Security, Availability and Accountability in Cloud Computing

Ozalp Babaoglu

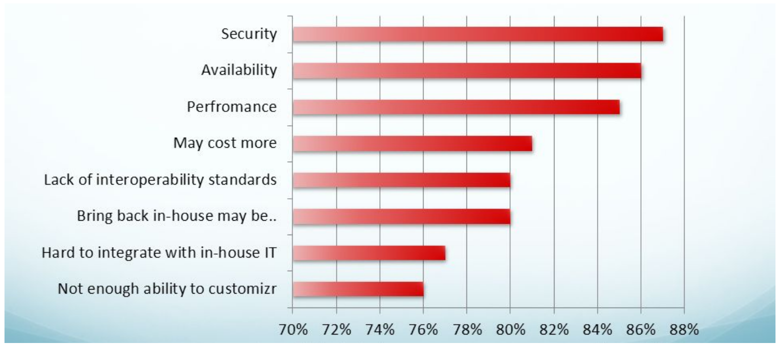
Cloud Security

- There are ***conflicting views*** of security in a cloud computing setting
- Some believe that moving to a cloud ***frees*** an organization from all concerns related to computer security and ***eliminates*** a wide range of threats to their data
- By placing cloud security in the ***hands of experts*** (cloud provider), they believe that they are ***better protected*** than when using ***on-premise*** computing systems

Cloud Security

- Others believe that handing over data and programs to a cloud provider inherently ***reduces the security*** of an organization's IT operations
- Cloud users accustomed to operating ***inside a secure perimeter*** protected by corporate firewalls now have to extend their ***trust*** to the ***cloud service provider*** if they wish to benefit from the economical advantages of utility computing
- The transition away from a model where users have ***full control*** over where their sensitive information is ***stored*** and ***processed*** is a difficult one
- Virtually all surveys report that ***security is the top concern*** of cloud users

Cloud Computing Concerns



% responded ***concerned*** or ***very concerned***

Cloud Computing Security Concerns

- Data breaches
- Lack of control over data lifecycle
- Data loss
- Hijacking of accounts
- Insider threats
- Insecure APIs
- Malware injection
- Distributed-denial-of-service attacks (DDoS)

Cloud Computing Security Concerns

- Security concerns associated with cloud computing derive from ***two sources***:
  - Issues faced by ***cloud providers***,
  - Issues faced by ***their customers***
- Yet, the responsibility is ***shared***: the provider must ensure that their ***infrastructure is secure*** and that their clients' data and applications are protected, while users must take measures to ***fortify their applications***, use ***strong passwords*** and other ***authentication measures***

Cloud Computing Security Concerns Cloud Provider

- The ***cloud provider*** is responsible for
  - ***Physical security***: hardware infrastructure guarded against unauthorized access, theft, fires, floods, power outages and other catastrophic events
  - ***Personnel security***: security screening of potential employees, security awareness and training programs
  - ***Identity management***: integrate customer's identity management system with the provider's own infrastructure, use a federation or single-sign-on technology, biometric-based identification system
  - ***Up-to-date infrastructure***: hardware and software systems free of all known vulnerabilities

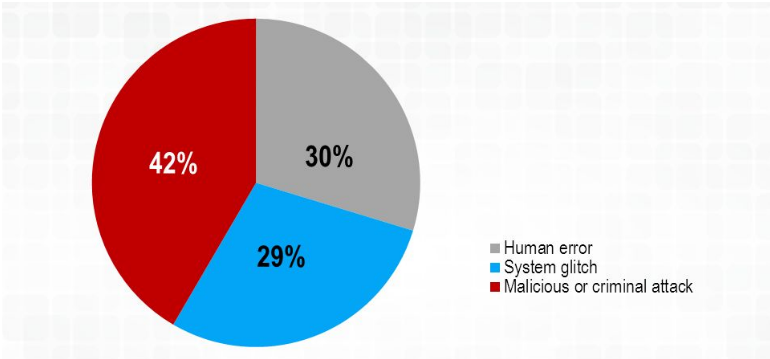
Cloud Computing Security Concerns Cloud Provider

- The ***cloud provider*** is also responsible for
  - ***Integrity*** and ***availability*** of ***user's data*** — user data is ***not corrupted*** and ***continues to be available*** despite unforeseen events (disk crashes)
  - ***Availability of services*** — cloud applications deployed by users ***continue to be available*** despite various ***disruptions*** (power outage, fire, flooding) and ***cyberattacks*** (denial-of-service attacks)

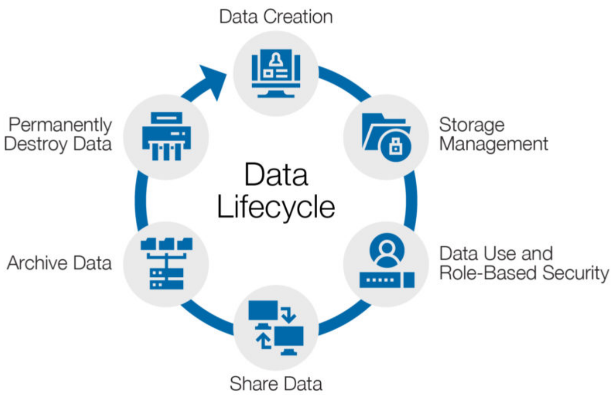
Cloud Computing Security Concerns Cloud Provider

- The ***pooled*** nature of the ***shared infrastructure*** resources that are necessary to facilitate elasticity can be a source for additional security concerns (data leakage)
- Software ***virtualization*** technologies that are necessary to provide the ***isolation*** among users introduce an ***additional layer*** that itself must be ***properly configured***, ***managed*** and ***secured***

Root Causes of a Data Breach



Data Lifecycle

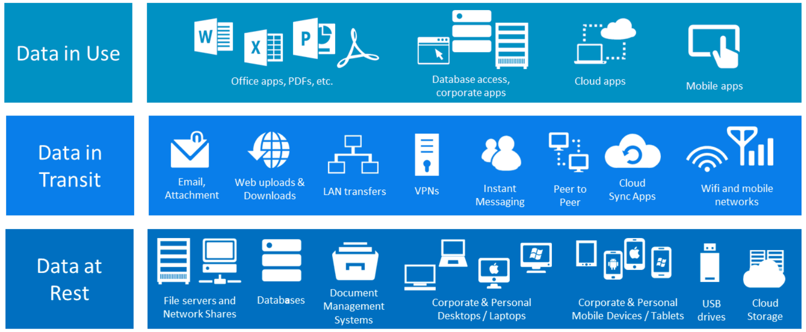


Data Lifecycle

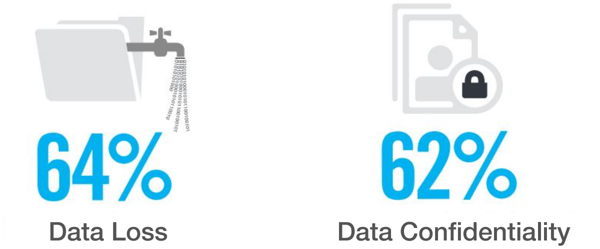
- ***Control*** over the lifecycle of data in a cloud environment is difficult
- Typically, it is impossible for a user to
  - control where data is ***stored***,
  - control if and where it is ***backed up***,
  - determine if data that ***should have been deleted*** was ***actually deleted***
- When data is deleted, there is no guarantee that the media was ***wiped out*** and the next user is not able to recover confidential data
- Cloud providers often rely on ***backups***, typically performed without the user's knowledge or consent, to prevent accidental data loss

States of Data

- While alive, data can be in one of three states:



Cloud Computing Security Concerns Cloud User



Cloud Computing Security Concerns Cloud User

- *Cloud users* have several tools to preserve the ***confidentiality*** of their data and ***minimize risk of data loss*** in cloud settings
- Key technologies that can be deployed in isolation or in combination:
  - *Data encryption*
  - *Data replication*

Cloud Data Encryption

- Encryption of sensitive data is a critical defense against ***unauthorized access*** and ***data theft***
- Encrypted data—even if accessed or stolen —is ***useless*** to third parties without the encryption keys necessary to decrypt it
- A ***cloud data encryption policy*** needs to answer:
  - *What* data needs encryption?
  - *When* does data need encryption?
  - *Where* should cloud encryption be deployed?
  - *Who* should hold the encryption keys?

Cloud Data Encryption

- *What* data needs encryption? Need to consider
  - Does the data fall under ***regulatory compliance requirements***, such as health records (HIPAA), financial data (PCI, SOX), privacy acts (GDPR), or other legal or contractual obligations?
  - Is the data ***personally identifiable*** information?
  - Does the data contain sensitive ***intellectual property***?
  - Is the data ***essential*** to the operation of the organization?

Cloud Data Encryption

- *When* does data need encryption?
  - Encrypting ***data at-rest*** — data saved on disk or other media — is ***essential***
  - Data that moves between the ***user organization*** and the ***cloud provider*** or ***between different clouds*** — ***data in-transit*** — is also vulnerable
  - Communication protocols such as ***SSL***, ***TLS***, ***IPSec***, ***virtual private network*** (VPN) should be used to secure data in-transit

Cloud Data Encryption

- *Where* should cloud encryption be deployed?
  - *Client-Side Encryption* — Encrypt data client-side before uploading it to the cloud
  - *Server-Side Encryption* — Request cloud provider to encrypt your data before saving it on disks in its data centers. Most major cloud providers offer data-at-rest encryption (Amazon S3 with AES-256)
  - *Cloud Application Encryption* — Many software-as-a-service (SaaS) application vendors provide de facto or optional encryption of data. Risk of vendor lock-in
  - *Cloud Security Service Software Encryption* — As a part of their protection services, third-party security software companies offer encryption technologies (*Gemalto SafeNet ProtectV*)

Cloud Data Encryption

- *Who* should hold the encryption keys?
  - Encryption keys can be ***managed*** either by the ***cloud provider*** or by the ***users***
  - Regulatory ***compliance considerations*** may come into factor
  - Regardless of who holds the keys, organizations should make certain that key access is through ***multi-factor authentication*** and that key storage is itself secure and backed
  - Moreover, organizations should keep their keys on storage media ***separate*** from their data

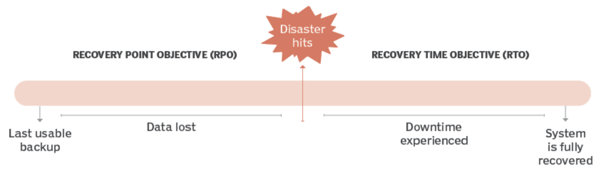
Cloud Data Encryption

- Sensitive data is ***safe while at-rest***, provided that it is encrypted with strong encryption
- To be ***processed***, encrypted data must be ***decrypted*** and this opens a ***window of vulnerability***
- Processing data in its ***encrypted state*** without decrypting is a long-time goal of cryptography and several attempts such as ***homomorphic encryption***, ***searchable symmetric encryption*** and ***order-preserving encryption*** exist

Disaster Recovery

- ***Recovery Time Objective*** (RTO) is the duration of time within which a business process ***must be restored*** after a disaster in order to avoid unacceptable consequences associated with a break in continuity
- ***Recovery Point Objective*** (RPO) describes the interval of time that might pass between your last data backup and a disaster before the ***quantity of data lost*** during that period causes ***serious damage*** to your business

Disaster Recovery



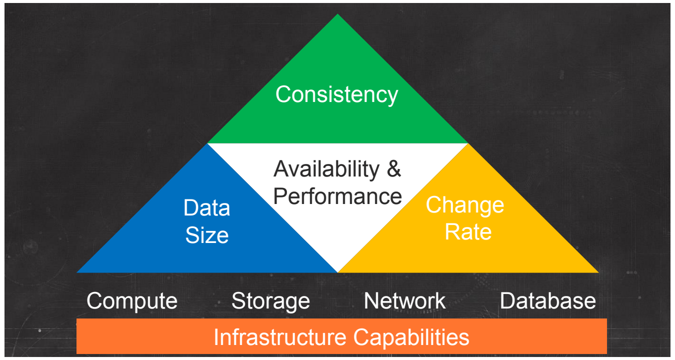
- These objectives can guide an organization in choosing optimal data backup and replication strategies
- Help setting ***frequency of backups*** or employing more aggressive ***data replication*** strategies

Data Replication Strategies

- ***Single copy*** — no replication
- ***Periodic backups*** — a backup is like a replica but limited to ***disaster recovery*** and not suitable for ***normal access***
- ***Independent copies*** — any copy can be ***read*** (for increased throughput) but ***not written***. Not suitable if data can change (be ***written***)
- ***Master-Slave*** — any copy can be ***read***, but writes limited to a ***master*** that assumes the responsibility to ***propagate*** the changes to slaves
- ***Fully distributed*** — any copy can be ***read***, any copy can be ***written*** subject to different consistency obligations

Data Replication Strategies

- Factors to consider when selecting a replication strategy



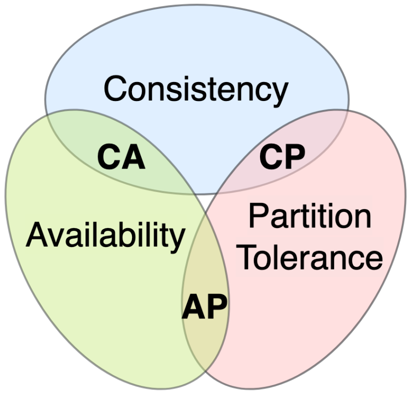
Data Consistency Models

- How should the replicated version of data behave when compared to its non-replicated counterpart?
  - *Strict consistency* — all updates to an item are seen by all copies in the same order (copies always return the value of the last update)
  - *Sequential consistency* — updates to an item by any given writer are seen by all copies in the same order
  - *Causal consistency* — only updates that are causally related are seen by all copies in the same order
  - *Eventual consistency* — if no new updates are made to a given data item, eventually all copies of that item will return the last update value

CAP Theorem

- *CAP theorem*, also known as Brewer's theorem (after Eric Brewer), states that any distributed data store can provide *only two* of the following three guarantees:
  - *Consistency*: Every read receives the most recent write or an error
  - *Availability*: Every request receives a (non-error) response, without the guarantee that it contains the most recent write
  - *Partition tolerance*: The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes

CAP Theorem



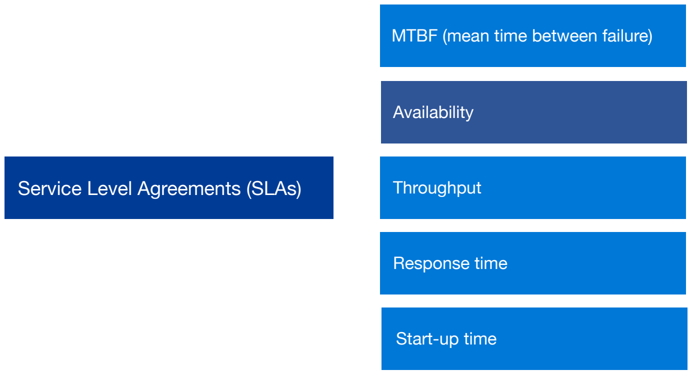
AWS Data Replication

- Amazon ***Simple Storage Service*** (S3) achieves high availability by replicating data ***across multiple servers*** within AWS ***data centers*** or ***Availability Zones***
- S3 can be configured to further replicate objects across ***different AWS Regions*** using ***Cross-Region Replication*** (CRR) or between buckets in the ***same AWS Region*** using ***Same-Region Replication*** (SRR)
- *CRR* use cases
  - ***Data residency requirements*** — Amazon S3 stores data across multiple geographically distant ***Availability Zones*** by default, but compliance requirements might dictate that data be stored at even greater distances
  - ***Latency performance*** — by maintaining object copies in AWS Regions that are geographically closer to end-users in two geographic locations, latency in accessing objects can be minimized

AWS Data Replication

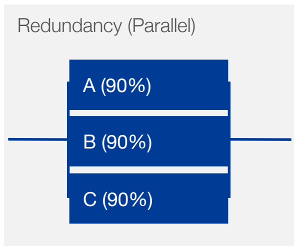
- *SRR* use cases
  - ***Replication between developer and test accounts*** — the same data can be shared between multiple accounts and SRR can be used to change account ownership for the replicated objects to protect data from ***accidental deletion***
  - ***Abide by data sovereignty laws*** — Often customers are required to store data in separate AWS accounts while being barred from letting the data leave a given country. In such circumstances, SRR can be used to ***backup critical data*** while remaining within national boundaries

Cloud Quality of Service Metrics

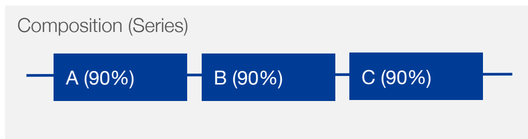


Availability

$$Availability = \frac{uptime}{uptime + downtime}$$



$$Availability = 1 - (1 - 0.9)^3 = 99.9\,\%$$



$$Availability = 0.9^3 = 72.9\,\%$$

Meaning of 9s

Service availability (%)	System type	Annual down minutes	Quarterly down minutes	Monthly down minutes	Practical meaning	FAA rating
90	Unmanaged	52,560	13,140	4,383	Down 5 weeks per year	
99	Managed	5,256	1,314	438	Down 4 days per year	ROUTINE
99.9	Well managed	525	131	43.83	Down 9 hours per year	ESSENTIAL
99.99	Fault-tolerant	52	13	4.38	Down 1 hour per year	
99.999	High availability	5.26	1.31	0.44	Down 5 minutes per year	CRITICAL
99.9999	Very high availability	0.53	0.13	0.04	Down 30 seconds per year	
99.99999	Ultra availability	0.05	0.01	0.004	Down 3 seconds per year	SAFETY-CRITICAL

Accountability and Trust

- A large number of current cloud customers are ***governments***, ***banks***, ***pharmaceuticals companies*** and other ***large corporations*** that outsource only ***small pieces*** of their enterprise that deal with ***less sensitive*** data to the cloud
- What prevents corporations and government organizations from realizing the full potential of cloud computing?
- Lack of ***accountability***, and as a consequence, lack of ***trust***
- Moreover, the ***Service Level Agreements*** often do not provide adequate ***legal protection*** for cloud users who are often left to deal with events beyond their control

Amazon Compute SLA

"AWS will use commercially ***reasonable efforts*** to make the *Included Services* each available for each AWS region with a *Monthly Uptime Percentage* of at least 99.99%, in each case during any monthly billing cycle (the "*Service Commitment*"). In the event any of the *Included Services* do not meet the *Service Commitment*, you will be eligible to receive a *Service Credit* as described below"

Amazon Compute SLA

Monthly Uptime Percentage	Service Credit Percentage
Less than 99.99% but equal to or greater than 99.0%	10%
Less than 99.0% but equal to or greater than 95.0%	30%
Less than 95.0%	100%

Amazon Compute SLA Exclusions

"The *Service Commitment* and *Hourly Commitment* do not apply to any unavailability, suspension or termination of an *Included Service*, or any other *Included Service performance issues*: (i) caused by ***factors outside of our reasonable control***, including any ***force majeure event*** or ***Internet access*** or ***related problems*** beyond the demarcation point of the applicable *Included Service*; (ii) that result from any ***actions or inactions*** of ***you or any third party***, including failure to acknowledge a recovery volume"

Amazon Compute SLA Exclusions

"(iii) that result from ***your equipment, software or other technology*** and/or third party equipment, software or other technology (other than third party equipment within our direct control); or (iv) arising from ***our suspension or termination*** of your right to use the applicable *Included Service* in accordance with the Agreement (collectively, the "*Amazon Compute SLA Exclusions*"). If availability is impacted by factors other than those used in our *Monthly Uptime Percentage* calculation, then we may issue a *Service Credit* considering such factors ***at our discretion***"

Some Notable Cloud Outages

- *20 March 2021:* WhatsApp, Facebook, Instagram suffer major global outage that lasts several hours
- *May 2019:* Salesforce faced one of its biggest service disruptions when the deployment of a database script to its Pardot Marketing Cloud ended up granting elevated permissions to regular users
- *June 2019:* Cascading errors created a network congestion problem that brought down many Google Cloud services for roughly four hours, in addition to large GCP customers like Snapchat and Shopify
- *August 2019:* an Amazon AWS US-EAST-1 datacenter in North Virginia experienced a power failure leading to the datacenter's backup generators to start failing

Some Notable Cloud Outages

- *July 2019:* many iCloud users across the globe briefly got the message of "Service Unavailable – DNS failure" for several hours
- *June, 2016:* The storms that battered Sydney in June, 2016, also shook AWS services. An extensive power outage led to the failure of a number of Elastic Compute Cloud (EC2) instances and Elastic Block Store (EBS) volumes, many of which hosted critical workloads for big brands
- *September, 2013:* Infamously called the "Friday the 13th outage," a load balancing issue led to some regional customers being hit for a period of two hours across one availability zone in Virginia
- *December, 2012:* The Christmas of 2012 was not so merry after all, especially for those affected by the much-talked-about AWS failure. As a result of the outage, Netflix was down on Christmas Eve

Accountability and Trust

- Cloud customers actually ***exert far more control*** over their vendors than traditional software customers
- Cloud application customers pay a ***recurring subscription fee*** and cloud vendors are typically held to monthly service level agreements (SLAs)
- This provides a financial motivation for cloud vendors to earn their ***customers' business every month***—by maintaining excellent support and operations, and high customer satisfaction

Accountability and Trust

- Traditional software vendors are paid a big ***upfront license fee*** in exchange for a perpetual license
- They have ***fewer obligations*** once the software has been deployed
- Whether the software works or not becomes the customer's problem
- The ongoing subscription model ensures that cloud application vendors ***remain accountable*** on a ***continual basis*** to their customers—unlike traditional software vendors that sell software and move on

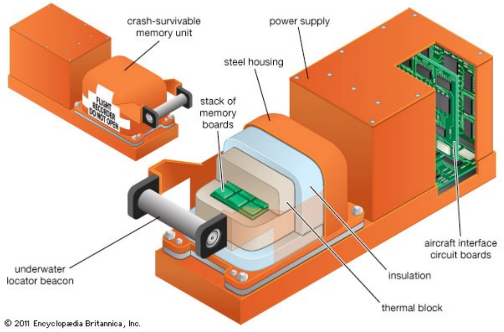
Accountability and Trust

- ***Accountability*** in a cloud computing environment needs to address:
  - Who is responsible when ***data is lost, corrupted or disclosed***?
  - Who is responsible when applications ***return no results, late results or erroneous results***?
  - What are the ***legal implications*** of data and applications being held by ***third parties***, possibly in ***multiple judicial domains***?
  - How can disputes be settled ***impartially by third parties***?

Civil Aviation

- Look to the ***safety*** aspects of ***Civil Aviation*** for inspiration
- Civil Aviation is a very complicated system of mutually suspecting agents set in a complex technological, economical, international, regulatory and legal context
- Yet it works surprisingly well and flying as a mode of transportation enjoys a high level of *trust* among its customers
- An important factor of this trust in flight safety rests with the requirement (by international law) that airlines render their flight operations *accountable*
- The famous "black box"

Flight Data Recorder



Flight Data Recorder

- Specification regulated by the *International Civil Aviation Organization* (ICAO)
  - Tamper proof
  - Withstand an acceleration of 3400 g's
  - Withstand extreme high and low temperatures
  - Withstand immersion to a depth of up to 6,000 meters
- Since the 1960's, mandatory for all commercial aircraft under internationally-agreed regulations
- Recorded data can be extracted and analyzed by the *Flight Data Analysis Service* of the *International Air Transport Association* (IATA)

Flight Data Recorder

- In the aftermath of an accident, the recovered *Flight Data Recorder* (together with the *Cockpit Voice Recorder*) are typically sufficient to attribute the cause of the accident to
  - Airline (pilot, cabin crew, ground personnel, maintenance, etc.)
  - Aircraft Manufacturer (design, manufacturing, materials, etc.)
  - Other parties (air traffic controller, another aircraft, etc.)
  - External Factors (weather, birds, volcanoes, etc.)
- Evidence typically stands in court and is the basis for legal settlements

"Flight Recorder" for the Cloud

- What we need is a "cloud flight recorder" (CFR)
- Integral part of a technical infrastructure along with a legal regulatory framework for making cloud computing ***accountable***, and ultimately making cloud services ***mutually trustworthy*** for customers and providers

Actors

- Actors in a CFR-enabled cloud setting:
  - *P* - the cloud service provider
  - *U* - end user
  - *Q* - regulatory organization, equivalent to the ICAO
  - *R* - certified CFR provider
- The logs maintained by the CFR should be *self extracting* and *self describing* such that the equivalents of the *National Transportation Safety Board*, *Flight Data Analysis Service* and lawyers/judges are not needed

Technical Challenges

- How to extract from a service contract formal descriptions of
  - rules that state the rights, obligations and prohibitions of providers and customers,
  - specifications for a CFR as a list of events and their attributes that must be logged
- Requirements for a CFR logging facility:
  - *Fine-grained* to allow backtracking of "incidents"
  - *Tamper resistant*
  - *Trustworthy*
  - *Non-reputable*
  - *Non-intrusive*
  - *Closed* (does not rely on services outside the cloud itself)

Peer-to-Peer Clouds

- How to *dynamically allocate* and *share* huge collections of commodity resources among many peer-to-peer applications?
- Not unlike "multiplexing" a distributed infrastructure in a totally decentralized manner to create a p2p "timesharing" system
- Wrote up the idea as a position paper:

O. Babaoglu, M. Jelasity, A-M Kermarrec, A. Montresor, M. van Steen. *Managing clouds: a case for a fresh look at large unreliable dynamic networks*, ACM SIGOPS Operating Systems Review, 2006

Peer-to-Peer Clouds

- Is it possible to build a *cloud computing* platform as a *peer-to-peer* system?
- Extreme point in the spectrum of cloud computing architectures from centralized-to-federated-to-p2p
- The architecture inherits characteristics of p2p systems:
  - Total decentralized
  - Self organized
- "Poor man's" cloud computing platform

Peer-to-Peer Clouds

- In our *Managing Clouds* paper, we used the "cloud" metaphor to highlight *granularity* and *fluidity*:
  - huge number of water droplets or ice particles,
  - individually insignificant but aggregated significant,
  - in a state of flux with constantly changing boundaries,
  - yet, maintaining an identifiable shape

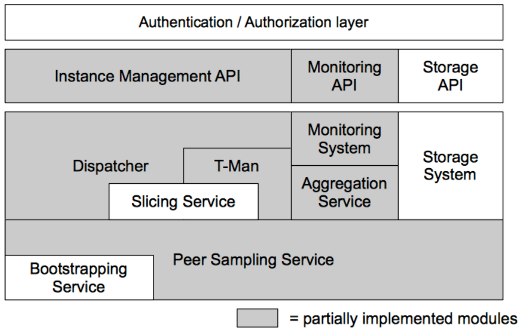
Peer-to-Peer Clouds

- What are these "water droplets" in practice?
- Range from set-top boxes to ADSL/Broadband modems to game consoles to laptops to multi-core PCs
  - have onboard computing and storage resources,
  - are owned and operated by different individuals,
  - are physically located at individuals' homes,
  - remain "mostly on" but can be powered off or unplugged from the network

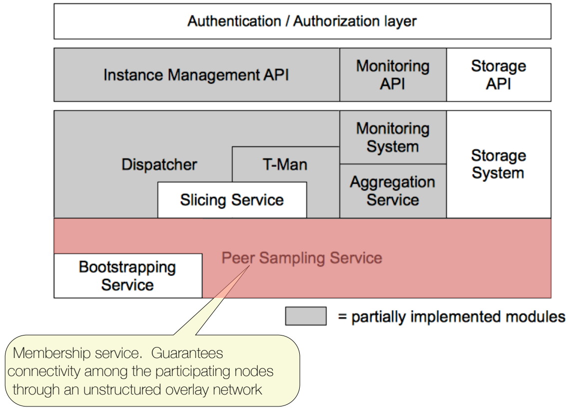
Peer-to-Peer Clouds

- The infrastructure we envision is similar to a classical p2p system and, as a basis for cloud computing, offers
  - Very low initial investment costs,
  - Distributed power consumption,
  - Distributed heat generation/dissipation,
  - Distributed network connectivity
- The challenge is to maintain a coherent abstraction over this *large-scale, distributed*, *unreliable* and *dynamic* infrastructure in a totally decentralized and self-organizing manner

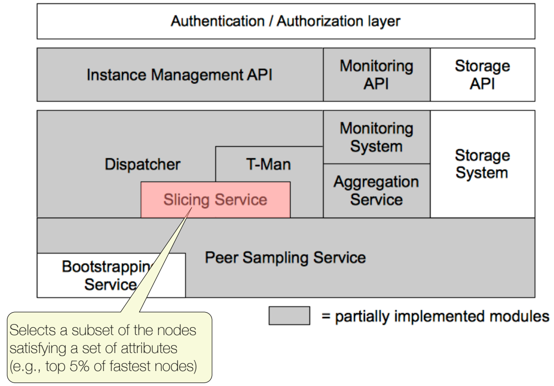
P2P Cloud - Architecture



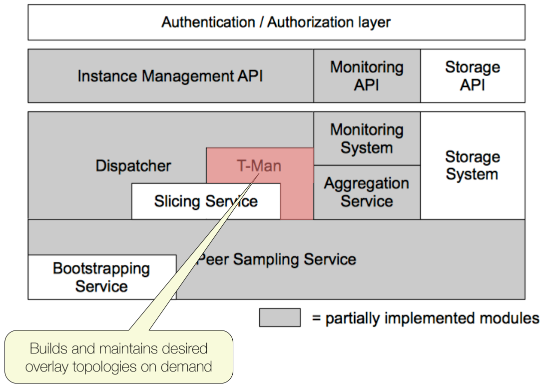
P2P Cloud - Architecture



P2P Cloud - Architecture

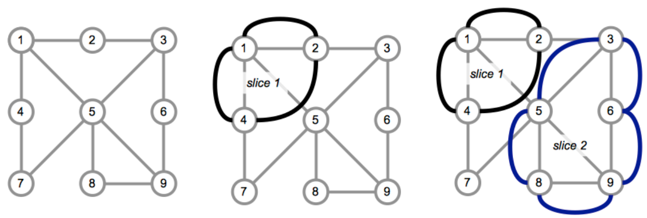


P2P Cloud - Architecture



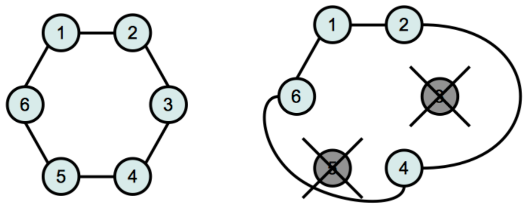
Sub-Clouds through Slicing

- Slicing builds sub-clouds as disjoint ring overlays on top of the unstructured membership layer

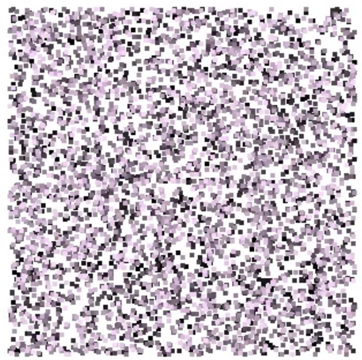


T-Man

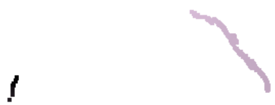
- Sub-clouds are maintained in the presence of churn using the T-Man overlay protocol



Building a ring with T-Man



Repairing a ring with T-Man



Repairing a ring with T-Man

P2P Cloud - Architecture