Market-based management of clouds -

The real potential of cloud computing resides in the fact that it actually facilitates the establishment of a market for trading IT utilities.

Market oriented cloud computing (MOCC) -

MOCC is the presence of a virtual marketplace where IT services are traded and brokered dynamically.

A reference model for MOCC -

MOCC originally from the coordination of several components - service consumers, service providers, and other entities that make trading between these two groups possible.

A global view of MOCC -

Three major components of cloud exchange one -

1. **Directory** - The market directory contains a listing of all the published services that are available in the cloud marketplace.

2. **Auctioneer** - The auctioneer is in charge of keeping track of the running auctions in the marketplace and of verifying that the auctions for services are properly conducted and that malicious market players are prevented from performing illegal activities.

3. **Bank** - The bank is the component that takes care of the financial aspect of all the auctions happening in the virtual marketplace.
Four major components of the architecture are:

1. Users and Brokers - They originate the workload that is managed in the cloud data center.
2. SLA resource allocator - The allocator represents the interface between the data center and the cloud service provider and the external world. It has:
   a) Service Request Examiner and Admission Control module - This module originates in the front-end and filters user and broker requests in order to accept those that are feasible given the current status of the system and the workload that is already processing.
   b) Pricing module - This module is responsible for charging users according to the SLA they signed.
   c) Accounting module - This module maintains the actual information on usage of resources and stores the billing information for each user.
   d) Dispatcher - This component is responsible for the low-level functions that are required to realize admitted service requests.
   e) Resource Monitor - This component monitors the status of the computing
Resource, either physical or virtual.

(vi) Service Request Monitor - This component keeps track of the execution progress of service requests.

(3) Virtual Machines (VMs) - VM constitute the basic building blocks of a cloud computing infrastructure, especially for IaaS providers.

(4) Physical Machines - At the lowest level of the reference architecture resides the physical infrastructure that can comprise one or more data centers.

2. **Federated cloud / Intellcloud**

There are enablers for M0CC since they provide means for interoperability among different cloud providers.

**Characterization and Definition**

The terms cloud federation and Intellcloud, often used interchangeably, convey the general meaning of an aggregation of cloud computing providers that have separate administrative domains.

Cloud federation manages consistency and access controls when two or more independent geographically distinct clouds share either authentication, file, computing resources, commands, and control or access to storage resources.

Intellcloud (Cloud of clouds) refers mostly to a global vision in which interoperability among different cloud providers is governed by standards, thus creating an open federated platform where applications can shift workloads and fully compare services from different sources.

**Cloud federation stack**

- **Conceptual level**
  - Motivations, Advantages, Opportunities, Obligations

- **Logical and Operational level**
  - Federation Model, Cloud Service Provider, Agreements
  - Market and Pricing Models, SLAs

- **Infrastructure level**
  - Protocol, Interfaces, Standards, Programmatic Interoperation
  - Federation Platforms (RESERVOIR, Intellcloud)
Each cloud federation level presents different challenges and operates at a different layer of the IT stack.

(1) Conceptual level - It addresses the challenges in presenting a cloud federation as a favorable solution with respect to the use of services offered by multiple cloud providers. Elements of concern at this level are-
(i) Motivations for cloud providers to join a federation.
(ii) Motivations for service consumers to leverage a federation.
(iii) Advantages for service-consumers/providers in using their services on other providers.
(iv) Obligations of providers once they have joined the federation.
(v) Trust agreements between providers.
(vi) Transparency versus concerns.

(2) Logical and Operational level - It identifies and addresses the challenges in defining a framework that enables the aggregation of providers that belong to different administrative domains within a content of a single overlay infrastructure, which is the cloud federation. It is important at this level to address the following challenges-
(i) How should a federation be presented?
(ii) How should we model & represent a cloud service, provider, or an agreement?
(iii) How should we define the rules & policies that allow providers to join a federation?
(iv) What are the mechanisms in place for settling agreements among providers?
(v) What are the provider responsibilities with respect to each other?
(vi) What should providers & consumers take advantages of the federation?
(vii) Which kinds of services are more likely to be leased or bought?
(viii) How should we price resources that are leased, and which fraction of resource should be we lease?

(3) Infrastructure level - It addresses the technical challenges involved in enabling heterogeneous cloud computing systems to interoperate seamlessly. At this level, it is important to address the following issues-
(i) What kind of standards should be used?
(ii) How should design interfaces and protocols be designed for interoperability?
(iii) Which are the technologies to use for interoperability?
(iv) How can we realize a software system, design platform components, and services enabling interoperability?

3: Third party cloud services

This can be the result of adding value to existing cloud computing services, thus furnishing customers with a different and more sophisticated service.

Meta CDN

It parallels nas with the high-level services of a Content Delivery Network (CDN) for content distribution and integrates with the low-level interfaces of storage clouds to optimally place the user content in accordance with the enriched geography of its demand.

Four deployment options are:

(i) Coverage and performance-optimized deployment
(ii) Direct deployment
(iii) Cost-optimized deployment
(iv) DDoS-optimized deployment

---

**Diagram:**

```
+-----------------+ +-----------------+ +-----------------+
| USERS           | ---| SLA           | ---| SLA           |
|                 |    |               |    |               |
|                 |    | WEB PORTAL    |    | WEB SERVICE ACCESS |
|                 |    |               |    |               |
|                 |    |               |    | Meta CDN      |
|                 |    | Meta CDN LOAD REDIRECTOR | Meta CDN DATABASE | Meta CDN MANAGER |
|                 |    | CONNECTOR     |    |               |
|                 |    | CLOUD         |    | Meta CDN ARCHITECTURE |
```

---

ShortCloud

It is an online portal that implements a virtual marketplace, where sellers and buyers can register and trade cloud computing services.

For service consumers, it acts as a market directory where they can browse.

For service providers, it constitutes an opportunity for advertising their offerings.
CASE STUDY -

Google App Engine -

It is a PaaS implementation that provides services for developing and hosting scalable web applications.

Application image of Google resources and services is modeled on App Engine, which bills users when their applications finish their free quotas.

Architecture -

Google App Engine Infrastructure

Four major components -

1) Infrastructure - server usage is efficient, also responsible for monitoring application performance and collecting statistics on which the billing is calculated.

2) Run Time Environment - Represents the execution context of applications hosted on App Engine. Provide sandboxed runtime support to run Python, Java, and Go.

3) Storage - Provides in-memory cache, storage for semi-structured data and long-term storage for static data.

4) Services - Application and compute services.

my companion
Application Xacive - URLfetch, Memcache (Database), Mail and Instant messaging, account management, image manipulation

Compute services - Task Queue, Job jobs (Schedule the required operations)

2. **Microsoft Azure**
   - It is a cloud operating system built on top of Microsoft data center infrastructure and provides developers with a collection of services for building applications with cloud technology.

3. **Amazon Web Services**
   - It is a platform that allows the development of flexible applications by providing solutions for elastic infrastructure scalability, managing and data storage.

**Management Portal**

```
<table>
<thead>
<tr>
<th>Compute</th>
<th>Storage</th>
<th>Networking</th>
<th>Identity</th>
<th>Marketplace</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Azure</td>
<td>AppFabric Service Bus</td>
<td>AppFabric Integration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Enterprise Level Infrastructure**

Services can be managed and controlled through the Windows Azure Management Portal.

**Compute services** - Web role, Worker role, VM role

**Storage services** - Blob (Binary large objects), Azure drive, Table, Queue

Core infrastructure: AppFabric - middleware for developing, deploying, and managing applications on the cloud.
Two popular services of Amazon are Amazon Elastic Compute (EC2) and Amazon Simple Storage Service (S3).

- **Compute Services**: Amazon machine images, EC2, Amazon Elastic MapReduce, AWS Elastic Beanstalk, AWS CloudFormation, Auto Scaling.
- **Storage Services**: S3, Elastic Block Store, Elastic Cache, SimpleDB, Relational Database Service (RDS), CloudFront, Import/Export.
- **Communication Services**: Simple Queue Service (SQS), Simple Notification Service (SNS), Simple Email Service (SES), Direct Connect, Route 53.
- **Additional Services**: CloudWatch, Dev-Pay, Flexible Payment Service (FPS), Fulfillment Web Service (FWS), S3 Transfer Acceleration.

### Hadoop - (developed by Apache)

Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware.

- It provides massive storage for any kind of data, enormous processing power, and the ability to handle virtually limitless concurrent tasks and jobs.
- It is written in JAVA.
- Concept of MapReduce are used here.

### Anuka - (developed by Microsoft)

Anuka is a platform and a framework for developing distributed applications on the cloud.

- One of the key features of Anuka is the ability of providing different ways for enpointing distributed applications by offering different programming models.

**Two key components** -

1. **SDK (Software Development Kit)**
2. **Runtime Engine and Platform**
   - Concept of MapReduce are used here.