CSC7970 – NEXT-GENERATION NETWORKING

CLOUD COMPUTING

Instructor: Susmit Shannigrahi
sshannigrahi@tntech.edu

Today: What is cloud computing?

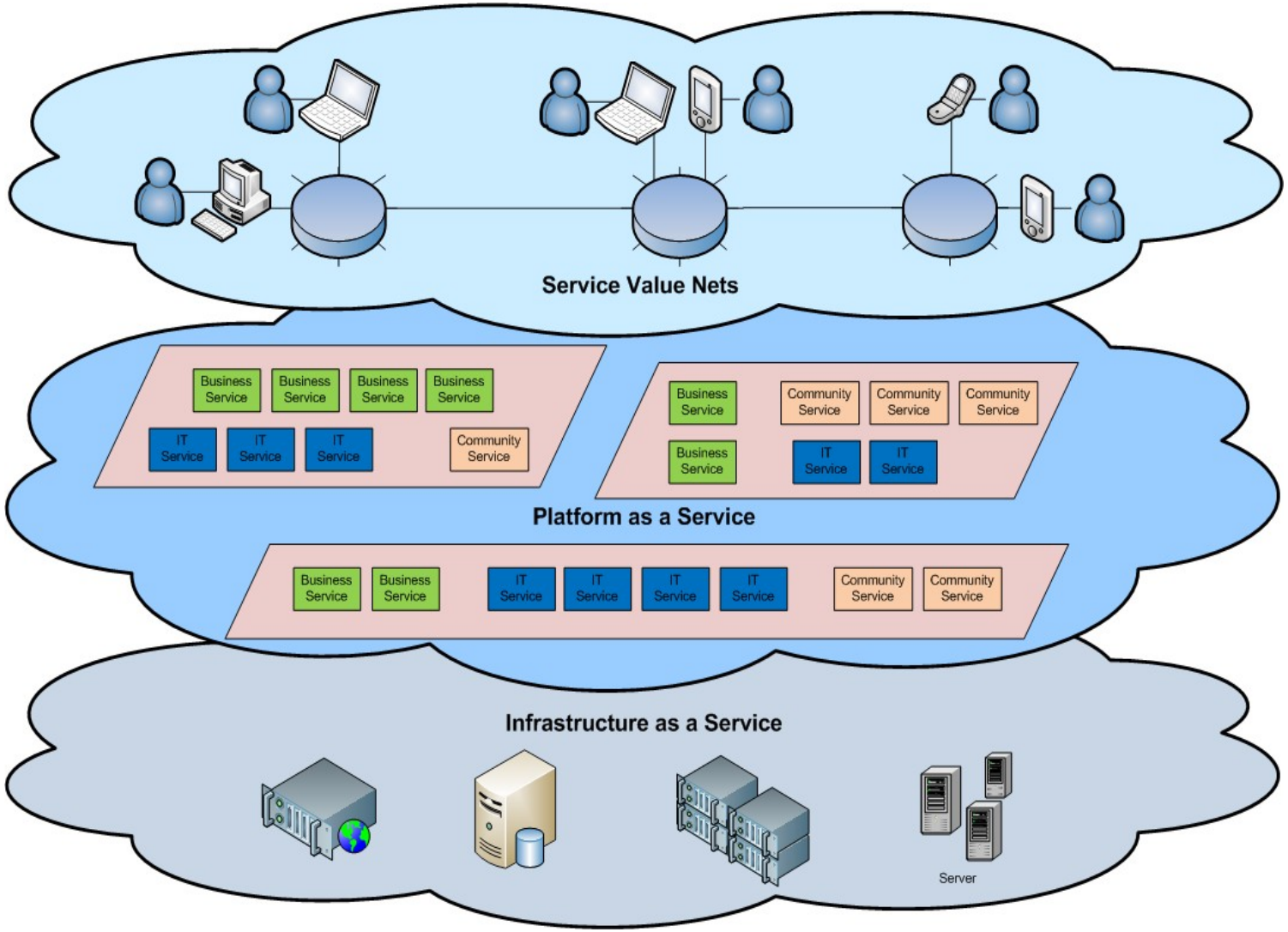
- What is cloud computing?
  - Idea goes back to the 60s
  - More recently popularized by Amazon EC2 in 2006

- Computing resources located somewhere on the Internet
  - Why would that be useful?
    - Users do not have to buy and maintain their own servers
    - Users can scale up or down resources based on need
    - Users pay based on their usage
    - Resources are **(almost)** always on and available
    - Hide complexity of underlying infrastructure from users

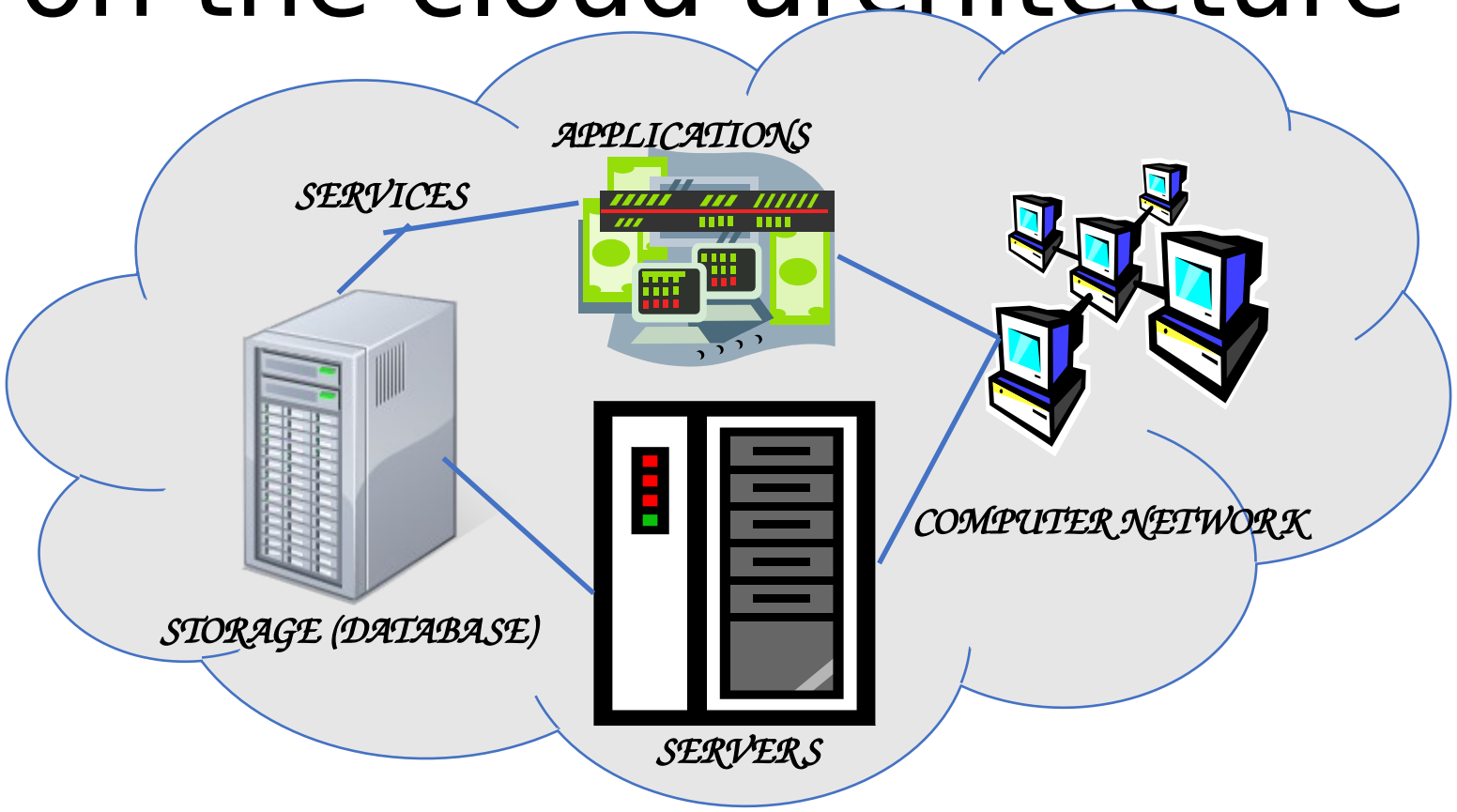
Cloud computing summary

- Cloud computing is a general term used to refer to services and infrastructure offered on the Internet
  - Services are remotely hosted
  - **Pervasive:** services are available (almost) all the time from anywhere
  - **Commodified:** goal is to offer a "utility" computing model similar to that of traditional utilities, such as gas and electricity – users pay for what they want and use!
- Basic idea
  - Hard to create and maintain your own infrastructure
  - Pay someone to do it for you
- Clouds galore
  - Google, Microsoft, Amazon, Cloudflare…..

Cloud Architecture



More on the cloud architecture



- Shared pool of configurable computing resources
- On-demand network access
- Provisioned by the Service Provider

Cloud design characteristics

- Massive scale
  - Users all around the world may need to allocate massive amounts of computing resources
- Geographic distribution
  - Low delay, make sure resources are “close" to users
- Resilience
  - Resources are still available when servers in a data-center or multiple data-centers fail
- Virtualization
  - Users do not know where exactly the resources are located or how many other customers use the resources
- Service-oriented
  - Offer seamless integration with a variety of services (e.g., load balancing, rest APIs, etc.)

Cloud Service Models

Software as a Service (SaaS) | Platform as a Service (PaaS) | Infrastructure as a Service (IaaS)

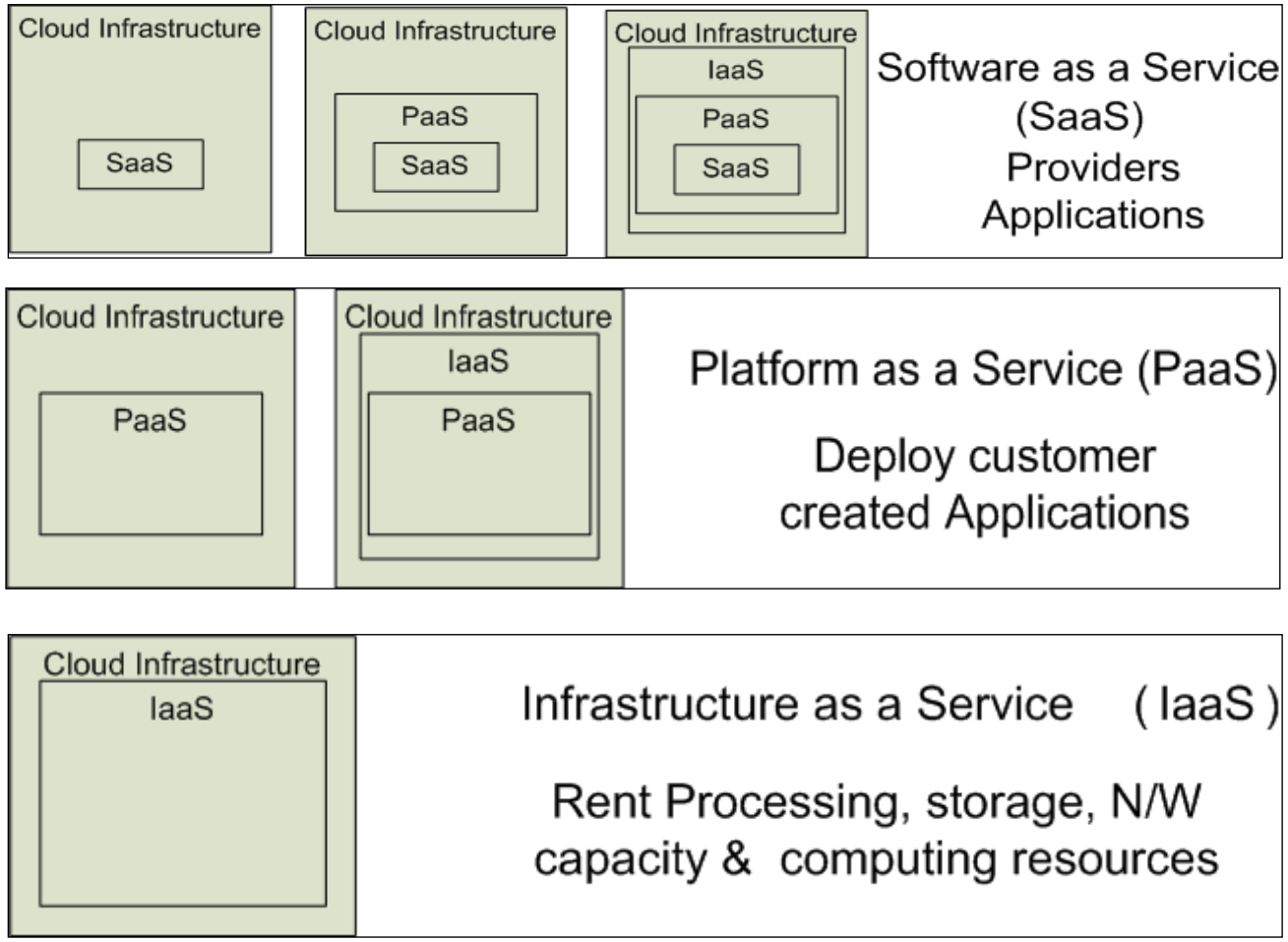
SalesForce CRM

Google Apps

Microsoft Azure

amazon web services

rackspace HOSTING



Why Amazon in-between PaaS and IaaS?

Software as a Service (SaaS)

- Users do NOT need to manage any software or hardware
  - Software is managed by provider
  - Software is typically accessed by users through a browser
  - Software runs on cloud servers of the software provider or a third party cloud

- Software updates, outages, scaling are transparent to users
  - Software provider or cloud provider has to figure it out
  - Depends on who manages the cloud infrastructure

- For this “convenience”, users pay a fee
  - Different billing models
  - Examples: based on usage, software features needed, number of user accounts, etc.

Platform as a Service (PaaS)

- Platforms offered on top of (potentially) virtualized infrastructure
  - Examples: storage, database
  - Platforms are not fun to manage!

- Users do NOT need to manage the infrastructure or the platform
  - They pick the platform they want and use it on a cloud
  - Platform runs on cloud servers of the platform provider or a third party

- Users do not need to bother about updates, outages, scalability
  - This comes with a fee as before

Infrastructure as a Service (IaaS)

- Users do not need to manage the infrastructure
  - They select infrastructure that matches their requirements (e.g., CPU power, memory, hard disk, etc.)
  - They select the OS of their preference
  - Cloud provider makes the required infrastructure available for users
  - Users do not need to maintain/care about this infrastructure

- The notion of infrastructure extends to networking components too!
  - Load-balancing, routers, firewalls, etc.
  - All that comes again for a fee

Virtualization

- What is it?
- Users feel like having complete control/being the only entity using the cloud resources
  - True or not?
- In reality: multiple users might occupy at the same time the same physical resources
  - Why we need to do that?

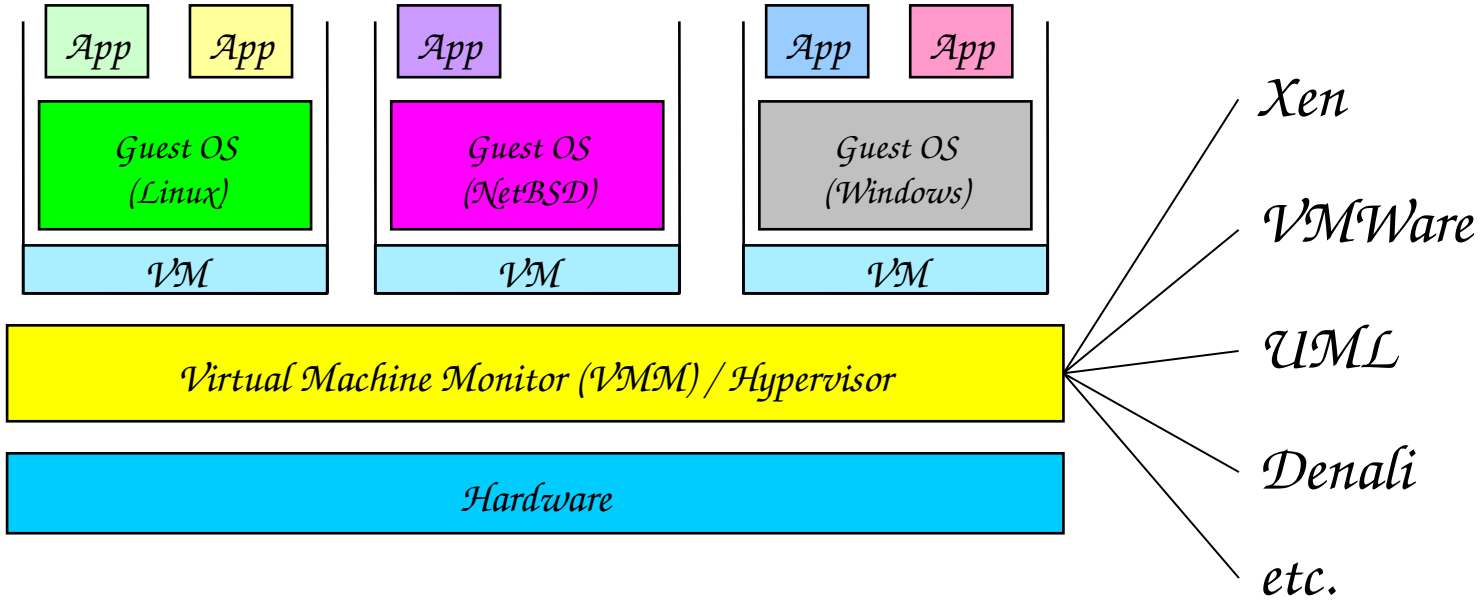
Virtualization (cont'd)

- Rarely all the provisioned customers will use the resources at the same time
  - Easier scalability, lower cost, more money for cloud provider!

- "Overprovisioning": Cloud resources "overbooking"
  - These rare times that indeed all the provisioned customers will use the resources simultaneously!
  - What can go wrong?
  - Things go ***really slow*** --> Frustration :-(
  - Pay more!

Virtual Machines

- How to achieve virtualization:
  - Run multiple virtual “workspaces”, one per provisioned user
  - Any problems/challenges having multiple customers use the same physical hardware?



Multi-tenancy

- Tenant (in real life): you rent an apartment, you occupy the apartment for a certain time-period as long as you pay the rent

- Same in cloud computing: users (tenants) “occupy” resources for a certain period of time
  - ***Multi-tenancy:*** multiple tenants occupy and can potentially use the same physical resources at the same time!
  - Bad or good? Why?

Tenant isolation

- Analogy: multiple tenants in the same apartment
  - Tenants still need some privacy
  - Tenants occupy different bedrooms (isolation)

- In cloud computing tenants may have their personal data or other confidential info on the cloud
  - Cloud providers need to make sure that a tenant (malicious or not) does not have access to the data of other tenants
  - Virtual machines and hypervisors are ways of achieving isolation

Cloud Transparency

- Users should not have to do anything “special” when it comes to using the cloud
  - (Almost) same set of actions like running something locally
- In this sense: cloud should be as transparent to users as possible
  - No (minimal) overhead for customers
- Internally, the cloud can be built in whatever way the provider wants
  - Typically: clusters of servers and off-the-shelf components + open-source and proprietary software

Cloud deployment models

- Private and public clouds
  - Why?
  - Some organizations do not trust storing their data/intellectual property/software on third-party cloud
  - **This might not be the case about the data of their own users though!**
- Private cloud: Access only to a single organization
  - Typically maintained by the organization itself
- Public cloud: Cloud infrastructure available to the general public
  - Typically owned by an organization that sells cloud services

Data replication

- User data needs to be available in the face of server/network failures
  - Single copy of user data will not work
  - Cloud server fails, user data gets lost

- Solution: replicate user data across multiple cloud servers
  - Unlikely all the involved cloud servers to fail at the same time
  - Servers within the same data center or in different data centers?

Data replication challenges

- Challenge: selecting the servers for replication
  - Where the servers are located?
  - How many times data is replicated?
  - How many copies do you need?
  - What about data consistency?

- Challenge: how to synchronize all these servers when user data is updated
  - All-time classic problem in distributed systems
  - Paxos and other techniques

Cloud security

- 2 parts
  - Network/communications security
  - Security within an individual server

- Network/communications security
  - Securing communication between user and cloud resources
  - Securing communication (e.g., for data replication purposes) among servers
  - Secure communication channels through TLS/SSL

Cloud server security

- **It is so hard to secure a server!**
  - **Why?**

- Vulnerabilities can be anywhere!
  - OS, applications, browsers, programming languages, malicious tenants
  - Impossible to keep up with all the different things running on a server
  - We might never realize that a server is compromised

- **Trusted execution:** can we provide to cloud users security guarantees in terms of how safe their data/code might be?
  - Since it is hard to keep up with all the things running on a server
  - Even if the server is compromised but the provider does not know!

Trusted execution environments

- How to build trusted execution environments?
  - Techniques in software and/or hardware
- Software
  - Dual operating system images
  - Sand-boxes: protected OS space for the execution of unknown programs
- Hardware
  - Trusted external hardware modules (e.g., for encryption/decryption, mathematical operations)
  - Trusted execution mode protected by CPU directly (e.g., Intel SGX, ARM Trustzone)

The not so “bright” side of cloud

- Use-case:
  - Application provider uses PaaS (third-party platform) running on top of the cloud infrastructure of another third-party

- Data breach happens. Who is responsible for it?
  - What if you have HIPAA data?

The not so “bright” side of cloud

- Data lock-in
  - Easier to get data in, much harder to move out
- Cost
  - Can exponentially increase with data volume
- Interoperability
  - Providers don’t play nicely with each other

Economics: Cloud Service Level Agreements

- Users sign with cloud provider a Service Level Agreement (SLA)
  - Typical case when the cloud user is a corporation

- SLA = understanding of:
  - What the user wants (e.g., hardware specification, software accounts)
  - What the provider is responsible for (e.g., maximum allowed annual downtime, what happens in a case of a data breach)
  - Billing model/policies

Summary

- Cloud:
  - Convenient
  - Hassle-free (well.. for users..)
  - Pay-as-you-go models available
  - (almost) Always available
  - Globally scalable

- Problems:
  - Security
  - Privacy
  - Downtime (cloud does fail!)
  - Just another dependency we cannot live without (?)
  - Who has control over user data/communication?