1. Introduction

1.1 Cloud Computing

A cloud is a pool of computing resources (i.e. set of machines, software, hardware, application programs, network) that can be shared with the use of internet, users of a cloud can easily access these shared resources and this is known as cloud computing. And it eliminates the overhead of installing and providing a suite of computing resources for each system individually. It is based on pay as you go model.

In contrast to managing resources locally within a college or some organization, cloud computing deals with highly scalable computing resources that are to be delivered over the internet. Applications, we services, infrastructure networks these are the resources offered by cloud computing. Cloud computing is very cost effective as we only have to pay for the resources that we required.

Flexibility is another interesting feature of cloud computing, with changing needs of IT cloud computing has made it possible for the organizations to quickly adapt these changes.

The impact of cloud computing over IT companies is such that, these organizations are now making best use of web applications, infrastructure and interoperability offered by computing entities to deliver IT services.

Technical expertise of cloud providers provide them the ability to deal with the management, backups, recovery, maintenance etc. Because of this, cloud services clients may experience improved reliability.

As to cope up with the changing needs and demands of IT industries, cloud computing makes it easier for the organizations to adapt to the changing needs. Standard protocols and methods are required so as to achieve effective communication between the cloud computing entities.

The web services provided by cloud computing are offered to:

- Organizations, general public, companies, business corporations.

1.1.1 Cloud Services Styles:

IaaS provides hardware resource services.

PaaS provides programming and runtime environment e.g. language runtime/framework.

SaaS delivers software application programs.

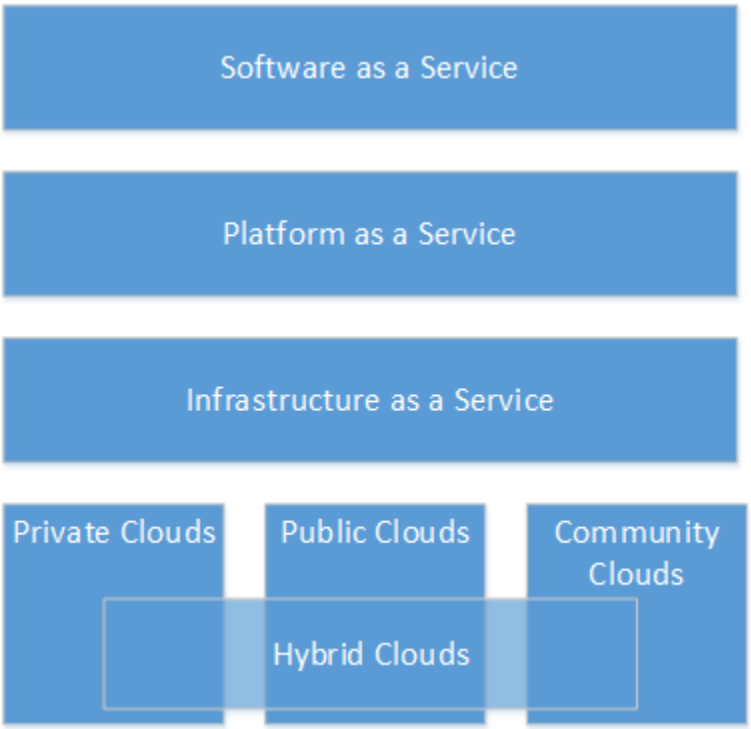


Fig.1.1. Cloud Services Style and types

1.1.2 Cloud Types:

Public cloud is a cloud which is managed by a service provider and is highly consistent. Its services are based on pay as go basis. Deployment of public cloud is much easier, cost effective and scalable. And offer services that can be accessed by anyone from anywhere.

Private cloud limits its access only to a particular organization or an enterprise. It provides enhanced security, privacy and greater flexibility and its services are open only for a limited number of people. Private cloud appears to be emerging as a favorite for organizations that do

not want to depend on services offered by a third party but like to have a proper control over their data.

Hybrid cloud is based on the concepts of both, public cloud and the private cloud. Hybrid cloud can be implemented by any of the following method: a service cloud provider contains a private cloud which is responsible to establish an association with public cloud provider, or an association can be formed between a public cloud provider and a service provider that offers private cloud platforms.

Virtualization has brought up the concept of virtual machines in cloud computing by making the software implementation of the physical machines. The functionality of virtual machines is similar to that of existing machine, as each VM is comprised of its own kernel, operating system, applications etc.

Hypervisor is responsible for concurrent execution of multiple virtual machines on top of it. Scalability is one of the most crucial feature of cloud computing so in order to achieve scalability, virtualization is the key.

Benefits of virtual machines:

- Operating system can be run in the absence of physical hardware.
- Easier to build new machines.
- It's easier to migrate virtual machines.
- More machines can be imitated compared to their physical availability.

Opportunities:

- In order to access web services offered by cloud computing, users do not require knowledge of underlying infrastructure.
- It lowers the cost to install a software suit on each and every system individually.
- It's based on pay as you go model.
- Remotely stored web services and resources can be accessed from all over the world.

Cloud computing is cost effective because in order to implement web services we don't require a high computing power. When users have to deal with web based applications, not much storage capacity is required.

In cloud computing as there are a less number of programs placed in the memory, it will automatically increase the performance of the system. When the user deals with web-based applications, cloud computing provides automatic updation of software and the latest version of the software is provided.

In cloud computing, there is no such issue of compatibility when users have to share some documents or applications. Cloud computing offers web services for public use, anyone can access these services or resources from anywhere via internet.

Users can freely share documents and applications, this will result in better collaborative environment.

1.1.3 Cloud Benefits:

- **Dynamic:** Dynamic framework of cloud offers scalable, virtualized and a more secure environment.
- **Flexible:** In order to cope up with fast changing enterprises, cloud computing delivers economic efficiency and flexibility.
- **Quality of service:** It deals with:
    - Guarantee a specific level of performance.
    - Virtual distribution of resources.
    - Every customer must have dedicated servers.
- **Resilient:** Resilient nature of cloud deals with specific kind of failure and help to make the system fully functional from the customer point of view inspite of the failure.
- **Ubiquitous environment.**

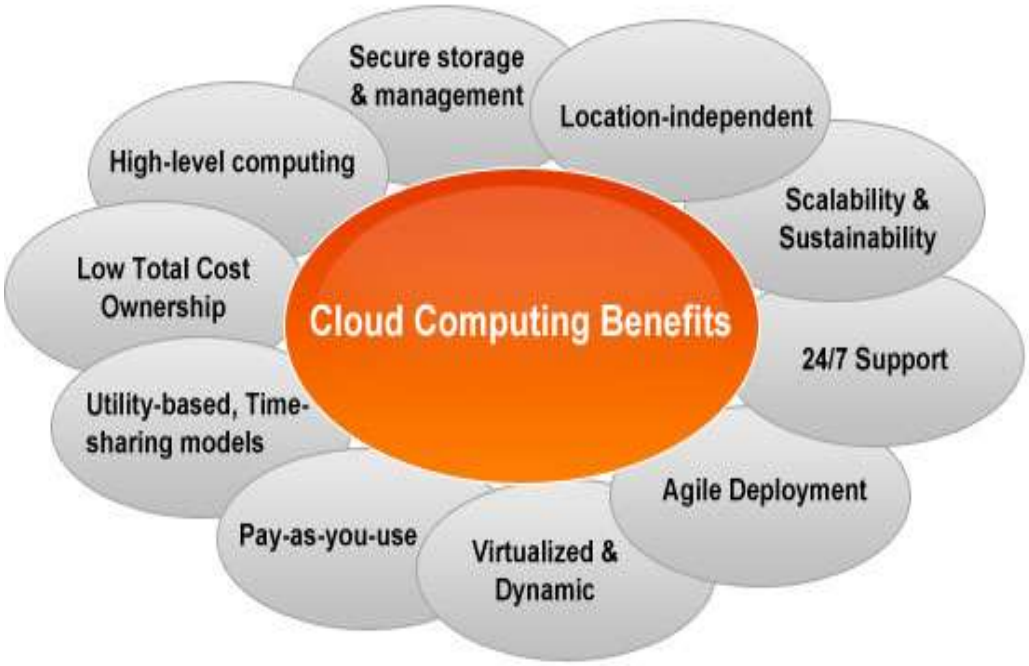


Fig.1.2. Cloud computing benefits

Cloud computing offers a number of advantages but there are some issues also:

The first one is the availability issue, second is the security issue, sometimes organizations and enterprises feel that their important data stored in someone else server over the internet is at risk or unprotected. And next is the reliability issue.

1.2 Volunteer Computing

Unlike cloud computing, volunteer computing [1] is a kind of distributed computing where idle computing resources are harnessed so as to accomplish distributed independent tasks. Volunteer computing propose an advantage of shared volunteer resources that can be used by various scientific and research projects in order to carry out the needs and requirements of the project.

Volunteer computing is based on exploiting the computing resources volunteered by individuals for a particular project. (eg. SETI@home involves more than 200,000 users and FOLDING@home is a collection of 450,000 CPU, PCs etc.).

Volunteer computing has been successful in collecting a massive number of resources from people across the world. In order to carry out computational needs of an organization for a particular project, SETI@home project offers a collection of computing resources donated by individuals across the globe.

Volunteer computing is based on a central server which is responsible for managing the resources and for allocating the jobs to the accessible computers.

Science and research projects requires extremely high computing power and for this purpose a large number of a large number of dedicated volunteers and their computing resources have turned out to be extremely useful. In volunteer computing multiple computers are responsible to carry out a complex problem which is further divided into various jobs.

To compete with the problems of scientific and research projects, high computing power is not sufficient, but the mutual intelligence power of volunteers is also required.

Large scale volunteer computing systems are based on client server architecture which is defined in the form of layers, where each layer has its own particular function to perform. In this architecture the concept of agent and broker is there, which is responsible for the communication between a project's client and server. It is also known as middleware layer with which the appropriate data that has been processed can be sent back to the scientists.

In order to perform scientific computing in a distributed way, volunteer computing make use of computational power proposed by general public. Volunteer computing is a technique that let its internet users to donate or offer their computing resources in a easier, faster and a cost-effective manner. And the users and the volunteers who offer their computing resources are considered to be as unidentified.

Volunteer computing is basically a composition where volunteers contribute their computing resources to various scientific and research projects so as to carry out distributed computing. Volunteers can be any user or organizations who may want to contribute their idle resources that are kept unused for a long period of time. But a trust relationship should be there between the volunteers and the projects. Projects include various scientific and research projects that require very high computational power.

Importance of volunteer computing:

- Volunteer computing is designed to offer massive computing power which is a result of a large number of PCs that exist in the world. This massive computing power is required to carry out scientific research projects and development.

- A research project that cannot afford expensive computing resources can make best use of volunteer computing as computing power offered by volunteer computing cannot be purchased, it can only be earned.

- People are now taking interest in various scientific and research projects and volunteer computing has made it possible.

Volunteer computing is a concept which is based upon centralized servers. There is no end to end communication where central server is not required. The main idea behind volunteer computing is that the computing resources that are managed by individuals or some organizations are kept unused for most of the time and while being idle, these computing resources can be used to solve complex problems as they need a huge amount of computing power.

Volunteer computing has made a magnificent effort in bringing a huge amount of computing power to researchers and also allows scientists and researchers to experience the strength of volunteer networks based on internet.With the help of computing power contributed by various volunteers, scientists have proposed some volunteer computing projects.

Working mechanism of volunteer computing projects is as follows: volunteer just have to download a block of data and need to execute it. After the execution, the processed block of data or the result will be uploaded on the central server of volunteer computing project.

Volunteer computing projects goals:

- For the betterment of research and development, volunteer computing projects main goal is to utilize the contributed resources volunteered by people so as to generate scientific outcomes.

- Another goal of volunteer computing projects is to reach out to people and make them aware about these scientific and research development. A volunteer associated with any one of volunteer computing project can learn a lot about the research agenda and also about the science behind these projects.

- A trust factor should be maintained so that volunteers of computing projects must not feel insecure about their sensitive data which is being used within the projects.

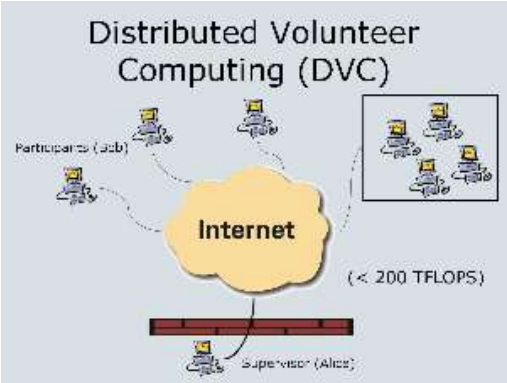


Fig.1.3. Volunteer computing

There are multiple distributed computing environment and each of them incorporate their own pool of resources.

- As in case of “grid computing”, the pool of resources is a combination of number of computers held by a university.
- In case of cloud computing, it’s the number of servers possessed by a company.
- And in case of volunteer computing, it’s the total number of attainable personal computers (PCs) that contributes towards forming a resource pool. In order to meet the computational requirements and to compete with the faster changing technology, it is difficult to upgrade or restore the super-computers but personal computers can easily be upgraded.

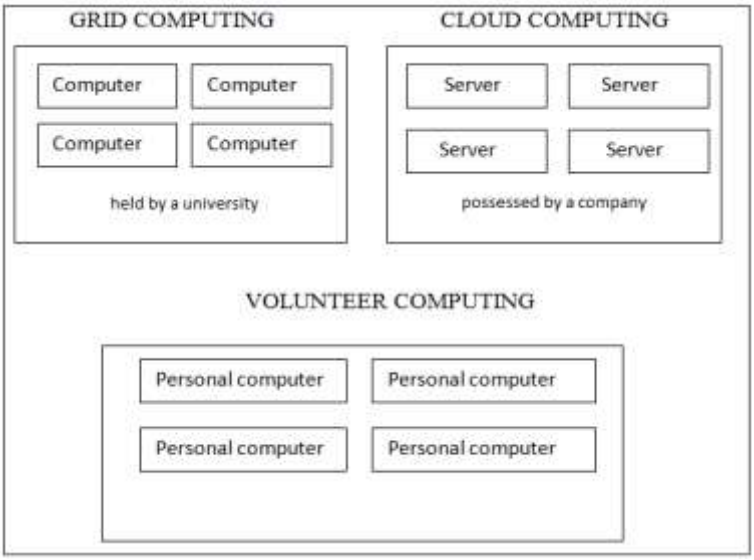


fig.1.4. pool of resources

Advantages of volunteer computing:

- Utilization of idle resources.
- Cost reduction.

1.3 Volunteer Cloud Computing

So here comes the concept of volunteer cloud computing [2][3], which is the combination of the concepts and the benefits of cloud computing and volunteer computing.

In volunteer cloud computing, idle and non-dedicated resources [1] that are not designed to be cloud infrastructure are harnessed so as to provide cloud capabilities or services. By volunteer cloud computing, users and contributors can experience the impact of cloud computing by allowing users to share any type of services be it physical resources, software resources or applications.

Volunteer cloud computing [3] is responsible for sharing network connected resources so as to support distributed computing.

VOLUNTEER CLOUDS vs. DISTRIUTED SYSTEMS

	Volunteer cloud	Commercial cloud	Grid	Desktop Grid
Elasticity	√	√	X	X
Virtualisation	√	√	X	X
Idle resources	√	X	X	√
Ease of use	√	√	X	X

Fig.1.5. Comparison between volunteer cloud and distributed system

The above figure is a comparison table which presents a comparison between volunteer clouds and distributed systems.

Idle, unused hardware and software resources are utilized or harnessed to enhance the quality of service. These highly distributed computing resources are made available so as to form a volunteer cloud. To accomplish distributed independent tasks and to deploy distributed applications, pool of resources are brought into the cloud.

There exists a large number of cloud users that are incapable of experiencing the impact of cloud computing and are failed to grasp the benefits of private and public clouds, so volunteer cloud computing has come up with a solution as it allow its users (consumers and contributors) to share and provide various computing resources and can run over an existing highly distributed network or hardware.

To be successful it is necessary for an organization to effectively use their data and to manage the data efficiently and must have the ability to utilize unused computing resources. So a volunteer cloud is based on the computing resources volunteered by the users of a network, which allows various organizations to collaborate while competing with each other.

Volunteer cloud computing has come up as a solution for researchers, scientists that have to deal with large computations exceedingly and need a large storage to store massive quantity of data.

Volunteer cloud computing is a technique that allow its clients to get the access by exploiting volunteer computing resources that are donated by individuals across the globe. To solve complex problems, IT companies are moving their solutions to the volunteer cloud, as volunteer clouds provides a way to cut down IT cost so that companies can plan a valuable financial management.

1.3.1 Benefits of Volunteer Cloud Computing:

- **Utilization of idle resources**:
  Volunteer cloud computing makes use of non-dedicated idle resources to build a cloud and so as to deploy cloud services. Hence, it overall increases the efficiency of the system by exploiting these underused resources.

- **Cost reduction [5]**:
  As volunteer cloud computing deals with the computing resources volunteered by individuals across the world, it prevents the organization, scientists and researchers from making any kind of investment in the resources. So it eliminates the requirement to have dedicated resources, as volunteer computing resources altogether generate a massive computing power which is sufficient to fulfill the needs and requirements of the projects and business.

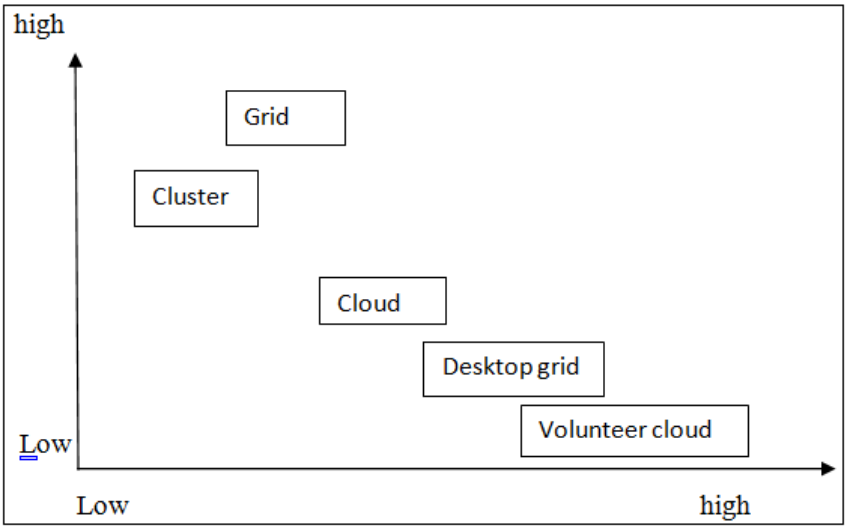


fig.1.6. platform performance vs. cost analysis

- **Reduce overall power consumption:**

  Presence of volunteer cloud decreases the need of particular framework for excessive power, cooling systems and battery backup etc.

- **Computing resources altogether provides a great computational power:**

  To solve complex problem, a large computing power is required, so if volunteer computing resources are used altogether then it can solve any kind of problem and can serve a possible solution for any business and scientific projects. These non-dedicated resources are contributed by various individuals and this is why volunteer cloud is also termed as distributed cloud because it deals with highly distributed non-dedicated resources. And these distributed services can be used in a combined fashion so as to implement services for any association.

- **Reduce data migration cost:**

  Volunteer cloud computing helps in reducing the cost related to data migration as volunteer cloud is a collection of highly distributed non-dedicated resources, it can act as an advantage also, as these resources can be allocated to clients according to their location. In this way the cost related to data migration can be reduced and hence improve the performance.

So in this way volunteer cloud computing contributes towards achieving high performance by utilizing the idle resources and buy cutting the IT cost so that organizations can make better use of their budget. It also makes available these volunteer computing resources for the public use so that clients can easily get the access of these resources.

1.3.2 Volunteer Cloud Computing Architecture:

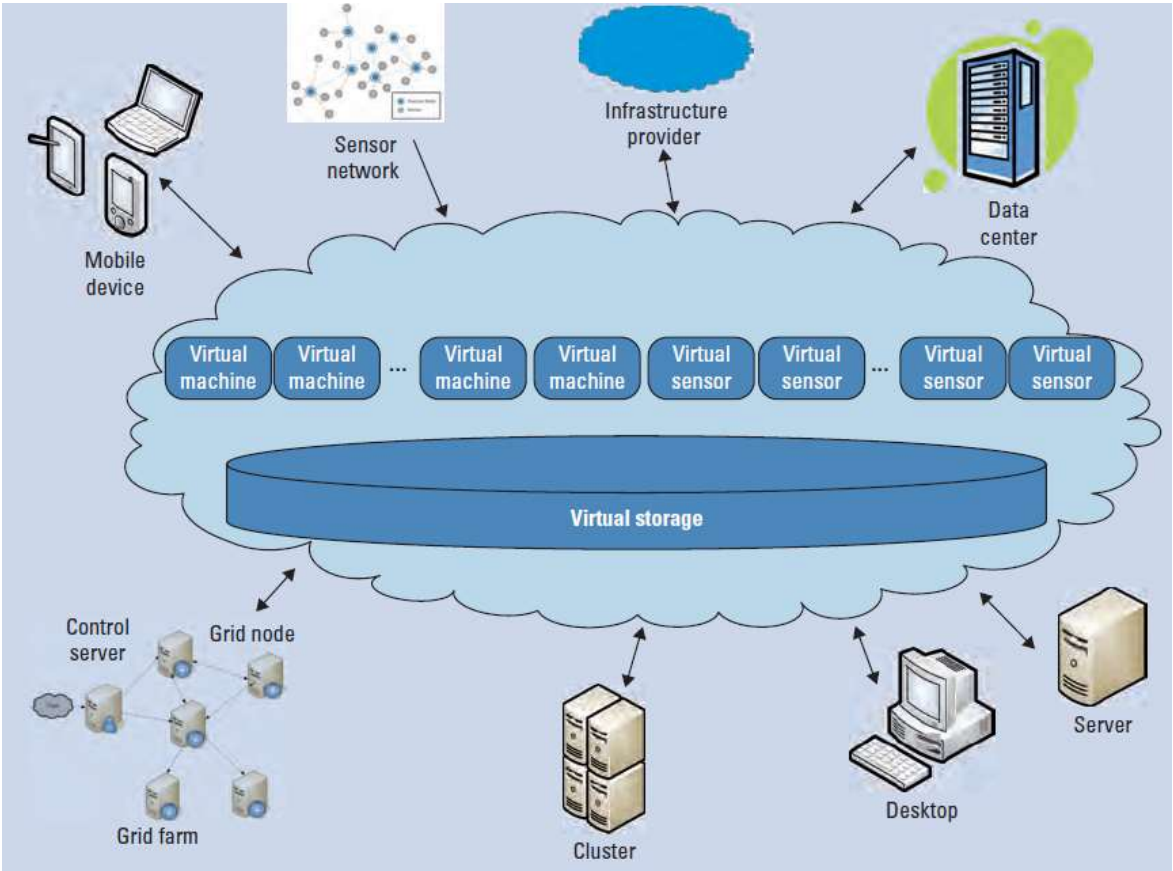


Fig.1.7. Volunteer cloud computing architecture

The above figure shows volunteer cloud computing architecture where there exist two types of clients: consumers who are responsible for consuming the volunteer computing resources and contributors who are allowed to contribute their idle resources and services in order to constitute and support a cloud. It is responsible for delivering the services to customers through an interface formed on SOA approach.

Clients of volunteer cloud can be both, consumer and contributor at the same time and can actively take part in building up a volunteer cloud so as to entrench an interactive session with the cloud. So enterprises can act as one of the clients that are free to contribute and consume non-dedicated computing resources. These enterprises can contribute their idle or unused resources to the cloud.

Scientific or research projects that cannot be able to have commercial cloud services can use volunteer clouds where contributors and users voluntarily share their resources.

For short and long term projects, our traditional IT companies have to spend a lot of time in deploying new hardware and software resources, by procuring and purchasing these new resources into their infrastructure.

Volunteer cloud computing offers cloud services that are based on non dedicated resources without charging and a way to cut down IT cost so that companies can benefit from the well planned budget and can make the best use of it. We can make use of volunteer clouds, if a company wants to endeavor a project for a short duration of time. So it's time to move complex solutions to a volunteer cloud that can offer comparatively much faster response to the needs and requirements of business and can help companies lower overhead.

Consider few scenarios:

- where an organization utilizes the services of a cloud provider A, but if the cloud provider A's server suffers a failure, then organization can make use of volunteer computing services offered by volunteer cloud and can even bring their problems to the cloud. So communication is needed so that clients can easily establish a communication with the volunteer cloud and can take benefit from its computing services.

- To effectively manage the volunteer cloud architecture, there should exists a proper communication channel so that various volunteer cloud entities can communicate with each other and can efficiently provide a coordination amongst the various components of the volunteer cloud.

An enhanced interoperable environment should be made within the volunteer cloud architecture so as to manage and control the volunteer cloud entities. In order to perform a specific task there should exists a proper coordination between the components so that they can carry out any task.

- Suppose a large number of computing resources are there in an organization which are kept unused for a long time, then organization can contribute their idle resources to the volunteer cloud so that these resources can be utilized in a proper way.

So there is a need to have a strong, reliable, secure communication to provide an interoperable environment between volunteer cloud entities and also between volunteer cloud and its clients.

1.4 Challenges:

Important challenges [6] that should be taken into consideration in order to achieve better communication between volunteer cloud entities and also between volunteer cloud and its clients or commercial clouds are as follows:

- **ONE DIRECTIONAL COMMUNICATION**

  When volunteer cloud entities interact with each other and also with other clients and commercial clouds, one way communication is the biggest hurdle. Because if two parties wants to communicate with each other effectively then there should be a proper two way communication so that consumers and contributors can easily access and provide their computing resources and also various components of volunteer cloud can coordinate with each other so as to carry out any specific task. So our middleware should be such that it can help to establish a proper and clean two way communication.
- **AVAILABILITY**

  Resources should be made available easily as per the needs of the users. Enterprises concerns about whether volunteer computing resources will have sufficient availability or not and also find out for how much time the system is running perfectly by making use of presence information. So there should be a mechanism that will allow an entity to broadcast its presence information or availability to other entities through

communication. So to manage the availability of volunteer cloud entities, presence information can be used.

- **SECURITY**

  While working on cloud computing we also face security concerns & these security concerns are mainly divided into two categories:

  Security Issues faced by Cloud Provider:

  These are the security concerns faced by the organization providing software platform or infrastructure as a service via cloud.

  Security Issues faced by their Customers:

  These are the security concerns faced by the end users/customers who used the cloud provider's services.

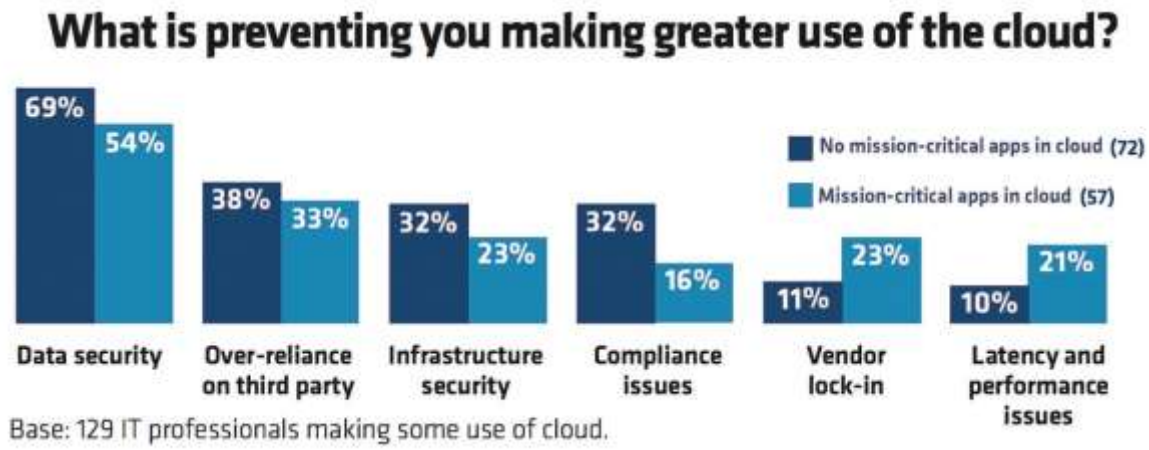


Fig.1.8. Challenges

In all the cases the cloud service provider should take care of the concern of the end user's security. They should thus ensure that their infrastructure is secure enough & the customer's data as well as application is securely protected. On the other hand the customers should also ensure that the service provider has taken proper measures to provide them the security & is following all the negotiated policies, constraints & requirements described in QoS.

The virtualization, apart from being a huge advantage of cloud computing, is also posing some serious concerns about security for the customers/end users who are using a public cloud services. Virtualization is known to change the relationship between the operating system & the hardware resources (it can be computing ability, storage capability or even routing or say networking power).

This whole phenomenon of virtualization infuses the need of one more layer called as virtualization layer. Until unless this layer is aptly configured, managed & protected, the security concerns will keep on hovering.

Some concerns that can keep haunting are the risk of compromising the Virtualization software or hypervisor. While these concerns seem to be theoretical, they also have proved their traces of existence. Say for example, the administrator's workstation's security gets compromised & thus the virtualization software gets malfunctioned & in turn affect the whole datacenter. This can also lead to the downtime for the service or it gets reconfigured as per attacker's will.

- Users should try to understand the risk involve in cloud computing like Cloud service provider's risks, their 3rd party service providers, probable attacks on data, downtime & monitoring mechanism to ensure tightly secured environment for the business.
- Till now there is nothing called uniform standards for fully protected data controllers. They are yet to evolve or say it's a on-going process of evolution of such standards.
- It is highly advised to know where your data is stored & what all local laws & jurisdiction is in place in the particular country.
- You'll lose the control over assets once they are out in the cloud.
- In order to trust your service provider, one should thoroughly go through the Service Level Agreement & be very clear about every policy, constraints & their own requirements.
- You not only lose control over your assets but also on the physical security of your Site.
- The assets can be seized by certain authorized government agencies if they found out that you or even the other companies with whom you are sharing the servers are violating any issues.

- If the cloud vendors are incompatible then in that there should be a mechanism in place so that you can retrieve & move your data.
- If certain set of data is encrypted, it should be very clear in the initial phase itself that who will shoulder the responsibility of encryption/decryption: you or your service provider.
- If your data transmitting over internet or stored in vendor's pool is it encrypted.
- Data Integrity speaks about the protection of data in its truest form. During e-commerce transaction you need to access Cloud provider's log & should therefore negotiate on the following:

  Data protection – how is your data protected?

  Identity management

  Physical and personnel security

  Availability

  Application security

  Privacy Issues

So when volunteer cloud entities communicate with each other or with other commercial clouds, clients, security is a matter of concern because volunteer cloud deals with highly distributed resources and it is also known as distributed cloud, so in order to achieve communication we need some effective mechanism that will provide support for authentication, encryption, data protection and integrity.

- **COMPATIBILITY**

  If we analyze today's scenario, no 2 clouds are similar when talk about both the aspect:

  1. Nature
  2. IT

  For example if we talk about Google's Cloud it is way too different than that of Microsoft's Cloud, which in turn different from any other cloud which exists today. And still we can't label proprietary as less useful.

  If we try to set up one analogy with the early age of personal computers, we'll found out why still we can't label proprietary as useless. During first wave of PCs, the market got

flooded with PCs by different makers with different hardware altogether & they were not even slightly compatible with each other. Programs written for one machine were not at all possessed any interchangeability with any other machine by any other maker. Be it Apple II with Atari, the Amiga or even the IBM PC itself.

There was no common platform available at that time which encouraged the cross compatibility among the machines. Even in such a grave anti cross compatibility environment, the huge potential developments did not stopped. Makers kept on moving with their remarkable developments & also in that era all these machines competed fiercely with each other based on their differences.

However it's totally understandable that today's situation is entirely different. Today users expect higher magnitude of cross compatibility between devices, between applications, between platforms & environment. But still there is something which makes each cloud entirely different, & surely it isn't much about the way the work on the inside, but the way interaction happens with the cloud, the way data gets into & out of cloud of each cloud, and management of functionality of each cloud is done very differently.

The first wave of cloud computing has a proprietary nature which is considered to be an evil necessity. Though it is good to make things less proprietary outside the cloud & when they interoperate among them. Also the standards that exists between cloud seems to be very odd since they are the development resulted from the consequences of what the users are actually using & not something which is developed in abstraction.

- **LATENCY**

There are always some physical laws in place which clouds have to follow. However following them makes running an application slightly problematic.

As we know all the clouds are different, but still there's one thing which is common among all of them & it's a general concept which says "Data is centralized but Users are distributed".

This concept therefore highlights the need of properly planned deployment which can thus avoid any significant latency issue between user & the application server. If we won't plan a proper deployment, it can increase the latency issue for the user.

There are following two things which we should take care of:

- A well planned & foresighted architecture & focused future planning is necessary because ultimately we are dealing with widely distributed & loosely integrated systems where all the user/contributors (data, application, user and machine) are situated in geographically diversified locations.

  Thus we should be ready with a design that can deal & accommodate successfully a mechanism to deal with the latency. Commonly used techniques are using buffers & cache mechanism & moving those components physically close which are interacting more.
- Try to minimize the chunk of information moving among clouds, this will help tackle the latency issue. Like within any organization, we've systems, bandwidth & performance to move huge information within the organization. Similarly if we are moving data within a single cloud based infrastructure it won't cause much of latency issue. But if the data is moving among the clouds or among various cloud providers then tackling latency will pose as an issue.
- The factor that takes cloud's latency to another level of complexity is distributed computing and virtualization. So in order to increase the performance we must try to achieve lower latency.
- Interaction between nodes of the same cloud is somewhat quicker than interaction between two different cloud infrastructures. So a middleware is required that can perfectly handle such delays.

1.5 Middleware

Middleware [7] is a software that can be epitomized as "software glue" that binds together various applications, operating systems and also allow software to be interoperable.

In distributed computing environments, middleware is mostly used as a software, that facilitate communication, interoperability and management of data.

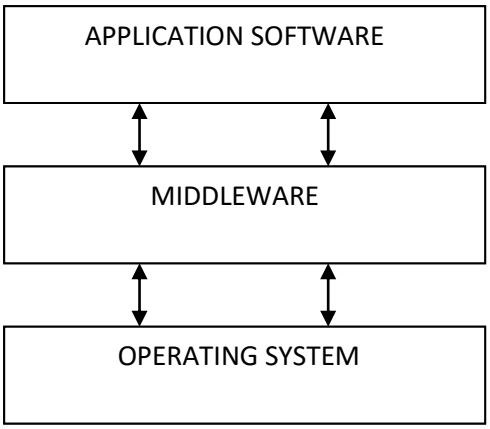


Fig.1.9. Middleware architecture

Multiple heterogeneous applications that are widely distributed in nature are brought into play by using middleware. It offers a connection between various applications and computing resources that can collectively form a large pool of resources to provide a massive computing power.

The services that are considered as middleware are as follows:

- Enterprise application integration.
- Data integration.
- Message oriented middleware.
- Object Request brokers.
- Enterprise Service Bus.

1.5.1 Messaging Middleware Architecture:

In order to compete with the complexities (reliability, availability, interoperability, security, maintaining quality of service) of distributed computing environment, there exist a service that can be considered as middleware, known messaging oriented middleware that can help towards a solution for above mentioned challenges.

Messaging oriented middleware (MOM) is a complete method which is used to develop an interoperable environment between different entities. MOM offers messaging capabilities with which client and server can communicate with each other via message passing. It is also responsible for transmission of asynchronous messages between client and server.

- **Message queuing**

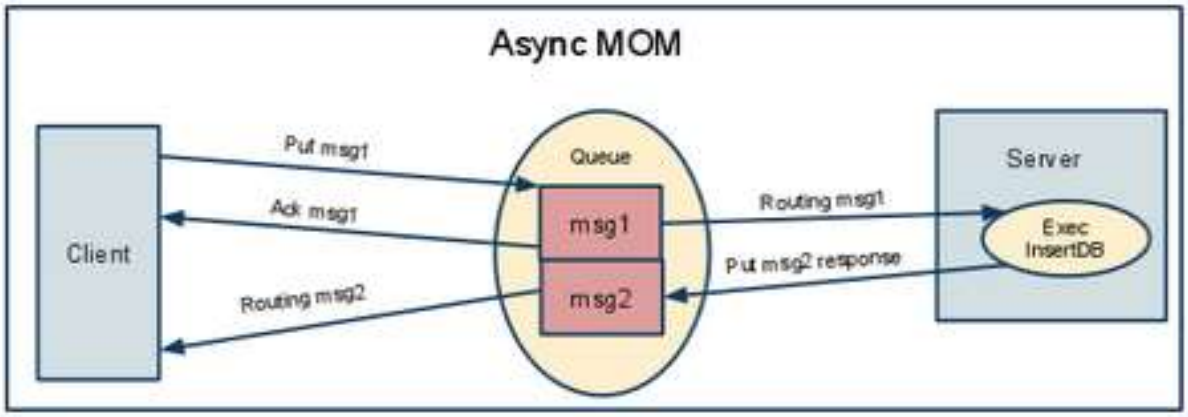


Fig.1.10. Message queuing

Message queue is one of the most important approach of messaging oriented middleware. With the message queue concept of MOM, clients are allowed to store messages in a queue and messages can be sent and received to and from a queue.

In order to establish asynchronous communication model, queues play a central role which is considered to be a destination for the clients to store their messages. So whenever a client wants to communicate with the server, client can place a number of requests in a queue without waiting for the server to response immediately. A particular format is used to sort the messages that are stored in a queue.

Messaging system includes a standard queue which is termed as FIFO (First In First Out) queue, as the name implies, the messages are retrieved from the queue in the same order in which they are sent. It means that the first message stored in the queue will be the first message to be fetched.

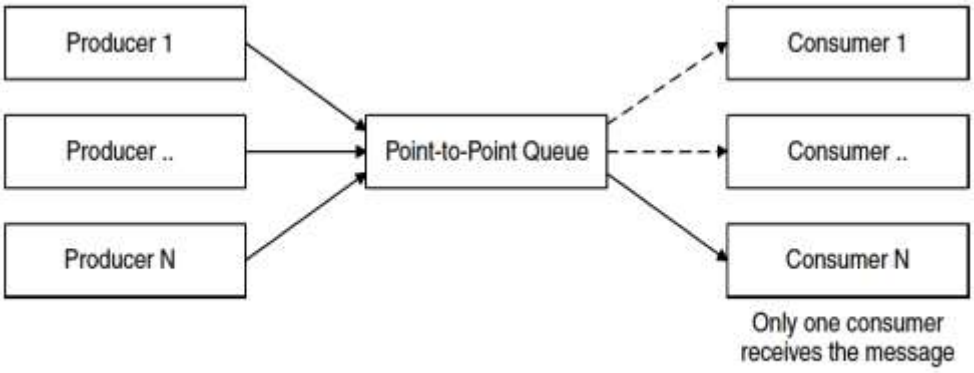
A queue is comprised of multiple attributes like name of the queue, size of the queue, algorithms, etc as it may be configured. MOM systems consist of various types of queues, with each queue performing different task.

There are two types of messaging models: point to point, publish subscribe. In both messaging models, messages are exchanged by using a queue.

- **Point to point**

This messaging model enables software entities to establish asynchronous communication by message exchange. This model is responsible for transmitting the messages from initiating clients to receiving clients through a queue.

In this model there can exists multiple initiating clients but at a time there should exist only a single receiving client.



Point-to-point messaging model

At a time a message can be delivered only to single receiver. There can be any number of receivers connected with the queue, but at a time only a single client can receive the message. This model focus on reliable delivery of messages and these messages are placed within the queue before a receiver receives them.

- **Publish subscribe**

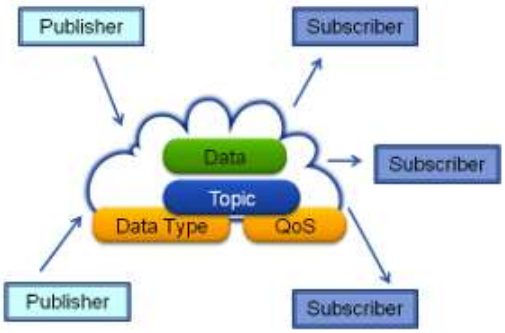


Fig.1.12. Publish subscribe model

This messaging model is a strong method which is used for broadcasting information between the communicating parties (initiating client and the receiving client). It is based on one-to-many and many-to-many cardinality.

There exist publish subscribe engine which is considered to be a destination for the communicating entities. Whenever a client (publisher) wants to send a message, it can publish that message over a particular channel and then the subscriber who wants to receive that message can subscribe to these channels.

A client can play a role of both the producer and the consumer. There exist various mechanism for publish subscribe model which provide support for multiple features.

A receiving client can receive messages by using two mechanisms: pull and push.

- **Pull**: Whenever a receiver wants to consume any message, it can ask the provider to look for the messages.
- **Push**: a consumer provides instructions to the provider to push messages as soon as it gets them.

1.6 SOA to the volunteer cloud

What role SOA (Service Oriented Architecture) will play in the volunteer cloud. SOA provides a flexible, connected platform that allows services to get merged and functions as SOA is also termed as service based platform. There are multiple attributes of SOA that are required to be implemented in order to place SOA with the volunteer cloud. As volunteer cloud computing paradigm includes services, so it is easier to place SOA with our cloud. SOA and volunteer cloud have many common features:

- Agility
- Flexibility
- Based on services
- Utilization of services

SOA architecture deals with reusability and sharing of services that it creates. Applications can be converted into services that are shared and reused.

The number of benefits offered by combining SOA with volunteer cloud are as follows:

- Reduce time
- Improved collaboration
- Customer satisfaction
- Business growth

In order to offer services into a volunteer cloud computing environment, we must require to take some decisions.

- **Reusability support**: at a time a service can be accessed by multiple clients.
- **Managing services:** it's the responsibility of SOA to manage the services within the volunteer cloud computing environment.
- **Security and access control:** in order to deploy such an environment there is a requirement for effective security and some control policies.

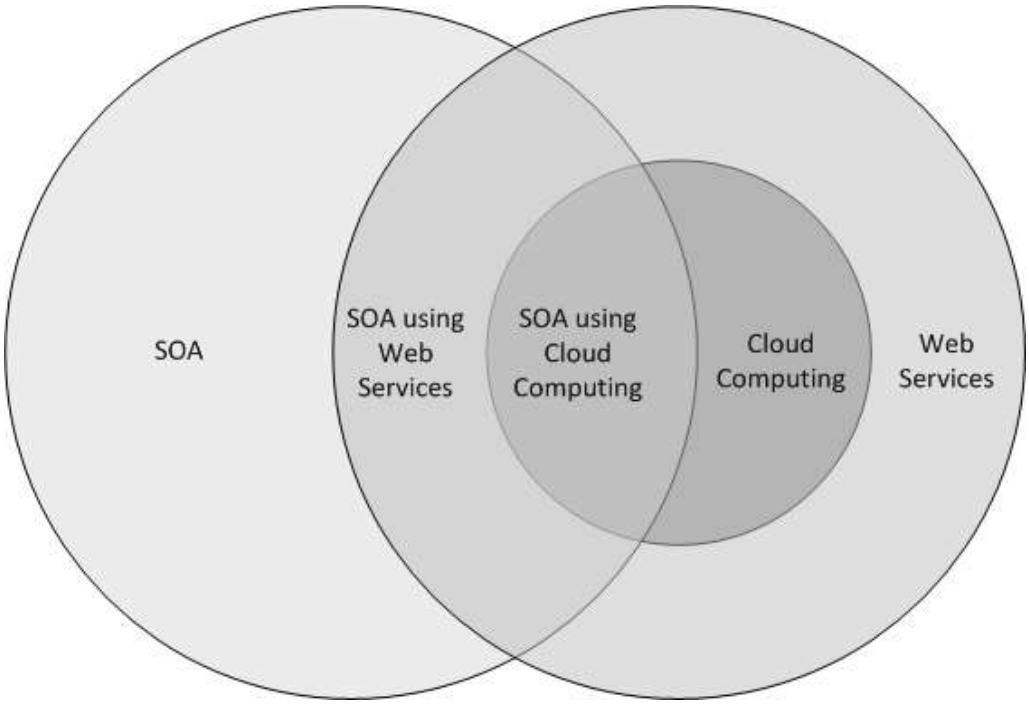


Fig.1.13. SOA to the cloud

At the volunteer cloud computing side, in order to deploy the services we need to follow some set of requirements:

- Services can be accessed easily
- Services usage must be **highly available**, **fast** and on **low cost**.
- **Service discovery**
- **Security and privacy**.

In order to cope up with the growing demands of IT industries, SOA comes into play as it provides reusability of services and also offer organizations the ability to merge their external and internal computing resources.

Via SOA interface, the volunteer cloud can be accessible. In order to execute mission critical applications at low cost, SOA can be an extremely powerful approach. To create and modify the volunteer cloud services SOA techniques can be useful. SOA techniques include:

- Creating and modifying the services.
- Services reusability
- Check out the process of SOA
- Bringing the services on a large scale.
- Discovering services.
- Eliminating services at the end.

1.7 Our Contribution

In this thesis, an enhanced interoperable environment is developed where volunteer cloud entities can freely communicate with each other and with other commercial clouds and clients also via XMPP (Extensible messaging and presence protocol). In this regard, XMPP protocol is implemented over middleware layer so as to form XMPP based messaging oriented middleware because XMPP is considered to be an ideal middleware protocol. And also it can help in overcoming the challenges mentioned above that should be taken into consideration. XMPP can be implemented over different kinds of levels so as to offer diverse services as well as support for authentication and encryption that can serve this purpose completely.

XMPP is based on XML (Extensible Markup Language) and offers a wide range of services.

- Firstly, it will uncover some solutions provided by XMPP for the challenges mentioned above.
- Next, it will include two different scenarios, one for the communication between the volunteer cloud entities. And the other one defines the communication between the volunteer cloud and the clients or we can say that clients residing on different network. Here, client can act like any of the following: commercial cloud, users, and contributors.
- The second scenario will further include the explanations using diagrams and the implementation.

Organization

This thesis is organized as follow:

- The next section illustrates the literature review or related work and that will include the work about Nebula, Volunteer cloud computing architecture and CLOUD@home project.
- Next section includes proposed work in which we present a working mechanism and the implementation.
- And finally end up with a result and a conclusion.

2. Literature Review

This section presents the following related work:

Nebula, volunteer cloud computing architecture, CLOUD@home. All these projects define the use of unused and idle non-dedicated resources to build a cloud.

2.1 Nebula

Nebula architecture [3] is a collection of database, a master node, data and execution nodes and each of these components has its own specific purpose. As the information about these nodes (master, data and execution nodes) are maintained using database by forming a list. From time to time the list keeps on updating so as to keep a record of the new nodes that are introduced to the cloud and the nodes that are no longer part of the cloud.

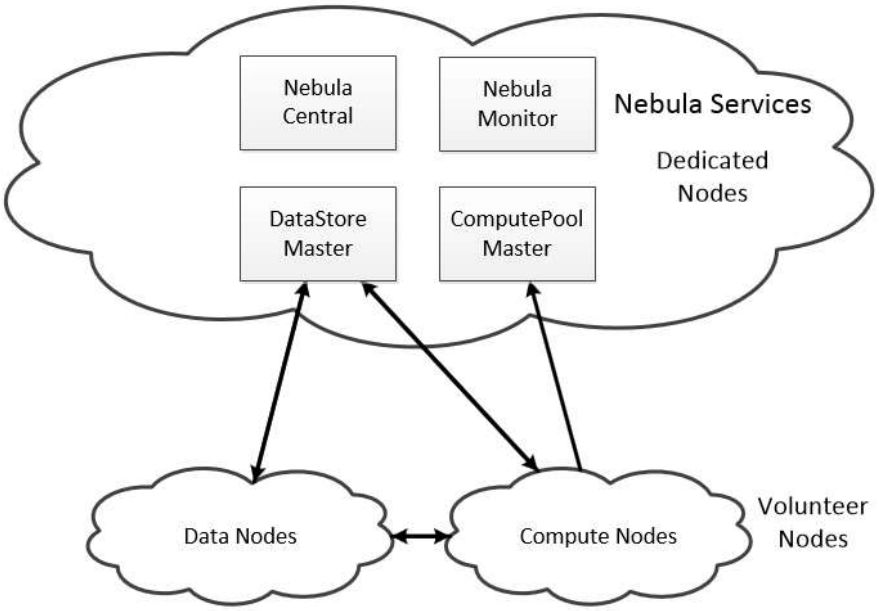


Fig.2.1. Architecture

- The management and allotment of jobs are the functions carried out by a master node that acts as an interface between the Nebula cloud and its clients.
- Data nodes are responsible for monitoring the data which will be executed by execution nodes.

- Execution node, it first gets a task from the master node and then execute that particular task and it is also responsible to fetch the data from the data nodes.

An experiment was conducted to compare the performance of Nebula cloud and commercial cloud (in this scenario Amazon) and according to the results, for parallel applications Nebula's performance was found more appropriate than Amazon's.

Chandra and Weissman comes up with a concept of Nebula that defines a combined pool of volunteer resources that are contributed by end-users so as to deploy cloud capabilities.

For Nebula architecture they describe some use cases:

- Nebula architecture can act as a testbed for preliminary cloud services.
- It can work in a way to implement distributed-data-absolute applications near to the location where the data is kept.
- It allows the services that are deployed to be accessed by others as a "public service".

The three requirements that are needed to carry out this work are as follows:

- The first requirement is about offering service-oriented performance differentiation. To fulfill this requirement resource discovery techniques are used on a wide scale to choose resources according to their capability of fulfilling the required performance of each service and to split up the tasks so as to fit into the capacity of the resources.

- Next requirement is to handle data coupling and computation. This requirement is accomplished by making use of P2P data location techniques and methods that are required to calculate network distance between the node which is used to deploy the cloud services and where the data is located.

- Small scale failures require robustness. So, the requirement is to offer robustness. This can be attained by implementing replication and aggressive check pointing techniques that are based on the information about reliability of each node.

2.2 Volunteer Cloud Computing (VCC) Architecture:

Abdulelah, Robert and Gary proposed VCC architecture [3] which makes it easier for volunteer cloud to be an applicable cloud solution.

This VCC architecture is composed of three layers:

- Service layer
- Virtual layer
- Physical layer

Each layer is responsible to carry out its own specific task. The middleware layer and physical layer are further divided into sub- components.

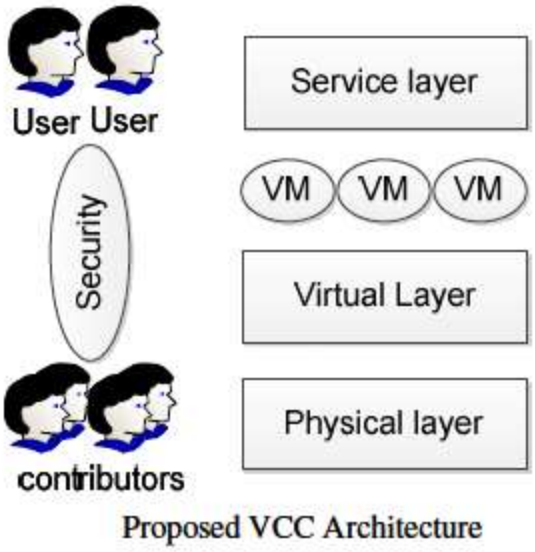


Fig.2.2. VCC architecture

- **Service layer**

  The service layer is responsible for delivering the services to customers through an interface formed on SOA approach. The way it deliver its services to customers is same as that of commercial clouds. Volunteer cloud computing clients are of two types: the consumer and the contributors. And a client can be both at the same time. VCC's contributors are responsible for contributing their resources and consumers can easily make use of these resources.

- **Virtual layer**

  In this layer resources are offered to the service layer. This layer is further divided into two parts: task management and QoS management.

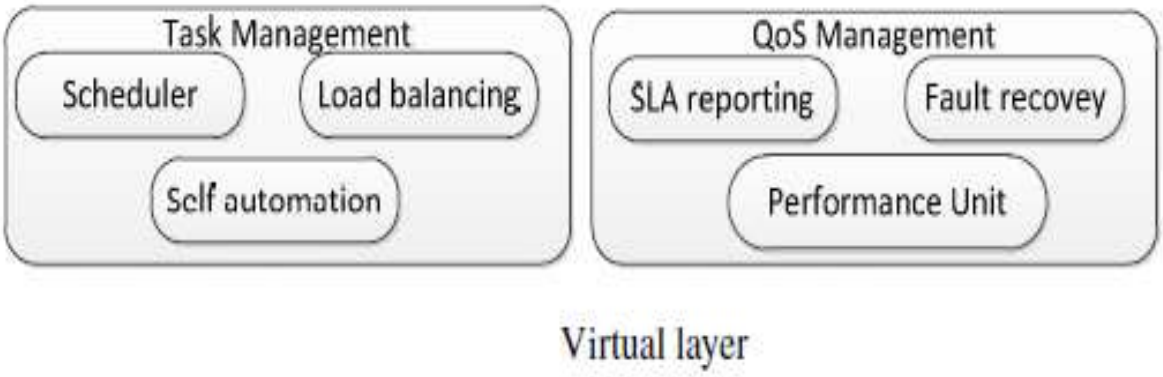


Fig.2.3. Virtual layer

Task management deals with tasks obtained from the service layer. The task scheduler is responsible for scheduling the tasks to appropriate resources that are provided in the physical layer. In order to lessen the time taken to execute a task, **load balancing** make sure that if the load is dispersed properly. In volunteer cloud computing **self automation** is used to offer rapid elasticity so that users can access services according to their requirements.

To maintain minimal quality level and performance, **QoS management** is there. **Performance monitor** ensures an acceptable performance level and thus it monitors the performance of every individual task. All this information is stated in the **SLA reporting**. There is another component of QoS management and that is **fault recovery,** it's the most essential component which is used to enhance the overall performance of volunteer cloud computing.

- **Physical Layer**

  Physical layer is further divided into three parts: allocator, aggregator and monitor.

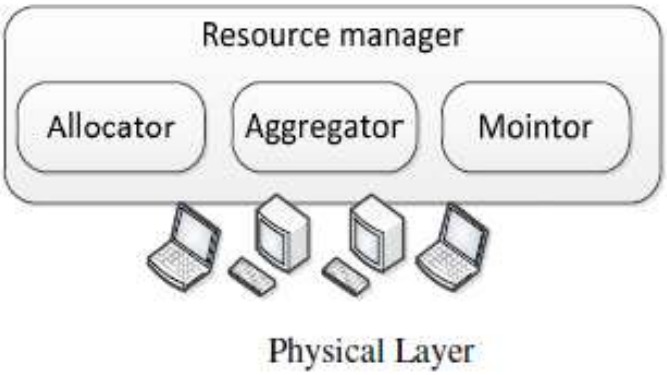


Fig.2.4. Physical layer

The combination of these three is known as **resource manager. Aggregator** component of resource manager is used to combine volunteer resources contributed by various individuals.

Depending upon various criteria resources are classified using an aggregation mechanism so as to achieve quality of service. The **allocator** component is responsible for allocating the tasks obtained from task management to suitable available resources.

The **monitor** component monitors the reliability of these allocated resources from time to time.

2.3 CLOUD@home

2.3.1 CLOUD@home idea:

The main thought which generates Cloud@Home [8][9] Idea is to build voluntary contributing clouds by refabricating/reusing domestic computing resources. The contributing cloud should not only be able to interoperate among them but also with the commercial Clouds infrastructure which is already in place. Cloud@Home concept gives huge advantage to users to actively provide their Resources/Services for power Cloud Computing & also passively submit request to exploit that power for them also.

In Cloud@Home concept, the 2 viewpoints of Cloud Computing Domain coexist i.e.

Commercial/Business Viewpoint:

This viewpoint is more inclined towards end user requests & in Cloud@Home is further extended to synergistic 2-way Cloud where users can buy and/or sell their Services/Resources.

Volunteer/Scientific Viewpoint:

This viewpoint is more about Grid Philosophy which suggests extending & enhancing few but large requests to open virtual organization. In both the viewpoint, we've to introduce the concept of quality often known as QoS (Quality of Service) in terms of negotiated policies, constraints & quality requirements.

Cloud@Home concept is the mature & generalized picture of already existing philosophy @Home: this concept builds up a context where there are no compatibility issues & the users can share their resources/services out of their own will. This helps us overcoming following two challenges of Grid & Volunteer computing:

Hardware Challenges:

Processor bits, endianness, architecture, network etc.

Software Challenges:

Operating system, libraries, compilers, applications, middleware

The Hardware & Software challenges were considered to be the barriers of grid & Volunteer computing which are eventually overcome in complete Cloud@Home concept.

Furthermore, the term Resources has to be widely defined in Cloud@Home concept so that it includes more generic sense of Cloud services. This implies that Cloud@Home not only help users to share the physical resources (as per the @home philosophy & also Grid environment) but also extend sharing to any kind of services.

The Cloud@Home concept is highly flexible & provides the ease of extension. These attributes of Cloud@Home helps to easily arrange, manage & ensure availability (Free/or by charge) of the huge computing resources (greater in magnitude when we compare it with Clouds in @Home concept or Grid Environment) to all the users who owns computer.

However the Cloud@Home concept can also proves to be the enhancement of Cloud Computing's Grid Utility version. According to this enhancement, the user's host do not play a role of passive interfaces to cloud services, but they are actively included in computing process.

Moreover according to this concept, even the single nodes & services are extensively used by Cloud@Home middleware to build up a very strong Own-Private Cloud Infrastructure that can promote the interoperability to the next level & thus help clouds to effectively interact among them.

The overall infrastructure/ or say architecture should able to accommodate the high level mechanism of its nodes/resources to provide permission to move data & reallocate data, tasks & job. Therefore it becomes a necessity to implement a very light weight middleware which is designed to handle effective & optimized migrations.

This throws light of yet again powerful attribute of Cloud@Home: a light middleware which has the ability to include the limited resource's device into the cloud as Consumer host which in turn can access the cloud through "thin client". In addition to that a light middleware also has the ability to make that resource (which is a device of limited resource included in the cloud) act as contributing host for some applications.

Also adding further, the light middleware of Cloud@Home does not interfere with code writing however in the other paradigms like Grid Environment or Volunteer Computing this was a major challenge of middleware.

One more purpose of action of Cloud@Home is *Security*. Let's first discuss the earlier philosophies:

Volunteer Computing: It has some open items or say holes as far as its security aspect is concerned which make it highly vulnerable.

Grid Computing: On the other hand the Grid Computing's security mechanism is highly complex leaving very little space to experiment.

The virtualization included in Clouds exercises the isolation of services but weakens when it comes to the protection of data from local access.

And therefore it becomes very essential for Cloud@Home to come up with very strong mechanism to tackle the protection of data from local access. As we understood by now that Cloud@Home is composed of the Resources which are conceivably more large & huge than commercial or proprietary Clouds therefore its reliability is easily comparable to Grid or the Volunteer Computing & greater than other Clouds.

Lastly the most highlighted attribute of Cloud@Home is *Interoperability*. This attribute will help the cloud to interoperate among them. Cloud interoperability will provide an environment where Cloud Infrastructure will come out be a transparent platform on worldwide level, a platform which will not restrict the applications to specific Enterprise Clouds & Cloud service provider. We must strive to develop a new benchmarks & simple & effective interface to leverage on the enhanced attributes of Portability & Flexibility of virtualized applications.

Up till now, a very eloquent discussion has been done about the Open standards for cloud computing. In this domain "open standard manifesto" speaks about the minimal set of principles or standards which will serve as elementary setup for introductory agreement as the Cloud computing community tries to lay down the foundation for standards & principles for this new philosophy.

However the Grid environment is facing enough heat when it comes to interoperability, so many trails had been given to resolve or address the issue, but still we are far from a viable solution.

On the other hand Interoperability is easier to implement & exercise in Cloud Computing environment. This is due to the fact that Cloud@Home virtualization does away with the major architectural, physical hardware & software barriers.

However there may arise an issue of compatibility among different virtual machine monitors popularly known as VMs. This must be aptly faced, as the Open Virtualization Format (OVF) 1 group is giving it a try.

The extremely important key points that need to be taken care of while implementing Cloud@Home are:

- **Resources & Services Management:**

There should be a very strong mechanism in place to manage the Resources & services provided by the Cloud. Resource and Services form the basic step stone for a cloud computing architecture & therefore a need arises of a very effective management of these building blocks.

- **Frontend:**

High level User Interface has to be facilitated to the users in order to depict the true potential of the computing system. This very key point shoulders the responsibility of interacting with the users & thus accepting the requests & submitting them. The frontend needs not to include a very high level dynamics which make end users uncomfortable & thus a UI less friendly, but a more interactive & friendly UI which will help service the Users requirement properly.

- **Security:**

Apt measure should be taken to ensure:

User Authentication or Validation: It makes sure that an access is provided to the intended users only & not to the users who are not authorized to log in.

Data & Resource Safekeeping: It makes sure that none of the crucial data/resource is lost while transacting or even when it is sitting idle on the cloud.

Data Confidentiality:

Data confidentiality speaks of the security of data from a different angle. For example a user A is authorized to login to the Cloud but he/she does not have permission to access a set of data which belongs to User B. For User B it is a data which belongs to him only & therefore it is confidential to him.

High level integrity:

Keeping the data safe is one aspect of security but keeping the data safe as well as in its true form is another aspect. It is to say that keeping data safe & correct is also one goal of security in Cloud@Home

- **Clouds Interoperability:**

The Clouds should possess the property of interoperability among themselves which in larger context means that they should be able to exchange and use information from each other. This attribute is extensively exercised in Cloud@Home & thus it helps to identify, allocate & ensure proper availability of required Resources/Services.

- **Standardizing Business Model:**

It's a necessary mandate to provide a Quality of Service (QoS) & Service Level Agreement (SLA) for both commercially viable Clouds & openly volunteered Clouds.

The SLA includes all the negotiated policies, constraints & benchmarks to test the quality of the service provided to the users. In case the service output diverted from the designed path in terms of satisfaction parameter laid down in SLA, there should also be a provision of workflows to set things in right perspective again.

2.3.2 Architecture:

Logical Abstract Model

This architecture differentiates between two types of Users depending upon the role they play in Cloud:

End Users: are those users who directly interact with the Cloud for submitting the requests.

Contributing Users: are those who contribute with resources & services for making the Cloud stand strong in terms of infrastructure which included both aspects Development as well as sustenance.

According to this theory, a Cloud is strengthen up with various *contributing hosts* provided by *contributing users* to the *end users* who interact & submit their requests with the help of *consumer hosts* to the Cloud.

In order to access the Cloud both users are required to validate themselves & therefore we have 2 user authentication: *End User Authentication* & *Contributing User Authentication.* Major enhancement which we can see in Cloud@Home is that a User/Host can act asna contributing

user/host as well as an end user/ consumer host at the same time, establishing a symbiotic mutual interaction with cloud.

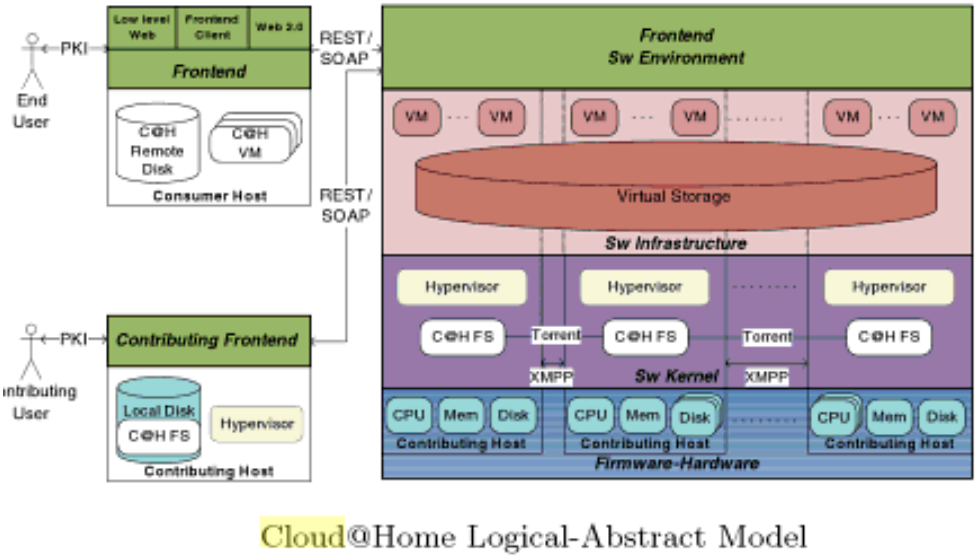


Fig.2.5. Cloud@home logical abstract model

- **Software Environment**

The high level user-infrastructure frontend interface is implemented by the Cloud@Home Software Environment. Its major task is to efficiently manage Resources & Services (for example: enrolling, discovery, allocation, coordination, monitoring, scheduling, etc). It also facilitates the tools, library & Application Program Interface (API) for mapping the user requirements into physical resource's requirement.

In addition to that, when we talk about commercial Clouds, it should also be able to come up with a negotiated agreement about the QoS policy to be applied (SLA). And this will help us monitoring its execution & in case of diverted results or unsatisfactory results, adopting the computing workflows to such QoS requirements.

If one Cloud is not able act on the request, the UI provides the process to raise a request for more Resources & Services from other Clouds. In more elaborated words, we can say Cloud@Home's UI/frontend executes interoperability among clouds & also keeps a check on availability & reliability of Service.

- **Software Infrastructure:**

Two very fundamental services provided by the software Infrastructure to the Software Environment as well as to end user are *Execution & Storage Services*.

The *Execution Service* authorizes to create & manage the virtual machines. One user who is sharing his Resources/Services in a specific Cloud can authorize/allow execution & management of the virtual machine by any other users of the Cloud locally at his node. But this should be in line with the policies & constraints laid down & monitored through the Software Environment.

The *Storage Service* implements Storage System sprawling across various hardware resources meant for storage. These storage hardware resources build up the crust of the Cloud & are independent since they store the replicated data in order to satisfy the policies & requirements of QoS.

- **Software Kernel:**

  In order to run Execution & Storage Services, we need software infrastructure, Process flows/Workflows (also known as mechanism) & tools for local management of the Cloud's Physical Resources.

  Cloud@Home comes to an agreement with the Users who want to join the Cloud about their contributions. This whole process involves Software Kernel to provide the tools for reserving the Execution and/or Storage Services/Resources for the Cloud. It also monitors these Execution & Storage Resources & services so that it can ensure that all the QoS constraints, requirements & policies are followed strictly.

- **Firmware/Hardware:**

  This layer of Cloud@Home includes a "cloud" of generic contributing nodes and/or geographically distributed devices which are sprawling across internet. This layer gives much needed support to the upper layers in terms of Physical hardware resources which in turn ensure effective as well as efficient implementation of Execution/Storage Services.

Application Scenario

There are several Cloud@Home scenarios that exist:

- **Extension of @Home Support:**

Cloud@Home possesses various properties like being Open, interoperable which in turn can benefit the support process of scientific research since it overcomes portability & compatibility issues exhibited by @Home concept.

This phenomenon of Open & Interoperable concept also helps public administration & open communities like social networking, peer-to-peer, & gaming community. To add more, the Volunteer computing has many limitations & issues when it comes to address the QoS requirements. Since these QoS requirements are of utter importance to many scientific projects, using volunteer computing becomes quite challenging.

The Cloud@Home concept overcome this difficulty by implementing the negotiated policies & constraints stated in QoS, this process includes rating the resources & providing services according to specification of QoS (using reward & credit system).

- **Expanding Enterprises:**

Implementing Cloud@Home concept the business environment will give huge benefits, specifically in case of Small & Medium enterprises. There are companies have their own Data Center, Local & off-the shelf resources.

This process is very much sprawled across many companies & therefore these companies go for stand-alone computing resources aligned to a very specific task (like designing, monitoring, automation etc).

These resources are utilized only during the office hours, so when they are idle we can combine them with other resources by using internet connection to realize the concept of Cloud@Home data center for sharing the services (file servers, web servers, archives, database & so on) & resources without any compatibility constraints or issues.

The interoperability property of the Cloud@Home concept states that a user can buy the computing resources as well as services from commercial/local Cloud provider and they can also sell their computing resources to the same/other cloud providers. Following the same concept the companies also can sell their computing resources for Cloud@Home concept.

This whole concept of sharing resources/services & interoperability will help businesses optimize their cost according to QoS & SLA & improve performance & reliability. Say for example we can make data center manage the average caseload, also during Peak hours, the company always has the option of using computing resources from other Cloud Providers.

In addition to that the companies can also securely manage their business processes online, submit requests asking for computing resources to Cloud provider & also provide their own computing resources to Cloud Provider for others to use.

Also by making them customizable we can also improve e-marketing & trading services. Since Cloud@Home possess the property of interoperability, private companies can act as sub-contractors who can buy & resale the computing resources.

Addressing Device

- **Heterogeneity and Mobility**
  In this approach of Cloud computing where the user has the power of owning & managing both software services & computing resources, the programmer gets the ease while addressing the issue of device heterogeneity. This ease gives mobile application designers' an opportunity to look at the more broad aspect: interaction with cloud instead of looking into present/prospective scenario of small device application.

There are more important issues which needs attention & can be very profitable if dealt properly like Service Discovery, brokering & reliability.

Researchers have proposed a new concept of *mobile dynamic virtual organization*; this concept has come up as a consequence of mobile users accessing the service oriented grid infrastructure.

This new concepts suggests to use the Cloud Computing concept in the world of wireless edge of the internet.

To support Mobile computing, mobile service cloud will enable dynamic instantiation, composition, configuration & reconfiguration of the services on an overlay network.

The Cloud computing can be considered as initial step towards a new computing paradigm "Mobile Computing" that involve active participation of the following:

- **Resources & Services:**

It includes any type of resources or services (desktops, laptops, mobile devices, clusters, data center, network infrastructure, storage, sensor networks, or the internet)

- **Distributed Paradigms:**

It includes all the popular & upcoming distributed concepts like peer-to-peer, cloud computing, grid architecture, volunteer computing)

- **Stakeholders:**

It includes all the users, whether they are end users or contributing users. They can be business or commercial users, academic users, public users or private users.

A lot has already been discussed, but still even after these approaches some challenges are still left that should be taken into consideration so as to achieve an interoperable environment. All these related work defines the utilization of unused and idle non-dedicated resources to build a cloud.

These projects deal with an idea of building a volunteer cloud with the computing resources contributed by various individuals across the world and also defines the architecture of volunteer cloud. But in our thesis, concentration is on how the communication will be carried out between the communicating parties.

3. Proposed Work

This thesis contributes toward establishing an enhanced interoperable environment where volunteer cloud entities are allowed to communicate with each other and also with other commercial clouds and clients via XMPP protocol (Extensible messaging and presence protocol). XMPP is considered to be an ideal middleware protocol so it is implemented over middleware layer to form XMPP based messaging oriented middleware which is the basis for communication between these entities and can help towards a solution for the challenges mentioned in section 1.4.

3.1 XMPP

XML (Extensible Markup Language) based open instant messaging protocol JABBER/XMPP was launched in January 1999 and its server implementation was also released which was known as Jabberd. XMPP, an enhanced version of JABBER protocols was brought up by IETF in the year 2004.

- XMPP offers a wide range of services and has emerged as scalable, more secure, implemented on a large scale, internet-scale technology, stateful, support a large number of end users, offer two way communication, enhance interoperability, extensible, and thousands of interoperable collection of code.

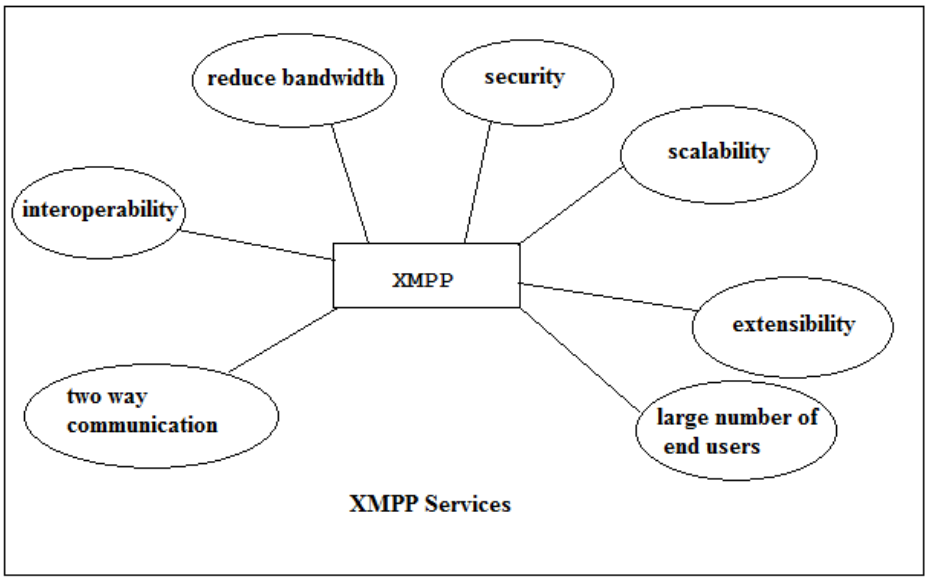
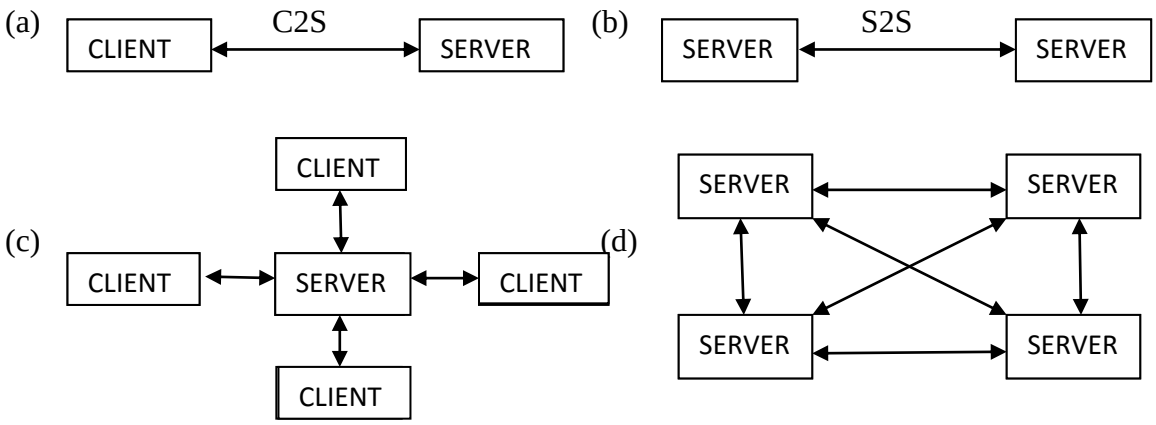


Fig.3.1. XMPP services

- **Security**: XMPP comes up with more improved features of TLS (Transport Layer Security) which deals with channel encryption and SASL (Simple Authentication and Security Layer) for authentication purpose. There is no issue of a single point of failure as it is based on a decentralized network. These are the features that make XMPP more secure.
- **Decentralized network**: XMPP technologies are implemented in a distributed architecture with millions of servers. Individual or any organization can have their particular XMPP server. Domain Name System (DNS) can be used so as to connect XMPP server to the remaining network.
- **Extensibility**: XMPP is responsible for transferring the XML messages from place to place. XMPP has come up as a strong provider for a large number of applications like instant messaging, data transfer, voice over IP, machine to machine communication, geolocation, gaming, system control, file sharing, group chat, middleware, real time communication and many more.
- **Scalability**: XMPP is designed to be scaled. XMPP overcomes the issue of long polling related with HTTP-based methods. HTTP-based protocols offers one way communication that makes the services non real time, won't extend. There is another problem of long polling with HTTP based protocols, it means that server has to wait until it receives an update and as soon as it receives an update, the server sends the response and then only client can send further request. In this regard, XMPP offers faster and easy two way communication and also eliminates long polling. Only one connection is established between the two communicating entities and therefore it reduces the bandwidth needed for communication.



XMPP provides a bidirectional communication between client and server; server and server; between multiple servers; one server and multiple client. In order to acquire high performance and scalability XMPP provides support for built-in publish subscribe functionality.

For clouds to interact with each other, a common format for messaging is used to allow the resources to be transmitted between the entities and manifest how their services can be utilized.

On different cloud network, if resources are not implemented by XMPP and if the communication is to be established, then there is XMPP gateway that can convert XMPP to a foreign protocol. Any user who wants to communicate with the user on the other network, that user will first have to register with one of the gateways for the authentication purpose and then only they can start the communication.

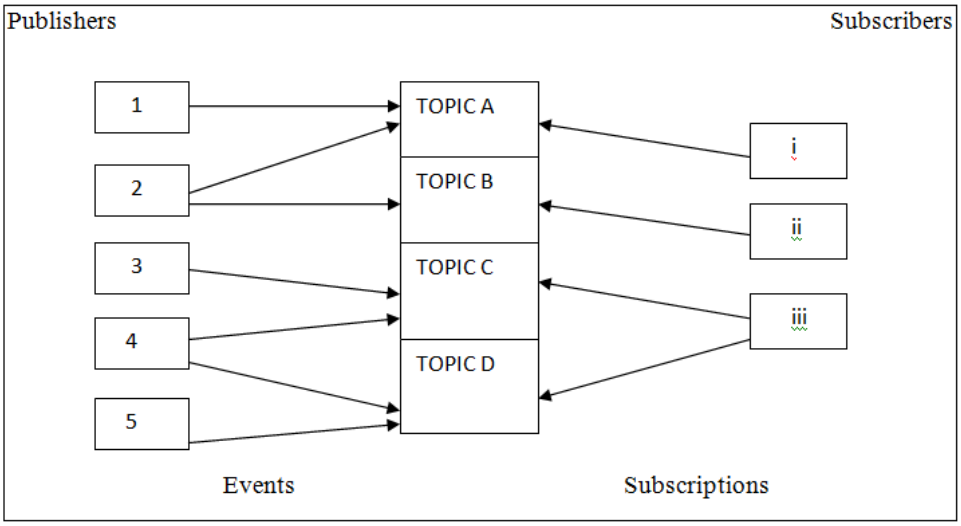


Fig.3.2. Publish subscriber

- The main asset of XMPP is its vigorous and amicable community. And this community is a collection of developers, end users, service providers etc.

XMPP is best suited for:

- Distributed and complex problems.
- Where there is low latency interaction.
- Dealing with complicated cloud services.

3.1.1 XMPP Architecture:

XMPP architecture is comprised of distributed, decentralized network of clients, servers, JABBER IDs, presence information of networks, structured data based on XML so as to provide asynchronous real time end to end communication. XMPP support continuous and stateful connection to transfer structured data from end to the other via XML streams.

- XMPP provides a separation of responsibilities between client and server, and allow clients to concentrate on user experience and severs to concentrate over reliability and performance, security related concerns like authentication, encryption.

- There is no such case of single point of failure as anyone can be a part of the network by having their own XMPP server. The association of XMPP has always followed the strategy of keeping the client simple and moving all the complexities to the servers.

- XMPP is an event-driven protocol which is held responsible for carrying out asynchronous communication between the entities. Unlike HTTP, XMPP offers easy two way communication and eliminates long polling as clients does not have to wait for the server to respond immediately in order to carry out further processing, client can freely send multiple requests to other entities and can get the response after the completion of these requests.
- In XMPP, XML is used to carry out transmission of structured data in a combination of streams. Suppose data is to be exchanged between client and server, so one stream is used for the communication from client to server and another one is used for server to client communication.
- There exists a direct communication between XMPP servers. For addressing purpose JABBER IDENTIFIERS (IDs) are used, JIDs indicates each entity that is the part of an XMPP network. XMPP brings up a format for the purpose of data transmission between entities that are willing to communicate.

In XMPP, XML STANZA is considered to be a basic unit of structured information which is being transmitted between communicating entities upon an XML stream. These XML stanzas are enclosed within an XML stream envelope. The core XMPP is composed of 3 kinds of stanzas: <message>, <presence>, <iq>. And clients and servers are responsible for managing each of these kinds of stanzas differently.

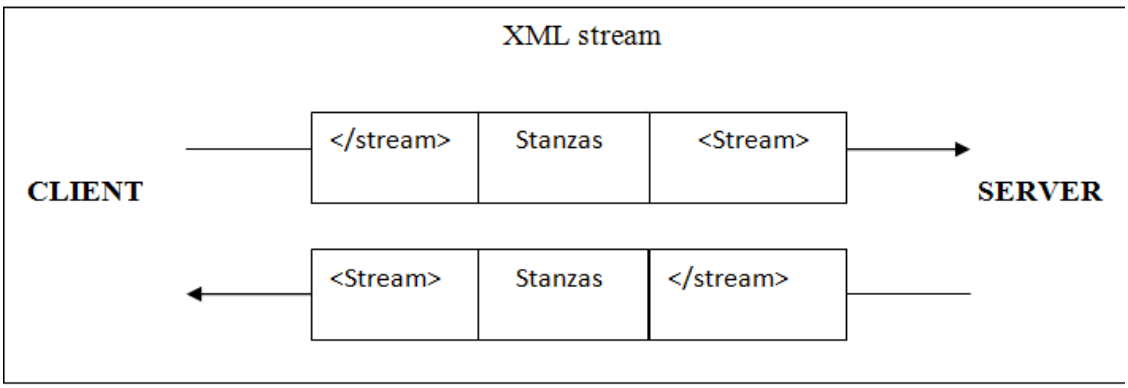


Fig.3.3. XML stanza

Message

<message/> stanza of XMPP is a kind of stanza which is strongly based upon “fire and forget” method for speeding up the transmission of information between communicating entities.

Presence

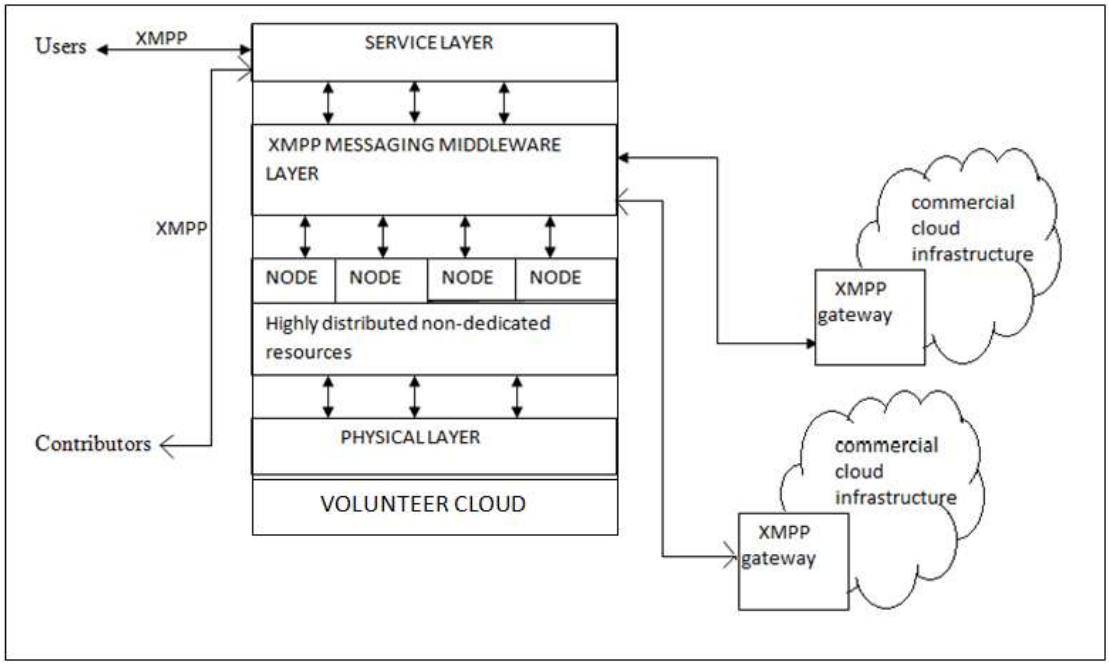
<presence/> stanza as the name suggests is used for managing and reporting the presence information of any entity. It allows an entity to broadcast its presence information or availability to other entities through communication. And moreover presence stanza is based on publish-subscribe mechanism for advertising network availability.

IQ

Like GET and POST methods of HTTP, XMPP Info/query stanza is based on request response method for communication. Id attribute which is managed by both the communicating parties are used to locate requests and responses. IQ stanza contains the following values: get or set (for requesting entity), result or error (for responding entity).

3.2 Working Process

A working procedure is presented over here to achieve better interoperable environment where volunteer cloud entities are free to communicate with each other and with other commercial clouds and clients also via XMPP protocol. In this regard, two different scenarios will come into play: first one deal with the communication between volunteer cloud entities using XMPP and second scenario deals with the communication between clients and volunteer cloud.



Fi.3.4. XMPP based messaging oriented middleware

3.2.1 First Scenario

XMPP protocol is implemented over middleware layer so as to form XMPP messaging oriented middleware layer that allow volunteer cloud entities or components to interact with each other by exchanging xml stanzas between components or entities. XMPP provides scalability which makes it easier for an infrastructure to extend by adding multiple resources, services and nodes to the network. It also overcomes the problem of single point of failure, as our volunteer cloud infrastructure can have multiple XMPP servers.

- Volunteer cloud infrastructure contains various components or entities that further incorporate multiple sub-components, each of them is responsible for performing a particular task. These entities perform task scheduling, performance monitor, resource management, data management, network management etc.
- All the communicating entities of volunteer cloud can make use of XMPP </presence> stanza which is based on publish-subscribe method as it is responsible for managing and reporting the presence information. It allows an entity to broadcast its network availability to other entities through communication. This can be made possible through XMPP based messaging oriented middleware.
- XMPP presence server receives presence information from XMPP clients. Presence state, addressing information and protocols are the present data distributed in XML streams. So to manage the availability of volunteer cloud entities, presence information is used and incoming and outgoing requests are handled by XMPP servers.
- Firstly, all the entities must connect with the XMPP server so as to carry out the communication.
- If a request is sent from one entity to the other, the receiving entity will check whether the request can be executed or not. It the receiving entity is capable of executing that request, the task is performed and the response is sent back to the requesting entity. Otherwise the requesting entity will send an error message to the requesting entity.
- Like this various components and sub- components of volunteer cloud infrastructure can communicate with each other via XMPP.
- Earlier XMPP was considered to be a perfect protocol for instant messaging between the users but with the emerging era of computing paradigms, there is a vital need for scalability, availability, security and transmitting computing resources, various tasks, virtual machines, larger services, storage from one entity to the other. In this regard, XMPP can be implemented over different kinds of levels so as to offer diverse services as well as support for authentication and encryption that can serve this purpose completely.

- **Flow chart for internal communication using XMPP**

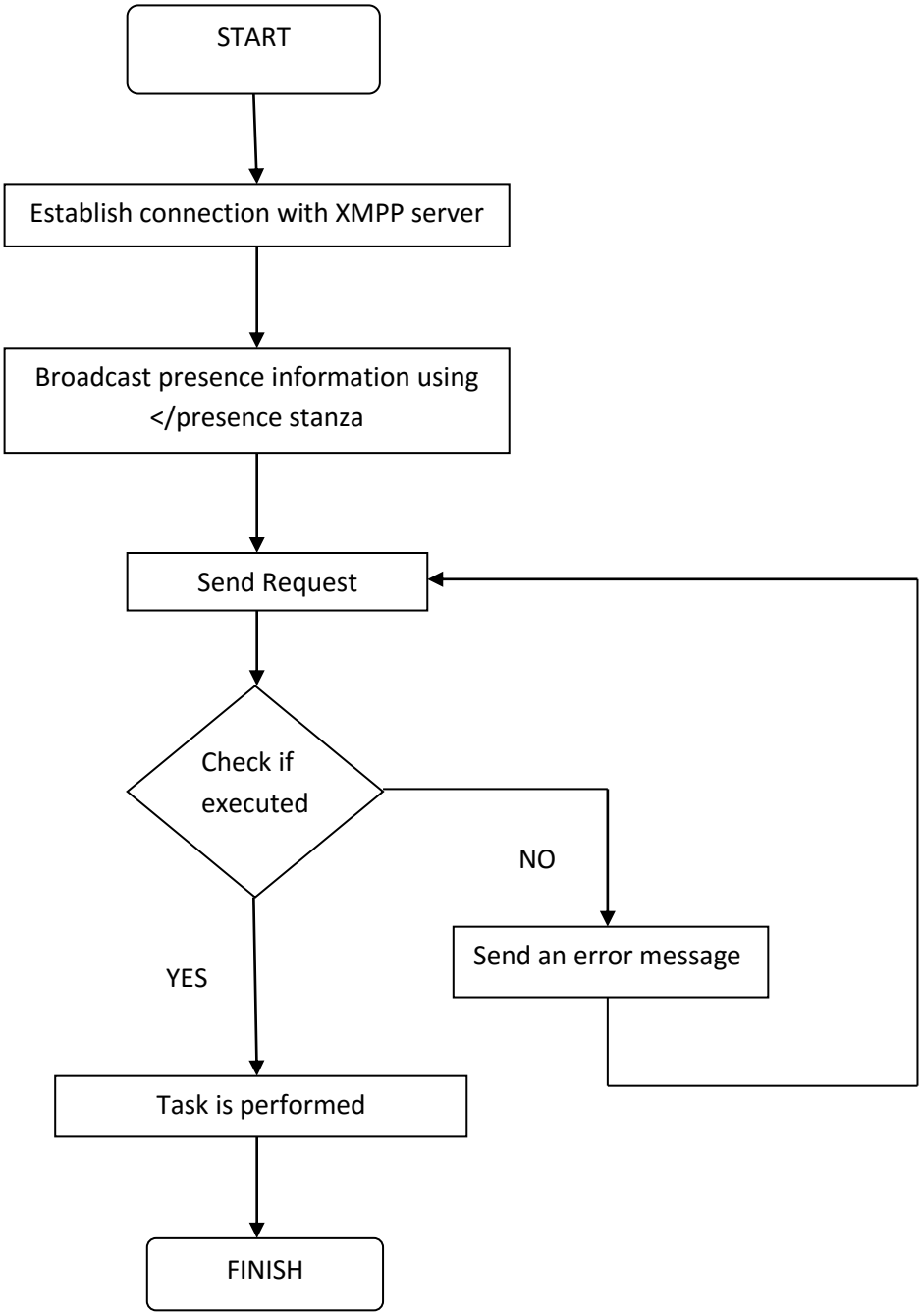


Fig.3.5. Flow chart for internal communication using XMPP

3.2.2 Second Scenario

- In this scenario, a communication process is defined between volunteer cloud and clients by using XMPP. It describes how the clients can interact with the volunteer cloud so as to fulfill their requirements and needs.

- Effective communication is not just related to exchanging messages but its more about having a reliable, scalable, secure communication between the communicating parties. And it also depends upon two way communication because when two parties try to interact with each other, there should exist a proper two way communication so that both the communicating entities should get involved in order to carry out a particular task. And XMPP has turned out to be a very strong technology that makes communication effective.

- First scenario deals with how the volunteer cloud entities can interact with each other, it shows the internal communication within the volunteer cloud. And in this scenario, concentration is on the external communication that shows how a client can interact with the volunteer cloud.

- A scenario is defined with which clients can effectively consume and contribute their resources to the volunteer cloud.

- So in order to accomplish this scenario, a Use Case Diagram is described first which depicts all the high level functionality of our entities and next a Sequence Diagram it is also termed as interaction diagram is presented and it shows what will be the sequence of the information that will flow between the objects. Here Enterprise Architect tool is used for designing these UML diagrams. And also display how the entities will be going to interact. Finally the implementation is defined.

USE CASE DIAGRAM

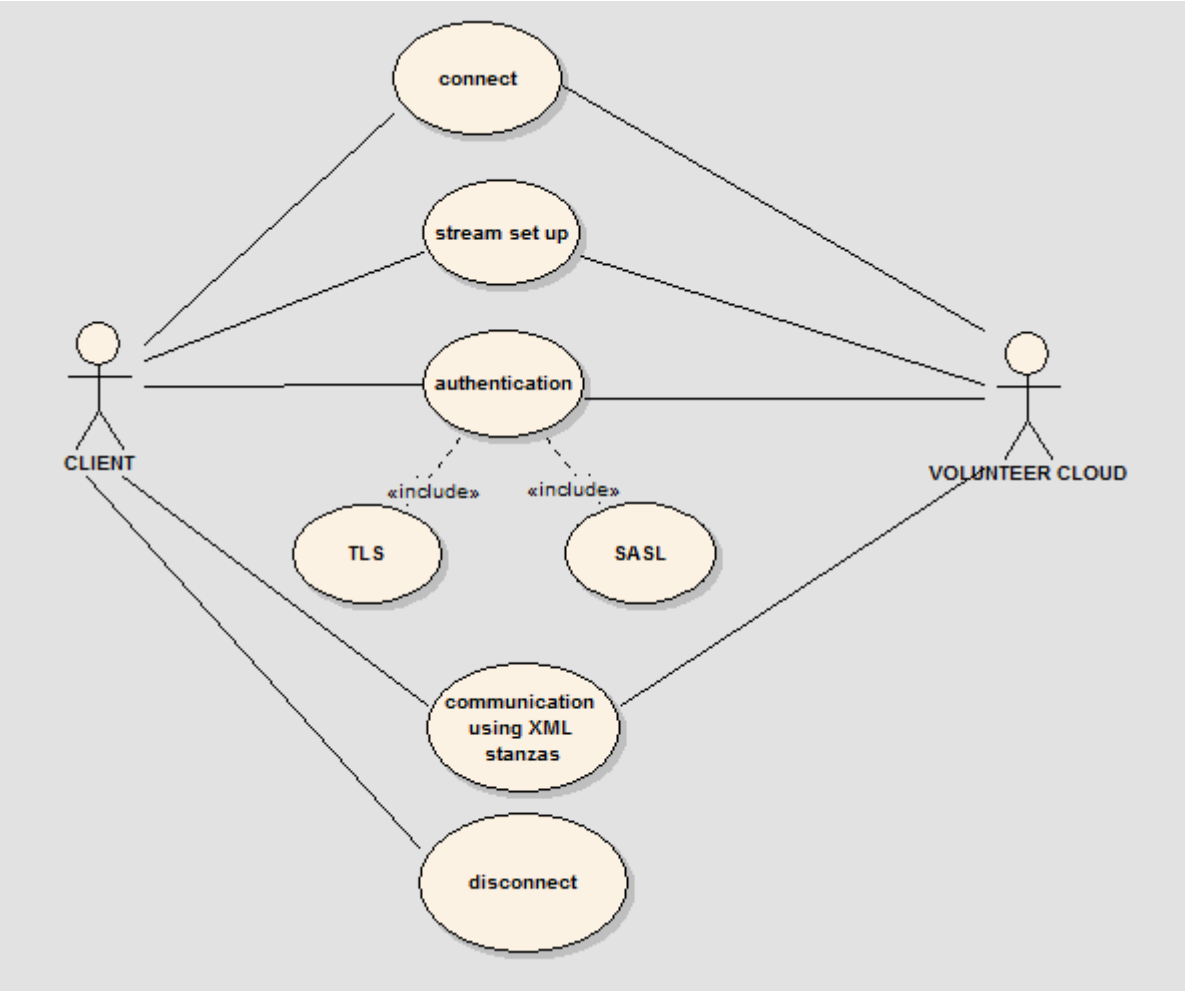


Fig.3.6. Use case Diagram

- The above Use Case diagram, display the roles or the actions that are carried out by our actors in order to establish a communication. It shows both the client and the volunteer cloud following the same set of actions for establishing the communication using XMPP protocol. A client can initiate the interaction by connecting with the volunteer cloud then authentication and exchanging messages.
- A client can be any of the following: commercial cloud, users, contributors, that are willing to interact with the volunteer cloud for two reasons:

    - Either to get the access of the volunteer computing resources for the purpose of fulfilling their needs and requirements.
    - Or to contribute their idle resources that are kept unused for a long period of time for the betterment of computing paradigms.
- Both the client and the volunteer cloud can establish an interaction by connecting their respective XMPP servers.

The actual description of Use Case:

Actors: Client, Volunteer Cloud

Description: The above use case initiates when the connection is made between the client and the volunteer cloud respective XMPP servers. Then an XML stream is established between the two communicating entities.

For further communication, both the entities must first authenticate themselves. Here the authentication use case is further split into two more use cases by using “include” relationship. So both the client and the volunteer cloud can carry out authentication action by using TLS (Transport Layer Security) and SASL (Simple Authentication and Security Layer) mechanisms.

Now, after the authentication operation is done, the communicating parties are allowed to communicate freely by exchanging a number of XML stanzas. XML stanzas are of three kinds:

- Message
- Presence
- IQ

So our entities are free to communicate with each other via passing any kind of XML stanzas. And at last when they are done with the exchange of information and when their work is done, the initiating entity can terminate the connection.

SEQUENCE DIAGRAM

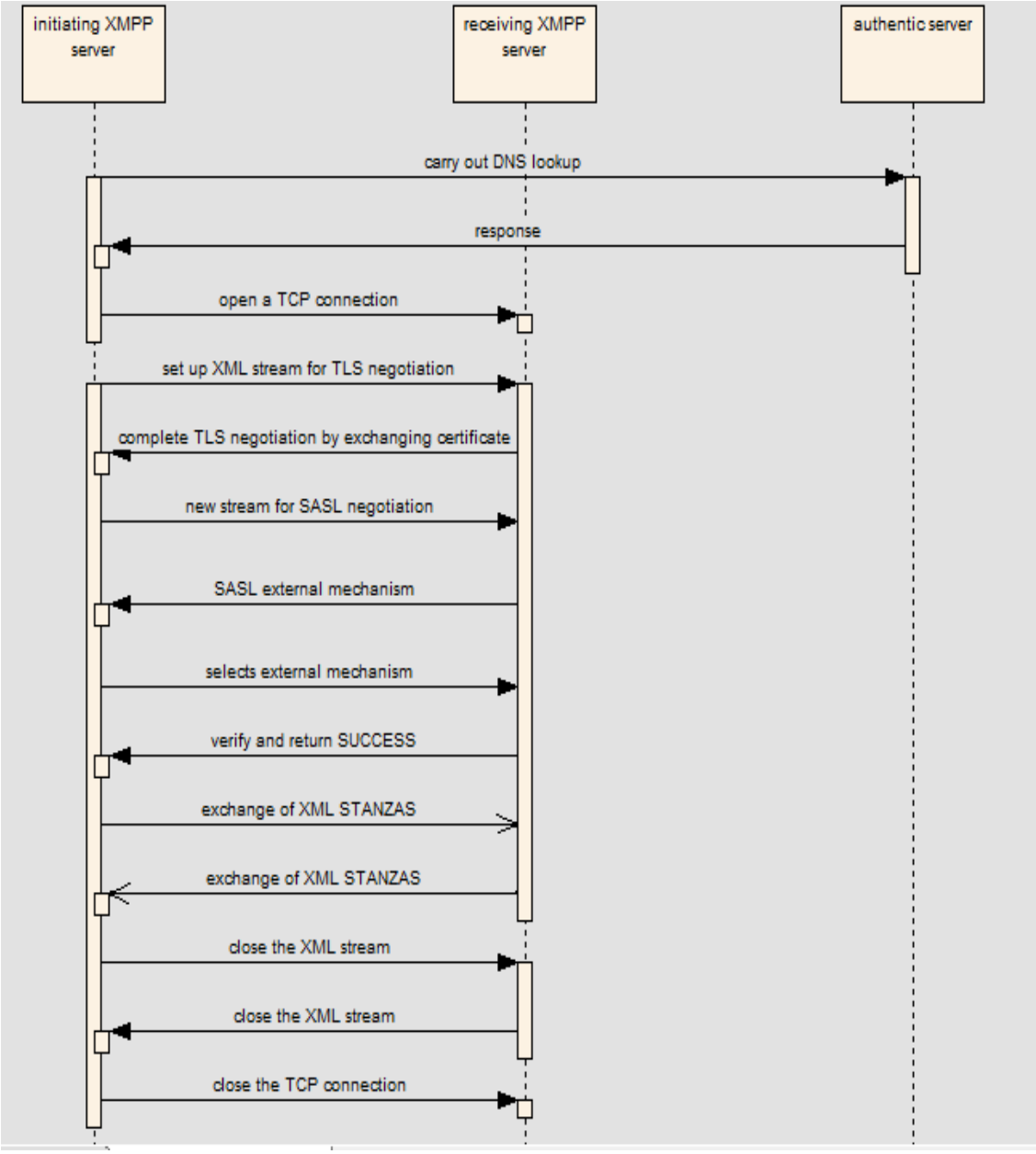


Fig.3.7. Sequence Diagram

The above Sequence Diagram, shows what will be the sequence of the actions that will flow between the objects. It also display how the interaction works between the XMPP servers. The

initiating and the receiving server negotiate with each other so as to carry out the communication effectively.

One object will be our initiating XMPP server, it is considered to be client's server which is responsible for initiating the communication, second one is the receiving XMPP server, it is considered to be volunteer cloud server with which the interaction is being setup, and the last one is the authentic server which is equivalent to the Domain Name System.

Sequence Diagram description:

Objects: initiating XMPP server, receiving XMPP server, authentic server.

Description:

- It will start with the initiating XMPP server carrying out DNS lookup with the authentic XMPP server.
- After receiving the response from the authentic server, initiating server establish a TCP connection with the server (receiving XMPP server) with which it wants to communicate.
- When the TCP connection is established its time to setup XML stream for TLS negotiation between the two servers.
- The two communicating server exchange certificates with each other in order to complete the TLS negotiation.
- Now a new stream is setup for the SASL negotiation then receiving XMPP server send a request for SASL External Mechanism to the initiating server, now the initiating server will select an External Mechanism and send it back to the receiving server. After the successful completion of the authentication process, the receiving server will send a SUCCESS response to the initiating server.
- After the authentication process both the servers are free to communicate with each other via exchanging XML stanzas.
- When they are done with the exchanging information and when their work is done, the initiating XMPP server can close the XML stream and TCP connection.

3.2.3 Implementation:

- Suppose ab.domainA.net is the initiating XMPP server or we can refer to it as server1 and domainB.net is the receiving XMPP server or we can call it as server2. Initially server1 has to perform DNS lookup with the authentic server and after clarifying a Service Record of _xmpp-server._tcp.domainB.net and then a request is sent from server1 to server2 in order to establish a TCP connection.

- Now server1 set up an XML stream with server2 in order to carry out TLS negotiation.

```
S1: <stream:stream
      from='ab.domainA.net'
      to= 'domainB.net'
      version='1.0'
      xmlns='jabber:server'
      xmlns:stream='http://etherx.jabber.org/streams'>
```

- Features of stream along with a stream header are sent as a response from server2 to server1.

```
S2: <stream:stream
      from='domainB.net'
      id='hTiXkW+ih9k2SqdGkk/AZi0OJ/Q='
      to='ab.domainA.net'
      version='1.0'
      xmlns='jabber:server'
      xmlns:stream='http://etherx.jabber.org/streams'>
```

```
S2: <stream:features>
      <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'>
        <required/>
      </starttls>
    </stream:features>
```

- In order to carry out TLS negotiation, a STARTTLS command is transmitted from server1 to server2.

```
S1: <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
```

- Now both the servers exchange certificates so as to complete TLS negotiation. After the successful completion of TLS negotiation, a new stream is set up by server1 for SASL authentication.

```
S1: <stream:stream
      from='ab.domainA.net'
      to= 'domainB.net'
      version='1.0'
      xmlns='jabber:server'
      xmlns:stream='http://etherx.jabber.org/streams'>
```

- Features of stream (SASL External Mechanism) along with a stream header are sent as a response from server2 to server1.

```
S2: <stream:stream
      from='domainB.net'
      id='RChdjlgj/TIBcbT9Keu31zDihH4='
      to='ab.domainA.net'
      version='1.0'
      xmlns='jabber:server'
      xmlns:stream='http://etherx.jabber.org/streams'>

S2: <stream:features>
      <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
        <mechanism>EXTERNAL</mechanism>
        <required/>
      </mechanisms>
    </stream:features>
```

- An external mechanism is selected by server1, this selected external mechanism is sent as a response to server2 along with an encoded identity.

```
S1: <auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl'
          mechanism='EXTERNAL'/>eG1wcC5leGFtcGxlLmNvbQ</auth>
```

- If the information given in the certificate matches with the encoded identity sent by server1, then server2 returns SUCCESS in response.

```
S2: <success xmlns='urn:ietf:params:xml:ns:xmpp-sasl'/>
```

- After the authentication process both the servers (client's server and volunteer cloud server) are free to communicate with each other via exchanging XML stanzas. The communicating servers now can send any number of XML stanzas with each other.

- Suppose abc@domainA.net of domainA.net wants to establish an interaction with xyz@domainB.net of domainB.net for consuming resources that are present on the volunteer cloud's server which we denote here as domainB.net. Now domainB.net check if the requested resources are available, if available then it grants the resources to the requesting party. It means that these resources are logically present in domainA.net but are physically located on domainB.net.

- If the servers of clients and volunteer cloud don't want to communicate any further, initiating XMPP server (server1) can close the XML stream by using handshake procedure.

```
S1: </stream:stream>
```

- The stream is closed by volunteer cloud server (server2) as well.

```
S2: </stream:stream>
```

- Finally, underlying TCP connecting can be closed by server1.

4. Result and Conclusion

The present study aimed at developing an effective interoperable environment so that volunteer cloud entities can interact with each other and also with other clients via XMPP. It also aimed at providing solutions to the challenges that should be taken into consideration in order to achieve effective communication and also provides a well defined sequence of steps that are required to carry out communication. This study also brings out the magnificent role XMPP play so as to establish effective communication between the communicating entities.

The significant findings include:

- XMPP has proved to be the most suitable protocol and has emerged to offer solutions for the challenges and also for carrying out a better communication between volunteer cloud entities and also with other clients as it offers a wide variety of services that includes scalability, more security, implemented on a large scale, internet-scale technology, stateful, support a large number of end users, offer two way communication, enhance interoperability, extensible, reduce bandwidth and thousands of interoperable collection of code.

Feature Comparison

Feature:	XMPP	HTTP+ REST	HTTP+ SOAP
Security	TLS	SSL	SSL
Reliability	YES	NO	NO
Authentication	SASL	NO	NO
P2P Support	YES	YES	YES
Easy to Integrate with Web	YES	YES	YES
Easy to Integrate with Operators	EASY	N/A	N/A
Identity Management	YES	NO	NO
Bi-Directional Communication	YES	NO	NO
Overhead	HIGH	HIGH	HIGHEST

Fig.4.1. Feature comparison

- Now we will compare XMPP pubsub with other HTTP based protocols based on some metrics:
    - Latency
    - Bandwidth
    - Network overhead
- **Latency:**

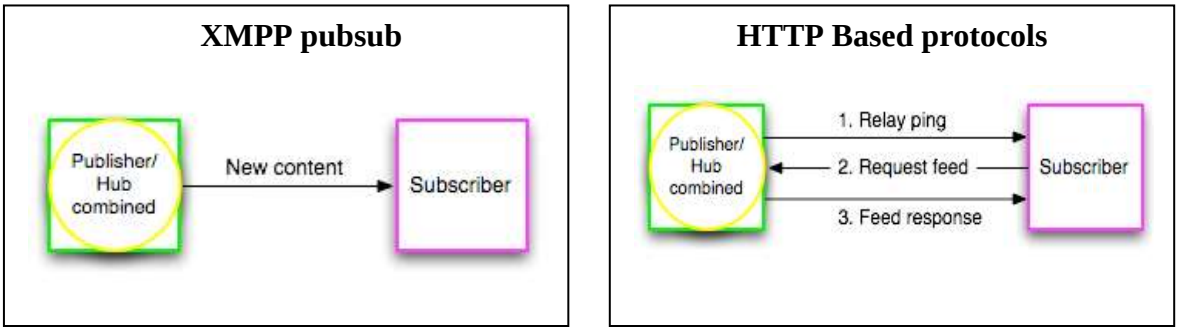
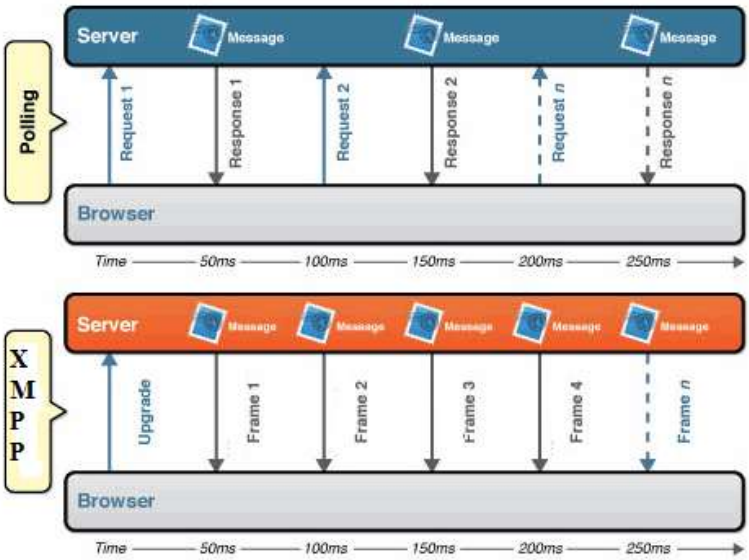


Fig.4.2. Latency Comparison

Minimum 3 network hops are required with HTTP based protocols whereas only a single hop is required with XMPP pubsub. So XMPP pubsub is 66% faster than the other one.

Latency Comparison between polling and XMPP applications.

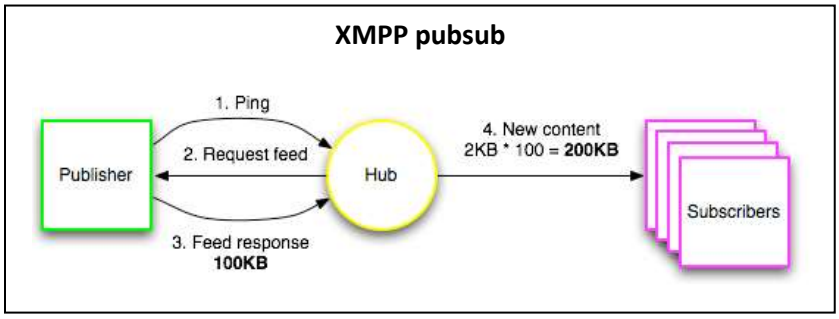


From the above figure it has been shown that the latency in the case of polling will be more as compared to XMPP latency. This is because message travels from server to the browser in about 50 milliseconds in case of polling and also the new request would be sent to the server as soon as the response will be completed. It means that another 50 ms would be taken by the new request and this will further increase the memory consumption of the server.

On the other hand, in case of XMPP, there is the reduction in latency as shown in the figure. Once there is an up gradation in the connection, messages can flow from the server to the browser the moment they arrive. It still takes 50 ms for messages to travel from the server to the browser, but the XMPP-TCP connection remains open so there is no need to send another request to the server.

- **Bandwidth:**

  Consider a case in which data of about 2KB along with 100 number of subscriber would be feed for a single new item.



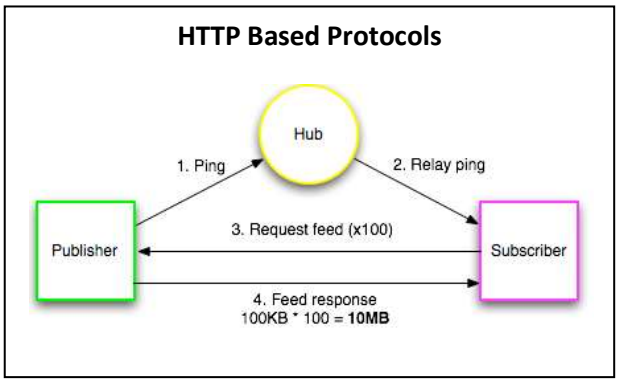


Fig.4.3. Bandwidth comparison

XMPP pubsub needs 98% less data (which means HTTP based protocols needs 50x more). 100x more HTTP requests are required to be delivered by publisher in case of HTTP based protocols.

- **Network Overhead:**

  Three different use cases:

HTTP POLLING

- **Use case A**: 1,000 clients polling every second: Network throughput is (500 x 1,000) = 500,000 bytes = 4,000,000 bits per second.
- **Use case B**: 10,000 clients polling every second: Network throughput is (500 x 10,000) = 5,000,000 bytes = 40,000,000 bits per second.
- **Use case C**: 100,000 clients polling every 1 second: Network throughput is (500 x 100,000) = 50,000,000 bytes = 400,000,000 bits per second.

XMPP

- **Use case A**: 1,000 clients receive 1 message per second: Network throughput is (42 x 1,000) = 42,000 bytes = 3,36,000 bits per second
- **Use case B**: 10,000 clients receive 1 message per second: Network throughput is (42 x 10,000) = 420,000 bytes = 3,360,000 bits per second
- **Use case C**: 100,000 clients receive 1 message per second: Network throughput is (42 x 100,000) = 4,200,000 bytes = 33,600,000 bits per second.

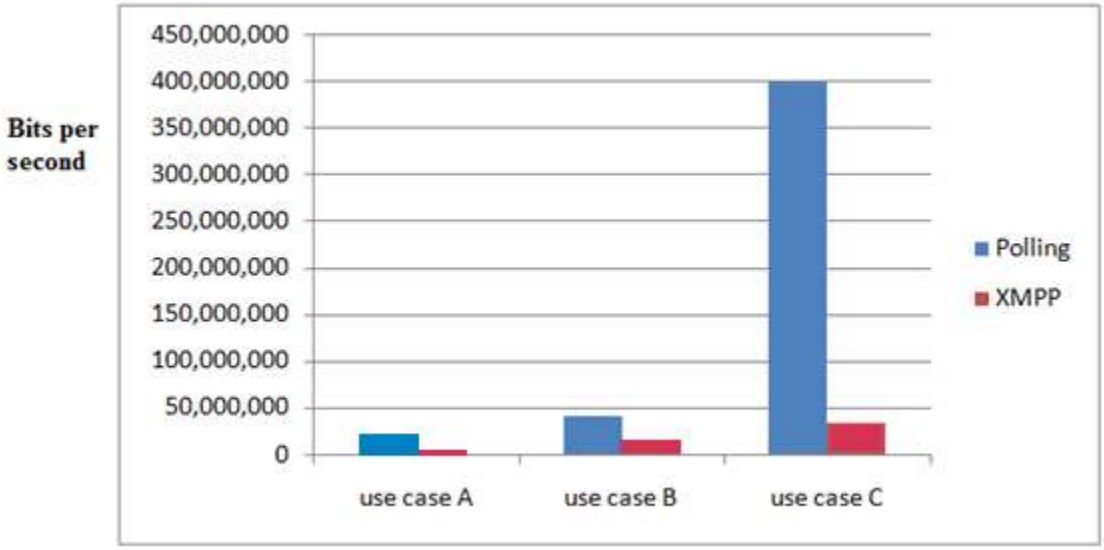


Fig.4.4. network overhead

- XMPP is not only suitable to carry out internal communication i.e. communication between the volunteer cloud entities but is also applicable to provide an effective external communication (the communication between volunteer cloud and clients) by exchanging XML stanzas.

- XMPP </presence> stanza which is based on publish-subscribe method makes it possible for all the components and sub-components of volunteer cloud to communicate with each other by broadcasting its network availability to other entities through communication. This has been made possible by implementing XMPP over middleware layer so as to form XMPP based messaging oriented middleware.

- The study also suggests that XMPP bring out a secure and more effective communication between a volunteer cloud and clients so that clients can effectively contribute and consume their resources to the volunteer cloud. Both the client and the volunteer cloud carry out an interaction by connecting their respective XMPP servers.

- The present study has brought out volunteer cloud computing to another level by providing an effective and more secure communication so that clients can freely consume and contribute their resources.

- This study has made it possible for the commercial clouds and other organizations to make use of large computing power offered by volunteer cloud in order to meet the changing requirements and needs of IT industries. It also suggests that to deploy short and long term projects, organizations can make use of the resources offered by volunteer cloud.

- The present study works for the betterment of the computing paradigms as by using XMPP, effective communication will be carried out which will help the organizations to cut down IT cost, and also help in reducing overall power consumption and also deduct the cost related to data migration.

- XMPP offer a communication channel with which organizations can even donate their unused computing resources. Scientists and researchers utilize the computing power provided by volunteer cloud for generating scientific outcomes by establishing this communication channel.

- This thesis is a valuable contribution towards establishing XMPP as the most suitable protocol for providing an interoperable environment so that clients can efficiently interact with the volunteer cloud and also volunteer cloud entities can freely communicate with each other.

5. References

[1] A. Andrzejak, D. Kondo, and D. P. Anderson, "Exploiting non dedicated resources for cloud computing," *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, pp. 341-348, 2010.

[2] A. Marosi, J. Kovács, and P. Kacsuk, "Towards a volunteer cloud system," *Future Generation Computer Systems*, Mar. 2012

[3] Abdulelah Alwabel, Robert Walters and Gary Wills "towards an architecture for IaaS volunteer cloud" digital research 2012.

[4] Fernando Costa, Luis Silva, Michael Dahlin, " volunteer cloud computing: mapreduce over the internet", Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on.

[5] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. Anderson, "Costbenefit analysis of Cloud Computing versus desktop grids", in *Proceedings of the 2009 IEEE international Symposium on Parallel & Distributed Processing*, pp. 1-12, May 2009

[6] T. Dillon, C. Wu, and E. Chang, "Cloud computing: Issues and challenges," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 2010, pp. 27–33.

[7] Edward Curry, "Message-oriented middleware".

[8] Vincenzo D. Cunsolo, Salvatore Distefano, Antonio Puliafito and Marco Scarpa, "Cloud@Home: Bridging the Gap between Volunteer and Cloud Computing", in *ICIC'09 Proceedings of the 5th international conference on Emerging intelligent computing technology and applications*, 2009.

[9] V. D. Cunsolo, S. Distefano, A. Puliafito, and M. Scarpa, "Applying Software Engineering Principles for Designing Cloud@Home," *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pp. 618-624, 2010.