System Architectures Outline Definition of System Architectures System Architectures Diagram Types of System Architectures Benefits of System Architectures System Architectures Definition System architecture is the conceptual model that defines the structure, behavior, and more views of a system. It's a representation of a system in which there is a mapping of functionality onto hardware and software components, a mapping of the software architecture onto the hardware architecture, and human interaction with these components. The architectural components and set of relationships between these components that an architecture describes may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people System Architectures Definition (Cont.) It describes the physical placement of software components. System architecture describes how all system components interconnect and how data links between them. Architecture can function as a guide or a framework for designing and developing a new system or can assist in defining a project's goals. System Architecture Diagram The system architecture diagram is a visual representation of the system architecture. It shows the connections between the various components of the system and indicates what functions each component performs. The general system representation shows the major functions of the system and the relationships between the various system components. Types of System Architectures Hardware Architecture Software Architecture Enterprise Architecture Hardware Architecture Hardware Architecture refers to the identification of the physical components and their interrelationships, which allows hardware designers to understand how their components fit into a system architecture. It provides software component designers important information needed for software development and integration. Parts of a computer hardware architecture: ● Central processing Unit ● Main Memory ● Secondary Memory ● I/O Devices ● Network Connection Hardware Architecture (Cont.) Central Processing Unit (CPU): is the part of the computer that is built to be obsessed with "what is next?". ● What is next means what instructions should I perform next. ● CPU performs insurrections that already stored in the main memory. Main Memory: is used to store information that the CPU needs in a hurry. ● The main memory is nearly as fast as the CPU. ● The information stored in the main memory vanishes when the computer is turned off. Hardware Architecture (Cont.) Secondary Memory is also used to store information, but it is much slower comparing to the main memory. The advantage of the secondary memory is that it can store information even when the power is off "Permanent". Ex; flash memory. I/O Devices are simply our screen, keyboard, mouse, microphone, speaker, touchpad, etc. They are all the ways we interact with the computer. Network Connection is used to retrieves information over a network. It is a slower and at times unreliable form of Secondary Memory. Hardware Architecture (Cont.) Software Architecture Software architecture refers to the logical organization of a distributed system into software components. Instead of one big monolithic application, distributed systems are broken down into multiple components. The way in which these components are broken down impacts everything from system performance to reliability to response latency. Software Architecture (Cont.) There are many architecture styles which can be applied on software, such as; ● Layered architecture ● Publish–Subscribe architecture Software Architecture (Cont.) Layered architecture In a layered architecture, components are organized in layers. Components on a higher layer make downcalls (send requests to a

lower layer). While lower layer components can make upcalls (send requests up), they usually only respond to higher layer requests. Google drive is a good example of layered architecture, where it shows clearly how the three layered are connecting; 1. Interface layer: you request to see the latest doc from your drive. 2. Processing layer: processes your request and asks for the information from the data layer. 3. Data layer: stores persistent data (aka your file) and provides access to higher-level layers. Software Architecture (Cont.) ● The data layer returns the information to the processing layer which in turn sends it to the interface where you can view and edit it. ● While it feels like one cohesive process, it's broken down into three (or more) components on three distinct layers. ● Each layer may or may not be placed on a different machine (this is a system architecture consideration). Software Architecture (Cont.) Publish-Subscribe architecture or Pub/Sub is a messaging service where the senders of messages are decoupled from the receivers of messages. There are several key concepts in a Pub/Sub service: ● Message: the data that moves through the service. ● Topic: a named entity that represents a feed of messages. ● Subscription: a named entity that represents an interest in receiving messages on a particular topic. ● Publisher (also called a producer): creates messages and sends (publishes) them to the messaging service on a specified topic. ● Subscriber (also called a consumer): receives messages on a specified subscription Software Architecture (Cont.) Publishers decide what topics their messages will belong to. Event bus filters the messages by topic before delivering them into the relevant subscribers. Ex; if publisher sends a massage with topic A, massage will be forwarded to any subscribers who have subscribed to topic A. Similarly, a massage with topic B will be delivered to sub scribers of Topic B. News papers, ads, and social networks are examples of this style. Enterprise Architecture Enterprise architecture (EA) is the practice of analyzing, designing, planning, and implementing enterprise analysis to successfully execute on business strategies. EA helps organizations to structure IT projects and policies to achieve desired business results, to stay agile and resilient in the face of rapid change, and to stay on top of industry trends and disruptions using architecture principles and practices, a process also known as enterprise architectural planning (EAP). It is especially useful for large businesses going through digital transformation, because it focuses on bringing legacy processes and applications together to form a more seamless environment. Enterprise Architecture EA is guided by the organization's business requirements and that will help lay out how information, business, and technology flow together. This has become a priority for businesses that are trying to keep up with new technologies such as the cloud, IoT, machine learning, and other emerging trends that will prompt digital transformation. Zachman Framework provides a structure for organizing information about an organization's business, processes, data, applications, and technology. one of the main framework used for enterprise architecture. Benefits of System Architectures ● Vision Development and implementation ● Facilitate Faster IT System Changes ● Ensure that the IT plans align with business programs ● Analyze potential cost-saving opportunities Virtualization Outline Definition of Virtualization Definition of Virtual Machine (VM) Role and types of Hypervisor Types of virtualization Benefits of virtualization Definition of Virtualization Virtualization is a process that allows for more efficient utilization of physical computer hardware and is the foundation of cloud computing. Virtualization uses software to create an abstraction layer over computer hardware that allows the

hardware elements of a single computer— processors, memory, storage and more—to be divided into multiple virtual computers, commonly called virtual machines (VMs). Each VM runs its own operating system (OS) and behaves like an independent computer Definition of Virtual Machine (VM) Virtual machine (VM) is a virtual environment that simulate a physical compute in software form. It normally comprise several files containing the VM's configuration, the storage for the virtual hard drive, and some snapshots of the VM that preserve its state at a particular point in time. A VM cannot interact directly with a physical computer. Instead, it needs a lightweight software layer called a hypervisor to coordinate between it and the underlying physical hardware. Role and types of Hypervisor A hypervisor is the software layer that coordinates/manage VMs. It serves as an interface between the VM and the underlying physical hardware, ensuring that each has access to the physical resources it needs to execute (1). It also ensures that the VMs don't interfere with each other by impinging on each other's memory space or compute cycles (2). Hypervisor has two types: Type 1 Type 2 Role and types of Hypervisor(Cont.) Type1 interact with the underlying physical resources, replacing the traditional operating system altogether. They most commonly appear in virtual server scenarios. Type2 run as an application on an existing OS. Most commonly used on endpoint devices to run alternative operating systems, they carry a performance overhead because they must use the host OS to access and coordinate the underlying hardware resources. Types of Virtualization Desktop Virtualization Network Virtualization Storage Virtualization Application Virtualization Types of virtualization (Cont.) Desktop Virtualization allow user to run multiple desktop operating systems, each in its own VM on the same computer. There are two types of desktop virtualization: ● Virtual desktop infrastructure (VDI) runs multiple desktops in VMs on a central server and streams them to users who log in on thin client devices. In this way, VDI allows an organization provide its users access to variety of OSs from any device, without installing OSs on any device. ● Local desktop virtualization runs a hypervisor on a local computer, enabling the user to run one or more additional OSs on that computer and switch from one OS to another as needed without changing anything about the primary OS. Types of virtualization (Cont.) Network Virtualization Network virtualization uses software to create a "view" of the network that an administrator can use to manage the network from a single console. It abstracts hardware elements and functions (e.g., connections, switches, routers, etc.) Software-Defined Networking (SDN) virtualizes hardware that controls network traffic routing (called the "control plane") Network Function Virtualization (NFV) virtualizes one or more hardware appliances that provide a specific network function (e.g., a firewall, load balancer, or traffic analyzer), making those appliances easier to configure, provision, and manage. Types of virtualization (Cont.) Storage Virtualization Storage virtualization enables all the storage devices on the network, whether they're installed on individual servers or standalone storage units to be accessed and managed as a single storage device. Storage virtualization makes it easier to provision storage for VMs and makes maximum use of all available storage on the network. Application Virtualization runs application software without installing it directly on the user's OS. This differs from complete desktop virtualization because only the application runs in a virtual environment, while the OS on the end user's device runs as usual. Types of virtualization (Cont.) Types of application virtualization: ● Local Application Virtualization The entire application runs on the

endpoint device but runs in a runtime environment instead of on the native hardware. ● Application Streaming The application lives on a server which sends small components of the software to run on the end user's device when needed. ● Server-based Application Virtualization The application runs entirely on a server that sends only its user interface to the client device. Benefits of virtualization Resource efficiency Easier management Minimal downtime Faster provisioning