

# BSE-3502

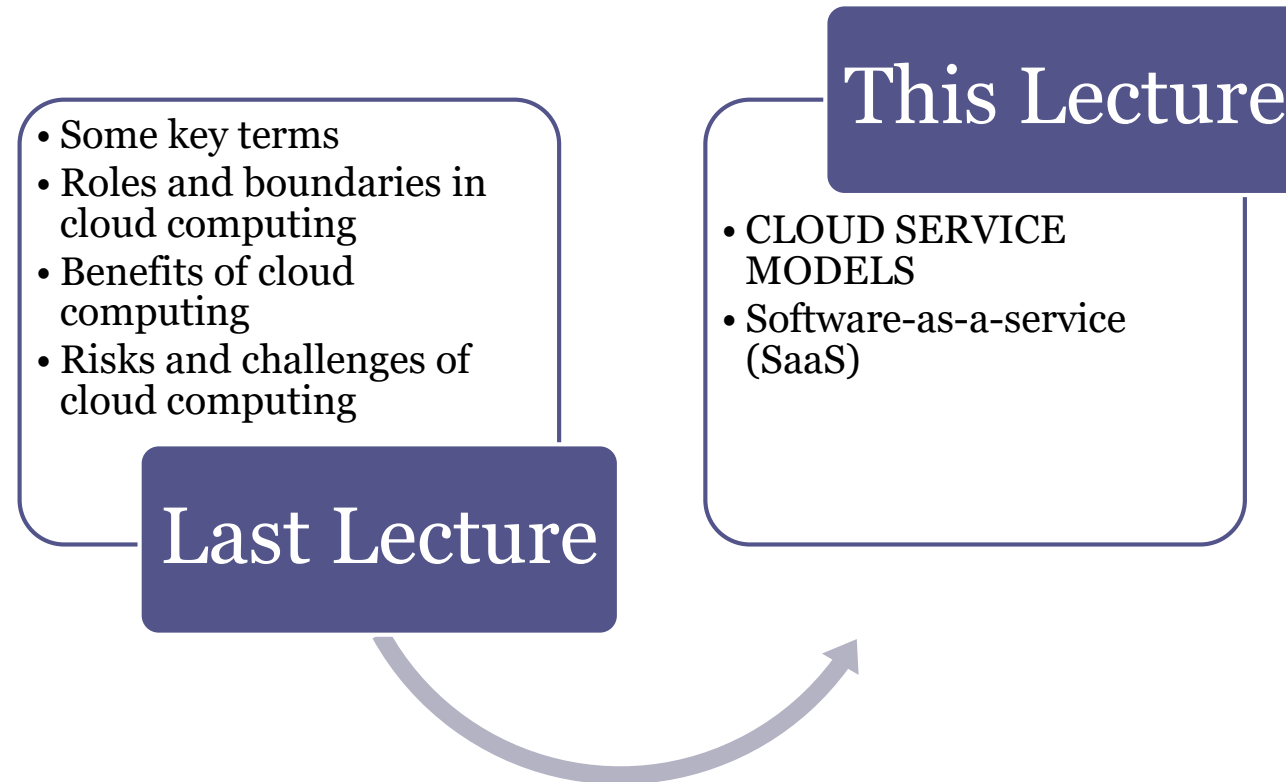
## Cloud Computing

### Lecture 04

Department of Software Engineering  
Mirpur University of Science and Technology (MUST)

# Lecture 04

## ROLES AND BOUNDARIES IN CLOUD COMPUTING



# CLOUD SERVICE MODELS

## IaaS, PaaS & SaaS Provisioning

- **IaaS:** The IT-resources are typically virtualized and packaged in a preplanned way.
- **PaaS:** Delivers a programming environment containing preconfigured tools to support the development lifecycle of custom applications.
- **SaaS:** Is the software hosted over cloud infrastructure and offered as a utility services.



# Software-as-a-Service (SaaS)

**SaaS: Is the software hosted over cloud infrastructure and offered as a utility services.**

- SaaS is software that runs on the cloud and is used through the internet.
- It works like a utility service (similar to electricity) that many people can use at the same time.
- SaaS is built and operated on platforms like IaaS or PaaS, and the creator becomes the service provider.
- Users have very limited control and can only use the features provided to them.



# Software-as-a-Service (SaaS)

**NIST (National Institute of Standards and Technology)** defines SaaS as:

**“Software deployed as a hosted service and accessed over the Internet.”**

- SaaS is a cloud-based software where both the **program (code)** and **data** run and stay on cloud servers.
- Users access SaaS through a **web browser** (Chrome, Edge, Firefox).
- A **cloud service consumer** is any program that temporarily uses a cloud service.

For now, we can say the **browser acts as the cloud service consumer** when accessing SaaS.

**•Common Examples of SaaS:**

- Google Docs
- Gmail
- Microsoft 365 Online
- Zoom
- Canva



# Software as a Service (SaaS) Overview

- SaaS removes the need for **on-premises applications**, meaning you don't have to keep software
  - or servers in your own data center.
- You don't need to handle **application management** or **data storage** — the SaaS provider does it all.
- Customers can use a **pay-as-you-go** rental model, paying only for what they use.
- SaaS provides **scalability**, so it can easily grow as your needs grow.
- SaaS can be accessed from **any device** (laptop, mobile, tablet) with an internet connection.
- The SaaS provider ensures the software is **well-tested, reliable, and fully supported**.



# Software as a Service (SaaS) Overview

- The notable **disadvantage** of SaaS is that the **data resides off-premises**. Therefore, the **data security** is of prime importance because the customers' data may be proprietary and business-sensitive.
- The SaaS provider offers SaaS apps executing over IT-resources. These resources can be from a physical servers or a VM owned/rented by the provider.
- Each instance of a SaaS app (consumed by a user) is allocated separate set of IT-resources.



# Software as a Service (SaaS) Overview

- Classes of SaaS:
  - **Business logic:** Connect the suppliers, employees, investors and customers.
    - Example: Invoicing, fund transfer, inventory management, customer relationship management (CRM)
  - **Collaboration:** Support teams of people work together.
    - Examples: Calendar systems, email, screen sharing, conference management and online gaming.
  - **Office productivity:** Office environment support.
    - Examples: word processors, spreadsheets, presentation and database software.
  - **Software tools:** For the support of developing software and solving compatibility problems.
    - Examples: format conversion tools, security scanning, compliance checking and Web development.





# Software as a Service (SaaS) Overview

- Software that are not suitable for public SaaS offerings (according to NIST):
  - **Real-time software:** They require precise response time. Due to variable response time and network delays, these software are not suitable to be offered as SaaS. Such as flight control systems and factory robots etc.
  - **Bulk-consumer data:** When extremely large amount of data is originating physically at the consumer's side such as physical monitoring and patient monitoring data. It is not feasible to transfer this data in real time over WAN to SaaS provider.
  - **Critical software:** A software is labeled *critical* if its failure or delay in handling can cause loss of life or loss of property. These software are not suitable for SaaS because achieving a continuous acceptable reliability for critical software in public SaaS is quite challenging due to (unreliable) public network based access.



# Software as a Service (SaaS) Overview

- SaaS billing: Based on
  - Number of users
  - Time in use
  - Per-execution, per-record-processed
  - Network bandwidth consumed
  - Quantity/duration of data stored



# SaaS Examples

- Salesforce.com SaaS for Customer Relationship Management (CRM)
  - Manage sales contacts and leads.
  - Centralize the contact. information and project details.
  - The sales reports from any place any time.
  - Manages and syncs sales contacts and meetings with other tools such as Microsoft Outlook.



# SaaS Examples

- Taleo SaaS for Human Resources Management (HRM):
  - Recruitment tools to manage the applicants' data for hiring purposes.
  - Performance management and tracking tools for employees' evaluation.
  - Compensation tools for rewarding the employees according to performance.
  - Workforce training and professional development tools



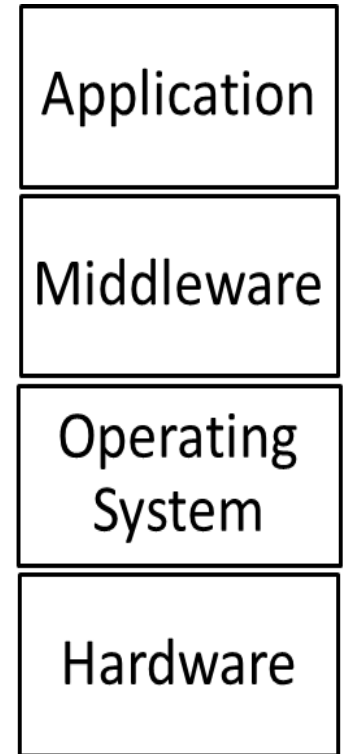
# SaaS Examples

- ADP SaaS for Payroll Processing and HRM:
  - Cloud solution for time management, employees benefits calculation, worker compensation and HR issues.
- Carbonite SaaS for File Backups:
  - Provides backup services for precious business data and personal data. The data is kept securely and redundantly.
- Microsoft Office 365 SaaS for Document Creation, Editing and Sharing:
  - In order to provide the documentation tools at affordable price and to compete with the freeware solutions, Microsoft offers its flagship software suite on monthly rental basis.



# SaaS Software Stack

- The provider controls most of the software stack.
- **Application:** Email
- **Middleware:** software libraries, run time environments (Java, Python)
- Service **provider** has admin control over application and total control over the rest of the layers.
- Service **consumer** has limited admin control over the application and no control over the rest of the stack.
- A consumer can create, send and manage the emails and even the email accounts. But the email provider has absolute control over the SaaS software stack in order to perform its duties such as provisioning, management, updates and billing in email app.



SaaS Software Stack



# SaaS Benefits

- **Modest software tool footprint:** There is no need for complex installation procedures because the SaaS applications are accessible through web browsers. This is one of the reasons of widespread use of SaaS applications.
- **Efficient use of software licenses:** The license issuance and management procedure is quite efficient. A single client is issued a single license for multiple computers. This is because the software is running directly on provider's infrastructure and thus can be billed and monitored directly.
- **Centralized management and data:** The consumer's data is stored in cloud. The provider assures the security and availability of data. The data seems centralized for the consumer may in fact be distributed and replicated by the provider. Data backup is provided at possibly additional charges.



# SaaS Benefits

- **Platform responsibilities managed by providers:** Consumer does not has to bother about operating system type, hardware and software configurations, software installation and upgrades.
- **Savings in up-front costs:** (as discussed before) the up-front costs such as equipment acquisition and hardware provisioning etc. are avoided by SaaS consumer. The provider is responsible for operational issues such as backups, system maintenance, security software, upgrades, trouble shooting in software, physical security and hardware management etc.





# Issues and Concerns

- The NIST has identified few issues and concerns about SaaS. Most of these issues are due to network dependency of SaaS.
  1. **Browser based risks and remedies:** Since the SaaS is accessed through browser installed on consumers' device, the inherent vulnerabilities of the web browsers do have impact over SaaS security.
    - Although the browsers apply encryption upon network traffic, yet various network attacks such as brute force and man in the middle attacks are possible upon the SaaS data.
    - The resources leased by a consumer can be hijacked by malicious users due to poor implementation of cryptographic features of browsers.



# Issues and Concerns

- If the consumer's browser is already infected with a security threat (due to a visit to malicious website) then later, the same browser is used for SaaS access, then the SaaS data might get compromised.
- If a single consumer accesses multiple SaaS services using browser instances, then the data of these SaaS instances may get mixed up.
- A few suggestions by NIST:
  - Use different browsers to access each different SaaS.
  - Do not use the same web browser for web surfing and SaaS access.
  - Use a VM to access the SaaS.



# Issues and Concerns

2. **Network dependence:** SaaS application depends upon reliable and continuously available network.
  - The reliability of a public network (Internet) can not be guaranteed as compared to dedicated and protected communication links of private SaaS applications.
3. **Lack of portability between SaaS clouds:** It may not be trivial to import export data among different SaaS applications deployed over different clouds due to customized development and deployment of SaaS applications and data formats.
4. **Isolation vs. Efficiency (Security vs. Cost Tradeoffs):** The SaaS provider has to make a trade-off decision as to deploy separate IT-resources (such as VMs) for each client or concurrently server multiple clients through a single deployment of SaaS application.



# NIST Recommendations for SaaS

- **Data protection:** The consumer should analyze the data protection, configuration, database transaction processing technologies of SaaS provider. Compare them with the confidentiality, integrity, availability and compliance requirement of the consumer.
- **Client device/application protection:** The consumer's client device (browser running over a computer) should be protected to control the exposure to attacks.
- **Encryption:** Strong encryption algorithm with key of required strength should be used for each web session as well as for data.
- **Secure data deletion:** The data deletion through consumer's request should be reliably done.

