A Distributed-SOA model for Unified Communication Services

Shrikant Kanaparti
Samsung Electronics
shrix@ieee.org

Abstract

This paper describes an architectural framework adopting the principles of Service Oriented Architecture for deploying unified communications over the Internet. Also discuss the possible ecosystem based on this approach, and the business model that could probably emerge using such a model.

In this architecture, we explore the benefits of the Open Services Gateway Initiative (OSGi) framework for providing us with an ideal service execution platform on the servers, with subscriber client devices on the Internet deploying their services dynamically on them. With a cluster of such networks, we form a distributed communication network for different regions of deployment, with an interaction model between them.

1. Introduction

Over the last couple of years, we have observed substantial progress in communication technologies. We now have a lot more choice in the way we communicate, and interact with each other; some of the common forms include voice, instant messaging, email, video, etc. The proliferation of these forms of communications have been enabled by the usage of Internet Protocol (IP) technologies, which also helped deploy it more rapidly into the market, with the Internet providing the necessary backbone for connectivity.

The Internet happens to be an ideal platform for communications for several reasons, one of them being its ubiquitous accessibility. It allows a myriad of devices that could be connected over a common protocol, which is universally accepted. Another reason could be the simplicity of these protocols, which are transparent across a range of access medias, both wired and wireless.

World Wide Web (WWW) revolutionized the Internet, as it introduced one of the most simplistic ways of developing required services; including deployment was not difficult over it. Consider for example, most of the Webmasters are not technical persons, yet are able to do a great job. This level of simplification could be more appreciated when one considers the alternate efforts required to perform an equivalent job. Perhaps humans have come to a stage now, where the development on the Web is taken for granted. Perhaps that is the stage we would like to be in, developing our future communications ecosystem as well.

Web Services have brought in a vast change in the way we interact and do business on the Web, which has evolved as a natural transition in the way we did our work traditionally. Built over a simple model utilizing the Simple Object Access Protocol (SOAP) message to access, control and integrate various services remotely. Web Services were designed for e-business applications primarily, having a one-way, stateless, request/response interaction pattern between the various services, and loosely coupled.

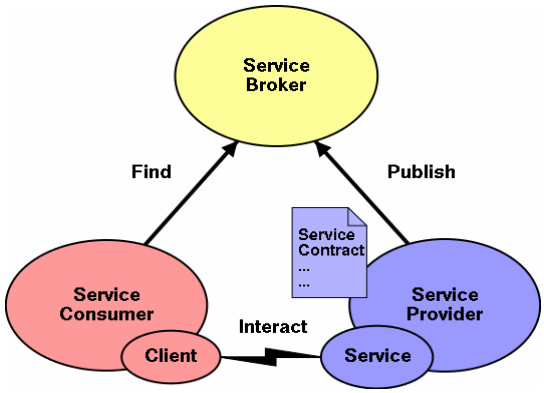

Fig.1**:** The Web Services architecture

Service Oriented Architecture (SOA) enabled over the Web Services, brings in an architectural paradigm for the deployment of dynamic services. Interactions amongst the various services specified by means of generic interfaces between them, for performing certain specific tasks. While the application-specific semantics are carried across using messages, the data itself complies with the specific protocols that it adheres to.

There are perhaps various architectures based around the principles of SOA. One of the architectures that could suit our need for a communication application specifically is the Open Services Gateway Initiative (OSGi). We adopt this as the reference SOA platform for explaining our system through out this paper. Although other frameworks having similar characteristics may also be suitable.

978-0-7695-3310-0/08 $25.00 © 2008 IEEE
DOI 10.1109/ICWS.2008.127

The benefits that the OSGi specification brings in are its standardized, component-oriented, computing environment specifically suited for networked services with the required security, reliability, quality of service, communications capability, etc. built into it. Other aspects include the lifecycle management of the software components on a device from anywhere on the network, which can be dynamically installed, updated, or removed on the fly, and without disrupting the device, or any of its other services. We take advantage of this unique characteristic for obtaining the functionality of distributed SOA, deployed over the Internet, providing a unified communication services framework.

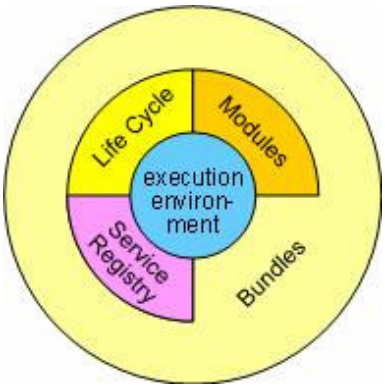

Fig.2**:** OSGi Service Platform – for dynamic SOA

- At the core of the framework is the execution environment which is responsible for the execution of all applications, deployable by any devices on the network, and can get connected to this Service Platform
- The module layer is a policy manager for the execution of all objects present within. It is fully integrated with the security architecture, enabling the option to deploy as closed systems, walled gardens, or completely user managed systems dynamically.
- The lifecycle layer adds bundles (i.e. applications in OSGi) to be dynamically installed, started, stopped, updated and uninstalled, introducing the dynamics that are normally not part of an application.
- The service registry provides a cooperation model for the bundles to share objects, and comes with a well-defined set of events to handle the coming and going of services on the Platform. The service security model provides an elegant way for a secure communication between bundle passes.

The OSGi Service Platform provides the functions to change the composition dynamically on the device of a variety of networks, without requiring restarts. To minimize the coupling, as well as make these couplings managed, the OSGi technology provides an SOA that enables these components to dynamically discover each other for collaboration. And comes with a number of standardized (thou optional) services including the Logging, Configuration, HTTP, XML, Wiring, IO, Event Handling, Device Discovery & driver loading, User Authentication & Authorization, Preferences, Device Management, Component Wiring, Event Management, UPnP and many more. Also provides a number of other standardized utilities. These services, along with a framework to execute them reliably, make the OSGi Service Platform an ideal option for deploying secure communication services.

The intent of this paper is to take advantage of these benefits of a dynamic SOA framework, along with its ability to distribute these services across, thus giving us an ideal platform for unified communications. The focus here is not on any specific communication service per se, but rather giving an end-to-end infrastructure, and the methodology for deploying and delivering these services.

2. Services, Trends & Concerns

Personal communications in all forms continues to be the driving factor for all social interactions. People like to stay connected always, keeping in touch with their family, friends, colleagues, etc. when they are physically away from each other. The earlier form of simple voice communication based on the analog switching has transitioned to the digital packet switching.

On the Internet – a worldwide, publicly accessible network for devices that adheres to the standard IP – we have seen a rapid progress, with a range of communication services deployed over the WWW, mostly disparate. The services that initially started with simpler ones like file-sharing and email, eventually led into other forms of communication such as instant messaging, voice, video, etc. The unique benefit that this platform brings is the seamless integration of most of these services on a single IP plane, thus giving scope for a unified communications platform. Blending services with other information, such as location, presence and context of the user, has lead into more exciting forms of communication.

Developments in broadband and wireless technologies have vastly contributed to the rapid growth of communications over the Internet. Increasing bandwidths of the broadband pipe has led the way for video services, which were not previously possible. Progress in wireless technologies has given the benefit of communicating on the move, from anywhere, anytime.

Communication network operators have been constantly finding innovative ways to deploy their services, and prefer to have them all integrated, giving their consumers an advantage of a single-point deployment, with benefits to interconnect across them.

E.g., a voice call can be transferred across as an instant message, or a fax can be transferred as an email, etc.

Coming to concerns, an aspect observed is the trend in mobile communications technology that keeps changing so often. This brings in changes in the deployment platform, and the user to upgrade on to a newer hardware device for taking benefit of these newer services. These changes seem to only go on and on, with no end in sight. Thus taxing the consumers continually, leaving no option but to be stuck with this model.

While improvements in technology and introduction of newer services are most welcome, there certainly needs to be a better way perhaps. Thus, a need for us to look ahead for a system that brings in a more flexible way of deploying newer services, without the user having to make any changes at their end, whatsoever. The only way this seems to be feasible is by enabling all necessary services as software, and a generic hardware that is deployable over a network that is universally present – the Internet. This infrastructure needs to be well adaptable, scalable for all the needs of future communication, without compromising on the security, as these communications are most often restricted to a small group.

A dynamic SOA comes in as a promising solution for this purpose. An aspect that needs focus is the way these services are deployed in the future. It is also foreseen, that in the future the services should be deployable by anyone who is on the network, from anywhere, anytime. With a pricing model appropriately based on the usage of services, rather than the usual slab subscription model, and perhaps also be supported by alternate business models, such as the advertising for instance.

3. Communications Services Architecture

An SOA approach has been a proven model in the enterprise world, and the advantages that it brings in for a major establishment are perhaps well known. In this architecture, we try to extend on this very concept, taking a distributed approach for addressing the requirements of the communications world, thus introducing the Distributed-SOA (D-SOA). The D-SOA is an appropriate paradigm for organizing and utilizing distributed capabilities for deploying dynamic Services across a cluster of Service Platforms, while it also gives the required control for different ownership domains to function independently, if required.

This distributed approach allows us to bring in Services depending on the communication range that a particular technology can offer as in the case of wireless, or may be configured by the Network Operator as required. Distributed characteristics are not with respect to the processing or sharing of resources, as perceived normally. In this system, it is with regard to the distributed SOA peers in a large grid network instead. The only relation that SOA Service Platform maintains with its peer is for the handoff signaling, that is required for a mobile consumer going out of a particular range.

The D-SOA framework is a client-server based grid architecture, where we have a server deploying the Service Platform required for the execution of the various communication services. And the devices around utilize this platform as their hub for specific communication, or task-organizer needs, etc. Some of the components for deploying a given Service are as illustrated in Fig.3.

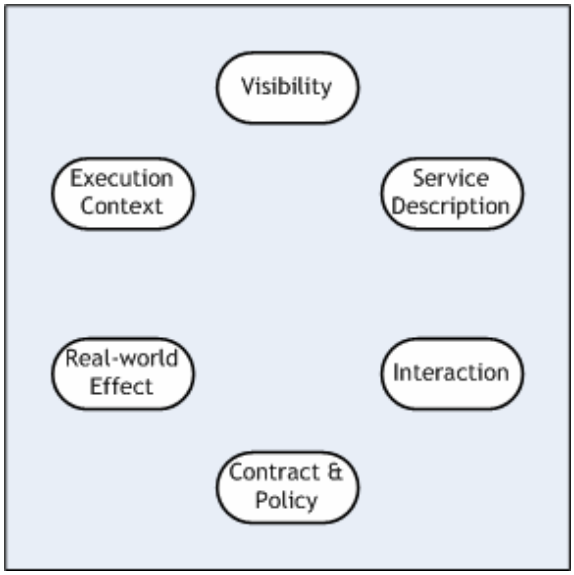

Fig.3**:** Some components of a Service in D-SOA

With an SOA platform introduced by the OSGi Service Platform, we could deploy various Services dynamically on the fly on a Server device. Building more such Service Platforms to distribute Services for its newer subscribers in a region as and when required, could be an ideal approach to enhance the subscriber base. An end-to-end architecture for deploying communication services in the D-SOA is as depicted in Fig.4.

The model describes the following network entities:

1. Central Servers
2. Edge Servers

The actors involved are as follows:

1. Network Operator } Infrastructure enablers
2. Service Provider } Subscribers / Infrastructure users
3. Service Consumer }

A Service Provider provides Services over the Network Operator's infrastructure – Edge Server – following the guidelines provided by the services deployment framework.

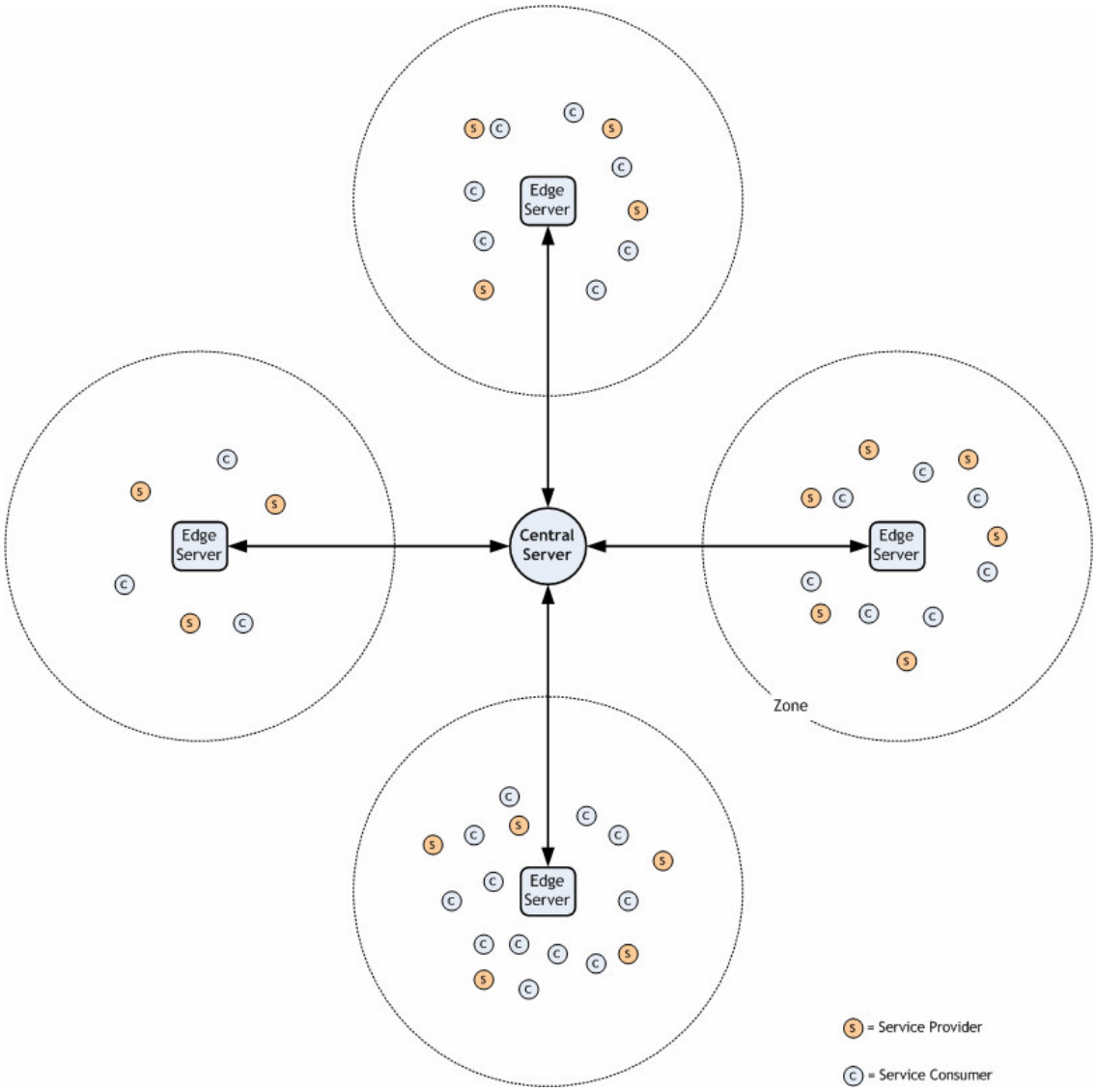

Fig.4**:** D-SOA's end-to-end architecture

3.1 Central Server

Central Server is a centrally located device, which is responsible for maintaining required database of all Subscribers (both Service Providers & Consumers), including their Subscription, Personalization information, and other details. It is also the centralized Authentication, Authorization & Accounting server, for all transactions performed by its Subscribers.

The Personalization information of Subscribers may include their address book, their preferences in terms of specific categories, and acting as their central storage server maintaining all of their other data. These Subscriber Preferences with respect to a category may be gathered over a period of time, observing their behavioral patterns on the system. Further, it could also build a correlation of data with the vast information that is available, between various Subscribers, based on specific patterns of information. This could help enhance the business between the various Subscribers, both Service Providers and Consumers, while acting as a service broker connecting all these entities, with the knowledge base generated.

Central Server's responsibilities also include providing transfer of Services across the Edge Servers when requested. In some cases, it may serve as a backup of these Services in the event of a particular Service failing to execute on a Service Platform (Edge Server) at any point of time. But the Central Server is never an execution plane for any of these Services thou it provides necessary backend support for Services, so as to be up and running always, for execution on the Service Platforms.

The Central Servers maintains live information about a Subscriber, and the respective Edge Server to which he belongs to at any given point of time. This information could be shared across with any other Edge Server as required for establishing a connection across. E.g., a voice call from a caller in a particular Zone needs to get routed to a callee in a different Zone.

Whenever an Edge Server senses the presence of a particular (mobile) Subscriber in its Zone, it could request all necessary details before launching Services across for him, and rather customizing them specific to the Consumer's Preferences, depending on their profile. E.g., it may require building relations between a Service Consumer and his respective banking Service Provider, and provide the necessary authentication for all transactions between them.

It may happen that the Services a particular Consumer gets when in a particular Zone (apart from basic), may not be available for him when he moves out of it. This is a dynamic characteristic, as Services are localized to each specific Zone. Sometimes it could also depend on the Service coverage that a Service Provider desires, and could be extended across different Zones though not belonging to them, with the help of Central Server for a wider coverage. E.g., when a Service Consumer enters the downtown area, he gets Service details such as the latest movie in the theater around there, etc. Alternatively, in the case of Services floating across the Zones, we could think of a scenario wherein an eBank gives Services across the Zones for its Consumers.

Central Servers may be hosted on high-end servers to cater to the details mentioned above. Also in very large networks, there could be a cluster of these (synchronized) acting as regional hubs, for a cluster of Edge Servers.

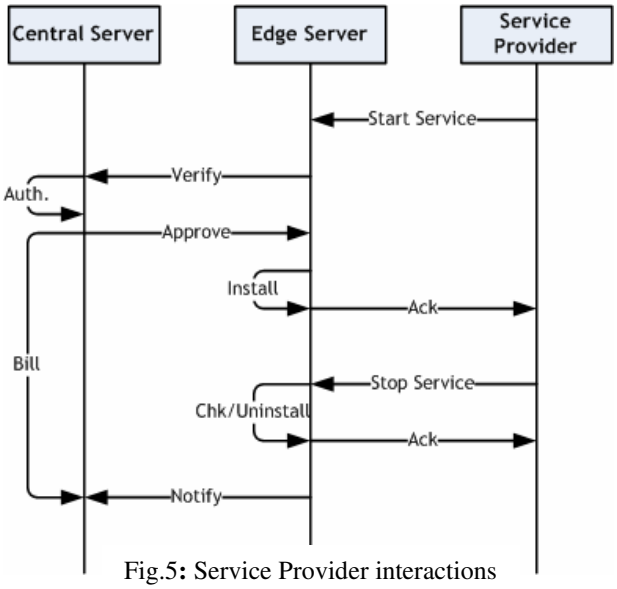

Fig.5**:** Service Provider interactions

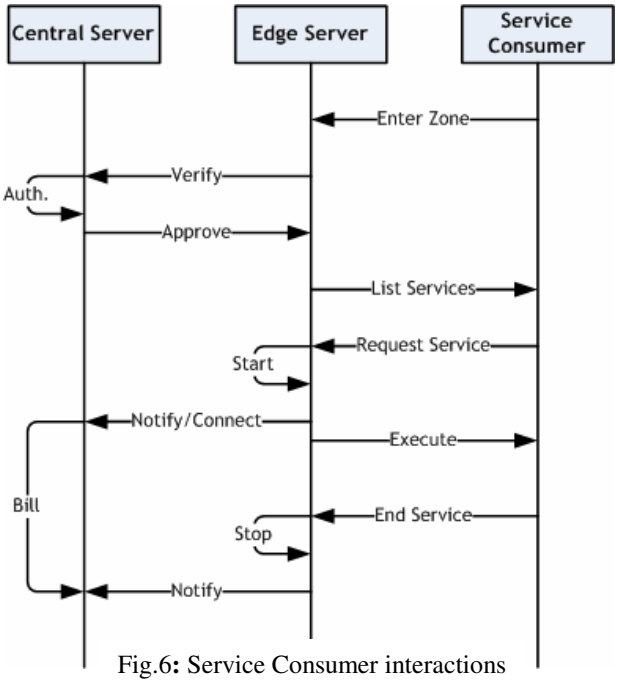

Fig.6**:** Service Consumer interactions

3.2 Edge Server

Edge Servers could be on mid-range servers located across different Zones. A Zone being a boundary within which it caters its deployed Services. These boundaries could depend on the technologies used (considering wireless), or at the discretion of the Network Operator in other cases. They are the Service Platforms providing the necessary D-SOA framework components to launch, pause, or remove Services dynamically.

In most cases, the Network Operator might choose to launch a few basic Services on this infrastructure directly, for voice communication, messaging, email, Web-access, etc. Service Providers in the Zone shall launch other Services dynamically, relating to their respective business, such as navigation, banking, restaurants, etc. In certain cases, Service Providers may deploy Services in other Zones too, in coordination with the Central Server.

Edge Servers host Services deployed on its Service Platform to all the Service Consumers within its Zone. Service consumption may be on-demand by a Consumer, or pushed sometimes, advertising for instance. Services are customized specifically depending on the Consumer's profile - Subscription, Preferences, etc. For interactions across Edge Servers, they coordinate with the respective Central Server, thus establishing required link.

These servers also monitor when a particular Service Consumer is within its Zone, or leaving (mobile devices), and run Services dynamically, based on this location information on the Service Platform. When the Service Consumer comes into a Zone, its signed personal identity bundle is obtained from the Central Server, and is

launched on the Service Platform. Based on the properties, respective Services its eligible for are exposed to this Service Consumer, on his client device.

The Consumer's bundle may incorporate his Personal info, and other data like To Do, Calendar, etc., which could use Service Platform's execution plane to run them, while results gets rendered on user's client device. Moreover, when a Consumer moves out of a Zone, its current state shall be informed to the Central Server, ready for the next requesting Edge Server depending on the Zone he might get into.

The Network Operator depending on the Subscriber's profile controls all interactions – Services launched by a Provider, or consumption by a Consumer. The Service Platform provides necessary operation syntax, and the semantics of the Service, normal as in any SOA.

To support access to a Service from the largest number of Consumers, a policy mechanism can be introduced. While the functional aspects are described in the Service interface, the orthogonal, non-functional capabilities and needs are specified using these policies. The formats for interactions on this platform need to be based on common open-standards, and hence XML. The data formats shall be based on the most commonly supported types, such as the MP3/AAC for audio, H.264 for video, SVG for graphics, etc. The data formats suggested need not be mandatory though.

This Service Platform shall take care of remote stop requests from respective Service Providers, and shall delete their bundles appropriately. And if there are any instances still consuming these Services, deletion shall be staggered until their completion, although no new instances would be created.

Fig.5 gives the sequence of interactions of the Service Provider with the system to start / stop its Services on the Edge Server. Fig.6 similarly gives the sequence of interactions with the system when a Service Consumer enters a Zone of his Edge Server, and requests / ends Services. The figures indicate the Central Server interaction for authentication and billing of Subscribers (Service Providers & Service Consumers).

Both these servers are access-agnostic, and support XML-over-HTTP for signaling, and the media/data transfer itself could be specific to its protocol.

3.3 Connectivity & interactions

Service Provider/Consumer to the Edge Server: D-SOA Service Platform deployed on the Edge Servers is reachable for the Service Providers/Consumers by appropriate forms of physical connectivity – wired or wireless. Service Providers could mostly be stationary entities, and hence their network link with the Edge Servers is a fixed broadband. Service Consumers could be fixed or mobile devices, and hence their network link with the Edge Servers is either fixed or mobile broadband. Fig.8 gives us an idea of this scenario.

Edge Servers offer a connectivity service across to its other peers - immediate or otherwise - as indicated in Fig.7. Central Servers provide the necessary information for an optimal hop to the Edge Servers on request. Edge Servers have a direct network link with their Central Server. D-SOA could be implemented the Web Services way with a Universal Description, Discovery and Integration (UDDI), although we look at Services with a direct interaction model here, to establish links between the Central Server, the Edge Servers and its Subscribers, with all of them binding to each other appropriately.

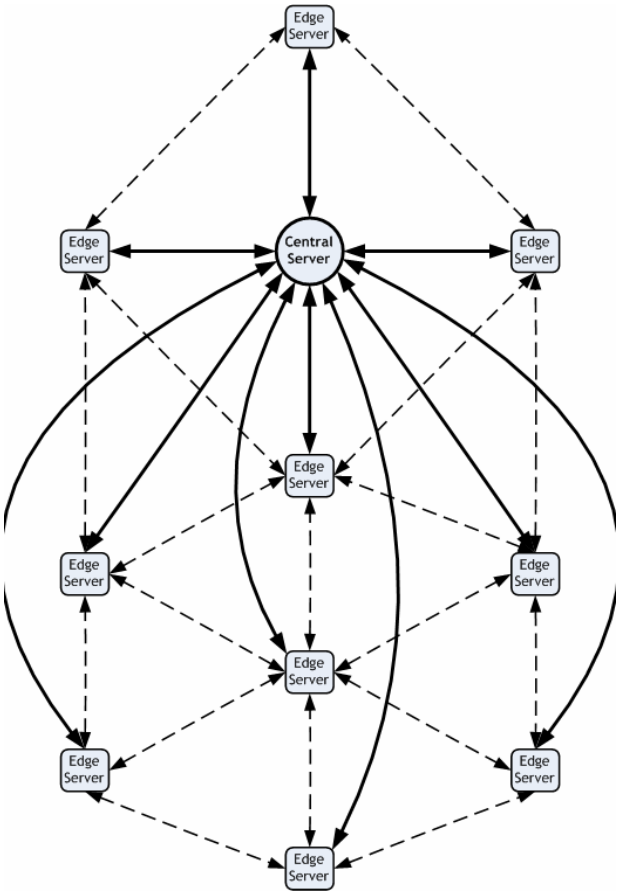

Fig.7**:** Interaction between the Servers

3.4 Roles & responsibilities

Fig.8 shows the typical objects in a D-SOA framework for deploying Services in a communication network, their relations and responsibilities. The Network Operator is responsible for all the server infrastructure, running required housekeeping Services for maintaining the Edge Server as a Service Platform for execution of the bundles

deployed on it. Network Operator may ideally take the responsibility of all basic communication services.

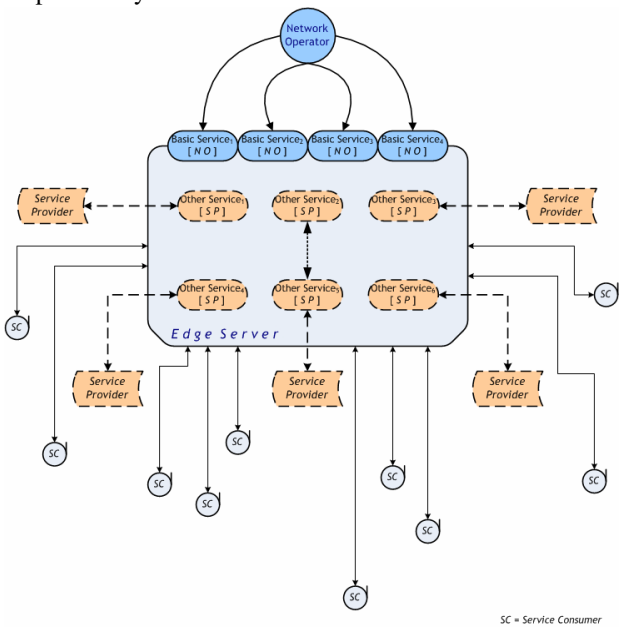

Fig.8**:** Services – provisioning & consumption

A Service Provider loads his Services, deploying them on the Edge Server, when required dynamically. It is the responsibility of this Service Platform to launch these, and exposing its respective characteristics - Visibility, Interaction, etc., and be accessible to all Service Consumers in the Zone. The Network Operator shall provide this well-controlled environment, and take responsibility of the framework. Depending on the respective Service contracts of the Service Provider/Consumer, the Service Description of the particular Service and the execution context, the various interactions are uniquely performed, which get monitored for billing purposes by the Network Operator, details logged on the Central Server's database.

Service consumption happens by the Consumer using Web-client devices that connect to the respective Edge Server, and access the Services. The Central Server is responsible for pushing any updates there on to the respective Edge Server if any, with respect to its specific Subscriber. In the case of an OSGi-based-SOA framework, these Services are the bundles that consist of Subscriber profile, downloaded on the Edge Server, which takes into consideration the Consumer's Personalization-info. E.g., bank details (bank name, login details, security key, etc.). All communication Services get represented as widgets on the Consumer's Web-console, which is updated dynamically in accordance with the Services being exposed by the Edge Server in that Zone. Whenever a Consumer requests for consumption using Service widgets, a specific instance of the Service is created for its execution. Consumer's profile bundles interact with other bundles offering Services, based on which appropriate actions are performed. E.g., we could envisage interesting scenarios - if on the Consumer's profile there is a "To Do" item to buy a particular item, and one of the Providers is selling this as part of its Service, the user gets alerted about it on his client device.

4. Other features

Some of the other features that can be envisaged with this architecture, apart from those described above are as follows:

4.1 C2C model

Consumer-to-Consumer features could be based on the respective Personalization settings, location, etc. of the respective Consumer, and derive benefit of the Zones in some situations. This approach could be a bit different from the regular Provider-to-Consumer approach, as here the Consumer isn't responsible for directly launching a Service on the Edge Servers at any time, but the network instead monitors the Active state of his Events, and helps giving a solution appropriate to his specific needs.

E.g., a person having movie tickets in a particular Zone is unable to make it to the theater, and in such a situation the network shall prompt him that there is a buyer waiting to purchase his tickets if he wants to sell them.

The scenario described above requires that the following steps be performed:

- Monitor all transactions of its Consumers, and its respective parameters such as Time, Location, Personalization-info, etc.
- Build a correlation with respect to the state of each of the Consumer's Events, such as those on "To Do", "Calendar", etc., (if made accessible)
- Prompt events in case there's an opportunity / failure (e.g. buy / sell tickets in the above case)
- Since the Consumers aren't launching Services, the Service Platform acts as a broker for this purpose

4.2 Network Intelligence

An aspect mentioned above can extend to all Subscribers (Service Providers & Consumers) in general. A system can constantly track their actions on the network, their behavior patterns and suggest appropriate solutions. This is a dynamic process and changes continuously.

E.g., the network understands its Consumer has purchased a flight ticket for the evening, but with the intelligence built into the platform could also inform the Consumer appropriately that there is a delay in this flight in an automated way, when the Service Provider i.e. the

airline company updates its Services on the network. The Platform updates all the affected Service Consumers in return, as it knows about the Consumers who had utilized the Services of this particular Provider prior.

4.3 Utility services

Apart from the specific D-SOA guidelines about the Service and interfaces, especially on the Internet it is difficult to impose restrictions with respect to other aspects such as the data formats, etc. Nevertheless, the Network Operator on the Edge Server can provide utility services to transcode data appropriately with respect to the client device specifications.

E.g., assuming a Consumer's client device supports only H.264 format of video, while the Service Provider has deployed a video Service using a different format. The Edge Server can detect the device characteristics and transcode it on the fly using these utility Services present on the Service Platform seamlessly.

4.4 Dynamic interfaces

The dynamic nature of Services launched and removed on the Edge Server needs to reflect seamlessly on the client device of the Service Consumer, to avoid wrong representation. An ideal technology for this purpose is Ajax (Asynchronous JavaScript and XML), in order to maintain constant synchronization between the Services provisioned and their widgets respectively.

5. Business models

There could be a number of different business models drawn above this D-SOA framework, depending on the specific markets addressed. And the range of Services that can be enabled on such a platform depends on the market targeted. Briefly touch upon from the Network Operator's point of view below.

5.1 Service Provider

Services can get deployed dynamically by its Service Providers and charged appropriately, based on its type, the amount of resources utilized and the amount of time. In some cases, the Network Operator could also demand a share in the revenue depending on the number of Consumers subscribing to this specific Service, and may not charge for the Service deployment itself on the Provider.

5.2 Service Consumer

The billing model can be pay-per-use, with a minimal fee to register for deploying these Services. Hence, the Network Operator monitors the specific Services used, and its durations, based on which the Consumer gets billed appropriately. Not all Services could have a flat rate, but depend on what exactly they offer, and come at different premiums.

6. Conclusions & Future Work

Following are a few advantages perceived over the currently existing communication models:

- Remove the scattered approach for the various forms of communication, and bring the focus on integrating all forms of communication and related Services on to a more economical and extensive platform
- Ever-increasing demand for communication Services and the low-cost of deployment should be beneficial for both the Service Provider and Consumer, and with the framework it supports them all dynamically, and pay appropriately. E.g., a store could advertise its Services only during the sale period of two weeks.
- Provides a simplistic way for a Provider to deploy their Services, much faster and yet be client-device independent. The clients only need to have a Web-client, perhaps with an appropriate browser.
- Correlating the profile, behavior patterns, etc. information of the various Subscribers, could open up newer opportunities for an operator
- Utility Services providing data conversion / transcoding utilities can prove to be very useful

- Define the functional components to give a wider acceptability for a range of communications, and their anticipated future patterns
- Explore possibilities of implementing this framework in a peer-to-peer way (like Skype), with an operator only taking the responsibility of the Central Server
- A layered plug-in architecture, with intricately designed Service interfaces, would also help build mashups, enabling newer, interesting Services from these collocated Services on an Edge Server.
- Multi-service bundles could collaborate across to provide richer solutions by adding applications, and extending these applications so that they operate across different services
- Explore the one-to-many communication scenario in greater detail

7. References

[1] W3C - http://www.w3.org/

[2] OSGi Technology - http://osgi.org/osgi_technology/

[3] OASIS SOA Reference Model TC (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm)