IntellectCloud –cloud application's ménage

Deepti Ghuge
Information Technology & Computer Science Department,
VPM's RZ Shah College of Arts, Science and Commerce
Mulund(E), Mumbai, India

Girish Tere
Department of Computer Science,
Thakur College of Science and Commerce,
Mumbai, India

ABSTRACT
With the increasing use of cloud computing, enterprises are facing the enormous challenges in setting up an efficient and elegant infrastructure for providing cloud based services. Due to huge demand and curiosity by the entire community, many small scale and middle scale enterprises are exploring the various possibilities of doing business using cloud computing. Another considerable emerging concept in IT industry is Social Networking. Popularity of Orkut, Facebook, Twitter, MySpace, LinkedIn and now Google+ is tremendous. Portal such as Facebook has become essential part of Internet savvy individual. Industry has also utilized this amazing concept, social networking to advertise their products and reach out to end customers. Enterprises are very keen to do more with social networking and cloud computing in near future. Just like website is essential part of branding of any business, in near future, having own social networking portal for each organization will become essential. It will help an organization to increase their brand value using social networking portal. We propose, a model which allows deploying any social application over the cloud based architecture named IntellectCloud. The architecture is distributed in nature, utilizing smallest computing device such as iPhone or Laptop to contribute to cloud computing resources requirement. This paper outlines our vision of IntellectCloud, which can accommodate almost any cloud based application. It also focuses on core storage architecture and framework required for IntellectCloud.

General Terms
Cloud Computing

Keywords
Cloud Computing, Grid computing, Social Networking, Cloud Architecture, IntellectCloud.

1. INTRODUCTION
Web 2.0 has changed the way, we uses technologies for our day to day life. Emergence of Social Network in recent years has revolutionized the way of using internet and computers. Social Networking portal such as Facebook has exceeded user base more population of large countries. Facebook has over 800 million active users[1]. Enterprises are looking at social network to advertize and market their products, many industries are successfully using Twitter or Facebook to publish their services to market[2]. Social Networking portal has not remained just a social meeting place rather it has span boundaries of communication with individuals separated apart by physical and geographical boundaries.

Another noticeable technology in the world of Computer Science is Cloud Computing. Topmost Information Technology giants are putting substantial efforts in developing cloud based resources and applications. Cloud computing is a style of computing which is having dynamically scalable virtualized resources provided as a service over the Internet. It reduces the time required to procure heavy resources and boot new server instances in minutes, allowing one to quickly scale capacity, both up and down, as one's requirement changes. Cloud terminology is burning in the market and is ready to cater to the small and medium business segment.
Cloud computing customers do not generally own the physical infrastructure required for software platform to run. This physical infrastructure is rented from third party cloud service provider. This technique saves lot of expenditure as compare to having own infrastructure. The service provider normally rents out scalable and huge infrastructure to multiple customers which bring them revenues. Cloud computing consumers consume resources as a service and pay only for resources that they use. Currently Google[5] , Amazon, Microsoft, Oracle, force.com, rackspace, IBM etc are leading cloud service providers.

In this paper, we outline our vision of IntellectCloud – An architecture for Cloud Applications. And elaborating on how it can be used for effectively implementing Social Network over cloud environment and customizing it for any type of organization. All enterprises are creating digital content such as creating profiles, pages, groups, competitions on social networking portals [2,3] such as Facebook, Twitter etc. Such social networking portals provide better facilities to publish the content as photos or videos and share it across the globe with the few mouse clicks. Of course, these social networks were not designed to do product advertising or branding, however industry has started utilizing the tremendous power of these tools. As in year 1998 onwards, website were become essential for each enterprise to add value to their brand name in market, in near future, having own social network would be essential requirement of each enterprise. It would not be possible for everyone to develop own social networking portal due to complex technical and development requirements for IT projects and cost associated with it[4]. We here propose a model to effectively customizing own social network using cost effective terminology such as cloud computing. We put

social network over the cloud and customize it as per the client requirement on fly using open source technologies and our IntellectCloud architecture. Further sections elaborate on entities of CloudIntellect and Cloud Object Persistence Framework for storing objects in cloud.

2. RELATED WORK

There are multiple instances of Social network and Cloud computing integration. However, most examples uses cloud to rent out applications to client and that are hosted on huge capacity data server. Some example creates scalable applications within the Social network. For example, Facebook users can build scalable Cloud based applications hosted by Amazon Web Services[4]. However, there is no literate which uses smallest computing devices and provide ability to host large amount of data service requirement.

Similar efforts has been made by Eucalyptus[9] and Cloud.com with their software and CloudStack-open source cloud computing[10] which allows us to deploy and manage large networks of virtual machines are highly available, scalable cloud computing platform. However, cloud stack mainly focuses of IaaS(Infrastructure as Service) and provides effective way to manage and monitor cloud entities and client.

3. THEOROTICAL ASSUMPTIONS

IntellectCloud allows us to customize any application for variety of clients. We demonstrate it using Customized social networking which is cloud based service for Social Networking. Any organization wish to have their own social network to span their own community, can subscribe to IntellectCloud. In order to use this service, there will be minimal configurations required for new client such as their logo, themes, menus, layout plans and additional or unnecessary fields. Each customer will give these details while subscribing for the service and their own social network can be ready within few minutes. This will essentially minimize cost of development of own social networking portals and infrastructure required to maintain this portal. It will also save time required for development and testing of the portal. We will have cloud based storage system for storing data of the entire client in IntellectCloud (explained in next section). Generally all client data will sit in single database having some segregation keys to differentiate between clients. Each customer would be given some unique credential during subscription to IntellectCloud. This credential will be key to separate data of each customer with another customer. For example, member schema will have clientID column which will keep ID of the client to which the member belongs. This way, client ID can be maintained in all the schema of the database. There will be way to identify the client during application launch for each member. This ID once identified, can be retained throughout the browsing session of the member of that client and it can be passed along with every request made by the client to the database. All queries of the database will have additional clause to filter data for the client using client credential. This technique will ensure that each member can see their own client's data. This way, member would be unaware of shared cloud architecture.

4. INTELLECT CLOUD ARCHITECTURE

IntellectCloud, as depicted in below figure, made of some entities which take part in smooth operation of IntellectCloud. IntellectCloud model is designed on top of distributed system4. It uses distributed nodes (called cloud entities) each, having their own functionality. IntellectCloud is designed using Java.

Architecture supports various computing devices such as high capacity data and application servers, laptops, desktops and even small computing devices such as mobiles and ipad. These all devices can participate in IntellectCloud services with some agreed upon terms and conditions and can serve as active node in the cloud. IntellectCloud architecture is shown in Figure 1. Following section briefly explains role of each entity in IntellectCloud:

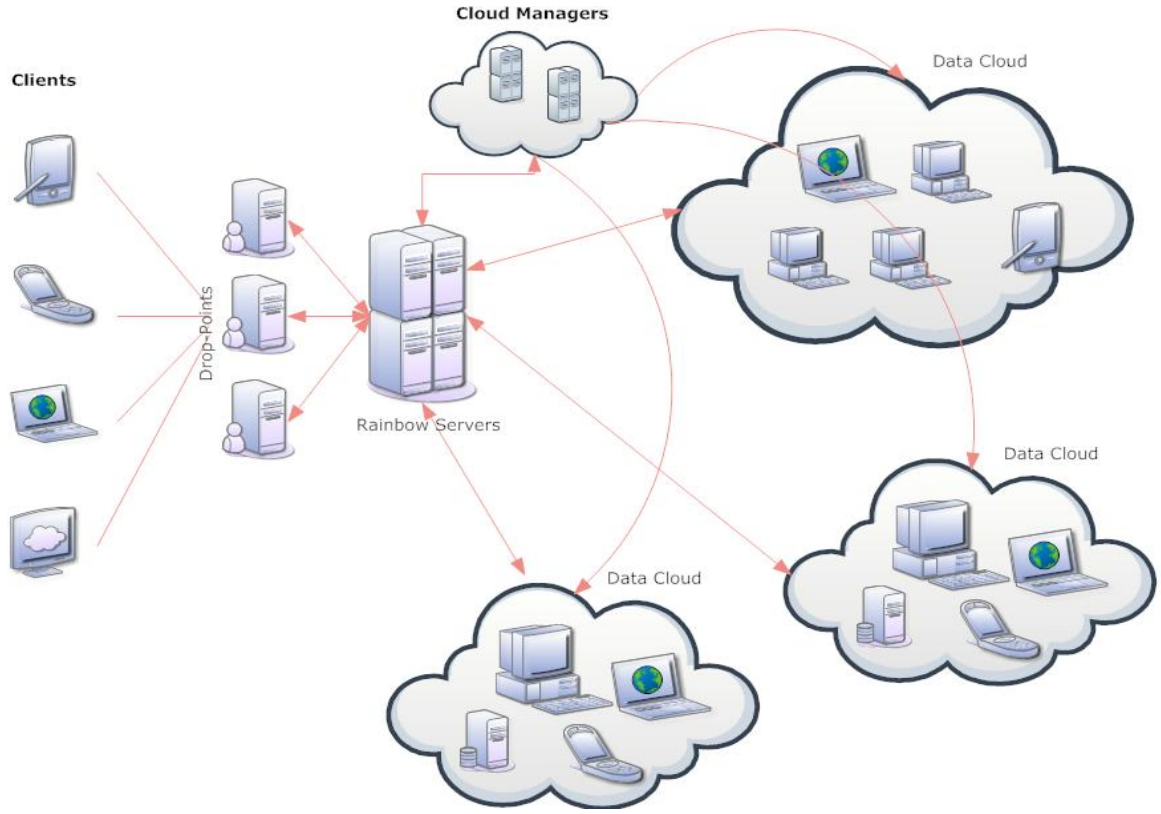

Fig 1: IntellectCloud Architecture

4.1 Cloud Managers

Cloud Manager itself is group of entities which exists in cloud and manages the entire architecture. They are responsible for startup, security management, performance tuning, load balancing and shutdown of Data clouds. Cloud Managers communicates with Rainbow server to manage the Data Cloud. Security credentials, data encryption and decryption keys, storage policies, passive and active status of nodes, terms and condition management, criticality of data are some of the key factors managed by cloud managers. They communicate with entire IntellectCloud using faster, asynchronous communication API[7]. The communication is partially peer to peer which remove dependency and risk of server failures.

4.2 Data Cloud

This is one of the essential entities of the IntellectCloud architecture. This entity hosts the data of the clients. The data could be stored in file system or in RDBMS managed by cloud. Data is hosted in encrypted format whose control is available only at Cloud Managers. Data cloud is very proactive entity in the IntellectCloud. It serves data just by looking at sparks and sends it to Rainbow servers. Spark is one of the operations of IntellectCloud. When spark occurs each cloud learns, that there would be request for data. Then data-cloud identifies what data is requirement for data. Each data-cloud then reads the query from IntellectCloud and searches data in their own data-store. Once they get the data, they send it to Rainbow server. Data cloud are competitive in nature with other data clouds, they compete to serve data faster to Rainbow server. Any device can act as member of Data Cloud group. The device which has space to store data, can participate in IntellectCloud with some agreed upon terms and conditions. Cloud-Managers decide which Data-Cloud the entity will go. Cloud Managers also setup policy of data (whether entity would hold critical data or non-critical data). Each entity in Data-Cloud faster communication API developed on top of Core TCP and UDP sockets in java as well as web services or web-sockets to communicate remote entities in the group. Data Cloud uses Cloud Object Persistence Framework to store objects.

4.3 Rainbow Server

It's the most essential entity of IntellectCloud architecture. As name suggests, they hide cloud from end users. Rainbow servers itself is group of processing servers which can communicate asynchronously with Drop-Points(interface to the clients), Cloud Managers and Data Clouds. They handle routing of data and acts as middleware and processing units of IntellectCloud. They are responsible for doing *Sparks* (Data Query Request) of the query coming from client. The spark is broadcast of required data by client to the Data-Cloud. *Sparks* is learned by Data-Cloud and then they serve the data spontaneously. Once *spark* is read by Data-Cloud, processing triggers at Data-Cloud, to serve data to Rainbow Server. Rainbow servers are multithreaded in nature and can handle millions of requests at one moment. Rainbow servers maintain

the complete data about Cloud-Data entities. This information is essential for communicating and routing data in the IntellectCloud.

4.4 Rainbow Server

Drop points are request handlers for clients. They could be web applications, web service endpoint or desktop based applications etc. They accept requests from client and send it across the IntellectCloud via Rainbow Servers. They communicate with Rainbow servers to exchange data. Drop points are hosted on specific application servers and configured with IntellectCloud.

5. CLOUD OBJECT PERSISTENCE FRAMEWORK (COPF)

As we know, any application is made of User Interfaces, Application Logic (Code) and Backend (Database), each entity of IntellectCloud performs specific roles. Cloud Object Persistence Framework[8] is an API for storing and retrieving objects effectively in IntellectCloud architecture.

5.1 Defining Schema

Each application has schema to store its data. The first requirement to use COPF is to define schema of the application. IntellectCloud provides simple web interface to define the schema. Schema is defined in Object Oriented fashion. Each Entity is represented as one class having attributes as instance variable of that class.

For example to define customer and Address entities, we create two separate classes Customer and Address using COPF Web Interface as shown in figure 2 below.

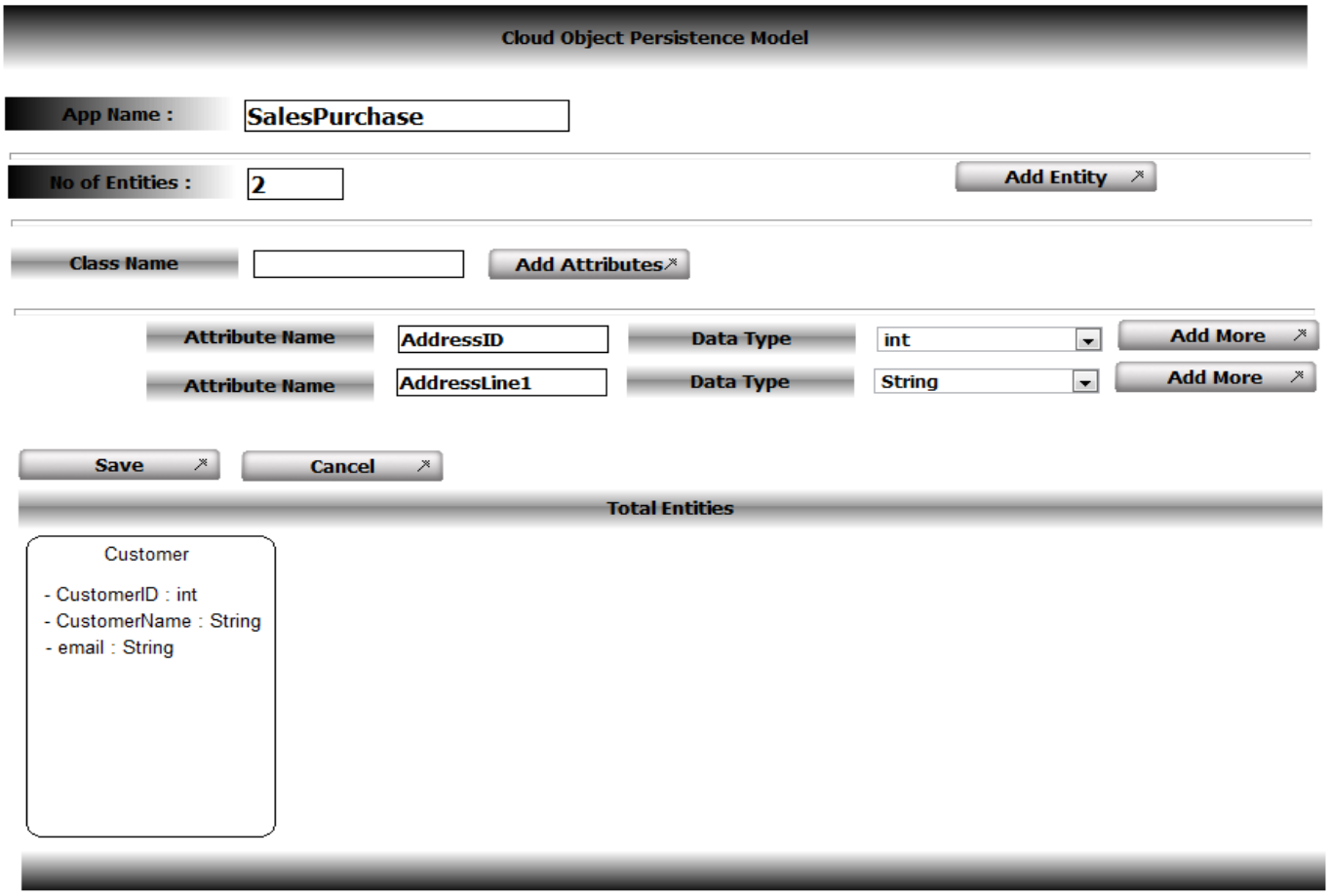

Fig 2: COPF Web Schema Interface

Once the schema is created, its associated with client application and stored at Rainbow servers . Rainbow server uses this schema to query objects using COPF.

5.2 Operations of COPF

There are eight major operations defined in COPF. These operations help in querying and storing data in IntellectCloud.

5.2.1 CREATE – Creates new object entry into IntellectCloud

Syntax:
COPF.CREATE(<objectName>,<ObjectInstance>)

<ObjectName> - Name of the Class
<ObjectInstance> - Actual instance of the object
It saves the given Object into IntellectCloud.

5.2.2 DELETE – Delete the object entry from IntellectCloud.
Syntax:
COPF.DELETE(<objectName>,<ObjectInstance>)

<ObjectName> - Name of the Class
<ObjectInstance> - Actual instance of the object
It deletes the specific object from the IntellectCloud.

5.2.3 UPDATE – Update the object from IntellectCloud.
Syntax:
COPF.UPDATE(<objectName>,
<ObjectInstance>,
<ByAttributeName>)
<ObjectName> - Name of the Class
<ObjectInstance> - Actual instance of the object
<ByAttributeName> - Name of the attribute by which object needs to be updated.

It updates the specific object in IntellectCloud by specific object and attribute.

5.2.4 READ – Read single object from IntellectCloud.
Syntax:
COPF.READ(<objectName>,
<ByAttributeName>.
<AttributeVAL>)

<ObjectName> - Name of the Class
<ByAttributeName> - Name of the attribute by which object needs to be read.
<AttributeVAL> - Value of the attribute by which object needs to be searched.

It read the specific object form the IntellectCloud by given attribute name and value.
5.2.5 READALL – Read all objects from IntellectCloud.
Syntax:
COPF.READALL(<objectName>,<[COUNT]>)

<ObjectName> - Name of the Class
[<CLOUNT>]- Total no of Objects to Read

It read all the object from the IntellectCloud. Optionally we can specify the maximum object we need.

5.2.6 SELECT [<OBJ.K1, OBJ.K2,.. OBJ.KN>] - Read specific attributes of the object(s) from IntellectCloud.
Syntax:
COPF.SELECT(<objectName1.AttributeName1>
<objectName1.AttributeName2>,
(<objectName1.AttributeName3>
.
.
(<objectName2.AttributeName1>
MATCH(<ObjectName1.AttributeName
=ObjectName2.AttributeName)

<ObjectName[X]> - Name of the Class
AttributeName[X] –Name of the attribute of the class

It selects the specific field from the specific object. It also supports joining of multiple objects via some relationship

5.2.7 REMOVE <OBJECTNAME> - Remove particular class from Schema definition.
Syntax:
COPF.REMOVE(<objectName>)

<ObjectName> - Name of the Class
It removes the entire class from the schema.

5.2.8 DROP <AppName> - Drop the entire Schema structure of the IntellectCloud.
Syntax:
COPF.DROP(<AppName >)

<AppName> - Drops the entire schema.
It removes the entire schema defined for application.

5.3 Query Processing

Query processing of COPF is explained below with the help of Class diagram shown in Fig. 3. Each query is read by *QueryReader* where query is lexically analyzed for Syntax.

If query is passing lexical and syntax analysis, then its forwarded to QueryParser, where query is identified as one of the operation. There are specific implementation classes for each type of operations where complexity of each operation is taken care. Once the query is formed to be executed, then it is handed over to QueryEngine. QueryEngine uses Communicator to communicate the query to DataClouds. QueryEngine is also responsible for receiving data from the DataCloud and gives the returned objects to OutputProcessor. OutputProcessor formats the objects into required format and sends back to the DropPoint calling routine.

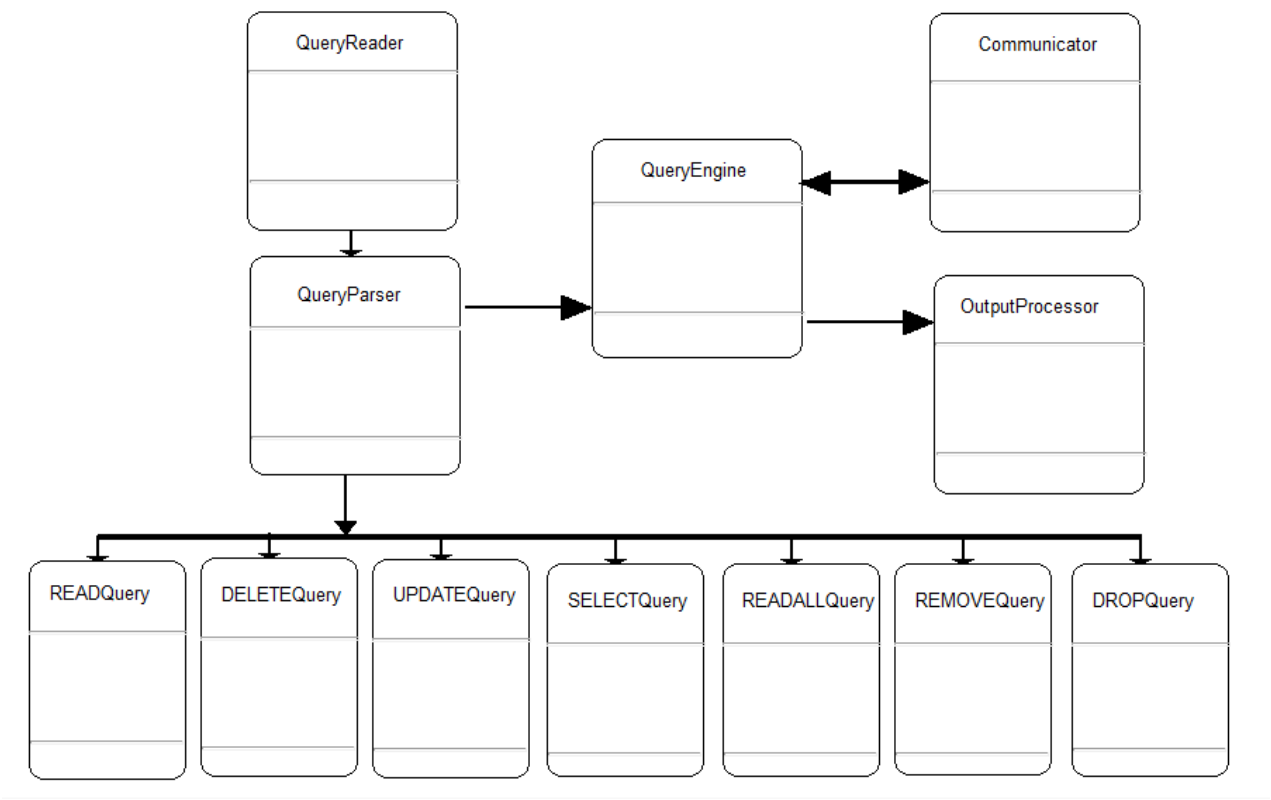

Fig. 3 Class Diagram for Query Processing Engine

5.4 Storage Format

Currently Cloud Object Persistence Framework stores the object in binary format in Hard disk of the Data Cloud. We are using Java ObjectOutputStream and ObjectInputStreams to read and write the objects to Secondary storage. Objects are indexed in special database at each DataCloud using their primary keys. Indexing is done as

<ObjectName>
<PrimaryKeyAttr>
<PrimaryKeyVAL>
<ObjectFileName>

Where, <*ObjectName*> is name of the object which is being persisted. <*PrimaryKeyAttr*> is the Value of primary key attribute of the object, <*PrimaryKeyVAL*> is value of the primary key of the object and <*ObjectFileName*> is the name of the file of an object. When objects are requested via some query, index table is checked and associated file name is retrieved. Object is read from that file and returned to the Query Engine.

6. CASE STUDY- SOCIAL NETWORK USING CLOUD

IntellectCloud provides social network as Service to customers. Social network is an application developed to be customizable as per the client[3]. Rather than hosting this app on their own servers, we demonstrate how it can be shared across different enterprises and organizations using IntellectCloud. Each organization or enterprise or an individual willing to have their own social network branded with their name can registers for Social Network at IntellectCloud. The client is given the specific credentials which is key to segregate clients from others. The data from different clients would be stored at one place using IntellectCloud but differentiated by Client ID. Drop Points have Client specific processor which will differentiate requests for different client and would route the data accordingly. Each client's social network can be differentiated by Look and Feel, data, information and business rules as stated by client during configuration. Social network user is transparent about this segregation and would feel like using dedicated social networking portal for his/her enterprise/organization. Client interface to IntellectCloud is shown in Fig. 4.

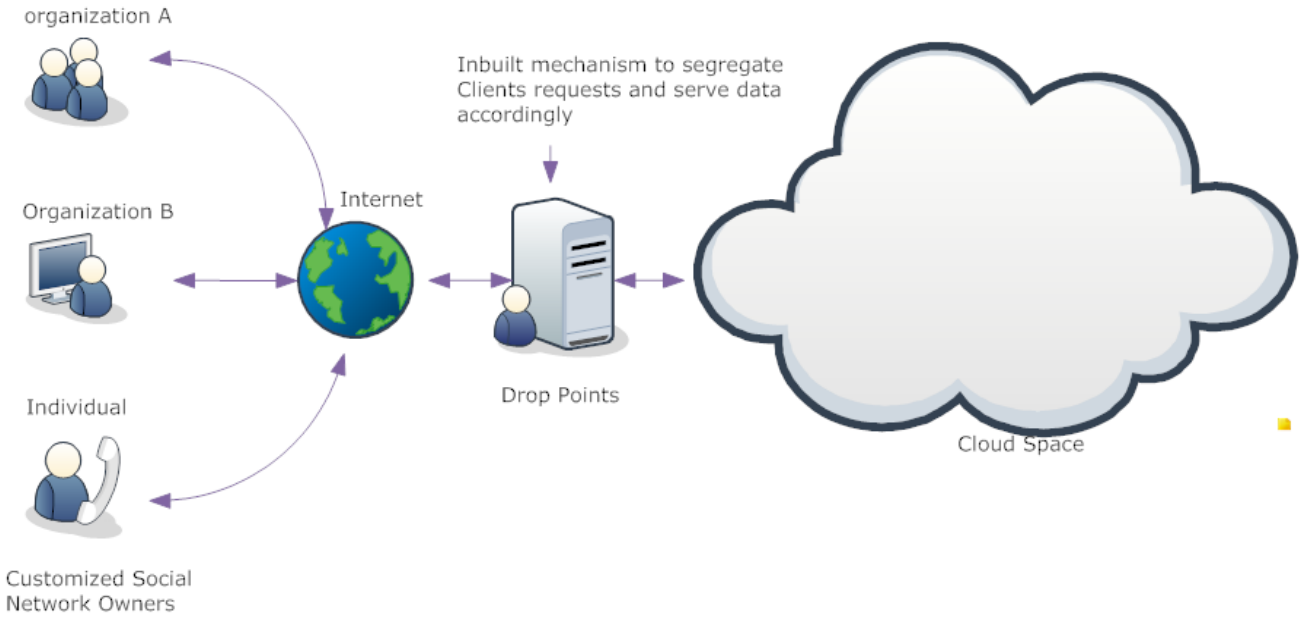

Fig. 4 Client Interface toIntellectCloud

7. CONCLUSION AND FUTURE WORK

This paper has presented the architecture and implementation of IntellectCloud; cloud application ménage, where we can host any application and provide it as cloud service. We have also proposed an API called Cloud Object Persistence Framework which is used to move object across the cloud entities to effectively store data. Using our architecture, any hosting providers can setup cloud hosting with minimal setup. Enterprises having many computing devices such as user desktops in IT firms, can setup IntellectCloud to utilize unused storage and processing capacities of these devices. We have also shown that Cloud Object Persistence Framework can be used for cloud based object oriented data storage.

Our future work aims to extend Cloud Based Object Persistence framework more to add complex database and querying operations and enhance security in moving objects across the cloud entities. Additionally, intellect cloud would be enhanced to deliver complete virtual desktop to client, giving feel of Cloud Operating System to the client.

8. REFERENCES

[1] Angela Bartels, Picking the Right Option for Hosted Storage,2011, http://www.rackspace.com/knowledge_center/white paper_hostedstorageoptions

[2] Chard, K.; Bubendorfer, K.; Caton, S.; Rana, O.; Social Cloud Computing: A Vision for Socially Motivated Resource Sharing; IEEE Transactions on Services Computing, Volume: PP , Issue: 99, 2011 , Page(s): 1-6

[3] Dikaiakos, M.D.; Katsaros, D.; Mehra, P.; Pallis, G.; Vakali, A.; Cloud Computing: Distributed Internet Computing for IT and Scientific Research, Internet Computing, IEEE, Volume: 13 , Issue: 5, 2009 , Page(s): 10 - 13

[4] Hansen, J.A.; Adding privacy and currency to social networking, International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010, Page(s): 607 - 612

[5] Jesse James Garrett, "Ajax: A New Approach to Web Applications," 2005, http://www.adaptivepath.com/ideas/essays/archives/000385.php

[6] Microsoft, "Windows Azure Storage Services REST API Reference," Copyright 2011 Microsoft, http://msdn.microsoft.com/en-us/library/dd179355.aspx.

[7] Nurmi, Daniel, Rich Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, Dmitrii Zagorodnov, "The Eucalyptus Open-source Cloud-computing System," Proceedings of the 2009 IEEE/ACM International Symposium on Cluster Computing and the Grid, May 2009.

[8] CloudStack, open source cloud computing, www.cloudstack.com/cloudstack.html.