

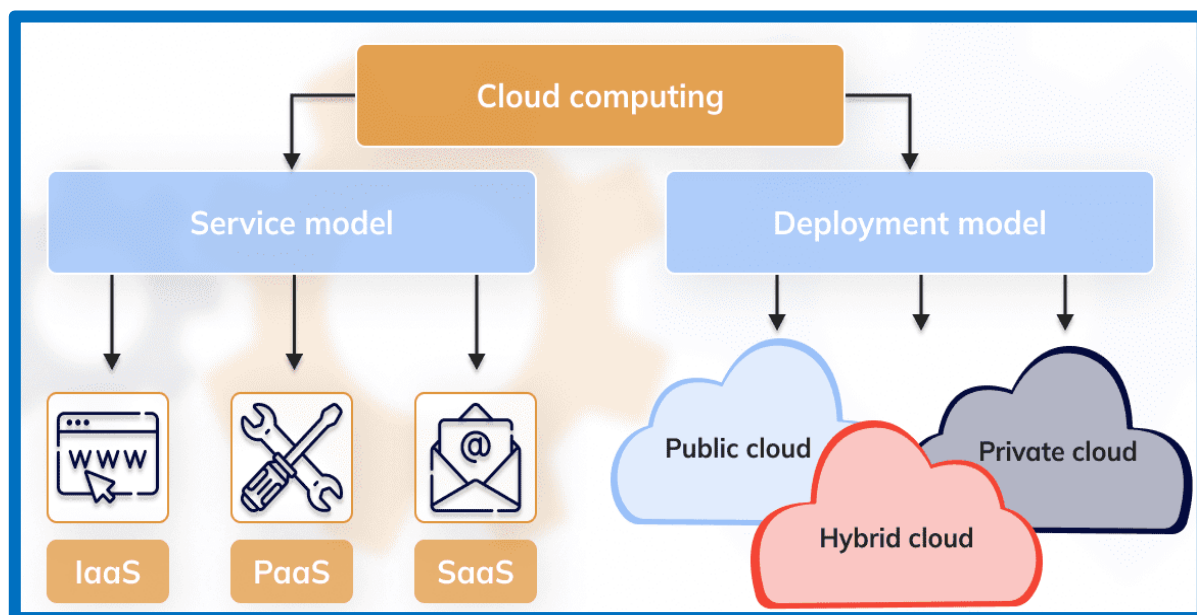


THE NEW COLLEGE



(An Autonomous Institution Affiliated to the University of Madras & Accredited by NAAC with 'A++' Grade of 3.61/4 in the 4th Cycle)
Chennai – 600 014, Tamil Nadu, India.

DEPARTMENT OF COMPUTER APPLICATIONS



E-CONTENT: UNIT V NOTES

Subject : CLOUD COMPUTING

Class : III BCA

BATCH : 2021-2024

Dr. J. Abdul Rasheedh M.C.A., M.Phil., Ph.D.

Assistant Professor,

P.G. Department of Computer Science,

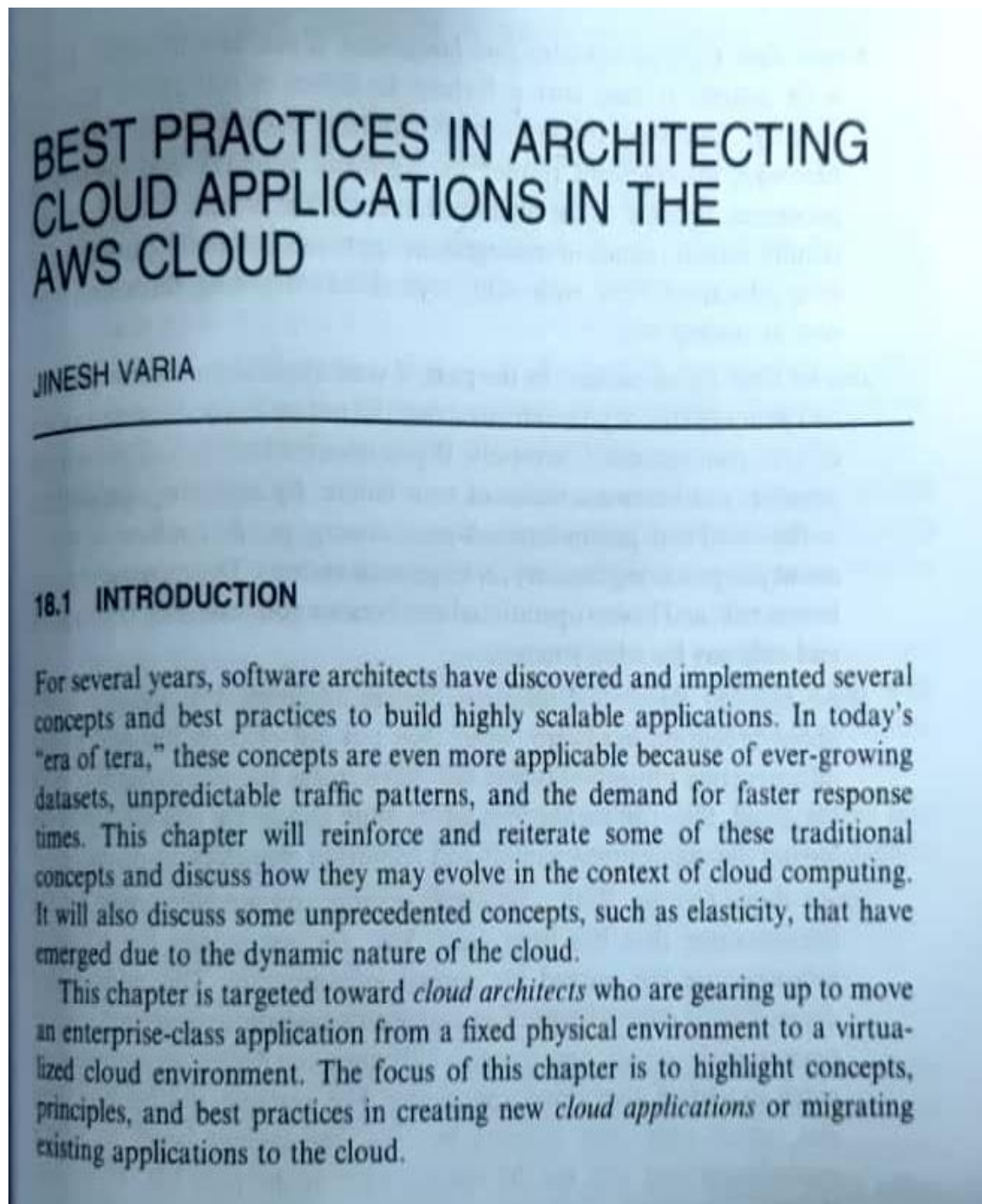
The New College, Chennai-14.

UNIT 5 –CLOUD COMPUTING

UNIT-5:

Applications and Case Studies: Best Practices in Architecting Cloud Applications in the AWS Cloud-Data Security in the Cloud-Legal Issues in Cloud Computing.

-
Application and case studies:



18.2.1 Business Benefits of Cloud Computing

There are some clear business benefits to building applications in the cloud. A few of these are listed here:

Almost Zero Upfront Infrastructure Investment. If you have to build a large-scale system, it may cost a fortune to invest in real estate, physical security, hardware (racks, servers, routers, backup power supplies), hardware management (power management, cooling), and operations personnel. Because of the high upfront costs, the project would typically require several rounds of management approvals before the project could even get started. Now, with utility-style cloud computing, there is no fixed cost or startup cost.

Just-in-Time Infrastructure. In the past, if your application became popular and your systems or your infrastructure did not scale, you became a victim of your own success. Conversely, if you invested heavily and did not get popular, you became a victim of your failure. By deploying applications in-the-cloud with just-in-time self-provisioning, you do not have to worry about pre-procuring capacity for large-scale systems. This increases agility, lowers risk, and lowers operational cost because you scale only as you grow and only pay for what you use.

More Efficient Resource Utilization. System administrators usually worry about procuring hardware (when they run out of capacity) and higher infrastructure utilization (when they have excess and idle capacity). With the cloud, they can manage resources more effectively and efficiently by having the applications request and relinquish resources on-demand.

Usage-Based Costing. With utility-style pricing, you are billed only for the infrastructure that has been used. You are not paying for allocated infrastructure but instead for unused infrastructure. This adds a new dimension to cost savings. You can see immediate cost savings (sometimes as early as your next month's bill) when you deploy an optimization patch to update your cloud application. For example, if a caching layer can reduce your data requests by 70%, the savings begin to accrue immediately and you see the reward right in the next bill. Moreover, if you are building platforms on the top of the cloud, you can pass on the same flexible, variable usage-based cost structure to your own customers.

Reduced Time to Market. Parallelization is one of the great ways to speed up processing. If one compute-intensive or data-intensive job that can be run in parallel takes 500 hours to process on one machine, with cloud architectures [1], it would be possible to spawn and launch 500 instances and process the same job in 1 hour. Having available an elastic infrastructure provides the application with the ability to exploit parallelization in a cost-effective manner reducing time to market.

18.2.2 Technical Benefits of Cloud Computing

Some of the technical benefits of cloud computing includes:

Automation—"Scriptable Infrastructure": You can create repeatable build and deployment systems by leveraging programmable (API-driven) infrastructure.

Auto-scaling: You can scale your applications up and down to match your unexpected demand without any human intervention. Auto-scaling encourages automation and drives more efficiency.

Proactive Scaling: Scale your application up and down to meet your anticipated demand with proper planning understanding of your traffic patterns so that you keep your costs low while scaling.

More Efficient Development Life Cycle: Production systems may be easily cloned for use as development and test environments. Staging environments may be easily promoted to production.

Improved Testability: Never run out of hardware for testing. Inject and automate testing at every stage during the development process. You can spawn up an "instant test lab" with preconfigured environments only for the duration of testing phase.

Disaster Recovery and Business Continuity: The cloud provides a lower cost option for maintaining a fleet of DR servers and data storage. With the cloud, you can take advantage of geo-distribution and replicate the environment in other location within minutes.

"Overflow" the Traffic to the Cloud: With a few clicks and effective load balancing tactics, you can create a complete overflow-proof application by routing excess traffic to the cloud.

18.4 CLOUD BEST PRACTICES

In this section, you will learn about best practices that will help you build an application in the cloud.

18.4.1 Design for Failure and Nothing Will Fail

Rule of Thumb: Be a pessimist when designing architectures in the cloud; assume things will fail. In other words, always design, implement, and deploy for automated recovery from failure.

In particular, assume that your hardware *will* fail. Assume that outages *will* occur. Assume that some disaster *will* strike your application. Assume that you *will* be slammed with more than the expected number of requests per second some day. Assume that with time your application software will fail too. By being a pessimist, you end up thinking about recovery strategies during design time, which helps in designing an overall system better.

If you realize that things fail over time and incorporate that thinking into your architecture, as well as build mechanisms to handle that failure before disaster strikes to deal with a scalable infrastructure, you will end up creating a fault-tolerant architecture that is optimized for the cloud.

Questions that you need to ask: What happens if a node in your system fails? How do you recognize that failure? How do I replace that node? What kind of scenarios do I have to plan for? What are my single points of failure? If a load balancer is sitting in front of an array of application servers, what if that load balancer fails? If there are master and slaves in your architecture, what if the master node fails? How does the failover occur and how is a new slave instantiated and brought into sync with the master?

Just like designing for hardware failure, you have to also design for software failure. Questions that you need to ask: What happens to my application if the dependent services changes its interface? What if downstream service times out or returns an exception? What if the cache keys grow beyond memory limit of an instance?

Build mechanisms to handle that failure. For example, the following strategies can help in event of failure:

1. Have a coherent backup and restore strategy for your data and automate it.
2. Build process threads that resume on reboot.
3. Allow the state of the system to re-sync by reloading messages from queues.
4. Keep preconfigured and preoptimized virtual images to support strategies 2 and 3 on launch/boot.
5. Avoid in-memory sessions or stateful user context; move that to data stores.

Good cloud architectures should be impervious to reboots and re-launches. In GrepTheWeb (discussed in the next section), by using a combination of Amazon SQS and Amazon SimpleDB, the overall controller architecture is very resilient to the types of failures listed in this section. For instance, if the instance on which controller thread was running dies, it can be brought up and resume the previous state as if nothing had happened. This was accomplished by creating a preconfigured Amazon machine image, which, when launched, dequeues all the messages from the Amazon SQS queue and reads their states from an Amazon SimpleDB domain on reboot.

Designing with an assumption that underlying hardware will fail will prepare you for the future when it actually fails.

This design principle will help you design operations-friendly applications, as also highlighted in Hamilton's paper [19]. If you can extend this principle to proactively measure and balance load dynamically, you might be able to deal with variance in network and disk performance that exists due to the multi-tenant nature of the cloud.

AWS-Specific Tactics for Implementing This Best Practice

1. **Failover gracefully using Elastic IPs:** Elastic IP is a static IP that is dynamically remappable. You can quickly remap and failover to another set of servers so that your traffic is routed to the new servers. It works great when you want to upgrade from old to new versions or in case of hardware failures.
2. **Utilize multiple availability zones:** Availability zones are conceptually like logical datacenters. By deploying your architecture to multiple availability zones, you can ensure high availability.
3. **Maintain an Amazon Machine Image** so that you can restore and clone environments very easily in a different availability zone; maintain multiple database slaves across availability zones and set up hot replication.

4. Utilize Amazon CloudWatch (or various real-time open source monitoring tools) to get more visibility and take appropriate actions in case of hardware failure or performance degradation. Set up an Auto scaling group to maintain a fixed fleet size so that it replaces unhealthy Amazon EC2 instances by new ones.
5. Utilize Amazon EBS and set up cron jobs so that incremental snapshots are automatically uploaded to Amazon S3 and data are persisted independent of your instances.
6. Utilize Amazon RDS and set the retention period for backups, so that it can perform automated backups.

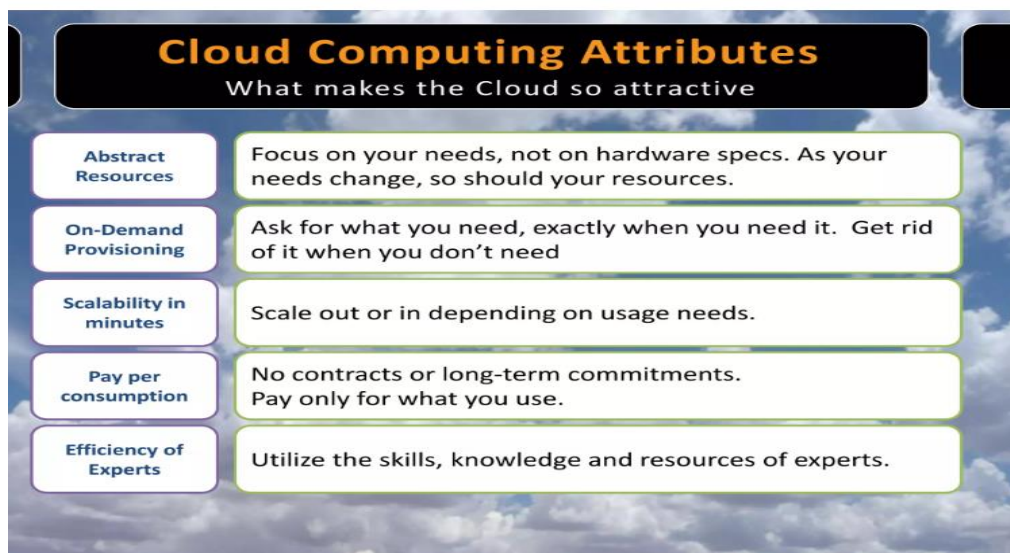
What is an application in cloud computing?

More specifically, a cloud application is **software that runs its processing logic and data storage between 2 different systems: client-side and server-side**. Some processing takes place on an end user's local hardware, such as a desktop or mobile device, and some takes place on a remote server

Best practices in Architecting Cloud Application in the AWS

Which of the following is an architectural best practices recommended by AWS?

Design Principles **implement a strong identity foundation**. Enable traceability. Apply security at all layers. Automate security best practices.

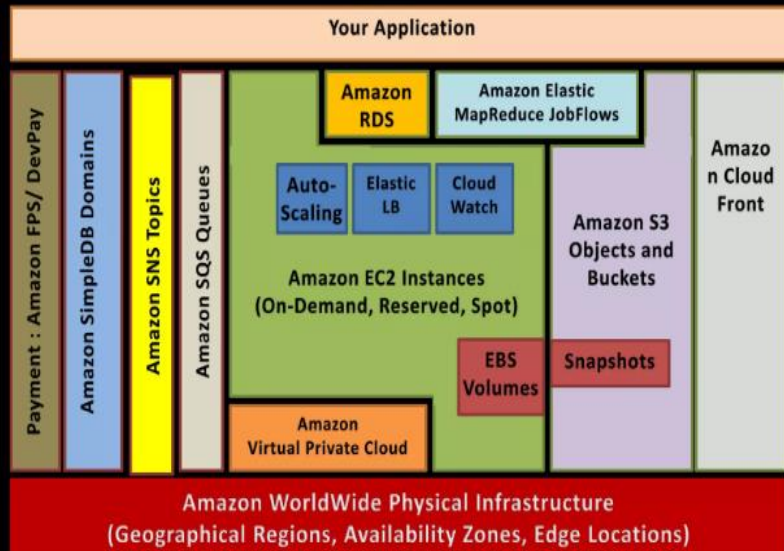


The “Living and Evolving” Cloud

AWS services and basic terminology

Most Applications Need:

1. Compute
2. Storage
3. Messaging
4. Payment
5. Distribution
6. Scale
7. Analytics



Scalability

Build Scalable Architecture on AWS

A scalable architecture is critical to take advantage of a scalable infrastructure

Characteristics of Truly Scalable Service

Increasing resources results in a proportional increase in performance

A scalable service is capable of handling heterogeneity


A scalable service is operationally efficient

A scalable service is resilient

A scalable service becomes more cost effective when it grows


Cloud Architecture Lessons

using Amazon Web Services

- 
1. Design for failure and nothing fails
 2. Loose coupling sets you free
 3. Implement "Elasticity"
 4. Build Security in every layer
 5. Don't fear constraints
 6. Think Parallel
 7. Leverage different storage options

1. Design for Failure

and nothing will really fail




"Everything fails, all the time"
Werner Vogels, CTO Amazon.com

Avoid single points of failure
Assume everything fails, and design backwards
Goal: Applications should continue to function even if the underlying physical hardware fails or is removed or replaced.

Design for Failure with AWS

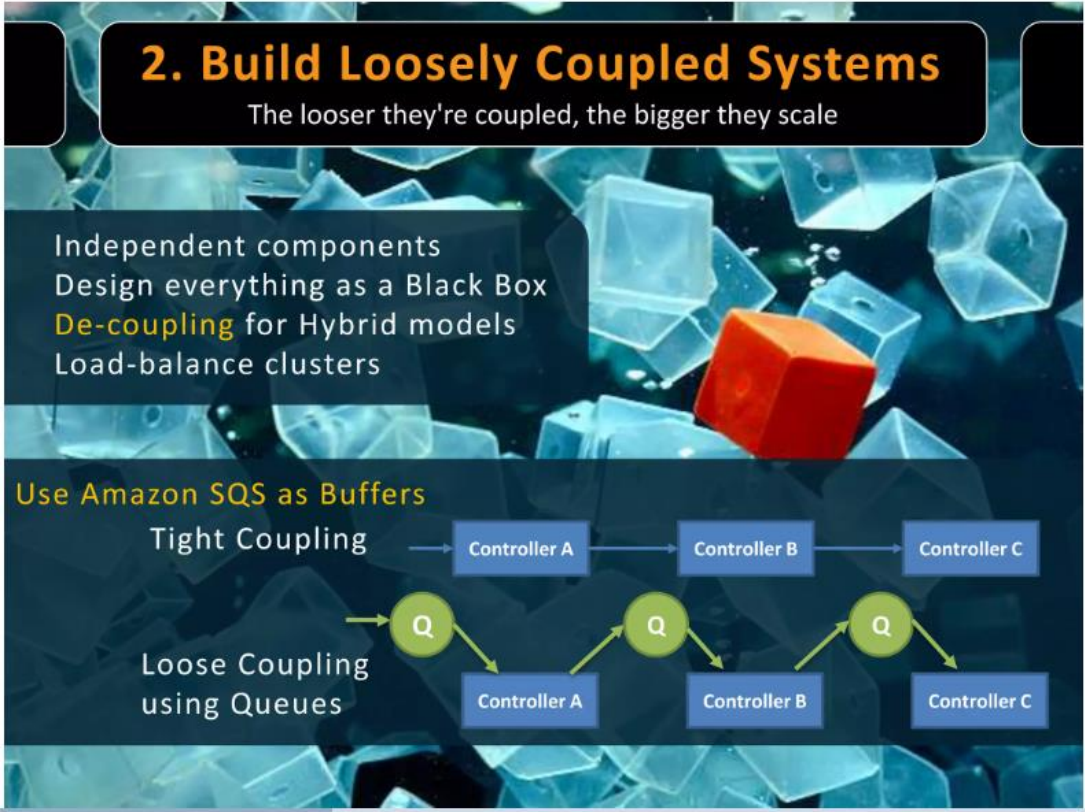
Tools to make your life easier



- Use Elastic IP addresses for consistent and re-mappable routes
- Use multiple Amazon EC2 Availability Zones (AZs)
- Create multiple database slaves across AZs
- Use real-time **monitoring** (Amazon CloudWatch)
- Use Amazon Elastic Block Store (EBS) for persistent file systems

2. Build Loosely Coupled Systems

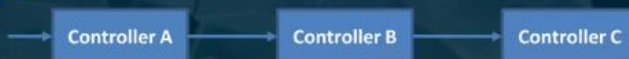
The looser they're coupled, the bigger they scale



- Independent components
- Design everything as a Black Box
- De-coupling** for Hybrid models
- Load-balance clusters

Use Amazon SQS as Buffers

Tight Coupling



Loose Coupling
using Queues



Data Security in the cloud:

What is Cloud Data Security?

Cloud data security refers to the technologies, policies, services and security controls that protect any type of data in the cloud from loss, leakage or misuse through breaches, exfiltration and unauthorized access. A robust cloud data security strategy should include:

Ensuring the security and privacy of data across networks as well as within applications, containers, workloads and other cloud environments

Controlling data access for all users, devices and software

Providing complete visibility into all data on the network

The cloud data protection and security strategy must also protect data of all types. This includes:

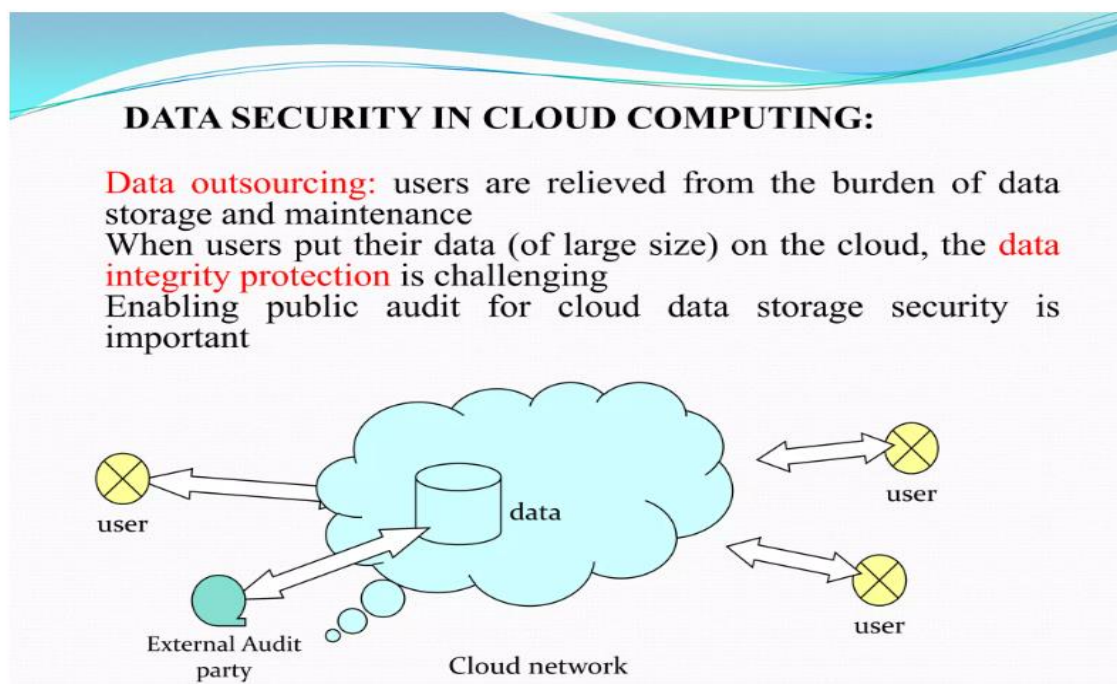
Data in use: Securing data being used by an application or endpoint through user authentication and access control

Data in motion: Ensuring the safe transmission of sensitive, confidential or proprietary data while it moves across the network through encryption and/or other email and messaging security measures

Data at rest: Protecting data that is being stored on any network location, including the cloud, through access restrictions and user authentication

How does data security work in cloud computing?

data security is the combination of technology solutions, policies, and procedures that the enterprise implements to protect cloud-based applications and systems, along with the associated data and user access.



There are three core elements to data security that all organizations should adhere to: **Confidentiality, Integrity, and Availability**.

Why is data security important in cloud computing?

You need a secure way to immediately access your data. Cloud security ensures your data and applications are readily available to authorized users. You'll always have a reliable method to access your cloud applications and information, helping you quickly take action on any potential security issues.

Types of data security

Encryption

Using an algorithm to transform normal text characters into an unreadable format, encryption keys scramble data so that only authorized users can read it. File and database encryption solutions serve as a final line of defense for sensitive volumes by obscuring their contents through encryption or tokenization. Most solutions also include security key management capabilities.

Data Erasure

More secure than standard data wiping, data erasure uses software to completely overwrite data on any storage device. It verifies that the data is unrecoverable.

Data Masking

By masking data, organizations can allow teams to develop applications or train people using real data. It masks personally identifiable information (PII) where necessary so that development can occur in environments that are compliant.

Data Resiliency

Resiliency is determined by how well an organization endures or recovers from any type of failure – from hardware problems to power shortages and other events that affect data availability (PDF, 256 KB). Speed of recovery is critical to minimize impact.

What are the legal issues involved in cloud computing?

Legal issues that can arise “in the cloud” include liability for copyright infringement, data breaches, security violations, privacy and HIPAA violations, data loss, data management, electronic discovery (“e-discovery”), hacking, cybersecurity, and many other complex issues that can lead to complex litigation and ...

What are the legal issues in cloud computing?

Besides this cloud computing creates new cloud-based services for generating employment. Well, several legal issues are associated with cloud computing like **privacy, data security, contact issues, and the issues related to the location of data**.

Cloud Computing: Legal Challenges

- Liability
- Security
- Risk allocation
- Data Retention Issues
- 3rd party contractual limitations
- Regulatory compliances
- Control over physical location of the data
- Security breach
- Trade secret protection
- Hacking of cloud provider
- Financial liability of cloud vendor
- Legal/practical liability for force majeure events
- IPR issues
- Jurisdiction and court of law

5

LEGAL ISSUES IN CLOUD COMPUTING

Introduction:

cloud computing is here to stay, all thanks to its several benefits. When connected to the right cloud infrastructure, you can save costs and enjoy broad network access and rapid elasticity.

While it is easy to be clouded by the benefits of cloud computing, you must also consider some legal issues. Doing so would ensure that you make an informed decision, especially in your choice of Cloud Service Provider (CSP). Plus, you can adequately protect yourself from the adverse effects of these legal issues in cloud computing. Today, I'll be discussing some of the issues you need to look out for.

List of legal issues in cloud computing:

i) Data Protection

Data protection is one of the most critical legal issues you must consider when using the cloud for your operations. It is especially important if your business includes handling the personal data of individuals in any form. There are data protection regulations with strict provisions on how you handle the personal data of individuals.

Under most of these regulations, including the General Data Protection Regulation, which deals with handling data of EU citizens, you can't just export citizens' personal data to the cloud without obtaining the necessary consent. You must also comply with the data protection standards as stipulated by these regulations. Failure to do so would attract strict sanctions.

You need to understand what the law says about data protection in your jurisdictions.

ii) Data Privacy and Security

Another essential legal issue in cloud computing that you should pay attention to is data privacy and security. If a third party receives unauthorized access to private information about your clients, it can damage your company's reputation. Your business risks losing sensitive and corporate confidential information in the case of a security breach. You may also have to compensate your customer for violating their data privacy, which would cost your business a lot.

Make sure you engage a CSP that would offer you the highest privacy and security standard possible. You should also ensure that there are necessary firewalls to prevent a security breach.

iii) Data Ownership (Intellectual Property Rights)

It is safe to assume that you own all the rights to data sent to the cloud by your company. However, it is advisable that your Service Level Agreement (SLA) with the CSP expressly indicates that your company has full rights to the data stored in the cloud and can retrieve it whenever you want. It is also essential to have these provisions in place, especially concerning data generated inside the cloud. The CSP (Communication Service Providers) may want to claim newly generated data because it was generated in the cloud through a data analytics solution.

Let the SLA provide that data generated in and out of the cloud by your company belongs to your company.

iv) Jurisdiction Issues

The issue of differences in laws applicable across different jurisdictions is one of the legal issues in cloud computing. For instance, the government can require CSPs to disclose client data in some jurisdictions. However, in some other jurisdictions, there is express protection for data stored in the cloud, and in those jurisdictions, governments cannot access it without following due process.

You may want your SLA (Service Level Agreement) to contain express provisions that the CSP can only hold your data in specific jurisdictions.

Conclusion

While these legal issues are not exhaustive, they are some of the most important ones you need to consider when you decide to use the cloud for your business operations. As much as you want to use the cloud for its several benefits, don't ignore the legal issues I've highlighted. Keep your business adequately protected at all times.