University of Science and Technology
Faculty of Computer Science and Information Technology
Information Security Master Program
Course Title: Cloud Computing Security

Lecture (1): Introduction to Cloud Computing

Reference: Cloud Computing Security and Privacy, by Tim Mather, Subra and Lattif, Chapter (2)

Instructor: Prof. Noureldien Abdelrahman

This lecture describes:

- A Cloud computing definition
- Cloud computing technology components
- Cloud services delivery models
- Cloud deployment modes
- Key drivers for adopting the cloud
- Barriers to cloud computing adoption in the enterprise

1.1 What is Cloud Computing?

Our definition of cloud computing is based on five attributes: **multitenancy** (shared resources), **massive scalability**, **elasticity**, **pay as you go**, and **self-provisioning of resources**.

1- Multitenancy (shared resources)

Unlike previous computing models, which assumed dedicated resources (i.e., computing facilities dedicated to a single user or owner), cloud computing is based on a business model in which resources are shared (i.e., multiple users use the same resource) at the network level, host level, and application level.

2- Massive scalability

Although organizations might have hundreds or thousands of systems, cloud computing provides the ability to scale to tens of thousands of systems, as well as the ability to massively scale bandwidth and storage

space.

3- Elasticity

Users can rapidly increase and decrease their computing resources as needed, as well as release resources for other uses when they are no longer required.

4- Pay as you go

Users pay for only the resources they actually use and for only the time they require them.

5- Self-provisioning of resources

Users self-provision resources, such as additional systems (processing capability, software, storage) and network resources.

1.2 Relevant Technologies in Cloud Computing

Cloud computing isn't so much a technology as it is the combination of many preexisting technologies. These technologies have matured at different rates and in different contexts, and were not designed as a coherent whole; however, they have come together to create a technical ecosystem for cloud computing. The following are examples for technologies that create the cloud.

1. **Cloud access devices**

The range of access devices for the cloud has expanded in recent years. Home PCs, enterprise PCs, network computers, and mobile phone devices.

2. High-speed broadband access

A critical component of the cloud is the broadband network, which offers the means to connect components and provides one of the substantial differences from the utility computing concept of 30 years ago.

Broadband access is now widely available, especially in global metropolitan areas. Nearly pervasive wireless access (e.g., WiFi, cellular, emerging WiMAX) is available, which has established mobile devices as entry points to the IT resources of the enterprise and the cloud.

3. Data centers and server farms

Cloud-based services require large computing capacity and are hosted in data centers and server farms. These distributed data centers and server farms span multiple locations and can be linked via internetworks providing distributed computing and service delivery capabilities.

4. Storage devices

Decreasing storage costs and the flexibility with which storage can be deployed have changed the storage landscape. The fixed direct access storage device (DASD) has been replaced with storage area networks (SANs), which have reduced costs and allowed a great deal more flexibility in enterprise storage. SAN software manages integration of storage devices and can independently allocate storage space on demand across a number of devices.

5. Virtualization technologies

In computing, **virtualization** means to create a virtual version of a device or resource, such as a server, storage device, network or even an

operating system where the framework divides the resource into one or more execution environments.

Virtualization technologies enable multitenancy cloud business models by providing a scalable, shared resource platform for all tenants. More importantly, they provide a dedicated resource view for the platform's consumers.

From an enterprise perspective, virtualization offers data center consolidation and improved IT operational efficiency. Today, enterprises have deployed virtualization technologies within data centers in various forms, including OS virtualization (VMware, Xen), storage virtualization (NAS, SAN), database virtualization, and application or software virtualization (Apache Tomcat, JBoss, Oracle App Server, WebSphere).

6. APIs

A suitable application programming interface (API) is another enabler for the cloud computing services delivery model. PIs empower users by enabling features such as self-provisioning and programmatic control of cloud services and resources.

APIs offered by cloud service providers (CSPs) such as Amazon EC2, Sun Cloud, and Go-Grid allow users to create and manage cloud resources, including compute, storage, and networking components.

1.3 The Cloud Services Delivery Model

A commonly agreed upon framework for describing cloud computing services goes by the acronym "SPI." This acronym stands for the three major services provided through the cloud: Software-as-a-service (*S*aaS), Platform-as-a-service (*P*aaS), and Infrastructure-as-a-service (*I*aaS). Figure 2-3 illustrates the relationship between services, uses, and types of clouds.

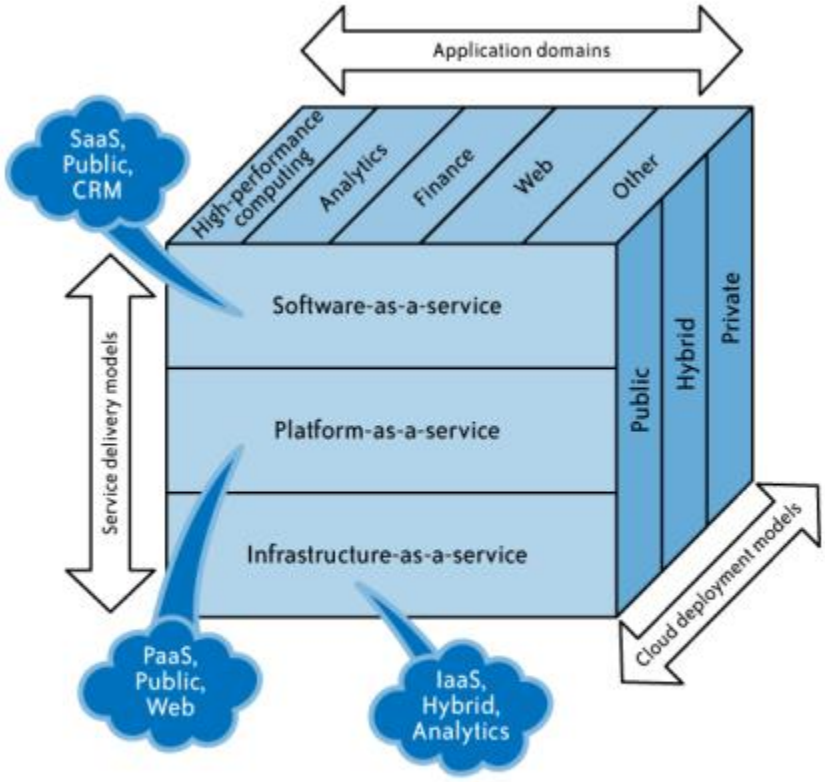

FIGURE 2-3. SPI service model

1. The Software-As-a-Service Model

Traditional methods of purchasing software involved the customer loading the software onto his own hardware in return for a license fee (a capital expense, known as *CapEx*). The customer could also purchase a maintenance agreement to receive patches to the software or other

support services. The customer was concerned with the compatibility of operational systems, patch installations, and compliance with license agreements.

In a SaaS model, the customer does not purchase software, but rather rents it for use on a subscription or pay-per-use model (an operational expense, known as *OpEx*).

Key benefits of a SaaS model include the following:

• SaaS enables the organization to outsource the hosting and management of applications to a third party (software vendor and service provider) as a means of reducing the cost of application software licensing, servers, and other infrastructure and personnel required to host the application internally

• SaaS enables software vendors to control and limit use, prohibits copying and distribution, and facilitates the control of all derivative versions of their software.

• Applications delivery using the SaaS model typically uses the one-to-many delivery approach, with the Web as the infrastructure. An end user can access a SaaS application via a web browser; some SaaS vendors provide their own interface that is designed to support features that are unique to their applications.

• A typical SaaS deployment does not require any hardware and can run over the existing Internet access infrastructure. Sometimes changes to firewall rules and settings may be required to allow the SaaS application to run smoothly.

• Management of a SaaS application is supported by the vendor from the end user perspective, whereby a SaaS application can be configured using an API, but SaaS applications cannot be completely customized.

What is the architectural difference between traditional software model and SaaS model?

1. The single most important architectural difference between the traditional software model and the SaaS model is the number of tenants the application supports.

The traditional software model is an isolated, single-tenant model, which means a customer buys a software application and installs it on a server. The server runs only that specific application and only for that single customer's end user group.

The SaaS model is a multitenant architecture model, which means the physical backend hardware infrastructure is shared among many different customers, but logically is unique for each customer.

2. The Platform-As-a-Service Model

In a platform-as-a-service (PaaS) model, the vendor offers a development environment to application developers, who develop applications and offer those services through the provider's platform.

The provider typically develops toolkits and standards for development, and channels for distribution and payment. The provider typically receives a payment for providing the platform and the sales and distribution services. This enables rapid propagation of software applications, given the low cost of entry and the leveraging of established channels for customer acquisition.

PaaS is a variation of SaaS whereby the development environment is offered as a service. The developers use the building blocks (e.g., predefined blocks of code) of the vendor's development environment to create their own applications.

PaaS solutions are development platforms for which the development tool itself is hosted in the cloud and accessed through a browser. With PaaS, developers can often build web applications without installing any tools on their computer, and can then deploy those applications without any specialized system administration skills.

At a minimum, a PaaS solution should include the following elements:

• A PaaS development studio solution should be browser-based.

• An end-to-end PaaS solution should provide a high-productivity integrated development environment (IDE) running on the actual target delivery platform so that debugging and test scenarios run in the same environment as production deployment.

• A PaaS solution should provide integration with external web services and databases.

• A PaaS solution must provide comprehensive monitoring of application and user activity, to help developers understand their applications and effect improvements.

• Scalability, reliability, and security should be built into a PaaS solution without requiring additional development, configuration, or other costs. Multitenancy (the ability for an application to automatically partition state and data to service an arbitrary number of users) must be assumed without additional work of any sort.

• A PaaS solution must support both formal and on-demand collaboration throughout the entire software life cycle (development, testing, documentation, and operations), while maintaining the security of source code and associated intellectual property.

• A PaaS solution should support pay-as-you-go metered billing.

What is the architectural difference between traditional Software development model and PaaS Model?

PaaS platforms also have functional differences from traditional development platforms, including:

Multitenant development tools

Traditional development tools are intended for a single user; a cloud-based studio must support multiple users, each with multiple active projects.

Multitenant deployment architecture

Scalability is often not a concern of the initial development effort and is left instead for the system administrators to handle when the project deploys. In PaaS, scalability of the application and data tiers must be built-in (e.g., load balancing and failover should be basic elements of the developing platform).

Integrated management

Traditional development solutions (usually) are not associated with runtime monitoring, but in PaaS the monitoring ability should be built into the development platform.

Integrated billing

PaaS offerings require mechanisms for billing based on usage that are unique to the SaaS world.

3. The Infrastructure-As-a-Service Model

In the traditional hosted application model, the vendor provides the entire infrastructure for a customer to run his applications. Often, this entails housing dedicated hardware that is purchased or leased for that specific application.

The IaaS model also provides the infrastructure to run the applications, but the cloud computing approach makes it possible to offer a pay-peruse model and to scale the service depending on demand.

Features available for a typical IaaS system include:

Scalability The ability to scale infrastructure requirements, such as computing resources, memory, and storage (in near-real-time speeds) based on usage requirements.

Pay as you go

The ability to purchase the exact amount of infrastructure required at any specific time.

Best-of-breed technology and resources

Access to best-of-breed technology solutions and superior IT talent for a fraction of the cost.

1.4 Cloud Deployment Models

The term *cloud* is a metaphor for the Internet and is a simplified representation of the complex, internetworked devices and connections that form the Internet. Private and public clouds are subsets of the Internet and are defined based on their relationship to the enterprise. Private and public clouds may also be referred to as *internal* or *external* clouds; the differentiation is based on the relationship of the cloud to the enterprise.

Public Clouds

Public clouds (or external clouds) describe cloud computing in the traditional mainstream sense, whereby resources are dynamically provisioned on a fine-grained, self-service basis over the Internet, via web applications or web services, from an off-site, third-party provider

who shares resources and bills on a fine-grained, utility-computing basis. A public cloud is hosted, operated, and managed by a third-party vendor from one or more data centers. The service is offered to multiple customers (the cloud is offered to multiple tenants) over a common infrastructure.

In a public cloud, security management and day-to-day operations are relegated to the third party vendor, who is responsible for the public cloud service offering. Hence, the customer of the public cloud service offering has a low degree of control and oversight of the physical and logical security aspects of a private cloud.

Private Clouds

Private clouds and *internal clouds* are terms used to describe offerings that emulate cloud computing on private networks. These products claim to deliver some benefits of cloud computing without the pitfalls, capitalizing on data security, corporate governance, and reliability concerns. Organizations must buy, build, and manage them and, as such, do not benefit from lower upfront capital costs and less hands-on management. The organizational customer for a private cloud is responsible for the operation of his private cloud.

Private clouds differ from public clouds in that; the network, computing, and storage infrastructure associated with private clouds is dedicated to a single organization and is not shared with any other organizations (i.e., the cloud is dedicated to a single organizational tenant).

As such, a variety of private cloud patterns have emerged:

Dedicated

Private clouds hosted within a customer-owned data center or at a collocation facility, and operated by internal IT departments.

Community

Private clouds located at the premises of a third party; owned, managed, and operated by a vendor who is bound by custom service-level agreements (SLAs) and contractual clauses with security and compliance requirements.

Managed

Private cloud infrastructure owned by a customer and managed by a vendor.

In general, in a private cloud operating model, the security management and day-to-day operation of hosts are relegated to internal IT or to a third party with contractual SLAs. By virtue of this direct governance model, a customer of a private cloud ***should have a high degree of control and oversight of the physical and logical security aspects of the private cloud infrastructure***—both the hypervisor and the hosted virtualized OSs. With that high degree of control and transparency, it is easier for a customer to comply with established corporate security standards, policies, and regulatory compliance.

Hybrid Clouds

A hybrid cloud environment consisting of multiple internal and/or external providers is a possible deployment for organizations. With a hybrid cloud, organizations might run non-core applications in a public cloud, while maintaining core applications and sensitive data in-house in a private cloud. (see Figure 2-9)

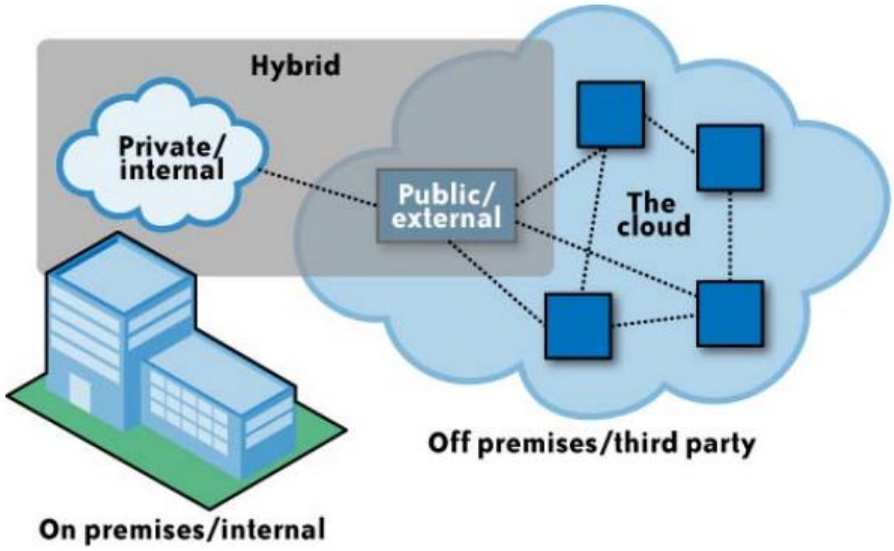

FIGURE 2-9. Hybrid cloud

1.5 Key Drivers to Adopting the Cloud

Table 2-3 illustrates some of the benefits cloud computing offers: lower IT costs, faster time to go live and reduced complexity. However, with cloud computing it is critical to understand how to integrate the cloud solution into existing enterprise architecture.

TABLE 2-3. Cloud computing: A customer's perspective

Dedicated/traditional IT	Cloud computing
High upfront IT investments for new builds	Low upfront IT investments; pay-for-use model
High cost of reliable infrastructure	Reliability built into the cloud architecture
High complexity of IT environment	Modular IT architecture environments
Complex infrastructure	No infrastructure

The following subsections describe a number of compelling reasons to move operations toward cloud computing.

Small Initial Investment and Low Ongoing Costs

Public cloud computing can avoid capital expenditures because no hardware, software, or network devices need to be purchased. Cloud usage is billed on actual use only, and is therefore treated more as an expense.

Economies of Scale

Most development projects have a sizing phase during which one attempts to calculate the storage, processing power, and memory requirements during development, testing, and production. It is often difficult to make accurate estimates; under- or overestimating these calculations is typical. With the flexibility that cloud computing solutions offer, companies can acquire computing and development services as needed and on demand, which means development projects are less at risk of missing deadlines and dealing with the unknown.

Open Standards

Some capabilities in cloud computing are based on open standards for building a modular architecture that can grow rapidly and can change when required. The flexibility to alter the source code is essential to allow for continued growth in the cloud solution. Open source software is the foundation of the cloud solution and is critical to its continued growth.

Sustainability

CSPs have invested considerable expense and thought into creating a resilient architecture that can provide a highly stable environment. Cloud computing allows companies to rely on the CSP to have limited points of failure, better resilience via clustering, and the ability to invest in state-of-the-art resilience solutions.

1.6 Barriers to Cloud Computing Adoption in the Enterprise

Although there are many benefits to adopting cloud computing, there are also some significant barriers to adoption. Two of the most significant barriers to adoption are security and privacy, and we discuss them extensively in the following lectures. The other barriers, besides security and privacy, are significant, but are outside the scope of this course.

Security

Because cloud computing represents a new computing model, there is a great deal of uncertainty about how security at all levels (e.g., network, host, application, and data levels) can be achieved. That uncertainty has consistently led information executives to state that security is their number one concern with cloud computing. The subsequent lectures present a detailed examination of those concerns to determine whether they are grounded.

Privacy

The ability of cloud computing to adequately address privacy regulations has been called into question. Organizations today face numerous different requirements attempting to protect the privacy of individuals' information, and it is not clear (i.e., not yet established) whether the cloud computing model provides adequate protection of such information, or whether organizations will be found in violation of regulations because of this new model. Privacy in cloud will be discussed in details in this course.

Connectivity and Open Access

The full potential of cloud computing depends on the availability of high-speed access to all. Such connectivity, rather like electricity availability, globally opens the possibility for industry and a new range of consumer products.

Reliability

Enterprise applications are now so critical that they must be reliable and available to support 24/7 operations. In the event of failure or outages, contingency plans must take effect smoothly, and for disastrous or catastrophic failure, recovery plans must begin with minimum disruption. Each aspect of reliability should be carefully considered when engaging with a CSP, negotiated as part of the SLA.

Interoperability

The interoperability and portability of information between private clouds and public clouds are critical enablers for broad adoption of cloud computing by the enterprise. Many companies have made considerable progress toward standardizing their processes, data, and systems through implementation of ERPs. Depending on the application, it is critical to integrate with traditional applications that may be resident in a separate cloud or on traditional technology.

Independence from CSPs

Examples exist of IT outsourcing contracts that have effectively locked a customer into a service that does not meet current or evolving needs at a speed and cost that are acceptable to meet business goals. This could be caused by a number of factors, and is a concern if limited options exist for quickly engaging an alternative provider supplier to meet the needs without large transition or penalty costs.

Political Issues Due to Global Boundaries

In the cloud computing world, there is variability in terms of where the physical data resides, where processing takes place, and from where the data is accessed. Given this variability, different privacy rules and regulations may apply. Because of these varying rules and regulations,

by definition politics becomes an element in the adoption of cloud computing.