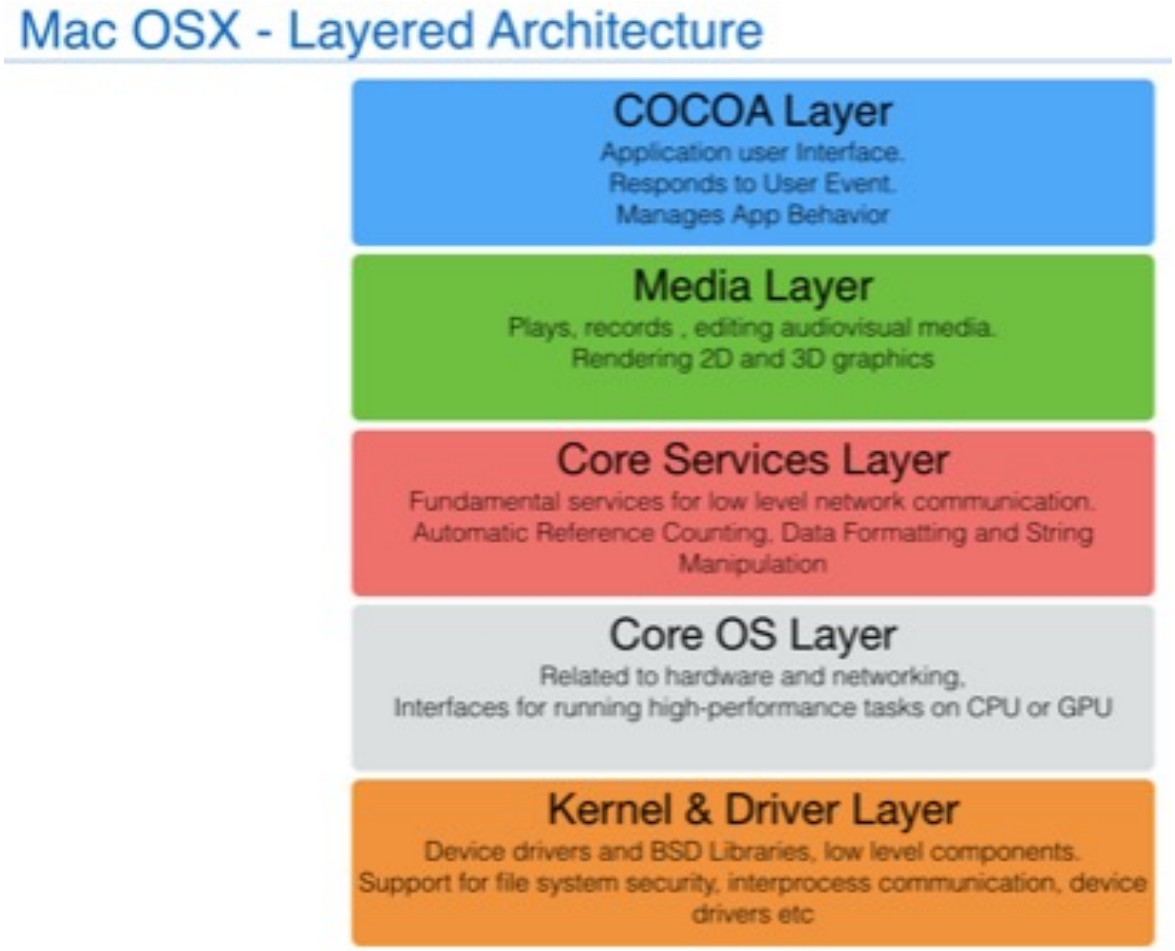
Layered Architecture

Layered Architecture

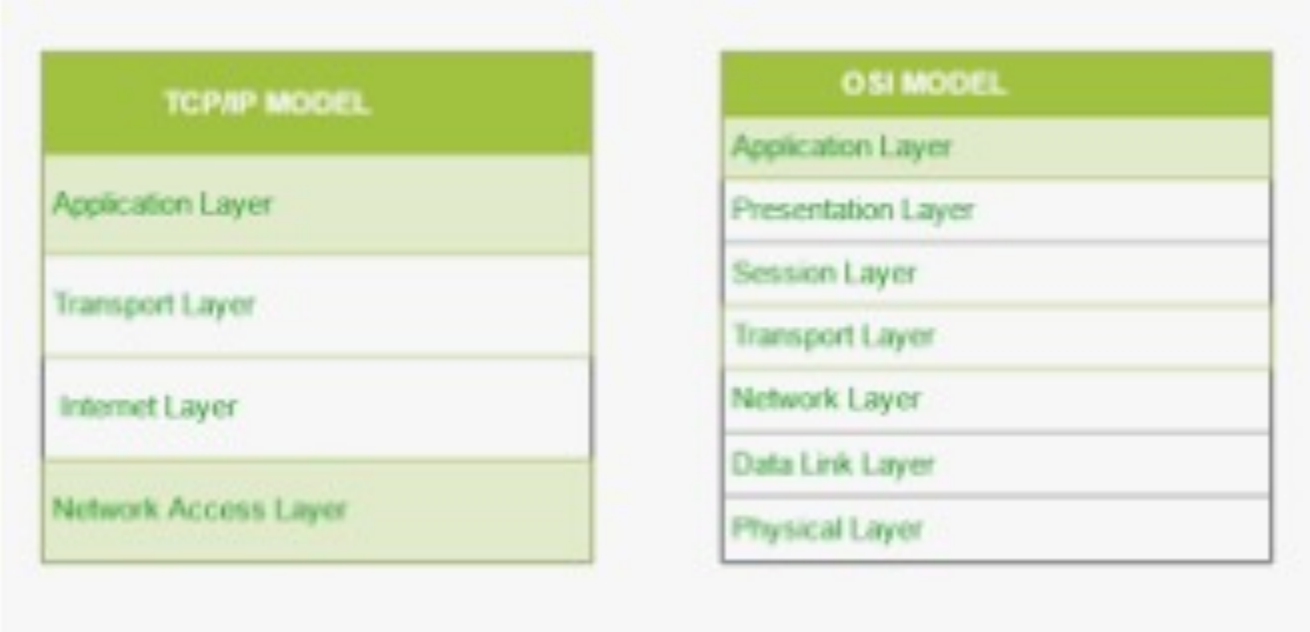
- Divide the responsibilities of the system into distinct layers.
  - Lower layers tend to be closer to the hardware, system, or resources
  - Upper layers tend to provide high-level abstractions and are closer to interacting with the user
- Layers only interact with the layer immediately above or immediately below it.
  - API definitions are critical.



Mac OSX



Computer Network Structure



Layered Architectures - Advantages

- Easy to understand
- Quick to locate where some tasks must be done
- Allows us to change the implementation of one layer without affecting the whole system
- Strong cohesion in each layer and loose coupling among layers

Layered Architectures - Disadvantages

- Overhead in traversing layers when two far-apart layers requires some interaction
- Deploying a new program in an existing layer framework may require you to create “empty” layer implementations just to fit the model
- Locks in a particular structure and size of program
  - Introducing many new responsibilities may require a refactoring of the architecture

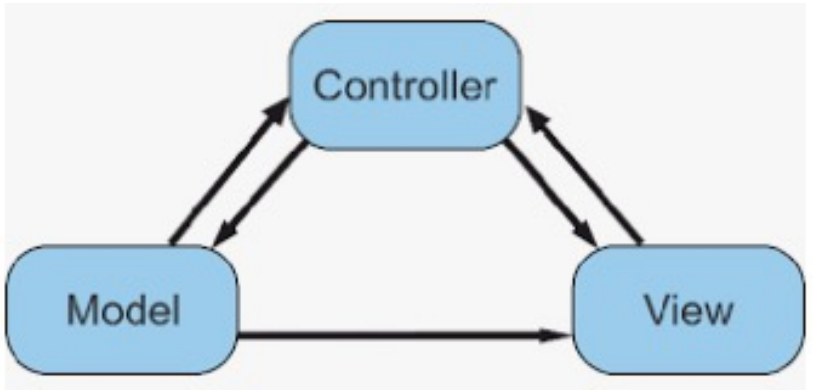
Layered Architectures

- Common when the system
  - Has a big range of responsibilities
  - Spans from low-level details to high-level user concepts

Event Driven Architecture

Model-View-Controller Architecture

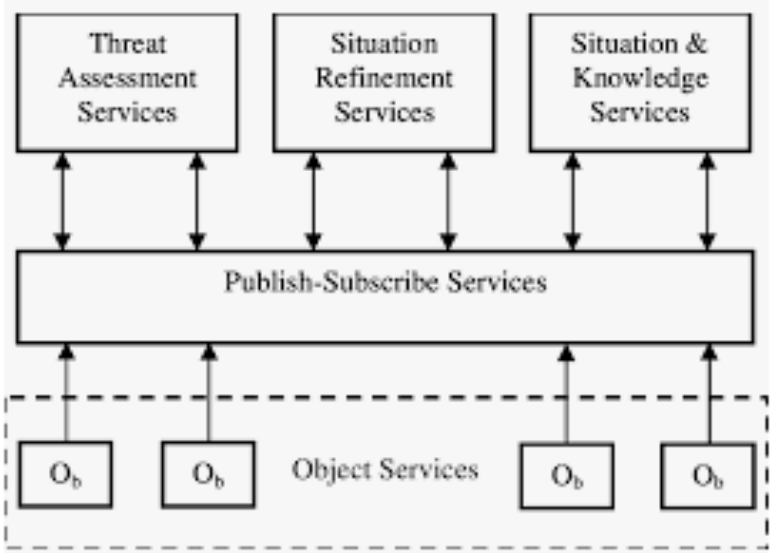
- An event-driven architecture developed for user interfaces.
- 3 main components:
  - The model does all of the calculations and manages the business rules
  - The view shows information to the user
  - The controller monitors for events from the user, from the model, or externally, and adapts the behavior of the model and the view accordingly



DALHOUSIE UNIVERSITY
Inspiring Minds

Publish-Subscribe Architecture

- Similar to event-driven work, except that agents in the system may only want to see a subset of events.
- The architecture allows agents to ask to receive (subscribe) only specific events.
- Agents that generate events only distribute them to other agents who have asked for the events.



Publish-Subscribe Architecture

- Typically needs some framework to manage the publishing and subscribing
  - Don’t want to build it on your own.
  - Early system called “Jabber” used pub-sub for a messaging system

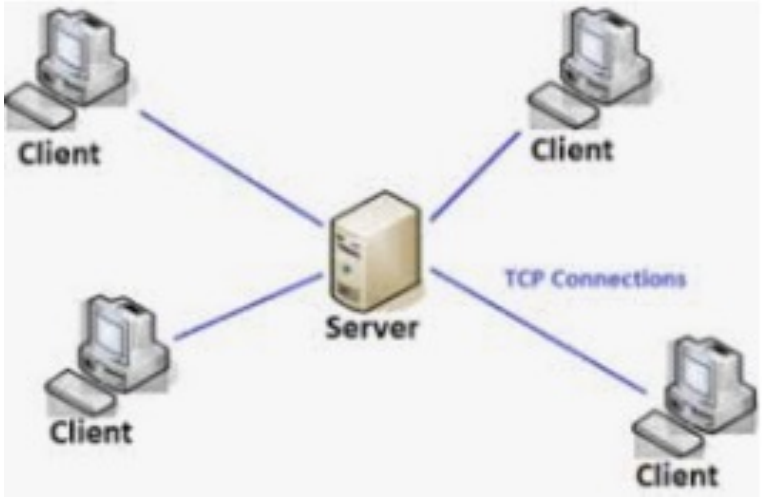
Distributed system architectures

- Client-server
  - 2-tier
  - 3-tier (presentation, domain logic, data storage)
    - Common for the web (client tier, web server tier, DBMS tier)
- Peer-to-peer
- Representational state transfer (REST)
- Service-oriented
  - Precursor to cloud systems

Client Server Architecture

Client Server Architecture

- Characterized by two different programs that communicate across the network:
  - The server has the data or service to offer
  - The client wants the data or service
- The server can speak with multiple clients at the same time.



DALHOUSIE UNIVERSITY
Inspiring Minds

Client Server Architecture

- The server
  - Has all the data / services
  - Is always present on the network
  - Waits around to be contacted by clients
    - Doesn't initiate contact with clients
- The client
  - Wants the data or service
    - Has no data or service to offer to others
  - Initiates the contact with the server
  - Can come and go
    - Is unreliable as far as the network is concerned

Client Server Advantages

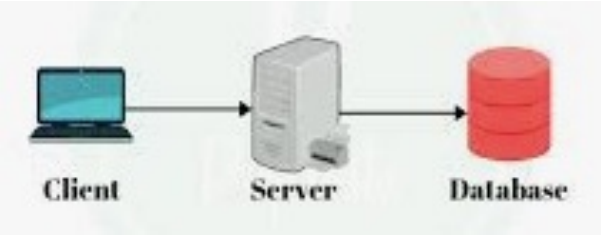
- Centralizes the data or services
- Is easy to locate the service
- Provides authoritative data by the server
- Clearly defined roles for the client and the server

Client Server Disadvantages

- Server risks being overloaded by too many clients
  - Use a distribution scheme across servers
- Loss of opportunity when you shut down the server for maintenance
  - Use a replication system to mitigate

3 Tier Architecture

- A variant of the Client Server architecture
- Add a third element which is a database to store information
- The client never accesses the database directly
- Deploy with
  - The database server in a network not accessible to the general Internet
  - The server in a protected area (called a demilitarized zone)
  - The client in the general Internet

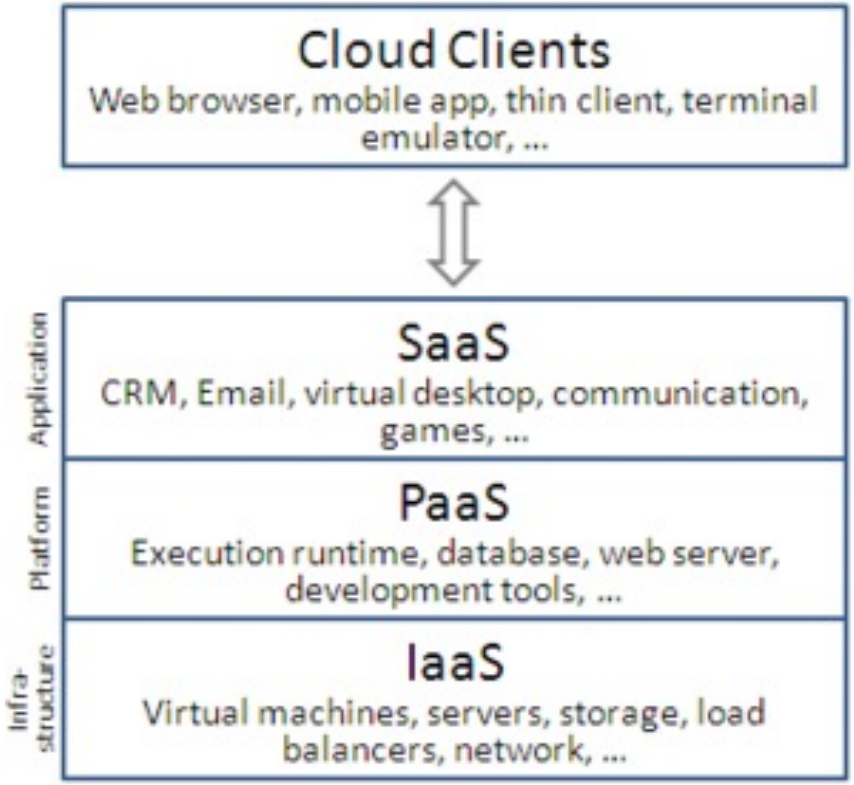


DALHOUSIE UNIVERSITY
Inspiring Minds

Cloud Architecture

Network “Architectures”

- Cloud
  - Your entire program is run / serviced on someone else’s hardware
  - Infrastructure as a service (IaaS)
    - eg. Amazon EC2, IBM Blue Cloud, FlexiScale
  - Platform as a service (PaaS)
    - eg. Google App engine
  - Software as a service (SaaS)
    - eg. Salesforce.com, Google docs, webmail



Cloud Computing

- **Definitions vary:**
  - "A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet" [Foster et al., 2008]
  - "A style of computing where scalable and elastic IT capabilities are provided as a service to multiple external customers using Internet technologies." [Plummer et al., 2008]
  - "Cloud computing is a model for enabling 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." [Mell and Grance, 2010]

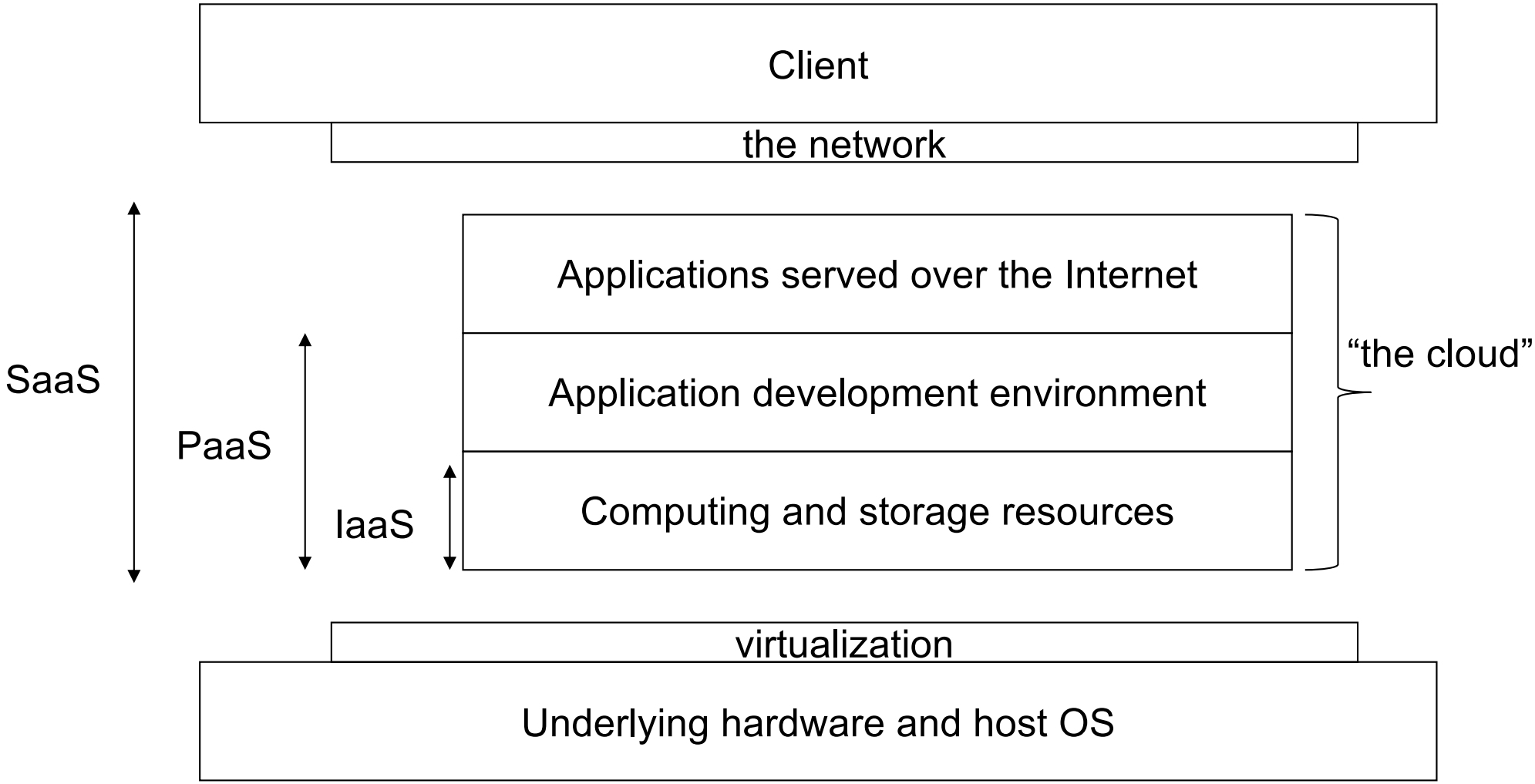
Cloud Computing Characteristics

- **Common elements:**
  - **Virtualization**
    - hardware can host many independent simulated servers
  - **Multi-tenancy**
    - multiple clients can occupy the same physical hardware
  - **Security**
    - clients are protected from each other and their data is secure
  - **Elasticity**
    - resources can be added and removed in real-time, often at the request of the client and without the intervention of the service provider

Cloud Computing Characteristics

- **Common elements:**
  - **Availability**
    - the service provider gives performance / QoS guarantees
  - **Reliability**
    - failure of any piece still allows the services to be offered
  - **Agility**
    - the resource allocations can adapt dynamically
  - **Pay-as-you-go**
    - the client just pays for the resources used
- **Clients use the same way to get to the service no matter how or where the service is deployed**

Cloud Computing



Hybrid Architectures

Hybrid Architectures

- Often, no single architecture fits what we want
  - System is big enough that sub-parts have different characteristics
  - Some disadvantages of an architecture need to be overcome
- We can combine architectures to create a hybrid architecture
  - Use one architecture for the high level and others for lower levels
  - Use two systems in parallel to complement one another, each system with a different architecture

Hybrid Architectures

- Good hybrid architectures have the elements complement one another
  - Eg. Client-server index for a peer-to-peer system
    The Mac OS that is layered and each layer has its own sub-architecture (microkernel for the core OS)

- Bad hybrid architectures lose the clarity of the component architectures and can add confusion

Databases

What are we dealing with?

- **Data**
  - Stored representations of objects and events that have meaning and importance in the user's environment
- **Information**
  - Data that has been processed in such a way as to increase the knowledge of the person who uses the data
- **Metadata**
  - Data that describe the properties or characteristics of end-user data, and the context of that data.
  - Eg: name, type, range restrictions on numeric data, …
- **Database management system (DBMS)**
  - A software system that is used to create, maintain, and provide controlled access to user databases.

Definitions from "Modern Database Management" by Jeffrey Hoffer, Mary Prescott, and Heikki Topi, 9th edition

Database basics

- Concerned with entities and the relations between the entities.
  - An entity is a person, place, object, event, or concept in the user environment about which the organization wishes to maintain data.
- We will focus on relational databases
  - A database that represents data as a collection of tables in which all data relationships are represented by common values in related tables.

Sample table of data
(from excel)

Term	Subject	Course Nu	Section	CRN	Schedule	Monday	Tuesday	Wednesday	Thursday	Friday	Begin Time	End Time	Building	Room
201810	ACAD	20	1	18188	L			W			935	1025	HALEY INSTIT	116
201810	ACAD	1050	1	18112	L	M		W			1005	1125	BANTING BU	32
201810	ACSC	4703	1	14095	L		T		R	F	1035	1125	CHASE BLDG	319
201810	ACSC	4720	1	14096	L	M		W		F	1135	1225	CHASE BLDG	319
201810	ACSC	4950	1	18296	L									
201810	AGRI	1000	1	14097	L		T		R		1005	1125	AGRICULTUR	24
201810	AGRI	1000	B01	14098	B					F	1235	1425	AGRICULTUR	260
201810	AGRI	1000	B01	14098	B					F	1235	1425	AGRICULTUR	261
201810	AGRI	1000	B01	14098	B					F	1235	1425	AGRICULTUR	262
201810	AGRI	1000	B02	14099	B					F	1435	1625	AGRICULTUR	260
201810	AGRI	1000	B02	14099	B					F	1435	1625	AGRICULTUR	261
201810	AGRI	1000	B02	14099	B					F	1435	1625	AGRICULTUR	262
201810	AGRI	4000	1	14102	L			W			1735	2025	HALEY INSTIT	110

- **Regard data about specific entities as relations**
  - **Represent data as tuples where each component of a tuple has some domain**
    - **Eg. (201810, ACAD, 20, 1, 18188, …) where**
      201810 is a string that represents the term
      ACAD is a 4 character string that is one of a set of course subjects
      20 is an integer as the course number
      1 is string
      18188 is a 5 digit integer

Advantages of a DBMS

- Program-data independence
- Planned data redundancy (or removal of redundancy)
- Improved data consistency
- Improved data sharing
- Increased productivity of application development
- Enforcement of standards
- Improved data quality
- Improved data accessibility and responsiveness
- Reduced program maintenance
- Improved decision support

Disadvantages of a DBMS

- New, specialized personnel
- Installation and management cost and complexity
- Conversion cost
- Need for explicit backup and recovery
- Organizational conflict

Some more terminology…

- **Schema**
  - The structure that contains descriptions of objects created by a user, such as base tables, views, and constraints, as part of a database.
- **Catalog**
  - A set of schemas that, when put together, constitute a description of a database.

Schemas

- **External data model**
  - The view of users of the database
    - Some users may operate through a *database view* and not see all data.
- **Conceptual schema**
  - A detailed, technology-independent specification of the overall structure of the organizational data.
    - Covers all external views of the data.
- **Internal schema**
  - Logical schema
    - The representation of a database for a particular data management technology
  - Physical schema
    - Specifications for how data from a logical schema are stored in a computer's secondary memory by a DBMS