**CC Assignment – 2**

**Q.1) What is the walrus storage controller? Explain the working of this particular module in detail**.

Ans.)

* Walrus is also called “WS3” and is the storage service provided by Eucalyptus. The Storage Service provides simple storage functionality, which is exposed by ReSTful and Soap APIs.
* Walrus takes care of storing the virtual machine images, storing the snapshots and serving Files. As with all other public facing Services in Eucalyptus, these Services are based on the Amazon Web Services API.
* Containers in Walrus Storage are called „Buckets“ and they have to be unique across accounts, just like it is with Amazon Web Services (AWS). Some naming restrictions are:

->Containers can contain lowercase letters, numbers, periods (.), underscores (\_), and dashes (-)

->Container Names must start with a number or letter

->The Length of a Name must be between 3 and 255 characters long

->It is not allowed to use an IP-Address as Name (e.g., 265.255.5.4)

* The maximum File Size in a Walrus Container is 5 Terabytes and Files can either be public or private. If the Container should be deleted, a container must be empty, which means that all files have to be deleted prior to deleting the container. Files are identified via unique Keys represented by Uniform Resource Identifiers (URIs).
* Common Actions performed on the Walrus storage are the creation of containers, store data in containers, download data and grant or deny permissions. These Actions can be performed via the ReSTful or SOAP Interfaces.
* The Walrus Storage distinguishes two major read options: consistent read or eventually consistent read. The later one is faster but might server inconsistent data whereas the first one might have higher latency but data is always consistent.
* The Storage Controller takes care of the creation of persistent EBS devices. Block Storage Devices are typically provided over over the ATAoverEthernet or iSCSI protocol to the instances.
* Its main function is:

To store the machine images that can be used by the cloud controller

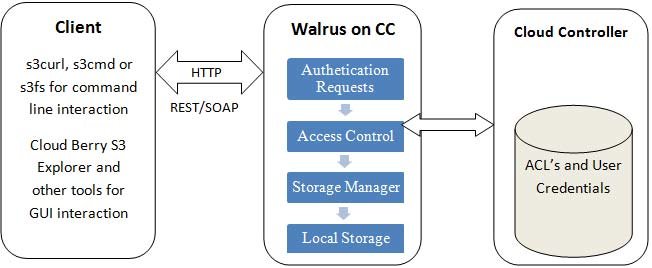
To access and store the data from anywhere.

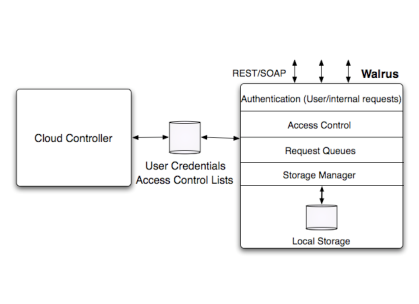
It provides file level storage system and doesn’t provide the locking of a file. It doesn’t allow concurrent file to change the status of a single file that is getting modified by the user.

It saves the state of the images.

* Walrus uses a database to manage metadata for buckets and objects as well as ACLs and policies. Walrus has its own database but is currently co-hosted by the Cloud Controller (CLC) with all of the other databases that Eucalyptus uses. The current implementation leverages a PostGreSQL database although version prior to Eucalyptus 3.1 ran on MySQL.

Walrus, like all Eucalyptus components written in Java, utilizes Hibernate to interact with the db layer and database HA is handled via a replicating Hibernate JDBC connection.





**Q.2) List and explain the different services of CSB (Cloud Services Brokerage)**

Ans.) Cloud services brokerage (CSB) is an IT role and business model in which a company or other entity adds value to one or more (public or private) cloud services on behalf of one or more consumers of that service via three primary roles including aggregation, integration and customization brokerage. A CSB enabler provides technology to implement CSB, and a CSB provider offers combined technology, people and methodologies to implement and manage CSB-related projects.

There are three primary areas a cloud service broker can address in accelerating the adoption of the cloud:

1. Aggregation – enabling the consumption of cloud by end users via a cloud application marketplace approved by the company
2. Integration – ensuring cloud applications exchange data with each other and with on-premise applications to orchestrate business processes
3. Customization – augmenting cloud services with changes to data schema or enhanced security and compliance

The challenge for IT is that the cloud is relatively immature compared to on-premise enterprise software. By adding customized capabilities on top of cloud services, the enterprise can realize the benefits of cloud, while also meeting its other business objectives including data security and compliance. In particular, organizations are looking to augment the cloud and achieve the following:

1. Reduce risk with more robust security and compliance capabilities
2. Add value and visibility with analytics
3. Centralize functionality for audit trails and policy enforcement
4. Streamline the selection process of cloud services

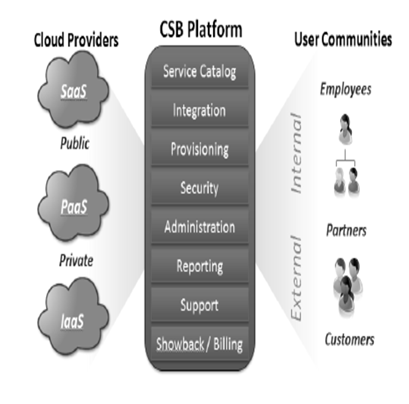
Advantages of CSB:

1) Broader Technical Expertise

2) Lower Total Cost of Ownership – Financial Returns

3) Operational efficiencies

4) Better options in dealing with risk, compliance and governance



**Cloud Service Aggregation** - Aggregation in Cloud Service Brokerage creates a virtual service provider, offering normalized data models across cloud services and enabling data portability and security across a multitude of services. Cloud Service Aggregators should empower flexibility and portability between providers.

**Cloud Service Arbitrage** - Arbitrage in Cloud Service Brokerage is a complementary function of Aggregation, wherein flexibility to move between cloud services and a single point of access enable end-users to select the best choice, based on metrics. Offering choice in services is only beneficial with portability to take advantage of performance and cost savings.

**Cloud Service Integration** - Integration Cloud Service Brokerage is a function which maintains the data fidelity for organizations using multiple on-demand B2B software services, SaaS, PaaS, or IaaS and the resulting silos they create. Cloud Service Integration can be complex and require effort from not only the brokerage, but B2B vendors and infrastructure providers alike.

**Cloud Service Intermediation** - Intermediation Cloud Service Brokerage provides specific value added services to enhance the capabilities of an existing cloud services. Examples might include identity or access management to multi-cloud services. Ask if your cloud broker is able to intermediate based on your requirements, or if there is a limitation on new services which can be added.

**Q.3) Explain the functioning of Cloud Storage Gateways with the help of a diagram.**

Ans.)

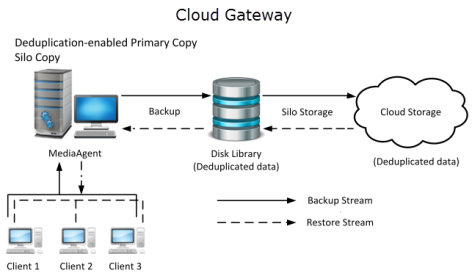
* A cloud storage gateway is a hardware- or software-based appliance located on the customer premises that serves as a bridge between local applications and remote cloud-based storage.
* A cloud storage gateway provides basic protocol translation and simple connectivity to allow the incompatible technologies to communicate transparently. The gateway can make cloud storage appear to be a NAS filer, a block storage array, a backup target or even an extension of the application itself.
* The gateway may be a stand-alone computing device or a virtual machine (VM) image that provides basic protocol translation and connectivity that allows incompatible technologies to communicate transparently.
* The need for a bridge between cloud storage systems and enterprise applications arose because of an incompatibility between the protocols used for public cloud technologies and legacy storage systems. Most public cloud providers rely on Internet protocols, usually a RESTful API over HTTP, rather than conventional storage area network (SAN) or network-attached storage (NAS) protocols.
* Many of today's cloud storage gateway products provide data deduplication and compression capabilities to make use of available bandwidth efficiently and move data as quickly as possible. Reducing the digital footprint also lowers cost, because cloud providers charge for over-the-wire transfers as well as for storage space. Other popular features include snapshots and version control, the ability to use local storage as a cache, automated tiered storage and encryption. As the market has evolved, some vendors have dropped the word "gateway" in favour of the word "controller" to emphasize the idea that their gateway products do more than just serve as a bridge.

**Functionality:**

In enterprise infrastructures NFS is mainly used by Linux systems whereas Windows systems are using SMB. Object storage needs data in form of objects rather than files. For all cloud storage gateways it is mandatory to cache the incoming files and destage them to object storage on a later step. The time of destaging is subject to the gateway and a policy engine allows functions like :

* pinning = bind specific files to the cache and destage them only for mirroring purpose
* content based destaging = move only files with specific characteristics to object storage e.g. all MP3 files
* multi-cloud mirroring = mirror all files to two different object stores
* Least Recently use = fill the local cache to maximum, move all files to object storage and delete files in cache on a LRU algorithm
* encrypt prior of destage = files are encrypted on the cloud storage gateway and destaged to object storage in an encrypted form
* compress and / or dedupliation prior of destage = files are deduplicated and/or compressed prior of destaging
* backup data in a native backup format

Combinations of these functions are usual. Default sorting schematics spanning the retrieval interface generally rely on zero-fault content processing, which carries the obvious requirement that two or more of the above functions are synchronized.



**Q.4) Describe in detail the working of the AAA model in cloud computing.**

Ans.)

**AAA stands for Authentication, Authorization, and Accounting.**

Authentication Authentication is the process of identifying an individual, usually based on a username and password. Authentication is based on the idea that each individual user will have unique information that sets him or her apart from other users.

• Refers to confirmation that a user who is requesting a service is a valid user. • Accomplished via the presentation of an identity and credentials. • Examples of credentials include passwords, one-time tokens, digital certificates, and phone numbers (calling/called).

**Authorization**

Authorization is the process of granting or denying a user access to network resources once the user has been authenticated through the username and password. The amount of information and the amount of services the user has access to depend on the user's authorization level.

• Refers to the granting of specific types of service (including "no service") to the users based on their authentication. • May be based on restrictions, for example, time-of-day restrictions, or physical location restrictions, or restrictions against multiple logins by the same user. • Examples of services include, IP address filtering, address assignment, route assignment, encryption, QoS/differential services, bandwidth control/traffic management, etc.

**Accounting**

**Accounting** is the process of keeping track of a user's activity while accessing the network resources, including the amount of time spent in the network, the services accessed while there and the amount of data transferred during the session. Accounting data is used for trend analysis, capacity planning, billing, auditing and cost allocation. • Refers to the tracking of the consumption of network resources by users. • Typical information that is gathered in accounting include the identity of the user, the nature of the service delivered, when the service began, and when it ended. • May be used for management, planning, billing, etc.

RADIUS is an example of an AAA service.

**Accounting Techniques**

Accounting is an increasingly critical step in the overall AAA process. Regulatory controls are starting to mandate better auditing of network access. The last stage of AAA, accounting simply records which clients accessed the network, what they were granted access to, and when they disconnected from the network.

Accounting has always been widely used in the Internet Service Provide (ISP) space because auditing network access is the basis for billing ISP customers. Increasingly, accounting is being used as a way to correlate client attribute

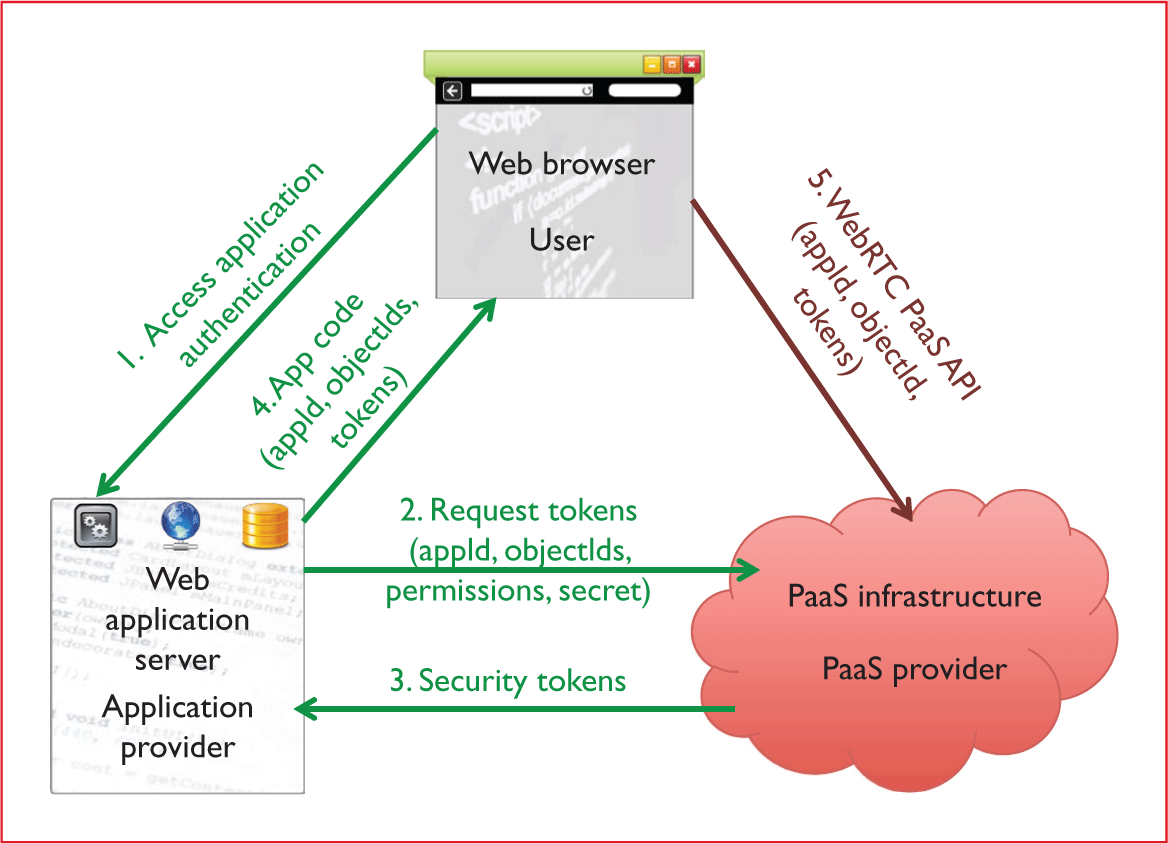
information (username, IP address, etc.) with actions and events on the network.

This correlation can make other systems that are not user-aware more intelligent in the security decisions that they make. For example, a network Intrusion Detection System (IDS) can learn a lot about the behavior of a given IP address.

However, when that information is correlated with the user assigned to that IP address—and the permissions that user should have—the relevance of the IDS data increases dramatically.

One of the design considerations of accounting systems is that, given the centralized nature of audit and the decentralized nature of access, they are generally out-of-band with the client's normal communications.

This makes them excellent resources to refer to when the network administrator wants to know when the client connected and what the client was granted access to. However, their out-of-band nature makes them poor resources for determining what the client actually did while connected to the network.



**Q.5) Explain the structure of Google App Engine’s Data Store and its underlying technologies.**

Ans.)

**Data Store :**

Cloud Datastore is a highly scalable NoSQL database for your applications. Cloud Datastore automatically handles sharding and replication, providing you with a highly available and durable database that scales automatically to handle your applications' load. Cloud Datastore provides a myriad of capabilities such as ACID transactions, SQL-like queries, indexes, and much more.

* **Simple and integrated**

With Cloud Datastore's RESTful interface, data can easily be accessed by any deployment target. You can build solutions that span across App Engine and Compute Engine and rely on Cloud Datastore as the integration point.

* **Fast and highly Scalable**

Focus on building your applications without worrying about provisioning and load anticipation. Cloud Datastore scales seamlessly and automatically with your data, allowing applications to maintain high performance as they receive more traffic.

* **Easy-to-use query language**

Cloud Datastore is a schemaless database, which allows you to worry less about making changes to your underlying data structure as your application evolves. Datastore provides a powerful query engine that allows you to search for data across multiple properties and sort as needed.

**Features :**

* **Rich admin dashboard**

View entity statistics, query your database, view indexes, and back up/restore your data.

* **Diverse data types**

Cloud Datastore supports a variety of data types, including integers, floating-point numbers, strings, dates, and binary data, among others.

* **Multiple access methods**

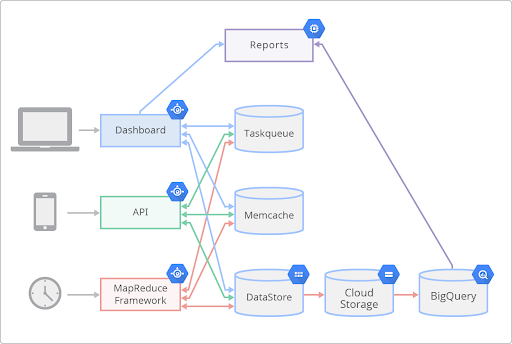
Access your data via our JSON API, open source clients, or community maintained ORMs (Objectify, NDB).

* **ACID transactions**

Ensure the integrity of your data by executing multiple datastore operations in a single transaction with ACID characteristics, so all the grouped operations succeed or all fail.

* **Fully manage**

Cloud Datastore is fully managed, which means Google automatically handles sharing and replication in order to provide you with a highly available and consistent database.



App Engine Flow diagram

**Entities :**

Objects in Datastore are known as entities. An entity has one or more named properties, each of which can have one or more values. Property values can belong to a variety of data types, including integers, floating-point numbers, strings, dates, and binary data, among others. A query on a property with multiple values tests whether any of the values meets the query criteria. This makes such properties useful for membership testing.

**Kinds, keys, and identifiers**

Each Datastore entity is of a particular kind, which categorizes the entity for the purpose of queries; for instance, a human resources application might represent each employee at a company with an entity of kind Employee. In addition, each entity has its own key, which uniquely identifies it.

**Ancestor paths**

Entities in Cloud Datastore form a hierarchically structured space similar to the directory structure of a file system. When you create an entity, you can optionally designate another entity as its parent; the new entity is a child of the parent entity (note that unlike in a file system, the parent entity need not actually exist). An entity without a parent is a root entity. The association between an entity and its parent is permanent, and cannot be changed once the entity is created. Cloud Datastore will never assign the same numeric ID to two entities with the same parent, or to two root entities (those without a parent).

**Queries and indexes**

In addition to retrieving entities from Datastore directly by their keys, an application can perform a query to retrieve them by the values of their properties. The query operates on entities of a given kind; it can specify filters on the entities' property values, keys, and ancestors, and can return zero or more entities as results. A query can also specify sort orders to sequence the results by their property values.

**Transactions**

Every attempt to insert, update, or delete an entity takes place in the context of a transaction. A single transaction can include any number of such operations. To maintain the consistency of the data, the transaction ensures that all of the operations it contains are applied to Datastore as a unit or, if any of the operations fails, that none of them are applied.

**Transactions and entity groups**

Only ancestor queries are allowed within a transaction: that is, each transactional query must be limited to a single entity group. The transaction itself can apply to multiple entities, which can belong either to a single entity group or (in the case of a cross-group transaction) to as many as twenty-five different entity groups.

**Cross-group transactions**

A transaction on entities belonging to different entity groups is called a cross-group (XG) transaction. The transaction can be applied across a maximum of twenty-five entity groups, and will succeed as long as no concurrent transaction touches any of the entity groups to which it applies.

**Datastore writes and data visibility**

Data is written to Datastore in two phases:

In the Commit phase, the entity data is recorded in the transaction logs of a majority of replicas, and any replicas in which it was not recorded are marked as not having up-to-date logs.

The Apply phase occurs independently in each replica, and consists of two actions performed in parallel:

* The entity data is written in that replica.
* The index rows for the entity are written in that replica. (Note that this can take longer than writing the data itself.)

**Datastore statistics**

Datastore maintains statistics about the data stored for an application, such as how many entities there are of a given kind or how much space is used by property values of a given type.