Cloud Computing

Computer Systems (CS2000)

Trimester 2 2020

Overview

- What is meant by
  - Cloud Computing
  - Utility Computing
  - {Infrastructure, Platform, Software} as a Service
- Why do corporations need to pay attention
- General principles
- Research

NIST Definition

- **July 5, 2011**
- *The NIST Definition of Cloud Computing identified cloud computing as:*

  a model for enabling ubiquitous, convenient, ondemand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Cloud Characteristics

- On-demand self-service
- Ubiquitous network access
- Location independent resource pooling
- Rapid elasticity
- Pay per use

Replace Delivery Models

- The IEEE defines several types of public cloud computing:[1]
  - Infrastructure as a service (IaaS)
  - Platform as a service (PaaS)
  - Software as a service (SaaS)
  - Storage as a service (STaaS)
  - Security as a service (SECaaS)
  - Data as a service (DaaS)
  - Test environment as a service (TEaaS)
  - Desktop as a service (DaaS)
  - API as a service (APIaaS)

Monaco, Ania (7 June 2012 [last update]). "A View Inside the Cloud". theinstitute.ieee.org (IEEE). Retrieved September, 2012.

Software Stack

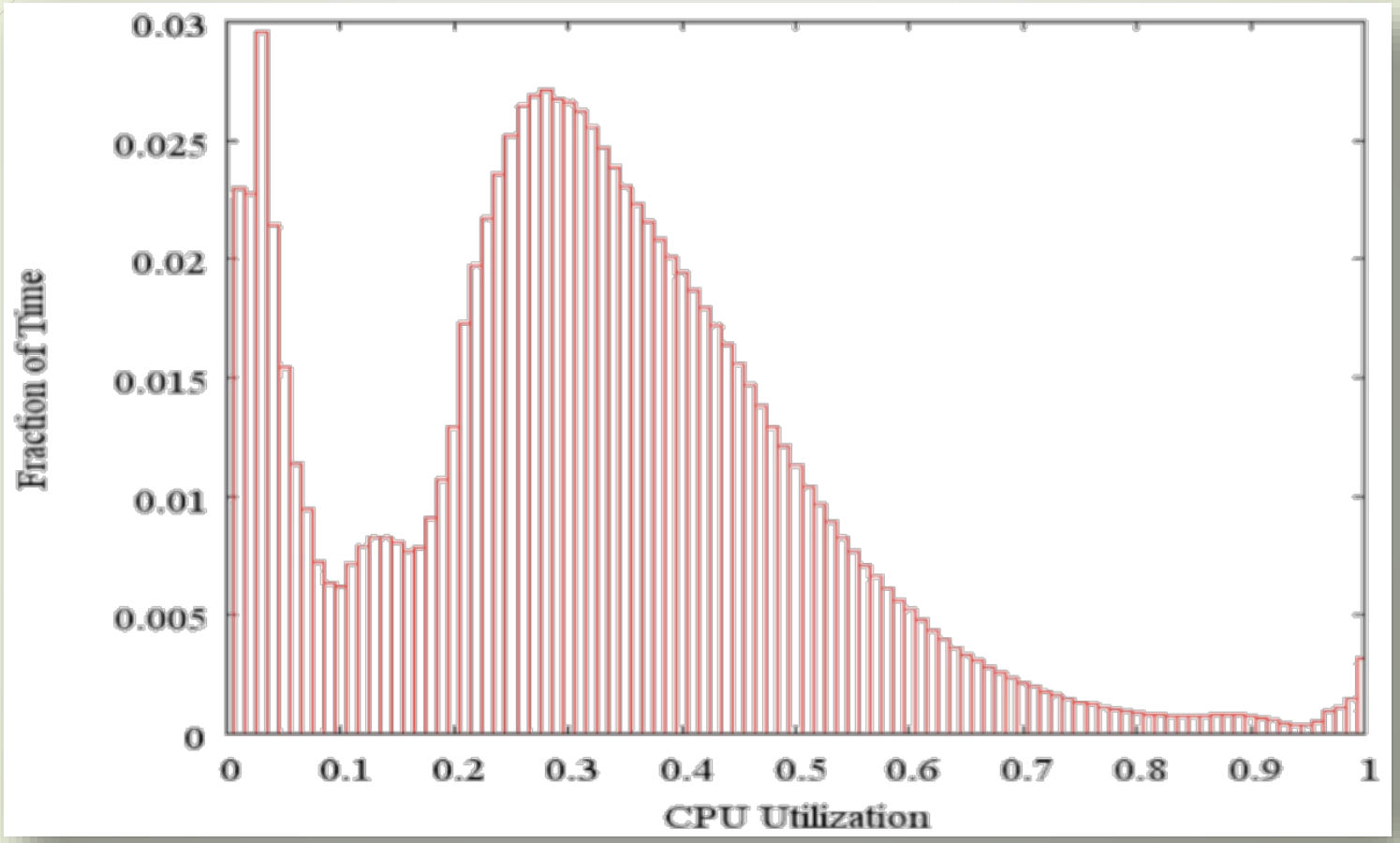
- Mobile (Android), Thin client (Zonbu) Thick client (Google Chrome)
- Identity, Integration Payments, Mapping, Search, Video Games, Chat
- Peer-to-peer (Bittorrent), Web app (twitter), SaaS (Google Apps, SAP)
- Java Google Web Toolkit, Django, Ruby on Rails, .NET
- S3, Nirvanix, Rackspace Cloud Files, Savvis,
- Full virtualization (GoGrid), Management (RightScale), Compute (EC2), Platform (Force.com)

Perils of Corporate Computing

- Own information systems
- However
  - Capital investment
  - Heavy fixed costs
  - Redundant expenditures
  - High energy cost, low CPU utilization
  - Dealing with unreliable hardware
  - High-levels of overcapacity (Technology and Labor)

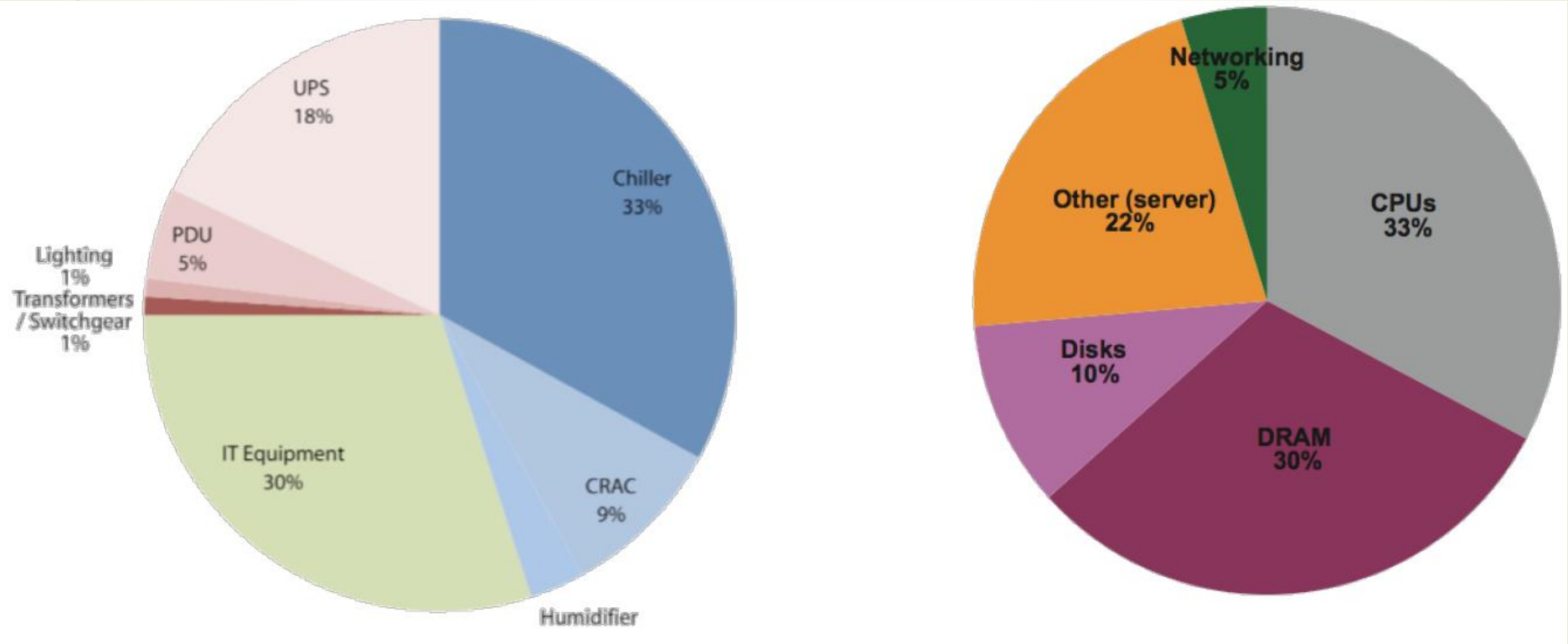
NOT SUSTAINABLE ?

Google: CPU Utilisation



Activity profile of a sample of 5,000 Google Servers over a period of 6 months

Google: Energy Overhead



Utility Computing

- Let economy of scale prevail
- Outsource all the trouble to someone else
- The utility provider will share the overhead costs among many customers, amortising the costs
- You only pay for:
  - the amortised overhead
  - Your real CPU / Storage / Bandwidth usage

Cloud Interoperability Standards

- Open Cloud Computing Interface – Infrastructure
- EC2 API
- Simple Storage Service (S3) API
- Windows Azure Storage Service REST APIs
- Windows Azure Service Management REST APIs
- Rackspace Cloud Servers API
- Rackspace Cloud Files API
- Cloud Data Management Interface
- vCloud API
- GlobusOnline REST API

Web Services in the Cloud

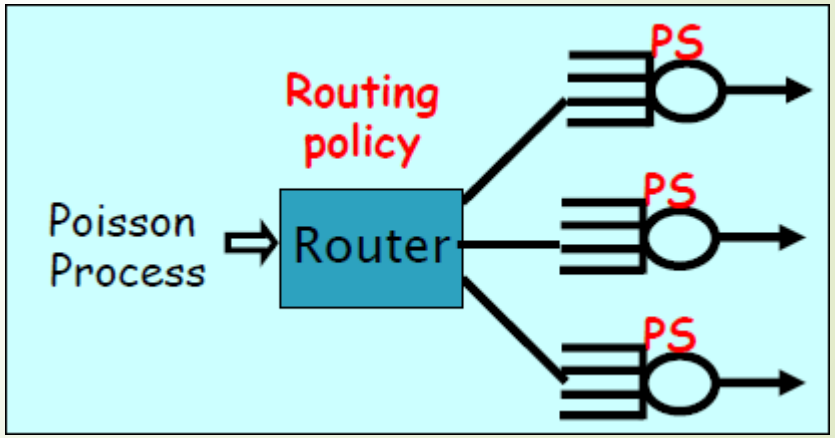
- What are web services in the cloud?
- How are they different from traditional web services?
- What challenges are there?

Web Services in the Cloud: User End

- Search engines: Google, Bing
- Social networks: Facebook, Twitter
- Collaborative filtering: Yelp
- Maps / GPS: GoogleMap, MapQuest
- On 24/7
- Accessed through Internet, often http
- Interactive applications
- Mobile devices
- Fast response time is crucial
- Not very different from web services

Web Services in the Cloud: Provider End

- It is very different from traditional web services
- A traditional web service cluster
  - 10s – 100 servers
  - Serve the same content
  - Most work in putting bits on the pipe from the server one is connected to (not much computation involved)



Web Service in the Cloud: Provider End

- Web services in the Cloud
  - 100s – 1000 servers
  - Content is different for different user (personalised, on-demand)
  - A lot of work goes into the computation to extract the information from large amount of data

Web Services in the Cloud

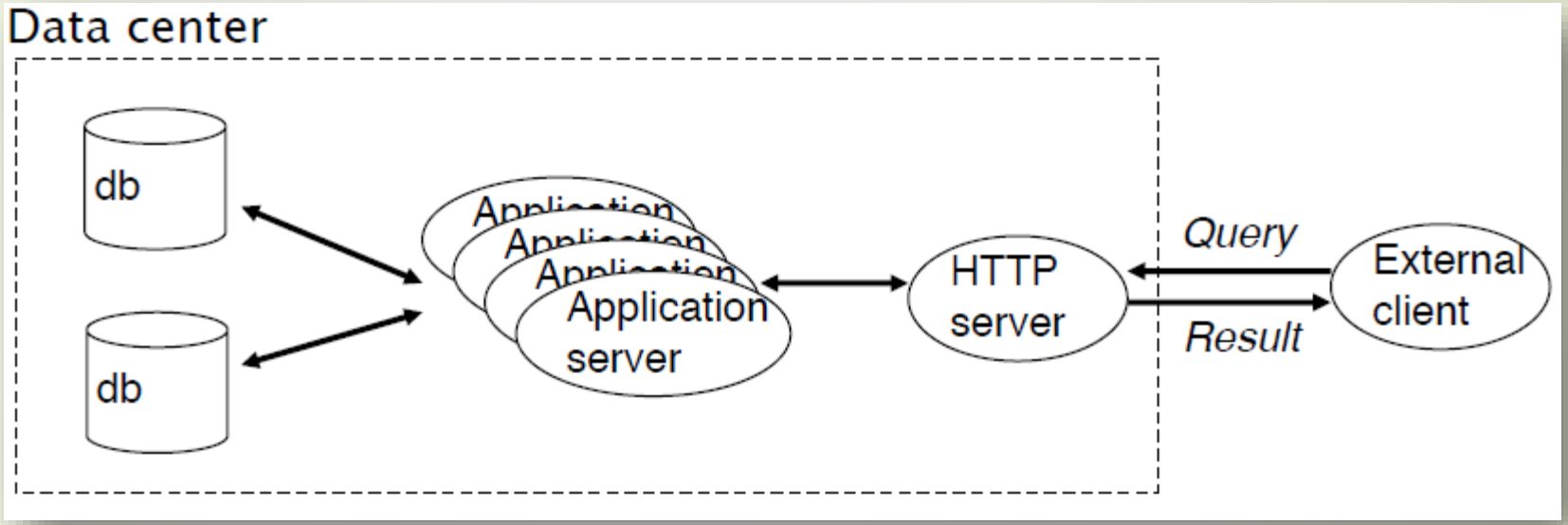
- Cloud providers:
  - Amazon, Google, IBM, Microsoft
- Content providers move to the cloud
  - Dynamic scaling: pay-as-you-go
  - Multi-tenant environment
- Currently
  - Design and operation are ad hoc
  - Excess over-provisioning to ensure good response time, does not scale

Challenges

- Distributed load balancers
- Services and request model
- Multi-tenant environment
- Multi-tier architecture
- Dynamic scaling
- Persistent connections
- Data locality problem
- Data partitioning problem

- Scheduling and routing problems
- Scheduling: which task to process first
- Routing: where to direct a task in a server farm
- Algorithm design
- Simulation
- Analysis
- Implementation

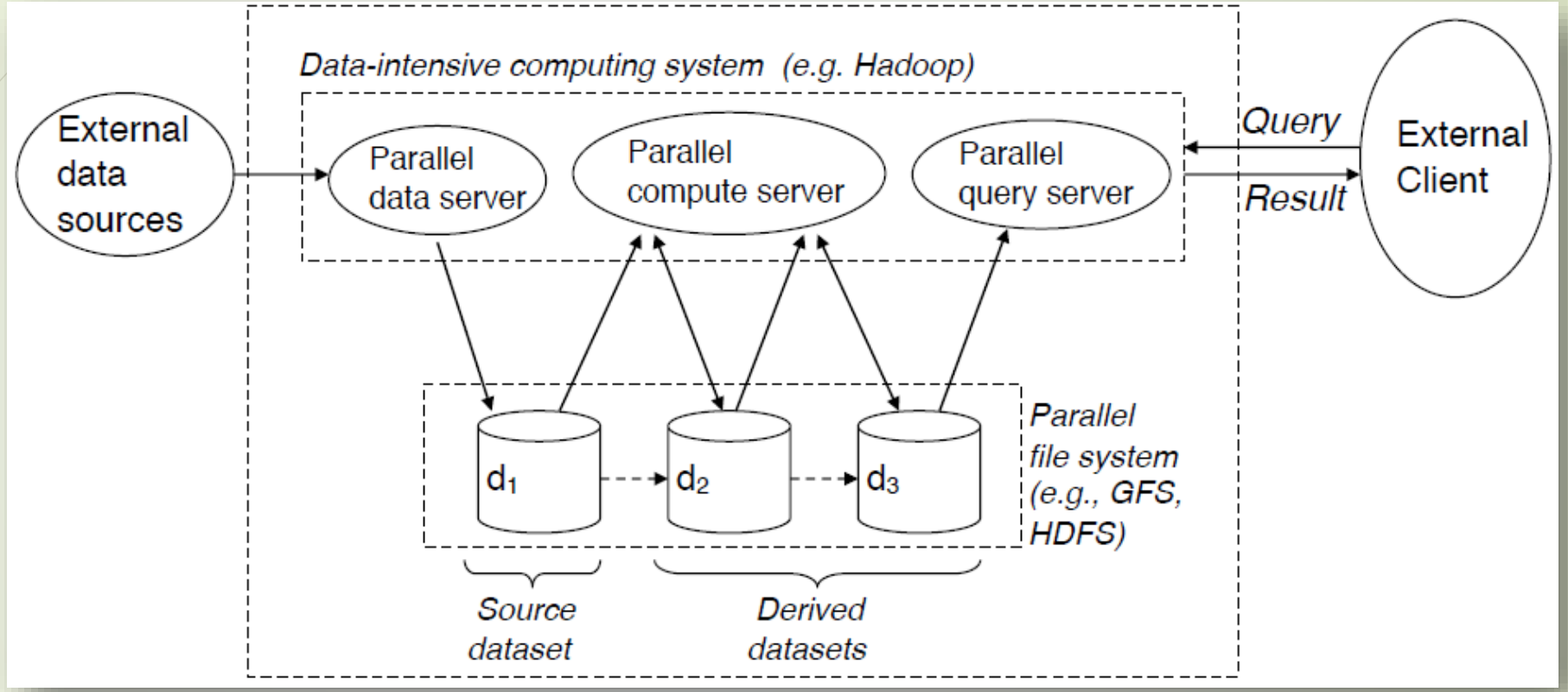
Typical Web Service



Characteristics:
Small queries and results
Little client computation
Moderate server computation

Examples:
Web sites serving dynamic content

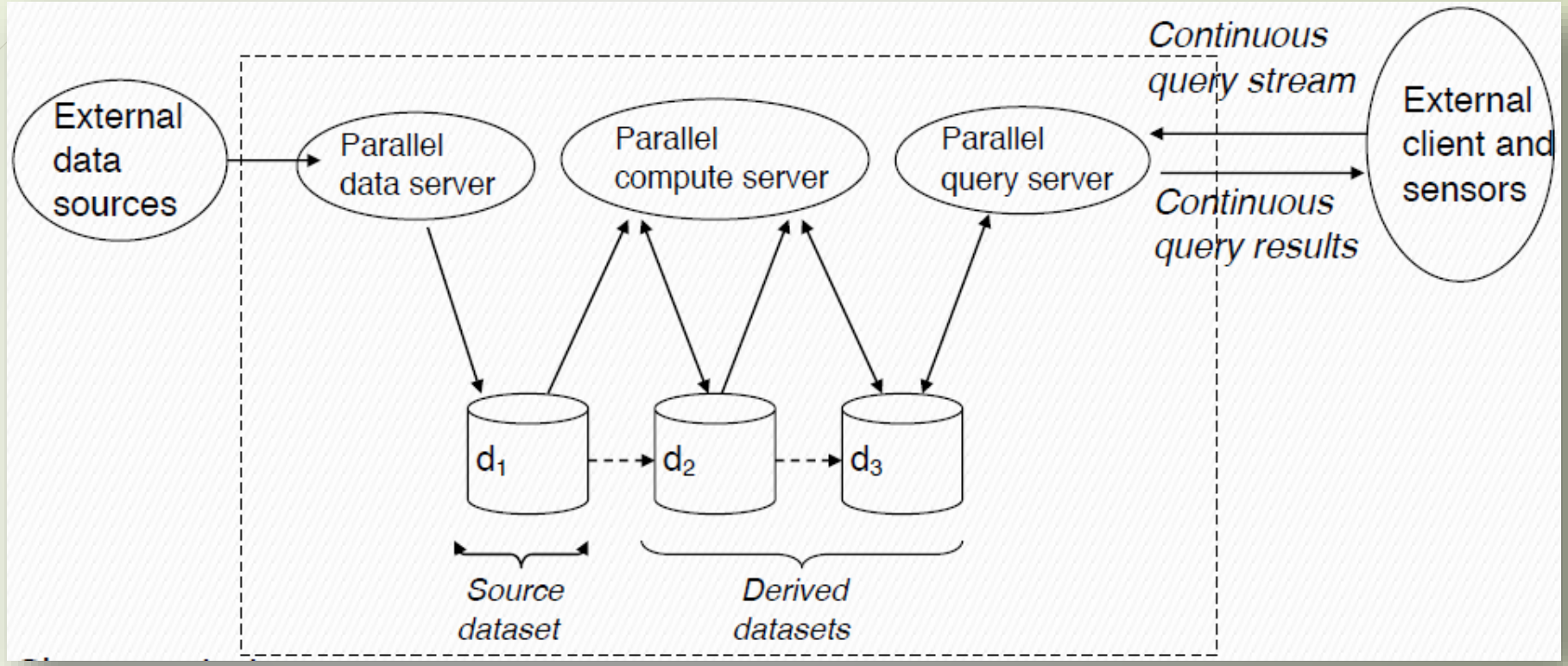
Big Data Service



Characteristics:
Massive data and computation on server, small queries and results

Examples:
Search, scene completion service, log processing

Streaming Data Service



Characteristics:
Application lives on client
Client uses cloud as an accelerator
Highly variant, latency sensitive HPC on server
Often combines with Big Data service

Examples:
Computational perception on high data-rate sensors: real time brain activity recognition, food recognition, activity recognition, object rec.

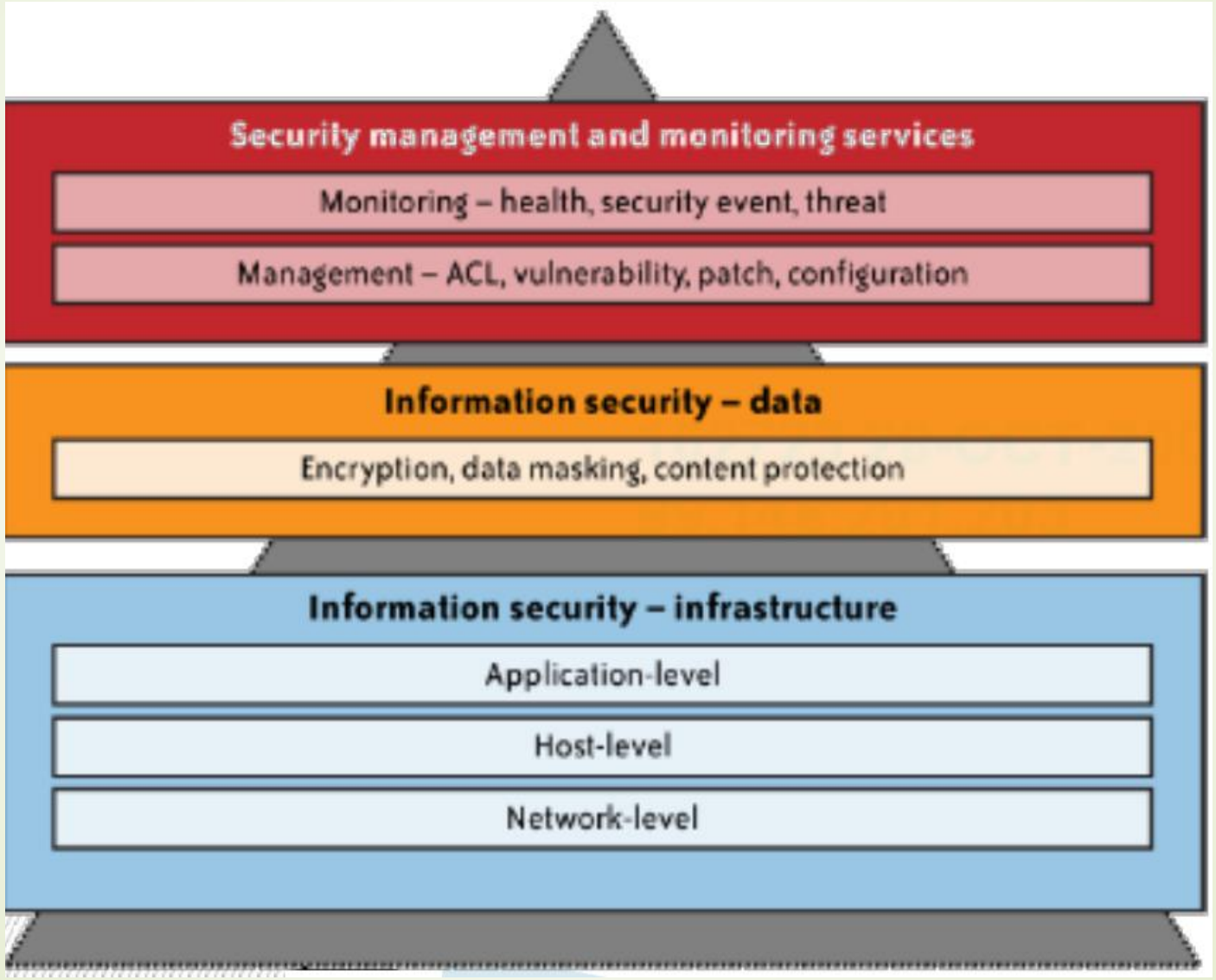
Cloud Security: Infrastructure, Data Security, and Access Control

Adapted from slide by Keke Chen

How Does Cloud Security Differ?

- What makes Cloud Security different from Normal Cyber Security Systems?

Overview


Security management and monitoring services
Monitoring – health, security event, threat
Management – ACL, vulnerability, patch, configuration
Information security – data
Encryption, data masking, content protection
Information security – infrastructure
Application-level
Host-level
Network-level

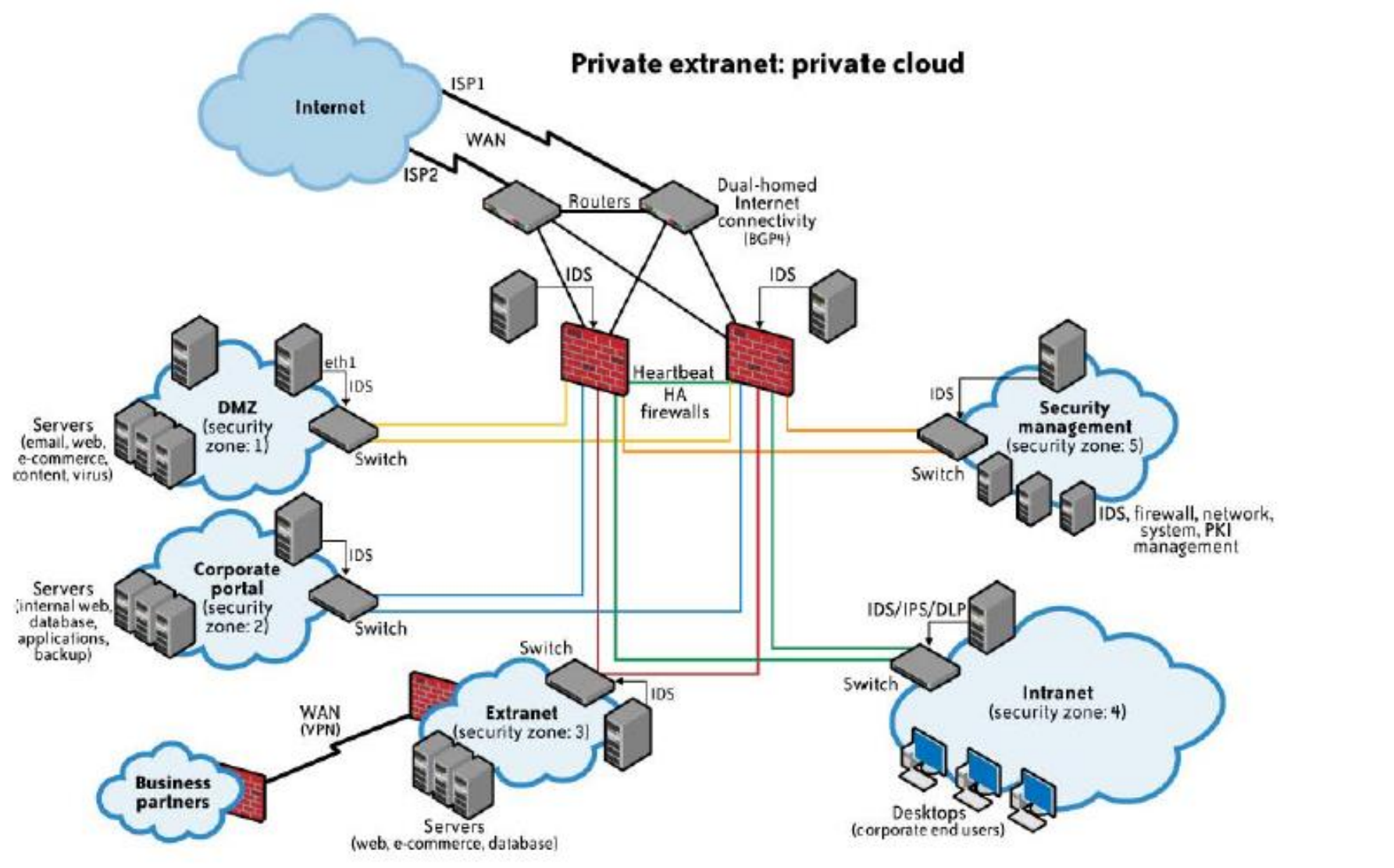
Infrastructure Security

- Infrastructure
  - IaaS, PaaS, and SaaS
- Focus on public clouds
  - No special security problems with private clouds – traditional security problems only
- Different levels
  - Network level
  - Host level
  - Application level

Network Level

- Confidentiality and integrity of data-in-transit
  - Amazon had security bugs with digital signature on SimpleDB, EC2, and SQS accesses (in 2008)
- Less or no system logging /monitoring
  - Only cloud provider has this capability
  - Thus, difficult to trace attacks
- Reassigned IP address
  - Expose services unexpectedly
  - Spammers using EC2 are difficult to identify
- Availability of cloud resources
  - Some factors, such as DNS, controlled by the cloud provider.

Private Cloud Network Security



Host Level (IaaS)

- Hypervisor security
  - "zero-day vulnerability" in VM, if the attacker controls hypervisor
- Virtual machine security
  - SSH private keys (if mode is not appropriately set)
  - VM images (especially private VMs)
  - Vulnerable Services

Application Level

- SaaS application security
  - Example: In an accident, Google Docs access control failed. All users can access all documents

What are the problem with Data Security in a Cloud?

- What are the issues?
- How does a Cloud make things worse?
- What techniques should be used?

Data Security

- Data-in-transit
  - Confidentiality and integrity
- Data-at-rest & processing data
  - Possibly encrypted for static storage
  - Cannot be encrypted for most PaaS and SaaS (such as Google Apps) & prevents indexing or searching
    - Research on indexing/searching encrypted data

Data Lineage

- Definition: tracking and managing data
- For audit or compliance purpose
- Data flow or data path visualization
  - E.g. data transferred to AWS on date x1 at time y1 and stored in a bucket on S3 example.s3.amazonaws.com, then processed on date x2 at time y2 on EC2 in ec2-67-202-51-223.compute-1.amazonaws.com, then stored in another bucket, example2.s3.amazonaws.com, then brought back locally on date x3 at time y3, …
- Time-consuming process even for inhouse data center
  - Not possible for a public cloud

Data Provenance

- Origin/ownership of data
  - Verify the authority of data
  - Trace the responsibility
  - e.g., financial and medical data
- Difficult to prove data provenance in a cloud computing scenario

Data cont.

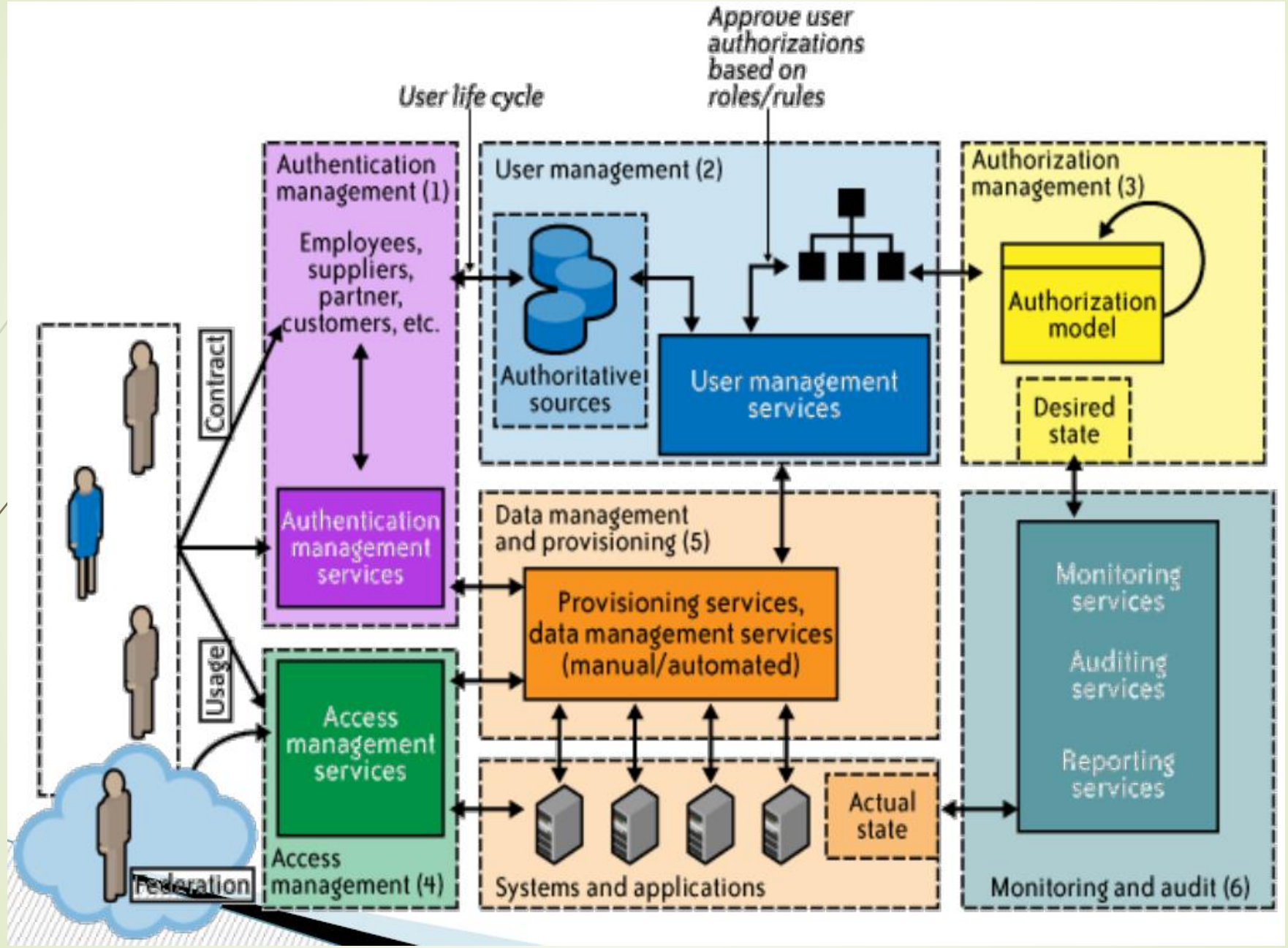
- Data remanence
  - Data left intact by a nominal delete operation
    - In many DBMSs and file systems, data is deleted by flagging it.
  - Lead to possible disclosure of sensitive information
- Provider's data and its security
  - The provider collects a huge amount of security related data
    - Data possibly related to service users
    - If not managed well, it is a big threat to users' security

Identity and Access Management

- Traditional trust boundary reinforced by network control
  - VPN, Intrusion detection, intrusion prevention
- Loss of network control in cloud computing
- Have to rely on higher-level software controls
  - Application security
  - User access controls - IAM

Identity and Access Management

- IAM components
  - Authentication
  - Authorization
  - Auditing
- IAM processes
  - User management
  - Authentication management
  - Authorisation management
  - Access management – access control
  - Propagation of identity to resources
  - Monitoring and auditing


Approve user authorizations based on roles/rules
User life cycle
Authentication management (1)
Employees, suppliers, partner, customers, etc.
Authentication management services
User management (2)
Authoritative sources
User management services
Authorization management (3)
Authorization model
Desired state
Contract
Usage
Federation
Data management and provisioning (5)
Provisioning services, data management services (manual/automated)
Access management services
Access management (4)
Systems and applications
Actual state
Monitoring services
Auditing services
Reporting services
Monitoring and audit (6)

IAM Standards and Specifications

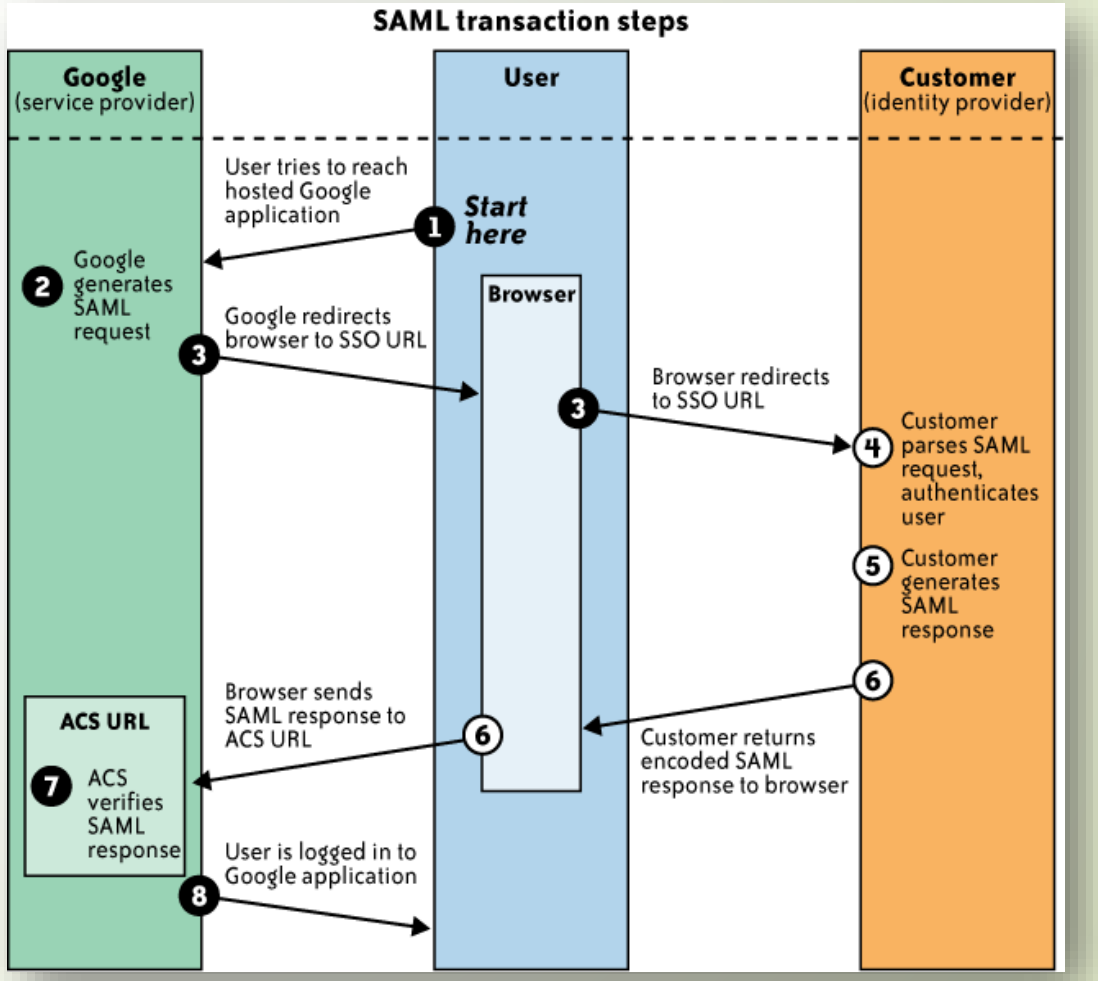
- Avoid duplication of identity, attributes, and credentials and provide a single sign-on user experience
- SAML(Security Assertion Markup Lang). http://shibboleth.internet2.edu/docs/internet2-mace-shibbolethарch-protocols-200509.pdf
- Automatically provision user accounts with cloud services and automate the process of provisioning and deprovisioning
  - SPML (service provisioning markup lang).

  http://www.oasis-open.org/standards#spmlv2.0
- Provision user accounts with appropriate privileges and manage entitlements
  - XACML (extensible access control markup lang).
- Authorise cloud service X to access my data in cloud service Y without disclosing credentials
  - Oauth (open authentication).

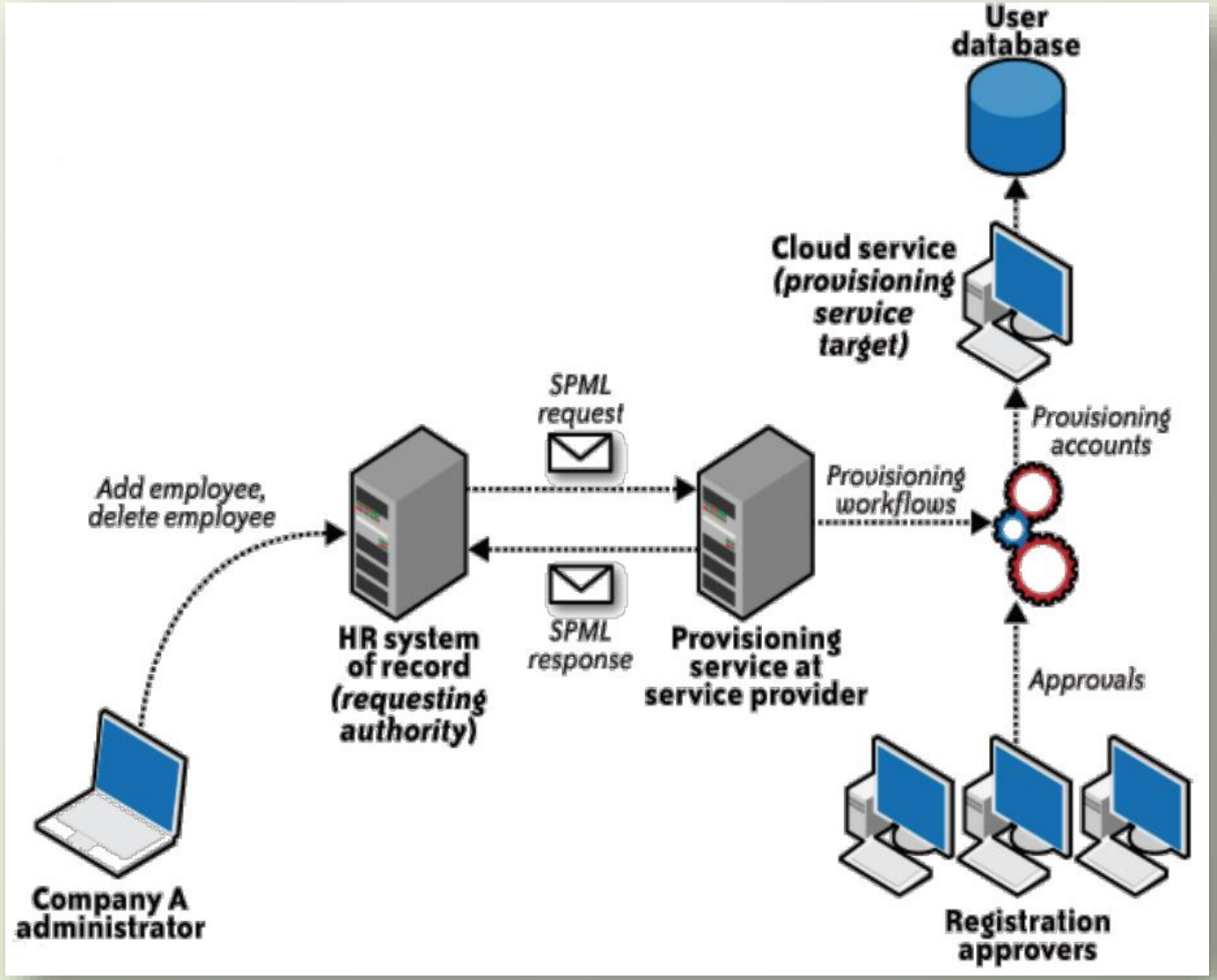
SAML Example

ACS: Assertion Consumer Service

SSO : single sign-on



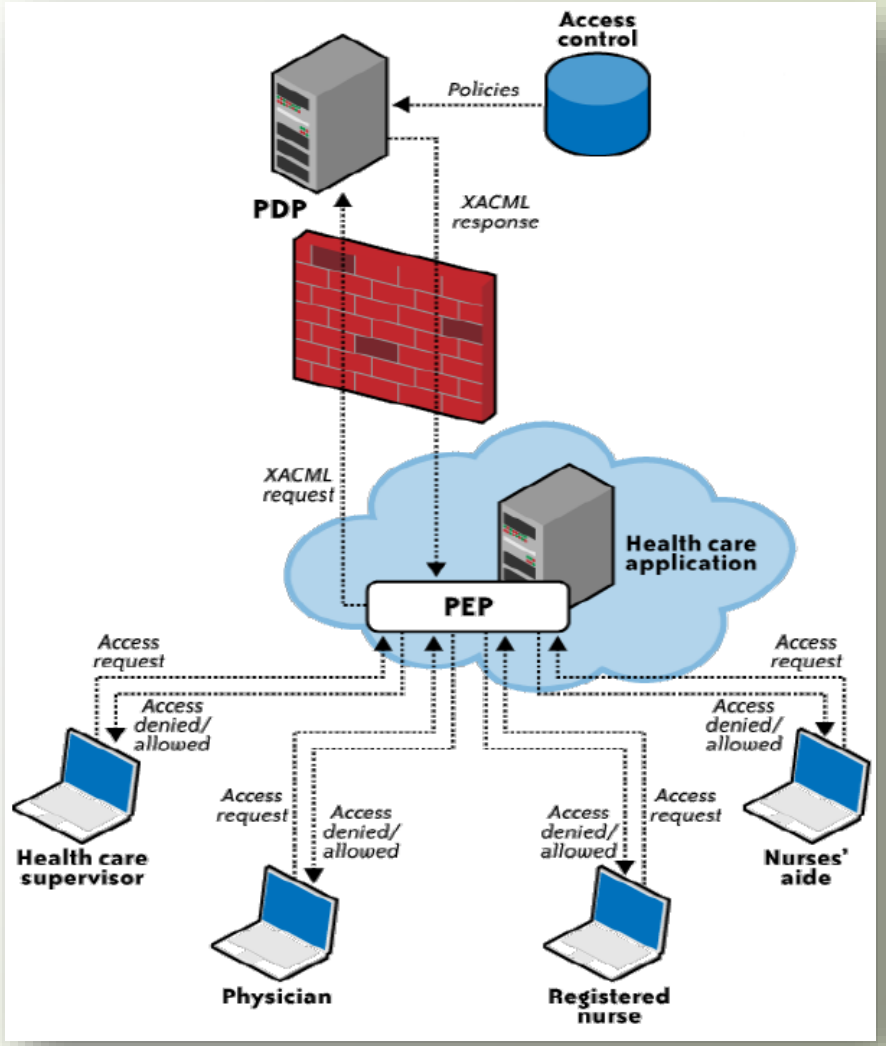
SPML Example



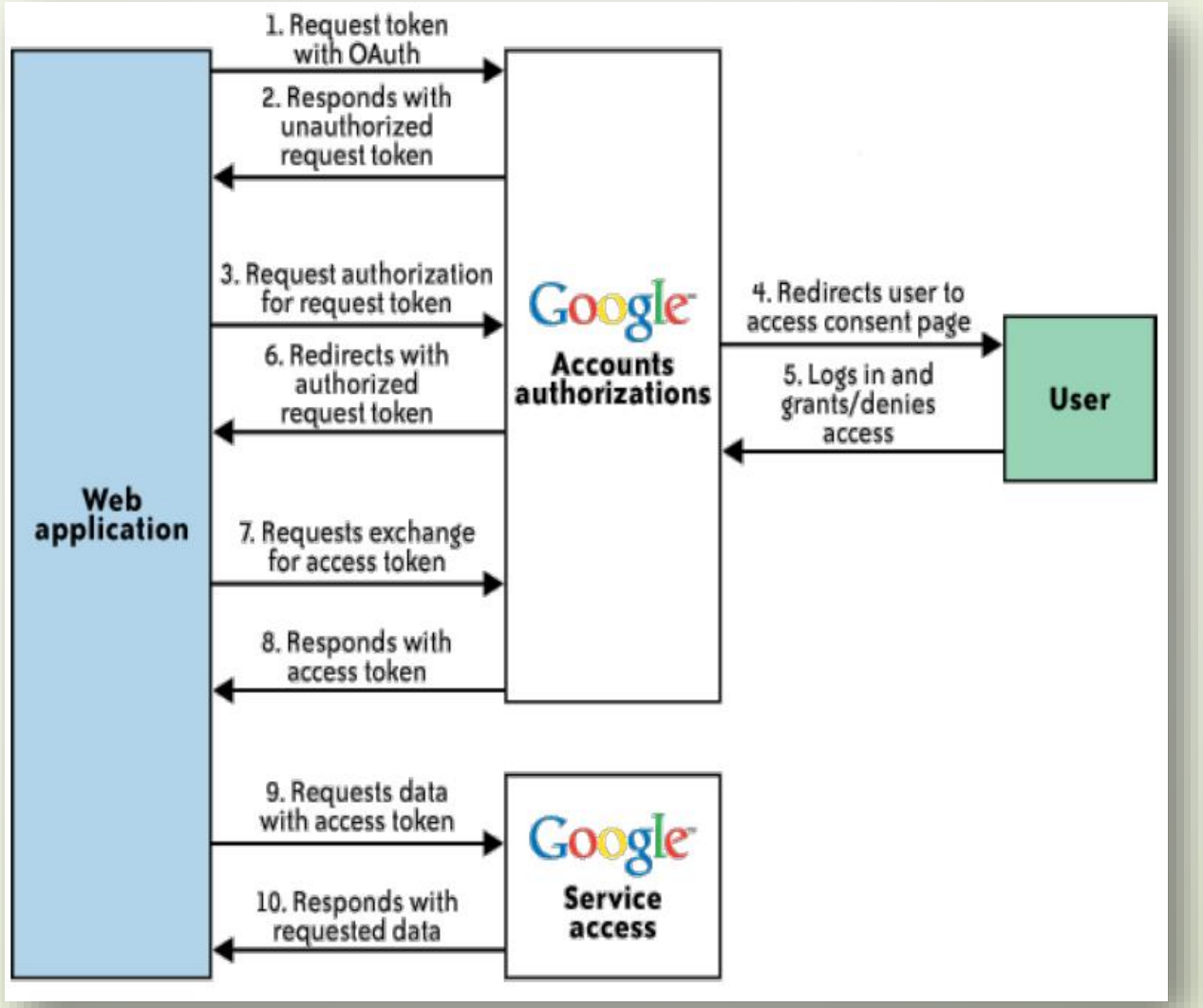
XACML Example

PEP: policy enforcement point (app interface)

PDP: policy decision point



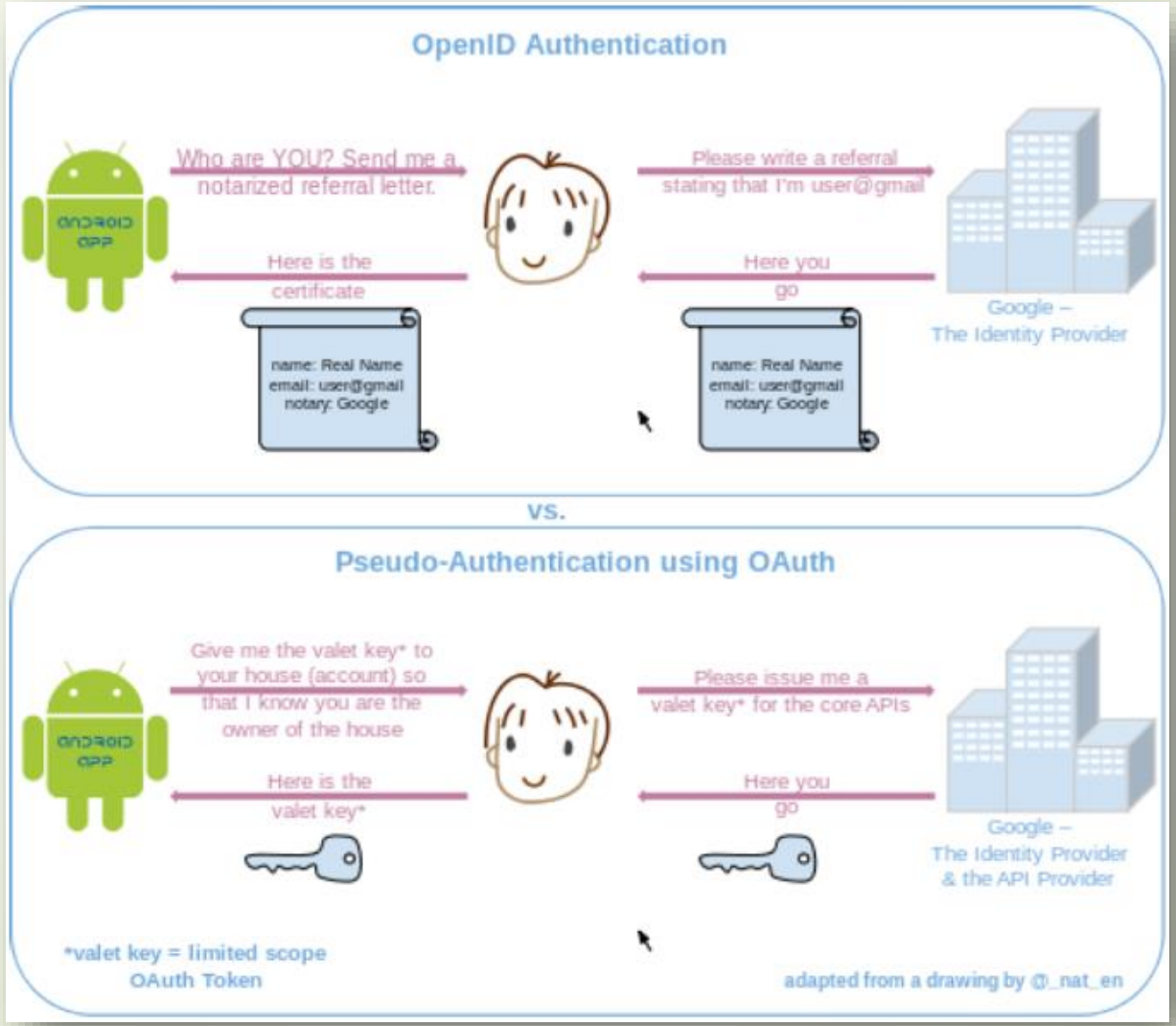
OAuth Example



IAM Standards/Protocols

- OpenID
- Information Cards
- Open Authentication (OATH)
- Issues for OpenID
  - Phishing – malicious relaying party forwards end user to bogus identity provider authentication page
  - Allows sniffing of certificate and replay

Difference Open ID versus Oauth (Thanks to Wikipedia)



Amazon Web Services

A Cloud Example

http://aws.amazon.com/

FAQs

Storage

- Storage
  - Simple Storage Service (S3)
  - Elastic Block Store (EBS)
  - AWS Import/Export
  - AWS Storage Gateway
- Compute
- Database
- Content Delivery
- Deployment & Management
- Messaging
- Network
- Web Traffic
- Workforce
- Payment and Billing

Simple Storage Service (S3)

- "Objects Storage for the Internet"
- Write, read, and delete unlimited number of objects, containing from 1 byte to 5 terabytes of data each
- Each object stored in a bucket and retrieved via a unique, developer assigned key
  - E.g: An object named photos/puppy.jpg and stored in the johnsmith bucket, is addressable using the URL http://johnsmith.s3.amazonaws.com/photos/puppy.jpg
- Public or Private objects, access control
- Simple Interfaces
  - REST
    - HTTP PUT, GET
  - SOAP
  - Bittorrent
- Pricing a combination of:
  - Per Storage GB, Per # of requests

Region: Asia Pacific (Singapore)

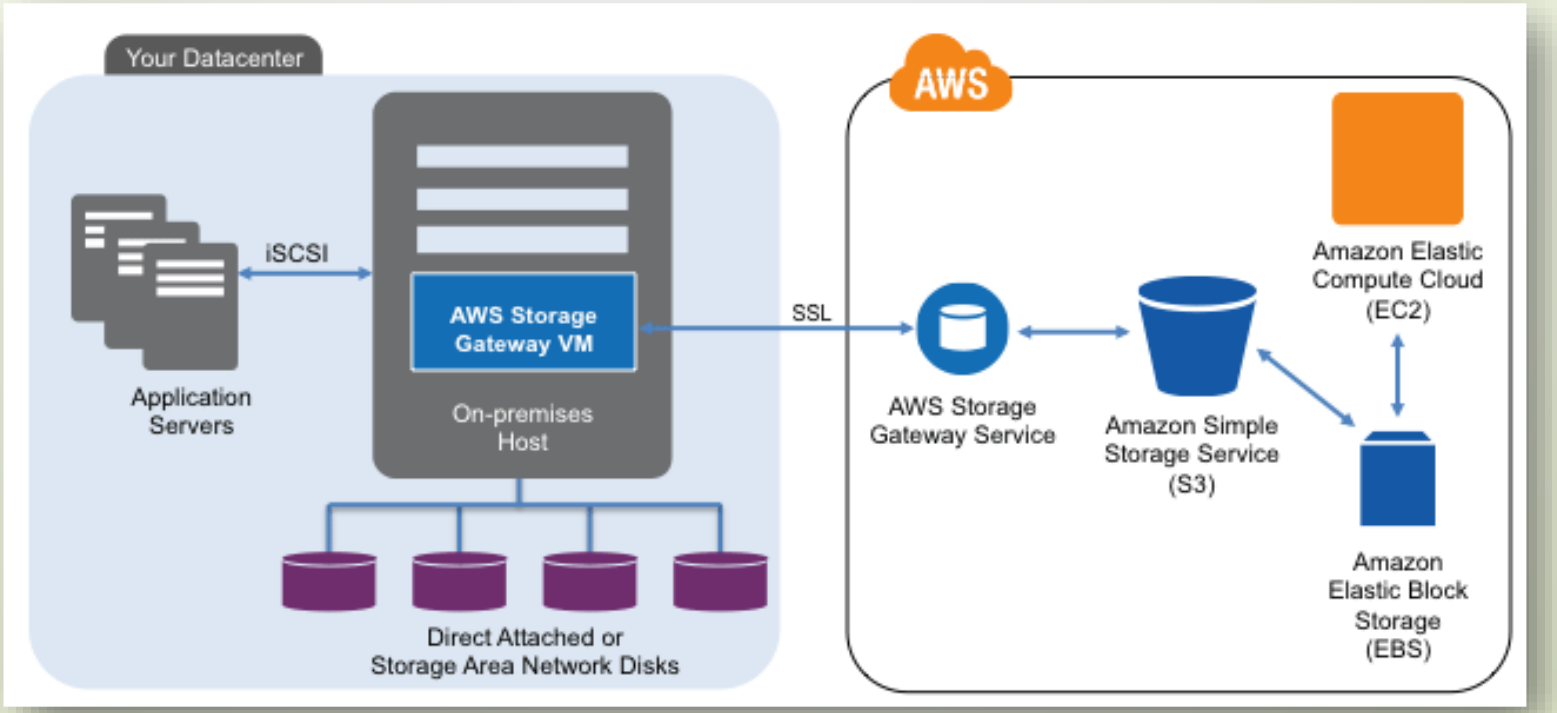
	Standard Storage	Reduced Redundancy Storage
First 1 TB / month	$0.125 per GB	$0.093 per GB
Next 49 TB / month	$0.110 per GB	$0.083 per GB
Next 450 TB / month	$0.095 per GB	$0.073 per GB
Next 500 TB / month	$0.090 per GB	$0.063 per GB
Next 4000 TB / month	$0.080 per GB	$0.053 per GB
Over 5000 TB / month	$0.055 per GB	$0.037 per GB

Elastic Block Store (EBS)

- "Cloud-based virtual hard drives"
- Block level storage volumes for use with Amazon EC2 instances
- Off-instance storage, persists independently from the life of an instance
- Can be attached to a running Amazon EC2 instance and exposed as a device within the instance
  - 1 GB to 1 TB
- Amazon CloudWatch
  - Monitors bandwidth, throughput, latency, …
- EBS can be (incrementally) backed up on S3
- Higher throughput than Amazon EC2 instance stores for applications performing a lot of random accesses
- Can attach multiple volumes to an instance and stripe across the volumes (RAID0) to achieve further increases in throughput.

AWS Storage Gateway

- Service for hybrid cloud storage
- Provides for "cloud-bursting"
- Designed for Enterprise storage and backup
- Use a gateway VM to connect to the cloud



Compute

- Storage
- Compute
  - Elastic Compute Cloud (EC2)
  - Elastic MapReduce
  - Auto Scaling
  - Elastic Load Balance
- Database
- Content Delivery
- Deployment & Management
- Messaging
- Network
- Web Traffic
- Workforce
- Payment and Billing

Elastic Compute Cloud: EC2

- Virtual machines running on Amazon's Datacenters
- Manage through CLI API or web-based tools
- On-demand (Pay-per-hour) or Reserved (Annual + discounted pay-per-hour)
- Instance Types
  - Micro, small, Medium, large, extra large, High-mem, …
- One EC2 Compute Unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor. This is also the equivalent to an early-20061.7 GHz Xeon processor

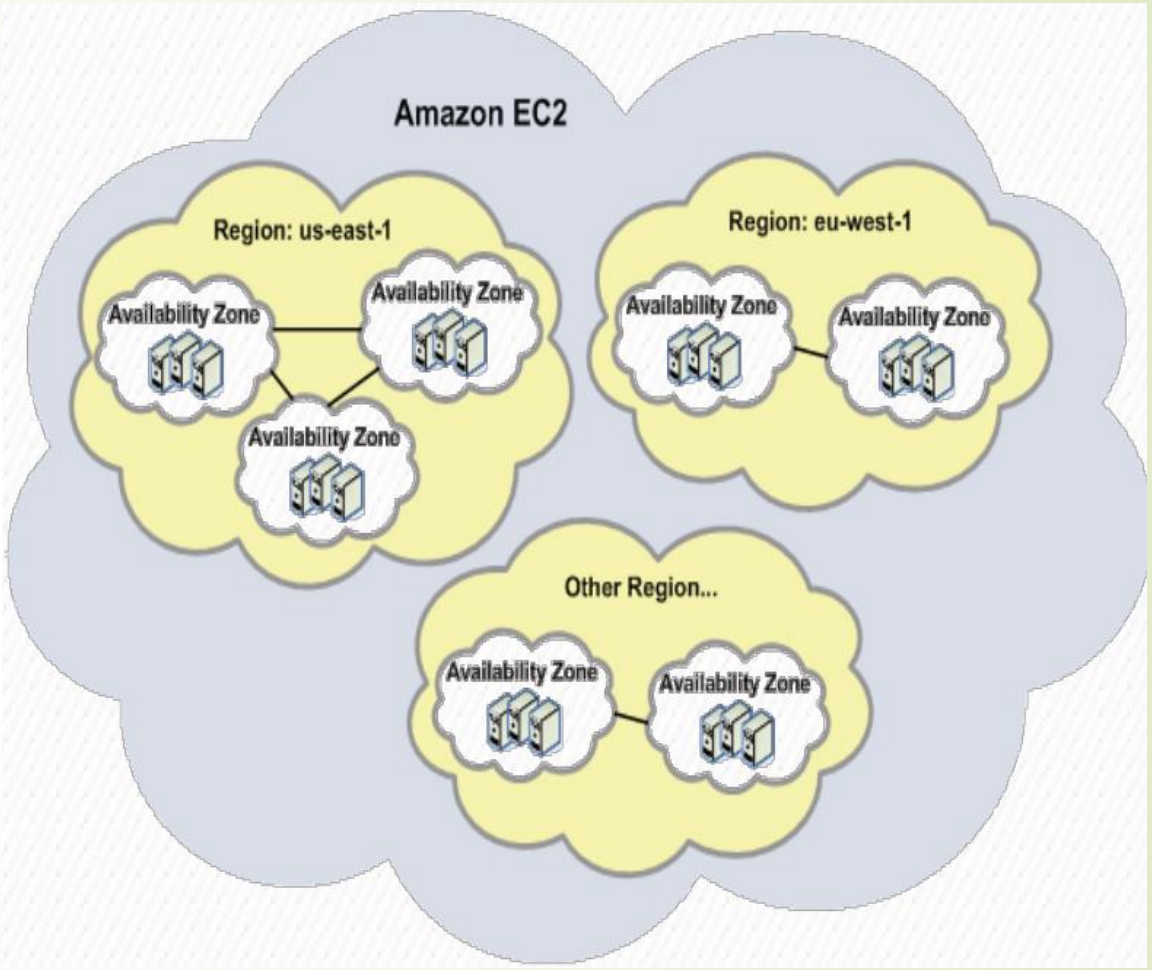
Region: Asia Pacific (Singapore)

	Linux/UNIX Usage	Windows Usage
Standard On-Demand Instances		
Small (Default)	$0.085 per Hour	$0.115 per Hour
Medium	$0.170 per Hour	$0.230 per Hour
Large	$0.340 per Hour	$0.460 per Hour
Extra Large	$0.680 per Hour	$0.920 per Hour
Micro On-Demand Instances		
Micro	$0.020 per Hour	$0.020 per Hour
High-Memory On-Demand Instances		
Extra Large	$0.506 per Hour	$0.570 per Hour
Double Extra Large	$1.012 per Hour	$1.140 per Hour
Quadruple Extra Large	$2.024 per Hour	$2.280 per Hour
High-CPU On-Demand Instances		
Medium	$0.186 per Hour	$0.285 per Hour
Extra Large	$0.744 per Hour	$1.140 per Hour
Cluster Compute Instances		
Quadruple Extra Large	N/A*	N/A*
Cluster GPU Instances		
Quadruple Extra Large	N/A*	N/A*

* Cluster and High I/O Instances are not available in all regions

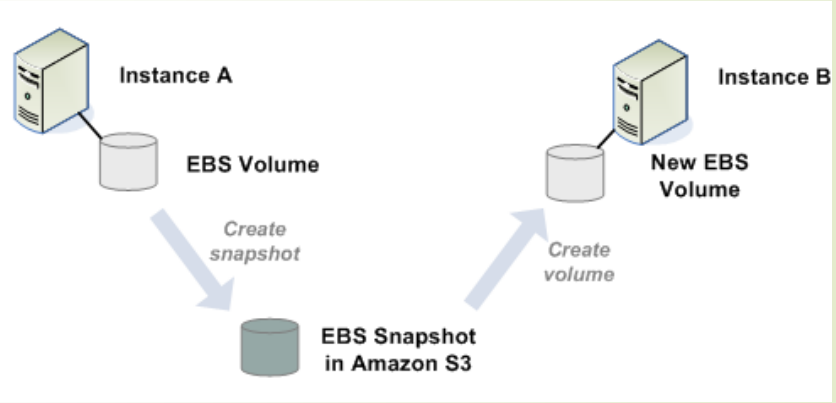
Elastic Compute Cloud: EC2

- Regions
  - Amazon has data centers in different areas of the world (e.g., North America, Europe, Asia, etc.)
- Closer to specific customers or to meet legal or other requirements
- Each Region contains multiple distinct locations called Availability Zones
- Availability Zones are isolated from failures in others
- Inexpensive, low-latency network connectivity to other zones in the same Region
- Launching instances in separate Availability Zones protect applications from failure in a single location



Elastic Compute Cloud: EC2

- Amazon Machine Images
  - Basically a Xen VM image: operating system, application server, and applications
  - Launch instances: run copies of the AMI
  - Runs until you stop or terminate them or if it fails
- Storage
  - Store the AMI images in S3
  - EBS: essentially hard disks that you can attach to a running instance



Auto Scaling & Elastic Load Balance

- Auto Scaling
  - Monitor the load on EC2 instances using CloudWatch
  - Define Conditions
  - Spawn new instances when there is too much load or remove instances when not enough load
- Elastic Load Balance
  - Automatically distributes incoming application traffic across multiple EC2 instances
  - Detects EC2 instance health and divert traffic from bad ones
  - Support different protocols
    - HTTP, HTTPS, TCP, SSL, or Custom
- They can work together

Database

- Storage
- Compute
- Database
  - Relational Database Service (RDS)
  - SimpleDB
  - DynamoDB
  - ElastiCache
- Content Delivery
- Deployment & Management
- Messaging
- Network
- Web Traffic
- Workforce
- Payment and Billing

Relational Database Service (RDS)

- Preconfigured EC2 instances with MySQL or Oracle installed
  - 1. Create an RDS instance
  - 2. Dump your database into it
    - mysqldump acme | mysql --host=hostname --user=username --password acme
  - 3. Update SQL connection strings in your application (which might be running anywhere, including EC2 VMs)
- Features:
  - Pre-configured
  - Monitoring and Metrics (CloudWatch)
  - Automatic Software Patching
  - Automated Backups
  - DB Snapshots
  - Changing the instance type (= increase computer power)
    - Through EBS snapshots
  - Multi-AZ Deployments
  - Read Replicas
    - Scaling for read-heavy database workloads
  - Isolation and Security

SimpleDB

- A NoSQL database, non-relational
- Eventual consistency, no ACID compliance
- Data model is comprised of domains, items, attributes and values
  - Large collections of items organised into domains
  - Items are little hash tables containing attributes of key, value pairs
- Use Put, Batch Put, & Delete to create and manage the data set
- Use GetAttributes to retrieve a specific item
- Attributes can be searched with various lexicographical queries
- The service manages infrastructure provisioning, hardware and software maintenance, replication, indexing of data items, and performance tuning
- Tables limited to 10 GB, typically under 25 writes/second
- User manages partitioning and re-partitioning of data over additional SimpleDB tables

SimpleDB	S3
Indexes all the attributes	Stores raw data
Uses less dense drives	Users dense storage drives
Better optimised for random access	Optimised for storing large objects

DynamoDB

- Amazon Dynamo paper (2007) -> Open-source Apache Cassandra project -> DynamoDB (1/2012) *
  - Dynamo is a highly available, key-value structured storage system
- Fully managed NoSQL non-relational Database
- Data model is comprised of domains, items, attributes and values (similar to SimpleDB)
  - Domains are collections of items that are described by attribute-value pairs
- Pay by throughput, not storage
- Run on solid state disks (SSDs)
- There are no limits on the request capacity or storage size for a given table.
  - DynamoDB automatically partitions data and workload over a sufficient number of servers to meet the scale requirements

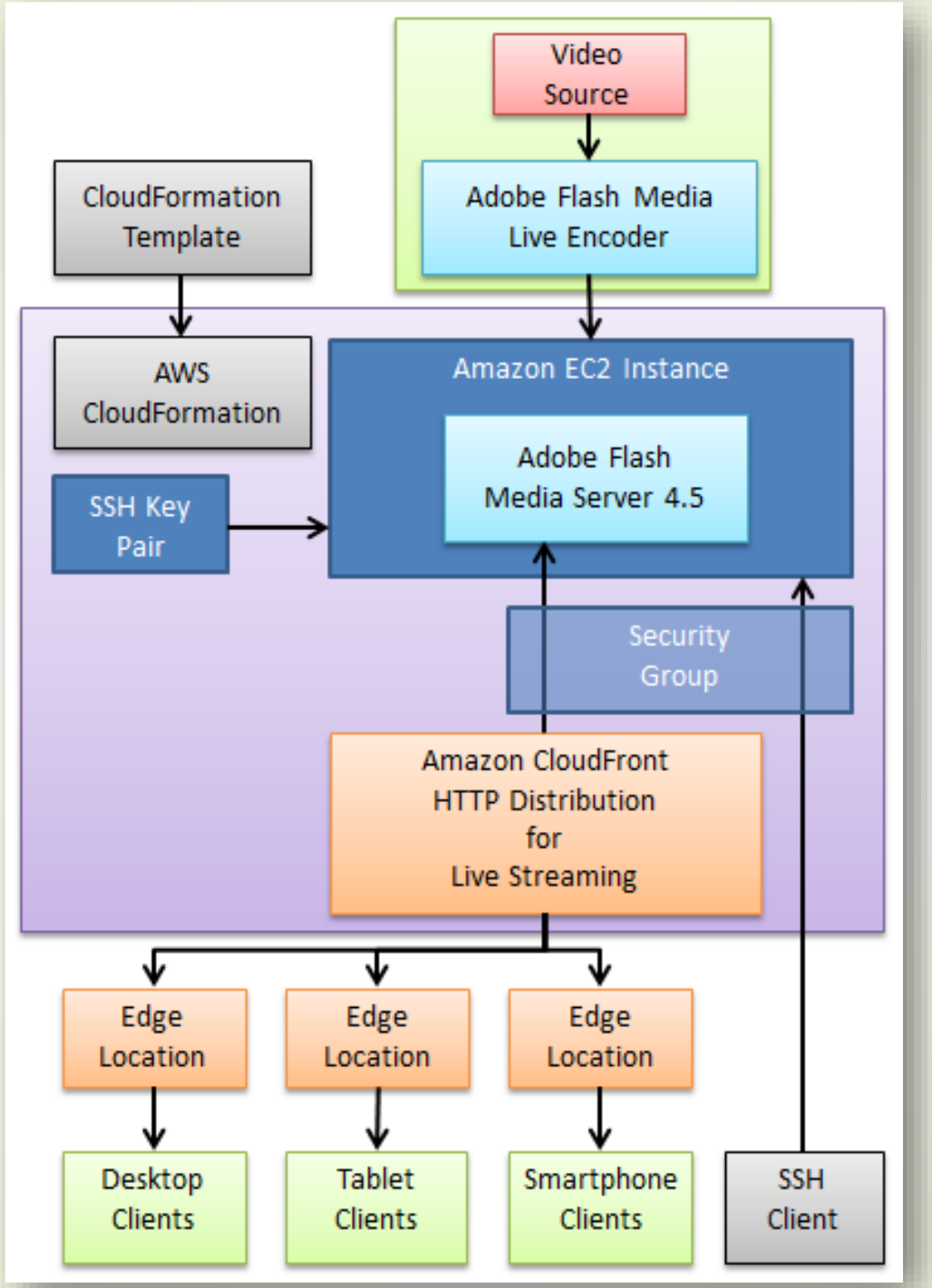
*http://www.datastax.com/dev/blog/amazon-dynamodb

Content Delivery

- Storage
- Compute
- Database
- Content Delivery
  - CloudFront
- Deployment & Management
- Messaging
- Network
- Web Traffic
- Workforce
- Payment and Billing

CloudFront – Content Delivery

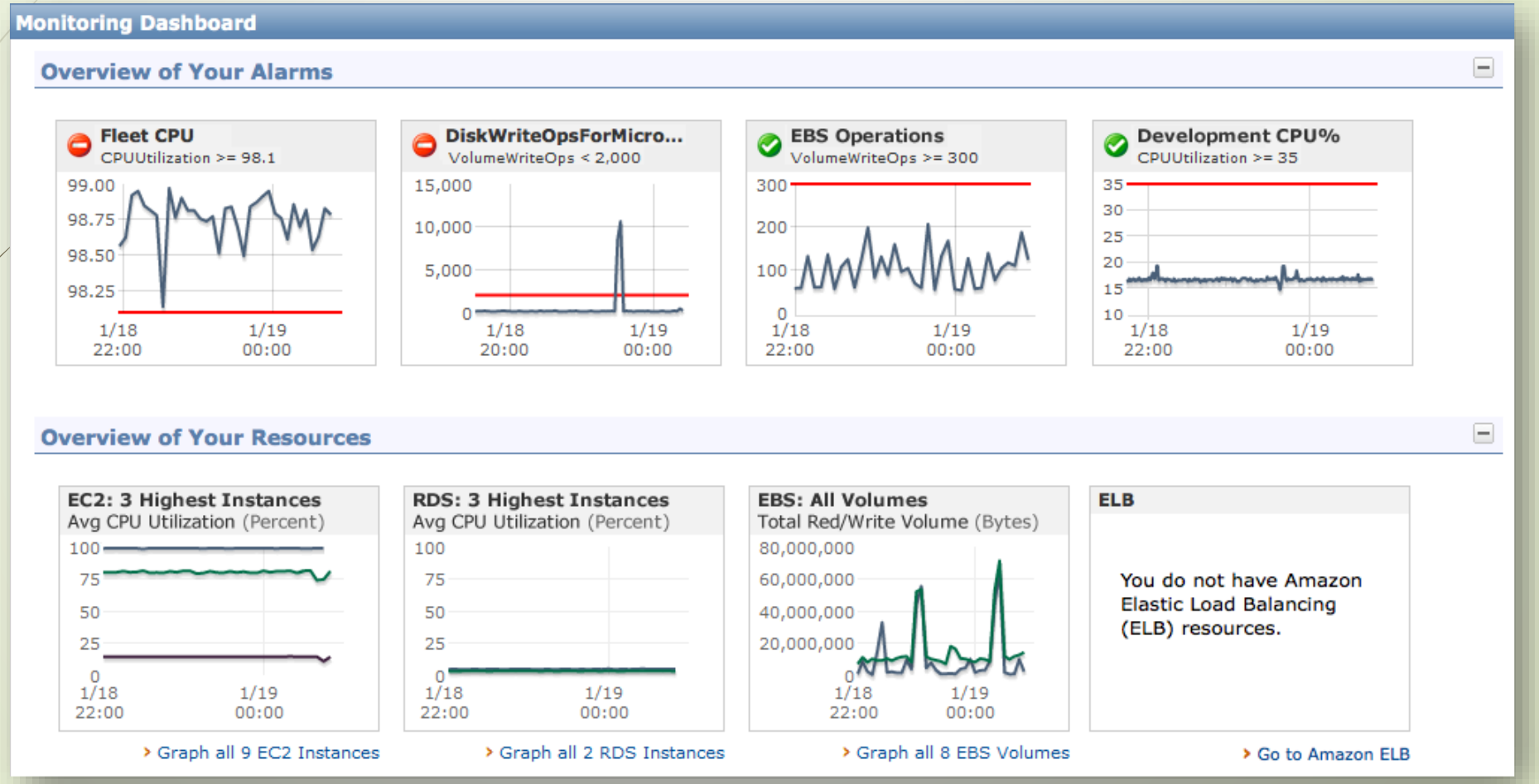
- Delivers static and streaming content using a global network of edge locations
- Store the original versions of your files on an origin server.
  - Amazon S3 bucket, Amazon EC2 instance, or your own server
- Register the origin server with CloudFront through a simple API call
- When users request an object using the original domain name, they are automatically routed to the nearest edge location


Video Source
Adobe Flash Media Live Encoder
CloudFormation Template
AWS CloudFormation
Amazon EC2 Instance
Adobe Flash Media Server 4.5
SSH Key Pair
Security Group
Amazon CloudFront HTTP Distribution for Live Streaming
Edge Location
Edge Location
Edge Location
Desktop Clients
Tablet Clients
Smartphone Clients
SSH Client

Amazon CloudWatch

- Monitor AWS resources automatically
  - Monitoring for Amazon EC2 instances: seven pre-selected metrics at five-minute frequency
  - Amazon EBS volumes: eight pre-selected metrics at five-minute frequency
  - Elastic Load Balancers: four pre-selected metrics at one-minute frequency
  - Amazon RDS DB instances: thirteen pre-selected metrics at one minute frequency
- Custom Metrics generation and monitoring
- Set alarms on any of the metrics to receive notifications or take other automated actions
- Use Auto Scaling to add or remove EC2 instances dynamically based on CloudWatch metrics

CloudWatch


Monitoring Dashboard
Overview of Your Alarms
Fleet CPU
CPUUtilization >= 98.1
99.00
98.75
98.50
98.25
1/18 22:00
1/19 00:00
DiskWriteOpsForMicro...
VolumeWriteOps < 2,000
15,000
10,000
5,000
0
1/18 20:00
1/19 00:00
EBS Operations
VolumeWriteOps >= 300
300
200
100
0
1/18 22:00
1/19 00:00
Development CPU%
CPUUtilization >= 35
35
30
25
20
15
10
1/18 22:00
1/19 00:00
Overview of Your Resources
EC2: 3 Highest Instances
Avg CPU Utilization (Percent)
100
75
50
25
0
1/18 22:00
1/19 00:00
RDS: 3 Highest Instances
Avg CPU Utilization (Percent)
100
75
50
25
0
1/18 22:00
1/19 00:00
EBS: All Volumes
Total Red/Write Volume (Bytes)
80,000,000
60,000,000
40,000,000
20,000,000
0
1/18 22:00
1/19 00:00
ELB
You do not have Amazon Elastic Load Balancing (ELB) resources.
Graph all 9 EC2 Instances
Graph all 2 RDS Instances
Graph all 8 EBS Volumes
Go to Amazon ELB

Elastic Beanstalk

- Solution for Enterprise server-side java application deployment
- Create your application (e.g. Eclipse).
- Package deployable code into a standard Java Web Application Archive (WAR file).
- Upload the WAR file to Elastic Beanstalk using the AWS Management Console, …
- Deploy the application
  - Elastic Beanstalk handles the provisioning of a load balancer and the deployment of the WAR file to one or more EC2 instances running the Apache Tomcat application server
- Access the application at a customized URL (e.g. http://myapp.elasticbeanstalk.com/).

Messaging

- Storage
- Compute
- Database
- Content Delivery
- Deployment & Management
- Messaging
  - Simple Queue Service (SQS)
  - Simple Notification Service (SNS)
  - Simple Email Service (SES)
- Network
- Web Traffic
- Workforce
- Payment and Billing

OpenStack Cloud Computing

General Introduction

Open-Source Software Solution

- We have a mix of different APIs—most proprietary— making it difficult or infeasible to deploy and to evaluate security.
- What if we had a standard API that was open and freely available?
- What is “the Linux of Cloud Computing Platforms?”

Cloud Computing: OpenStack

- *"The OpenStack project has been created with the audacious goal of being the ubiquitous software choice for building cloud infrastructures."*

  —Ken Pepple, Deploying OpenStack, O'Reilly

- *"Cloud computing is a computing model, where resources such as computing power, storage, network and software are abstracted and provided as services on the Internet in a remotely accessible fashion. Billing models for these services are generally similar to the ones adopted for public utilities. On-demand availability, ease of provisioning, dynamic and virtually infinite scalability are some of the key attributes of cloud computing."*

  — docs.openstack.org

- *"OpenStack is a collection of open source software projects that enterprises/service providers can use to setup and run their cloud compute and storage infrastructure."*

  — docs.openstack.org

- The OpenStack Consortium has grown rapidly in the past year:
  - NASA
  - Rackspace
  - Citrix
  - Dell
  - AMD
  - Intel
  - Cisco
  - HP
  - Over 140 Others

- OpenStack services can be made available via Amazon's S3 and EC2 APIs. Applications written for Amazon Web Services can work with OpenStack.

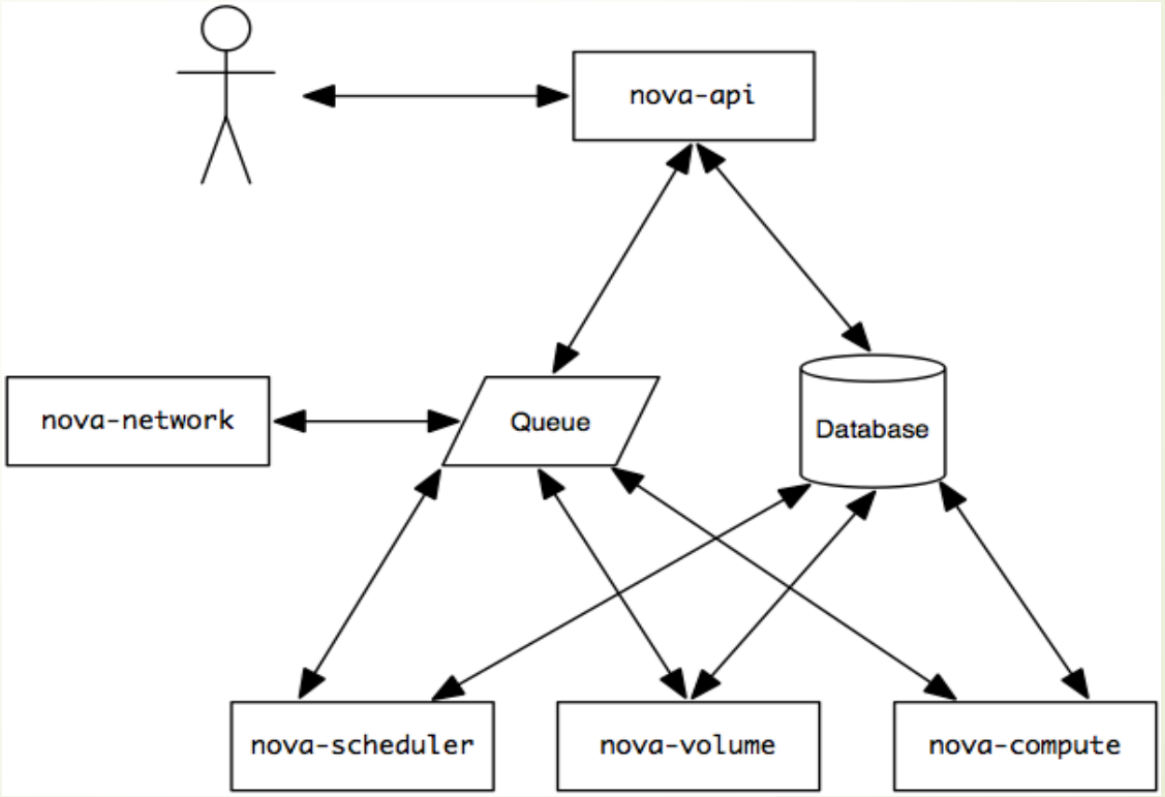
OpenStack's Core Components

- Compute ("Nova")
  - Orchestrates large networks of Virtual Machines.
  - Responsible for VM instance lifecycle, network management, and user access control.
- Object Storage ("Swift")
  - Provides scalable, redundant, long-term storage for things like VM images, data archives, and multimedia.
- Image Service ("Glance")
  - Manages VM disk images.
  - Can be a stand-alone service.
  - Supports private/public permissions and can handle a variety of disk image formats.

OpenStack Nova

- Nova was contributed by NASA from the Nebula platform.
- Nova allows users to create, destroy, and manage virtual machines using user-supplied images.
- Corresponds to Amazon's EC2.
- Users can use OpenStack API or Amazon's EC2 API.
- Uses Python and Web Server Gateway Interface (WSGI).

OpenStack Nova: Architecture



OpenStack Nova: nova-api

- A daemon that is the workhorse of Nova.
  - Handles API requests.
  - Manages most orchestration.
  - Enforces some policies.
- If it can, it will handle the request on its own with help from the database.
- Otherwise, it will delegate to the other nova daemons using the message queue as well as the database.

OpenStack Nova: nova-compute

- Worker that does the actual work of starting and stopping virtual machine instances.
- Takes its orders from the message queue and executes the appropriate VM API calls to accomplish the task.
- Commonly uses “libvirt” (RedHat), but can use Xen, vSphere (VMware), or Windows Management Interface.