USN										

CELEBRATING 25 YEARS
CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test III – May 2019
Answer Solution

Sub:	Cloud Computing			Sub Code:	**18SCS23**	Branch:	CSE
Date: 14/05 /19	Duration: 90 min’s	Max Marks:	50	Sem / Sec:	M. Tech (CSE)/ II SEM		OBE

Answer any FIVE FULL Questions

MARKS | CO | RBT

1. Explain security risks with respect to cloud. **[10]** **CO5** **L2**

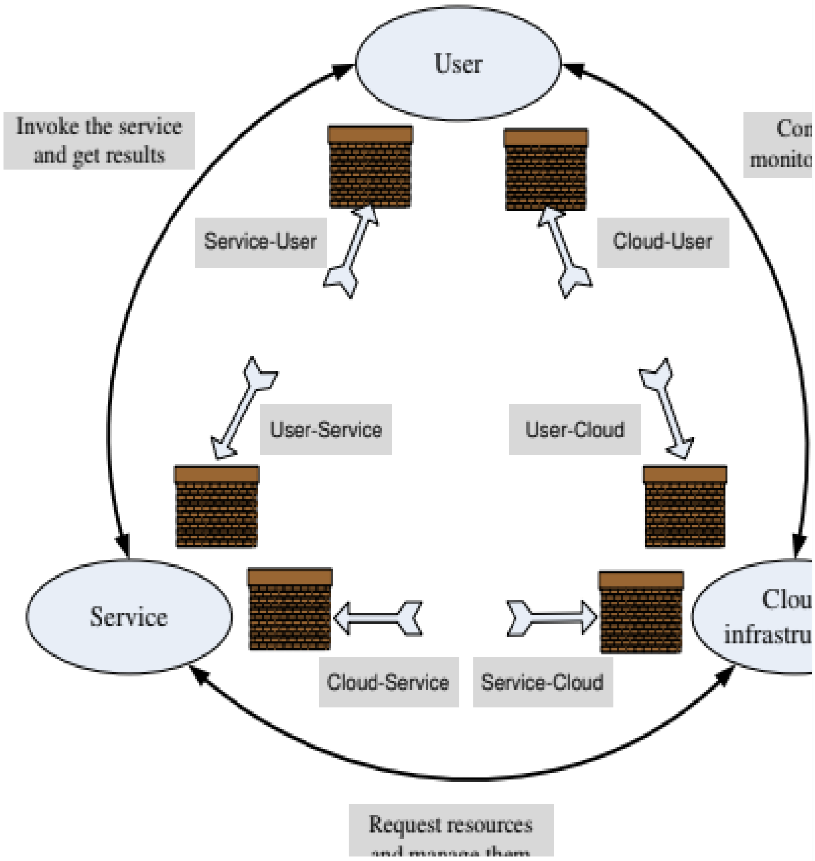
- There are multiple ways to look at the security risks for cloud computing.: traditional security threats, threats related to system availability, and threats related to third-party data control.
- **Traditional threats** are those experienced for some time by any system connected to the Internet, but with some cloud-specific twists. The impact of traditional threats is amplified due to the vast amount of cloud resources and the large user population that can be affected.
- The fuzzy bounds of responsibility between the providers of cloud services and users and the difficulties to accurately identify the cause of a problem add to concerns of cloud users.
- The traditional threats begin at the user site; the user must protect the infrastructure used to connect to the cloud and to interact with the application running on the cloud.
- This task is more difficult because some components of this infrastructure are outside the firewall protecting the user.
- The next threat is related to the **authentication and authorization** process. The procedures in place for one individual does not extend to an enterprise. In this case the cloud access of the members of an organization must be nuanced; different individuals should be assigned distinct levels of privilege based on their role in the organization.
- It is also nontrivial to merge or adapt the internal policies and security metrics of an organization with the ones of the cloud.
- Moving from the user to the cloud we see that the traditional attacks have already affected cloud service providers. The favorite means of attack are: distributed denial of service (DDDS) attacks which prevent legitimate users to access cloud services, phishing, SQL injection, or cross-site scripting.
- Cloud servers host multiple VMs and multiple applications may run under each VM.
- Multi-tenency in conjunction with VMM vulnerabilities could open new attack channels for malicious users. Identifying the path followed by an attacker is much more difficult in a cloud environment.
- Traditional investigation methods based on digital forensic cannot be extended to a cloud where the resources are shared among a large user

population and the trace of events related to a security incident is wiped out due to the high rate of write operations on any storage media.

- **Availability of cloud services** is another major concern. System failures, power outages, and other catastrophic events could shutdown cloud services for extended periods of time;
- control generates a spectrum of concerns caused by the lack of transparency and limited user control. For example, a cloud provider may subcontract some resources from a third party whose level of trust is questionable.
- There are examples when subcontractors failed to maintain the customer data. There are also examples when the third party was not a subcontractor but a hardware supplier and the loss of data was caused by poor quality storage devices.
- Storing proprietary data on the cloud is risky as cloud provider espionage poses real dangers. The terms of contractual obligations usually place all responsibilities for data security with the user.
- The Amazon Web Services customer agreement does not help user's confidence as it states "We ...will not be liable to you for any direct, indirect, incidental,....damages.... nor...be responsible for any compensation, reimbursement, arising in connection with: (A) your inability to use the services... (B) the cost of procurement of substitute goods or services..or (D) any unauthorized access to, alteration of, or deletion, destruction, damage, loss or failure to store any of your content or other data."
- It is very difficult for a cloud user to prove that data has been deleted by the service provider.
- The lack of transparency makes auditability a very difficult proposition for cloud computing. Auditing guidelines elaborated by the National Institute of Standards (NIST) such as the Federal Information Processing Standard (FIPS) and the Federal Information Security Management Act (FISMA) are mandatory for US Government agencies.
- The first release of the Cloud Security Alliance (CSA) report in 2010, identifies seven top threats to cloud computing. These threats are: the abuse use of the cloud, APIs that are not fully secure, malicious insiders, shared technology, account hijacking, data loss or leakage, and unknown risk profile.
- According to this report the IaaS delivery model can be affected by all threats. PaaS can be the affected by all, but the shared technology, while SaaS is affected by all, but abuse and shared technology.
- The abuse of the cloud refers to the ability to conduct nefarious activities from the cloud, for example use multiple AWS instances or

applications supported by IaaS to launch distributed denial of service attacks or to distribute spam and malware.

- Shared technology considers threats due to multi-tenant access supported by virtualization.
- VMMs can have flaws allowing a guest operating system to affect the security of the platform shared with other virtual machines.
- Insecure APIs may not protect the users during a range of activities starting with authentication and access control to monitoring and control of the application during runtime.
- The cloud service providers do not disclose their hiring standards and policies thus, the risks of malicious insiders cannot be ignored. The potential harm due to this particular form of attacks is high.
- Data loss or leakage are two risks with devastating consequences for an individual or an organization using cloud services.
- Maintaining copies of the data outside the cloud is often unfeasible due to the sheer volume of data. If the only copy of the data is stored on the cloud, then sensitive data is permanently lost when cloud data replication fails followed by a storage media failure. As some of the data often includes proprietary or sensitive data access to such information by third parties could have severe consequences.
- Account or service hijacking is a significant threat and cloud users must be aware of and guard against all methods to steal credentials. Lastly, unknown risk profile refers to exposure to the ignorance or underestimation of the risks of cloud computing.

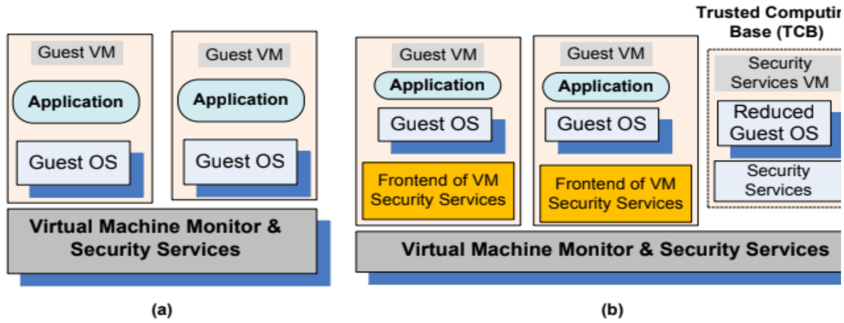

User
Invoke the service
and get results
Con
monito
Service-User
Cloud-User
User-Service
User-Cloud
Service
Clou
infrastru
Cloud-Service
Service-Cloud
Request resources
and manage them

2. **Define Trust? Explain how virtual machine security is implemented** **[10]** CO5 L2

- According to the Merriam-Webster dictionary **trust** means "assured reliance on the character, ability, strength, or truth of someone or something."

Virtual machine security:

- Hybrid and hosted VMs, expose the entire system to the vulnerability of the host OS.
- In a traditional VM the Virtual Machine Monitor (VMM) controls the access to the hardware and provides a stricter isolation of VMs from one another than the isolation of processes in a traditional OS.
- A VMM controls the execution of privileged operations and can enforce memory isolation as well as disk and network access.
- The VMMs are considerably less complex and better structured than traditional operating systems thus, in a better position to respond to security attacks.
- A major challenge a VMM sees only raw data regarding the state of a guest operating system while security services typically operate at a higher logical level, e.g., at the level of a file rather than a disk block.
- A secure TCB (Trusted Computing Base) is a necessary condition for security in a virtual machine environment; if the TCB is compromised then the security of the entire system is affected.



(a) Virtual security services provided by the VMM; (b) A dedicated security VM.

VMM-based threats

- Starvation of resources and denial of service for some VMs.
- Probable causes:

  (a) badly configured resource limits for some VMs.

  (b) a rogue VM with the capability to bypass resource limits set in VMM.
- VM side-channel attacks: malicious attack on one or more VMs by a rogue VM under the same VMM.
- Probable causes:

  (a) lack of proper isolation of inter-VM traffic due to misconfiguration of the virtual network residing in the VMM.

  (b) limitation of packet inspection devices to handle high speed traffic,

  e.g., video traffic.

  (c) presence of VM instances built from insecure VM images, e.g., a VM image having a guest OS without the latest patches.

VM-based threats

- Deployment of rogue or insecure VM. Unauthorized users may create insecure instances from images or may perform unauthorized administrative actions on existing VMs.
- Probable cause:improper configuration of access controls on VM administrative tasks such as instance creation, launching, suspension, re-activation and so on.
- Presence of insecure and tampered VM images in the VM image repository. Probable causes: (a) lack of access control to the VM image repository.(b) lack of mechanisms to verify the integrity of the images, e.g.,digitally signed image.

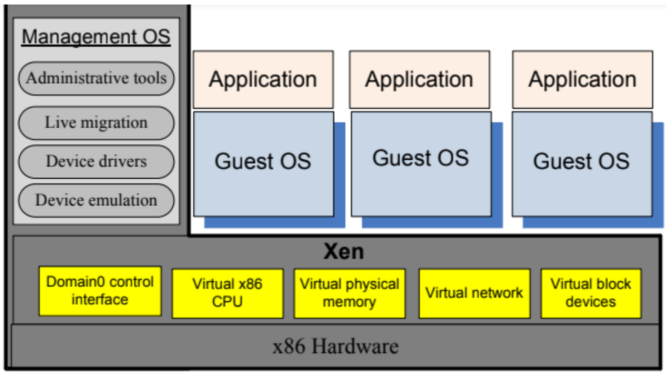
3. **Describe the concept of security a) posed by shared images** **[05]** CO5 L2

- Image sharing is critical for the IaaS cloud delivery model. For example, a user of AWS has the option to choose between.
- Amazon Machine Images (AMIs) accessible through the Quick Start. Community AMI menus of the EC2 service.
- Many of the images analyzed by a recent report allowed a user to undelete files, recover credentials, private keys, or other types of sensitive information with little effort and using standard tools.
- A software vulnerability audit revealed that 98% of the Windows AMIs and 58% of Linux AMIs audited had critical vulnerabilities.
- Security risks:
    - Backdoors and leftover credentials.
    - Unsolicited connections.
    - Malware.

b) posed by a management OS **[05]** CO5 L2

- Virtual machine monitor, or hypervisor, is considerably smaller than an operating system, e.g., the Xen VMM has ~ 60,000 lines of code.

- The Trusted Computer Base (TCB) of a cloud computing environment includes not only the hypervisor but also the management OS.
- The management OS supports administrative tools, live migration, device drivers, and device emulators.
- In Xen the management operating system runs in Dom0; it manages the building of all user domains, a process consisting of several steps:
- Allocate memory in the Dom0 address space and load the kernel of the guest operating system from the secondary storage.
- Allocate memory for the new VM and use foreign mapping to load the kernel to the new VM.
- Set up the initial page tables for the new VM.
- Release the foreign mapping on the new VM memory, set up the virtual CPU registers and launch the new VM.



The trusted computing base of a Xen-based environment includes the hardware, Xen, and the management operating system running in Dom0. The management OS supports administrative tools, live migration, device drivers, and device emulators. A guest operating system and applications running under it reside in a DomU.

Possible actions of a malicious Dom0

- At the time it creates a DomU: Refuse to carry out the steps necessary to start the new VM.
- Modify the kernel of the guest OS to allow a third party to monitor and control the execution of applications running under the new VM.
- Undermine the integrity of the new VM by setting the wrong page tables and/or setup wrong virtual CPU registers.

- Refuse to release the foreign mapping and access the memory while the new VM is running.
- At run time: Dom0 exposes a set of abstract devices to the guest operating systems using split drivers with the frontend of in a DomU and the backend in Dom0.
- We have to ensure that run time communication through Dom0 is encrypted.
- Transport Layer Security (TLS) does not guarantee that Dom0 cannot extract cryptographic keys from the memory of the OS and applications running in DomU.

How to deal with run-time vulnerability of Dom0

- To implement a secure run-time system, we have to intercept and control the hypercalls used for communication between a Dom0 that cannot be trusted and a DomU we want to protect.
- New hypercalls are necessary to protect:

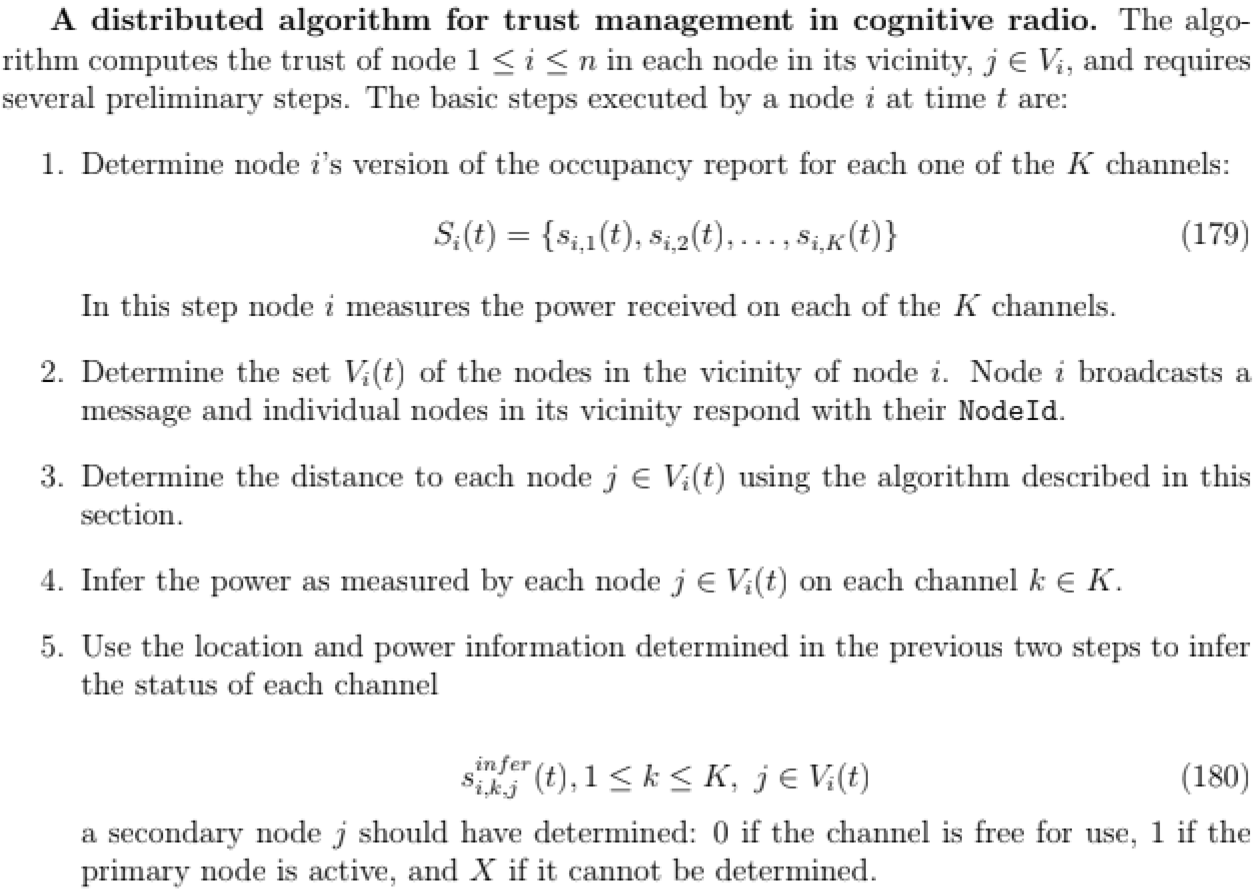
  □ The privacy and integrity of the virtual CPU of a VM. When Dom0 wants to save the state of the VM the hypercall should be intercepted and the contents of the virtual CPU registers should be encrypted. When DomU is restored, the virtual CPU context should be decrypted and then an integrity check should be carried out.
- The privacy and integrity of the VM virtual memory. The page table update hypercall should be intercepted and the page should be encrypted so that Dom0 handles only encrypted pages of the VM. To guarantee the integrity, the hypervisor should calculate a hash of all the memory pages before they are saved by Dom0.
- An address translation is necessary as a restored DomU may be allocated a different memory region.
- The freshness of the virtual CPU and the memory of the VM. The solution is to add to the hash a version number.

4. **Explain the security rules for application and transport layer protocols in EC2** **[10]** CO6 L2

- A client must know the IP address of a virtual machine in the cloud, to be able to connect to it.
- Domain Name Service (DNS) are used to map human-friendly names of computer systems to IP addresses in the Internet or in private networks.
- DNS is a hierarchical distributed database and plays a role reminiscent of a phone books in the Internet.
- In late 2010, Amazon announced a DNS service called Route 53 to route users to AWS services and to infrastructure outside of AWS.
- A network of DNS servers scattered across the globe, which enables customers to gain reliable access to AWS ans place strict controls over who can manage their DNS system by allowing integration with AWS Identity and Access Management(IAM).
- For several reasons, including security and the ability of the infrastructure to scale up,the IP addresses of instances visible to the outside world are mapped internally to private IP addresses.
- A virtual machine running under Amazon's EC2 has several IP addresses:EC2 Private IP Address: The internal address of an instance; it is only used for routing within the EC2 cloud.
- EC2 Public IP Address: Network traffic originating outside the AWS network must use either the public IP address or the elastic IP address of the instance.
- The public IP address is translated using the Network Address Translation (NAT) to the private IP address when an instance is launched and it is valid until the instance is terminated.
- Traffic to the public address is forwarded to the private IP address of the instance.
- EC2 Elastic IP Address: The IP address allocated to an AWS account and used by traffic originated outside AWS. NAT is used to map an elastic IP address to the private IP address.
- Elastic IP addresses allow the cloud user to mask instance or availability zone failures by programmatically re-mapping a public IP addresses to any instance associated with the user's account. This allows fast recovery after a system failure; for example, rather than waiting for a cloud maintenance team to reconfigure or replace the failing host, or waiting for DNS to propagate the new public IP to all of the customers of a web service hosted by EC2, the web service provider can re-map the elastic IP address to a replacement instance.
- Amazon charges a fee for unallocated Elastic IP addresses.
- Amazon Web Services use security groups to control access to user's virtual machines.

- A virtual machine instance belongs to one, and only one, security group, which can only be defined before the instance is launched. Once an instance is running, the security group the instance belongs to cannot be changed.
- The following steps allow the user to add a security rule:
  1. Sign in to the AWS Management Console at http://aws.amazon.com using your Email address and password and select EC2 service.
  2. Use the EC2 Request Instance Wizard to specify the instance type, whether it should be monitored, and specify a key/value pair for the instance to help organize and search.
  3. Provide a name for the key pair, then on the left hand side panel choose Security Groups under Network & Security, select the desired security group and click on the Inbound tab to enter the desired rule.

5.	**Describe the distributed algorithm for trust management in cognitive radio**	**[10]**	**CO6**	**L2**



A distributed algorithm for trust management in cognitive radio. The algorithm computes the trust of node $1 \leq i \leq n$ in each node in its vicinity, $j \in V_i$, and requires several preliminary steps. The basic steps executed by a node i at time t are:

1. Determine node i's version of the occupancy report for each one of the K channels:

$$S_i(t) = \{s_{i,1}(t), s_{i,2}(t), \ldots, s_{i,K}(t)\} \tag{179}$$

   In this step node i measures the power received on each of the K channels.

2. Determine the set $V_i(t)$ of the nodes in the vicinity of node i. Node i broadcasts a message and individual nodes in its vicinity respond with their `NodeId`.

3. Determine the distance to each node $j \in V_i(t)$ using the algorithm described in this section.

4. Infer the power as measured by each node $j \in V_i(t)$ on each channel $k \in K$.

5. Use the location and power information determined in the previous two steps to infer the status of each channel

$$s_{i,k,j}^{infer}(t), 1 \leq k \leq K,\ j \in V_i(t) \tag{180}$$

   a secondary node j should have determined: 0 if the channel is free for use, 1 if the primary node is active, and X if it cannot be determined.

$$s_{i,k,j}^{infer}(t) = \begin{cases} 0 & \text{if secondary node j decides that channel k is free} \\ 1 & \text{if secondary node j decides that channel k is used by the primary} \\ X & \text{if no inference can be made} \end{cases} \tag{181}$$

6. Receive the information provided by neighbor $j \in V_i(t)$, $S_{i,k,j}^{recv}(t)$.

7. Compare the information provided by neighbor $j \in V_i(t)$

   $$S_{i,k,j}^{recv}(t) = \{s_{i,1,j}^{recv}(t), s_{i,2,j}^{recv}(t), \ldots, s_{i,K,j}^{recv}(t)\} \tag{182}$$

   with the information inferred by node i about node j

   $$S_{i,k,j}^{infer}(t) = \{s_{i,1,j}^{infer}(t), s_{i,2,j}^{infer}(t), \ldots, s_{i,K,j}^{infer}(t)\} \tag{183}$$

8. Compute the number of matches, mismatches, and cases when no inference is possible, respectively,

   $$\alpha_{i,j}(t) = \mathcal{M}\left[S_{i,k,j}^{infer}(t), S_{i,k,j}^{recv}(t)\right] \tag{184}$$

   with $\mathcal{M}$ the number of matches between the two vectors,

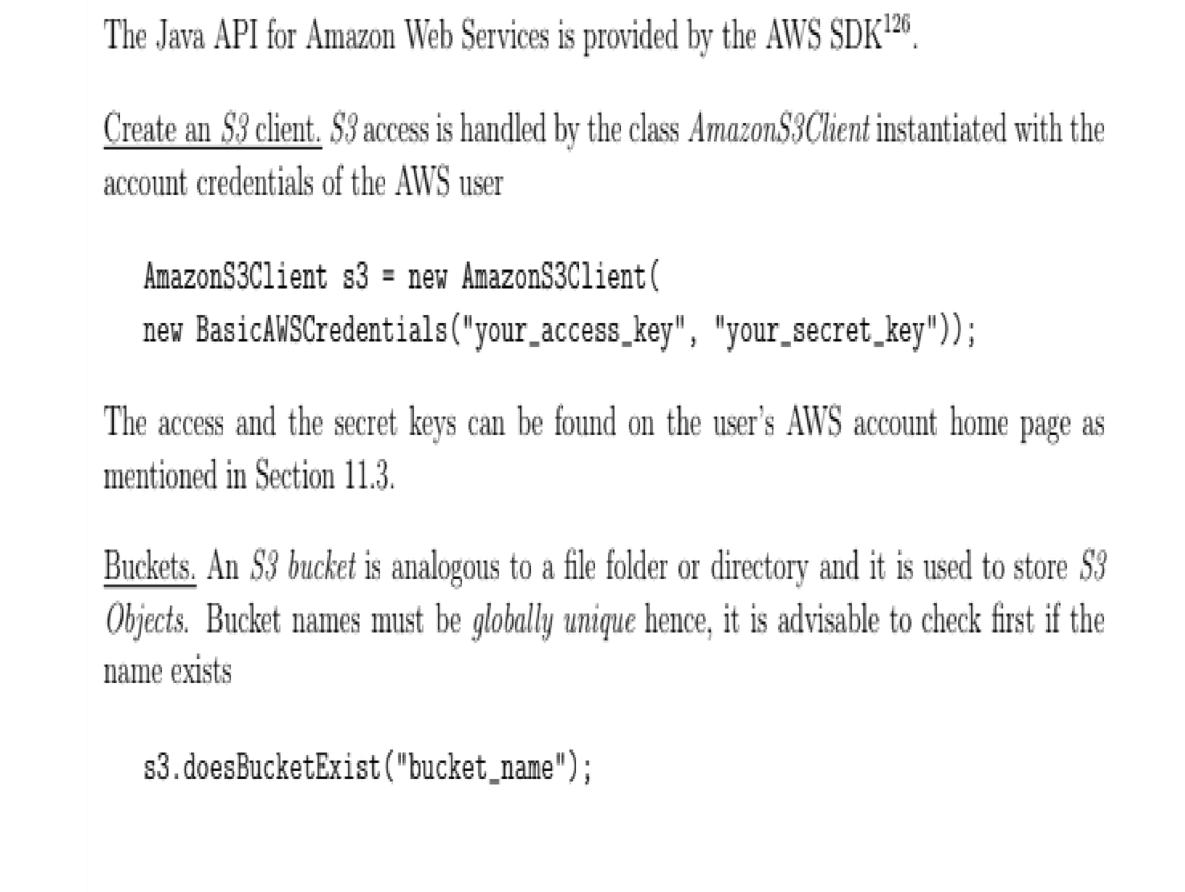
   $$\beta_{i,j}(t) = \mathcal{N}\left[S_{i,k,j}^{infer}(t), S_{i,k,j}^{recv}(t)\right] \tag{185}$$

   with $\mathcal{N}$ the number of mismatches between the two vectors, and $X_{i,j}(t)$ the number of cases where no inference could be made.

9. Use the quantities $\alpha_{i,j}(t)$, $\beta_{i,j}(t)$, and $X_{i,j}(t)$ to assess the trust in node j. For example, compute the trust of node i in node j at time t as

   $$\zeta_{i,j}(t) = [1 + X_{i,j}(t)] \frac{\alpha_{i,j}(t)}{\alpha_{i,j}(t) + \beta_{i,j}(t)} \tag{186}$$

6. **How to use S3 in java** **[10]** CO6 L2



The Java API for Amazon Web Services is provided by the AWS SDK[126].

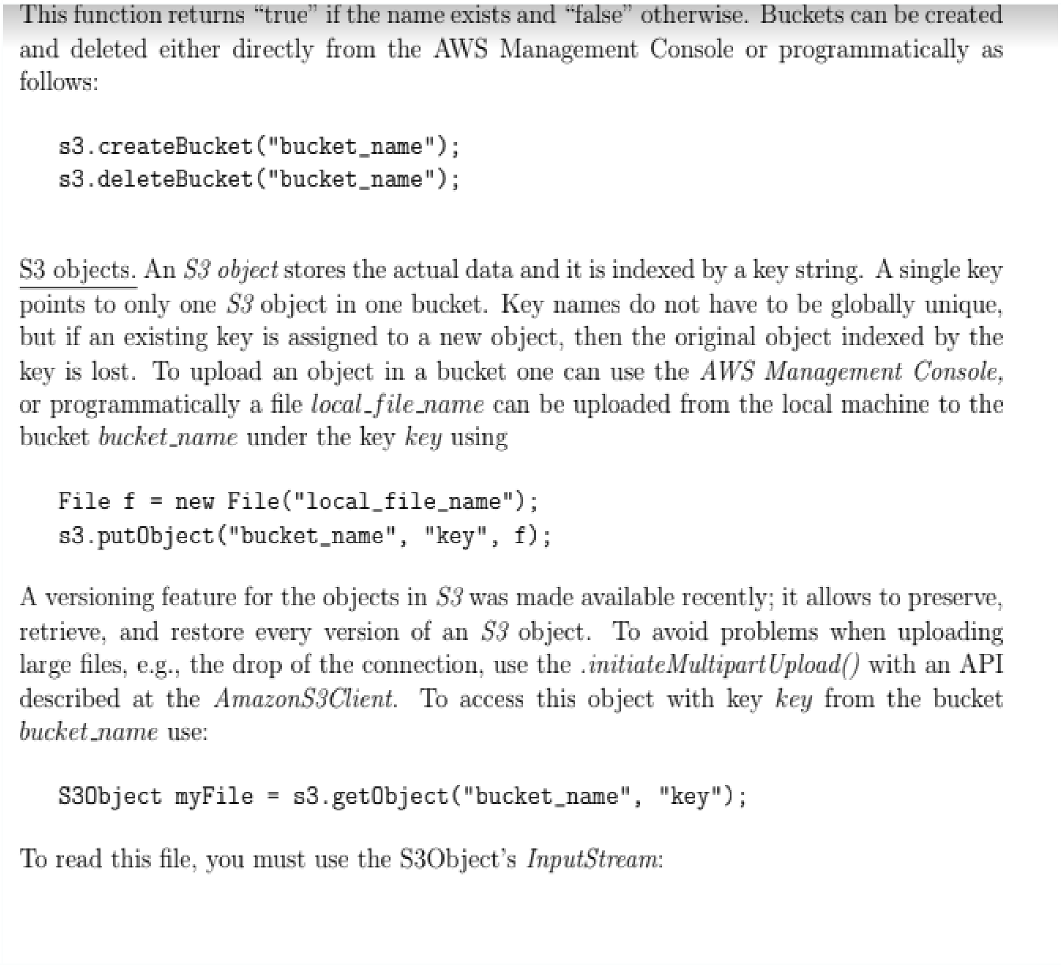
Create an S3 client. *S3* access is handled by the class *AmazonS3Client* instantiated with the account credentials of the AWS user

```
AmazonS3Client s3 = new AmazonS3Client(
new BasicAWSCredentials("your_access_key", "your_secret_key"));
```

The access and the secret keys can be found on the user's AWS account home page as mentioned in Section 11.3.

Buckets. An *S3 bucket* is analogous to a file folder or directory and it is used to store *S3 Objects*. Bucket names must be *globally unique* hence, it is advisable to check first if the name exists

```
s3.doesBucketExist("bucket_name");
```



This function returns "true" if the name exists and "false" otherwise. Buckets can be created and deleted either directly from the AWS Management Console or programmatically as follows:

```
s3.createBucket("bucket_name");
s3.deleteBucket("bucket_name");
```

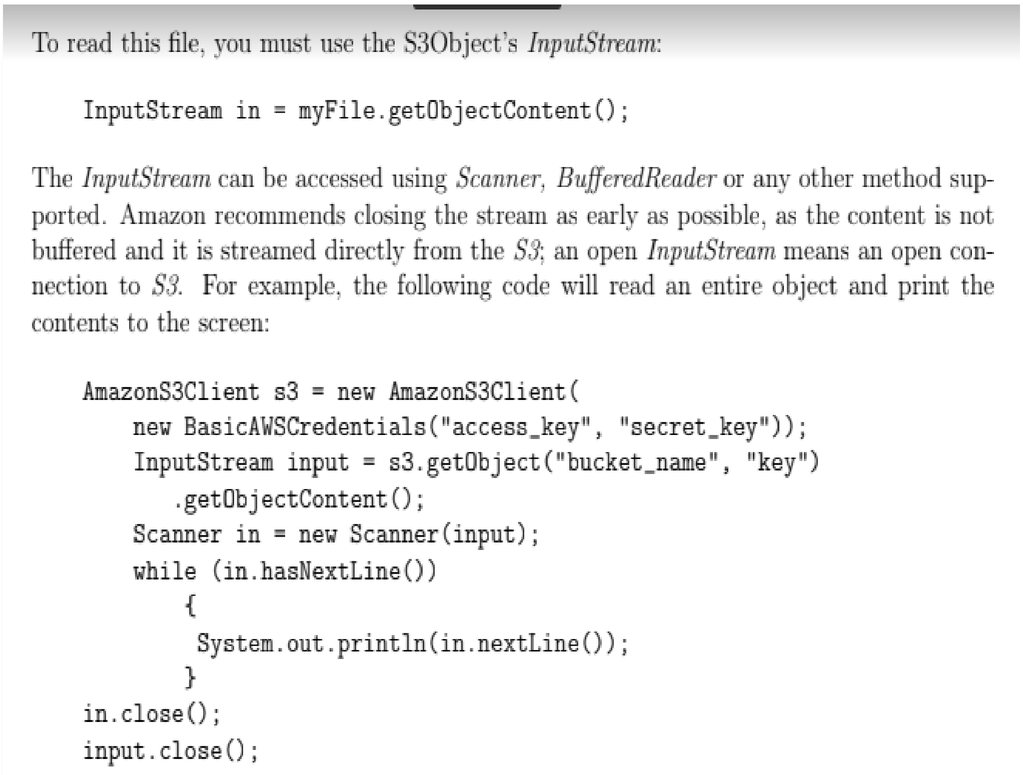
S3 objects. An *S3 object* stores the actual data and it is indexed by a key string. A single key points to only one *S3* object in one bucket. Key names do not have to be globally unique, but if an existing key is assigned to a new object, then the original object indexed by the key is lost. To upload an object in a bucket one can use the *AWS Management Console*, or programmatically a file *local_file_name* can be uploaded from the local machine to the bucket *bucket_name* under the key *key* using

```
File f = new File("local_file_name");
s3.putObject("bucket_name", "key", f);
```

A versioning feature for the objects in *S3* was made available recently; it allows to preserve, retrieve, and restore every version of an *S3* object. To avoid problems when uploading large files, e.g., the drop of the connection, use the *.initiateMultipartUpload()* with an API described at the *AmazonS3Client*. To access this object with key *key* from the bucket *bucket_name* use:

```
S3Object myFile = s3.getObject("bucket_name", "key");
```

To read this file, you must use the S3Object's *InputStream*:



To read this file, you must use the S3Object's *InputStream*:

```
    InputStream in = myFile.getObjectContent();
```

The *InputStream* can be accessed using *Scanner*, *BufferedReader* or any other method supported. Amazon recommends closing the stream as early as possible, as the content is not buffered and it is streamed directly from the *S3*; an open *InputStream* means an open connection to *S3*. For example, the following code will read an entire object and print the contents to the screen:

```
    AmazonS3Client s3 = new AmazonS3Client(
        new BasicAWSCredentials("access_key", "secret_key"));
        InputStream input = s3.getObject("bucket_name", "key")
           .getObjectContent();
        Scanner in = new Scanner(input);
        while (in.hasNextLine())
            {
             System.out.println(in.nextLine());
            }
    in.close();
    input.close();
```

7.

Describe the concept a) Cloud based optimal FPGA synthesis **[05]** CO5 L2

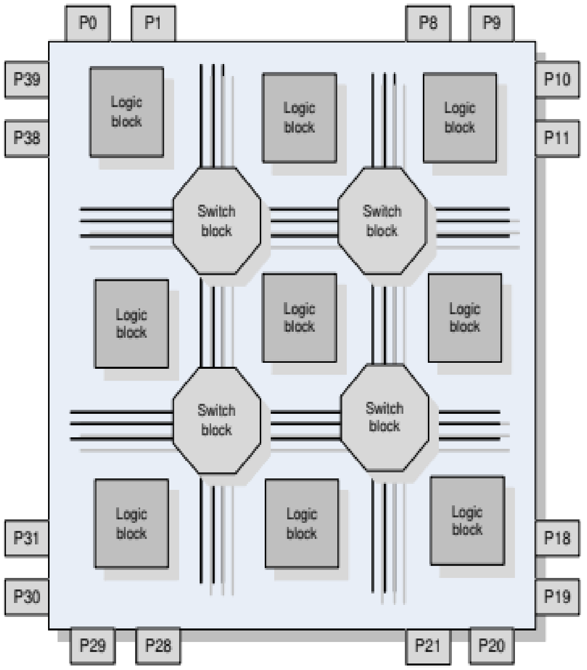
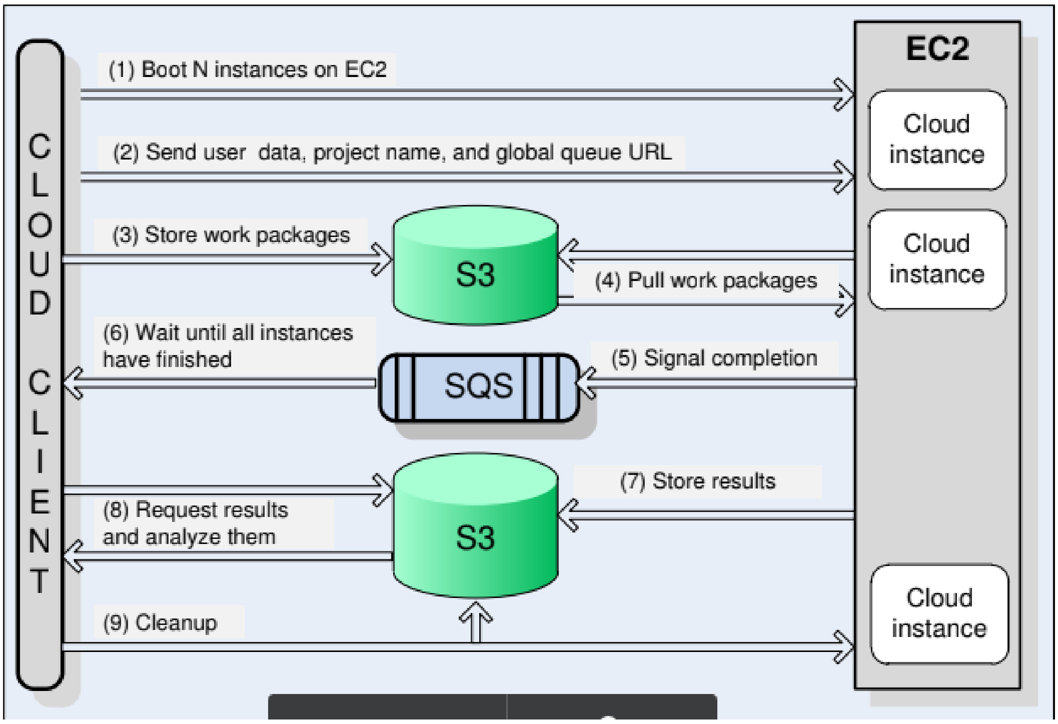


Figure 118: The structure of an FPGA with 30 pins, $P1$-$P29$, 9 logic blocks and 4 switch-blocks.

A cloud is an ideal running environment for scientific applications which involve model development; in this case multiple cloud instances could concurrently run slightly different models of the system. When the model is described by a set of parameters, the application can be based on the SPMD (Same Program Multiple Data) paradigm combined with an analysis phase when the results from the multiple instances are ranked based on a well-defined metric. In this case there is no communication during the first phase of the application, when partial results are produced and then written to storage server; then individual instances signal the completion and a new instance to carry out the analysis and display the results is started. A similar strategy can be used by engineering applications of mechanical, civil, electrical, electronic, or any other system design area. In this case the multiple instances run concurrent design for different sets of parameters of the system.


EC2
CLOUD CLIENT
(1) Boot N instances on EC2
Cloud instance
(2) Send user data, project name, and global queue URL
(3) Store work packages
S3
Cloud instance
(4) Pull work packages
(6) Wait until all instances have finished
SQS
(5) Signal completion
(7) Store results
(8) Request results and analyze them
S3
Cloud instance
(9) Cleanup

A cloud application for optimal design of field-programmable gate arrays (FPGAs) is discussed next. As the name suggests, an FPGA is an integrated circuit designed to be configured/adapted/programmed in the field to perform a well-defined function [310]. Such a circuit consists of *logic blocks* and *interconnects* that can be "programmed" to carry out logical and/or combinatorial functions, see Figure 118.

The first commercially viable FPGA, XC2064, was produced in 1985 by Xilinx. Today FPGAs are used in many areas including digital signal processing, CRNs, aerospace, medical imaging, computer vision, speech recognition, cryptography, and computer hardware emulation. FPGAs are less energy efficient and slower than application-specific integrated circuits (ASICs). The widespread use of FPGAs is due to their flexibility and the ability to reprogram them.

Hardware description languages (HDLs) such as VHDL and Verilog are used to program FPGAs; HDLs are used to specify a register-transfer level (RTL) description of the circuit. Multiple stages are used to synthesize FPGL.

b)Cloud service for adaptive data streaming **[05]** CO5 L2

- Data streaming involves three entities, the sender, a communication network, and a receiver.
- The resources necessary to guarantee the timing constraints include CPU cycles and buffer space at the sender and the receiver and network bandwidth.
- Adaptive data streaming determines the data rate based on the available resources.
- Lower data rates imply lower quality, but reduce the demands for system resources.
- Adaptive data streaming is possible only if the application permits tradeoffs between quantity and quality.
- Such tradeoffs are feasible for audio and video streaming which allow lossy compression, but are not acceptable for many applications which processes a continuous stream of data collected by sensors.
- Data streaming requires accurate information about all resources involved and this implies that the network bandwidth has to be constantly monitored; at the same time, the scheduling algorithms should be coordinated with memory management to guarantee the timing constraints.
- Adaptive data streaming poses additional constraints because the data flow is dynamic. Indeed, once we detect that the network cannot accommodate the data rate required by an audio or video stream we have to reduce the data rate thus, to convert to a lower quality audio or video. Data conversion can be done on the fly and, in this case, the data flow on the cloud has to be changed.
- Accommodating dynamic data flows with timing constraints is non-trivial; only about 18% of the top 100 global video web sites use ABR (Adaptive Bit Rate) technologies for streaming.
- This application stores the music files in S3 buckets and the audio service runs on the EC2 platform. In EC2 each virtual machine functions as a virtual private server and is called an instance; an instance specifies the maximum amount of resources available to an application, the interface for that instance, as well as, the cost per hour.
- EC2 allows the import of virtual machine images from the user environment to an instance through a facility called VM import.

- It also distributes automatically the incoming application traffic among multiple instances using the elastic load balancing facility.
- EC2 associates an elastic IP address with an account; this mechanism allows a user to mask the failure of an instance and re-map a public IP address to any instance of the account, without the need to interact with the software support team.
- The adaptive audio streaming involves a multi-objective optimization problem. We wish to convert the highest quality audio file stored on the cloud to a resolution corresponding to the rate that can be sustained by the available bandwidth; at the same time, we wish to minimize the cost on the cloud site, and also minimize the buffer requirements for the mobile device to accommodate the transmission jitter.
- Finally, we wish to reduce to a minimum the start-up time for the content delivery.
- A first design decision is if data streaming should only begin after the conversion from the WAV to MP3 format has been completed, or it should proceed concurrently with conversion, in other words start as soon as several MP3 frames have been generated; another question is if the converted music file should be saved for later use or discarded.
- To answer these question we experimented with conversion from the highest quality audio files which require a 320 Kbps data rate to lower quality files corresponding to 192, 128, 64, 32 and finally 16 Kbps.
- If the conversion time is small and constant there is no justification for pipelining data conversion and streaming, a strategy which complicates the processing flow on the cloud. It makes sense to cache the converted copy for a limited period of time with the hope that it will be reused in the next future.
- Another design decision is how the two services should interact to optimize the performance; two alternatives come to mind:
- 1. The audio service running on the EC2 platform requests the data file from the S3,converts it, and, eventually, sends it back. The solution involves multiple delays and it is far from optimal.
- 2. Mount the S3 bucket as an EC2 drive. This solution reduces considerably the start-up time for audio streaming.