IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

CLOUD COMPUTING

CLOUD SECURITY I

PROF. SOUMYA K. GHOSH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

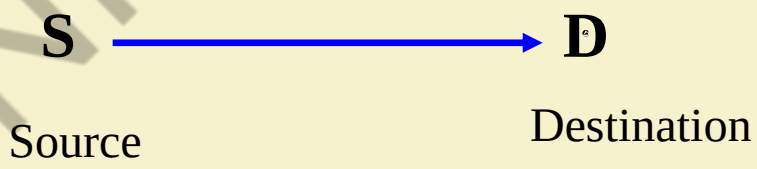
IIT KHARAGPUR

Security - Basic Components

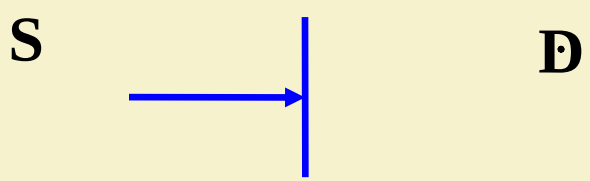
- Confidentiality
  - Keeping data and resources hidden
- Integrity
  - Data integrity (integrity)
  - Origin integrity (authentication)
- Availability
  - Enabling access to data and resources

Security Attacks

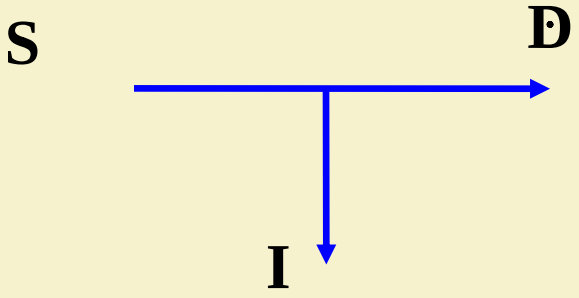
- Any action that compromises the security of information.
- Four types of attack:
  1. Interruption
  2. Interception
  3. Modification
  4. Fabrication
- Basic model:



Security Attacks (contd.)



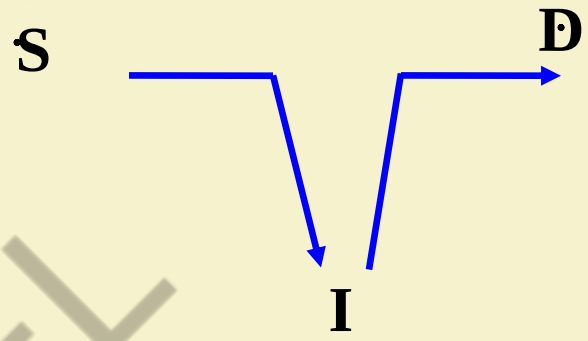
- Interruption:
  - Attack on availability



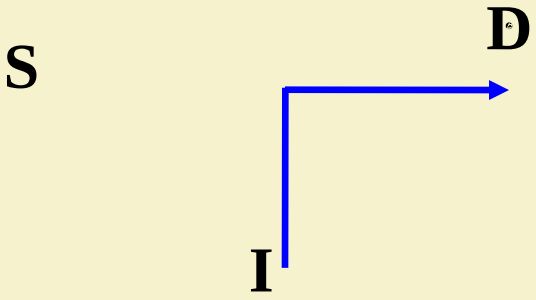
- Interception:
  - Attack on confidentiality

Security Attacks

- Modification:
  - Attack on integrity



- Fabrication:
  - Attack on authenticity



Classes of Threats

- Disclosure
  - Snooping
- Deception
  - Modification, spoofing, repudiation of origin, denial of receipt
- Disruption
  - Modification
- Usurpation
  - Modification, spoofing, delay, denial of service

Policies and Mechanisms

- Policy says what is, and is not, allowed
  - This defines "security" for the site/system/*etc*.
- Mechanisms enforce policies
- Composition of policies
  - If policies conflict, discrepancies may create security vulnerabilities

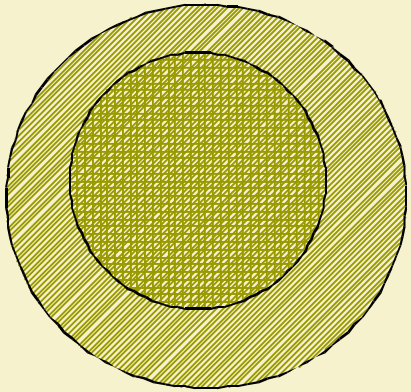
Goals of Security

- Prevention
  - Prevent attackers from violating security policy
- Detection
  - Detect attackers' violation of security policy
- Recovery
  - Stop attack, assess and repair damage
  - Continue to function correctly even if attack succeeds

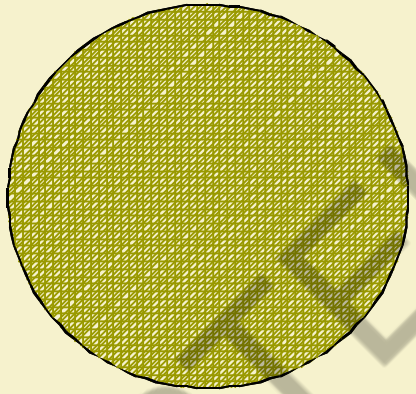
Trust and Assumptions

- Underlie all aspects of security
- Policies
  - Unambiguously partition system states
  - Correctly capture security requirements
- Mechanisms
  - Assumed to enforce policy
  - Support mechanisms work correctly

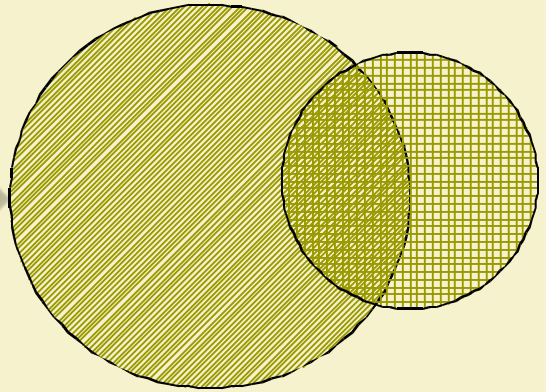
Types of Mechanisms



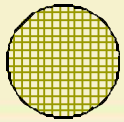
secure



precise



broad



set of reachable states

set of secure states

Assurance

- Specification
  - Requirements analysis
  - Statement of desired functionality
- Design
  - How system will meet specification
- Implementation
  - Programs/systems that carry out design

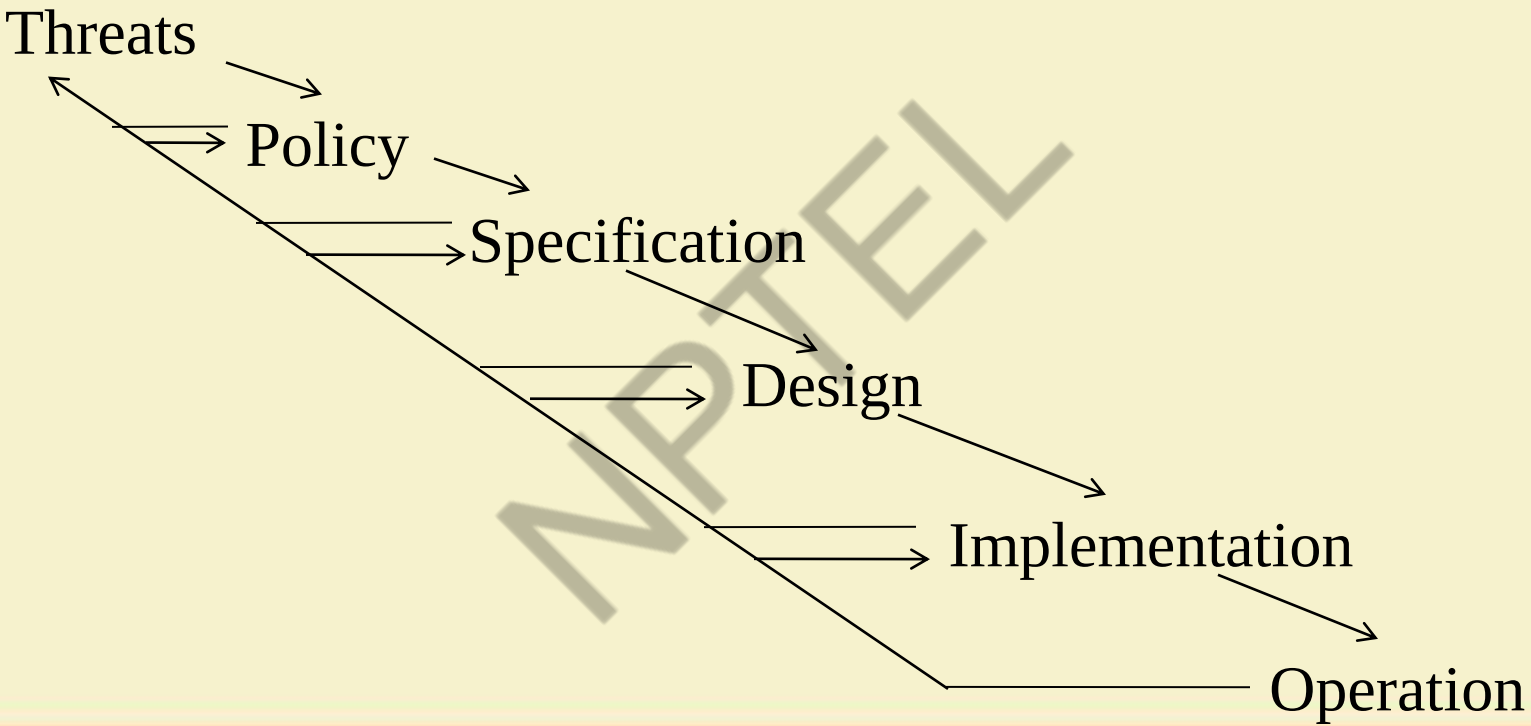
Operational Issues

- Cost-Benefit Analysis
  - Is it cheaper to prevent or recover?
- Risk Analysis
  - Should we protect something?
  - How much should we protect this thing?
- Laws and Customs
  - Are desired security measures illegal?
  - Will people do them?

Human Issues

- Organizational Problems
  - Power and responsibility
  - Financial benefits
- People problems
  - Outsiders and insiders
  - Social engineering

Tying Together



Passive and Active Attacks

- Passive attacks
  - Obtain information that is being transmitted (eavesdropping).
  - Two types:
    - Release of message contents:- It may be desirable to prevent the opponent from learning the contents of the transmission.
    - Traffic analysis:- The opponent can determine the location and identity of communicating hosts, and observe the frequency and length of messages being exchanged.
  - Very difficult to detect.

- Active attacks
  - Involve some modification of the data stream or the creation of a false stream.
  - Four categories:
    - Masquerade:- One entity pretends to be a different entity.
    - Replay:- Passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.
    - Modification:- Some portion of a legitimate message is altered.
    - Denial of service:- Prevents the normal use of communication facilities.

Security Services

- Confidentiality (privacy)
- Authentication (who created or sent the data)
- Integrity (has not been altered)
- Non-repudiation (the order is final)
- Access control (prevent misuse of resources)
- Availability (permanence, non-erasure)
  - Denial of Service Attacks
  - Virus that deletes files

Role of Security

- A security infrastructure provides:
  - Confidentiality – protection against loss of privacy
  - Integrity – protection against data alteration/ corruption
  - Availability – protection against denial of service
  - Authentication – identification of legitimate users
  - Authorization – determination of whether or not an operation is allowed by a certain user
  - Non-repudiation – ability to trace what happened, & prevent denial of actions
  - Safety – protection against tampering, damage & theft

Types of Attack

- Social engineering/phishing
- Physical break-ins, theft, and curb shopping
- Password attacks
- Buffer overflows
- Command injection
- Denial of service
- Exploitation of faulty application logic
- Snooping
- Packet manipulation or fabrication
- Backdoors

Network Security…

- Network security works like this:
  - Determine network security policy
  - Implement network security policy
  - Reconnaissance
  - Vulnerability scanning
  - Penetration testing
  - Post-attack investigation

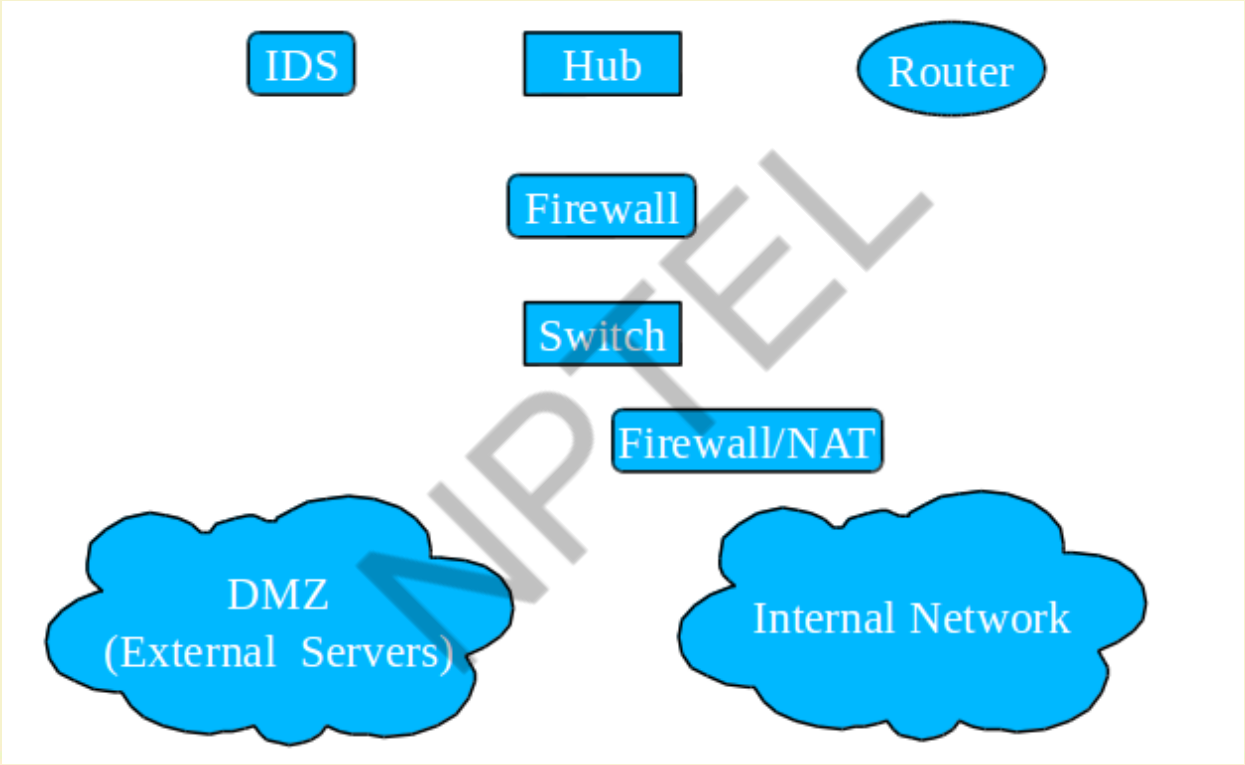
Step 1: Determine Security Policy

- A security policy is a full security roadmap
  - Usage policy for networks, servers, etc.
  - User training about password sharing, password strength, social engineering, privacy, etc.
  - Privacy policy for all maintained data
  - A schedule for updates, audits, etc.
- The network design should reflect this policy
  - The placement/protection of database/file servers
  - The location of demilitarized zones (DMZs)
  - The placement and rules of firewalls
  - The deployment of intrusion detection systems (IDSs)

Step 2: Implement Security Policy

- Implementing a security policy includes:
  - Installing and configuring firewalls
    - *iptables* is a common free firewall configuration for Linux
    - Rules for incoming packets should be created
      - These rules should drop packets by default
    - Rules for outgoing packets *may* be created
      - This depends on your security policy
  - Installing and configuring IDSes
    - *snort* is a free and upgradeable IDS for several platforms
    - Most IDSs send alerts to log files regularly
    - Serious events can trigger paging, E-Mail, telephone

Step 2: Implement Security Policy



Step 2: Implement Security Policy

- Firewall
  - Applies filtering rules to packets passing through it
  - Comes in three major types:
    - Packet filter – Filters by destination IP, port or protocol
    - Stateful – Records information about ongoing TCP sessions, and ensures out-of-session packets are discarded
    - Application proxy – Acts as a proxy for a specific application, and scans all layers for malicious data
- Intrusion Detection System (IDS)
  - Scans the incoming messages, and creates alerts when suspected scans/attacks are in progress
- Honeypot/honeynet (e.g. honeyd)
  - Simulates a decoy host (or network) with services

Step 3: Reconnaissance

- First, we learn about the network
  - IP addresses of hosts on the network
  - Identify key servers with critical data
  - Services running on those hosts/servers
  - Vulnerabilities on those services
- Two forms: passive and active
  - Passive reconnaissance is undetectable
  - Active reconnaissance is often detectable by IDS

Step 4: Vulnerability Scanning

- We now have a list of hosts and services
  - We can now target these services for attacks
- Many scanners will detect vulnerabilities (e.g. nessus)
  - These scanners produce a risk report
- Other scanners will allow you to exploit them (e.g. metasploit)
  - These scanners find ways in, and allow you to choose the payload to use (e.g. obtain a root shell, download a package)
  - The payload is the code that runs once inside
- The best scanners are updateable
  - For new vulnerabilities, install/write new plug-ins
  - e.g. Nessus Attack Scripting Language (NASL)

Step 5: Penetration Testing

- We have identified vulnerabilities
  - Now, we can exploit them to gain access
  - Using frameworks (e.g. metasploit), this is as simple as selecting a payload to execute
  - Otherwise, we manufacture an exploit
- We may also have to try to find new vulnerabilities
  - This involves writing code or testing functions accepting user input

Step 6: Post-Attack Investigation

- Forensics of Attacks
- This process is heavily guided by laws
  - Also, this is normally done by a third party
- Retain chain of evidence
  - The evidence in this case is the data on the host
  - The log files of the compromised host hold the footsteps and fingerprints of the attacker
  - Every minute with that host must be accounted for
  - For legal reasons, you should examine a low-level copy of the disk and not modify the original

Thank You!

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

CLOUD COMPUTING

CLOUD SECURITY II

PROF. SOUMYA K. GHOSH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IIT KHARAGPUR

Cloud Computing

- **Cloud computing** is a new computing paradigm, involving data and/or computation outsourcing, with
  - Infinite and elastic resource scalability
  - On demand "just-in-time" provisioning
  - No upfront cost … pay-as-you-go
- Use as much or as less you need, use only when you want, and pay only what you use

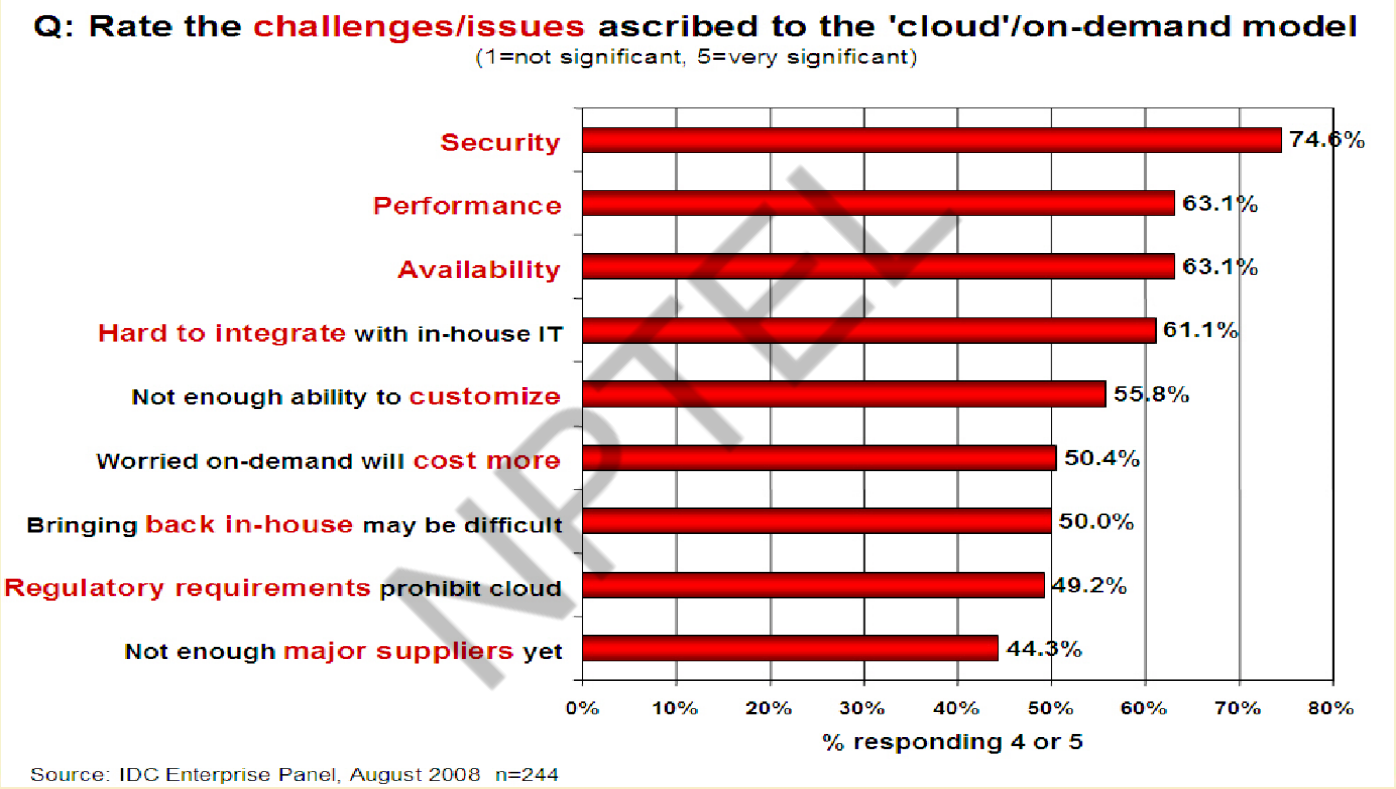
Economic Advantages of Cloud Computing

- For consumers:
  - No upfront commitment in buying/leasing hardware
  - Can scale usage according to demand
  - Minimizing start-up costs
    - Small scale companies and startups can reduce CAPEX (Capital Expenditure)
- For providers:
  - Increased utilization of datacenter resources

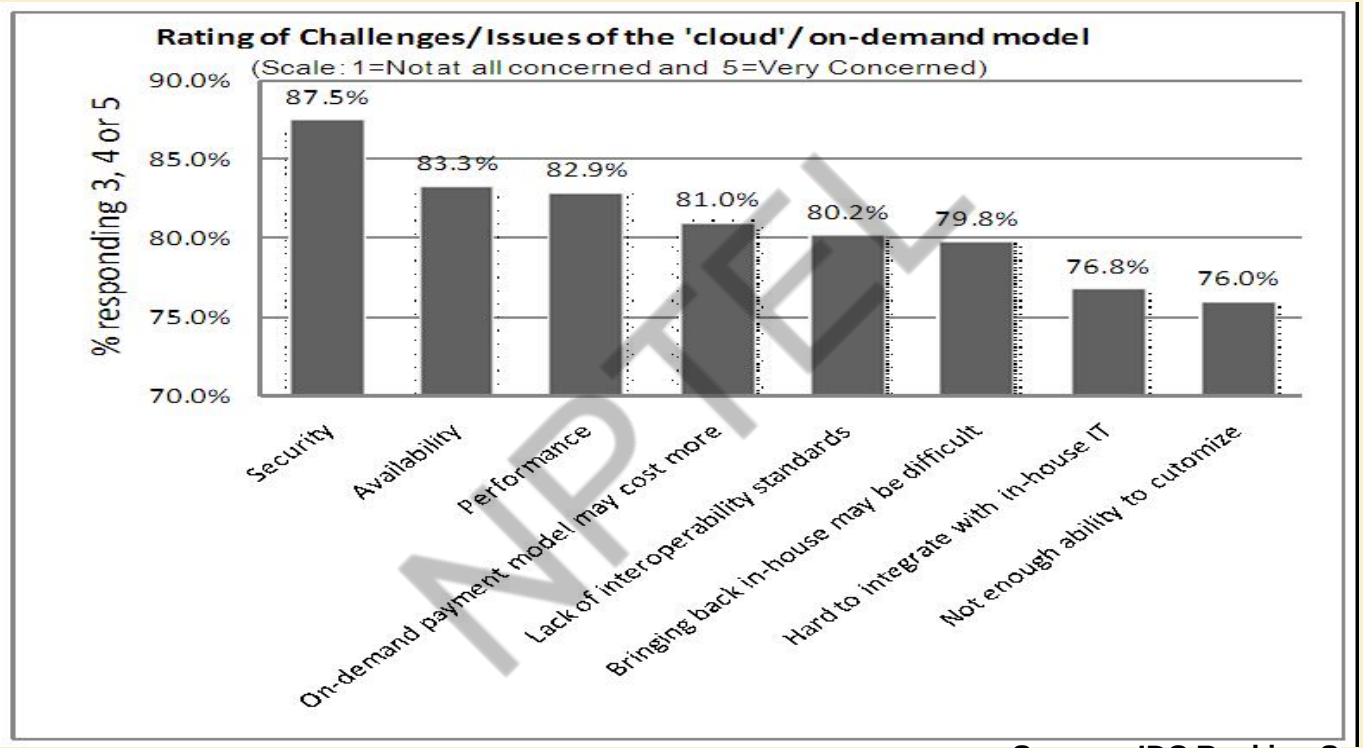
Why aren't Everyone using Cloud?

Clouds are still subject to traditional data confidentiality, integrity, availability, and privacy issues, plus some additional attacks

Concern…



Survey on Potential Cloud Barriers



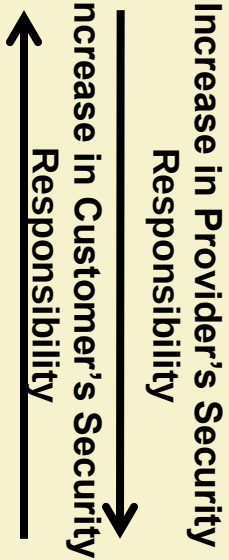
Source: IDC Ranking Security Challenges

Why Cloud Computing brings New Threats?

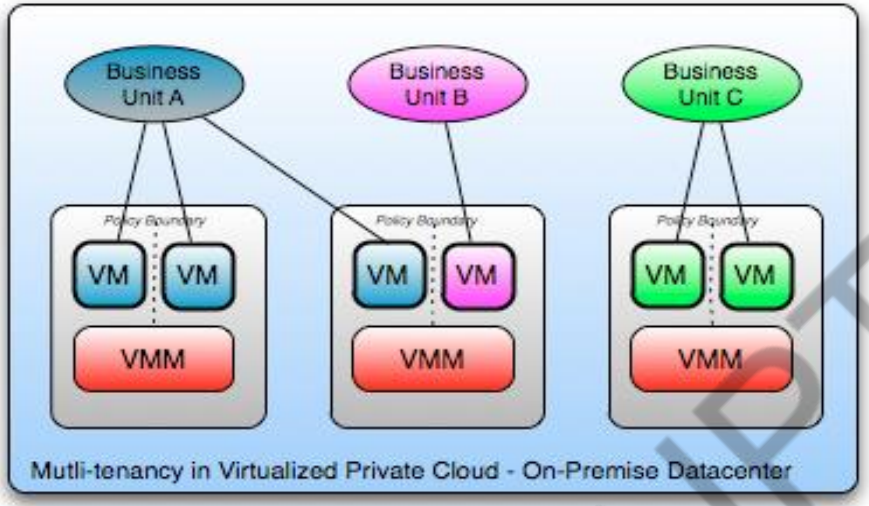
- Traditional system security mostly means keeping attackers out
- The attacker needs to either compromise the authentication/access control system, or impersonate existing users
- But cloud allows **co-tenancy**: Multiple independent users share the same physical infrastructure
  - An attacker can legitimately be in the same physical machine as the target
- Customer's **lack of control** over his own data and application.
- **Reputation fate-sharing**

Security Stack

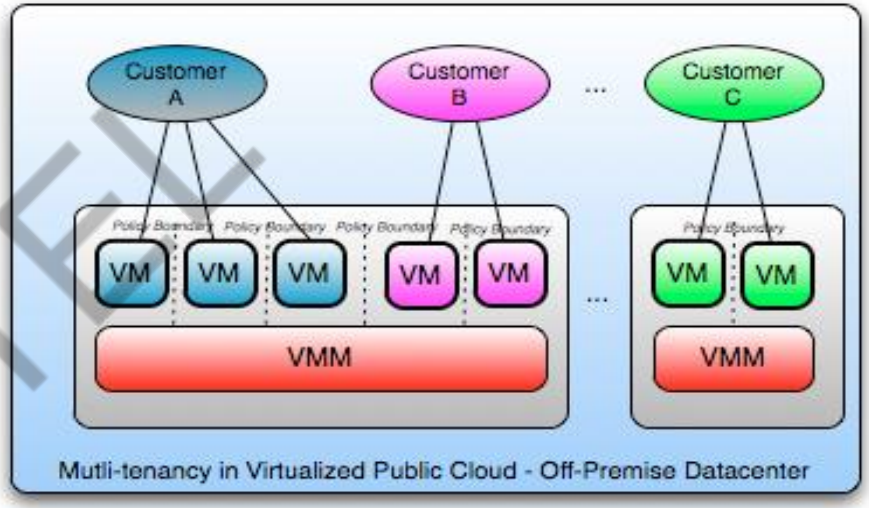
- **IaaS**: entire infrastructure from facilities to hardware
- **PaaS**: application, middleware, database, messaging supported by IaaS
  - Customer-side system administrator manages the same with provider handling platform, infrastructure security
- **SaaS**: self contained operating environment: content, presentation, apps, management
  - Service levels, security, governance, compliance, liability, expectations of the customer & provider are contractually defined



Sample Clouds



Private Cloud of Company XYZ with 3 business units, each with different security, SLA, governance and chargeback policies on shared infrastructure



Public Cloud Provider with 3 business customers, each with different security, SLA, governance and billing policies on shared infrastructure

Source: "Security Guidance for Critical Areas of Focus in Cloud Computing" v2.1, p.18

Gartner's Seven Cloud Computing Security Risks

- Gartner:
  - http://www.gartner.com/technology/about.jsp
  - Cloud computing has "unique attributes that require risk assessment in areas such as data integrity, recovery and privacy, and an evaluation of legal issues in areas such as e-discovery, regulatory compliance and auditing," Gartner says
- Security Risks
  - Privileged User Access
  - Regulatory Compliance & Audit
  - Data Location
  - Data Segregation
  - Recovery
  - Investigative Support
  - Long-term Viability

Privileged User Access

- Sensitive data processed outside the enterprise brings with it an inherent level of risk
- Outsourced services bypass the “physical, logical and personnel controls” of traditional in-house deployments.
- Get as much information as you can about the people who manage your data
- “Ask providers to supply specific information on the hiring and oversight of privileged administrators, and the controls over their access,” Gartner says.

Regulatory Compliance & Audit

- Traditional service providers are subjected to external audits and security certifications.
- Cloud computing providers who refuse to undergo this scrutiny are “signaling that customers can only use them for the most trivial functions,” according to Gartner.
- Shared infrastructure – isolation of user-specific log
- No customer-side auditing facility
- Difficult to audit data held outside organization in a cloud
  - Forensics also made difficult since now clients don’t maintain data locally
- Trusted third-party auditor?

Data Location

- Hosting of data, jurisdiction?
- Data centers: located at geographically dispersed locations
- Different jurisdiction & regulations
  - Laws for cross border data flows
- Legal implications
  - Who is responsible for complying with regulations (e.g., SOX, HIPAA, etc.)?
  - If cloud provider subcontracts to third party clouds, will the data still be secure?

Data Segregation

- Data in the cloud is typically in a shared environment alongside data from other customers.
- Encryption is effective but isn't a cure-all. "Find out what is done to segregate data at rest," Gartner advises.
- Encrypt data in transit, needs to be decrypted at the time of processing
  - Possibility of interception
- Secure key store
  - Protect encryption keys
  - Limit access to key stores
  - Key backup & recoverability
- The cloud provider should provide evidence that encryption schemes were designed and tested by experienced specialists.
- "Encryption accidents can make data totally unusable, and even normal encryption can complicate availability," Gartner says.

Recovery

- Even if you don't know where your data is, a cloud provider should tell you what will happen to your data and service in case of a disaster.
- "Any offering that does not replicate the data and application infrastructure across multiple sites is vulnerable to a total failure," Gartner says. Ask your provider if it has "the ability to do a complete restoration, and how long it will take."
- **Recovery Point Objective (RPO)**: The maximum amount of data that will be lost following an interruption or disaster.
- **Recovery Time Objective (RTO):** The period of time allowed for recovery i.e., the time that is allowed to elapse between the disaster and the activation of the secondary site.
- Backup frequency
- Fault tolerance
  - **Replication**: mirroring/sharing data over disks which are located in separate physical locations to maintain consistency
  - **Redundancy**: duplication of critical components of a system with the intention of increasing reliability of the system, usually in the case of a backup or fail-safe.

Investigative Support

- Investigating inappropriate or illegal activity may be impossible in cloud computing
- Monitoring
  - To eliminate the conflict of interest between the provider and the consumer, a neural third-party organization is the best solution to monitor performance.
- Gartner warns. “Cloud services are especially difficult to investigate, because logging and data for multiple customers may be co-located and may also be spread across an ever-changing set of hosts and data centers.”

Long-term Viability

- "Ask potential providers how you would get your data back and if it would be in a format that you could import into a replacement application," Gartner says.
- When to switch cloud providers ?
  - Contract price increase
  - Provider bankruptcy
  - Provider service shutdown
  - Decrease in service quality
  - Business dispute
- Problem: vendor lock-in

Other Cloud Security Issues…

- Virtualization
- Access Control & Identity Management
- Application Security
- Data Life Cycle Management

Virtualization

- Components:
  - Virtual machine (VM)
  - Virtual machine manager (VMM) or hypervisor
- Two types:
  - **Full virtualization**: VMs run on hypervisor that interacts with the hardware
  - **Para virtualization**: VMs interact with the host OS.
- Major functionality: resource isolation
- Hypervisor vulnerabilities:
  - Shared clipboard technology– transferring malicious programs from VMs to host

Virtualization (contd...)

- Hypervisor vulnerabilities:
  - Keystroke logging: Some VM technologies enable the logging of keystrokes and screen updates to be passed across virtual terminals in the virtual machine, writing to host files and permitting the monitoring of encrypted terminal connections inside the VM.
  - Virtual machine backdoors: covert communication channel
  - ARP Poisoning: redirect packets going to or from the other VM.
- Hypervisor Risks
  - Rogue hypervisor rootkits
    - Initiate a 'rogue' hypervisor
    - Hide itself from normal malware detection systems
    - Create a covert channel to dump unauthorized code

Virtualization (contd…)

- Hypervisor Risks
  - External modification to the hypervisor
    - Poorly protected or designed hypervisor: source of attack
    - May be subjected to direct modification by the external intruder
  - VM escape
    - Improper configuration of VM
    - Allows malicious code to completely bypass the virtual environment, and obtain full root or kernel access to the physical host
    - Some vulnerable virtual machine applications: Vmchat, VMftp, Vmcat etc.
  - Denial-of-service risk
- Threats:
  - Unauthorized access to virtual resources – loss of confidentiality, integrity, availability

Access Control & Identity Management

- Access control: similar to traditional in-house IT network
- Proper access control: to address CIA tenets of information security
- Prevention of identity theft – major challenge
  - **Privacy issues** raised via massive data mining
    - Cloud now stores data from a lot of clients, and can run data mining algorithms to get large amounts of information on clients
- Identity Management (IDM) – authenticate users and services based on credentials and characteristics

Application Security

- Cloud applications – Web service based
- Similar attacks:
  - **Injection attacks**: introduce malicious code to change the course of execution
  - **XML Signature Element Wrapping**: By this attack, the original body of an XML message is moved to a newly inserted wrapping element inside the SOAP header, and a new body is created.
  - **Cross-Site Scripting (XSS)**: XSS enables attackers to inject client-side script into Web pages viewed by other users to bypass access controls.
  - **Flooding**: Attacker sending huge amount of request to a certain service and causing denial of service.
  - **DNS poisoning and phishing**: browser-based security issues
  - **Metadata (WSDL) spoofing attacks**: Such attack involves malicious reengineering of Web Services' metadata description
- Insecure communication channel

Data Life Cycle Management

- Data security
  - Confidentiality:
    - Will the sensitive data stored on a cloud remain confidential?
    - Will cloud compromise leak confidential client data (i.e., fear of loss of control over data)
    - Will the cloud provider itself be honest and won't peek into the data?
  - Integrity:
    - How do I know that the cloud provider is doing the computations correctly?
    - How do I ensure that the cloud provider really stored my data without tampering with it?

Data Life Cycle Management (contd.)

- Availability
  - Will critical systems go down at the client, if the provider is attacked in a Denial of Service attack?
  - What happens if cloud provider goes out of business?

- Data Location
  - All copies, backups stored only at location allowed by contract, SLA and/or regulation

- Archive
- Access latency

Thank You!

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

CLOUD COMPUTING

CLOUD SECURITY III

PROF. SOUMYA K. GHOSH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IIT KHARAGPUR

Research Article

- Research Paper:
  - *Hey, You, Get Off of My Cloud! Exploring Information Leakage in Third-Party Compute Clouds*. by Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage. In Proceedings of CCS 2009, pages 199–212. ACM Press, Nov. 2009.
  - First work on *cloud cartography*
    - Attack launched against commercially available "real" cloud (Amazon EC2)
    - Claims up to 40% success in co-residence with target VM

New Risks in Cloud

- Trust and dependence
  - Establishing new trust relationship between customer and cloud provider
  - Customers must trust their cloud providers to respect the privacy of their data and integrity of their computations
- Security (multi-tenancy)
  - Threats from other customers due to the subtleties of how physical resources can be transparently shared between virtual machines (VMs)

Multi-tenancy

- Multiplexing VMs of disjoint customers upon the same physical hardware
  - Your machine is placed on the same server with other customers
  - Problem: you don't have the control to prevent your instance from being co-resident with an adversary
- New risks
  - Side-channels exploitation
    - Cross-VM information leakage due to sharing of physical resource (e.g., CPU's data caches)
    - Has the potential to extract RSA & AES secret keys
  - Vulnerable VM isolation mechanisms
    - Via a vulnerability that allows an "escape" to the hypervisor
  - Lack of control who you're sharing server space

Attack Model

- Motivation
  - To study practicality of mounting cross-VM attacks in existing third-party compute clouds
- Experiments have been carried out on real IaaS cloud service provider (Amazon EC2)
- Two steps of attack:
  - *Placement*: adversary arranging to place its malicious VM on the same physical machine as that of the target customer
  - *Extraction*: extract confidential information via side channel attack

Threat Model

- Assumptions of the threat model:
  - Provider and infrastructure to be trusted
  - Do not consider attacks that rely on subverting administrator functions
  - Do not exploit vulnerabilities of the virtual machine monitor and/or other software
  - Adversaries: non-providers-affiliated malicious parties
  - Victims: users running confidentiality-requiring services in the cloud
- Focus on new cloud-related capabilities of the attacker and implicitly expanding the attack surface

Threat Model *(contd…)*

- Like any customer, the malicious party can run and control many instances in the cloud
  - Maximum of 20 instances can be run parallel using an Amazon EC2 account
- Attacker's instance might be placed on the same physical hardware as potential victims
- Attack might manipulate shared physical resources to learn otherwise confidential information
- Two kinds of attack may take place:
  - Attack on some known hosted service
  - Attacking a particular victim's service

Addresses the Following…

- *Q1*: Can one determine where in the cloud infrastructure an instance is located?
- *Q2*: Can one easily determine if two instances are co-resident on the same physical machine?
- *Q3*: Can an adversary launch instances that will be co-resident with other user's instances?
- *Q4*: Can an adversary exploit cross-VM information leakage once co-resident?

Amazon EC2 Service

- Scalable, pay-as-you-go compute capacity in the cloud
- Customers can run different operating systems within a virtual machine
- Three degrees of freedom: *instance-type, region, availability zone*
- Different computing options (instances) available
  - m1.small, c1. medium: 32-bit architecture
  - m1.large, m1.xlarge, c1.xlarge: 64-bit architecture
- Different regions available
  - US, EU, Asia
- Regions split into availability zones
  - In US: East (Virginia), West (Oregon), West (Northern California)
  - Infrastructures with separate power and network connectivity
- Customers randomly assigned to physical machines based on their instance, region, and availability zone choices

Amazon EC2 Service *(contd…)*

- Xen hypervisor
  - Domain0 (Dom0): privileged virtual machine
    - Manages guest images
    - Provisions physical resources
    - Access control rights
    - Configured to route packets for its guest images and reports itself as a hop in traceroutes.
  - When an instance is launched, it is assigned to a single physical machine for its lifetime
- Each instance is assigned internal and external IP addresses and domain names
  - *External IP*: public IPv4 address [IP: **75.101.210.100**/domain name: **ec2-75-101-210-100.compute-1.amazonaws.com**]
  - *Internal IP*: RFC 1918 private address [IP: **10.252.146.52**/domain name: **domU-12-31-38-00-8D-C6.compute-1.internal**]
- Within the cloud, both domain names resolve to the internal IP address
- Outside the cloud, external name is mapped to the external IP address

Q1: Cloud Cartography

- Instance placing is not disclosed by Amazon but is needed to launch co-residency attack
- Map the EC2 service to understand where potential targets are located in the cloud
- Determine instance creation parameters needed to attempt establishing co-residence of an adversarial instance
- Hypothesis: *different availability zones and instance types correspond to different IP address ranges*

Network Probing

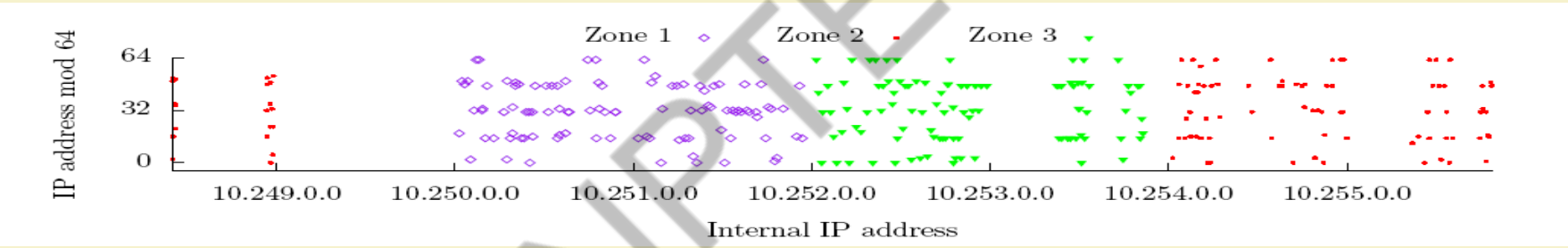
- Identify public servers hosted in EC2 and verify co-residence
- Open-source tools have been used to probe ports (80 and 443)
  - **nmap** – perform TCP connect probes (attempt to complete a 3-way hand-shake between a source and target)
  - **hping** – perform TCP SYN traceroutes, which iteratively sends TCP SYN packets with increasing TTLs, until no ACK is received
  - **wget** – used to retrieve web pages
- *External probe*: probe originating from a system outside EC2 and has an EC2 instance as destination
- *Internal probe*: originates from an EC2 instance, and has destination another EC2 instance
- Given an external IP address, DNS resolution queries are used to determine:
  - External name
  - Internal IP address

Survey Public Servers on EC2

- Goal: to enable identification of the instance type and availability zone of one or more potential targets
- *WHOIS:* used to identify distinct IP address prefixes associated with EC2
- EC2 public IPs: /17, /18, /19 prefixes
  - 57344 IP addresses
- Use external probes to find responsive IPs:
  - Performed *TCP connect probe* on port 80
    - 11315 responsive IPs
  - Followed up with *wget* on port 80
    - 9558 responsive IPs
  - Performed a *TCP scan* on port 443
    - 8375 responsive IPs
- Used DNS lookup service
  - Translate each public IP address that responded to either the port 80 or 443 scan into an internal EC2 address
  - 14054 unique internal IPs obtained

Instance Placement Parameters

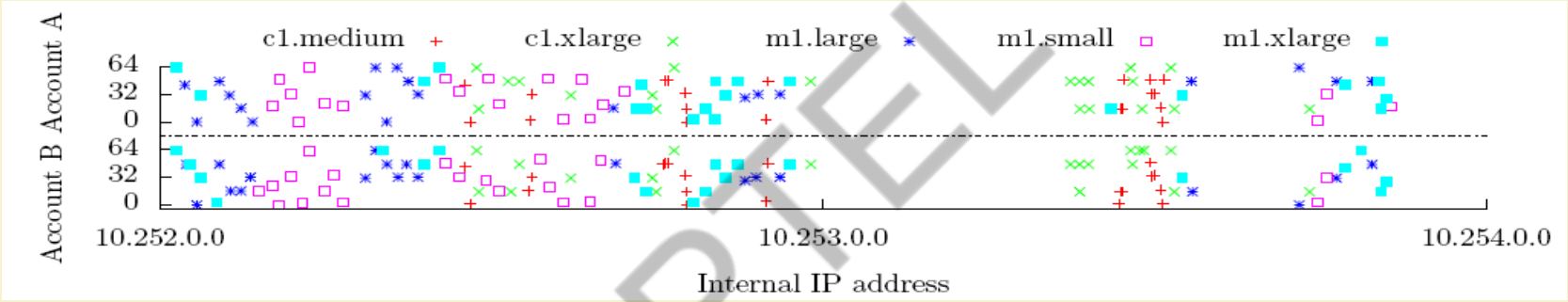
- EC2's internal address space is cleanly partitioned between availability zones
  - Three availability zone; five instance-type/zone
  - 20 instances launched for each of the 15 availability zone/instance type pairs from a particular account (Say, Account A)



- Samples from each zone are assigned IP addresses from disjoint portions of the observed internal address space
- **Assumption**: internal IP addresses are statically assigned to physical machines
  - To ease out IP routing
- Availability zones use separate physical infrastructure

Instance Placement Parameters *(contd…)*

- 100 instances have been launched in Zone 3 using two different accounts: A & B (39 hours after terminating the Account A instances)



- Of 100 Account A Zone 3 instances
  - 92 had unique /24 prefixes
  - Four /24 prefixes had two instances each
- Of 100 Account B Zone 3 instances
  - 88 had unique /24 prefixes
  - Six of the /24 prefixes had two instances each
- A single /24 had both an m1.large and m1.xlarge instance
- Of 100 Account B IP's, 55 were repeats of IP addresses assigned to instances for Account A

Q2: Determining Co-residence

- Network-based co-residency checks: instances are likely to be co-resident if they have-
  - **Matching Dom0 IP address**: determine an uncontrolled instance's Dom0 IP by performing a *TCP SYN* traceroute to it from another instance and inspect the last hop
  - **Small packet round-trip times**: 10 probes were performed and the average is taken
  - **Numerically close internal IP addresses (e.g., within 7)**: the same Dom0 IP will be shared by instances with contiguous sequence of internal IP addresses

Verifying Co-residency Check

- If two (under self-control) instances can successfully transmit via the covert channel, then they are co-resident, otherwise not
- Experiment: hard-disk-based covert channel
  - To send a 1, sender reads from random locations on a shared volume, to send a 0 sender does nothing
  - Receiver times reading from a fixed location on the disk: longer read times mean a 1 is set, shorter a 0
- 3 m1.small EC2 accounts: *control, victim, probe*
  - 2 control instances in each of 3 availability zones, 20 victim and 20 probe instances in Zone 3
- Determine *Dom0* address for each instance
- For each ordered pair (A, B) of 40 instances, perform co-residency checks
- After 3 independent trials, 31 (potentially) co-resident pairs have been identified - 62 ordered pairs
- *5 bit* message from A to B was successfully sent for 60 out of 62 ordered pairs

Effective Co-residency Check

- For checking co-residence with target instances:
  - Compare internal IP addresses to see if they are close
  - If yes, perform a TCP SYN traceroute to an open port on the target and see if there is only a single hop (Dom0 IP)
    - Check requires sending (at most) two *TCP SYN* packets
      - No full TCP connection is established
    - Very "quiet" check (little communication with the victim)

Q3: Causing Co-residence

- Two strategies to achieve “good” coverage (co-residence with a good fraction of target set)
  - Brute-force placement:
    - run numerous *probe* instances over a long period of time and see how many targets one can achieve co-residence with.
    - For co-residency check, the probe performed a wget on port 80 to ensure the target was still serving web pages
    - Of the 1686 target victims, the brute-force probes achieved co-residency with 141 victim servers (8.4% coverage)
    - Even a naïve strategy can successfully achieve co-residence against a not-so-small fraction of targets
  - Target recently launched instances:
    - take advantage of the tendency of EC2 to assign fresh instances to small set of machines

Leveraging Placement Locality

- Placement locality
  - Instances launched simultaneously from same account do not run on the same physical machine
  - *Sequential placement locality*: exists when two instances run sequentially (the first terminated before launching the second) are often assigned to the same machine
  - *Parallel placement locality*: exists when two instances run (from distinct accounts) at roughly the same time are often assigned to the same machine.
- *Instance flooding*: launch lots of instances in parallel in the appropriate availability zone and of the appropriate type

Leveraging Placement Locality (contd...)

- Experiment
  - Single victim instance is launched
  - Attacker launches 20 instances within 5 minutes
  - Perform co-residence check
  - 40% of the time the attacker launching just 20 probes achieves co-residence against a specific target instance

Q4: Exploiting Co-residence

- Cross-VM attacks can allow for information leakage
- How can we exploit the shared infrastructure?
  - Gain information about the resource usage of other instances
  - Create and use covert channels to intentionally leak information from one instance to another
  - Some applications of this covert channel are:
    - Co-residence detection
    - Surreptitious detection of the rate of web traffic a co-resident site receives
    - Timing keystrokes by an honest user of a co-resident instance

Exploiting Co-residence (contd…)

- Measuring cache usage
  - Time-shared cache allows an attacker to measure when other instances are experiencing computational load
  - Load measurement: allocate a contiguous buffer B of *b* bytes, *s* is cache line size (in bytes)
    - *Prime*: read B at *s*-byte offsets in order to ensure that it is cached.
    - *Trigger*: busy-loop until CPU's cycle counter jumps by a large value
    - *Probe*: measure the time it takes to again read B at *s*-byte offset
  - Cache-based covert channel:
    - Sender idles to transmit a 0 and frantically accesses memory to transmit a 1
    - Receiver accesses a memory block and observes the access latencies
    - High latencies are indicative that "1" is transmitted

Exploiting Co-residence (contd...)

- Load-based co-residence check
  - Co-residence check can be done without network- base technique
  - Adversary can actively cause load variation due to a publicly-accessible service running on the target
  - Use a priori knowledge about load variation
  - Induce computational load (lots of HTTP requests) and observe the differences in load samples

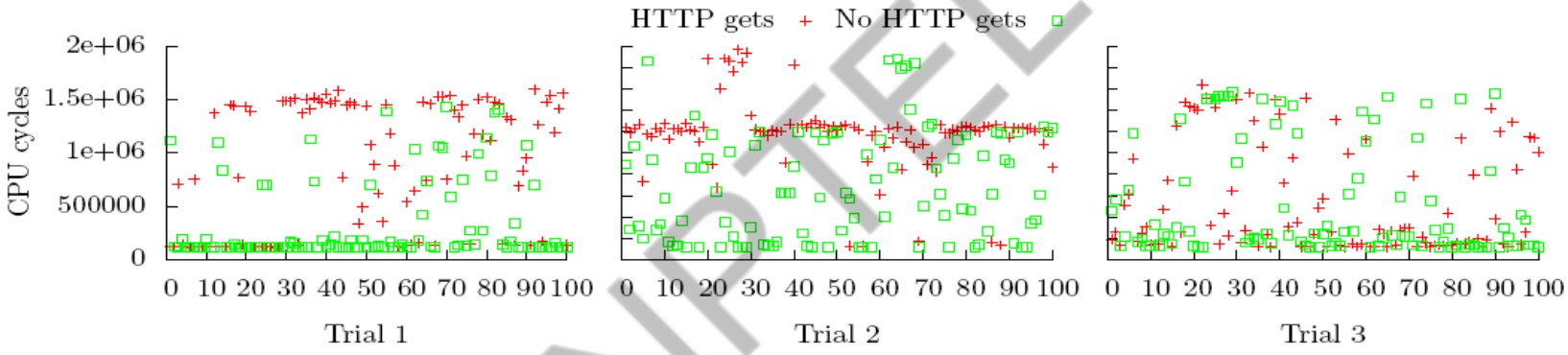


Figure 5: Results of executing 100 Prime+Trigger+Probe cache timing measurements for three pairs of m1.small instances, both when concurrently making HTTP get requests and when not. Instances in Trial 1 and Trial 2 were co-resident on distinct physical machines. Instances in Trial 3 were not co-resident.

- Instances in Trial 1 and Trial 2 were co-resident on distinct physical machines; instances in Trial 3 were not co-resident

Exploiting Co-residence (contd...)

- Estimating traffic rates
  - Load measurement might provide a method for estimating the number of visitors to a co-resident web server
  - It might not be a public information and could be damaging
  - Perform 1000 cache load measurements in which
    - no HTTP requests are sent
    - HTTP requests sent at a rate of (i) 50 per minute, (ii) 100 per minute, (iii) 200 per minutes

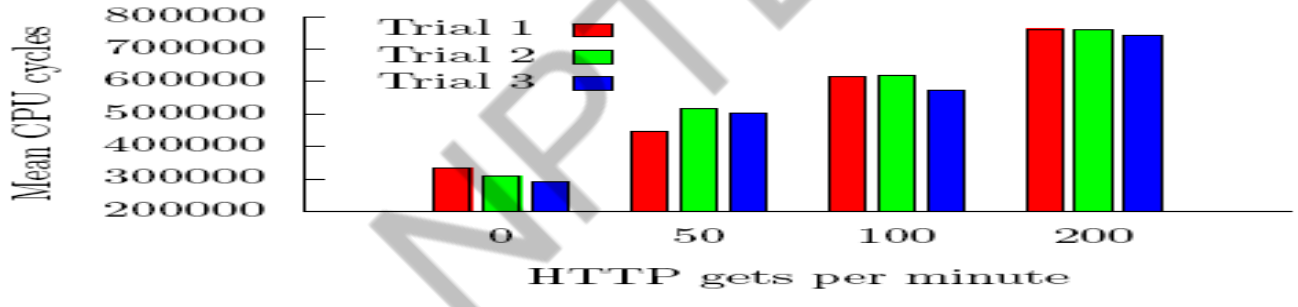


Figure 6: Mean cache load measurement timings (over 1 000 samples) taken while differing rates of web requests were made to a 3 megabyte text file hosted by a co-resident web server.

Exploiting Co-residence (contd...)

- Keystroke timing attack
  - The goal is to measure the time between keystrokes made by a victim typing a password (or other sensitive information)
  - Malicious VM can observe keystroke timing in real time via cache-based load measurements
  - Inter-keystroke times if properly measures can be used to perform recovery of the password
  - In an otherwise idle machine, a spike in load corresponds to a letter being typed into the co-resident VM's terminal
  - Attacker does not directly learn exactly which keys are pressed, the attained timing resolution suffices to conduct the password-recovery attacks on SSH sessions

Preventive Measures

- Mapping
  - Use a randomized scheme to allocate IP addresses
  - Block some tools (nmap, traceroute)
- Co-residence checks
  - Prevent identification of Dom0
- Co-location
  - Not allow co-residence at all
    - Beneficial for cloud user
    - Not efficient for cloud provider
- Information leakage via side-channel
  - No solution

Summary

- New risks from cloud computing
- Shared physical infrastructure may and most likely will cause problems
  - Exploiting software vulnerabilities not addressed here
- Practical attack performed
- Some countermeasures proposed

Thank You!

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

CLOUD COMPUTING

CLOUD SECURITY IV

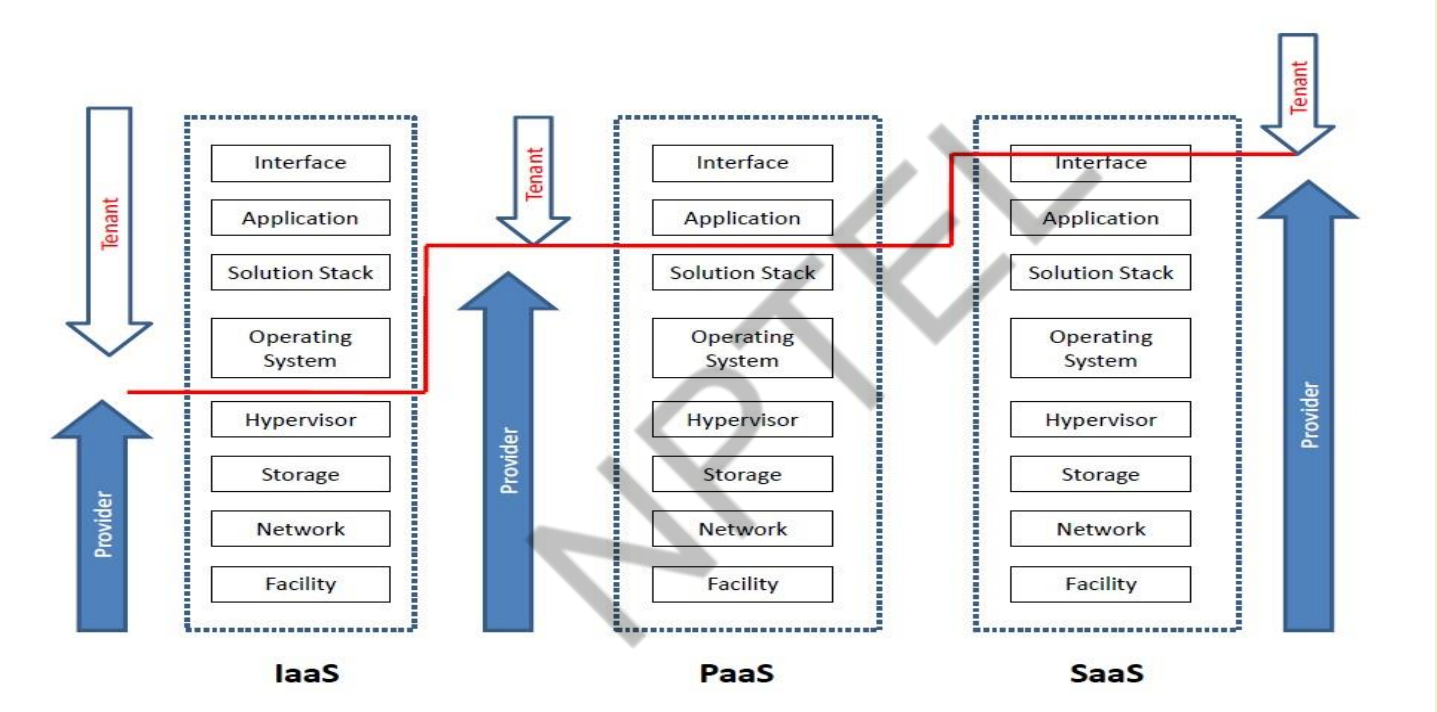
Security Issues in Collaborative SaaS Cloud

PROF. SOUMYA K. GHOSH
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
IIT KHARAGPUR

Security Issues in Cloud Computing

- Unique security features:
  - Co-tenancy
  - Lack of control on outsourced data and application
- General concerns among cloud customers [Liu'11]:
  - Inadequate policies and practices
  - Insufficient security controls
- Customers use cloud services to serve their clients
- Need to establish trust relationships
- Beneficial to both stakeholders

Security Responsibilities



SaaS Cloud-based Collaboration

- APIs for sharing resources/information
  - Service consumer(customers): human users, applications, organizations/domains, etc.
  - Service provider: SaaS cloud vendor
- SaaS cloud-centric collaboration: valuable and essential
  - Data sharing
  - Problems handled: inter-disciplinary approach
- Common concerns:
  - Integrity of data, shared across multiple users, may be compromised
  - Choosing an “ideal” vendor

SaaS Cloud-based Collaboration

- Types of collaboration in multi-domain/cloud systems:
  - Tightly-coupled or federated
  - Loosely-coupled
- Challenges: securing loosely-coupled collaborations in cloud environment
  - Security mechanisms: mainly proposed for tightly-coupled systems
  - Restrictions in the existing authentication/authorization mechanisms in clouds

Motivations and Challenges

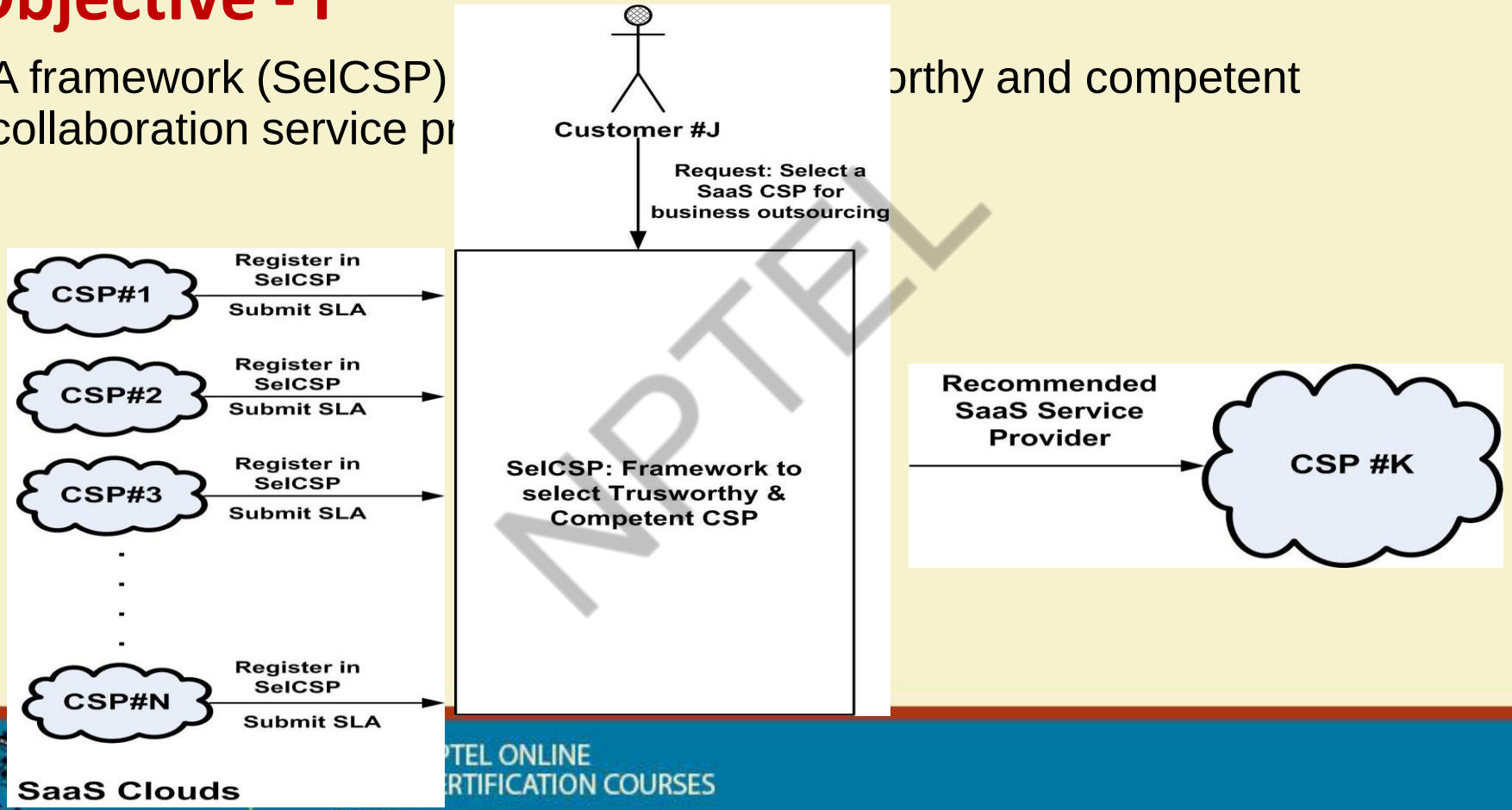
- SaaS cloud delivery model: maximum lack of control
- No active data streams/audit trails/outage report
  - **Security**: Major concern in the usage of cloud services
- Broad scope: *address security issues in SaaS clouds*
- Cloud marketplace: rapid growth due to recent advancements
- Availability of multiple service providers
  - Choosing SPs from SLA guarantees: not reliable
    - Inconsistency in service level guarantees
    - Non-standard clauses and technical specifications
- Focus: *selecting an "ideal" SaaS cloud provider and address the security issues*

Motivations and Challenges

- Online collaboration: popular
- Security issue: unauthorized disclosure of sensitive information
  - Focus: *selecting an ideal SaaS cloud provider and secure the collaboration service offered by it*
- Relevance in today's context: *loosely-coupled collaboration*
  - Dynamic data/information sharing
- Final goal (problem statement): *selecting an ideal SaaS cloud provider and securing the loosely-coupled collaboration in its environment*

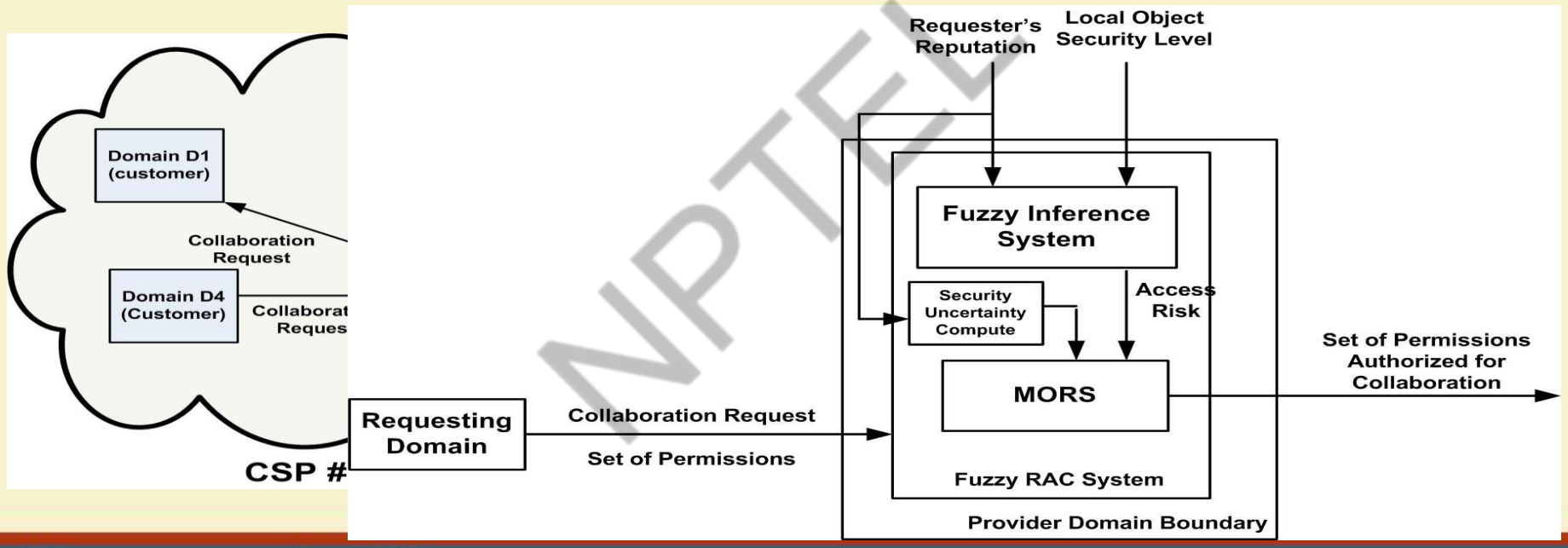
Objective - I

A framework (SelCSP) orthy and competent collaboration service pr



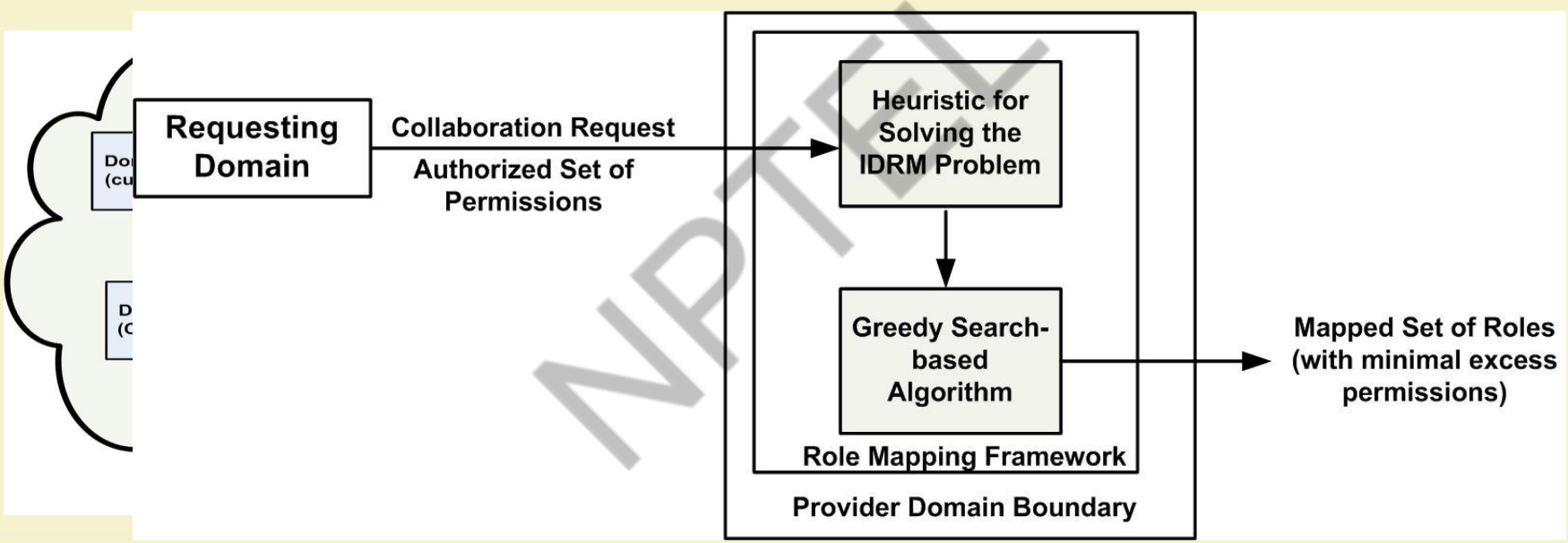
Objective - II

Select requests (for accessing local resources) from anonymous users, such that both access risk and security uncertainty due to information sharing are kept low.



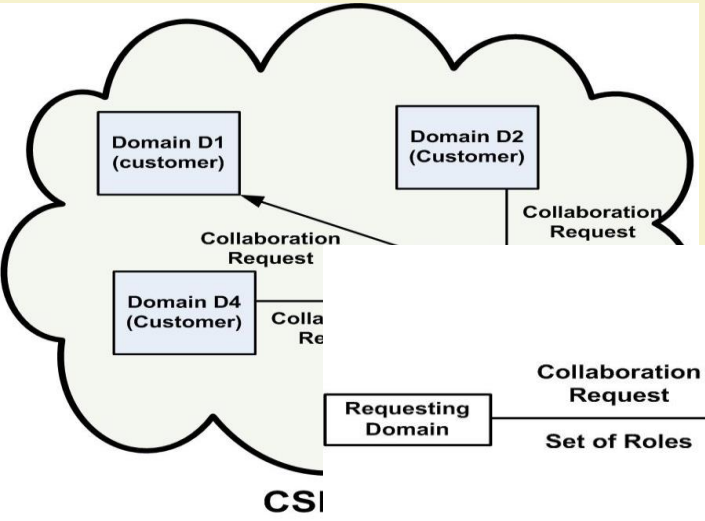
Objective - III

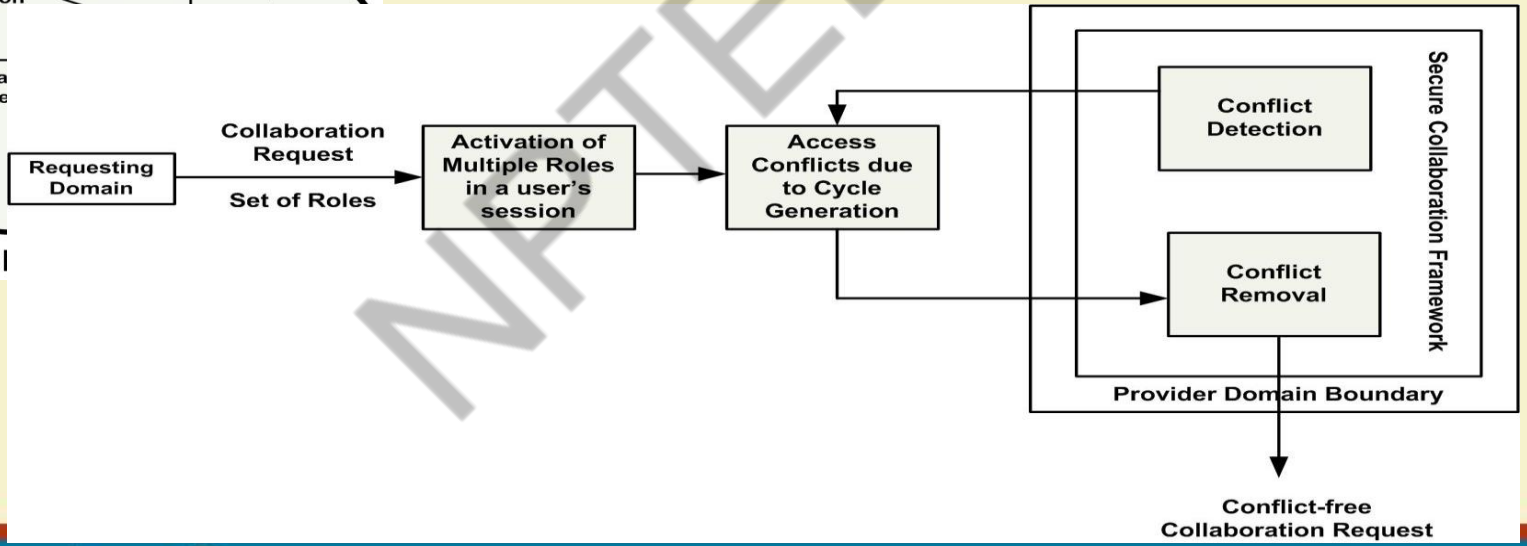
Formulate a heuristic for solving the IDRM problem, such that minimal excess privilege is granted



Objective - IV

A distributed secure collaboration framework, which uses only local information to dynamically detect and remove access conflicts.





Selection of Trustworthy and Competent SaaS Cloud Provider for Collaboration

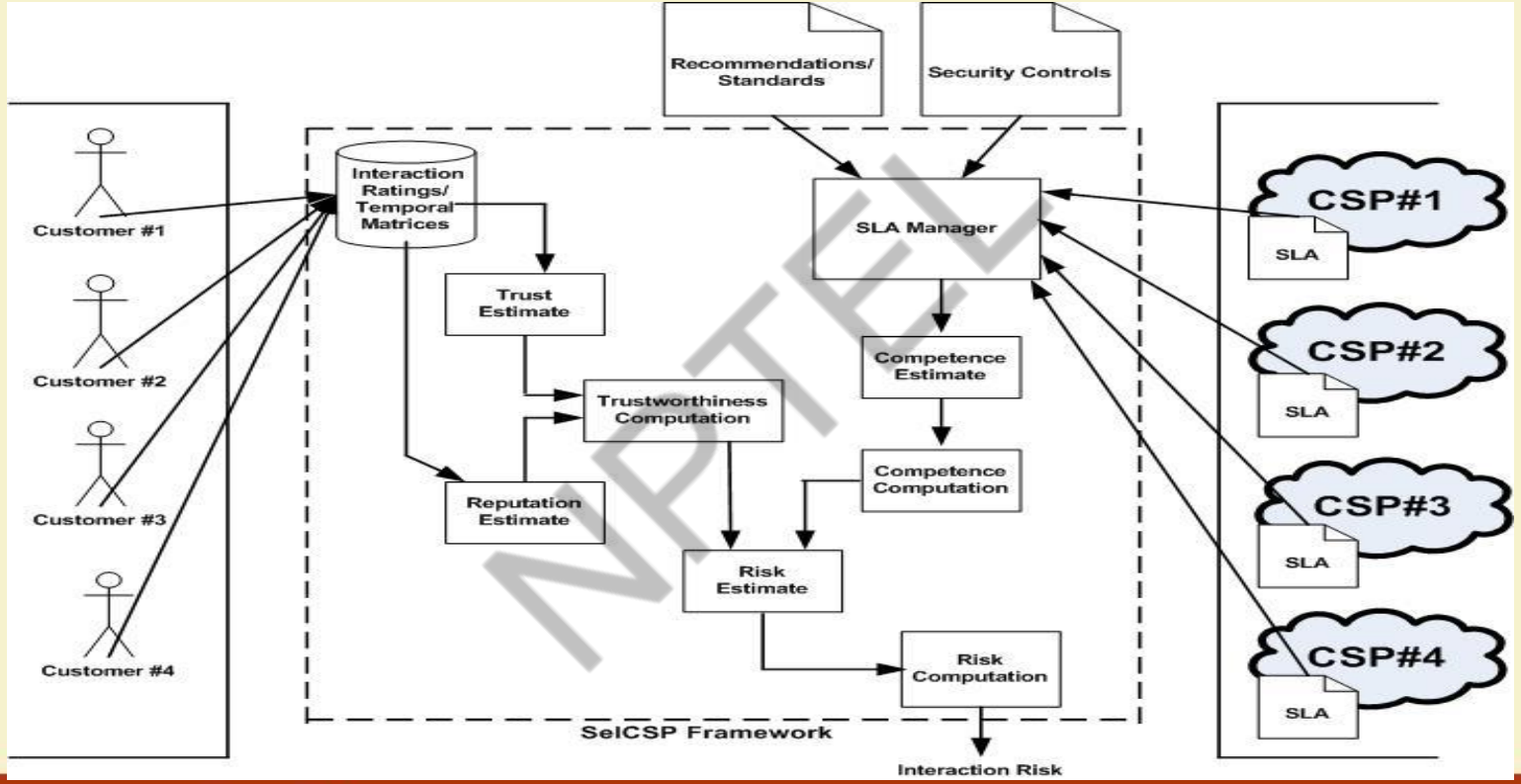
Trust Models in Cloud

- Challenges
  - Most of the reported works have not presented mathematical formulation or validation of their trust and risk models
  - Web service selection [Liu'04][Garg'13] based on QoS and trust are available
    - Select resources (e.g. services, products, etc.) by modeling their performance
- **Objective: Model trust/reputation/competence of service provider**

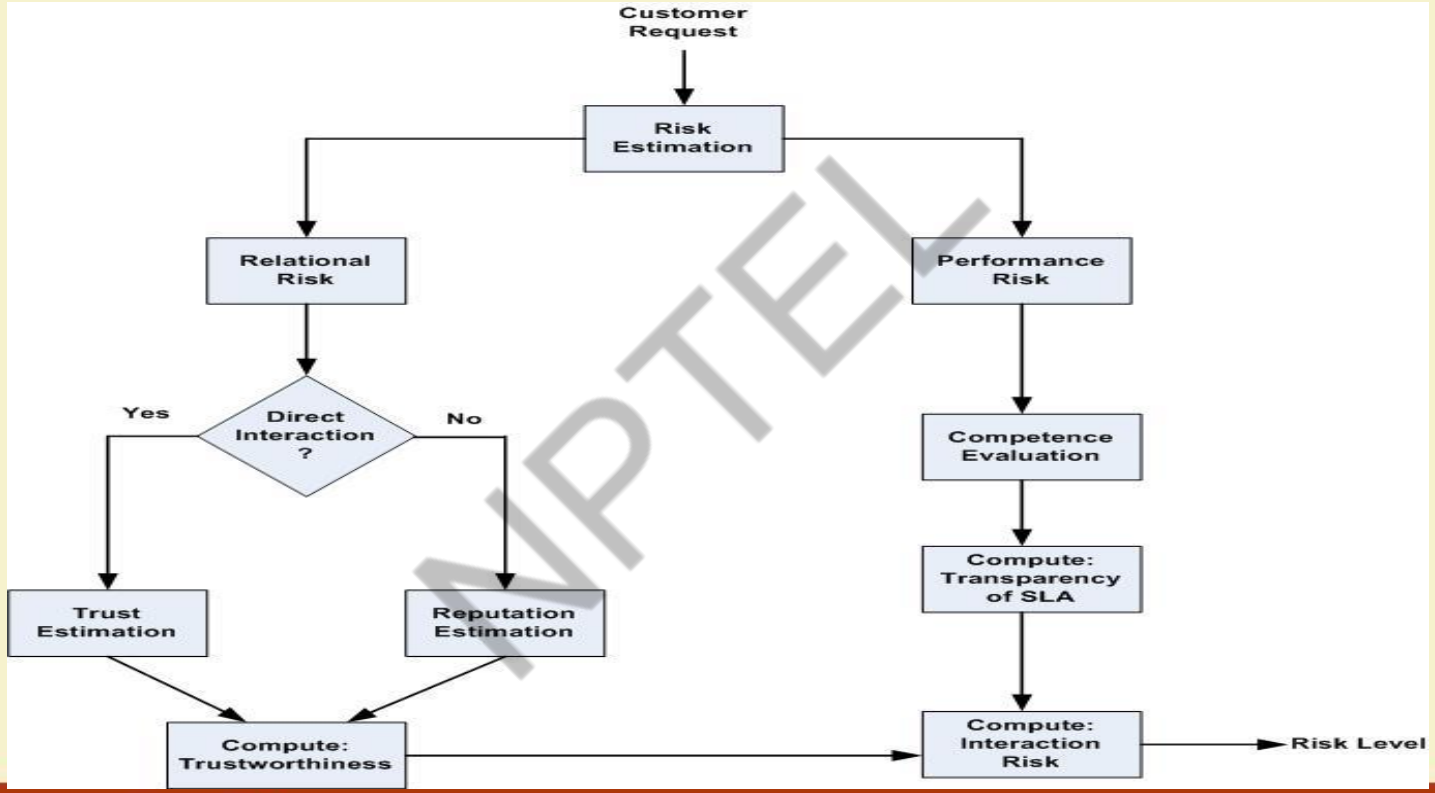
Service Level Agreement (SLA) for Clouds

- Challenges:
  - Majority of the cloud providers guarantee “availability” of services
  - Consumers not only demand availability guarantee but also other performance related assurances which are equally business critical
  - Present day cloud SLAs contain non-standard clauses regarding assurances and compensations following a violation[Habib’11]
- Objective: **Establish a standard set of parameters for cloud SLAs, since it reduces the perception of risk in outsourced services**

SelCSP Framework



SelCSP Framework - Overview

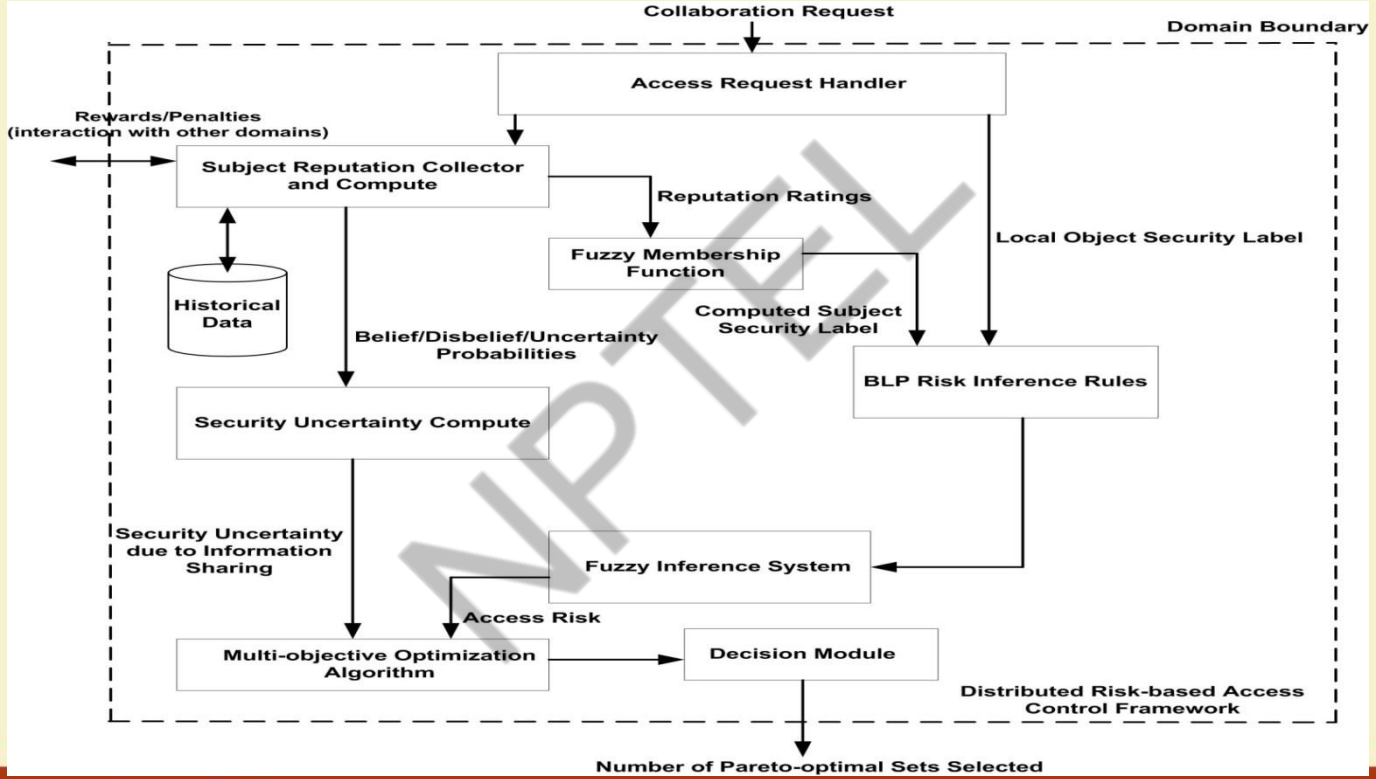


Recommending Access Requests from Anonymous Users for Authorization

Risk-based Access Control (RAC)

- RAC: Gives access to subjects even though they lack proper permissions
  - Goal: balance between *access risk* and *security uncertainty due to information sharing*
  - Flexible compared to binary MLS
- Challenges
  - Computing security uncertainty: not addressed
  - Authorization in existing RAC system: based on risk threshold and operational need.
    - Operational need: not quantified.
    - Discards many requests which potentially maximizes information sharing

Distributed RAC using Fuzzy Inference System

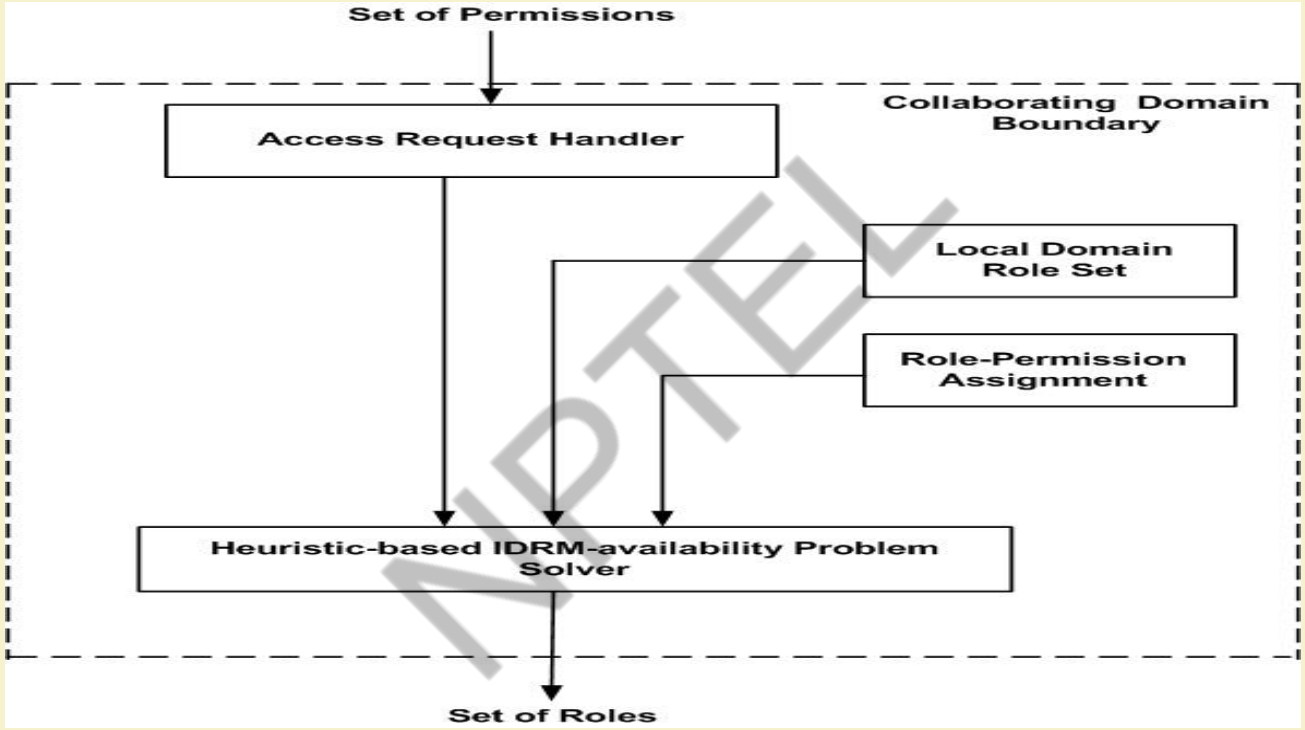


Mapping of Authorized Permissions into Local Roles

Inter-Domain Role Mapping (IDRM)

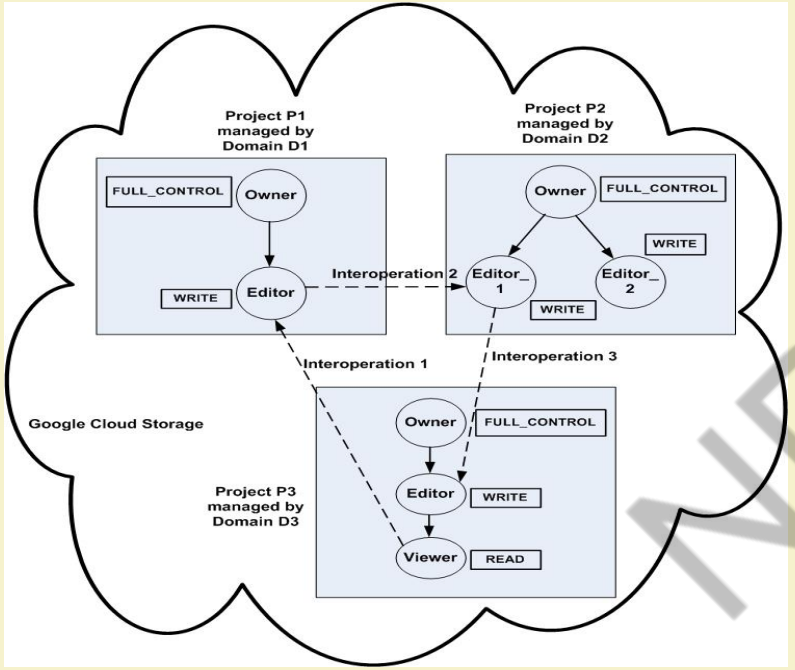
- Finds a minimal set of role which encompasses the requested permission set.
  - No polynomial time solution
  - Greedy search-based heuristics: suboptimal solutions
- Challenges:
  - There may exist multiple minimal role sets
  - There may not exist any role set which exactly maps all permissions
- Two variants of IDRM proposed: *IDRM-safety*, *IDRM-availability*
- Objective: formulate a novel heuristic to generate better solution for the IDRM-availability problem.
- Minimize the number of additional permissions

Distributed Role Mapping Framework

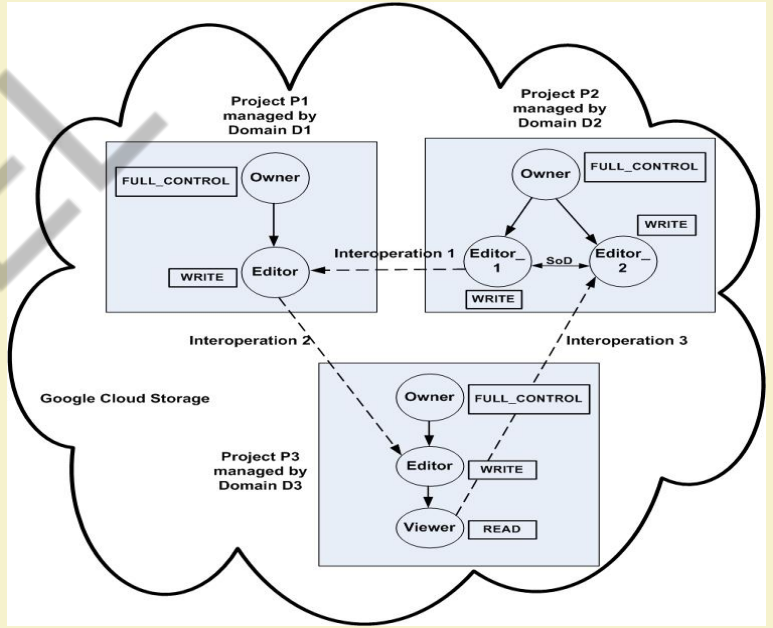


Dynamic Detection and Removal of Access Policy Conflicts

Access Conflicts



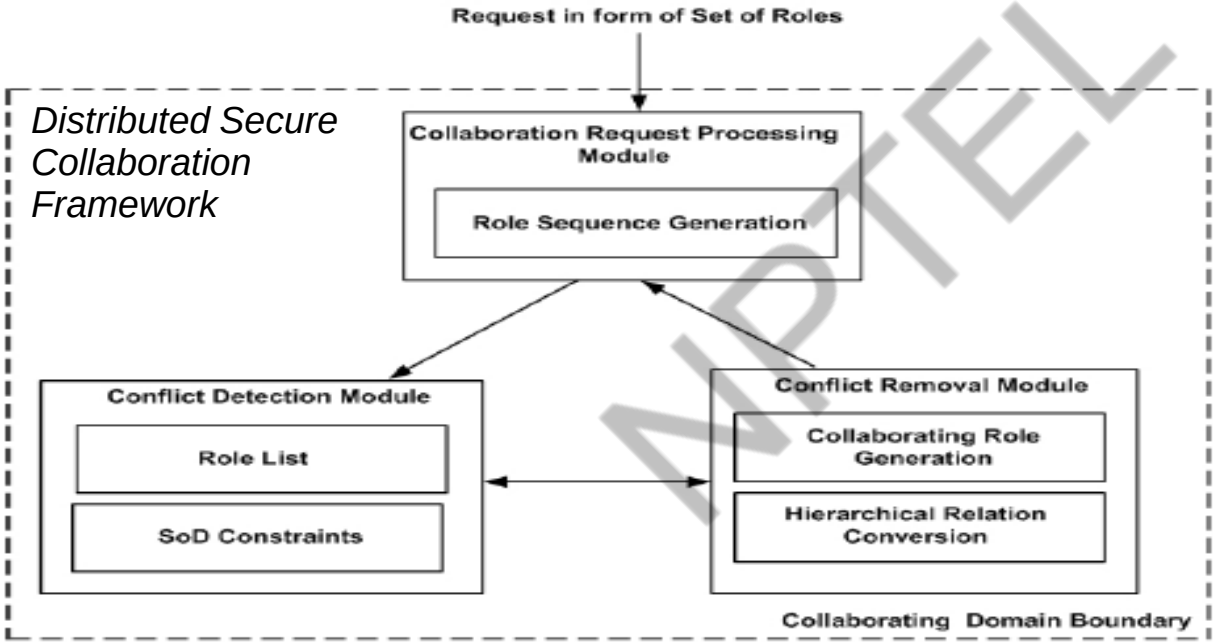
Cyclic Inheritance Conflict



Violation of SoD Constraint

Objective

- Dynamic detection of conflicts to address **security** issue
- Removal of conflicts to address **availability** issue
- Proposed: distributed secure collaboration framework



- Role Sequence Generation
  - Interoperation request: pair of *entry* (from requesting domain), *exit* (from providing domain) roles
  - Role sequence: ordered succession of entry and exit roles
  - Role cycle:
    - Safe role cycle
    - **Unsafe role cycle**

Conflict Detection

- Detection of inheritance conflict
  - Necessary condition: at least one exit role
  - Sufficient condition: current entry role is senior to at least one exit role
- Detection of SoD constraint violation
  - Necessary condition: at least one exit role
  - Sufficient condition: current entry role and at least one exit role forms *conflicting pair*

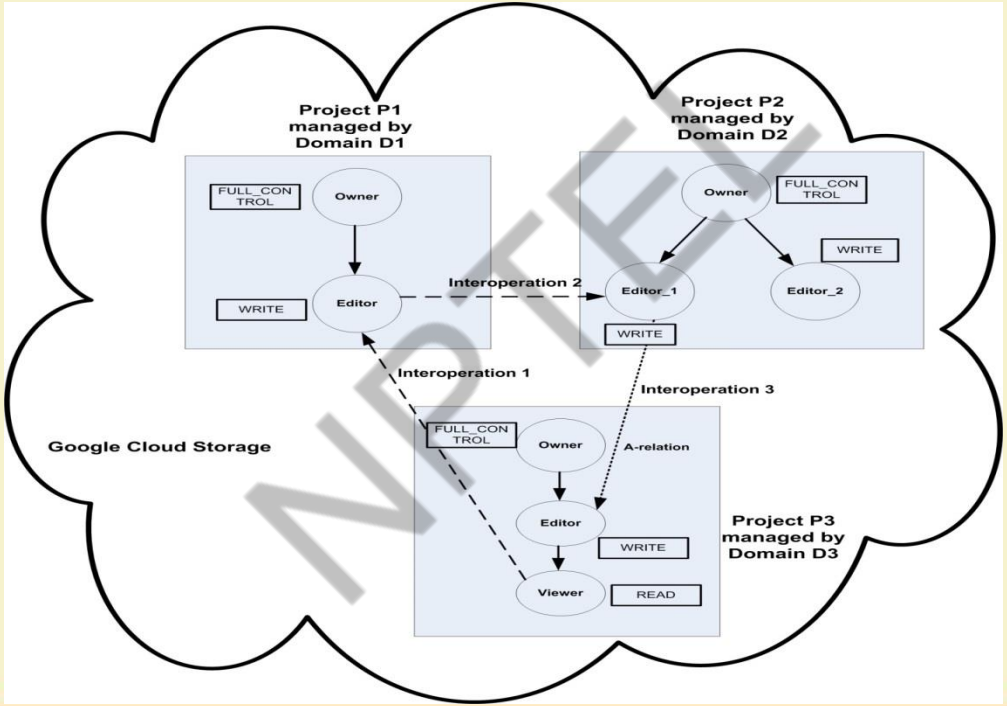
Conflict Removal

Cyclic Inheritance

- Two cases arise:
  - Exactly matched role set exists
    - RBAC hybrid hierarchy
      - *I-hierarchy, A-hierarchy, IA-hierarchy*
    - Replacing *IA-relation* with *A-relation* between exit role in previous domain and entry role in current domain
  - No-exactly matched role set exists
    - Introduce a virtual role

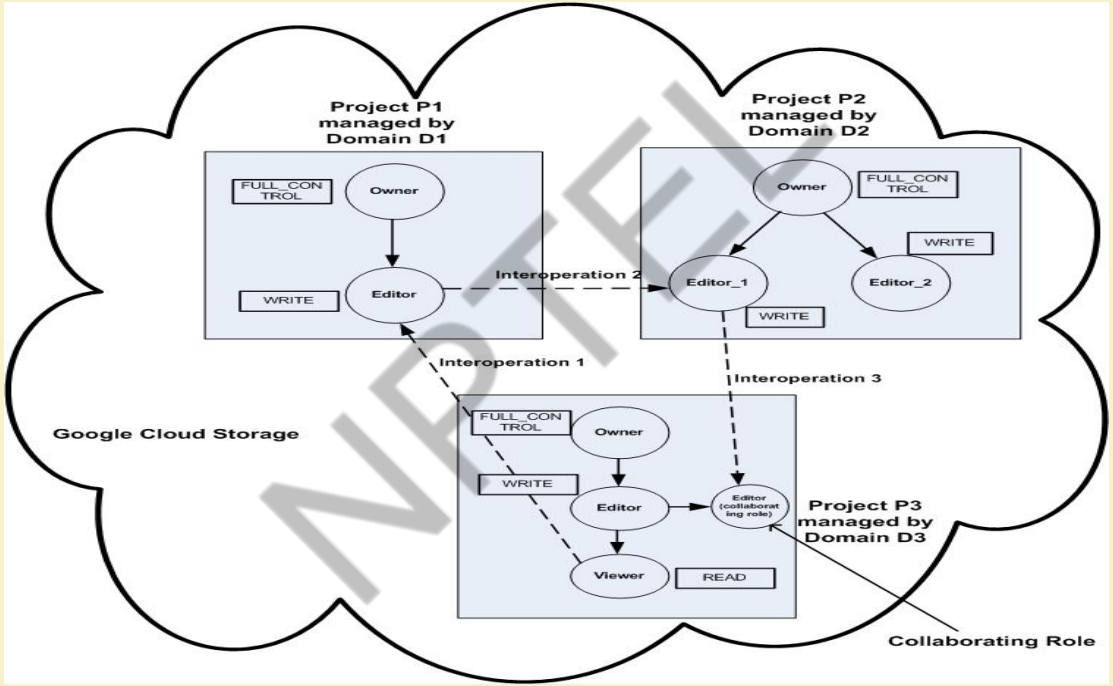
Conflict Removal

Cyclic Inheritance: Inheritance Conflict Removal Rule for Exactly Matched Role



Conflict Removal

Cyclic Inheritance: Inheritance Conflict Removal Rule for No-Exactly Matched Role



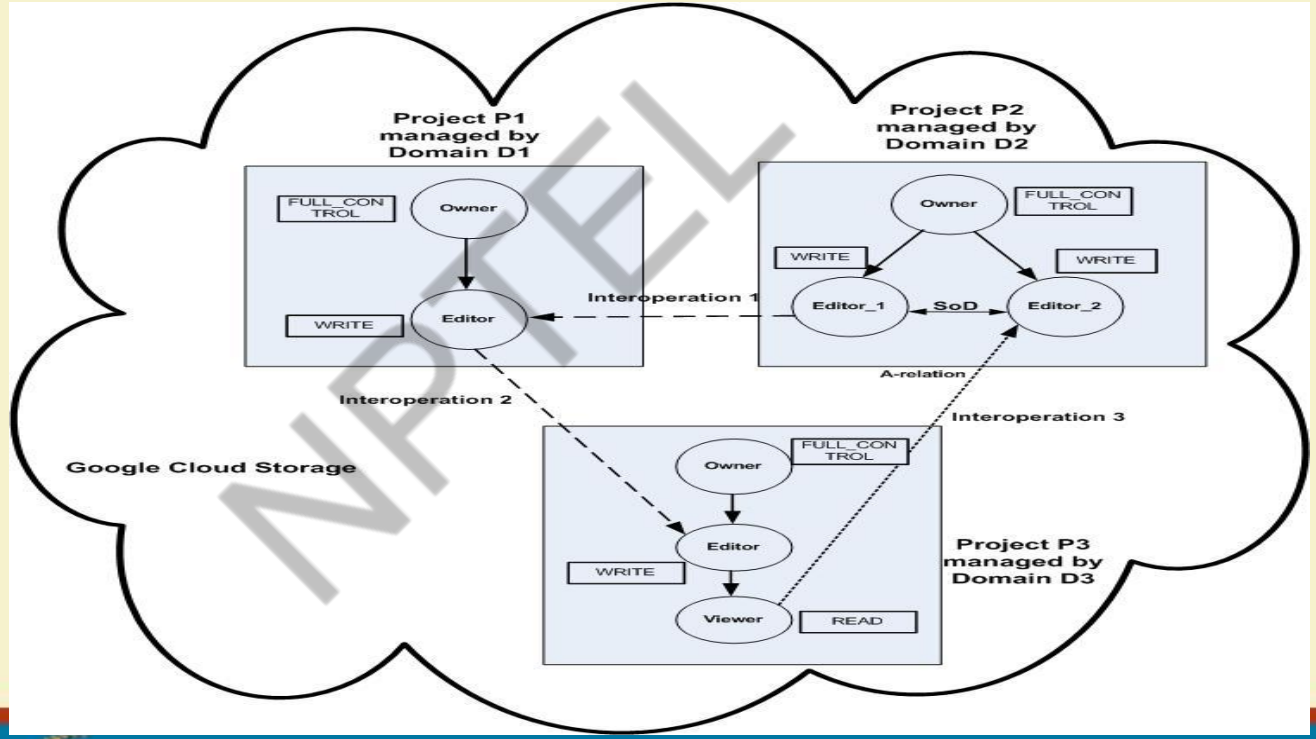
Conflict Removal

SoD Constraint Violation

- Two cases: similar to removal of inheritance conflict
  - Additional constraint: identifying *conflicting permission* between collaborating role and entry role in current domain
  - Conflicting permission
    - Objects are similar
    - Hierarchical relation exists between access modes
- Remove conflicting permission from permission set of collaborating role

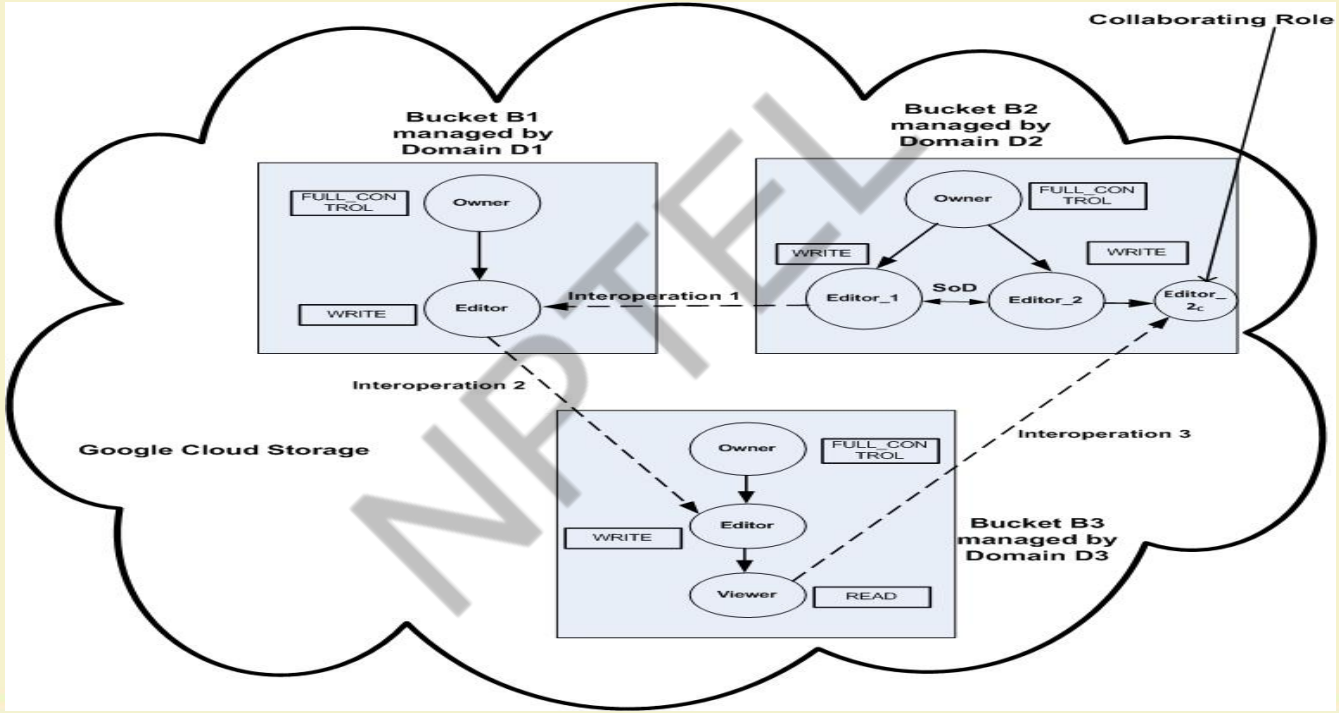
Conflict Removal

SoD Constraint Violation: SoD Conflict Removal Rule for Exactly Matched Role



Conflict Removal

SoD Constraint Violation: SoD Conflict Removal Rule for No-Exactly Matched Role



Summary

Secure Collaboration SaaS Clouds: A Typical Approach

- Selection of Trustworthy and Competent SaaS Cloud Provider for Collaboration
- Recommending Access Requests from Anonymous Users for Authorization
- Mapping of Authorized Permissions into Local Roles
- Dynamic Detection and Removal of Access Policy Conflicts

Thank You!

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

Cloud Computing :
Broker for Cloud Marketplace

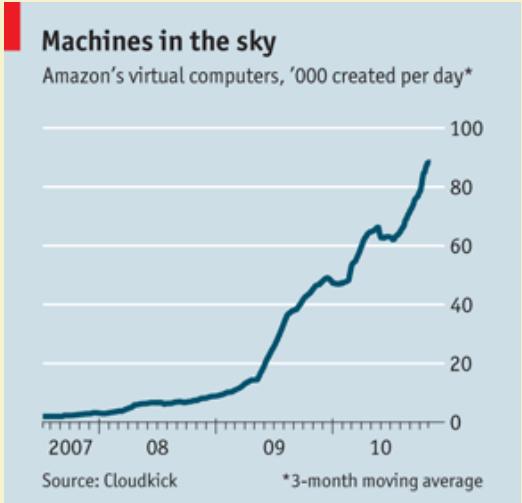
Prof. Soumya K Ghosh

Department of Computer Science and Engineering

IIT KHARAGPUR

INTRODUCTION

- Rapid growth of available cloud services
- Huge number of providers with varying QoS
- Different types of customer use cases – each with different requirements



INTRODUCTION

- Rapid growth of available cloud services
- Huge number of providers with varying QoS
- Different types of customer use cases – each with different requirements

- *Need for a "middle man" (Intelligent Broker!) to*
  - Suggest the best cloud provider to the customer
  - Safeguard the interests of the customer

MOTIVATION

- Flexible selection of cloud provider
- Trustworthiness of provider
- Monitoring of services
- Avoiding vendor lock-in

OBJECTIVES

- Selection of the most suitable provider satisfying customer's QoS requirements
- Calculation of the degree of SLA satisfaction and trustworthiness of a provider
- Decision making system for dynamic service migration based on experienced QoS

Different Approaches

- CloudCmp: a tool that compares cloud providers in order to measure the QoS they offer and helps users to select a cloud.
- Fuzzy provider selection mechanism.
- Framework with a measure of satisfaction with a provider for keeping in mind the fuzzy nature of the user requirements.
- Provider selection framework which takes into account the trustworthiness and competence of a provider.

CUSTOMER QoS PARAMETERS

Infrastructure-as-a-Service

Software-as-a-Service

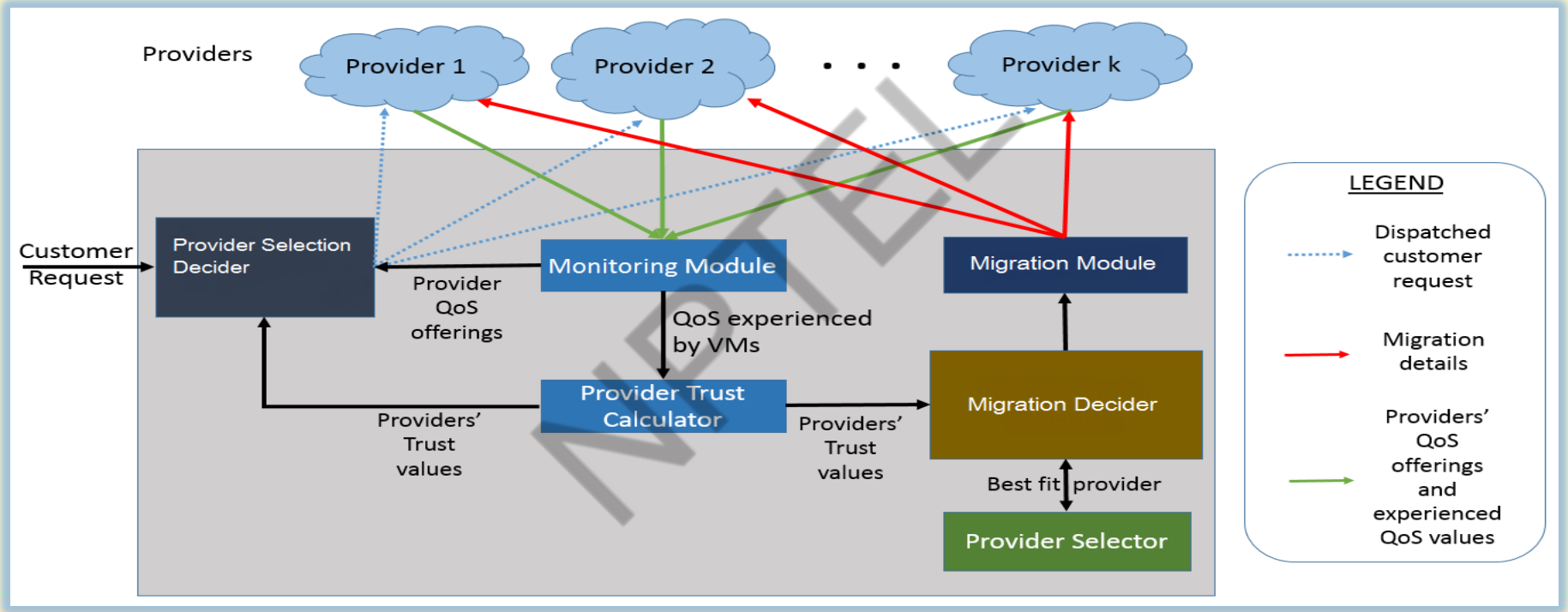
- *More QoS parameter can be added easily.*

PROVIDER

- Promised QoS values : $Prom_i{}^1, Prom_i{}^2, \ldots, Prom_i{}^L$
- Trust values : $TRUST_i{}^1, TRUST_i{}^2, \ldots, TRUST_i{}^L$

Note: They have been kept independent as they pertain to different parameters

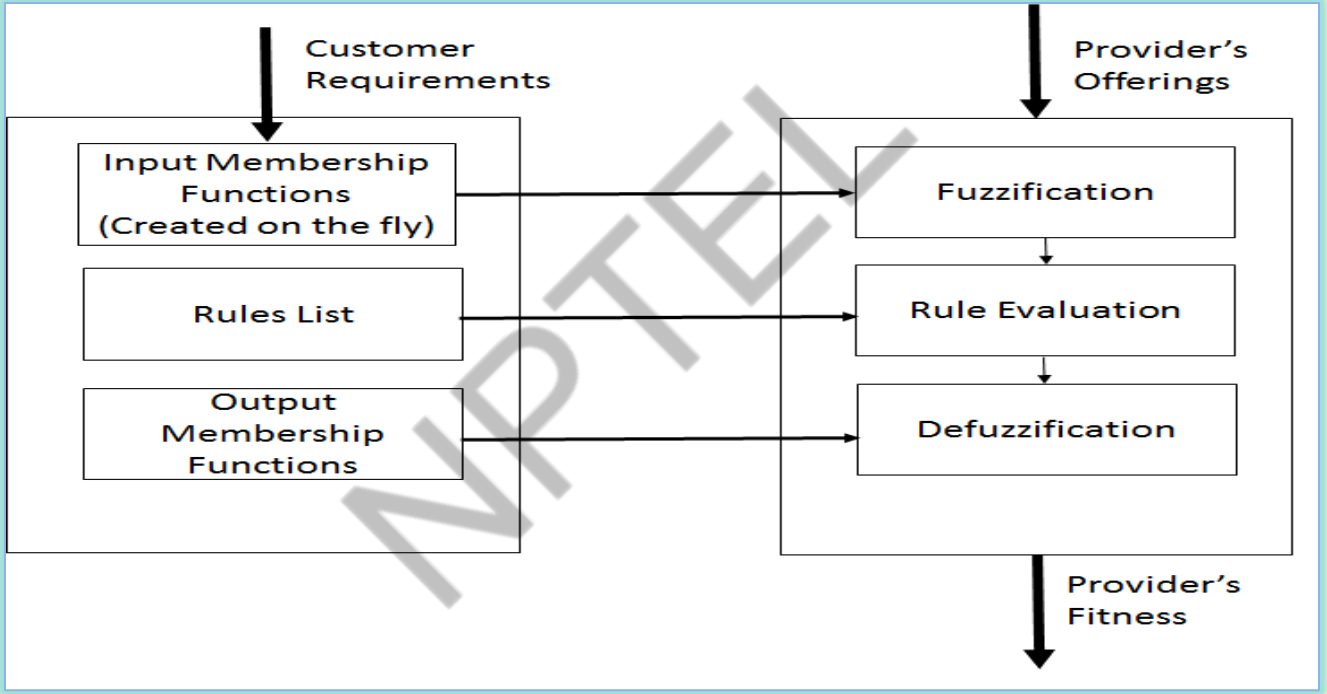
Typical MARKETPLACE Architecture



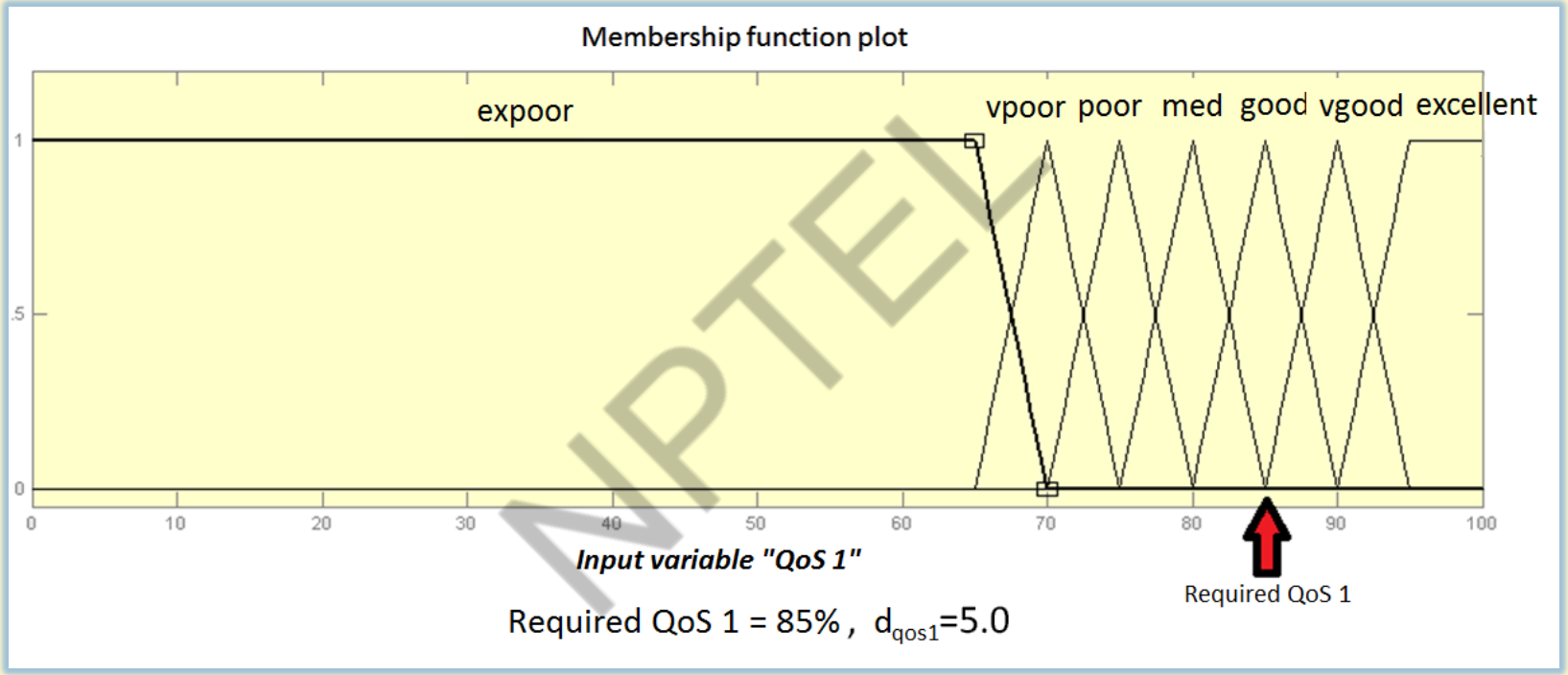
PROVIDER SELECTION

- Selection of provider is done using a fuzzy inference engine
- Input : QoS offered by a provider and its trustworthiness
- Output : Suitability of the provider for the customer
- Customer request is dispatched to provider with maximum suitability
- Membership functions are built using the user requirements

PROVIDER SELECTION

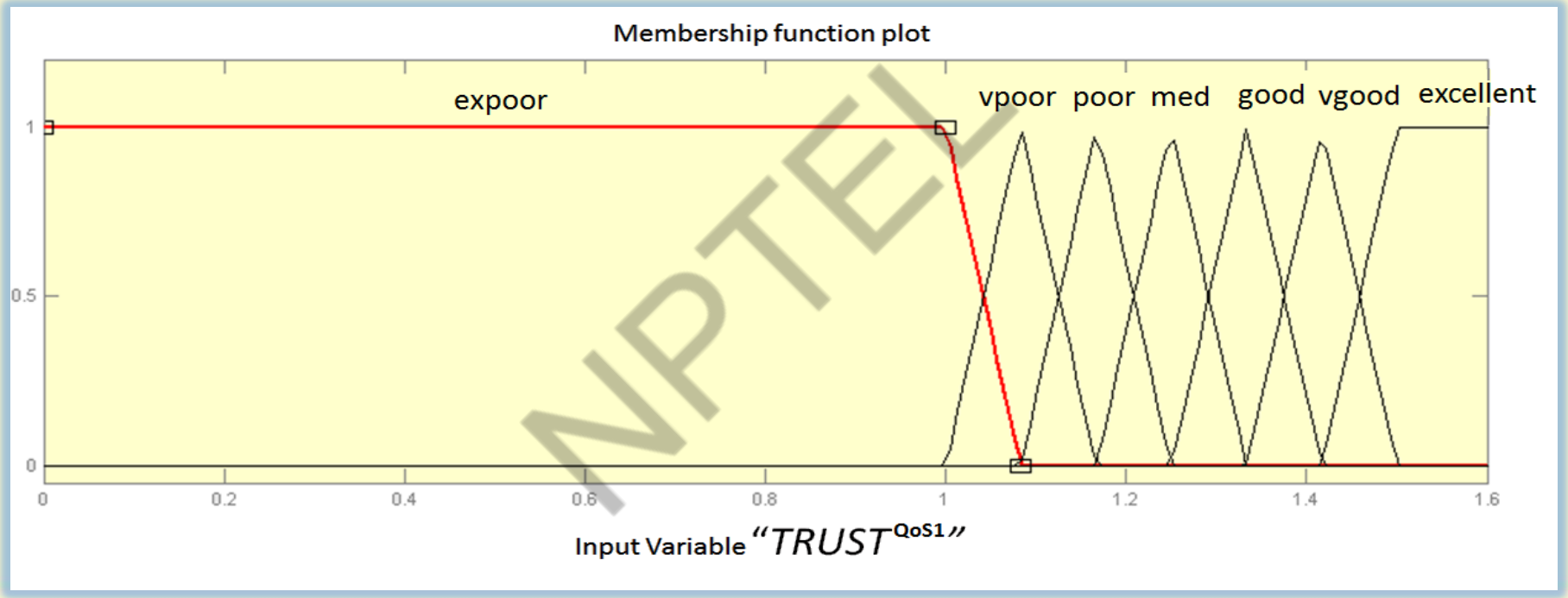


PROVIDER SELECTION – INPUT MEMBERSHIP FUNCTION

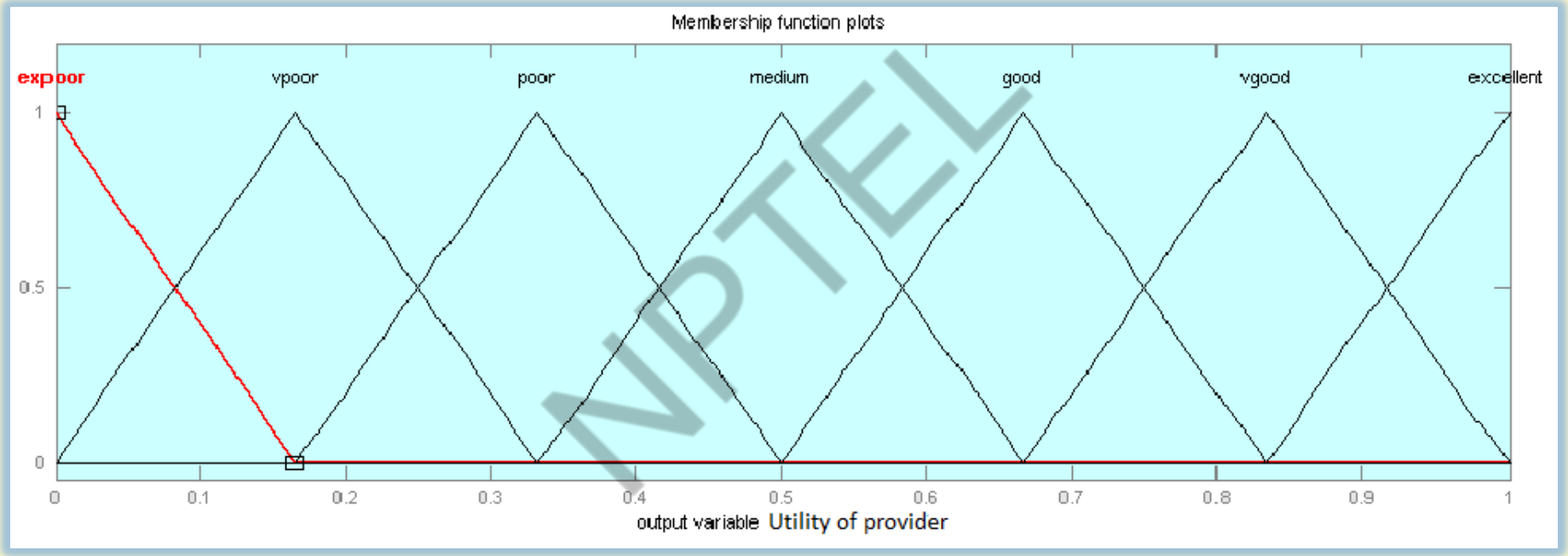


Required QoS 1 = 85% , d_{qos1}=5.0

PROVIDER SELECTION – INPUT MEMBERSHIP FUNCTION



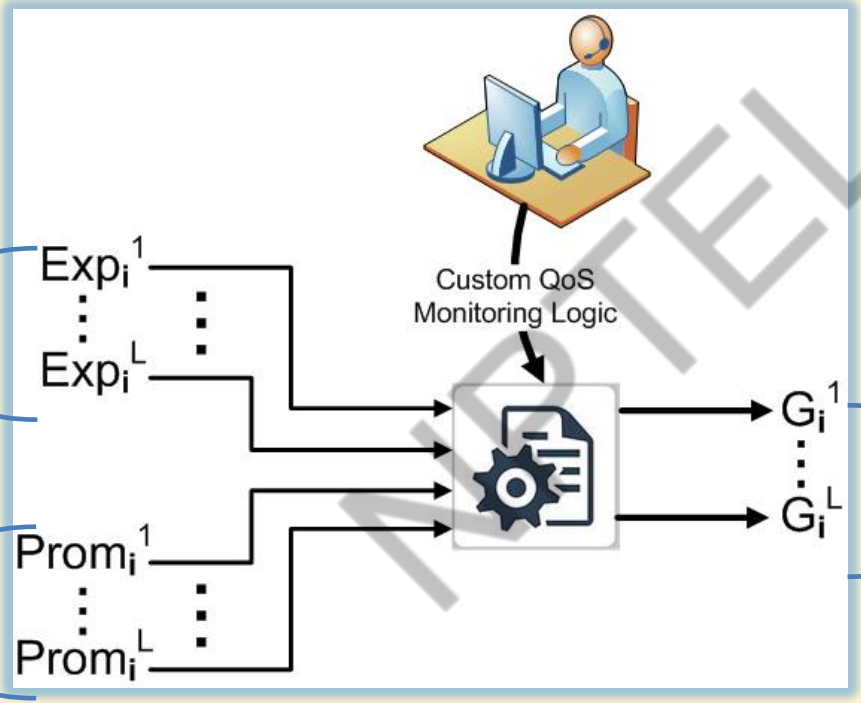
PROVIDER SELECTION – OUTPUT MEMBERSHIP FUNCTION



MONITORING MODULE

From provider/3rd party monitoring tool

From Repository



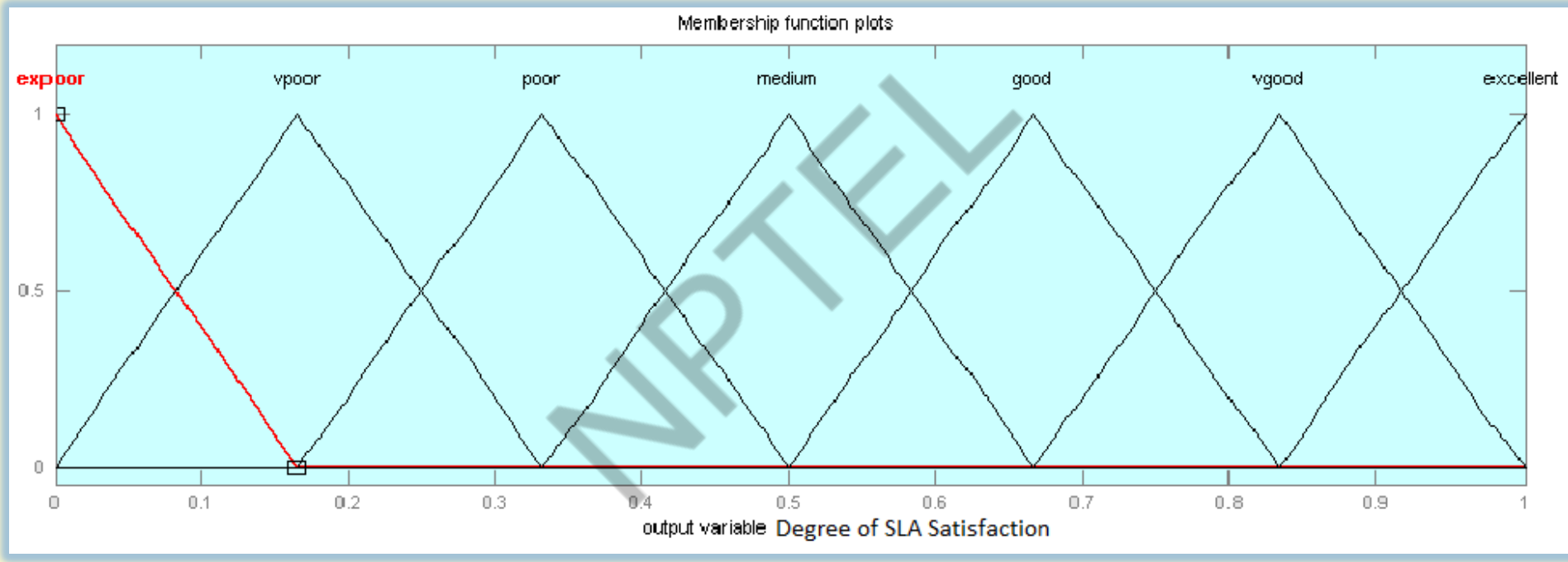
Symbol	Interpretation
$Prom_i^{avail}$	Promised availability for VM_i
$Prom_i^{bw}$	Promised bandwidth for VM_i
$Exp_i^{avail}(t)$	Availability experienced by VM_i at time t
$Exp_i^{bw}(t)$	Bandwidth experienced by VM_i at time t
$F_i^{avail}(t)$	Performance History in availability for VM_i at time t
$F_i^{bw}(t)$	Performance History in bandwidth for VM_i at time t
$G_i^{avail}(t)$	SLA satisfaction for VM_i in availability
$G_i^{bw}(t)$	SLA satisfaction for VM_i in bandwidth

Performance for SI_i in current monitoring period

MIGRATION DECIDER

- Makes use of a fuzzy inference engine
- Input : $F_i^1, F_i^2, \ldots, F_i^L$
- Output : *Degree of SLA Satisfaction* for SI_i
- If *Degree of SLA Satisfaction* < *threshold*, migrate

MIGRATION DECIDER – OUTPUT MEMBERSHIP FUNCTION



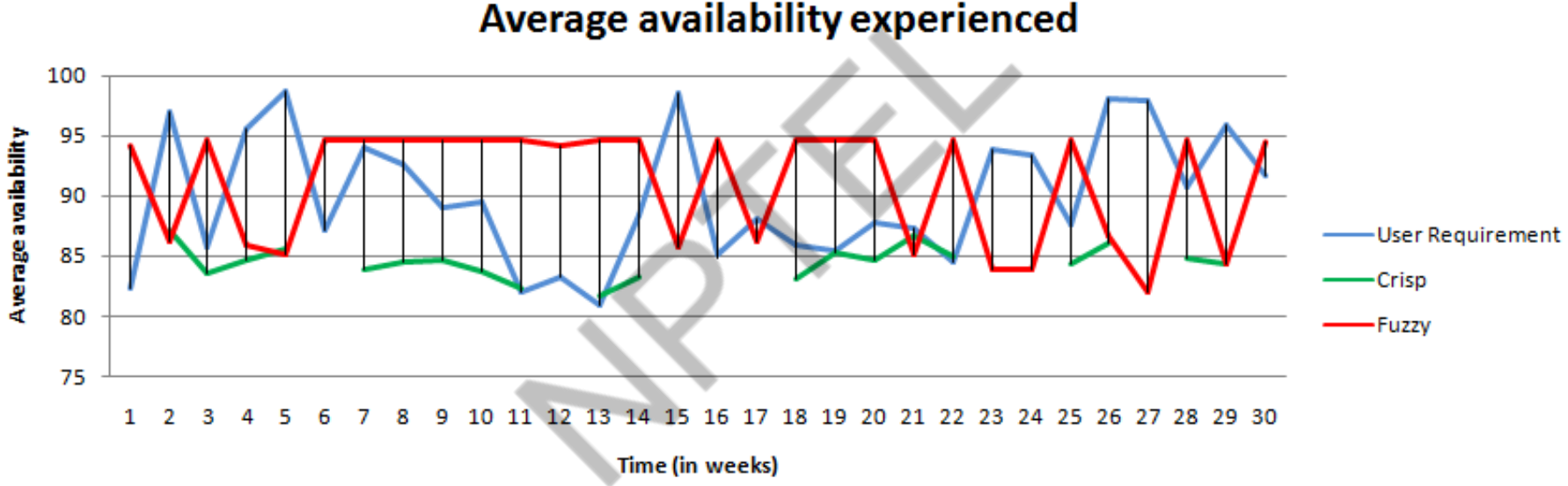
MIGRATION MODULE - SELECTION OF TARGET PROVIDER

- Similar to provider selection
- Selection done using a fuzzy inference engine

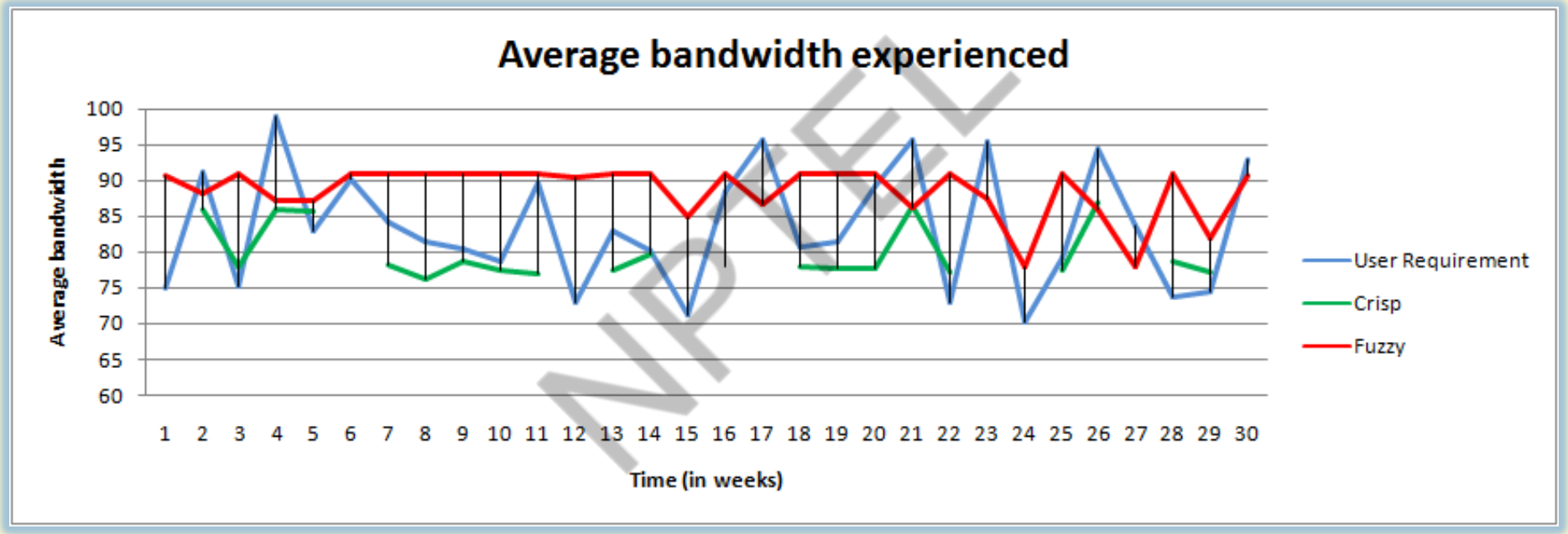
Case study on IaaS Marketplace

- 10 providers with varying offered QoS
- 500 requests for VMs
- Year long simulation
- Few providers exhibit performance degradation. Degraded QoS parameters follow a Gaussian distribution
- Comparison made with conventional (minimum cost) crisp broker

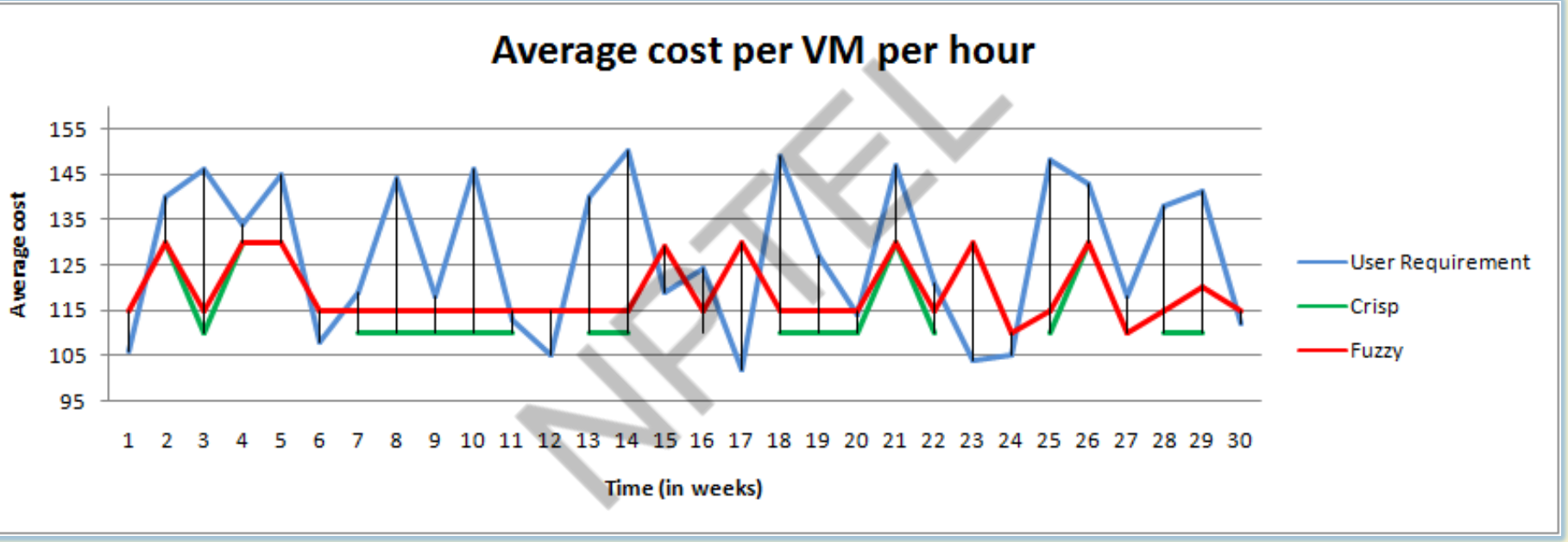
EXPERIMENTS AND RESULTS



EXPERIMENTS AND RESULTS



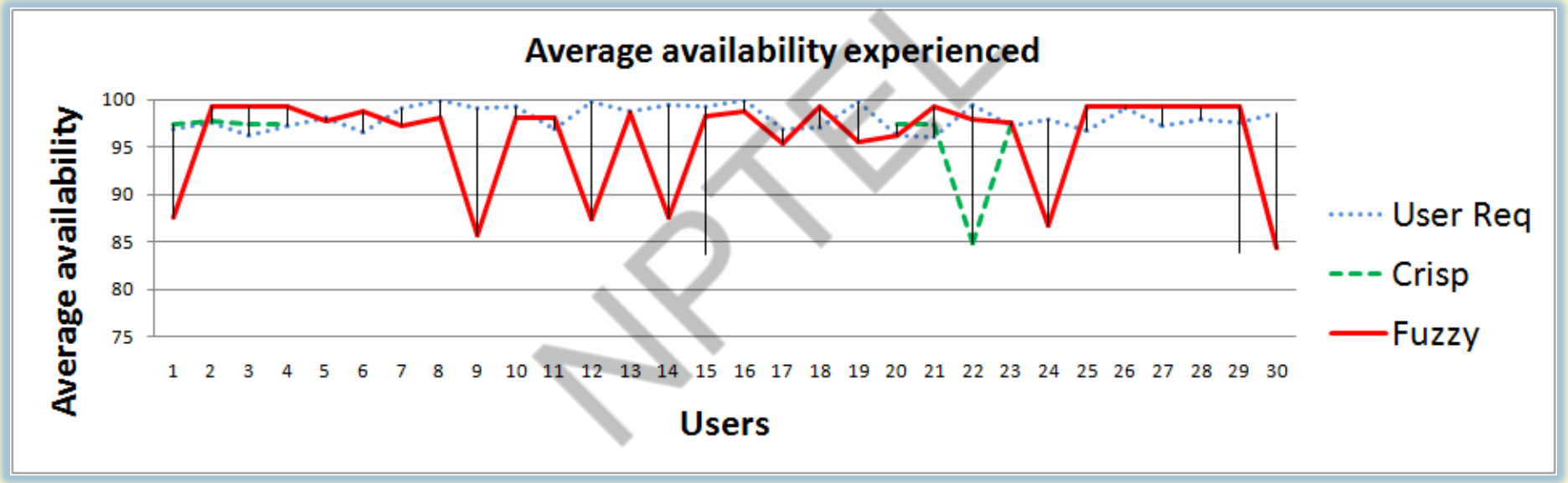
EXPERIMENTS AND RESULTS



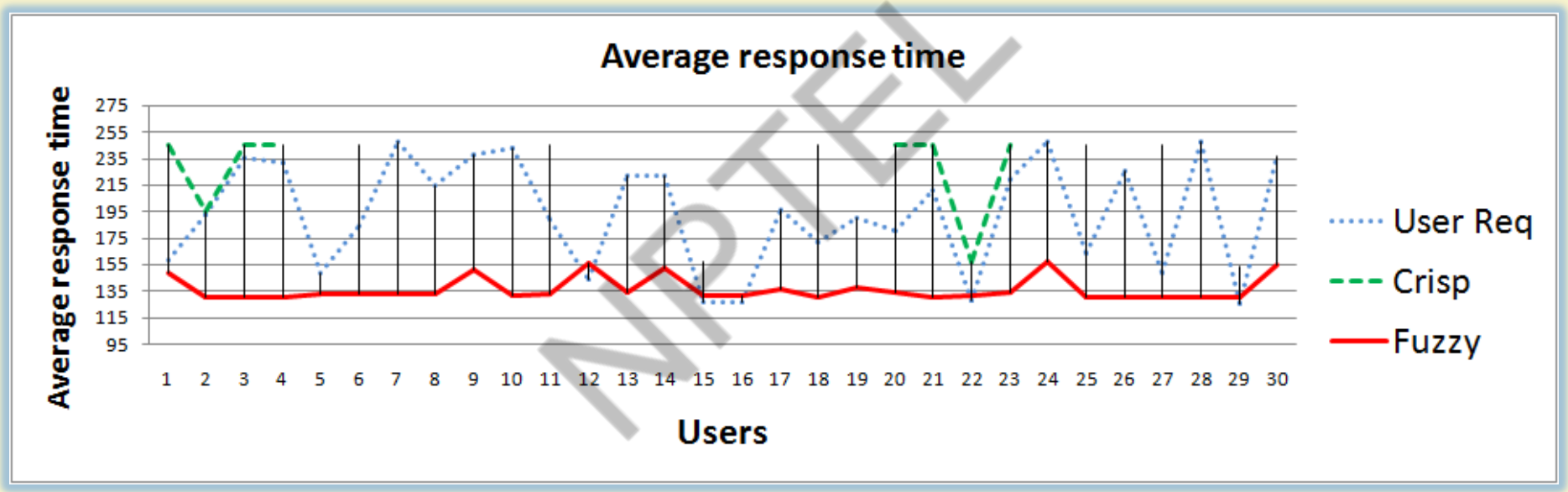
Case study on SaaS Marketplace

- 10 providers with varying offered QoS
- 500 service requests
- Year long simulation
- Few providers exhibit performance degradation. Degraded QoS parameters follow a Gaussian distribution
- Comparison made with conventional (minimum cost) crisp broker

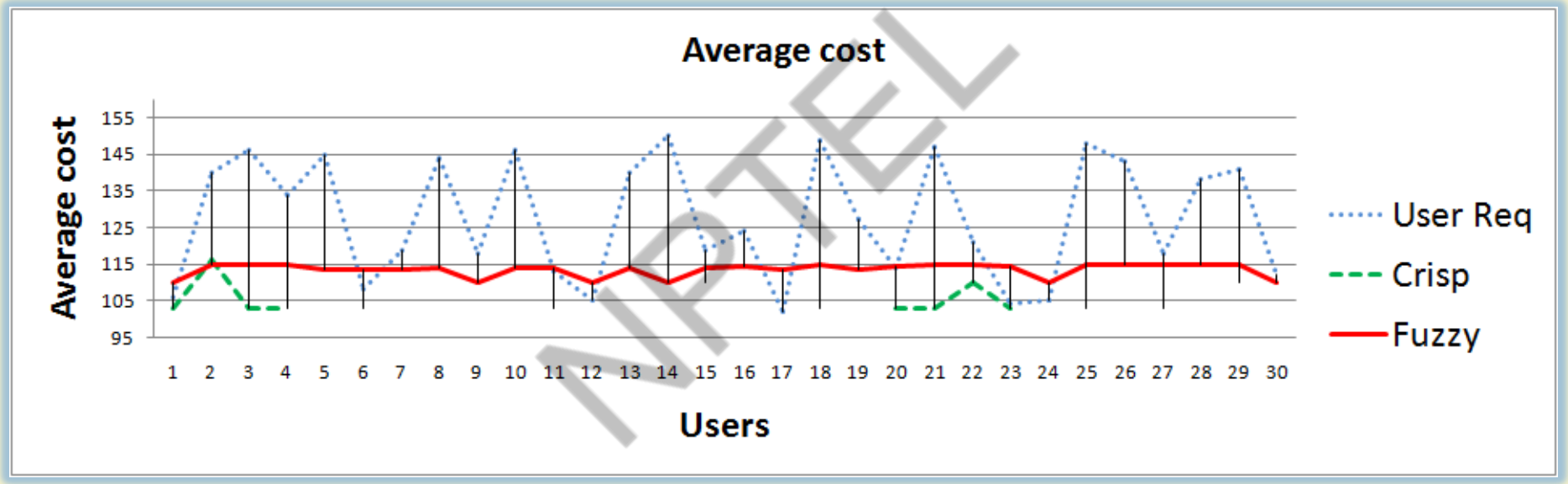
EXPERIMENTS AND RESULTS



Experiments and Results



EXPERIMENTS AND RESULTS



Future Scope

- Specification of flexibility in QoS requirements
- Comparison against existing approaches on production workload
- Service classes for customers

Thank You!!