# Kai Hwang : *Cloud Computing for Machine Learning and Cognitive Applications*

The MIT Press, June 2017

## Chapter 10: **Cloud Benchmark Performance, Security and Data Privacy Solutions**

**(60 Slides for 3-hour lectures)**

# Cloud Performance Issues

1) *Scaling measurement:* Cloud scaling is done with virtualized resources. Hence, the scale of computing power is decided at various abstraction levels of virtual resources

2) *Workload scenario:* Cloud aims to accommodate workload with large number of small jobs. Scaling strategies must match with such a workload scenario

# Cloud Performance Issues

3) *Performance attributes:* To benefit a large number of small jobs, performance concerns are the response time and throughput, rather than batch execution time

4) *Cloud productivity:* Productivity is tied to performance cost ratio. Tradeoffs do exist in high performance versus service costs to massive users

# Scale-Out Workloads

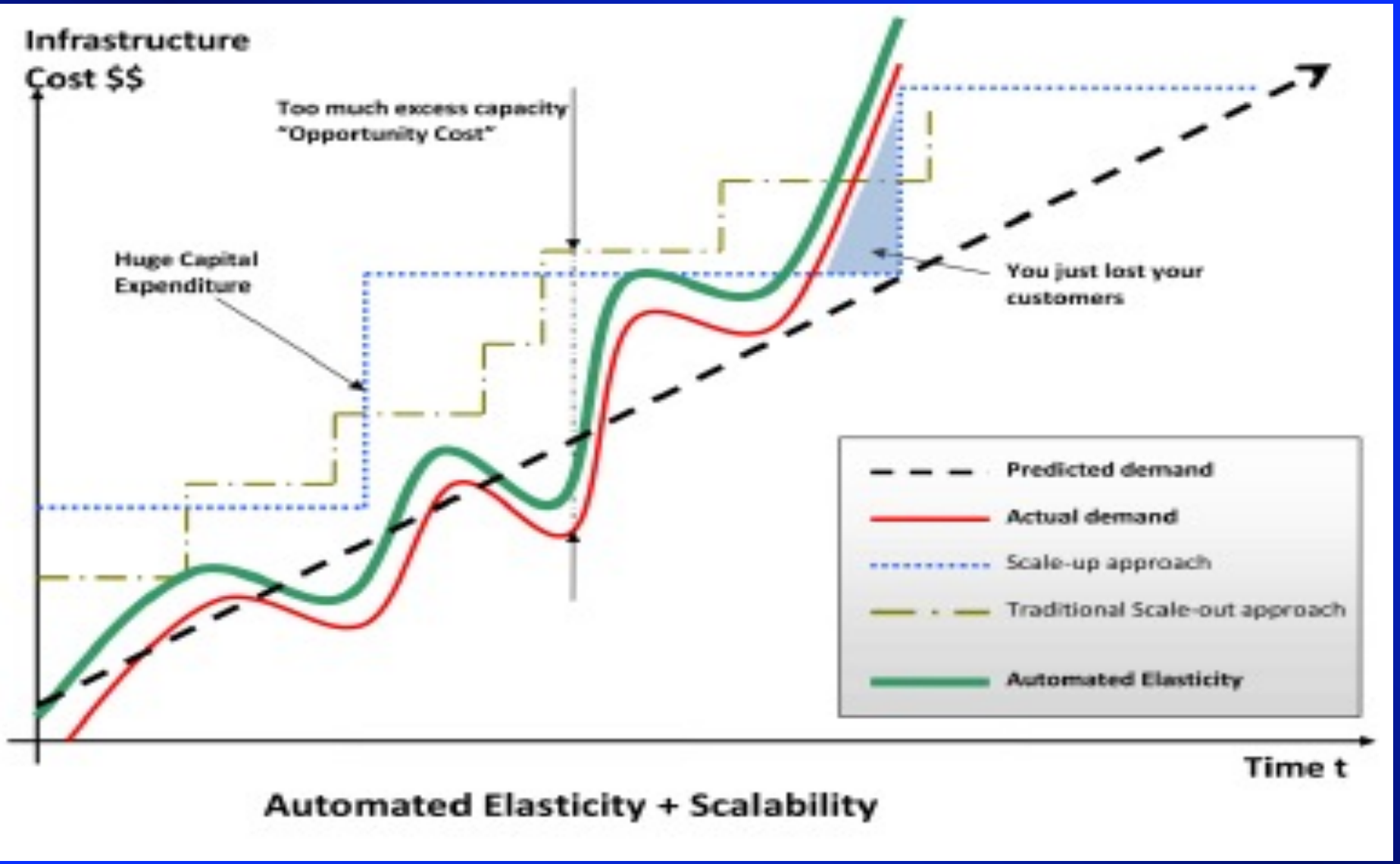
Cloud workloads are characterized by their dataset size, algorithms, memory-access pattern, and service model applied. Scaling techniques cover three cloud workload types:

- *Scale-out technique* allows adding more machine instances or processing nodes of the same type based on the quota agreed in the *service-level agreement* (SLA). Obviously, scaling out appeals more to the use of homogeneous clusters with identical nodes.

# Scale-Up and Mixed Workloads

- *Scale-up technique* is implemented with scaling the cloud from using small nodes to more powerful nodes equipped with better processor, memory or storage.
- *Mixed scale-up/scale-out technique* allows one to scale up or scale-down the instance type and adjust the instance quantity by scale-out (increasing) or scale-in (reducing) resources at the same time. Mixed scaling appeals better with using heterogeneous clusters.

# Understanding Elasticity in Cloud Resources



# Elastic Cloud Resources Provisioning for High Throughput Performance

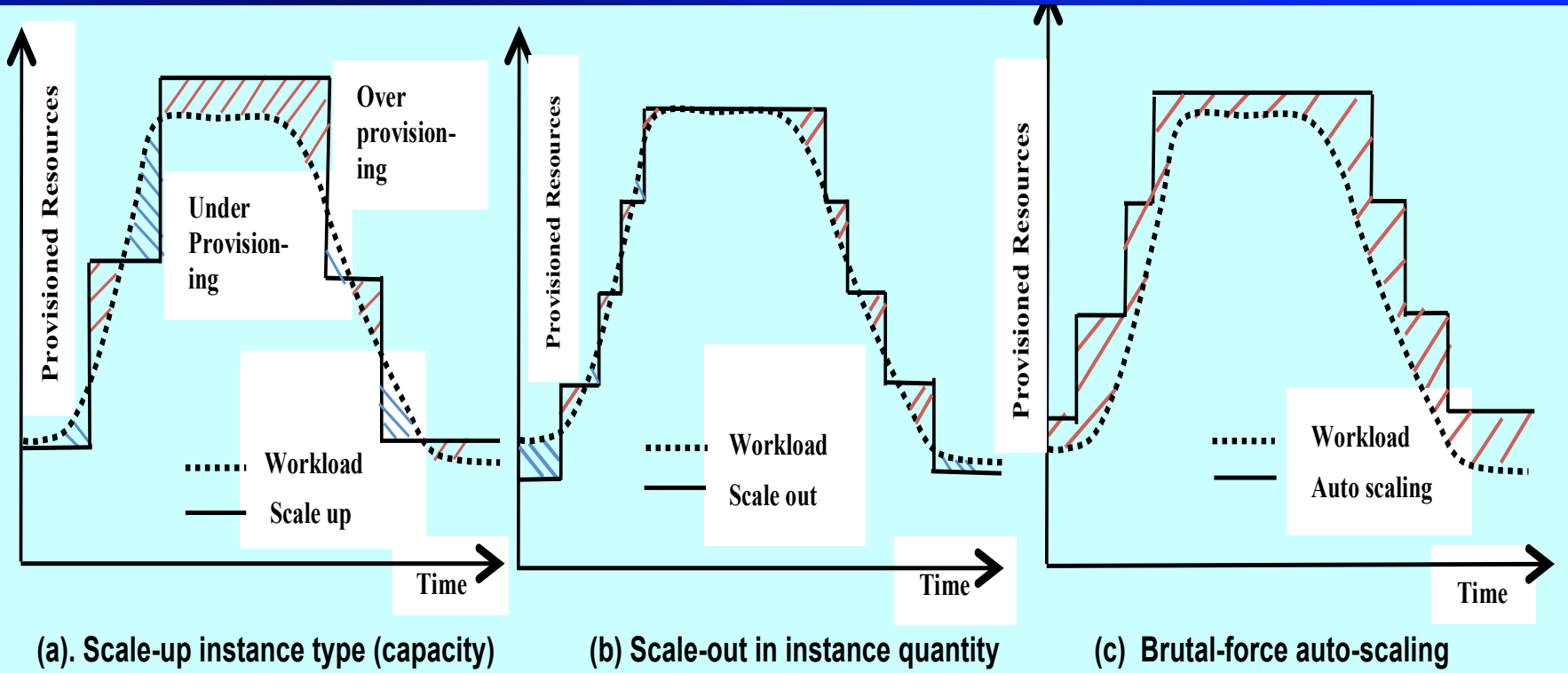



Figure 1: Auto-scaling, scale-out and scale-up machine instance resources in elastic IaaS clouds, where over-provisioning and under-provisioning of machine resources are shown in differently shaded areas above and below the workload curves. .

TABLE 2. PERFORMANCE, CAPABILITY AND PRODUCTIVITY METRICS FOR EVALUATING CLOUDS

| Abstraction Level | Performance Metric | Notation (Eq. #) | Brief Definitions with Representative Units or Probabilities |
|---|---|---|---|
| Basic Performance Metrics | *Execution time* | $T_e$ | Time elapsed during program or job execution, (sec., hours) |
| | *Speed* | $S_r$ | Number of operations executed per second, (PFlops, TPS, WIPS, etc.) |
| | *Speedup* | $S_u$ | Speed gain of using more processing nodes over a single node |
| | *Efficiency* | $E$ | Percentage of max. Performance (speedup or utilization) achievable (%) |
| | *Scalability* | $S$ (Eq.5) | The ability to scale up resources for gain in system performance |
| | *Elasticity* | $E_l$ (Eq.14) | Dynamic interval of auto-scaling resources with workload variation |
| Cloud Capabilities: | *Latency* | $T$ | Waiting time from job submission to receiving the first response. (Sec.) |
| | *Throughput* | $H$ | Average number of jobs/tasks/operations per unit time (PFops, WIPS.) |
| | *Bandwidth* | $B$ | Data transfer rate or I/O processing speed, (MB/s, Gbps) |
| | *Storage Capacity* | $S_g$ | Storage capacity with virtual disks to serve many user groups |
| | *Software Tooling* | $S_w$ | Software portability and API and SDK tools for developing cloud apps. |
| | *Bigdata Analytics* | $A_n$ | The ability to uncover hidden information and predict the future |
| | *Recoverability* | $R_c$ | Recovery rate or the capability to recover from failure or disaster (%) |
| Cloud Productivity | *QoS of Cloud* | *QoS* | The satisfaction rate of a cloud service or benchmark testing (%) |
| | *Power Demand* | $W$ | Power consumption of a cloud computing system (MWatt) |
| | *Service cost* | *Cost* | The price per cloud service (compute, storage, etc.) provided, ($/hour) |
| | *SLA/Security* | $L$ | Compliance of SLA , security, privacy or copyright regulations |
| | *Availability* | $A$ | Percentage of time the system is up to deliver useful work. (%) |
| | *Productivity* | $P$ , (Eq.4) | Cloud service performance per unit cost, (TFlops/$, WIPS/$, etc.) |

# Basic Performance Metrics

***Speed (S):*** **Number of *millions of operations per second* (*Mops*). The operation could be integer or floating-point like *MFlops*. The speed is also called *throughput* such as *millions of web interactions per second* (*MIPS*), etc.**

***Speedup ($S_u$):*** **Speed gain of using multiple nodes**

***Efficiency ($E_f$):*** **Percentage of peak performance achieved**

***Utilization (U):*** **Busy resources (CPU, memory, storage)**

***Scalability (S):*** **Scaling ability to upgrade performance**

# Cloud Capabilities are macroscopic metrics

*Latency* ($L$): System response time or access latency

*Bandwidth* ($B$): This is data transfer rate or I/O rate

*Elasticity* ($E_l$): The ability for cloud resources to scale up/down or scale in/out to match with workload variation

*Software* ($S_w$): Software portability, API and SDK tooling

*Big-data Analytics*:($A_n$): The ability to uncover hidden information or predict trends in big data

# .Cloud Productivity Measures

***Quality of Service (QoS):*** **Satisfaction on user services**

***System availability (A):*** **The *s*ystem up time per year**

***Service costs ($C_o$):*** **User renting costs and provider cost**

***Power Demand (W):*** **Cloud power consumption (MWatt)**

***SLA/Security (L):*** **Compliance of SLA, security, etc.**

***Productivity (P):*** **QoS-satisfied perf. per unit cost**

# Elasticity Analysis of Scalable Cloud Performance

- Elasticity refers to the stretchability of cloud resources like CPU power, storage capacity, etc.
- Elasticity is measured by the speed (or overhead) to reconfigure the cloud configuration
- Elastic resource provisioning cannot not be done with physical machine, only virtual machines can be refigured in real-time

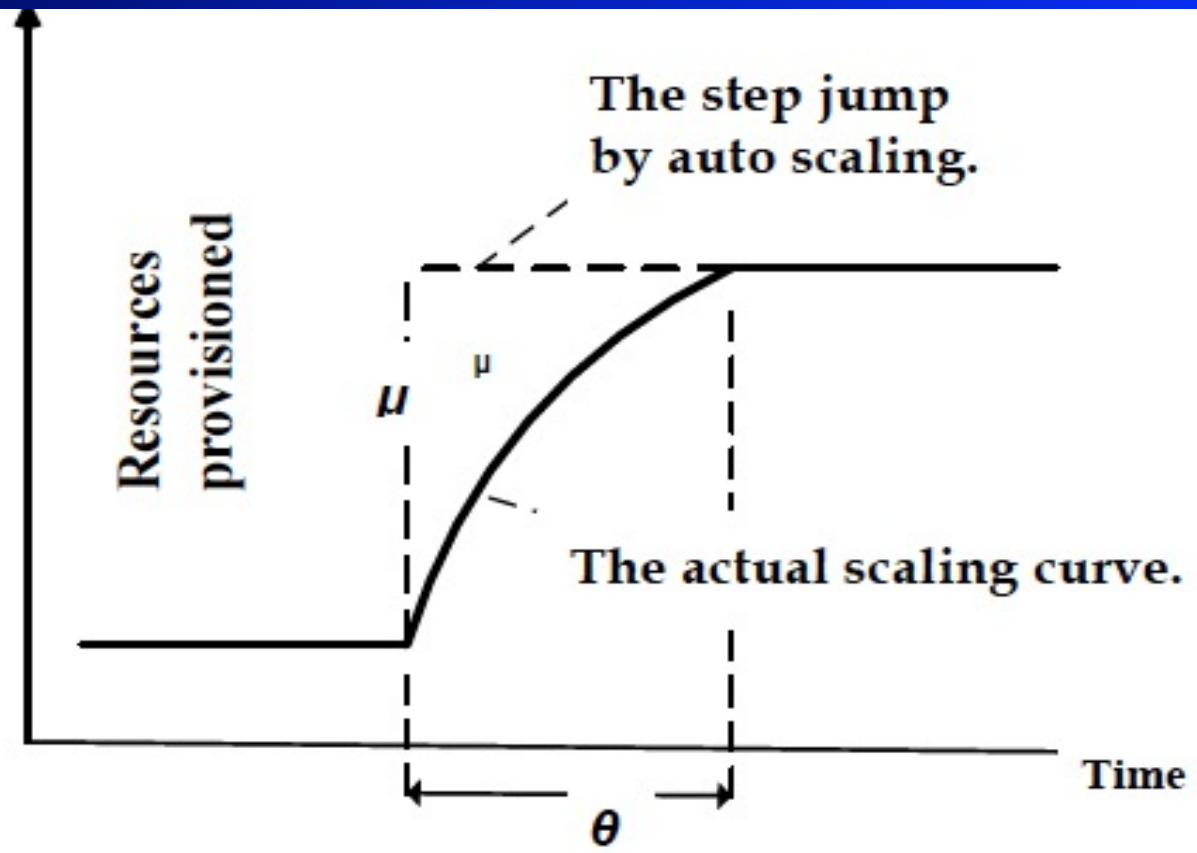



Figure 3. Illustration of cloud resource provisioning, where $\theta$ is the overhead time and $\mu$ is the offset between actual scaling and an auto scaling process.

# Elastic Compute Unit (ECU)

- Amazon AWS has defined a term: *Elastic (or EC2) Compute Unit* (ECU) as an abstract unit to quantify the computing capacity of a machine instance type
- By 2009 standard, the performance of a 1 ECU instance is roughly equivalent to the CPU capacity of a 1.2 GHz 2007 Xeon processor
- Each physical CPU (processor) can house a number a number of *virtual CPU* (vCPU). Also the memory and storage may affect the ECU count

# Performance Metrics: Speedup and Efficiency

Given a problem size $x$ and n-processors system, the *speedup*$(n,x)$ is the sequential execution time $Time(1,x)$ divided by parallel execution time $Time(n,x)$.

$$Speedup(n,x) = (Time(1,x))/(Time(n,x)) \quad (1)$$

The *efficiency*$(n,x)$ is defined by the following ratio :

$$Efficiency(n,x) = \frac{Speedup(n,x)}{n} = (Time(1,x))/(n \times Time(n,x)) \quad (2)$$

# Cloud Efficiency for Different EC2 Configurations

Consider a cluster configuration $\Lambda$. Let $T(1)$ be the execution time of an application code on a 1-ECU instance. Let $T(\Lambda)$ be the execution time of the same code on a virtual cluster $\Lambda$. The speedup is defined by $Speedup\ (\Lambda) = T(1) / T(\Lambda)$. Assume that the cluster is built with $n$ instance types. The type-$i$ has $n_i$ instances, each with an ECU count $c_i$. We calculate the total cluster ECU count by:

$$M(\Lambda) = \sum_{i=1}^{i=n} n_i \times c_i \qquad (5)$$

**Table 10.3**
The ECU rating of machine instance types in Amazon EC2 in 2014

| Instance Type | ECU | Virtual Cores | Memory (GB) | Storage (GB) | Price ($/hour) |
|---|---|---|---|---|---|
| ***m1.small*** | 1 | 1 | 1.7 | 1×160 | 0.044 |
| ***m1.medium*** | 2 | 1 | 3.7 | 1×410 | 0.087 |
| ***m3.medium*** | 3 | 1 | 3.75 | 1×4 SSD | 0.07 |
| ***m1.xlarge*** | 8 | 4 | 15 | 4×420 | 0.350 |
| ***m3.xlarge*** | 13 | 4 | 15 | 2×40 (SSD) | 0.280 |
| ***c1.xlarge*** | 20 | 8 | 7 | 4×420 (SSD) | 0.520 |
| ***c3.xlarge*** | 14 | 4 | 7.5 | 2×40 (SSD) | 0.210 |

# Cloud Efficiency for Different EC2 Configurations

This $N(\Lambda)$ count sets a ceiling of the cluster speedup. Now, we are ready to define the *cloud efficiency* for the cluster $\Lambda$ in question as follow:

$$Efficiency(\Lambda) = Speedup(\Lambda) / N(\Lambda)$$

$$= T(1) / \{T(\Lambda) \times \sum_{i=1}^{i=n} n_i \times c_i\} \quad (6)$$

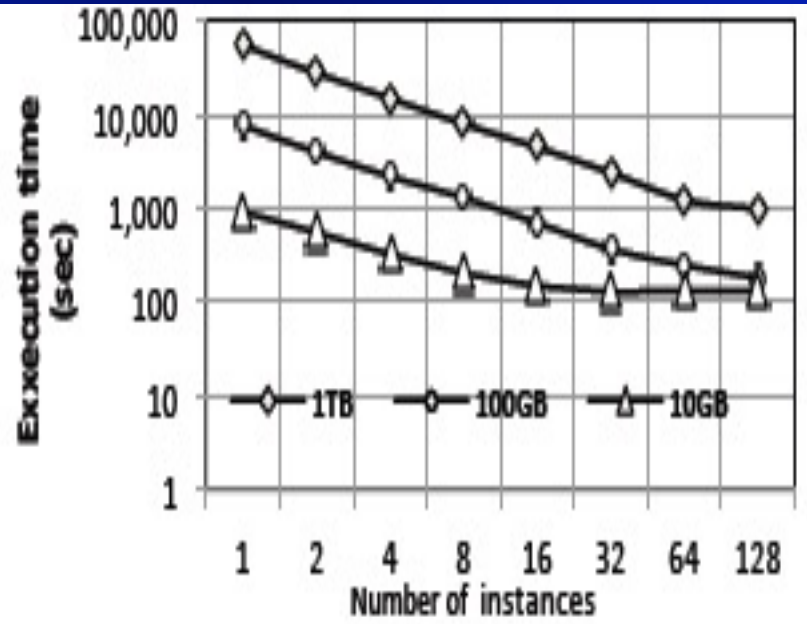
# Cloud Productivity

In general, the cloud *productivity* is driven by three technical factors that are related to the scaling factor.

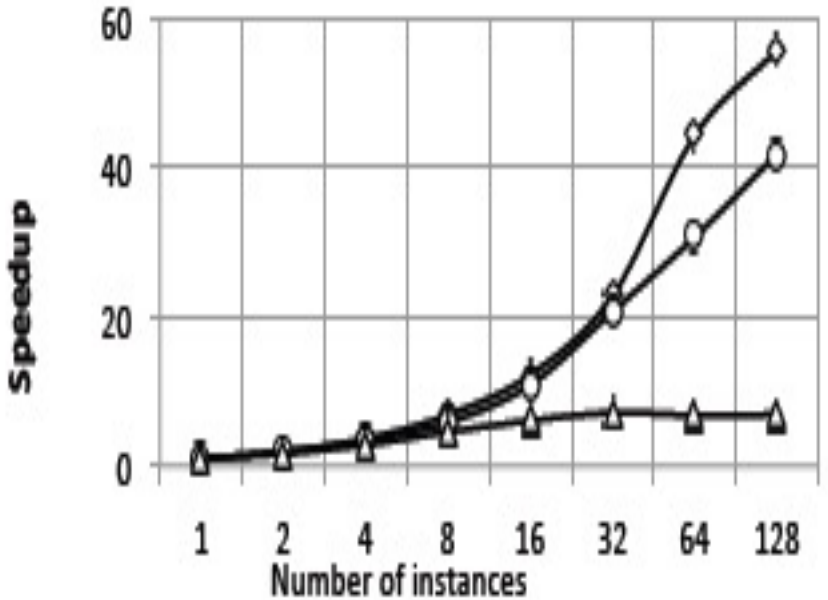
1) System performance such as throughput in terms of transactions per second or response time.

2) System availability as an indicator of QoS measured by percentage of uptime.

# Twitter Spam Filtering Results on EC2



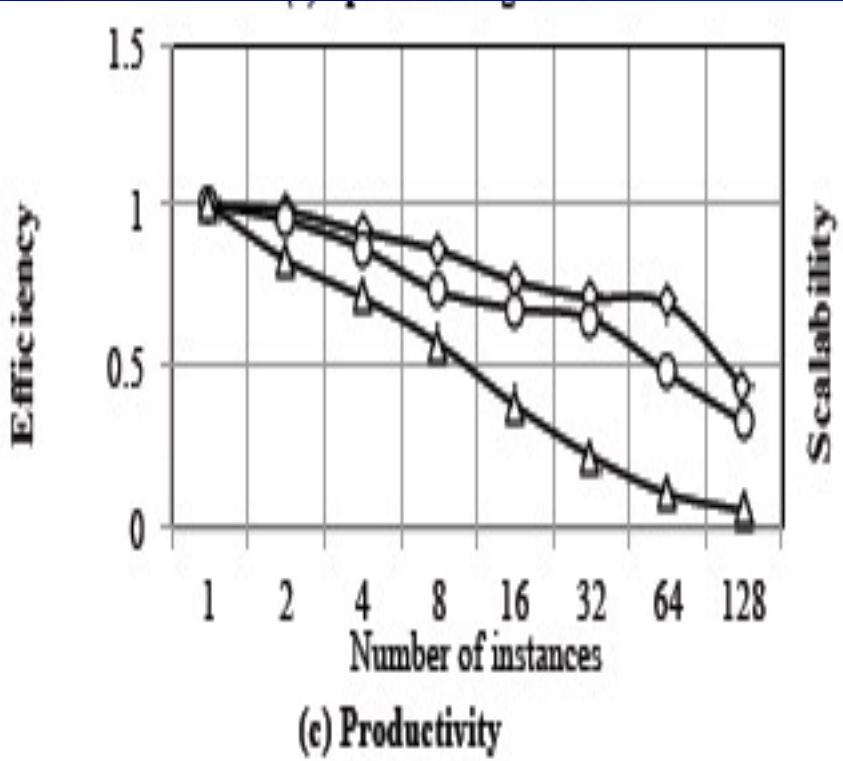

(a) Spam filtering time



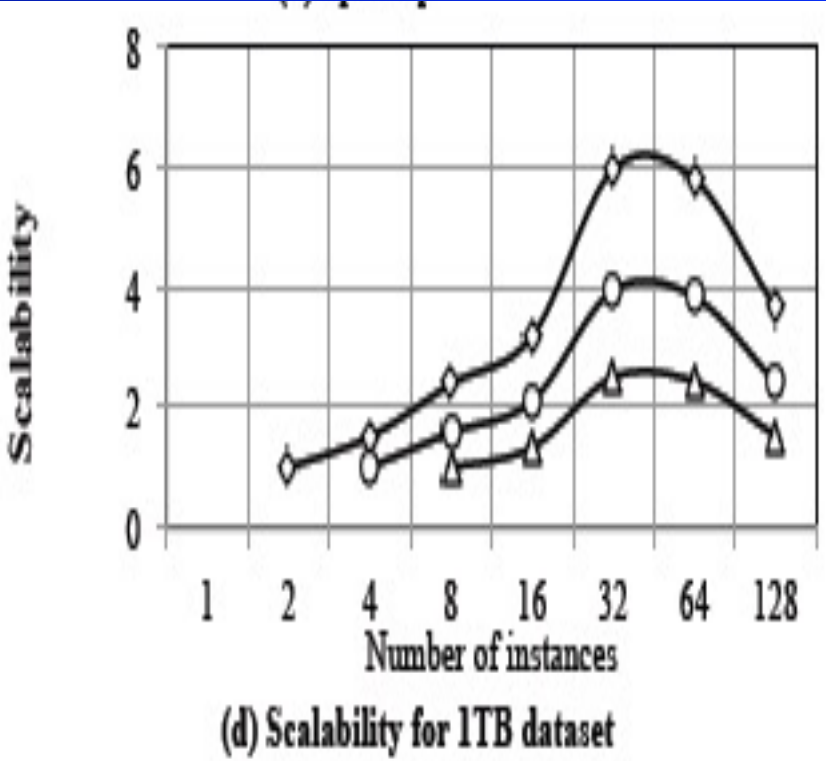

(b) Speedup

Figure 5. Scale-out BenchClouds results on MapReduce filtering twitter spams over AWS EC2 of various sizes. Parts (a, b, c) apply the same legend. Part &d) shows both scalability measures by scaling from 3 distinct instances.

# Twitter Spam Filtering on EC2




(c) Productivity




(d) Scalability for 1TB dataset

Figure 5. Scale-out BenchClouds results on MapReduce filtering twitter spams over AWS EC2 of various sizes. Parts (a, b, c) apply the same legend. Part &d) shows both scalability measures by scaling from 3 distinct instances.

**Table 10.4**
Scaling techniques based on HiBench benchmarking findings on the EC2

| Impact Factors | Scale-Out Technique | Scale-Up Technique | Mixed Scaling |
|---|---|---|---|
| **Elasticity Speed, Scaling Complexity, and Overhead** | Fast elasticity, possibly supported by auto-scaling and heuristics | High overhead to reconfigure and cannot support auto-scaling | Most difficult to scale with wide range of machine instances |
| **Effects on Performance, Efficiency, and Scalability** | Expect scalable performance if the application can exploit parallelism | Switching among heterogeneous nodes may reduce scalability | Flexible app, low efficiency and resource utilization |
| **QoS, Costs, Fault Recovery, and Cloud Productivity** | Cost the least, easy to recover, incremental productivity | More cost-effective, but reduced QoS may weaken productivity | High costs, difficult to recover, expect the highest productivity |

Table 2 Over-all Performance Demonstrated by Polygon Area on Radar Charts in Fig.5

| | | | |
|---|---|---|---|
| Scale-Out Mode (Figs.5a, d) | *Cluster Configurations* | *2 small nodes* | *16 small nodes* |
| | WordCount | 34.53 | 46.85 |
| | Sort | 17.02 | 23.65 |
| Scale-Up Mode (Figs.5b, e) | *Cluster Configurations* | *2 medium nodes* | *2 xlarge nodes* |
| | WordCount | 37.25 | 31.42 |
| | Sort | 41.84 | 21.22 |
| Mixed Scaling Mode (Figs. 5c, f) | *Cluster Configurations* | *4 medium and 4 small* | *3 large and 2 xlarge* |
| | WordCount | 23.39 | 18.28 |
| | Sort | 22.81 | 11.90 |

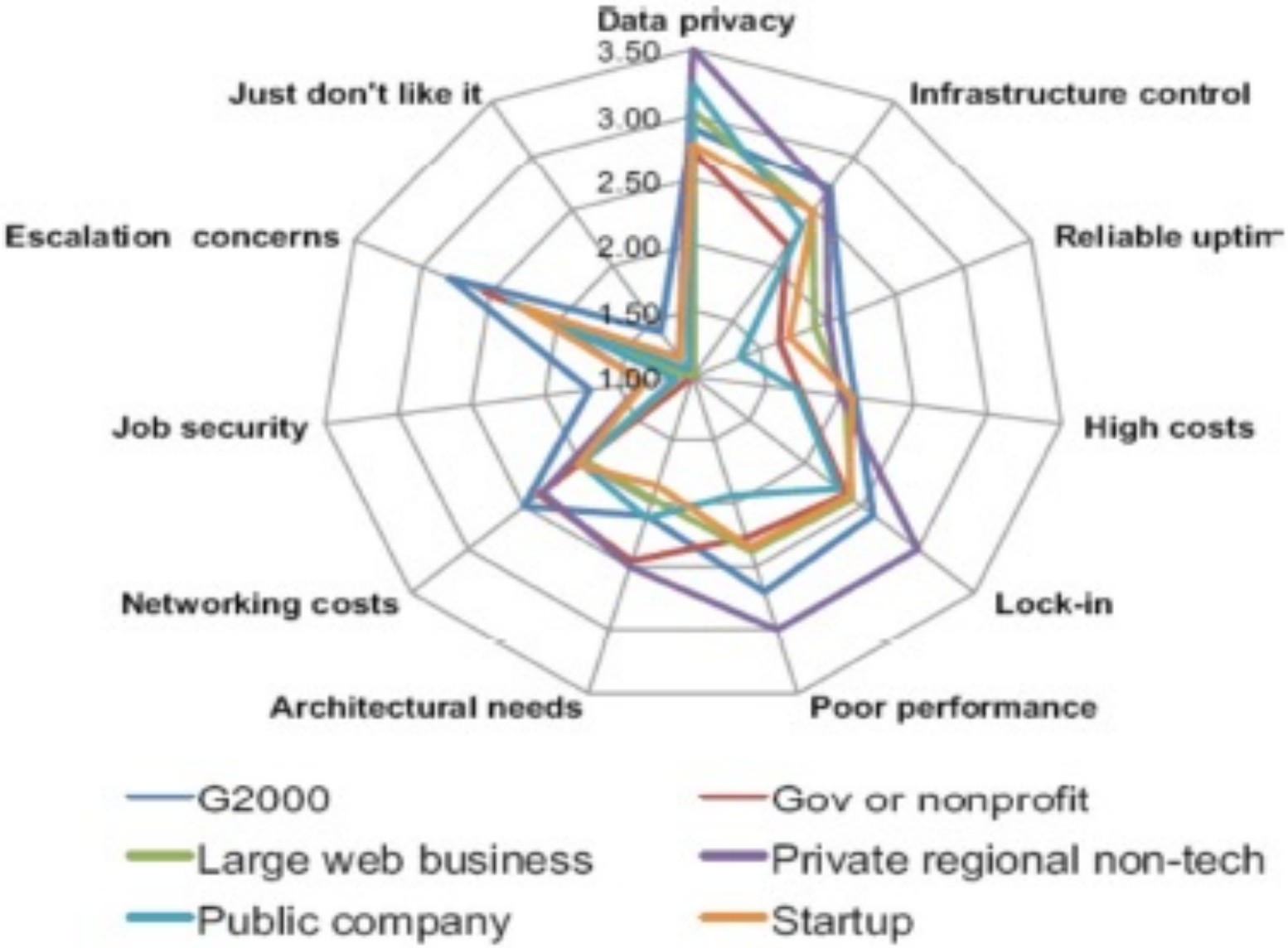
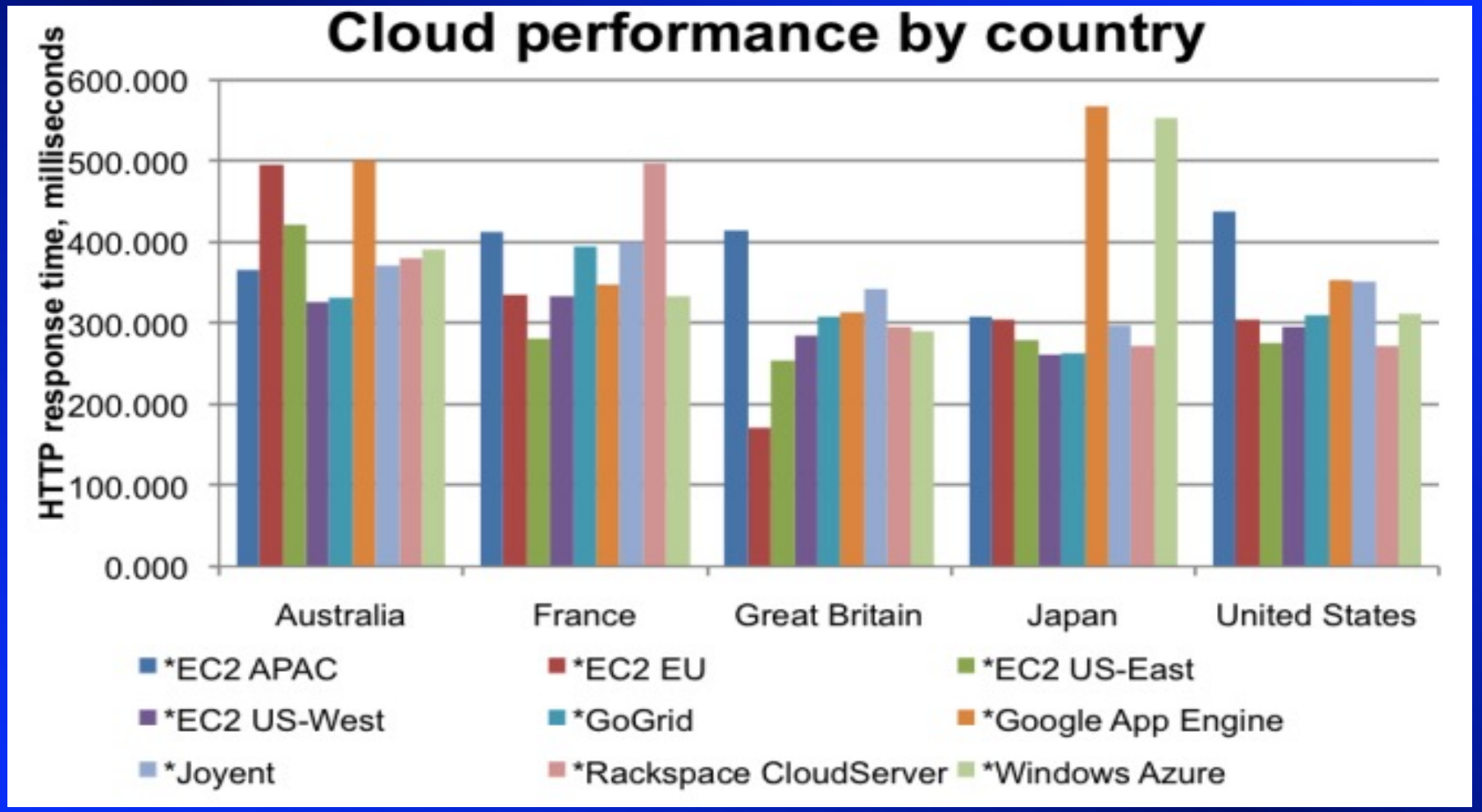
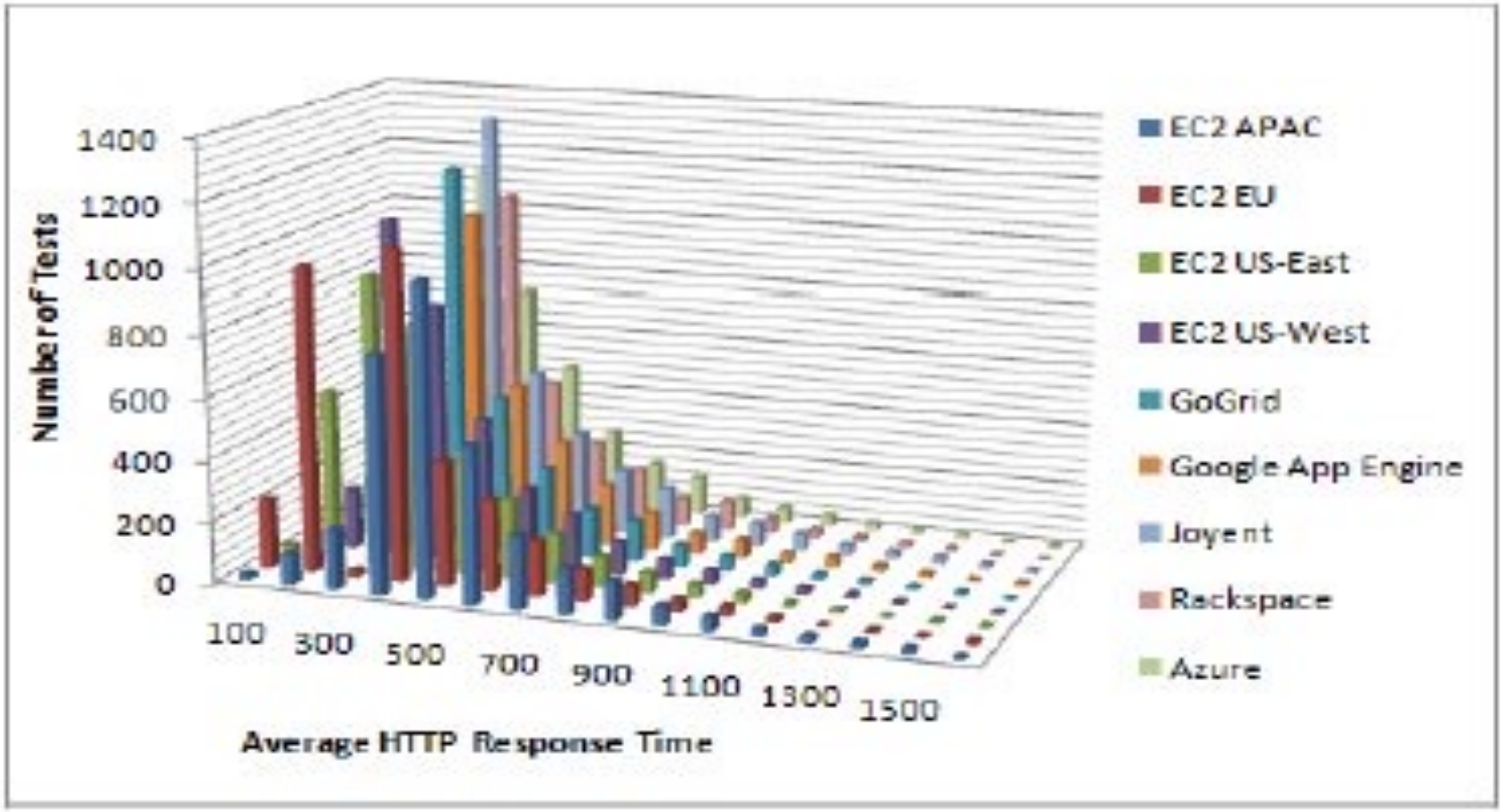



Figure 3 An example radar chart for expressing 11 concerns by 6 cloud user groups (Courtest by Bittorrent, Inc. 2010 [12])






**Average response times in ms of 9 cloud service reported in Bitcurrent Report (Courtesy of A. Croll [27])**

# HTTP Response Time of Working Clouds

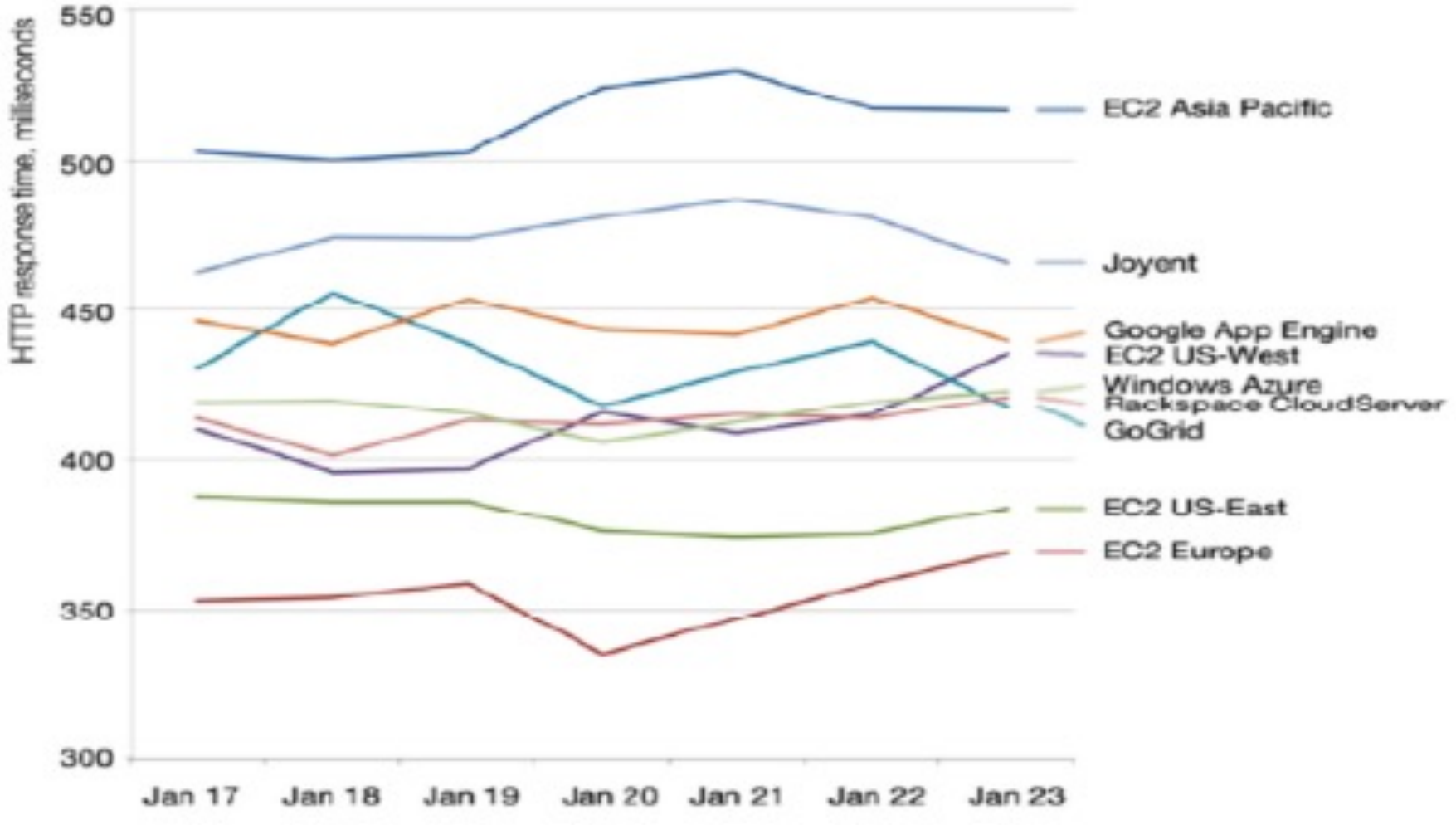
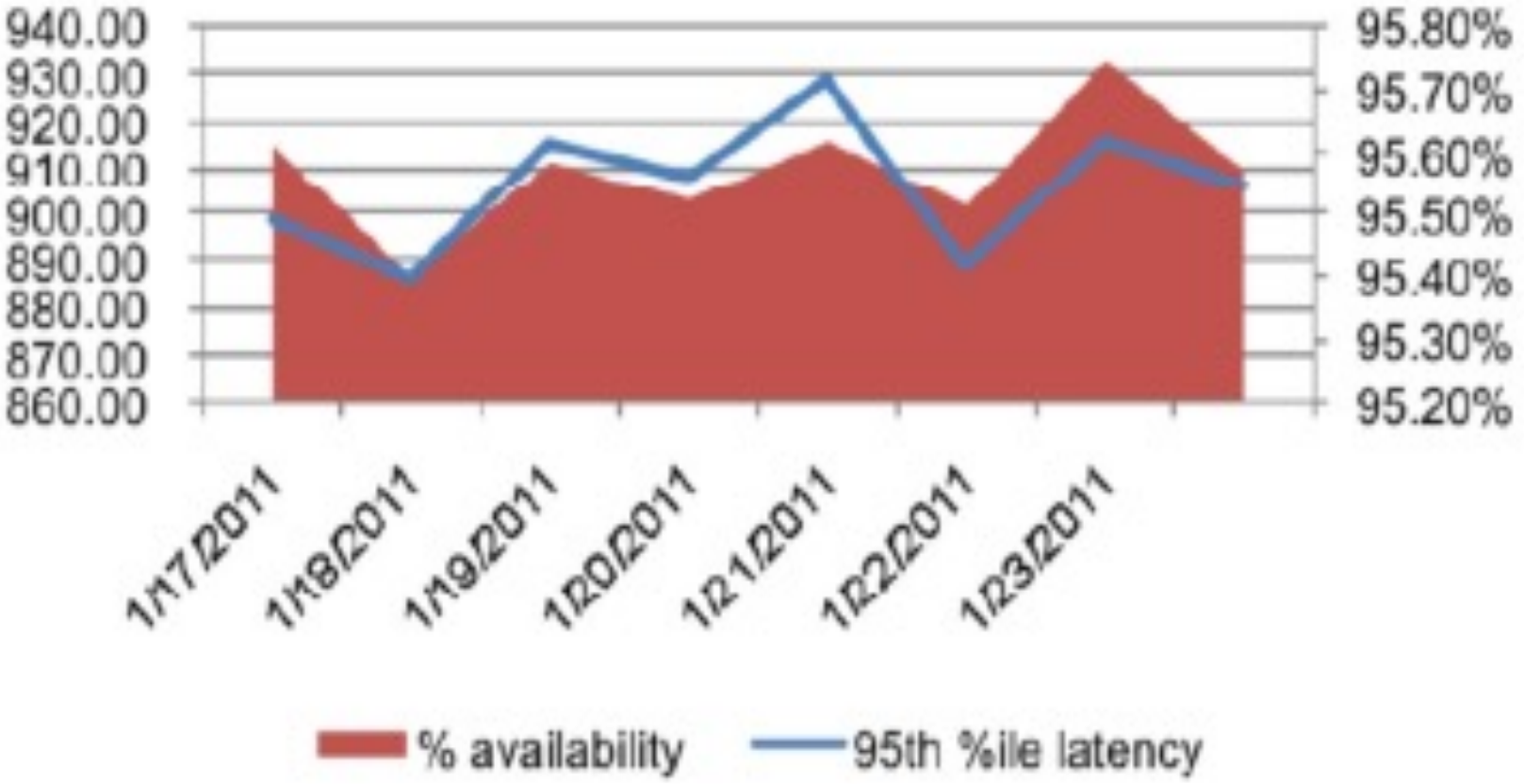
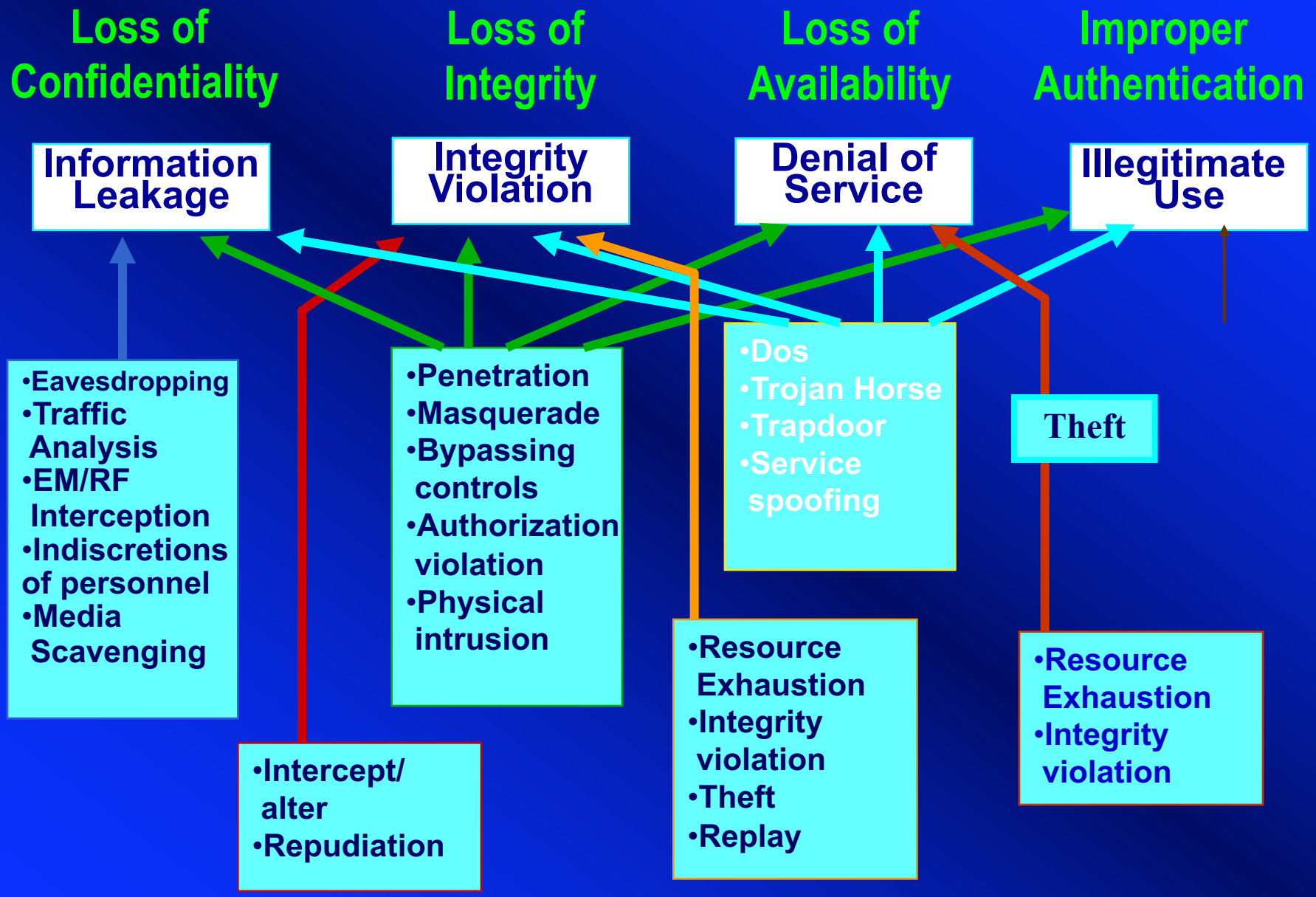



**Figure 7 Average HTTP response times in ms of nine cloud service sites in various regions of the world** (Courtesy of Bitcurrent 2011, [19])




(a) Average performance and availability of 9 cloud providers

# Cyber Security Threats



# Cyber Security: Critical Issues and Plausible Solutions

- Cyber threats and system vulnerability
- Internet security issues and infrastructures
- Automated Worm Signature Generation
- Collaborative DDoS Defense over multiple domains
- Copyright Protection in P2P Content Delivery
- P2P Reputation Systems for Internet Security
- Cloud Resources Protection and Data Coloring
- Trusted Zoning in Virtualized Cloud Resources
- Trust Chain for Cloud and IoT Mashup Services
- Cloudlet Mesh for Securing Mobile Clouds

# Distributed Security Enforcement over a DHT Overlay Network Developed at the USC

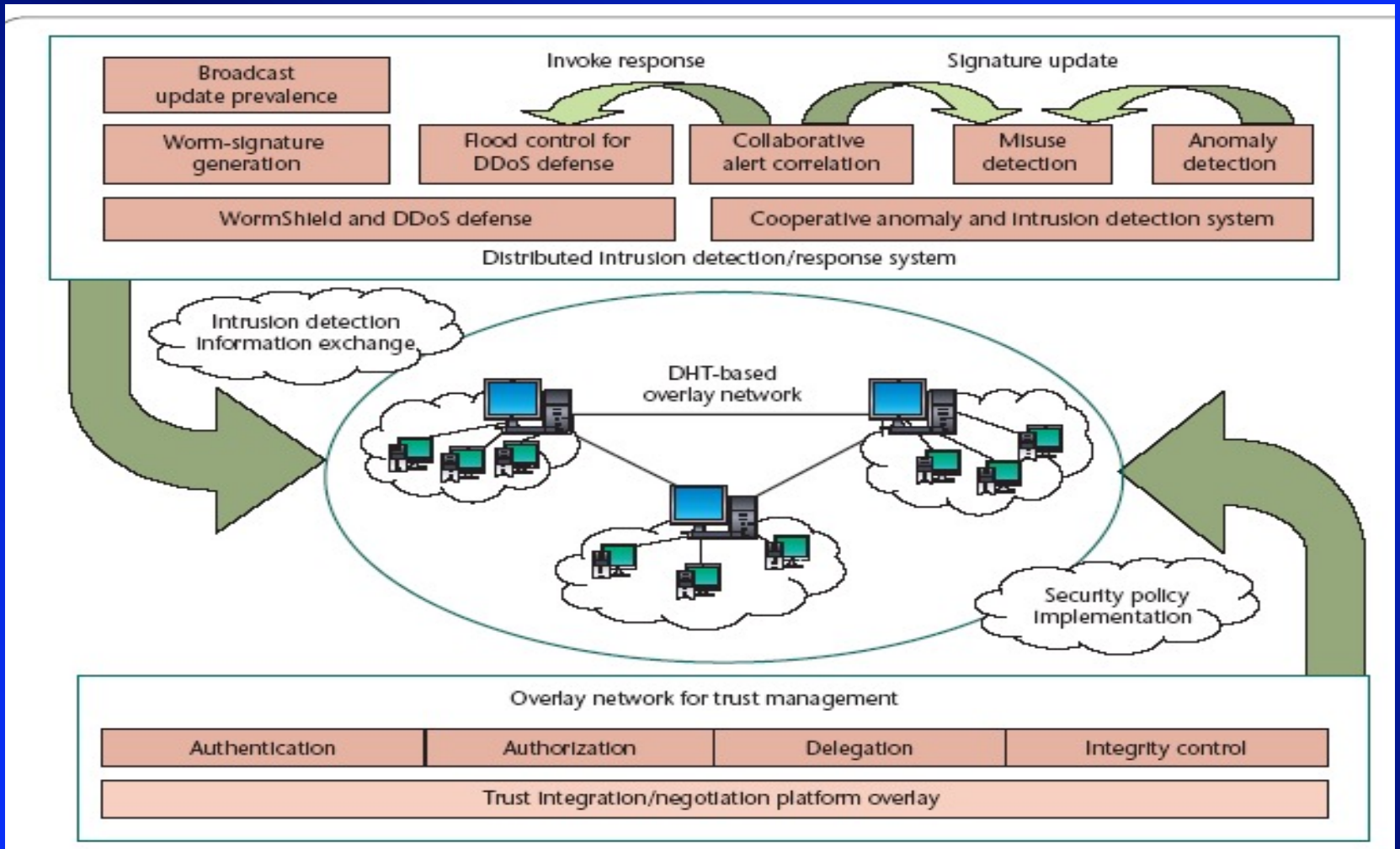
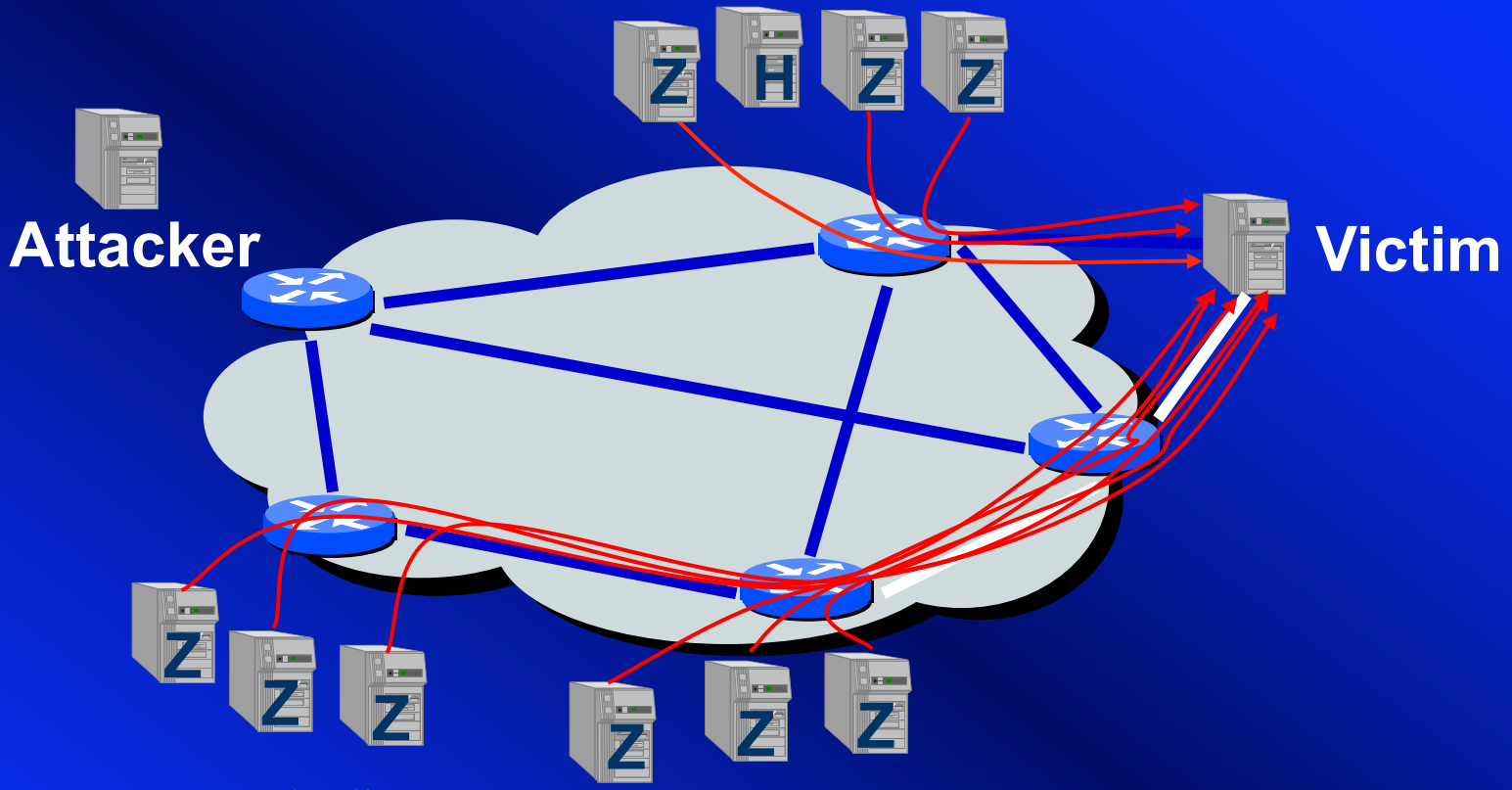


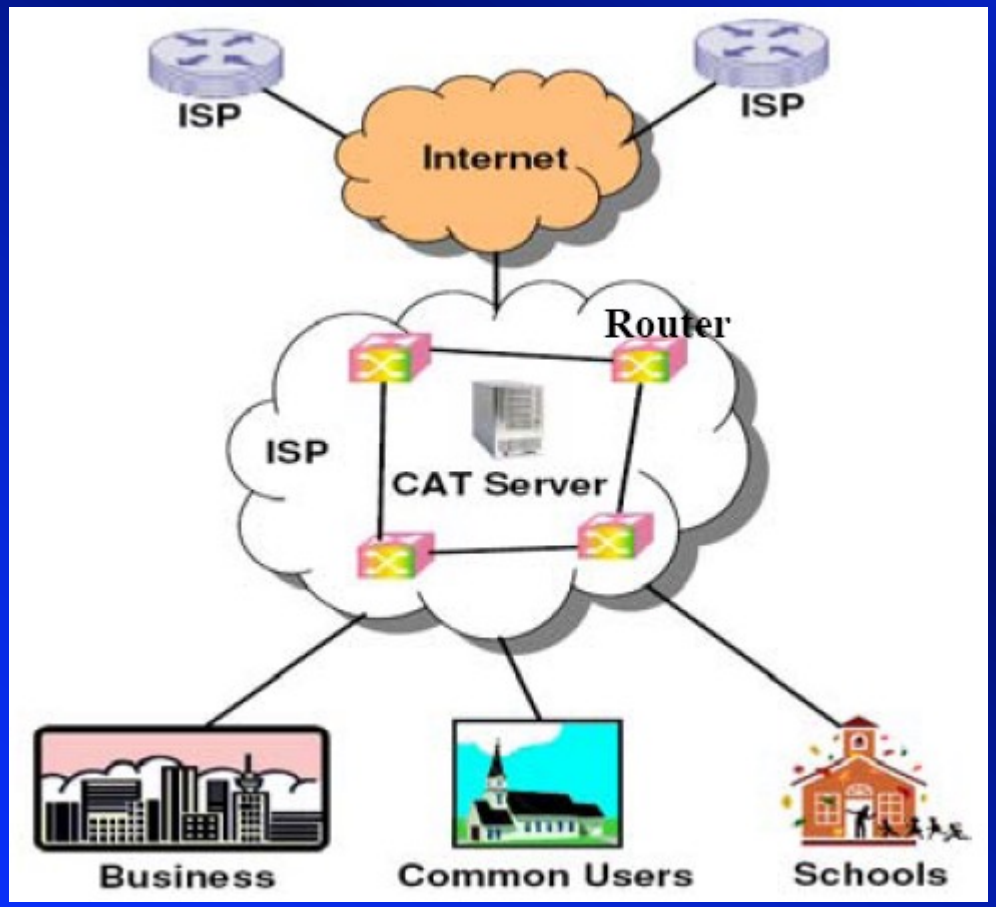

Figure 1. The NetShield system architecture and trust integration over a distributed hash table (DHT) overlay. The system performs trust management across multiple administrative domains, suppresses Internet worm outbreaks, and defends against flooding DDoS attacks.

# Distributed DoS (DDoS) Attacks



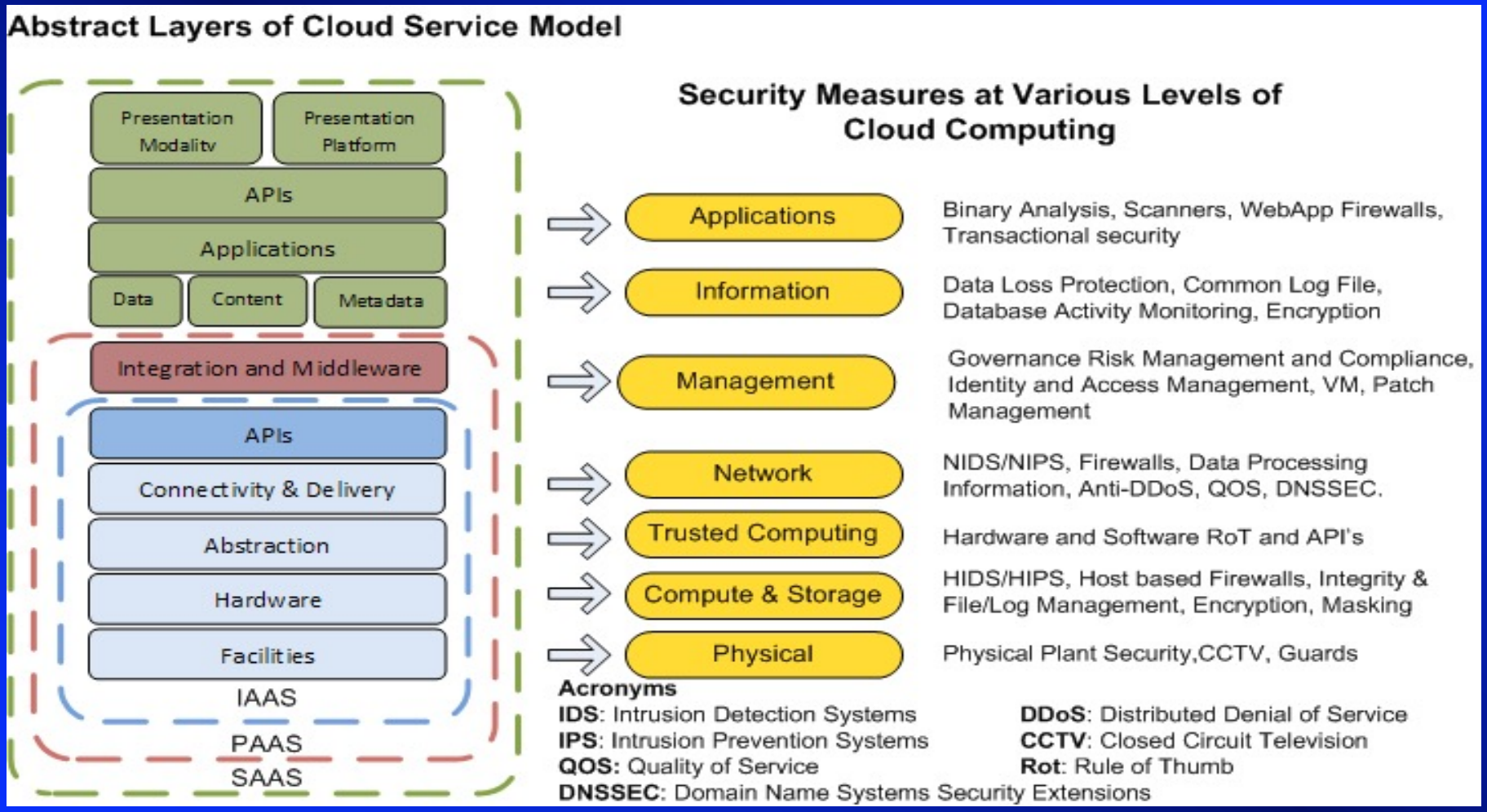

1. Attacker infiltrates hosts and commands a handler (H)
2. Handler sends commends to zombies (Z)
3. Zombies attack the victim, damages to CPU/Memory resources and to the network bandwidth resources

# Collaborative Detection Scheme




- Routers monitor the traffic and periodically report the individually observed pattern to the central CAT server
- The central CAT server analyzes the collected patterns and try to construct a CAT tree
- Once a CAT tree formed, we can detect the starting of a DDoS flooding attack

# Cloud Service Models and Their Security Demands



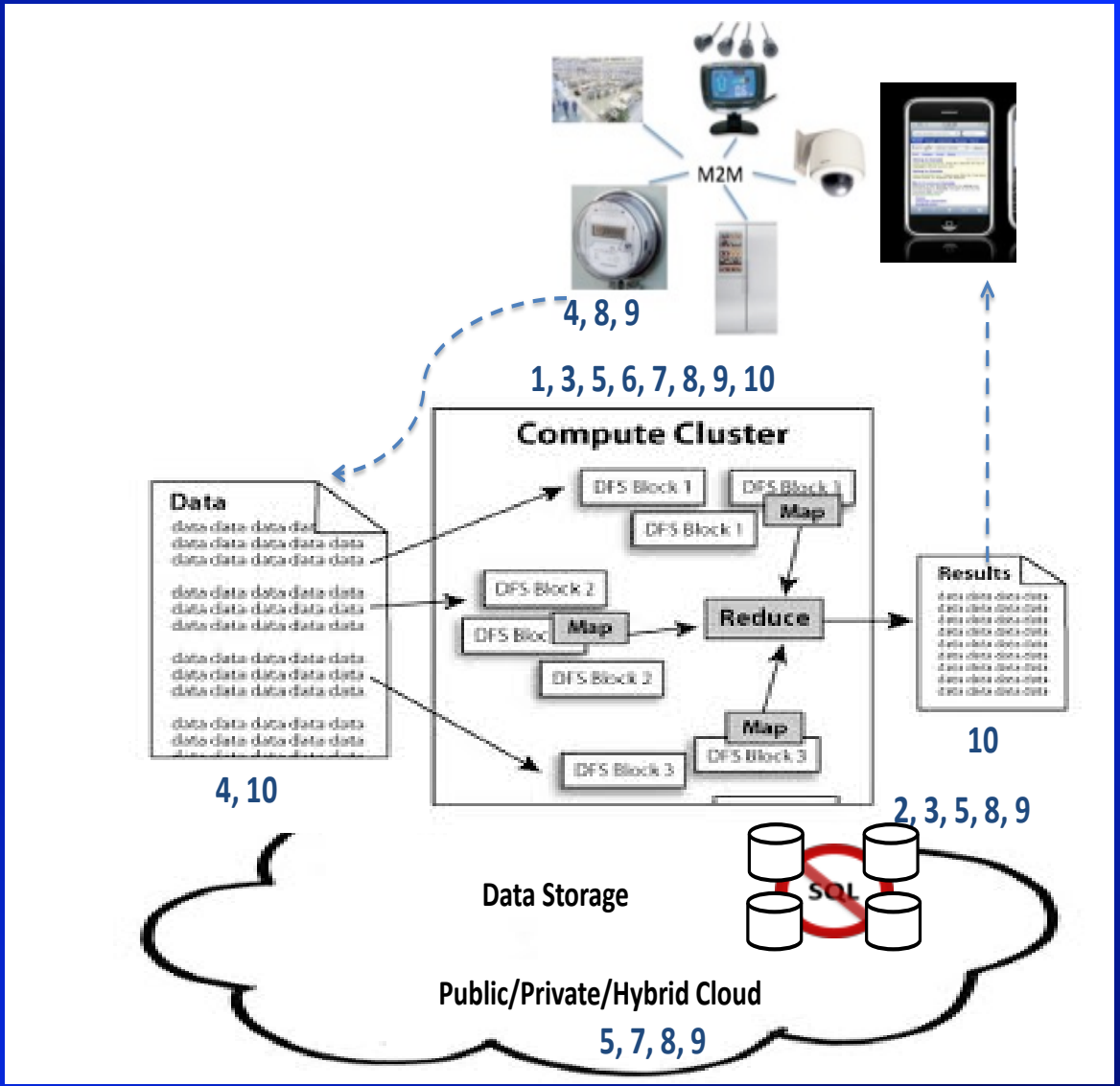

**Source:** K. Hwang and D. Li, " Trusted Cloud Computing with Secure Resources and Data Coloring", *IEEE Internet Computing,* Vol.14, Sept. 2010.

# CSA (Cloud Security Alliance): Top Ten Big Data Security and Privacy Challenges

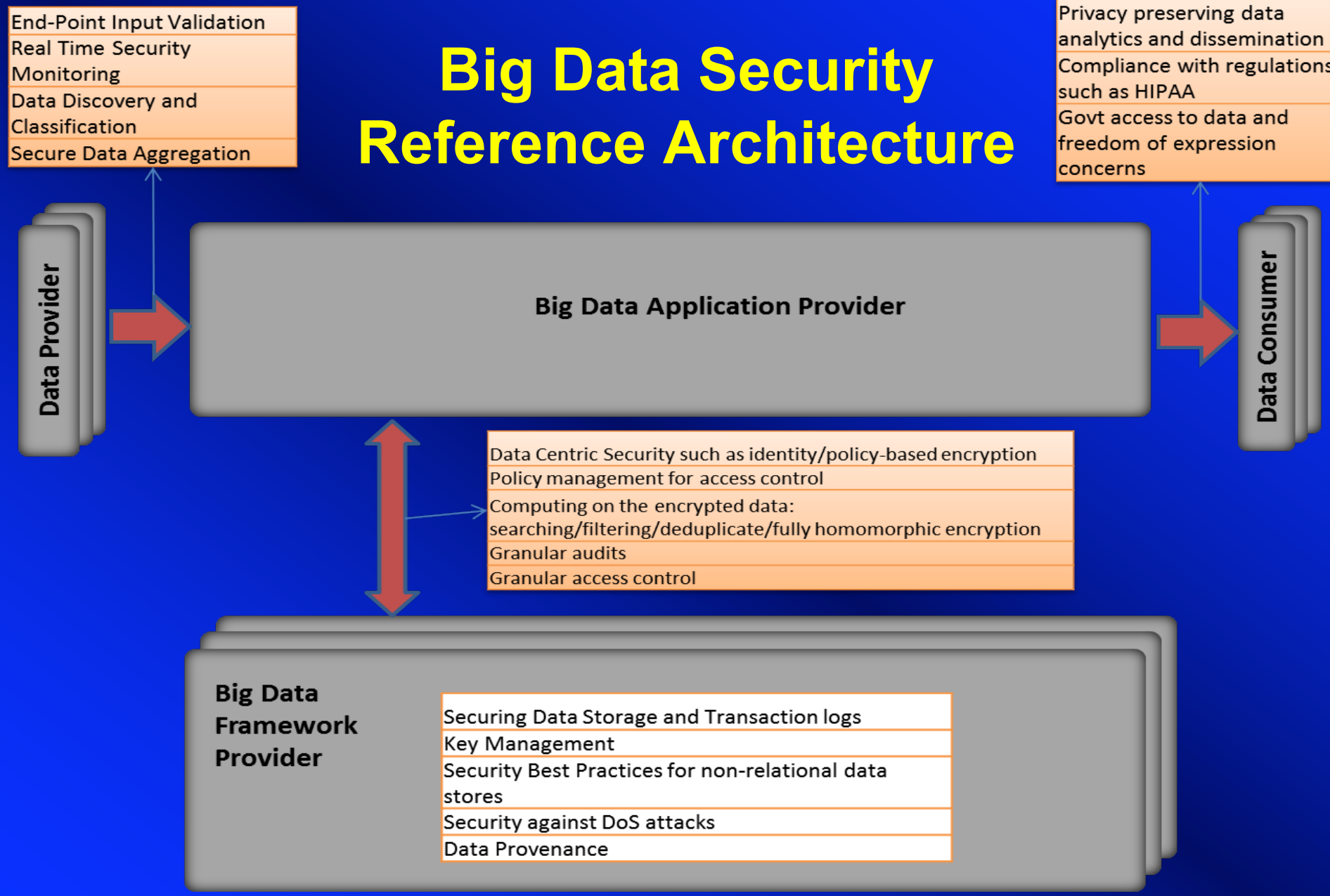
1. Secure computations in distributed programming frameworks
2. Security best practices for non-relational datastores
3. Secure data storage and transactions logs
4. End-point input validation/filtering
5. Real time security monitoring

# CSA Top Ten Big-Data S/P Challenges

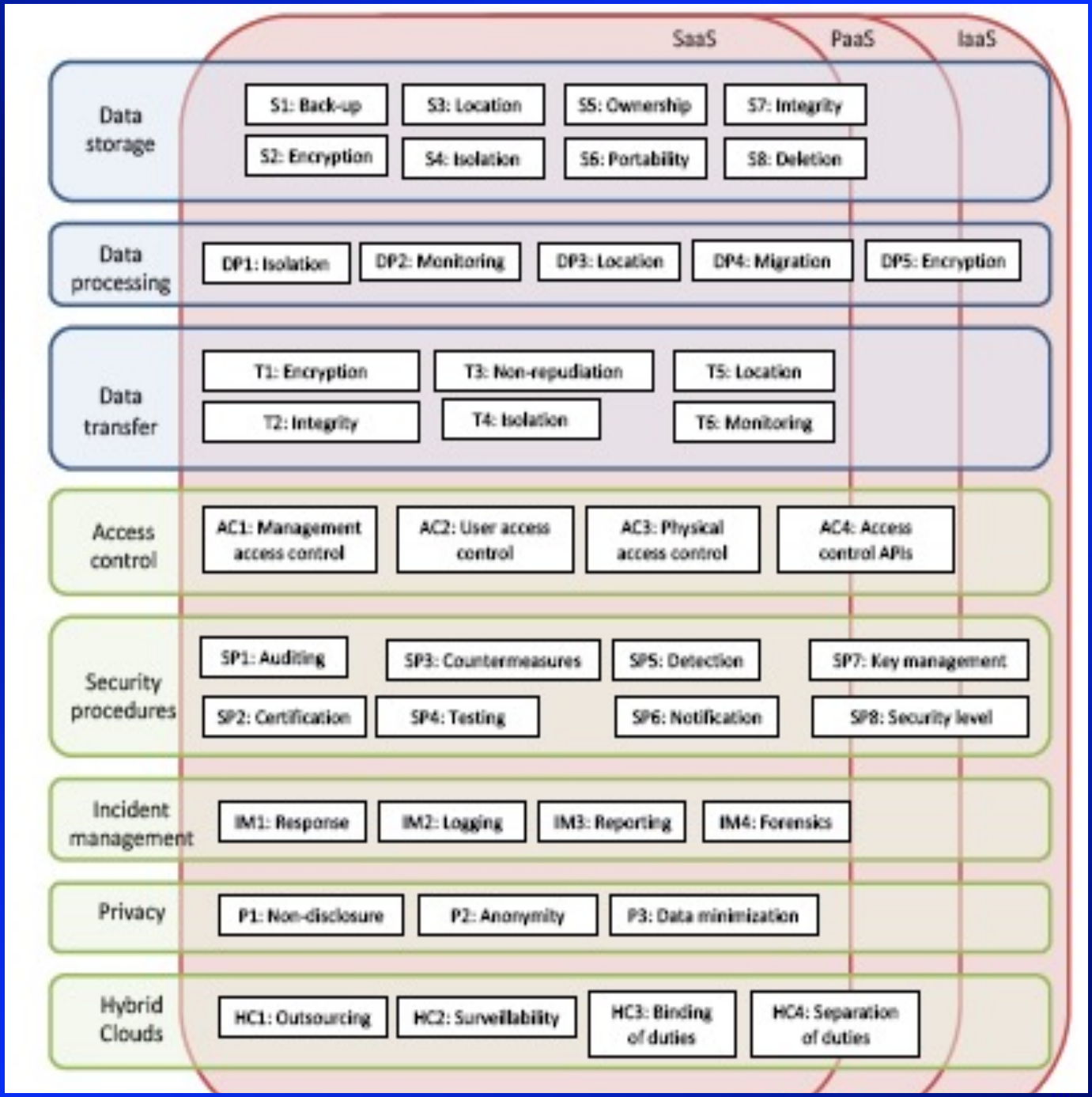
6. Scalable and composable privacy-preserving data mining and analytics
7. Cryptographically enforced access control and secure |communication
8. Granular access control
9. Granular audits
10. Data provenance



# Big Data Security Reference Architecture



# Big Data Security in Clouds



# The Future of a Trusted Cloud Computing Environment

- Due to economies of scale, a cloud provider can have a dedicated team of security specialists and cloud datacenters have physical protection on par with military installations
- Centralized data management also has its benefits — provided that one can manage the risk of data losses, portability issues, etc.
- In 2015, 10% of the overall IT security product capabilities will be delivered in the cloud – projection by Telenor Research, October 2013

$Ex' = \bar{X}$

# Security and Trust Barriers in Mobile Cloud Computing

- Protecting datacenters must first secure cloud resources and uphold user privacy and data integrity
- We suggested the use of a trust overlay network to build reputation systems for trusted cloud computing
- A watermarking technique is suggested to protect shared data objects and massively distributed software modules
- These techniques safeguard user authentication and tighten the data access-control in public clouds
- The new approach could be more cost-effective than using the traditional encryption and firewalls

# Concluding Remarks

- *Scalable killer applications* on Cloud/IoT systems
- Collaborative Warm Containment in The Internet
- *DDoS flooding defense* over multiple network domains
- *Copyright protection* in P2P content delivery
- P2P Reputation Systems for Cloud/Web Services
- Trusted Zoning for VM Clusters in Public Clouds
- Big Data Analytics for Security Alert Systems