Universitat Autònoma de Barcelona

etse

CREATION OF A CARRIER-GRADE TELCO BASED ON A VOICE OVER IP BACKBONE

Memòria del Projecte
d'Enginyeria en Informàtica
realitzat per
Esteve Vilardell Cànovas
i dirigit per
Romualdo Moreno Ortiz
Bellaterra,. a 11 de Juny de 2008.

CERTIFICACIÓ DE DIRECCIÓ

El sotasignat, Romualdo Moreno Ortiz

Professor de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Esteve Vilardell Cànovas

I per tal que consti firma la present.

Signat: ....
Bellaterra, 9 de Juny de 2008

Preface

Evolution: the human race seems to follow the same pattern of evolving generation after generation, developing and increasing its knowledge about the planet they live in and share with others. At first sight, humans seem repeatedly to commit cyclic historic mistakes but luckily have also shown the willingness and capacity of eventually learning from those.

There is a ubiquitous sense that humanity as a whole constantly tries to come up with new ways of improving our lives and in this century, it seems to have chosen technology as the tool to achieve it. Examples such as the daily invention of new drugs to improve our life expectancy, the impulse in space exploration, down to the thousands of gadgets that have taken a place in our lives, it is quite clear that technology has come to stay and has taken the driver's seat in the early 21st century that we are going through.

It also seems to be a fact that the greatest technological inventions have been either rushed or developed mainly driven by either social or financial needs. In the capitalist economy world that we live in there are thousands of thousands of companies and each single company tries to compete with and be more competitive than its peers. This triggers off invention and creativity and, so far, it seems to bide well with the rate of innovation that one would expect in a developed society.

In this project compilation, the author shall introduce his approach to implementing a Telephone and Advanced Services Carrier designed from scratch using early 21st century technology. The design, as intrinsically very generic, essentially designed with the commercial world in mind, could also be used to support any kind of upcoming network because that is what the author's main aim is as well, to design something capable of extending, expanding and keep up well with scalable and complex scenarios.

Worth saying that we are living in an exciting moment of history where technology brings people closer and where communication is king. In less than 20 years, people have advanced from writing and exchanging thoughts by paper mail to massively being able to see each other's in real time regardless of the distance using mobile phones. Developed and developing countries are somehow closer now than a few decades ago thanks to the availability of technology and it is the author's hope that the latter will eventually get even nearer as communication among humans improves. Let us hope that humans have learnt from past mistakes and this time they use technology for good rather than for evil purposes; our children and generations to come deserve no less.

Esteve Vilardell
Barcelona, May 2008

Table of contents

1 Introduction

1.1 Project Goal – General Overview

The purpose of this document is to gather, clearly identify and implement the technical and business requirements for the implementation of a Telecommunications Services Framework based on Voice Over IP – VoIP - technologies. In a 21st century environment, legacy Operating System and Support applications are quickly becoming obsolete and a need for a from-the-scratch design arises; consequently the author humbly intends to come up with new ideas for a smoother and modular replacement for old technologies currently in use in big Telecommunications companies worldwide.

The basic idea is to allow a company to align against bigger vendors by means of using a more scalable and affordable solution to run their voice, data and third-generation telecommunication services. With the current state-of-the-art technology this is achievable and the general idea outlined here is that it needs to be done to allow for the development of further ideas in the Telecommunication market. Scalable and affordable systems translate into more market players, which in turn, lead to further technological development and major options for the end-user of telecommunication services. Throughout this document the system shall be presented as it is designed and implemented.

Designing an application framework that will take care of providing telecommunication services for a company is this project goal. We will have achieved this target once we are in a position for using our framework to deploy new services in a fast and market-competing way. During the duration of this thesis, the author will outline and implement the functionality needed for this framework to allow for services to run on “top” of it. In other words, a successful completion of this work shall end with a framework capable of supporting the deployment of new telecommunication services running under its supervision in a distributed controlled-manner.

The power of achieving this goal shall enable a company to invent, develop and commercially launch new products for its customers. Initially these products might consist of voice, data and collaboration applications but they are extendable to every kind of application running on the Internet.

1.2 Introduction to the Project State of the Art

1.2.1 Operating Systems and Support (OSS)

From an engineering point of view, each Telecommunications company network can be regarded as the sum of entities interconnected and forming some kind of topology where there is a structure of nodes allowing information to flow from one edge of the network and reaching its destination by following a given path. This is the definition that a Network Designer would take into consideration when overseeing the issues concerned with implementing and running a communications network. As the first layer of abstraction, this is a fair-enough approach but one quickly realizes that it falls short as more and more pieces are introduced into the network. A network alone is not very useful about much as having a car without a steering wheel is.

The analogy of someone driving or controlling the entity is what we are trying here to convey to the reader. Regardless of the network size, without any kind of network intelligence, it is of very little use; exactly like having a fast car without the proper controls to drive it properly to the millimetre is not. Networks quickly tend to grow and extend over buildings, neighbourhoods; cities, countries and nowadays the big Telecommunications companies’ backbones can have nodes in different countries and territories and those can be made up of all kinds of physical links such as cable, fibre optic, and microwave links and so on.

It becomes quite natural that the Network Managers should need to have tools to properly being able to find out the network state in real time as well as the state of all the systems that form its parts. From simple alarm modules indicating that a node is down, a link on temporary offline state, or that a backup link has been activated to fulfil additional circuit capacity; those are only a few real world requirements and needs that current network carriers are constantly watching and provisioning, and trying to achieve.

What is even more important, the systems outlined are not only vital for managing the network and making sure that its entity is controlled and its status known at all times in order to assure a proper behaviour and availability; but those systems turn out to be the key piece for the business side of the company as well. They are the ones that will be used to define which services the network will implement to fulfil the business needs of coming up with new competitive services which customers will choose in favour of other companies' systems. In other words, they will make a distinctive point on the business side of the network.

If we recognize that almost everyone these days has a telephone, uses the *Internet*, and many carry a mobile gadget keeping track of their contact, personal information, schedules and allowing that person to reach and be reached at all times, one quickly concludes that there is a massive amount of information that travels and is exchanged at all times. Something really scary and daunting is that having no common reporting, controlling or in general interacting mechanism, one hits the wall and realizes that the need for a OSS is a requirement instead of just something 'nice-to-have'. Applying an engineering approach, the author performed an initial assessment of the complexity and resources needed to plug the network systems into the 'intelligence' layer wanted on the 'business side' and came up with a nice synergy as a result of elaborating and merging the two needs.

1.3 The Basic Foundations

Thus, what is an OSS? What exactly does OSS stand for and why is so important and present in every single telephone operator worldwide?

Having an airport without a control system would turn out easily and quickly to be chaotic and disastrous. The same goes for having a company unable to charge its customers for the services it provides. Even worse could be to bill improperly, inaccurately or even out of time as nowadays time is a competitive factor and a company's survival many times depends on its capacity to not only market its services but to properly bill them and provisioning them in a timely manner. Furthermore, errors generated in customer invoices and bills create a very negative company image and therefore could cause a big downturn in revenue if not carefully considered and planned.

Information and service management is therefore vital for a Telecommunication company's survival. An OSS is the system that carriers use to cope with this. The acronym stands for ***O*perating *S*ystems and *S*upport** and, from an engineering or software design point; it is the core or engine that keeps track of the user's activities and requests. OSS systems are usually found in the Network Core and perform the function of the main decision and billing engines. OSS systems can be found in any telephone carrier, ISP – Internet Service Provider -, Mobile Operator, Satellite Operator and in general in any Telco company providing services to customers and later having to bill for those.

In a real world scenario, the difference between very large established Telecommunications companies like Telefónica, MCI, British Telecom, Telstra, and relatively small ones, spin-offs, start-ups or the new Internet Telephony companies lies in the OSS systems. Whereas the former have the procedures and OSS systems in place to provision systems and services, a company candidate to become or just provide basic phone service will face the challenge of starting up from scratch, as they will not have any OSS system in place. This can make the difference between acting in a reliable way and providing a fast service to its customers or having primitive procedures in place that will lead to failure and being unable to keep up with the provisioning, technical support and service level agreements that customers would expect. Or in plain worlds: again, a differentiated factor or point that marks the line between being a good carrier or failing as the speed of reaction in the market can be too fast for the company to keep up with.

As in everyday decision, managers of a carrier have to face issues that affect their companies regularly: logistics and proper customer care is by definition essential to leading a company to success. Because of this worldwide CTO – Chief Technical Officers – know that a company's internal procedures cannot be left to pure chance and consequently have to be taken care of until they are properly tuned and looked after. This is just an engineering approach to resolving a purely business problem: using technology to help optimise the business procedures as the main goal to achieve.

In our telecom-like scenario, we have a heterogeneous number of systems in place and our engineers or people in charge of monitoring them take good care of that, but those highly coupled systems need to be associated with the human business side as well. That is fundamental as the capability of quickly accessing a customer's profile in order to market, sell and provision services to the customers will eventually define the degree of customer satisfaction and this information needs to be known at each moment in order to take business decisions in an accurate and prompt way.

1.4 The State of the Art: Reuse versus Redesign

Historically the big carriers developed some guidelines or approaches to assist them in this endeavor and they are currently still in production in thousands of networks of the 20th and 21st networks.

Old-established Telco started development on their OSS mainly due to their internal procedures requiring them to automate customer management tasks, service provisioning and mainly billing tasks. Above all, their OSS adapted to their needs and requirements and not the opposite. Their software was mainly developed in-house or hired to other software houses, but their requirements were always quite clear and the result pieces in most of the cases adapted to their needs fairly enough. The main problem with this approach is that those same big telephone companies ended up as monopolies and with OSS software so big and coupled that in the end they have become highly time-consuming when it comes to the system maintenance and new functionality additions.

Classic ERP - Enterprise Resource Planning -- packages traditionally implement the basis tools to handle a company's main needs: accounting, sales management, and human resources. Nevertheless, those are too global or generic and that is why there is always some effort needed to customize and adapt these ERP to the exact company's needs in accordance with their exact requirements and specifications.

Through adapting or customizing such packages to Telco needs, some companies base their OSS infrastructure on global or generic third-party software packages. New carriers therefore need to opt for using commercially available ERP, CRM – Customer Relationship Management – software or take the decision of putting a development team together and create a new application to support their operations exactly in the way they want to: i.e. being faithful and adapting to their narrowed-down specifications to the tiniest detail or using commercial packages that will need months of tuning to adapt to the Telco companies´ systems and procedures.

The second phase or wave in the OSS field is that software vendors started to focus specifically on telecommunications needs and adapted their software development methods to make software specifically suited to addressing the telephone companies' needs. Quite a few vendors have appeared in recent years, especially during the major explosion of the Internet (at its peak around 1994-99s, the magic days of Silicon Valley where every company on the block was doing software for either the Internet or for Telecoms).

These days a wide variety of commercial packages exists but its main downside is that either it consists of software pieces so complex and large that a very specialized knowledge is expected from the system implementers or that the very customization and adaptation to the telecommunications requirement can be very demanding in human resources and time. Even so, this is nowadays the pattern being followed by the big Telecommunication Companies as they are aware of the importance of unifying systems and make them more 'open' so that whichever new services they market, their OSS platform will be adaptable and cope with them.

Another motivating factor to make systems more open is that they will adapt more easily and integrate better with other companies' systems. As there seems to be a tendency for telephone and especially on mobile operators to acquire and merge with competitors to gain market share, it makes sense to take this very important factor into consideration and planning all the internal procedures accordingly.

This creates the debate of either engaging in a very new software development effort or delegating the main core of the business to vendors' software designed to perform this function. Whereas in the first case the task involves a much longer and steep effort in time, the former has the risk of failing if the acquired software does not perform well when implanted on the carrier's services management processes.

Now, we shall focus on what is our current study context within this project: a Telecommunications Company – or Telco - , defined as an entity that sells telephone services to customers that make calls using different devices, access methods and technology. That would be the very basis of defining what we need.

Those services can expand to multimedia-rich data, over the Internet new applications and so on, but mainly the foundations or common ground would be that users will use those services and that a certain company needs to charge certain defined amount of money for them. In other words, we will introduce a complete design of an OSS created from zero and targeted to bide well with the business, network, and customer's requirements.

1.4.1 AS-IS Situation

The state-of-the-art in voice gateways, network routers, network access switching technologies and related networking and hardware has evolved exponentially in the last decade; a factor which has helped and contributed to the apparition of the tools used by the author to implement his system.

Nevertheless, this is not necessarily good news as the core network systems have indeed come down in its cost and allow the author to deploy his Telco-grade voice-core system in place, but without the intelligence needed to coordinate these systems, the former effort can end up in nothing if no OSS system is behind the scene to back it up.

Bottom line here: it does not matter how good the routers, transmission technology, and even voice quality are if later, that service cannot be billed and reported in a very professional and outlined well as needed. In this very one sentence, we have summarized the very reason why this project is about OSS and why they are so important and essential for a carrier; this is so because customers do not only want a service but also a good quality of service and efficient and accurate billing. If a carrier fails in this department, it will collapse from a business side very shortly afterwards.

This is exactly the need the author had when implementing a system capable of sending voice from one place to another. After setting up the network, putting the appropriate quality of service measures in place to make the voice really 'travel' over the wires so that everything goes all right, then the other part of the puzzle was missing: the network intelligence to track down services, network usage and service billing.

That is why after establishing the necessary peering agreements with other carriers to exchange voice, the author quickly realized that a system could be really neat and sleek on the function it provides, but it lacks usability if it does not rely on a smart system controlling its functions down to the single second.

In other words, and even plainer, the analogy would be like having thousands of cars in a city without any traffic lights to control the traffic and no measures to limit or stop the cars if necessary. In this project's scenario, the analogy translates into having a good network capable of routing phone calls over IP but unable to perform any intelligence on these calls due to the lack of a system controlling what is flowing on the network.

A system, which should distinguish and keep track of the services that we are deploying for our users and that we want to be able to bill for, essentially that is what we need to achieve: to have a real system that we can exploit and get some revenue from.

Companies are founded to survive and in that context, survival consists of generating revenue from either products or services sold to people. It is in that context where the natural need of granularly knowledge and billing for services arises. Customers are used to being billed for the services they purchase and in most cases they are used to being billed in an accurate and elegant way as they have traditionally been by the major carriers which slowly but surely created their billing and operating support systems from scratch to support their business. In an IP-convergence world, this requirement has not changed at all, only the underlying voice-carrying technology has changed, moving from TDM to VoIP but that being no reason whatsoever not to present the customer with the same robust and tools they grew up used to when dealing with their traditional telephone service carriers.

In the next following chapters the author shall describe the implementation of the hardware and network platform running the voice-core of the Telco-kind of company and how he came up with such implementation using a 21st century approach and using the technology plus knowledge synergy to put in place a reliable, mission critical voice-core.

1.5 Introduction to a new Design Candidate

This chapter is about dealing with a complex system used to bill mainly telephone calls at this stage and capable of progressing to billing further telecommunication services. We shall walk through the next pages developing an understandable and appealing reading introducing the development as the core functionality of this dissertation as a small contribution to the world of communications.

In this thesis compendium, the author has developed his own OSS design and implementation, finishing it successfully and installing it as a production system as from early March 2005. As expressed in the previous chapters, this project was about defining a new class of Carrier not using the former PSTN – Public Switched Telephone Network – technology and implementing a conventional carrier relying on the new IP convergence paradigm instead. In the next chapters, the author shall outline the steps followed to implement such a system from the network and hardware approach. Now, we will proceed to outline and go into some detail on the design and implementation of the OSS developed for such system.

Many of the most successful current OSS architectures started as humble prototypes fulfilling urgent or important requirements that needed to be addressed. In this project, the design will begin analyzing and trying to address the imminent needs that came up when the network elements and basic Voice over IP and TDM circuits were put in place and operations started. Logically, the design began earlier on in order to hit the right timing with regards to the commercial launch so that all the pieces were assembled and ready to start at the given launch date. We will now proceed to outline and go through the design in a detailed way to introduce our suggested design work.

1.5.1 Design Ideas and Goals

Our main aim when designing an OSS platform from scratch is to address our business needs and make sure that those are perfectly implemented and in harmony with the services that we want to develop and market. Rather than relying on existing commercial packages, which do not suit our needs perfectly, we have opted to develop our own solution therefore making sure that at least our basic needs shall be covered in detail. Besides, controlling our own processes shall allow us to customize and adapt our resources to maximize our time in developing services we know that we shall be able to implement and launch quickly for our customers. When compiling our first requirements in the form of business ideas, we made emphasis on some basic points that became the main requirements; those may be outlined as follows:

- Business reaction is essential in the competitive Telecommunications field that we aim to work in; therefore the more agile, dynamic and flexible the OSS are, the better likelihood that the company will benefit from the advantages that these characteristics can yield. Thus, the system has to be flexible and scalable. Flexible in the sense that it needs to have an open architecture allowing to grow and implement future requirements, it also means that ideally it should be implemented using a standard or open technology that accepts interaction with other third-party modules and allows for integration with external systems. The latter also bides well to cover the need for scalability.

- The success of a business is the combination of service Provisioning + Reliability + Billing capacity. We shall act accordingly to fulfil these three properties, bearing them in mind as we come up with a design satisfying them as our primary target. We will additionally design the system to be open and scalable so that undetermined characteristics at this time can become part of the system and introduced as the need for those arises.

- No billing, no revenue: Even with a perfect state-of-the-art, network in place, it becomes ubiquitous and extremely obvious that without billing systems in operation, there is little use and benefit in commercially exploiting it, hence billing shall be our priority one and main requirement to implement.

- Manual processes are obsolete and should be avoided. Elimination of these in regards to the managing of our basic processes must be taken as an important milestone from the very beginning as the intrinsic slowness associated with them belongs to old-fashioned deprecated business processes.

- Detailed knowledge about our users: in order to serve our customers better, we need to analyse proactively their behaviour, needs, and service feedback. Our customers are our first priority and we have to reflect this in our Customer Relationship Management processes so that we know as much as we can inside the system about our customers, hence foreseeing their needs and fixing their possible issues even before they let us know. Proactively tools are therefore what we are highlighting at this point; those ranging from, for example, ways to monitor the network voice quality up to being able to narrow down a given service problem to a single user to make that user feel recognized and satisfactorily served.

- Security: A global network can be difficult to defend against unwanted integrity or fraud attacks, especially when dealing with voice calls that tend to be more appealing to the fraudsters. Strict security policies have to be defined and the system has to be capable of checking them in order to avoid customer abuse or system security breaches; therefore user authentication and tracking shall be one of the points to bear in mind as well when designing our system. Aside of using customer's information to serve better our customers, this may also be used to make sure no attacks on the network integrity succeeds. Fraud is a big deal on any business and especially on a Telecommunications Network where these networks are reachable and available for any single user with an Internet connection. A diligent policy and control systems need to be in place to oversee the good behaviour of all the users and network pieces involved.

1.6 BARA: Billing, Authentication and Routing Architecture

Based on the previous requirements and concluded premises we can conclude that a first candidate system should initially address our main concerns or priorities, these being: Billing, Authentication, and Routing. As we humbly intend to make our design not only useful to be used an stand-alone application but also to be referred as something more scalable and versatile, we shall call it Architecture to reflect our humble goal of ending up with something good, reliable and that allows for further modules and services to be written and built on top. This has been our aim from the beginning and will now elaborate the details of our candidate design.

Hence, these are our main basic modules that need to be granularly defined:

- **Billing:** This is a commercial architecture and therefore it aims to be used to run the basic needs of a Telecommunications company, and let us face it: these are: to earn money thanks to running a network by exploiting the services running on top of it. In order to get some revenue from these services, a Carrier – a Telecommunications Company – will present an invoice, bill or summary to the customer to charge this for the given service or like classically done, to bill the customer at the end of the month for the services and usage that this has consumed on them.

- **Authentication:** Multiple aggregations of services and availability of different commercial services via market packages means that there will be diverse profiles for users and that the relationship between a user and our company's services can be as simple as 1 to 1 or go to a user using a large number of those: 1 to N relationship. Practically, this translates into the fact that we should know for each user which services he/she has access to – has subscribed, signed up and so on, its profile quota limits for such services, his/her usage limits, whether the customer will be in a *prepaid* or *post-paid* service plan and many more parameters like attributes specifying how will the user be billed, his/her associated billing rate in order to know if the rating mechanism should apply different SLA – Service Level Agreements – such as special voice quality assurances, call or data-usage volume discounts and so on.

  There will exist therefore a repository with our customers' attributes in the form of service and authentication profiles that will allow us to uniquely and exhaustively deal with our users and bring along fraud detection mechanisms in case of some security policies being triggered.

- **Routing:** a Telecommunications Network is intrinsically a mission critical system; this defined as an entity that has to be available to its customer 24 hours a day, 365 days a year with a 99.99999% uptime value. No one expects a telephone line to be picked up and get no dial tone as well as humans has been used to use phones in a very reliable way and which they can rely on even for emergency situations like calling emergency services like the Police, Fire Department and urgent vital scenarios. We did grow up used to this kind of always-on reliable phone Networks and this has been the de-facto response ways of these networks up to now. Anecdotically therefore it is the recent reality where many start-ups companies have put in place communication networks which apparently provides a good replacement for the classical phone networks. Problem with these though is that even if they seem to be good replacements, they fail in the most needed cases, as in many cases the new carriers have no contingency or alternative routing for its network in place in case one of their links or systems go down.

In a technical way: these new networks implemented by eager-carriers in some cases do not have any kind of Fault-tolerance or HA – High Availability – system in place. Initially they launch their services and try to cope with the customer's load and network components longevity but as systems always fail, they eventually hit a moment in their operations where a vital system goes offline leaving the whole network unavailable for its customers. Again, in order to highlight the importance of this fact, this translates into users not being able to make any phone calls, not being able to reach the most vital emergency services and basically left stranded by the company they have subscribed their phone service with. This fatal consequence is the cause also eventually that the new carrier loose its customers base and can not survive in the competitive Telecommunications market as its reputation receives a very bad visibility when these service-interruption occurs. Bottom line: no user being left without service for many hours will have the same good perception of his/her company any more and reputation is a property difficult to gain and very easy to lose. We are sure that the company revenue would be hit terrifically after each no-service event.

1.6.1 BARA Must-Have Characteristics

Because of the reasons outlined we shall design an architecture with support for fail-over, fault-tolerance and with high-availability systems and policies in mind. Consequences of this are going to be that our design will allow for Multi-Carrier, Multi-route and route-failure detection mechanisms and recovery mechanisms. By implementing network peering with other carriers, we make sure that in case of our network routes being down; the system can switch and use the other carriers as alternate routes.

By choosing a Multi-route policy meaning that we will provision different routes for each required path in order to make sure we can always choose the most optimal one in terms of voice-quality, data packet latency or any other criteria that we might want to evaluate, we shall make sure that the failure of a network component or route does not bring our whole service or network down and affect therefore our customers.

This approach shall also become quite handy to implement a Service Level Agreement association with our customers being able to let a customer choose his/her desired level of service quality by means of charging them more or less according to his/her service requirements. Thus, a customer wanting a 90% of perfect voice quality with a foreign and distant country where communications are scarce and poor, might be charged premium amount if he is ready to do so, but our system will support such requirements and route this customer’s voice calls using the associated link.

Summing up, our OSS Platform will have the functions and it will implement high-availability in order to assure an almost mission-critical-like network uptime, and shall implement SLA as well to opt for different services qualities according to the marketing department definitions outlined in the company’s different services available to the customers. Different services can therefore have acute service quality characteristics and shall be billed different according to those.

Finally yet importantly, our OSS has the added word ‘architecture’ added to its acronym (BARA). We will humbly try to deserve such an adjective and have the software cycles deep inside our designers’ mind when approaching our design with openness and easiness to integrate system methodology. We want something easy, pluggable with other modules, which scales well and that provides public interfaces so that other third-party system can eventually integrate and easily couple with our managing system. In the pages to come we will traverse through some design figures which outline the main conceptual design blocks for such a candidate architecture.

An OSS architecture has to deal and manage service-level requirements as well as the pure information and transport side of the network picture, yet we shall initially focus on the Voice or Phone Calls handling and management in order to introduce our design details and platform specifics. The Network in place, which we shall introduce in the next chapters, mainly deals with voice calls flowing through the network, like any traditional Telco in the past. Even though this revenue source is showing some lose of strength as data services are taking over and replacing voice as the main revenue-generating origin, we have begun with an IP-based Network carrying Telephone and Intelligent Voice-Services using Voice Over IP technology; hence we shall now proceed to introduce a suitable design to deal with voice billing, routing and processing.

1.6.2 Service-Level Behavior

As laid down, we now want to focus on the way Voice Communications shall be flowing and controlled in our network. As far as we are concerned, we need to aim towards finding out what mechanisms and events affect us when dealing with this topic.

To bring this up and comment it with detail, we shall direct our attention focus now on the way Voice Communications are carried out on a Voice Network.

Phone calls, and voice communications in the old-fashion way, follow a call-establishment protocol in which the phone call or connection has to be initiated, established, and terminated. These three phases make up a normal telephone call and our OSS design shall introduce different modules to handle each one; therefore, during the call establishment of an in-progress call, the OSS shall participate authenticating, routing and lately billing for the call according to the rates and user profiles service-parameters.

For instance, the customer could have different prices depending on the time of the day that the call was made, discounts by voice minutes in form of phone-packages, off-peak and on-peak rates, between-friends, business-office programs and any other billing policy in place. These characteristics and flowchart, introduced in Figure 1-1, shall be looked upon in detail in the next chapters as we walk through our specific platform design and implementation. Most of our lower application layers systems shall cope with telephone-related network concepts. We shall also introduce these in the next chapters as we will need to know the associated present technologies used nowadays in telephone companies worldwide.

The OSS routing module will make sure the call is physically terminated by choosing the right network path in order to assure an either best effort or high reliable chance of call termination in order to extrapolate and drive the call to a good end. This module shall have communication and control over the network components in charge of sending the voice stream back and forth to the desired destination.

The routing module therefore shall have to be fault-tolerance in the sense that it shall have inherent capabilities to route a call using alternative paths other than the primary one with greater weight and decide to perform a fail over to a backup route in case of detecting problems on the main one.

In addition to the previous behaviour, the same logic can be used at any time as well to profile reasons rather than network fail-over issues. An example of this to have in mind is the possibility of a customer having a special minimum service-level agreement specifying a good voice quality or rate of failure calls, for instance. In these circumstances, the routing module will turn out to be decisive and shall be the driving engine.

Moreover, the service visibility highly depends on this module as if a phone call does not succeed and many routes fail at the same time, then it is up to the OSS Routing module to provide a good response to minimize this condition and hide that failure to the end-customers.

For these, the network should always be up and running on a 24h, 365 days shift and any network failures should be invisible and recovered by the Carrier OSS. One more time we shall remember the important emphasis of this recovery mechanism for the good business-health condition of the Carrier.

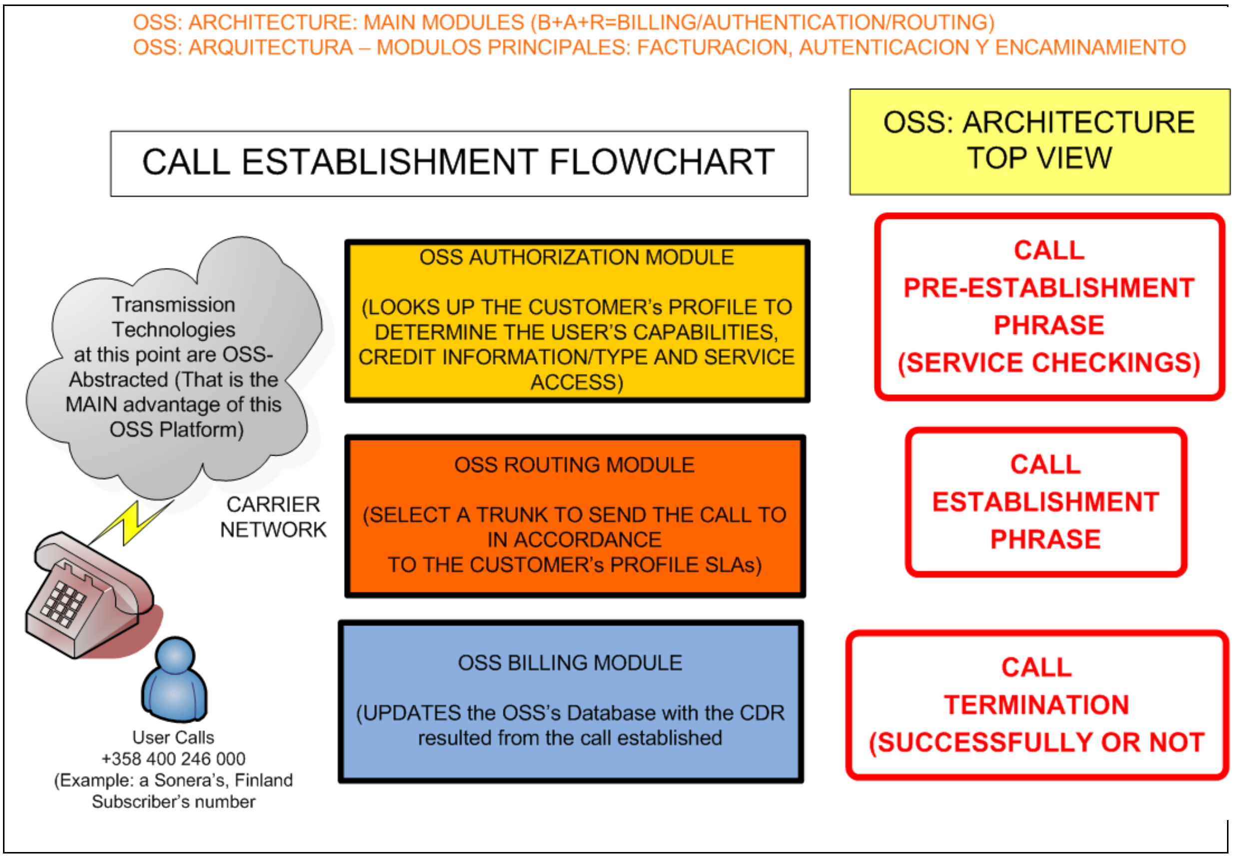


Figure 1-1: The OSS Architecture Basic Functions Diagram

2 The OSS Design: The System CORE functions

2.1 OSS Functions

As highlighted in the previous chapter, an OSS is the basis for the support of a viable system supporting a Telecommunications company infrastructure. Depicted earlier, we have enumerated and introduced the main modules of our OSS platform as: Billing, Authentication and Routing. In the next sections we shall get ourselves deeper into the development and come up with the requirement specification of these functions in order to achieve a clear understanding and define in a clear way the features that we are going to see implemented.

2.2 BARA Modules Definition

We proceed to define further, in a concise and more detailed form, the main modules as we introduced them in the previous chapter in Figure 1-1, which are summarized and enumerated as follows:

2.2.1 Call Establishment Phases

PRE-ESTABLISHMENT PHASE
AUTHENTICATION MODULE Module in charge of assessing the customer permissions: whether the user is allowed to use a prepaid or post-paid calling plan, how the user is going to be billed exactly, the exact billing rate, the units of time for which such user shall be charged and a comprehensive list of destinations that the user can dial out or receive calls from.

CALL-ESTABLISHMENT PHASE
ROUTING MODULE Module responsible of terminating a telephone call or data connection. It shall implement every fail-over policy and mechanisms to avoid system failure when initiating and/or terminating phone calls. It is also the one in charge to control the peering agreements as well as the Interconnection agreement with other Carriers. Lastly but not least important it communicates with the authentication user information module to grant the associated level of service for a given customer.

POST-ESTABLISHMENT PHASE
BILLING MODULE

Service-usage information as per user shall be granularly and exhaustively controlled by the billing module. Either in the prepaid or post-paid scenario, this module will keep track of all the customer's actions in regards to number of calls made, duration, billing plan and so on. In fact, the CDR or – Call Data Record -, containing the time and associated information for each system call, becomes the basic billing unit in any existing Telecommunications systems up to date.

2.2.2 Telco Nomenclature and Definitions

The majority of billing engines used in the telco environment define a billing canonical unit of usage information when dealing with phone calls or voice communications. For historical compatibility and also because it follows a good design pattern and encapsulates the basic usage information into a single unit object, we shall carry on basing most of our processes on this unit. Figure 2-1 shows a typical unit of *billing*.

OSS: BILLING FUNCTIONS: CDR – CALL DATA RECORD – BASIC CALL REPORTING UNIT
OSS: MODULO DE FACTURACION – UNIDAD BASICA DE REPORTE DE LLAMADAS

OSS: BILLING MODULE

CALL DATA RECORD (CDR)

IT IS THE PIECE THAT DEFINES THE FOUNDATION OF THE OSS BILLING MODULE AS IT KEEPS TRACK OF ALL THE TRANSACTIONS AND ACTORS OF OPERATIONS HAVING TO DO WITH CALL GENERATION.

Telco (Actors) Nomenclature and Definitions.

OSS CDR Schema

CDR (Main Schema) [Abridged]	
callid: int	did: string
cardnumber: string	didrate: decimal
parentaccount: string	effectiverate: decimal
callednumber: string	billcurrentrate: decimal
destination: string	trunkcurrentrate:decimal
callstart : date	billminsecs: int
durationseconds: int	trunkminsecs: int
cost: decimal	billincr: int
costpermin: decimal	trunkincr: int
application: string	billroundup: int
callerid: string	trunkroundup: int
trunk: string	timeinsystem: int
disposition: string	billroutelocalparms: boolean
billseconds: int	trunkroutelocalparms: boolean
trunkbillseconds: int	...
realcallseconds: int	...
billcost: decimal	...
netcost: decimal	...
billrate: string	...
dialedtime: int	...

Rate:

Voice is a revenue-generating commodity. As such it can be sold and therefore billed as a service. Also as commonly known, a telephone call can be invoiced in different ways in accordance with usage criteria such as call destination, call duration, carrier used to connect, time of the day, customer's assigned service level agreement, customer's discount or minutes plan, et cetera. A rate instantiates all these attributes into a global entity which is assigned to a customer's profile. Different rates imply different prices and levels of discount. Prices can be expressed in international currencies.

A Rate is the MOST IMPORTANT minimal entity in a BILLING system, hence the importance of a good OSS handling it in a correct, structured and dynamic way.

Figure 2-1: Billing Module, CDR Concept

In Figure 2-1, we observe that many attributes are committed into our billing systems storing on-call information such as duration, start date and time, finishing time, number of seconds for the call, billable seconds versus real call duration, billable cost to the user, net cost indicating the real cost for the carrier routing the call, the name of the routing carriers or internal routes used to terminate the call, the application that handled the call, and the disposition of such call – i.e.: connected, no answer, no route found, invalid number, etc.

Notably, we notice that some attributes exist specifying the billing increments for the call, that being the number of minimum seconds for which a given call shall be billed, or the number of units where the call shall be trimmed and consequently its cost calculated and that this information is available to assess the customer billable value, but also for the peers or carriers used to route that call.

The above means that not only our billing engine shall calculate costs and rates for the end-user but it is also in charge of doing so for our relationship with third-party systems used by us to interconnect and to hand them over some voice traffic. In that sense, our billing module becomes not only important in terms of billing our customers but it helps out and it will account for the money owed and earned by means of our interconnection agreements with other carriers. Knowing in real-time how much of our revenue is net and how much we own, our peers shall result in better synergies and efficiencies in regards to the decision taking processes.

Figure 2-1 also highlights an important definition is the concept of ***Rate*****,** this being defined as a straightforward interpolation of what we normally understand as rate in the colloquial language. A rate shall specify how much a service is. In other words and now coming to the voice or phone call specifics, each phone call can be associated with a cost in unitary units.

This cost shall depend on different factors and variables such as time of the day, duration of the call and so on, these terms already outlined earlier on, but also the call cost shall deeply depend on other non-call related attributes such as pure-marketing criteria such as special offers or number of minutes bundled in a service package, special offers for certain routes, discounts on a certain number of specific destinations and so on.

The business department shall define this policy-driven and random value and from a system point of view; this cost shall be evaluated as any other function cost. Once again, the main efficiency of having a rate object as an entity in our OSS is that this shall not only be used for billing the end-customer but also shall ease the tasks of calculating our interconnection cost with others. The latter property needs to be stressed to its highest mark as it can create the difference between a carrier succeeding due to its internal processes being reliable and fast versus using monolithic billing software unable to provide information in an acceptable time scale on the 21st century.

Internally our OSS shall contain a great number of objects to implement our outlined design. These objects or schemas shall relate to the Actors, or entities participant in our system. These entities shall be the customers, the interconnecting peers and any other party or system for which we shall exchange service information.

OSS: BILLING FUNCTIONS: CDR – CALL DATA RECORD – BASIC CALL REPORTING UNIT
OSS: MODULO DE FACTURACION – UNIDAD BASICA DE REPORTE DE LLAMADAS

OSS: BILLING MODULE

CALL DATA RECORD (CDR)

Telco Nomenclature and Definitions.

Trunk:

A Telco system is not globally capable in the sense that it needs to interact with other Carriers and Telco companies to terminate phone calls outside its network coverage. Practically, this means that when routing calls outside it might need another company to help out and share the international leg of the call. A trunk is defined as this entity and is normally a carrier for which a call exchange agreement has been negotiated. As in any commodities market, electronic bourses and universal agreements for dealing with call termination exist; otherwise one to one conversations need to be arranged to close traffic exchange agreements. A typical example could be Telefonica in Spain trying to establish a phone call with a customer using Sonera in Finland. In this typical scenario both carriers can either exchange the call in a previously agreed node or route the call thru other third-party carriers using alternative ways. The routing and its associated technical parameters such as call establishing time, voice quality and latency will influence and determine the final real net cost of the call.

Net Cost:

The real cost of a call. The technical cost of providing a point to point voice conversation. This includes all kind of taxes, cost of international links, cost of volume discounts, cost of negotiated agreements and so on. In brief: A Carrier routing at the net cost would have no benefits at all when charging for the service.

Figure 2-2: Telco OSS Nomenclature, I.

By all accounts, there is a general field agreement on the nomenclature and terms using defining OSS and general Telecommunications concepts. Since this asserts true, we can look up some more definitions on Figure 2-2 as our OSS design relies on these terms to base its functionality. Our system design shall outline the basis to deal with other carriers and encapsulating such behaviour in the form of different trunk routes for a given traffic to be forwarded throughout these trunks either on demand or on a regular basis following the defined system policies for routing. Practically, this shall mean that the OSS shall calculate the cost of these transactions all the times as well as the cost to be charged to the user.

Net cost shall apply for infrastructure cost and for calculating our cost when dealing with external entities, whereas billed costs shall translate into units charged to a customer, given his/her profile and applicable rate, and that shall be presented to the customer later on either electronically via web, sending it to his/her PDA, or any other form of electronic presentation, or classically and by default sending him/her a usage invoice at the end of each billing period.

Figure 2-3 introduces the concept of DID or *Direct Inward Dialling* node as still one more parameter to assess when calculating phone calls costs.

Traversing this design, we come across more and more functional requirements to consider on the billing process. We shall observe as we keep on walking through the next pages that the billing engine shall control and handle many different variables and subsequent policies at a given time, thus its intrinsic nature might turn into its main strength if designed and implemented coherently to our portrayed criteria.

OSS: BILLING FUNCTIONS: CDR – CALL DATA RECORD – BASIC CALL REPORTING UNIT
OSS: MODULO DE FACTURACION – UNIDAD BASICA DE REPORTE DE LLAMADAS

OSS: BILLING MODULE

CALL DATA RECORD (CDR)

Telco Nomenclature and Definitions.

Billed Cost:

The arbitrary cost assigned by the Telco to charge for the service. This varies in accordance to the rate assigned to a customer or provider. Basically it can be seen as the price or cost that a customer has to assume to use the service. This is the final price of the call made by a customer and it depends on the associated customer rate as this will define the parameters to use when applying the calculations to bill the customer. This is normally much above the net cost of a call as this is where the Carrier's benefit is as they are selling a service consisting of the capability of making phone calls and charging for it following the open market laws.

DID:

Phone service can be provided by means of direct (a Telephone line directly available to the customer) or via Indirect Access using what is it referred as DID – Direct Inward Dialing – dialing numbers. A customer calling a free number from a telephone booth is the classical example of a DID, acting this as a way to bypass the dominant operator network and reach another carrier network. Customers can jump from one to another carrier by specifying dial-prefixes in order to choose the carrier they want to use when calling long distance. A DID would be the logical number associated to the customer. DIDs are important as they form the basic of what is referred as the Intelligent Network (IN) or 800/900 numbers in Spain, for instance.

Figure 2-3: Telco OSS Nomenclature, II

2.2.3 The Billing Module: The Call Rating Process Overview.

Let us now turn to studying what we could describe as the core process within the billing module in our OSS: the *Rating* process. The verb rate is associated with giving a value to some kind of evaluated expression. In this environment, we shall use this form of action to evaluate and assess the price of the services used by the customer according to usage and quality-level parameters. The concept of *Rate*, or more specifically *Call Rating,* as we are calculating the amount of billable units that we shall tightly associate with a certain phone call, is a very important piece of information because when dealing with a commercial network implementation, revenue shall come precisely from these services, as the customer shall be billed for their usage. Down to earth, this means that we shall have a component or module that shall calculate the cost of a telephone call according to some constraints and attributes.

The prior process, needless to say, and even though it can look as obvious and straightforward at first look, has nothing but simple as what initially can be a simple calculation of a mere number or variables, and can quite quickly become a complex and fundamental piece of our business revenue. Knowing how much a user or customer needs to be charged and doing so without errors either on our side or from the user's side, it shall mark the difference between been an agile company or just being one in the tail of the competition.

The rating of a call starts in the very moment that a call is attempted to be established by a request from the routing OSS module. At that precise instant, the customer can already be checked by the Authentication module to see if he/she has the proper permissions to make the call; assuming that this is the case and that the call has entered our Billing module, this can be rated regardless of the call disposition state as a carrier can even have a policy in place to bill just for trying to terminate calls not having to assure the proper termination of those at all times.

As for the call disposition, this shall be a factor determining the end calculation of the phone call in progress. As shown in Figure 2-4, the CDR – Call Data Record – for a phone call – shall be created regardless of the call disposition. This means that all the system calls shall always traverse the Billing module even if they did not go through as the destination calleé was unavailable or the routing was unable to determine a proper route for such a call. What has to be clear is that the Rating shall perform always some calculations on the phone calls handled by our platform. Furthermore, in a strict Telecommunications Network environment, no calls shall be left out of the OSS and even left outside or unassimilated by the Billing module. Each telephone call has to and it will be rated. This is needed to bill the customer as well as to know and to determine the traffic and interconnection costs with our peers.

Introducing some own nomenclature at this point, we have come up with the term CRGB, which stands for *Call Rating Granular Billing*. This is the fruit of our own design conception and its aim is to introduce the capacity to handle granularly the billing of each call. In spite of getting away with mechanisms traditionally used in the big Telco – Telecommunications – companies, we want to innovate and design something more scalable and flexible, thus the need for the CRGB as it shall be just one more of our OSS Billing Module piece that shall allow us to rate a call using a multiple number of parameters, dynamically defined and very likely defined in an open grammar or format for easier and quicker definition.

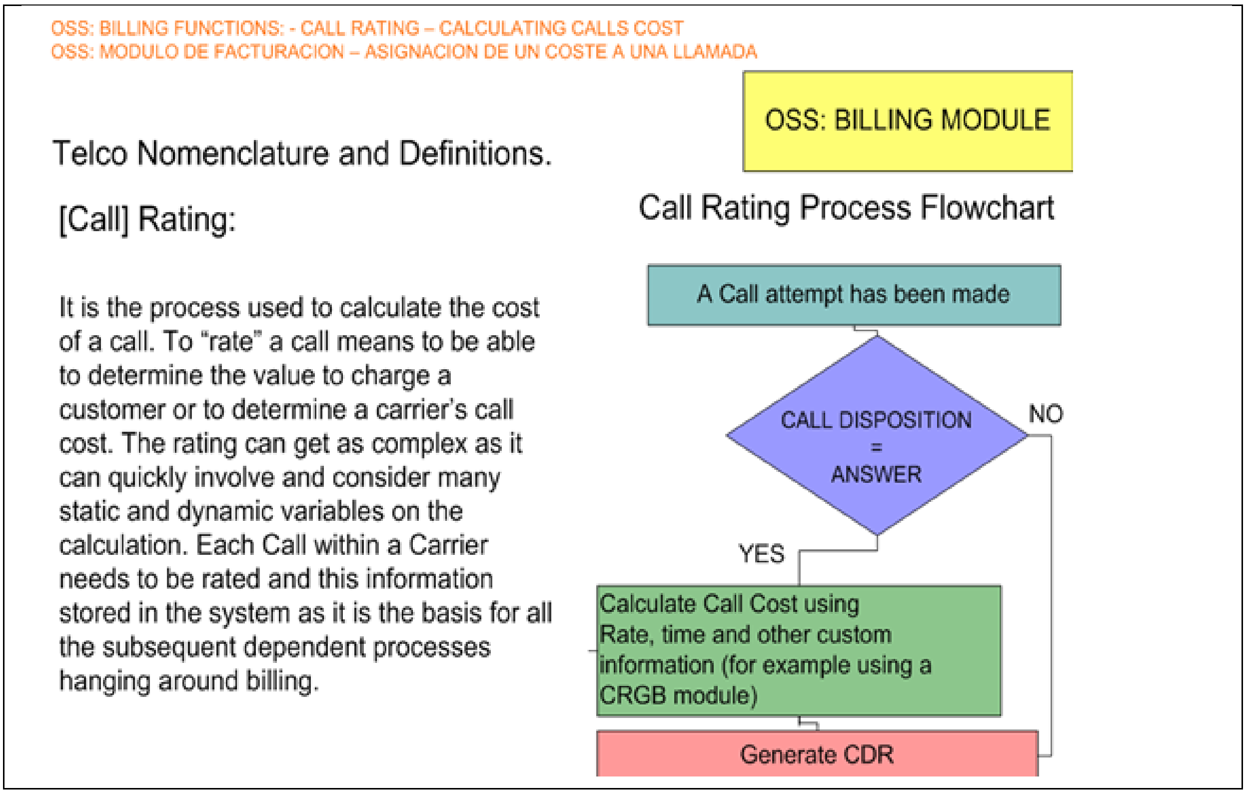


Figure 2-4: The Call Rating Process

To support our assertion that a powerful mechanism is needed to face the complex task of performing Call Rating, we shall take some simple phone calls records from a production system; as shown on Figure 2-5, we realize that even the smallest system shall quickly produce and compile a large amount of information stored in either databases or OSS repositories. This information containing the attributes, on which the rating calculation shall work on, becomes quite tedious to handle if the rating has to be always hard coded or implemented as a store procedure or something static. In the figure, we aim to calculate the resulted bill cost and net cost, being these the cost charged to the customer and the real cost of the call when handing it over to the associated trunk – or voice link -. In this case therefore as important is the call parameters as well as the trunk associated ones, as we want both values to be rated so that we can later on act accordingly. Also to be highlighted, and even though this belongs to the implementation section of our design, it is important to observe that we shall force all the network calls to have a unique identifier, and we shall emphasize and force this behaviour as a constraint to make sure that we can uniquely identify an event. This property shall later on prove to be a good design decision as it shall be ease the implementation and day-to-day management of the network.

In the initial CDR shown, we possess knowledge of a certain number of attributes associated to a phone calls like: the call identifier, start of the call, the trunk or carrier used to terminate the call, the customer id, to properly associate the call to a billable entity, the dialled number, that is a basic attribute to rate call, the call duration, and finally the variables of cost that are deliveries of our OSS Billing Module. CDRS shall be created as time goes by and more calls are made, therefore the database shall grow and grow up to the infinite, as it shall contain CDR information for every single call going through our platform.

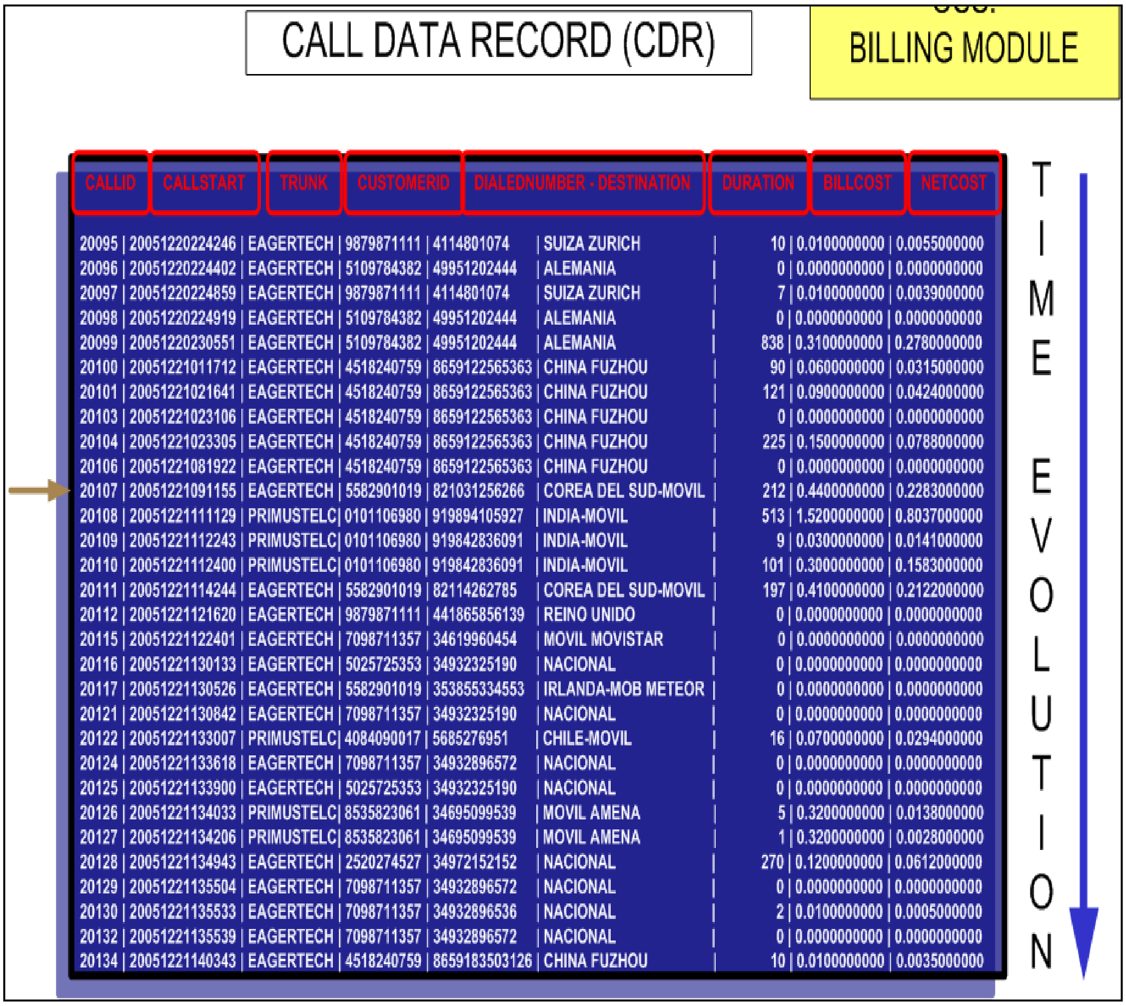


Figure 2-5: Billing CDRS Generation

In our OSS design, we shall enforce the support for multiple Rates in order to allow the carrier using our platform to have a vast range of services and marketing rates applicable to its customers. By defining this entity as abstract we can enjoy the benefits of having an entity as generic as it can be used for billing customers or as a tool to keep track of all our transactions with third-party systems. As mentioned before this is quite important and it shall turn out to be a nice benefit to have fruit of a properly design architecture. Figure 2-6 depicts the visual logic of the multiple rate support.

By allowing for instance granularly, one more time we end up having an entity which can be used to support multi-layered rates; that is different rates or sets of rates used for different purposes and group of users and trunks. Thus, our billing engine shall have as many defined rates objects as needed and desired and those can be associated to business layers or entities for which we interact. In an abstract way, we can imagine this as treating our customers, resellers, distributors, carriers, or any other third-party as the same entity or object at the end of the day.

The OSS billing platform shall treat them the same regardless of their business association but from a carrier's point of view; this shall translate into having a powerful engine that can be used from calculating and invoicing our customers, to calculate the funds earned or to be transferred and due/earned with our business peers.

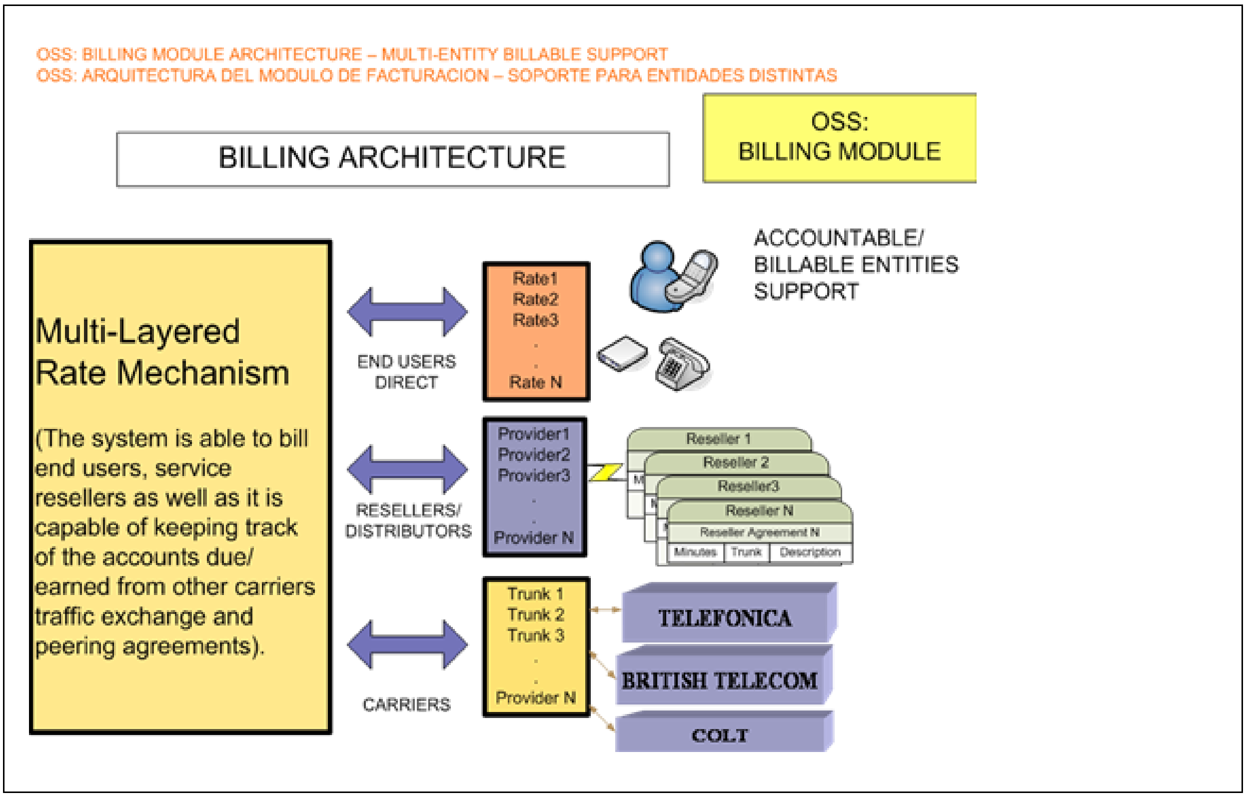
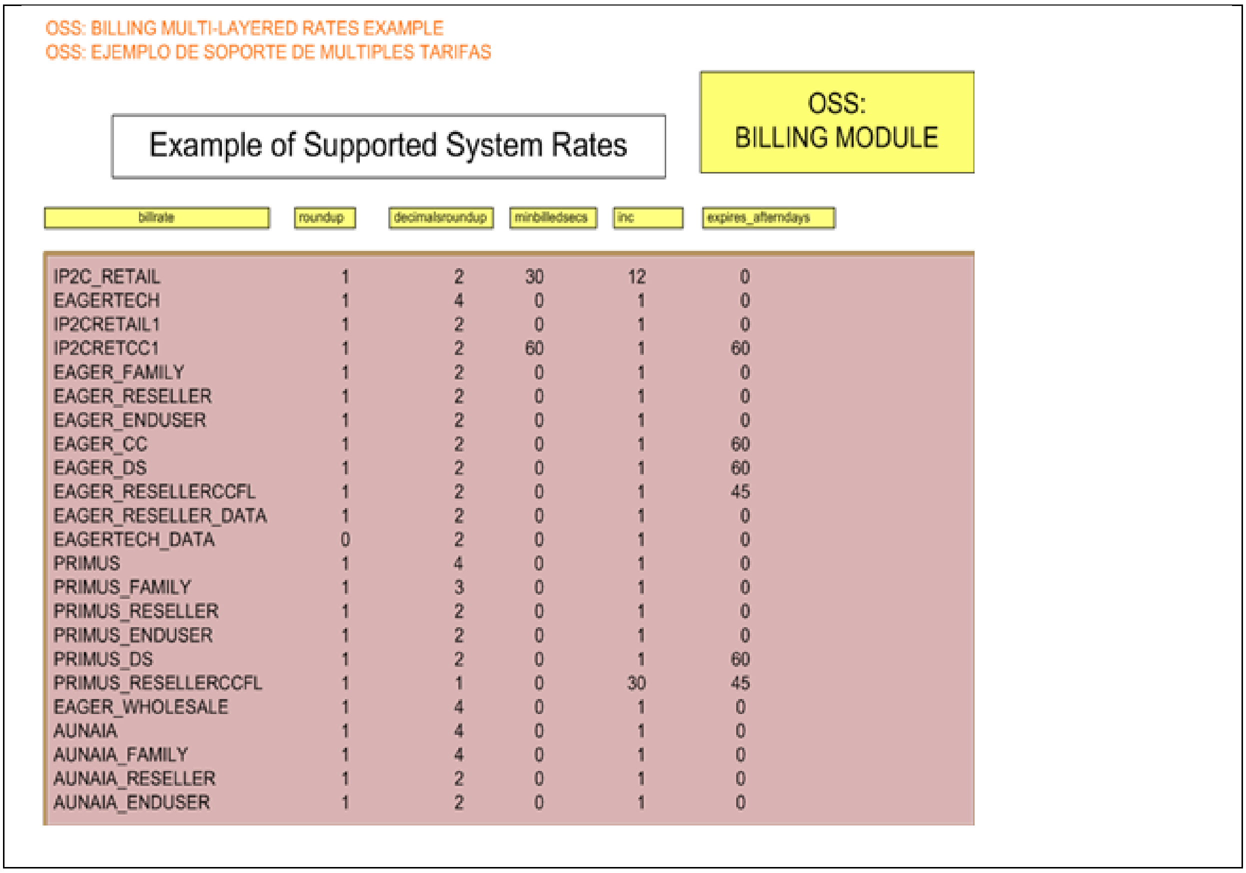


Figure 2-6: Support for Multi-Layered Rates

A rate can also be regarded as an object implementing and giving support for accountable objects being defined those as an entities used to manage our relationships with other systems. In this sense, those shall facilitate the tasks of managing our accounts or kind of giving some kind of CRM – Customer Relationship Management – functionality within our OSS platform.

Figure 2-7 lists an OSS system table containing different rates. These are defined to deal with different carriers as well as to define bill rates for end customers. The table has associated attributes that can be specified to narrow down the required behaviour when dealing with the rate object. For instance, some rates can have a minimum number of seconds to be charged to the trunk/user, whereas others can just have no default seconds at all. Others can have a different billing time increment to bill the entity according to that period of time, where additionally some of the rates can expire after a certain number of days or be always valid, on will. Additionally, the rounding process on the final cost is calculated according to an attribute, which specifies how this value has to be rounded. We might want to round up some value to two decimals when dealing with customers' billing but used the 4-decimal standard Telecommunications standard practice when exchanging billing information with other trunks/carriers, hence having a rate object with its variables and attributes that can grow on demand that shall allow us to implement future service in a straightforward way because the object schema in the OSS shall be easy to expand and shall inherit attributes and behaviour from its parent schema.



billrate	roundup	decimalsroundup	minbilledsecs	inc	expires_afterdays
IP2C_RETAIL	1	2	30	12	0
EAGERTECH	1	4	0	1	0
IP2CRETAIL1	1	2	0	1	0
IP2CRETCC1	1	2	60	1	60
EAGER_FAMILY	1	2	0	1	0
EAGER_RESELLER	1	2	0	1	0
EAGER_ENDUSER	1	2	0	1	0
EAGER_CC	1	2	0	1	60
EAGER_DS	1	2	0	1	60
EAGER_RESELLERCCFL	1	2	0	1	45
EAGER_RESELLER_DATA	1	2	0	1	0
EAGERTECH_DATA	0	2	0	1	0
PRIMUS	1	4	0	1	0
PRIMUS_FAMILY	1	3	0	1	0
PRIMUS_RESELLER	1	2	0	1	0
PRIMUS_ENDUSER	1	2	0	1	0
PRIMUS_DS	1	2	0	1	60
PRIMUS_RESELLERCCFL	1	1	0	30	45
EAGER_WHOLESALE	1	4	0	1	0
AUNAIA	1	4	0	1	0
AUNAIA_FAMILY	1	4	0	1	0
AUNAIA_RESELLER	1	2	0	1	0
AUNAIA_ENDUSER	1	2	0	1	0

Figure 2-7: Example of OSS Rates

The OSS should not really distinguish that much between an end-customer and an interconnected carrier. For the OSS, both kind of entities shall share the property of rate entity and the billing processes around them shall behave according to the object attributes, hence on the practical daily process, this shall mean that we will not really have to make such a big distinction between users of our network, as the OSS shall handle that automatically for us and make sure the information flow properly moves in the right modules and that each entity regardless of its owner, it is processed separately and passed over to the other modules in a transparent way.

Remarkable is this property of unifying all the system actors under the same umbrella of just OSS objects. For instance, in Figure 2-7, the system is coping with some 'transport' carriers – these bringing the voice communication down to its proper competition – or in telco nomenclature, those "terminating" the call; making sure the call travels a good path in order to the callée party's phone to ring and pick up the phone and start a human conversation. This mere session establishment can require the call to travel over multiple networks, traversing different Carrier's networks and finally reach its destination. Within our domain of control, our OSS shall be in control of our network edge limit and shall interact with the assigned trunk to deal with the call-termination process and final status, and finally bill according to it.

Some sample carriers are used: Eagertech and Primus as well as AUNA; all of them being alternative carriers providing International voice termination service. Some of them using VoIP pure protocols while the rest connected to our system using TDM circuits (specifically primary-rate 2 Mbps circuits at this time). This again highlights the fact that no matter what the transport physical technology in used is, the OSS shall abstract it from us and handle both from a logistics pure of way showing us all the carriers as just logical links used by the phone call to reach its destination. This approach totally free us from considering network implementation issues so that we can totally focus on methods to make sure the proper billing and routing of the phone calls are in operation.

The rates table is also used to define different levels of service as for example in the list we observe some rates valid for end-customers (pure end-users), and some customers with wholesale rates (discount per volume), and also rates for resellers and distributors, therefore the rate object with its scalable attributes shall let us play with a growing number of possibilities to define new services, new peers to make business with and a chain of distributors and resellers of the services we shall bring to market.

The granularity is the characteristic that shall permit us to narrow down the service characteristics down to a single user if desired. This shall translate in being able to fulfil the customer's need down to a single very specific user requiring a very specific profile. For such case, a rate can be defined as easy as any other and shall be treated by the OSS in a seamlessly way. This is the main advantage of defining and bringing a multi-layered rate support to our OSS from the very design stage: it shall make the dealing of users, trunks, carriers, resellers, distributors and any other future interacting party, transparent to the system as the OSS treats all the entities using a conceptual or object approach, which releases it from knowing the technical intrinsic characteristics laying underneath.

2.2.4 Billing Module: The Call Billing Flowchart

A telephone call shall enter the processing flow portrayed in Figure 2-8. First, the customer invokes a call, either an end-user using a telephone device or something more complex using a PBX, Virtual Call Centre, automatic call-generator or any kind of call-generating device. At this point, the call is handed over to the OSS to be tracked and controlled. In the Authorization OSS module the user profile is looked up to determine his/her permissions and the calling plan that he/she shall be covered on during this phone call.

Once authentication parameters such as finding out if the user is using pre-paid or post-paid, the quota of minutes associated to the service, and other multiple parameters defining and constraining the service, the call shall move to the establishment phase.

In this phase, the OSS Routing module shall determine the better candidate set of trunks to be used to terminate the current call request. As the call enters the Routing module, this shall intrinsically means that high-availability and fault-tolerant mechanisms shall be transparent and working throughout all the call handling process.

In the next chapter, we shall go into deeper detail as the software side of the implementation is concerned. At this moment, we convey to the reader to pay attention to Figure 2-8 as it introduces the flowchart of a telephone call bill process, which shall act as the basic requirement for the platform we have implemented and worked on during this dissertation work.

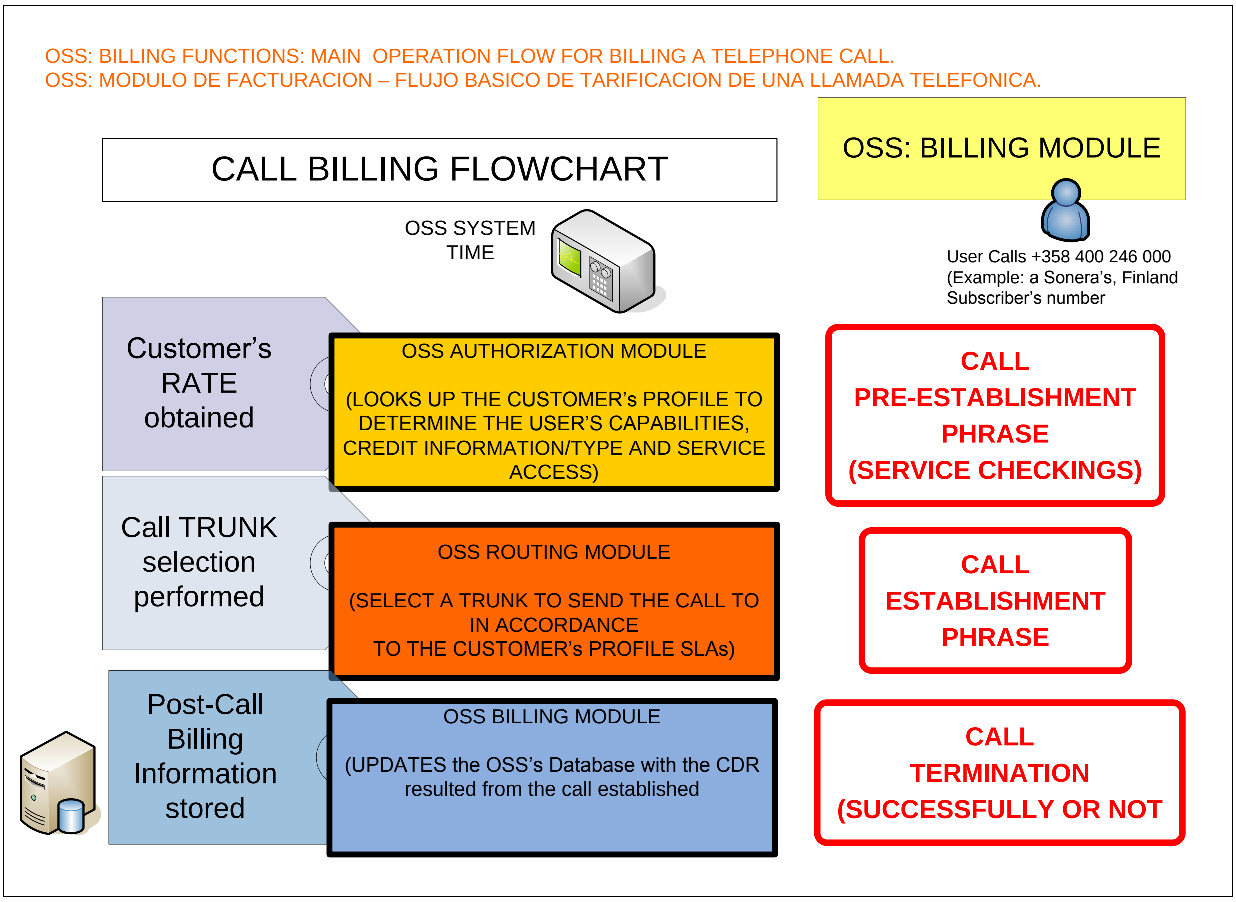


Figure 2-8: The Call Billing Flowchart Diagram

3 The Software Side: System CORE Implementation, Main Modules and Software Architecture Overview.

3.1 Software Development: methodology, tools and software engineering approach used.

Building an OSS platform from scratch is not a straightforward task. Tendency drags the developer to start coding rather than focus on the design side needed to develop any good software product. In this dissertation work, the author did have the main ideas of software design in mind whilst working on the system: *Object Oriented Design* tools and the concept of *Design Patterns*. Suffice it to say in plain language that the framework developed aimed to provide a sufficient level of abstraction in every single task supported: from database engine abstraction, different methods of authentication support for various underlying protocols, the idea of a generic bus to exchange information and using a uniformed language to exchange information among the various modules of the platform engine.

The software design part of this implementation was done using UML – Unified Modeling Language - which was used to prepare and finally generate code stubs for the C++ and Java languages. About 60% of the final implementation inherits the design approach in its final form, whereas the remaining 40% was initiated using a monolithic philosophy as basic plug-in tasks were needed. In other words: almost all the code used to interconnect the abstract layers with the underlying ones are done using procedural languages with some object-oriented support: basically, most of the code finalized bringing together the world of the soft switches together with our OSS platform was done using Perl - Practical Extraction and Report Language –. When the author worked on the first implementation, and due to the fact that the software used to implement the VoIP side of the architecture was an open-source implementation product – it is called *Asterisk* -, the first real work consisted in working on an abstract layer connecting this soft switch with a new higher abstract controller framework. As *Asterisk* is purely implemented in C and it supports a classical API-approach to extend its functionality, the language most suitable during the first parts of framework creation was chosen to be Pearl, due to its rapid development characteristics, being an interpreted language and with support for object oriented design, as well.

Thus, the majority of the initial implementation was done using the language and methodology mentioned in the last paragraph. Later on, and turning the initial work into a real dissertation and major-scope project, the author started to look at better and more scalable ways of developing what it would become the OSS platform controlling the overall Systems.

This was done using Java and C++ as the programming languages as they are very object-oriented rich languages ideal to be used with an Object Oriented Design and Abstractions ideas behind plus using Design Patterns to reuse code and framework-making ideas. As this was to become a *platform* or *framework,* it did make a lot of commonsense to shift the working time more towards a design part to later on ease up the programming and implementation phases.

This dissertation aim is neither to cover the daily programming tasks and pure coding of the modules, nor to detail the abstraction and design of the hundreds of objects and patterns gluing it all together.

The work described here strongly relies on common 21st century technology, commonly available protocols, distributed standards and data exchange formats as well as presentation and data manipulation tools.

In this dissertation, a framework bringing together heterogeneous telephone-company functions uses Internet-technology to implement them. The author´s aim was to use open standards in order for the final framework to possess the following characteristics: Scalability, Network and Modules Distribution and Security, in both sensitive-security protection and reliability and a wide range of fault-tolerance resilience. In previous years, technologies like Client/Server architectures, and later on distributed more serious and defined standards like CORBA – Common Object Request Broker Architecture - would have been used to implement our Telecommunications OSS framework. Long is due and passed for these technologies that have contributed with their concepts and ideas to the nowadays distributed standard: the Web standards. Everything has to run on the Internet now, or at least on top of an IP network. This is a must and requirement these days and consequently the distributed standards chosen or selected for this project work were the ones in use on these networks: SOAP – Simple Object Access Protocol – was assessed as a binding-protocol to establish our inter-modules communication interfaces, using HTTP – HyperText Transport Protocol – as the session and application protocol and the universal XML – Extensible Markup Language – was selected to serve as the *lingua franca* that our modules would understand and process for all the architecture requests.

An easy and very commonly adopted straightforward approach: the mentioned protocols to exchange information between our modules, object oriented languages to implement the module functionality and wrapping up the whole system with security protocols such as the SSL – Secure Socket Layer – and wider TLS – Transport Layer Security – ones that we relied on by means of reusing Java and C++ API. And for the design we did use UML to achieve a quicker – or at least more comfortable, rapid and with a higher degree of abstraction – this latter characteristic highly needed in order not to lose focus on our architecture framework design.

Current implementation consists of many Java, C++ and UML diagrams.; and also many SQL – Structured Query Language – as the framework heavily relies on databases to save and retrieves the objects´ persistence state during their execution. Finally, and very important as we implemented many of our visualization code using the technology we need to mention the contribution of the XSLT - Extensible Style Sheet Language Transformations – standard, as we used it also very heavily for all the visualization transformation carried out by our different platform modules. The language of communication between modules is XML but the language used to convert and process these data and turn into several kinds of format needed at different times and for different purposes, was XSLT. The “magic” that this standard incorporates can only be described as extremely considerable. Many of our internal displaying and processing of data was implemented using this good collaborative relationship between the XML and XSLT protocols.

3.2 Work History and Chronology

The work outlined in this chapter is the realization and documentation of an enterprise dream that started work on March, 14th 2005. The software and design described here has so far supported more than 1 million calls and processing more than 10 million of voice minutes and traffic currently keep on doubling on a two months basis, thus reaching soon an exponential growth. The author developed and deployed the first phase of this software a couple of years ago and it has been running in a production environment since then.

In this period, and using this dissertation as a good time to redesign and apply the learnt lessons from the engineering career, the author turned the initial development into an open architecture to support growth and future needs. As only our own software is used this means that our daily operations depend on ourselves for everything from voice processing to billing, routing and quality of service.

The initial software developed in Perl has gone through major revisions and improvement over the last 24 months and in this dissertation the underlying layers have been migrated and moved to the architecture depicted here; basically learning from our needs and fulfilling them in a glance before they occur. We would like to convey to the reader the fact that the main work here was to initially put all the pieces together; and later on to have the time and skills to reverse engineer our software and redesign it using the proper design methodologies learnt during the time spent studying Computer Science at the Universitat Autònoma de Barcelona. This is the important foundation; the rest can be observed as pure hours and hours of coding and design time.

Purpose	Technology/Tools
Documentation	JavaDoc, Eclipse, Open-source Tools, Rational Rose, Software Architect
Basic Framework	Perl and SQL
Modular Framework Redesign	UML
Development Tools	Software Architect, Rational Rose, Open-source Eclipse OOD tools, Microsoft Visual Studio
Object Oriented Design	UML
Programming Languages Used	C, C++, and SQL
Distributed Objects Technology	SOAP, Webservices
Security Protocols	SSL, TLS
Information Exchange Format	XML
Generic Transport Protocol	HTTP
Visualization and Transformation Protocols	XLS and XLST

Table 3-1: Summary of some of the tools and technologies used in this dissertation work

Table 3-1 shows some of the tools, languages and standards used during the conception and development of this dissertation work. It merely lists the most used items as many others have also been used but there are so many that they are into a second layer of already integrated protocols and concepts

In an initial phase of this project, the author assessed at least twenty different use-case and UML-supporting design tools. It was a very time-consuming task to find the right tools for our architecture development; additionally the same was done to find the appropriate IDE environment, compiles, programming languages, database engines, web servers, java containers, VoIP libraries and so. The amount of time involved in all these tasks can be summed up as thousands of hours.

3.3 Software Modules and Architecture

Our initial software was replaced by a totally redesigned architecture in mind: one with support for a common way of exchanging messages between modules using a standard message format and a communications distributed bus. Our structure and architecture relies on these ideas regardless of the module implemented: Billing, Routing and Authentication.

They all follow the same guidelines outlined in our previous paragraphs as design ideas and concepts. We shall now introduce the Billing Module implementation as a sample of the implemented concepts; the other core modules follow the same ideas in their implementation servicing as a strong pillar supporting the whole system.

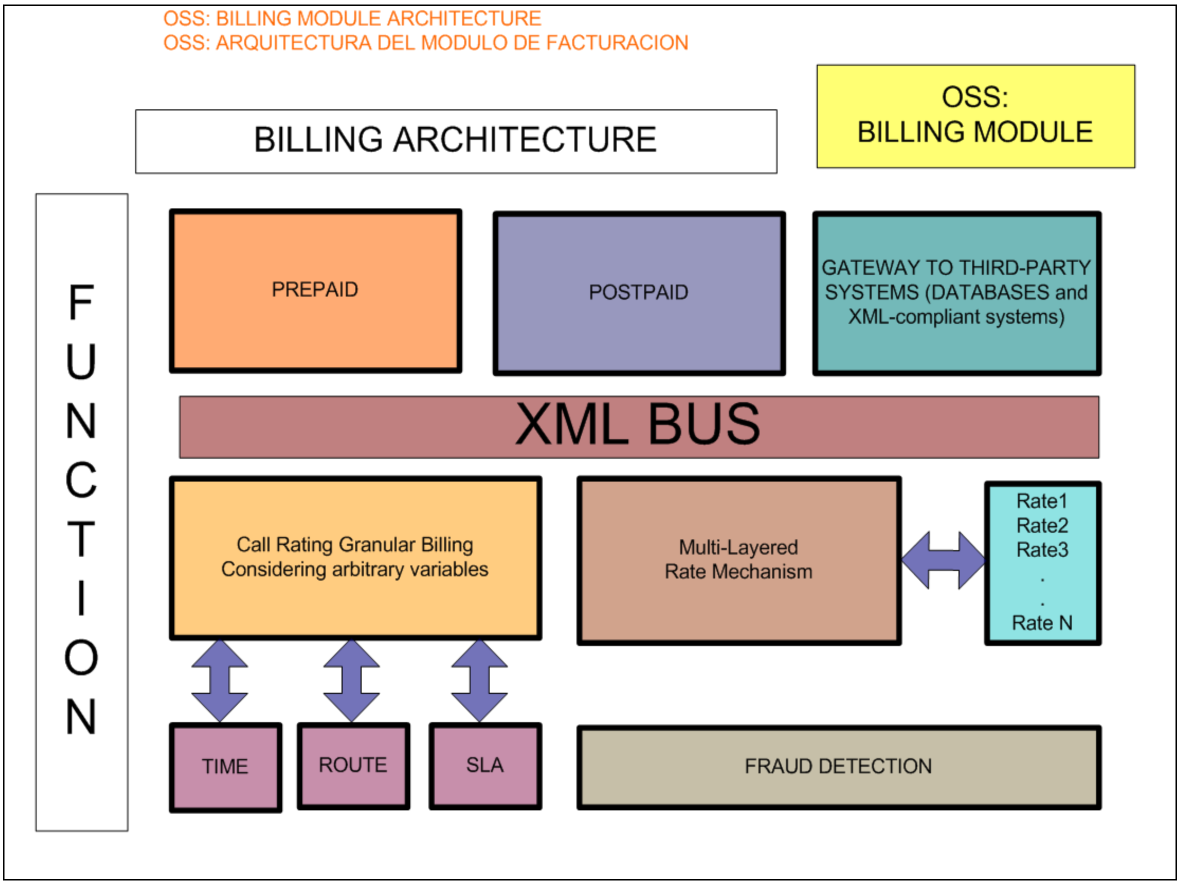


Figure 3-1: Billing Architecture Module Implementation

Supported implemented modules within the Billing Module can be summarized as graphically depicted in Figure 3-1 . Each module outlined in that figure is implemented by means of a set of abstract classes implementing the required functionality for each module through object methods.

These interact between themselves by using an XML Bus and SOAP to exchange functional information at every moment.

3.3.1 Generic Module Functionality Description

The following modules supported in our implementation are briefly described below:

Module	Performing Task
Prepaid	Controls, authenticates and validates a prepaid user. It communicates with the authentication parts of the system to get the users' credentials and determine which profile to assign to the current system session.
Post-paid	Performs authentication based on the same system policies but the billing information is populated into our databases modules to be processed and analysed when needed afterwards. Quota checking is still performed.
CRGB	Call Rating Granular Billing: this is a concept developed for this platform and it basically translates into being able to very granularly bill a user for its session actions and activities. An user can be billed using several variables predefined in the system. This allows for a very scalable way of define and bill future services as this was originally the idea: to be able to subsequently create new services without the need to redefine our platform.
Time	The variable time can be used as a factor for applying different billing polices.
Rate	A rate is associated to each user according to his/her system level, tariff, permissions, volume discounts, rapports and other billing and financial policies supported by our billing mechanism.
SLA	Service Level Agreement: The system allows for real-time determination of the best route and quality of service to provide depending on the users´ SLA attribute.
Multi-Layered Rate Mechanism	The system supports multiple rates. These can be defined according to the platform from where the user it is connecting, according to his/her billing attributes, his/her permissions, and multiple-layers of rates can be associated to the same users depending on several predefined variables. This allows for easiness of defining new services as it expands flexibility.
Fraud-Detection	A basic abstract module takes care of analysing the billing data in real-.time according to the policies and rules we define in the system. As the module is expandable, it is possible to define different algorithms and statistical patterns and data-mining approaches to determine if the event of fraud has occurred.
Gateway to Third-Party System	This module is meant to be the gate between our platform and third-party ones: as we use XML for communications and we support SOAP and HTTP for objects transport, we can communicate with other Webservice Platforms in order, for example, to exchange CDRS and quality of service metrics. This property also makes this architecture an open one as it has a way to interact with the rest of the telco systems available out there.
XML Bus	Conceptually an XML bus by currently implemented as a set of objects communications among themselves using XML. In the future we can implement it as a bus in the essence that we can allocate and assign Bus controllers to authenticate and double check ongoing operations.

Table 3-2: Billing Modules Overview

XML was chosen as it incorporates a good way of defining our own application messages. To implement the functionality of our OSS modules we did need to define several (actually a few dozens at the end of the day) *schemas* or XML schemas: set of containers that include a set of attributes to exchange information between our operational modules. As we did already mention earlier on, by doing this, we were able to utilize XML as the *lingua franca* or way to exchange commands and data between our different modules. One of the applications we implemented using this approach, was the exchange of billing data as in many occasions we need to either process or visualize this information either internally or transform it for user purposes. Figure 3-2 summarizes the process used.

When thinking about a way to easily and efficiently store and display billing data, XML turned out to be a very good way to encode this information into. We implemented functionally where the Billing module collects the session information, converts it to XML and pass it to either a subsequent transformation module to come up with the desired end format: either PDF, HTML, XHTML or even other application-specific formats such as Microsoft Excel.

We send XML messages. These are received by the appropriate displaying module, which processes the query and acts consequently as requested. The output format is then requested and the outcome is the data in the format we needed. Everything is automatic and all the modules just send messages between each other. A post-processing layer finally creates the suitable format ready to be sent to the final user. The *magic* of this is that it is abstracted and the essence of the functionality is auto-embedded in the message itself. Each module is aware of the operations to perform when it receives the message and it acts accordingly. We did strongly rely on XML Schemas to define our many sides of our internal processes.

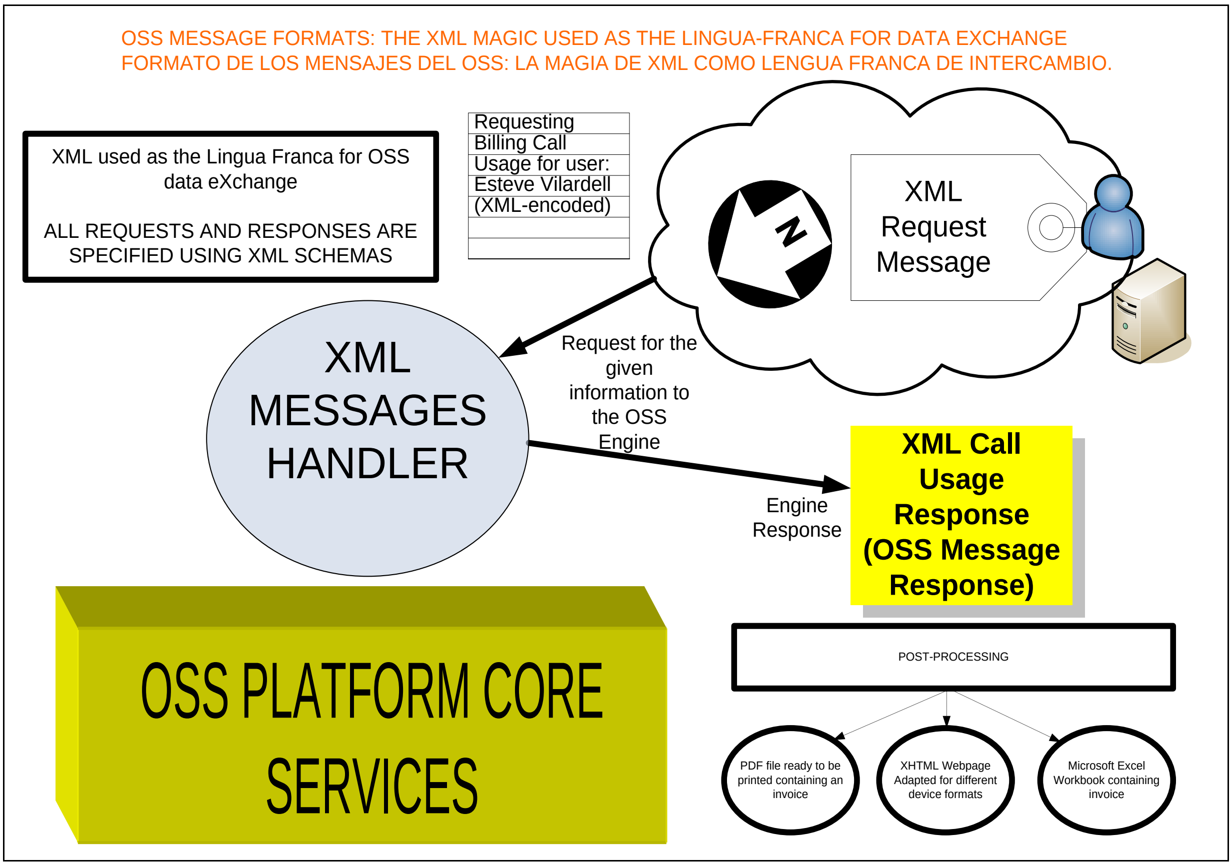


Figure 3-2: XML as our System Lingua Franca

3.3.2 The importance of the XML Schemas as the platform information bearers

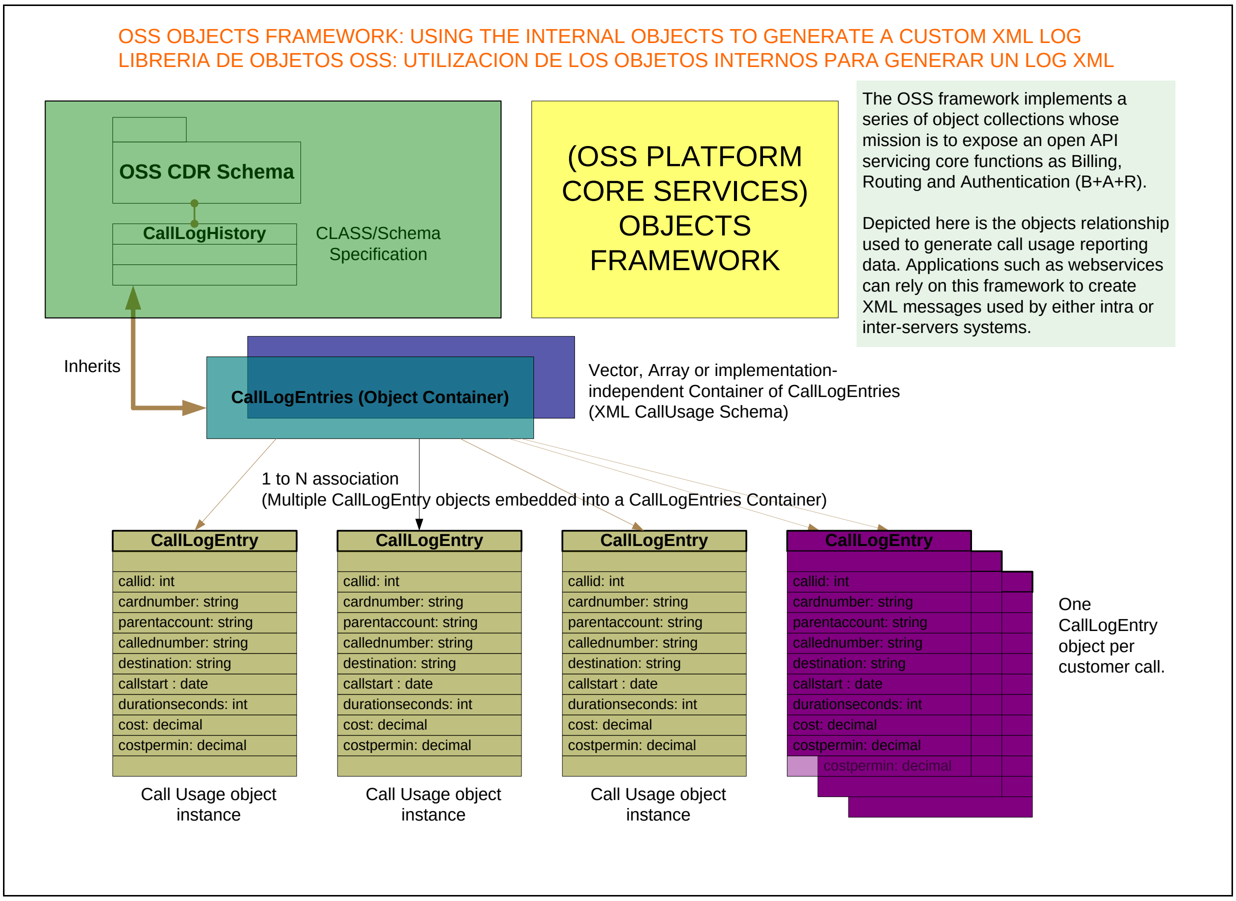


Figure 3-3: Sub-Level of our OSS Framework Objects

Figure 3-3 depicts one of the many uses of this methodology to implement our production services: one of these is the processing of CDRS, split into different object classes' containers, to generate a system log.

CDRS are collected from the call sessions controllers and their information stored into object classes to subsequently be processed and persistently stored in a call log container. This can then be passed to the associate class handler which shall therefore take care of generating the system logs. Thus, the initial raw data consisting of pure billing information is firstly, for example, memorized or stored into a vector, then it is passed to an object which reads its associated XML schema and pass it over to another object, which finally creates the final call object per customer or user's call.

By all means this example gives us an overview on how we internally pass information from one source to the other, processing it when appropriately by the associated module if needed.

As a rule, we need to have an XML Schema available for each of our architecture system functions. Without it, the XML would only contain raw data whose meaning would be unknown and therefore incapable of being processed. By associating a schema to our objects we are inheritably giving them the value, as from that moment each object turns to know its meaning by the system. In other words: the XML Bus and modules is capable of identifying each object and therefore performing the right operations on it. This level of abstractions gives the architecture a great deal of scalability, autonomy and awareness.

3.3.3 Message Schemas Example Used by the Platform Modules

OSS MESSAGE FORMATS: XML BEARER OF BILLING TELEPHONY INFORMATION (1 / 2)
FORMATO DE LOS MENSAJES DEL OSS: XML PORTADOR DE INFORMACION DE TARIFICACION (1 / 2)

Top Node of the XML Schema

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Copyright 2005 IP Square Communications. All rights reserved. -->
+ <callhistory>
```

Expanded Node containing the actual Schema attributes

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Copyright 2005 IP Square Communications. All rights reserved. -->
- <callhistory>
  + <ipsquareserviceinfo>
  + <calllogentries>
  </callhistory>
```

Expanded Node embedding operation result information for statistics

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Copyright 2005 IP Square Communications. All rights reserved. -->
- <callhistory>
  - <ipsquareserviceinfo>
      <operationresult>0</operationresult>
      <connectionused>0</connectionused>
    </ipsquareserviceinfo>
  + <calllogentries>
  </callhistory>
```

The platform's OSS core is basically a language-independent engine which proceeds requests from clients and is able to exchange service-related dynamic information with other internal or third-party systems. In order to perform this function, it relies on what it seems is becoming the 21st century lingua franca for data exchange: the XML (eXchange Markup Language) as the format for messages exchange.

Therefore, instead of relying on a proprietary format, it at all times generates and parses XML-compliant data. Internally, the OSS engine relies on a OSS framework which comprises all the needed functionality to be XML-savy and therefore 'undestands' any new messages implemented in the future.

This feature is what makes the OSS platform an open and adapting one as it is able to scale and adapt to future services implementations as the engine and framework can be implemented in any language and kept at any edge of the backbone, whereas the real value is that information is easily and ambiguously exchanged among all the parties and servers involved in the telco architecture.

Figure 3-4: XML Bearer of Billing Telephony Information Message

On Figure 3-4, the parent definition of our multiple XML Schemas is shown. The OSS works all the time by exchanging and processing messages in this format. For each different function encompassed in the framework an associated schema exists. In this specific instant the one the container of billing data is described and defined.

Later on further attributes shall acquire its operational meaning whilst processing calls, authentication and routing processes. For each of these actions the framework looks up in the system repository for the needed schema, does the binding and then passes the object over its handler. We shall next expand this into another example where the exact number of variables within the object class is shown. The object contains information about a just-finished call session and the message is passed to a handler that shall parse the data into XSTL format for performing the needed visualization and transformations,

The OSS initiates, authenticates, controls and post-processes each call flowing through the system regardless of its starting end-user device and destination. It acts as the master controller supervising all the real-time calls and assuring each call is correctly processed and that nothing gets out of control and its information lost. Billing is critical for a telephone company; consequently our implementation relies also on underlying database engines which replication enabled and with a supporting engine capable of rolling out data transactions and supporting a three-phase commit. We also implemented this in our framework: in our actual C++ implementation of the *Database Abstraction*

Layer. Again, for reasons of not being too extensive we shall postpone this module documentation for the day when we finish this software and turn it into a commercial product.

That is also the plan as the system has proved to be working correctly for many months now and it successfully performs the tasks that other commercial and much more expensive and complex software performs. Due to a good design simplicity and methodology we have achieved some of the same functions that only big telephone companies have.

3.3.3.1 Billing XML Schema In Detail: an example of XML being used for message exchange functions

OSS MESSAGE FORMATS: XML BEARER OF BILLING TELEPHONY INFORMATION (2/2)
FORMATO DE LOS MENSAJES DEL OSS: XML PORTADOR DE INFORMACION DE TARIFICACION (2/2)


```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Copyright 2005 IP Square Communications. All rights reserved.  -->
- <callhistory>
  + <ipsquareserviceinfo>
  - <calllogentries>
    - <calllogentry>
        <callid>18435</callid>
        <cardnumber>7098711357</cardnumber>
        <parentaccount>7098711357</parentaccount>
        <callednumber>34933905870</callednumber>
        <destination>NACIONAL</destination>
        <callstart>2005-11-30 16:56:37.0</callstart>
        <durationseconds>207</durationseconds>
        <cost>0.06</cost>
        <costpermin>0.015</costpermin>
      </calllogentry>
    - <calllogentry>
        <callid>18432</callid>
        <cardnumber>7098711357</cardnumber>
        <parentaccount>7098711357</parentaccount>
        <callednumber>34933905870</callednumber>
        <destination>NACIONAL</destination>
        <callstart>2005-11-30 16:48:44.0</callstart>
        <durationseconds>229</durationseconds>
        <cost>0.06</cost>
        <costpermin>0.015</costpermin>
      </calllogentry>
    - <calllogentry>
        .
        .
        .
```


Depicted is an OSS message containing Call Usage Billing Information.

An XML message implementing a custom OSS XML-Billing schema is the chosen way to share the data to external applications.

The self-containing XML message can be parsed by any application or can be filtered using XSLT – XML stylesheets templates to generate any desired output such as an XHTML webpage, a text document, a PDF or any needed format provided the filters are implemented.

Figure 3-5: Example of an XML Billing Message instantiated

Figure 3-5 shows a real message being sent across the system: an instance of one of the multiple XML schemas used in the framework this time used to propagate billing data. Call-related attributes such as called, caller id, duration, called number, applicable rate, cost, billing ranges and increments and further session data is embedded in the object itself.

The depicted object shall then traverse our processing engine to finally be converted to the final format we need for this transaction. For this exact instance of our billing object an invoice-generation process shall occur. Thus, basically the whole process of generating an on-the-fly invoice to be sent to the user is triggered by passing this message to the appropriate module handler.

3.4 XML Billing Message Implemented Process

During the phase of development of the needed requirements we implemented the processes depicted in Figure 3-6. One of our first real-world requirement was to implement a mechanism to invoice our customers as we remind the reader this dissertation project is based on a system which is already working in production for a few months so far, thus real world needs require modern and agile solutions: we did use XML and the process shown in this figure to implement this need. The result was that we achieved the capacity of generating invoice on-the-fly just by sending billing data, embedded in our parameterized XML-schema format, to our invoicing engine. The philosophy in use is always the same, take advantage of the abstraction capabilities to the maximum level of performance possible; this brings us a greater flexibility as each module in our architecture becomes independent and autonomous only interacting with its counterpart using a standard message known by all the entities involved in our OSS Platform.

The process is a straightforward one: information arrives in our system via an object instance, it lands in the system as an XML message, then the OSS parses the message, identifies its associated module and dispatch it for processing. If the involved process has to do with visualization and/or format conversion, the XML is parsed, its data retrieved and the resulted format stream generated. This minimizes the effort involved in document handling and it turns an old and time-consuming process into an agile and flexible one.

Our OSS is currently able to generate HTML, XHTML and self-embedded CDRS as CSV or other text files. Further on, it is also capable of generating PDF files from the XML sources.

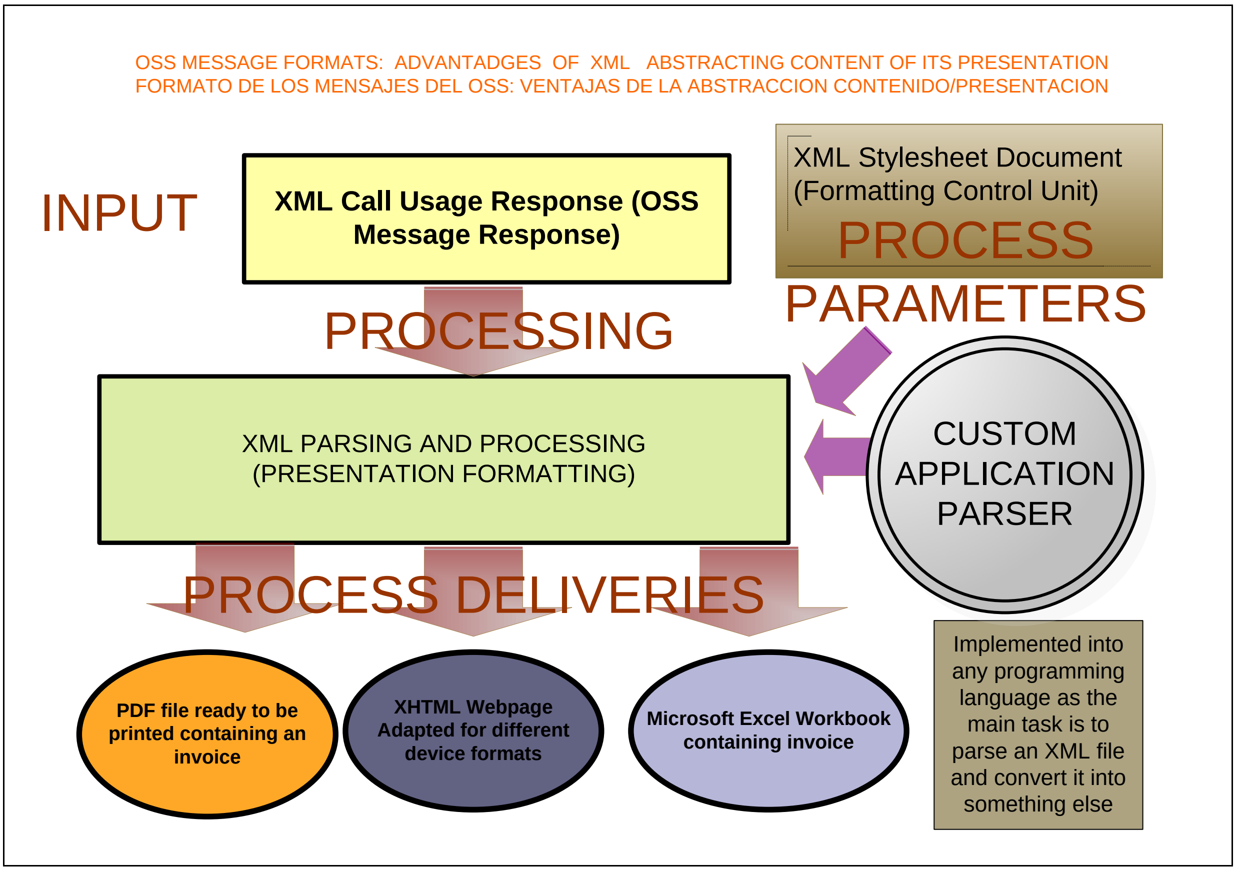


Figure 3-6: XML Presentation/Content Abstraction

The controlling part of this previous process is controlled by means of defining our output format in XLST code. This allows us to dynamically generate outputs: invoices for instance, by just modifying the given Style sheet.

The same applies for other formats as basically what we have achieved by implementing the process shown in Figure 3-6 is to abstract totally content from visualization. XML besides of being used as our platform lingua franca also provides us with the additional nice feature of isolating and abstracting the presentation layer. This is worth highlighting as it converts our Billing Module automatically into a presentation modeler and document creator.

The trigger or controller of this process is the XSTL definition. By defining one style sheet using XSTL we instruct our parser module to process an XML object in the desired way. This allows for the multiple document generation, changing of presentation options and artwork or look-and-feel and to present the data adjusted and formatted to the user´s end device: Mobile phone, PDA, iPhone, Web Browser, WAP Browser in WML format, Third-Party CRM and so on. XSTL controls the way the final information shall look and be processed.

Figure 3-7 shows a formatting control document, in XSTL format, which specifies the way one of our billing objects in XML shall be transformed for user visualization. Specifically, and as we shall see in the next pages, we shall convert our example data into an invoice ready-to-ship to the user/customer. Clearly the abstraction of data and visualization introduces a great flexibility as, from now on, we only need to customize and work on the final format look-and-feel for either the end-user or automated data handler. No more the data is together with the visualization which translates into the fact that we can just pass objects to our architecture modules and not having to parse and create data and visualization outputs all the time. This is now delegated to only the final phases when we really need to create and display human-readable or human-ready formats.

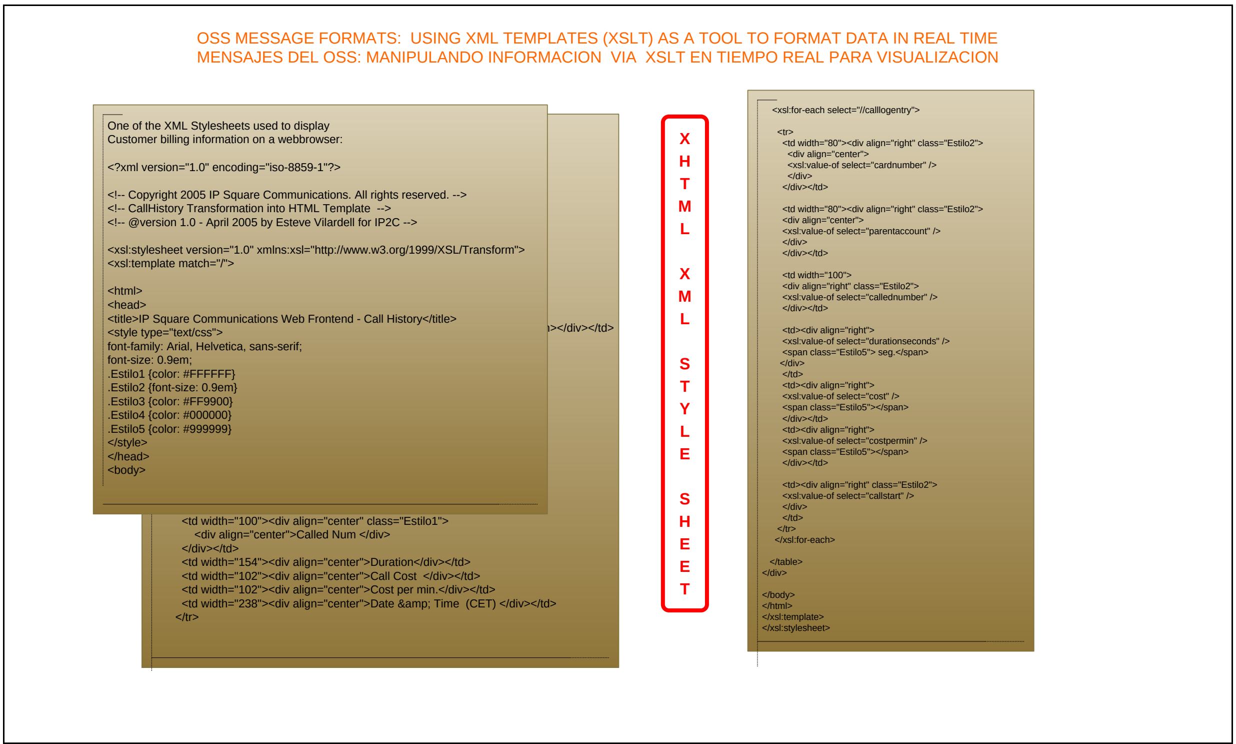
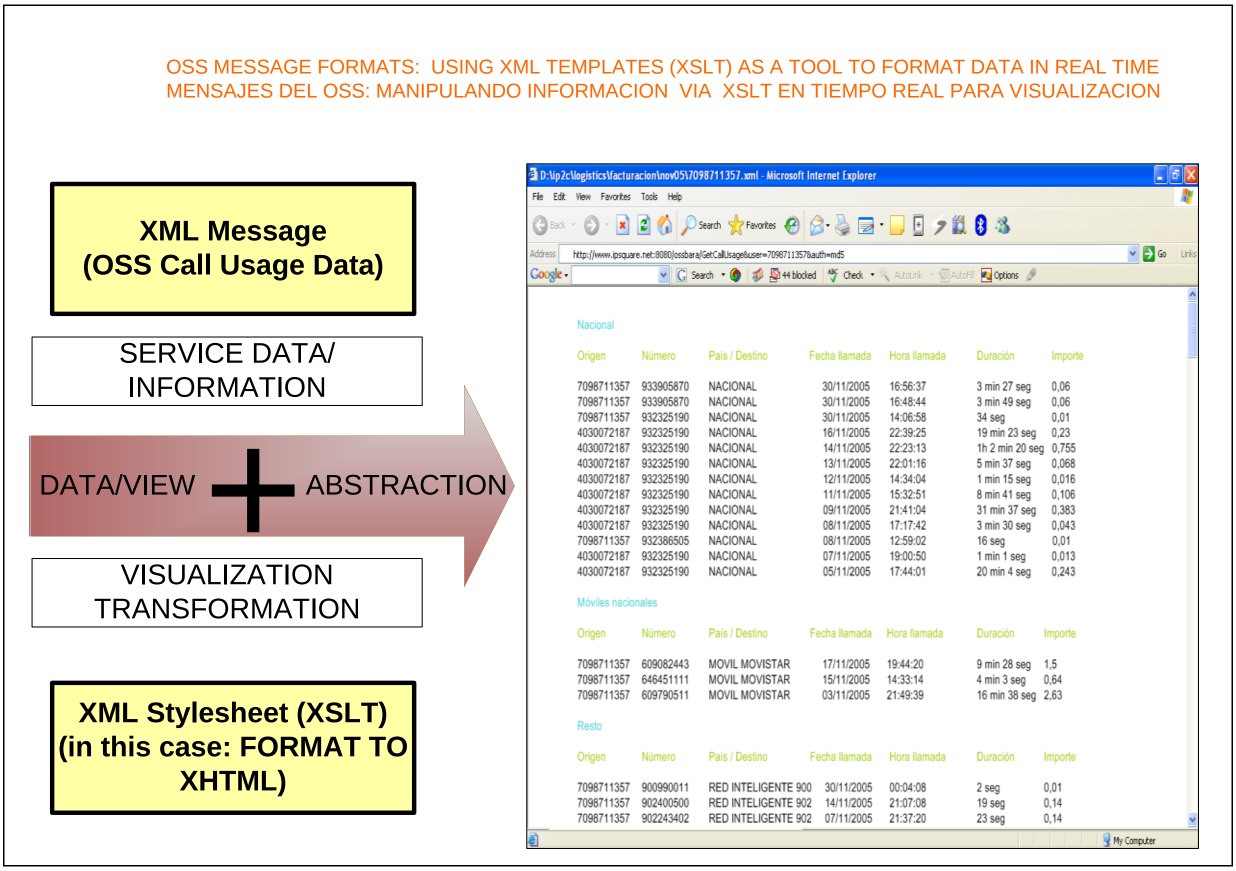


Figure 3-7: XSLT Visualization/Formatting Control

3.5 Automatic Invoice Generation Example using XML/XSLT

An OSS platform is expected to be able to perform Invoice generation, CDRS consolidation, and financial account mediation and so on. These are nice-to-have features expected in any contemporary platform. For illustrating the implementation that we executed, we shall now convey to the reader how we implemented the process of invoicing. Previously we described how we divided content and visualization transformations by abstracting both concepts and using different language definitions to deal with them. An actual example is a real invoice generation. In Figure 3-8 we use the platform to create an XHTML-compliant page. This shall be used by the system´s users to view their rates, balance and CDRS online. Again, by splitting content from visualization we are able to present and handle the data presentation format in different formats as needed whilst the data itself remains consistent along all the modules involved in our architecture.

We collect OSS Call Usage Data from the call session controller, then this data in XML format we feed it to our invoicing-generating object class, which identifies the requested service, parses the data, extracts it, takes the XSLT object containing the desired look-and-feel to be generated and finally generates the document in real-time in a dynamic way. Everything only by specifying what the data object is and what the visualization transformation is going to be, too. By using these two parameters our platform automatically performs the operations and creates an online CDRS ready to be sent by an HTTP Browser or converted to any other needed format for fulfilling the end-user needs.



Nacional

Origen	Número	País / Destino	Fecha llamada	Hora llamada	Duración	Importe
7098711357	933905870	NACIONAL	30/11/2005	16:56:37	3 min 27 seg	0,06
7098711357	933905870	NACIONAL	30/11/2005	16:48:44	3 min 49 seg	0,06
7098711357	932325190	NACIONAL	30/11/2005	14:06:58	34 seg	0,01
4030072187	932325190	NACIONAL	16/11/2005	22:39:25	19 min 23 seg	0,23
4030072187	932325190	NACIONAL	14/11/2005	22:23:13	1h 2 min 20 seg	0,755
4030072187	932325190	NACIONAL	13/11/2005	22:01:16	5 min 37 seg	0,068
4030072187	932325190	NACIONAL	12/11/2005	14:34:04	1 min 15 seg	0,016
4030072187	932325190	NACIONAL	11/11/2005	15:32:51	8 min 41 seg	0,106
4030072187	932325190	NACIONAL	09/11/2005	21:41:04	31 min 37 seg	0,383
4030072187	932325190	NACIONAL	08/11/2005	17:17:42	3 min 30 seg	0,043
7098711357	932386505	NACIONAL	08/11/2005	12:59:02	16 seg	0,01
4030072187	932325190	NACIONAL	07/11/2005	19:00:50	1 min 1 seg	0,013
4030072187	932325190	NACIONAL	05/11/2005	17:44:01	20 min 4 seg	0,243

Móviles nacionales

Origen	Número	País / Destino	Fecha llamada	Hora llamada	Duración	Importe
7098711357	609082443	MOVIL MOVISTAR	17/11/2005	19:44:20	9 min 28 seg	1,5
7098711357	646451111	MOVIL MOVISTAR	15/11/2005	14:33:14	4 min 3 seg	0,64
7098711357	609790511	MOVIL MOVISTAR	03/11/2005	21:49:39	16 min 38 seg	2,63

Resto

Origen	Número	País / Destino	Fecha llamada	Hora llamada	Duración	Importe
7098711357	900990011	RED INTELIGENTE 900	30/11/2005	00:04:08	2 seg	0,01
7098711357	902400500	RED INTELIGENTE 902	14/11/2005	21:07:08	19 seg	0,14
7098711357	902243402	RED INTELIGENTE 902	07/11/2005	21:37:20	23 seg	0,14

Figure 3-8: Real-Data formatting and visualization

3.5.1 Example of Invoice Creating by XSLT document feeding into our Engine

In our introduced visualization-transformation process the key object is the visualization format specified by means of an XSLT template. By defining the output we require, our implemented framework module takes care of the dispatched task and generates the submitted output. Figure-3-9 shows a Microsoft Visual Basic code that generates XSLT and passes the request to our Visualization engine. Then, by combining the generated XSLT controlling file along with the data contained in the XML object a final invoice is produced.

One of the many daily uses of this architecture in the operation of the commercially-funded company running as a licensed telco in Spain, is to generate invoices for the customers, billing for the provided services to them; consequently invoicing is key to adequately and periodically process and release payment to vendors and coordinate with all the suppliers that the company interacts with.

In other words: one of the platform first implemented service was one designed to create invoices: it was implemented using Microsoft Visual Basic to parse CDRS, then the application did produce XML files with the billing data, then it passed them to a C++ application that combines XML+XSLT and creates an XML file already formatted embedded in XHML. Finally, the Microsoft Visual Basic application takes this input and finalizes creating a Microsoft Excel file-compatible invoice that it is sent to the customer using electronic means. This allows to minimize and remove the hassle of the invoicing heavy tasks and help the company relying on our platform to run their operations on a quick and practical way thus being able to survive in the very competitive telecommunications market field.

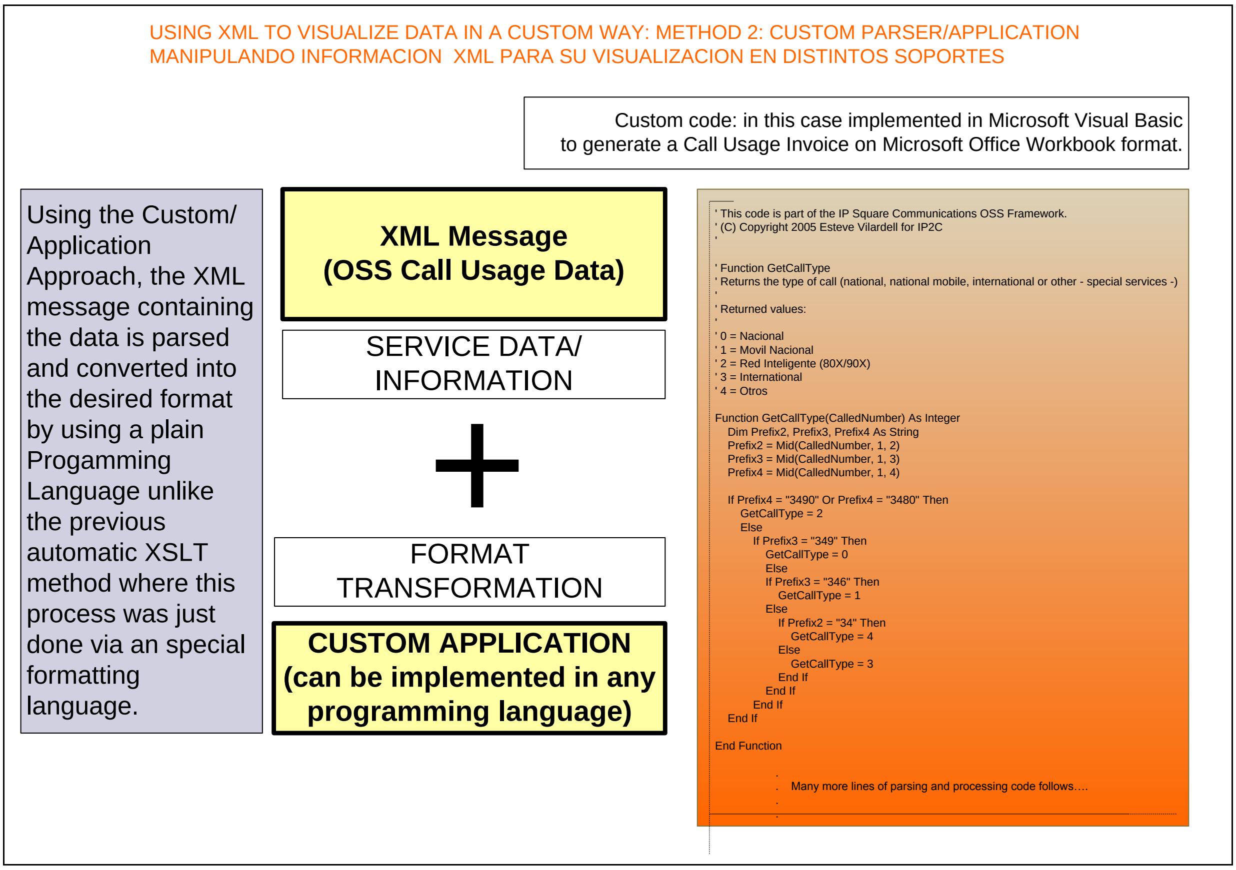


Figure-3-9: Visualization Example Overview

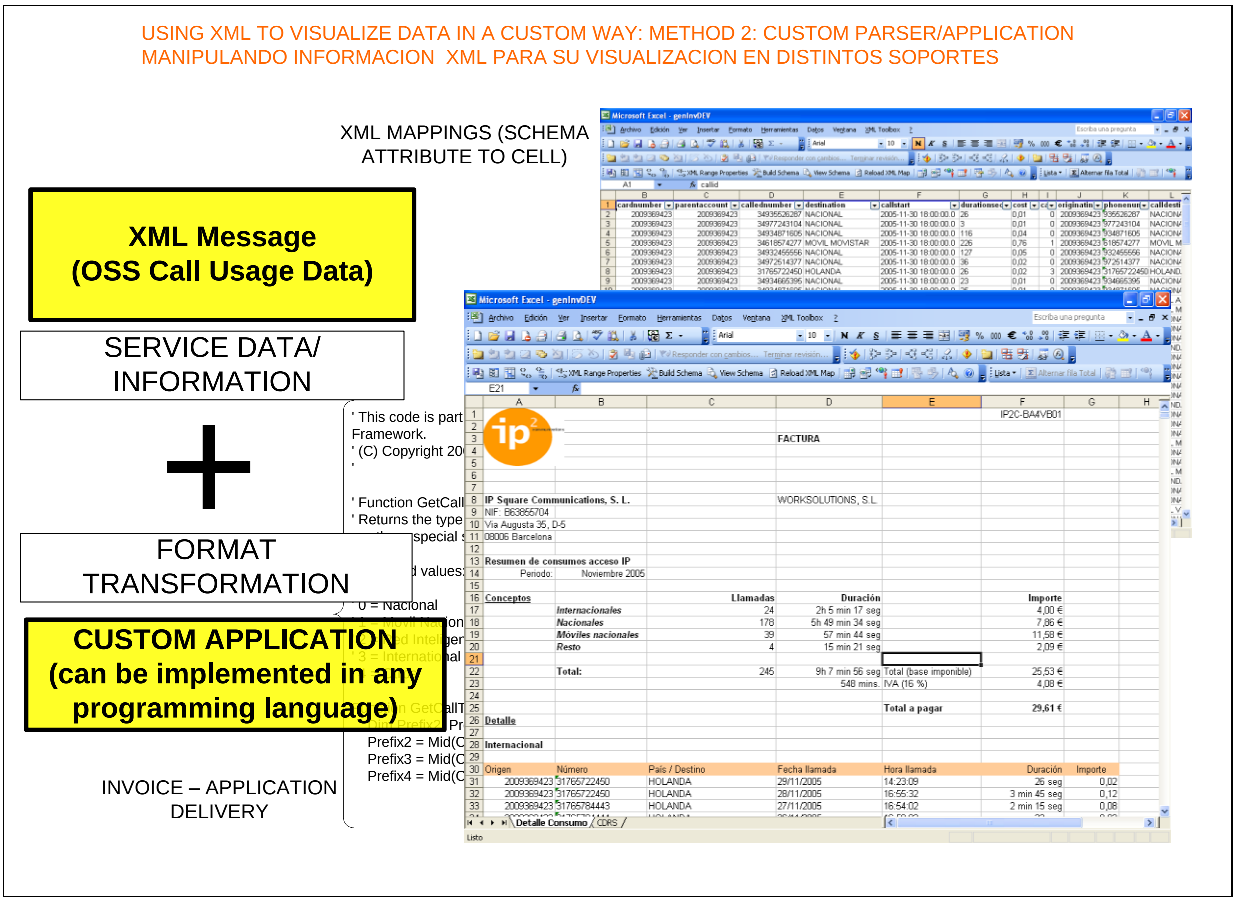
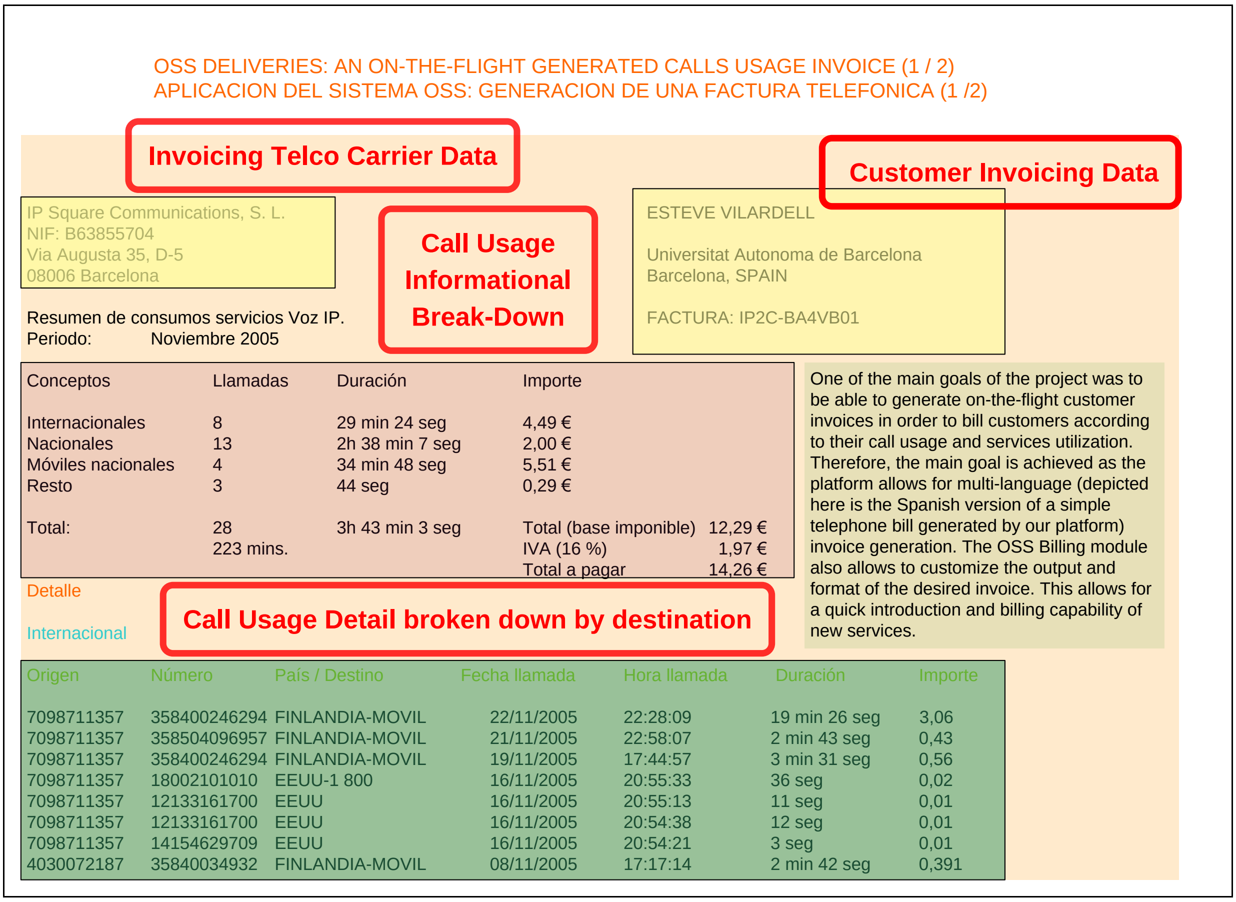


Figure 3-10: Automatically-Generation of Billing Invoices Architecture Service

Looking deeper in our path into the invoice generation example, we observe in Figure 3-10 a real example of a generated invoice. As summarized in previous sections, the XML data combined with an XSLT guideline style sheet allow us to automatically produce customer invoicing. The same applies for suppliers, vendors and partners as we are now in phase 2 of our OSS development, extending the initial scope of application to these actors.

One of the benefits of the implemented architecture is that it abstracts the functionality from the programming languages used to submit the action requests. This makes it possible to use different languages and approaches depending on the kind of final item we need to create. As the common framework basically deals with XML, any input being fed into the system shall be understood, processed and executed, whilst we can develop new functionality using any available programming language.

3.5.2 Invoice Generation Customizable Fields Usage


OSS DELIVERIES: AN ON-THE-FLIGHT GENERATED CALLS USAGE INVOICE (1 / 2)
APLICACION DEL SISTEMA OSS: GENERACION DE UNA FACTURA TELEFONICA (1 /2)

Invoicing Telco Carrier Data

Customer Invoicing Data

IP Square Communications, S. L.
NIF: B63855704
Via Augusta 35, D-5
08006 Barcelona

Call Usage Informational Break-Down

ESTEVE VILARDELL

Universitat Autonoma de Barcelona
Barcelona, SPAIN

FACTURA: IP2C-BA4VB01

Resumen de consumos servicios Voz IP.
Periodo: Noviembre 2005

Conceptos	Llamadas	Duración	Importe	
Internacionales	8	29 min 24 seg	4,49 €	
Nacionales	13	2h 38 min 7 seg	2,00 €	
Móviles nacionales	4	34 min 48 seg	5,51 €	
Resto	3	44 seg	0,29 €	
Total:	28	3h 43 min 3 seg	Total (base imponible)	12,29 €
	223 mins.		IVA (16 %)	1,97 €
			Total a pagar	14,26 €

One of the main goals of the project was to be able to generate on-the-flight customer invoices in order to bill customers according to their call usage and services utilization. Therefore, the main goal is achieved as the platform allows for multi-language (depicted here is the Spanish version of a simple telephone bill generated by our platform) invoice generation. The OSS Billing module also allows to customize the output and format of the desired invoice. This allows for a quick introduction and billing capability of new services.

Detalle

Internacional

Call Usage Detail broken down by destination

Origen	Número	País / Destino	Fecha llamada	Hora llamada	Duración	Importe
7098711357	358400246294	FINLANDIA-MOVIL	22/11/2005	22:28:09	19 min 26 seg	3,06
7098711357	358504096957	FINLANDIA-MOVIL	21/11/2005	22:58:07	2 min 43 seg	0,43
7098711357	358400246294	FINLANDIA-MOVIL	19/11/2005	17:44:57	3 min 31 seg	0,56
7098711357	18002101010	EEUU-1 800	16/11/2005	20:55:33	36 seg	0,02
7098711357	12133161700	EEUU	16/11/2005	20:55:13	11 seg	0,01
7098711357	12133161700	EEUU	16/11/2005	20:54:38	12 seg	0,01
7098711357	14154629709	EEUU	16/11/2005	20:54:21	3 seg	0,01
4030072187	35840034932	FINLANDIA-MOVIL	08/11/2005	17:17:14	2 min 42 seg	0,391

Figure 3-11: Another Invoice instance format Example

Provided that the previous concepts have been clearly assimilated it becomes clear that our engine is capable of creating multiple formats as needed. An immediate need to also be able to produce real physically-available invoices in paper quickly arose as the second real and practical application of this OSS framework. To fulfill such need, a PDF-creation object was developed to produce invoice in that format as most of the clients nowadays are used to receive all their bills, notifications and service information through an electronic medium, basically email most of the time.

Figure 3-11 depicts the real format and fields involved when billing a real user for his/her telecommunications services we provided. At the end of his/her billing period, an automatically XML/XSLT-process request is launched by the OSS platform and the whole process of bill creation, bill dispatching and electronic sending is dealt by it in a smooth and invisible way, therefore making it very easy for the platform administrator to focus on their business activities instead of on heavily time-consuming billing tasks.

OSS DELIVERIES: AN ON-THE-FLIGHT GENERATED CALLS USAGE INVOICE (2 / 2)
APLICACION DEL SISTEMA OSS: GENERACION DE UNA FACTURA TELEFONICA (2 / 2)

Rest of the Automatically-generated Invoice. Generated by the OSS Billing Module using dynamic XML information served by our BARA OSS Application Server running on a J2EE environment.

CALL USAGE BREAKDOWN

Nacional

Origen	Número	País / Destino	Fecha llamada	Hora llamada	Duración	Importe
7098711357	933905870	NACIONAL	30/11/2005	16:56:37	3 min 27 seg	0,06
7098711357	933905870	NACIONAL	30/11/2005	16:48:44	3 min 49 seg	0,06
7098711357	932325190	NACIONAL	30/11/2005	14:06:58	34 seg	0,01
4030072187	932325190	NACIONAL	16/11/2005	22:39:25	19 min 23 seg	0,23
4030072187	932325190	NACIONAL	14/11/2005	22:23:13	1h 2 min 20 seg	0,755
4030072187	932325190	NACIONAL	13/11/2005	22:01:16	5 min 37 seg	0,068
4030072187	932325190	NACIONAL	12/11/2005	14:34:04	1 min 15 seg	0,016
4030072187	932325190	NACIONAL	11/11/2005	15:32:51	8 min 41 seg	0,106
4030072187	932325190	NACIONAL	09/11/2005	21:41:04	31 min 37 seg	0,383
4030072187	932325190	NACIONAL	08/11/2005	17:17:42	3 min 30 seg	0,043
7098711357	932386505	NACIONAL	08/11/2005	12:59:02	16 seg	0,01
4030072187	932325190	NACIONAL	07/11/2005	19:00:50	1 min 1 seg	0,013
4030072187	932325190	NACIONAL	05/11/2005	17:44:01	20 min 4 seg	0,243

Móviles nacionales

Origen	Número	País / Destino	Fecha llamada	Hora llamada	Duración	Importe
7098711357	609082443	MOVIL MOVISTAR	17/11/2005	19:44:20	9 min 28 seg	1,5
7098711357	646451111	MOVIL MOVISTAR	15/11/2005	14:33:14	4 min 3 seg	0,64
7098711357	609790511	MOVIL MOVISTAR	03/11/2005	21:49:39	16 min 38 seg	2,63

Resto

Origen	Número	País / Destino	Fecha llamada	Hora llamada	Duración	Importe
7098711357	900990011	RED INTELIGENTE 900	30/11/2005	00:04:08	2 seg	0,01
7098711357	902400500	RED INTELIGENTE 902	14/11/2005	21:07:08	19 seg	0,14
7098711357	902243402	RED INTELIGENTE 902	07/11/2005	21:37:20	23 seg	0,14

Figure 3-12: Final Invoice Creation Example

In Figure 3-12 another instance, this time using a Java J2EE object to generate the submission request, is shown. As an example of a real invoice, detailed information about traffic, call duration, destination and other highlighted data is presented to the user. Once again, the magic of separating content and visualization gives the chance to implement any new service using any arbitrary programming language and IDE.

The other architecture modules follow the same approach. A new object instance is created for every need that arises from the business-side, and then it is fed into our OSS platform as a new automatically-available Web Service distributed service and as it uses XML as the communication way, everything is tied up together in a fast and very convenient way. We have introduced some of the very basic tasks implemented in the framework.

So far, the new services are introduced on a weekly-basis and the framework has to be extended very often. All the details of the exact implementation and the assessment and documentation of all the hundred objects involved in it, is out of the scope of these dissertation mainly due to space and time constraints. Documenting everything shall take a very large amount of time the author estimates. Up to this point we have walked the reader through some of our architecture ideas, aiming to transmit these and the resource-intensive need that running a telephone company service requires in both time and financial sources.

4 The System Hardware and Infrastructure: Technical Details and Topology.

4.1 The technology fundamentals: the invention of the Telephone, from Graham Bell to the 21st century.

In 1876, at the age of 29, Alexander Graham Bell invented the telephone; shortly afterwards, in 1877, he formed the Bell Telephone Company. We could adventure in our thoughts that, at the time of his invention, he could barely imagine the magnitude and scope that his invention would take in terms of technology evolution and the effects that it would bring to the human beings living in this planet. His invention kicked off the concept of remote communications that quickly evolved throughout the years into the creation of the Telecommunication Networks covering the whole world now. In the 21st century almost each part of the planet where we live is covered by a Network reaching it, using different methods of transmission and access. But the concept of global connectivity is now a reality.

The invention of the telephone started the massive rollout of communication devices during the next years, decades and centuries. The more telephones that were deployed, the higher the need to connect them among them, messed together in the concept of Network that we now come across in our daily lives. Networks did grow up and expand quickly, which extended to also sending data over the same networks, as other needs of media arose and consequently the need to send all kind of information regardless of its type also arose. Networks have evolved at light speed. In the last 30 years of the 20th century, the modern phone networks switched from its original analog form to the current digital way of exchanging information, which also evolved into bigger topologies, larger convergence and massive point of access for these networks. Finally, in the last years of the 1990s, a new protocol started to be massively used and bring all these communication networks together in what we now know as the Network of Networks: the Internet.

All started with the need to communicate as people´s deeper social needs makes them talk, exchange ideas, travel, and engage into activities that need remote communication. Then, the telephone was invented, the ball started to roll, and the digital technology took over the world as it was known before and evolved into an universal network that regardless of the place and underlying technologies is based in the concept of moving bits of information from one point of the world to the other. Digitalization of these symbols is the base for this information to flow in the world nowadays. We must thank the invention of the telephone and the development of the big Telecommunications Network by mostly the Telephone Companies during the 20th century that have brought us into the world of information that we live on these days.

4.2 Basics of telephony transmission from the 1970s to the 21st century: current legacy technologies.

Human voice can be studied as a signal both in time and in the frequency domains. Mathematics and engineers do know quite well about this kind of signals and how to analyze, convert and, in general, process them for the desired application. One of these applications is to transmit the voice signal from a sending point to a receiving point. This is the basics of telecommunication networks. The physical study and signal characteristics are out of the scope of this project and we shall assume the reader is familiarized with the topic. This project is about IPT – Internet Telephony – or VoIP – Voice Over Technologies. Basically this translates in that we are dealing with digitized voice streams being sent over the Internet or IP-based Networks. This would not be possible at all without the previous digitalization of the audio conversation into a format which can be passed over to a digital network for its transmission. Wide bibliography is available in the topics of signal processing, voice transmission and so on. For the sake of simplicity and in order to start getting familiar with the technology dealt with this, in this dissertation scenario, we shall bring us to the second part of the 20th century where the concept of voice digitalization was introduced by the Telephone Companies in order to pack many audio conversations into their transmission lines and send them all over their network. By getting into this process, they also introduced and technologically started the modern world of digital networks as by doing that, they abstracted emission and reception of information and then voice became just another type of media as the networks were able to send any kind of data as long as it was coded in digital format. We shall now describe the very fundamentals of a Telco Office circuit. How voice is encoded into digital format, the way and procedure used to send multiple audio or data streams and in general, the description of the networks as we know them since last century.

Audio is an analog signal. Analog signals can be digitized for the purpose of sending them over a medium. Telephone calls, being made every single second, by millions of people all over the world, are just a stream of digitized analog signals, converted to a digital format in order to send them through a single or multiple networks. The more networks involved, the more format conversions the signal shall suffer and therefore the more chances for audio degradation if not treated correctly.

4.2.1 Telephone Digital Conversion and Transmission Fundamentals

An audio signal can be approached as a frequency signal with a frequency threshold of about 20 KHz. As we know from the Nyquist theorem, every signal can be transformed into a discrete system by sampling it at the correct rate period. In addition to that, the medium, or transmission channels, always impose a physical constraint in the frequency bandwidth available for the signal being sent. Traditionally telephone signals started and, still now in more than 95% of the households worldwide, run on top of copper cables. In these the sampling and conversion equipment assume a channel bandwidth of about 4k, therefore normally sampling at about 8kbps for each voice or telephone conversation.

This is how the Telephone Companies began to do it to convert the signal to digital and, up to now, most of the carriers still use this method. Thus, this is the current technology in place to digitize a single voice stream. Technically and using the ITU nomenclature, this is a PCM – Pulse Code Modulation – channel. By sampling the signal at 8 KHz and dedicating 8 bits per sample, we obtain digital streams of 64 Kbps. Welcome to the basics of digital voice transmission in Telephone Networks as this is the very pillar of current transmission equipment. Almost every single telephone branch office in our neighborhoods uses this system. This is, and shall probably, still be for many years to come, the telephony transmission standard of the world. In Figure 4-1 we have schematized and summed up the process involved when quantifying electrical signals, sampling them in order to convert them to digital format and wrap that up into a universal recognized format, called PCM as we already know, that can be relayed throughout any point of a modern Telecommunications Network to deliver it to a remote user regardless of the distance between the two points.

Most of the National and International Peering Telephone Channels – voice conversations – being exchanged in thousands of telephone branch exchanges worldwide rely on this format and procedure to originate and terminate telephone calls. This had not changed considerably for about at least four decades, from mid 1960-to end of the 20th century, until recently with the arrival of the IP networks that have enabled legacy voice traffic to be carried in newer IP-based backbones. The change and transition is now quickly landing into each of the carriers and some of the big ones are already expected to have their backbones 100% IP by 2010. The remaining smaller carriers shall take years to migrate as a substantial investment in new equipment, mostly conversion gateways, is needed, but this is a non-stoppable tendency which will end up in having the world relying on All-IP networks in this century. We shall elaborate more on this topic further on as we advance in this project dissertation. At the moment we are describing the legacy technology as we need to support it in our designed platform so that we can 'talk' to the old world; talking in this concept translates into interconnecting with the telephone carriers as most of them still use last century´s digital hierarchies and digital protocols it becomes a need for us to have the needed equipment to be able to exchange voice traffic and signalization with them. In essence, we are creating a new system but have the constraint to interact with older platforms until they fade away in time getting replaced in a few decades by the new IP gear and IP-capable networks supporting the adequate levels of Quality of Service and traffic priorization. Until that requirement is achieved, the need for interconnecting with the old classical telephone networks still stands.

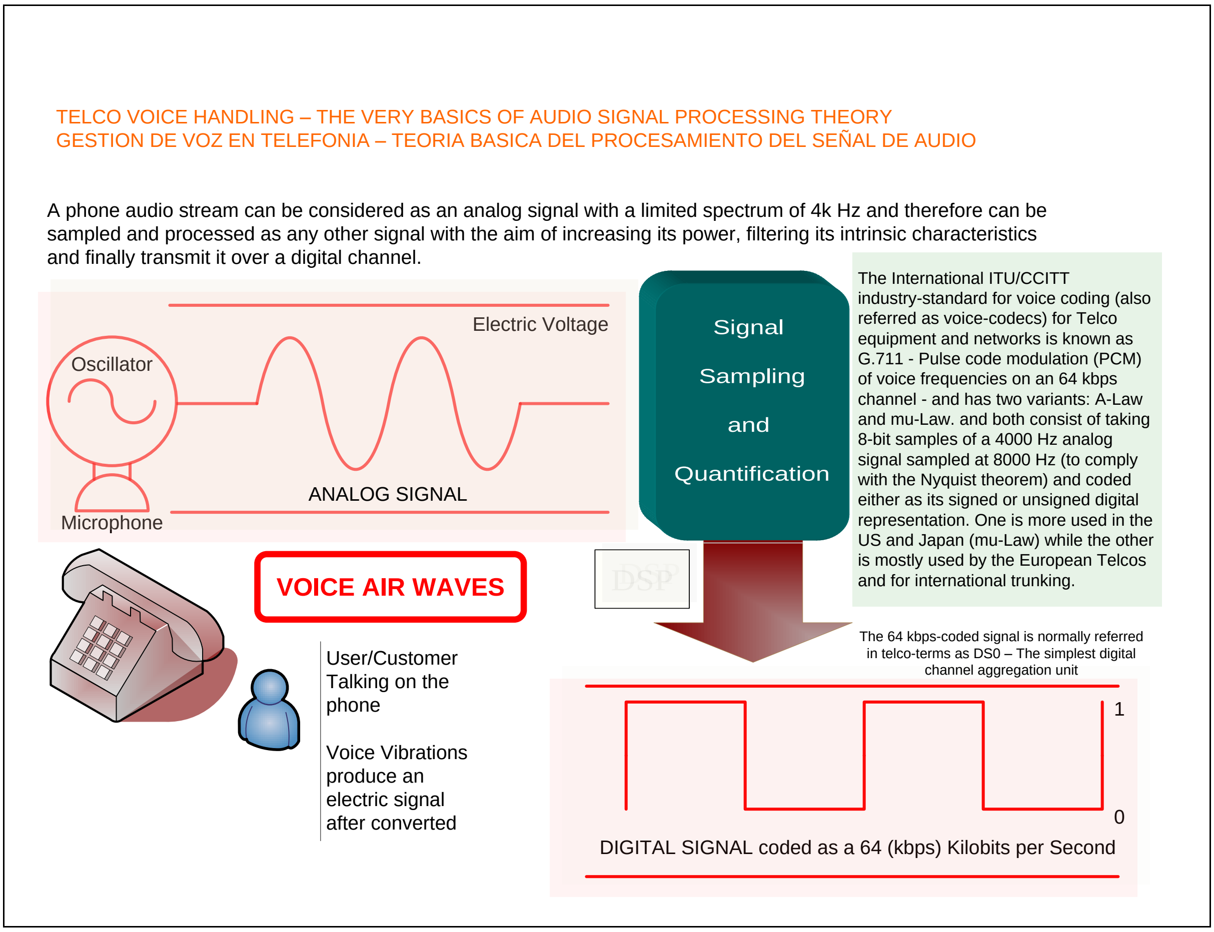


Figure 4-1: Telephony Channel Digitalization Standard - PCM

4.2.2 Multiplexation and Transmission Layers

PCM signals are normally referred as pure DS0 – The minimum Digital Hierarchy at Level 0 – channel. In countries like the United States audio signals are sampled and quantified resulting in 56 Kbps channels, but in most of the other countries the universal standard is the one from the ITU and CCITT which defines a more clearer and organized 64 Kbps digital channel. Moreover, as during the last decade of the 20th century, ISDN subscriber technologies were popular, especially in Europe and some developed Asian countries, the tendency of having DS0 channels make this digital framing the center of the de-facto digital hierarchies that we can be found in place around the Telephone companies of the world. We shall also remind to the reader, that in the context of ISDN, the basic channel is defined as BRI – Basic Rating Interface – which coincides also with the 64 Kbps coding of the voice signal. This was defined like this with the idea of encapsulating voice channels into the already Digital Multiplexing platforms in used worldwide by the Telephone Companies, therefore once again this can be perceived as a technology evolution bringing the old and new technologies to converge into an all-digital telephone world.

The CCITT and ITU international bodies outlined and consequently defined a standard meant to bring telephony into the new paradigm of digital networks while also incorporating new user and network functionalities. Needless to say, the digitalization of the subscriber and transmission backbone channels immediately allows for more generic content to flow throughout these networks: like video, digital fax, and computer data among others.

This opened the doors of the Data Networks to evolve quickly and to become what they are today: universal IP network accessible using different mediums as we shall elaborate in our further paragraphs, as we have got used to utilize and rely on it to run our daily activities. To develop this project we had to think and come up with the right topology, infrastructure and hardware equipment needed to allow our system to talk to the older digital networks while combining them with the all-IP ones, therefore quite a lot of work was done to provision circuits, define quality of service circuits, install and configure a large number of conversion equipment, voice gateways, routers, switches and so on.

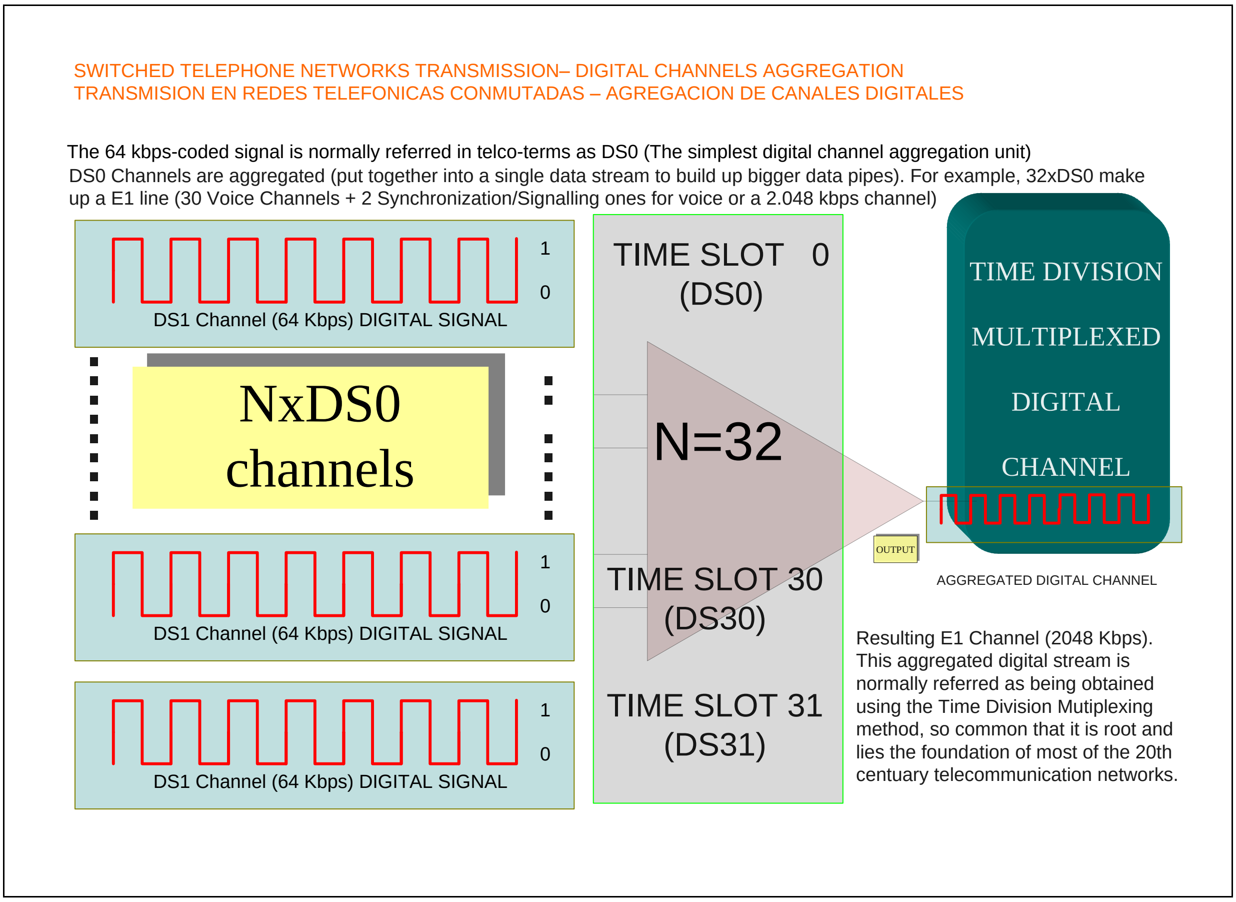


Figure 4-2: Digital Channels Aggregation – Primary Circuit Hierarchy

The sum of all these was needed for our logical software layer to act as gateways between the telephone and data worlds. In addition to this, IPT runs on top of the technologies outlined, therefore needing also a considerable amount of time to conveniently and accurately integrate them to our new OSS world.

The field of telecommunications has historically focused on studying the transmission of all kind of signals through a medium from distances ranging from small to very far. Tightly associated to processing information signals to be transmitted is the concept of modulation, multiplexation and channel aggregation. The modulation concept can be defined as the processing of the original signal in order to move it or adopt it to the medium characteristics, being the medium the air waves, radio signal, fiber optics and other kind of physical channels. Intrinsically related we immediately find the characteristic or property of aggregating multiple information channels together for the purpose of using the medium/channel at its maximum capacity by sending simultaneous channels of information at the same time.

Classical types of multiplexation are time and frequency division multiplexing, the first one has to do with sending a bit of each channel elapsing it throughout the time axis, whereas the frequency division approach uses the frequency spectrum of the physical channels to send these at a given time by using different channel frequencies. TDM – Time Division Multiplexing – is vastly used in the transmission of telephone-related signals, i.e: voice conversations or voice channels. As we introduced earlier on, a DS0 or basic digital channel hierarchy contains a live telephone conversation. TDM basically uses the idea of sending simultaneous channels thru the same medium sending each channel bit in a time-manner approach. In essence, the space between the time periods that the channel is empty is used to ‘fill it up’ with other channels/conversation data, consequently optimizing the transmission and sending a continuous stream of bits.

Figure 4-2 describes one of the most-used digital hierarchies used all around the world to aggregate telephone conversations in the developed countries: An E1 or *Primary Circuit.* It consists of the time multiplexation of 30 voice channels, sending one bit of each frame at the same time, therefore ending up with a digital frame of 2.048 Kbps as 2 additional DS0 channels are also used for signaling and synchronization purposes. The resulting aggregated result is known as a Primary Circuit and it can be a 2 Mbps circuit standard used in Europe, Asia and other countries or an American 1.544 Mbps equivalent used in Northern-America and other world locations. The resulting circuit speed is different but the processing and aggregation of voice channels is clear. Telephone Companies Equipment accepts these circuit standards as input for their transmitting and relaying processes. By standardizing these digital hierarchies equipment manufacturers are able to provide all kind of equipment for the Telephone Networks of thousands of Telephone Companies with Network backbones around the world. These circuits started to appear in the early 1970s initially used for branch and POP – Point Of Presence – branches, but quickly progressed and arrived into the final user premises afterwards.

We would like to emphasize that these digital hierarchies form the basis of any current telephone network nowadays. It also means that, for implementing our project dissertation work, we had to configure and implement our equipment with these hierarchies in mind as we did have to interconnect with many telephone companies to be able to carry out commercial business and physical interconnections with them. We deployed a system connecting two worlds: the Digital Hierarchies of the Telephone World of the 20th century with the All-IP Networks of the 21st century. This was not trivial to achieve because it means large expertise on both sides is needed. Until recently this required a lot of time and serious expertise. It still does somehow but at least, thanks to the apparition of more affordable IP gateways and IP-TDM conversion gear, the cost of this has gone down considerably. The author of this thesis at least managed to provision many circuits and interconnect to many telephone companies, first using TDM and lately moving on to doing so thru IP sessions and software switches. Important to highlight is the large amount of time needed to put these two worlds together. In the 20th century this required incredible amount of investment capital for both expensive telco equipment and to hire highly-skilled telco engineers. In this project the author managed to do the same in a much smaller scale but finally reaching a channels capacity big enough to deal with the telephone company from a point of view and business position impossible to imagine ten years earlier. This is a big achievement in the opinion of the author of this dissertation. We must be grateful to the big technology development for this opportunity to have become a reality.

4.2.3 Digital Hierarchies Standards

E1, T1 or generically Primary Circuit Hierarchies are only good to connect a relatively low number of voice channels, 30 in the case of an E1 circuit or even less in the case of an American standard T1 circuit, which brings it down to 24 channels. Due to this, the use of these circuits are mainly to reach the customer´s premises to provide that number of voice lines to the building, office, or physical location. These circuits are also sometimes used to interconnect customer´s PBX and even very small telephone branches in developing countries, but mainly its use is reserved for reaching the customer´s premises as we stated. The hierarchies need to be therefore extended to support serious traffic aggregation in the figures of thousands and even million of simultaneous voice transmissions in the way of digital channels that can, we highlight it one more time, later on be used either to carry telephone conversations or any other kind of digital information such as computer networks traffic. Thus, digital hierarchies are the essence to bring bit of data from one point to the other crossing physical and even country borders seaming less.

Figure 4-3 shows the main two types of digital hierarchies found in today´s transmission networks: plesiochronous and synchronous ones, the former were the first to appear during the 20th century and we are still cohabiting with them at the present days. The latter started to appear in the late 1990s and they have been taken over their digital counterparts taking more market percentage as times goes by. We could summarize that synchronous hierarchies are being deployed massively as they are mainly bases on higher-bandwidth medium based on fiber optics physics. As the world yells for more bandwidth and higher-transmission rates, the plesiouchronous networks let their synchronous counterparts take over and they quickly become the main underlying protocols in use in current networks.

By definition plesiochronous hierarchies are the ones containing a mechanism to synchronize their clocks on each side of the channel/medium whereas the synchronous ones use separated clocks to recover its framing and therefore synchronize with the information emitter. Most of the plesiochronous hierarchies used to be based on coaxial cables and even copper cables. As they were the first to appear and be used they used any available physical medium to let information flow from one point to the other of the telephone backbones. With the emersion of the fiber optics channels this quickly moved to transmission rates prepared for really high speed transmission and therefore a very concrete and accurate way of deriving the initial clock was needed.

Historically America and the European and Asian Counterparts have always tried to introduce their standards as the standard to be done universal, therefore always creating debate and ending up with different technologies in use depending on the physical location where the circuits are installed. Again, this was the case with Digital Hierarchies as in America the SONET – Synchronous Optical NETwork – was introduced by the Americans whereas in Europe the CCITT defined the SDH – Synchronous Digital Hierarchy – standard in their yellow book standards. Points in common are that both hierarchies are designed with fiber optics in mind, technologies to be deployed mainly in the telco backbones to connect larger clouds of digital channel aggregations: serious data transmission speeds, big enough to connect the world and bring third generation digital services to each household of our inhabited world.

These technologies are the supporting layer of information transmission in any actual carrier network. Initially used to carry phone conversations, they extended their usage to carry computer data and eventually connect circuits among them using layer 3 IP protocols. By doing this the Internet cloud was introduced. This is to highlight that the Internet and IP Networks run on top of other transmission networks: i.e.: they do run on top of Plesinchronous and/or synchronous transmission hierarchies. These are transparent for the already-used to an all-IP Network users and clients but very important as they carry the world´s traffic, at least until next generation networks such as Metropolitan Ethernet running directly on top of optic fibers are more develop and rolled out. But at the moment if somebody wants to interconnect to a telephone company to carry either data or voice information, interconnection with these standard circuits shall arise. This is what the author had to do to be able to exchange voice channels with the local telephone companies. Even if in an ideal world of IPT – Internet Protocol Telephony – connection to legacy system is not needed, in practice this requirement arises as most of the actual telephone companies still insist, require and most of them do not allow to interconnect to their backbones if not doing so by means of plugging your equipment with their networks speaking the above commented digital hierarchies standards. The tendency moves to having some big carriers totally IP by 2010-2015 but this only means that the other 95% shall still rely on these technologies to run their networks for many decades to come. Transitions are always slow and involve technical expertise and investment that needs to be justified, therefore only the big carriers do it first due to massively finding a justification in the terms of unification savings and managing synergies.

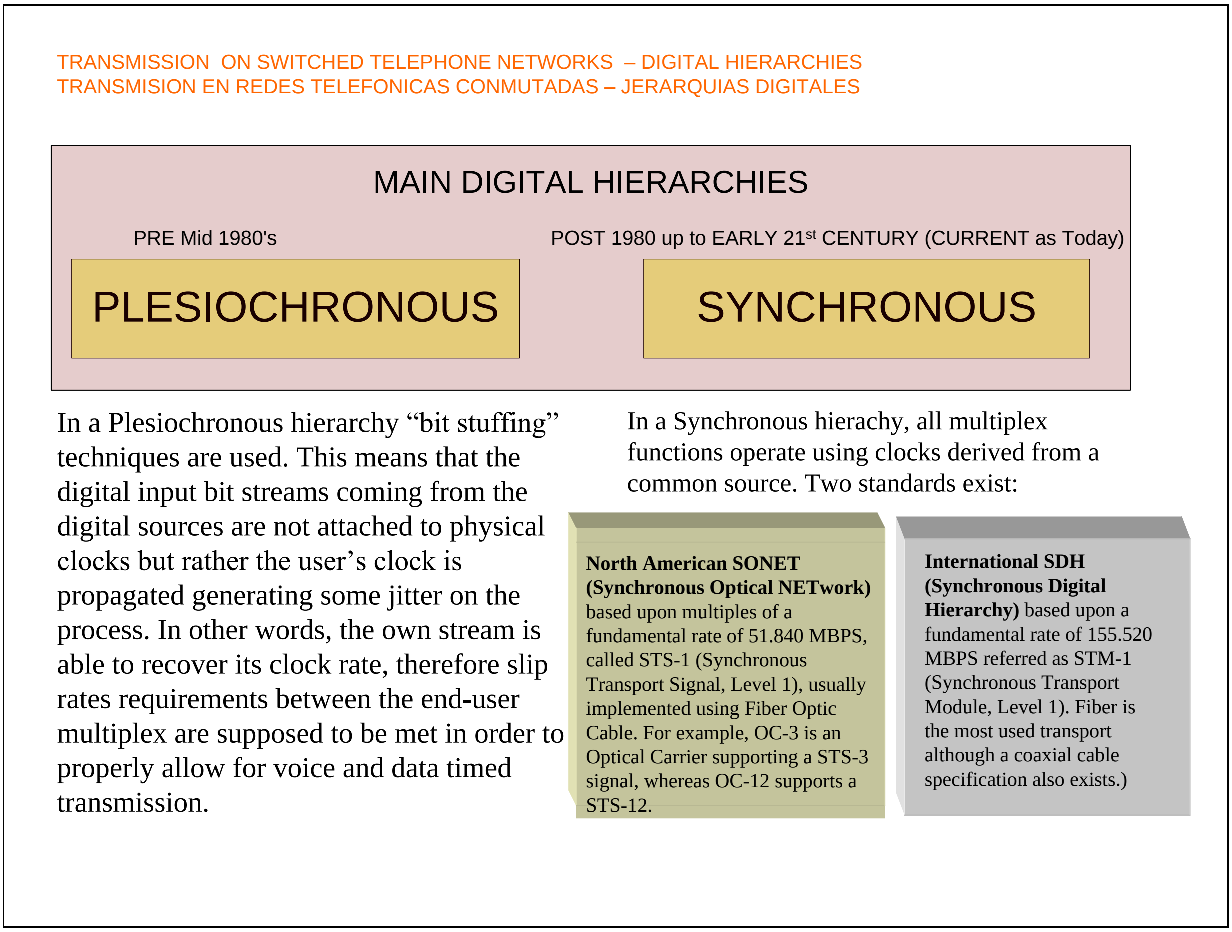


Figure 4-3: Main Digital Hierarchies in use

Figure 4-4 introduces and summarizes the most used digital hierarchies speeds of both the North American and the International (mostly used in Europe) Digital Hierarchies for the Plesiochronous type of Digital Hierarchy. They both have in common the basic 64 Kbps DS0 channel and from that channel capacity they go up in capacity by aggregating same-type digital channels. The difference between the two hierarchies is the number of channels used in each subsequent upper hierarchy, in the case of the American standard the inferior layers have got less capacity but then as the level goes up, more channels are aggregated, while the European counterpart aggregates channels in a more consistent way. The difference is only practical, as in the real world, conversion equipment exists and it is basically needed to perform international trunking between operators. A carrier lying down an international circuit out of its physical country shall most likely need to convert the framing to interconnect with another carrier following the other digital standard counterpart.

SWITCHED TELEPHONE NETWORKS TRANSMISSION – DIGITAL HIERARCHIES IN USED
TRANSMISION EN REDES TELEFONICAS CONMUTADAS – JERARQUIAS DIGITALES EN USO

PLESIOCHRONOUS HIERARCHY
DIGITAL AGGREGATION SPECIFICATION
PRE 1980's Industry De-facto Standard

The Digital Hierarchies basic element is the DS0 (64 Kbps) digital channel. Subsequent channel aggregations packs this unit into higher digital speeds. In the North-American naming standards, the DSn abbreviations are used, whereas for International, the En notation is used instead. Hence, higher digital speeds – implemented physically in most of the cases in either twister-copper pair or coaxial cable as fiber was not still highly developed to commercially proof standards until the mid 80's. That is why the plesynchronous hierarchy can be regarded as the omnipresent technology in all the main public phone operators backbones worldwide. This can be seen as the first hierarchy used to carry out digital data.

NORTH AMERICAN DIGITAL HIERARCHY

DS0	64 KBPS
DS1	1.544 MBPS
DS1C	3.152 MBPS
DS2	6.312 MBPS
DS3	44.736 MBPS
DS4	274.176 MBPS

INTERNATIONAL DIGITAL HIERARCHY

DS0	64 KBPS
E1	2.048 MBPS
E2	8.448 MBPS
E3	34.368 MBPS
E4	139.264 MBPS

Figure 4-4: Plesiochronous Digital Hierarchy

Synchronous Hierarchy is stated and summarized in Figure 4-5. As Synchronous transmission was defined with a fibre optic medium in mind, we quickly observe that starting digital aggregations ratings allows for much bigger rates than their plesiochronous predecessors. In Europe, an STM-1 or Synchronous Transport Module (STM) is commonly used to interconnect high-speed links between National IP Network Operators and International ones. In other words: for high-capacity trunking for either voice channels, pure IP traffic or a mixed of both types. In the US, the STS-1 performs the same analogue task. Higher rates aggregations are used to interconnect either whole countries or very large telephone companies and the large number of POP and interconnected circuit mesh is completely covering the world and it is the backbone connecting us to the present world

In this project, and in order to have our software platform, designed for this dissertation, and also in order to talk to the real telephone companies of this world, the author interconnected with some large carriers and operators using TDM circuits, basically E1 circuits as the author is based in Europe. Subsequently the voice channels needs did increase as the commercial viability outlined in this project succeeded and more channels were needed to give service to the platform and services users. Thus, at this moment the situation of already running out of capacity has been reached and that is why we are now looking to use STM-1 or probably IP running on dark fibre to expand our bandwidth to deal with more simultaneous voice conversations.

The exact details carried out during this project scope are unfortunately impossible to describe in detail as the down-to-the-field implementation has taken months of real work configuring E1 circuits, E2 circuits, routers, access switches, circuit patching between the author's platform and the telephone company's circuits. The provisioning of each single circuit requires a long and very considerable amount of time since the order is committed to the telephone company until the actual circuit is provisioned by them, and later on a transmission equipment is needed to be connected in order to manage and control the circuit and then the whole thing needs to be plugged into our platform to control billing, quality of the voice being carried and so on. The author also had to apply for a government's telecommunications license in order to do the full roll-out of the system. In other words: it is not exactly quickly to come up with a telephone company even if implemented using a software solution with the IPT world in mind. But one thing is clear: it takes less time and less resources and investment using the approached described in this dissertation, as years ago to achieve the same would have required out-of-reach investments for a single person, whereas in this dissertation the author has managed to launch a commercial service using the ideas, software and hardware and infrastructure systems mentioned alongside this dissertation.

SWITCHED TELEPHONE NETWORKS TRANSMISSION – DIGITAL HIERARCHIES IN USED
TRANSMISION EN REDES TELEFONICAS CONMUTADAS – JERARQUIAS DIGITALES EN USO

SYNCHRONOUS HIERARCHY
DIGITAL AGGREGATION SPECIFICATION
AS DEFINED BY THE ITU/CCITT

North American SONET

(Synchronous Optical NETwork)

Synchronous Transport Signal (STS)

STS-1	51.840 MBPS
STS-3	155.520 MBPS
STS-9	466.560 MBPS
STS-12	622.080 MBPS
STS-48	2488.320 MBPS
STS-192	9953.280 MBPS
STS-768	39813.120 MBPS

International SDH (Synchronous Digital Hierarchy)

Synchronous Transport Module (STM)

STM-1	155.520 MBPS
STM-3	466.560 MBPS
STM-4	622.080 MBPS
STM-16	2488.320 MBPS
STM-64	9953.280 MBPS
STM-256	39813.120 MBPS

Figure 4-5: Synchronous Digital Hierarchy

4.2.4 Backbone Technologies Evolution

Technology is unstoppable and that is known by every single serious engineer who loves and carries out his work in a professional and serious way. We love technology and we were born to use it, develop it and make use of it to invent and develop newer services one time after the other. That is the nuts and bolts of invention I would say.

During this dissertation the author started using Digital Hierarchies telephone equipment to integrate the work shown on this paper with the real world. Alongside his work technology evolves so fast and impressively optimal that legacy technologies are becoming obsolete pretty fast. Telephone Digital Hierarchies shall still last for many decades to follow, but newer technologies such as Metropolitan Ethernet linking sites spread far away thousands of kilometres and shall also turn these technologies into dinosaur ones. That linked with the current omnipresence of IP networks, translates into the fact that no matter which underlying transmission protocol information is used to reach one point: in the very near future, computers, telephone companies, telephone switches and end-user devices shall only talk a converged simple protocol: IP.

Given this point, the author confirms one more time that VoIP/IPT is the way to go to design and implement a modern telephone service or telephone company. No more the focus is to be given to the transport technologies but to the carrying protocol. Due to this the software managing side that the author implemented and named BARA becomes the very heart of the system therefore abstracting transport layer from the application and controlling one. This is one of the main characteristics that makes the system scalable and adaptive to the coming technologies: it is abstraction.

For the implementation of the interconnection between the BARA platform and the telephone companies' voice backbones, the author had to implement and deal with most of the technologies outlined in the associated Figure 4-6. For instance, all the circuits carrying voice in the BARA platform are either trunking with Pleosinchronous circuits, ATM ones and also to other newer IP-trunking capable telephone companies allowing for direct Switched Giga-Ethernet trunking. We therefore highlight the large amount of technologies needed to implement our system in a real world scenario.

Bottom line here: our IPT platform needs to interact with each single company and backbone out there: therefore needing to have Pleosincrhonous equipment, Synchronous one, Ethernet access switches and the right configuration and quality tools to be able to assure the correct Service Levels on each technology circuits as this is one main target to be reached on both the voice quality and up-time availability sides.

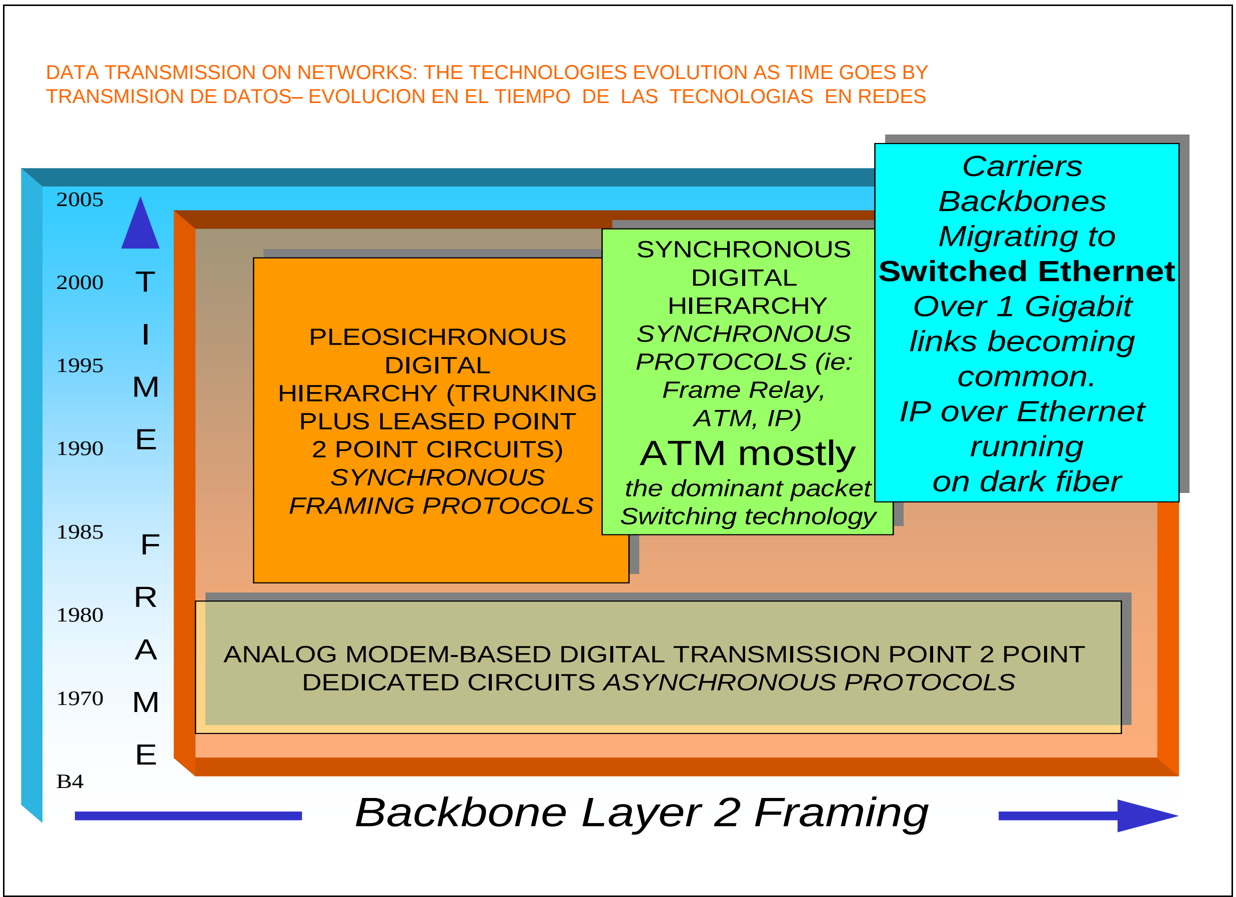


Figure 4-6: Transmission Backbone technologies Evolution

4.2.5 Telephone Infrastructure in our real-world Scenario of converging network technologies, the hardware and network modules.

This dissertation comprises a whole range of topics, from the application software side, to the hardware systems, network infrastructure, and down-to-earth daily maintenance and paperwork tasks. No real system attached to a business is easy to implement and run. The author of this dissertation did a full rollout of both software and hardware systems which took him a very considerable amount of time. The underlying main technology in use in this dissertation is Voice Over IP – VoIP – also known or referred in the telco business space as IPT or Internet Protocol Telephony. The OSS design initially in the author´s mind had to accomplish the tasks of controlling the IP modules, functions and protocols needed to implement a real business-like Telephone Company. Only a few decades ago even the consideration of building up a telephone system or service from the scratch was unavailable and totally out of reach for any non-large organization entity. This changed with the invention of VoIP/IPT and the author has tried to use this incredible technology to deploy an OSS platform and incorporate his own Telephone Company following the Spanish Business Laws.

This is, somehow, a small dream come true, the author´s dream of one day use his skills and knowledge to work on what he is enthusiastic about. With this main idea in his mind, he did go ahead and use the available tools out there to construct a telephone service backbone. Thanks to the environment of the All-IP Networks and great backbones and bandwidth almost-everywhere this was achievable; but still the task of putting all the modules together was needed. This is what this dissertation is about as the previous chapters covered the design of the software controlling part of the system whereas this chapter tries to describe the hardware, network and binding components needed to analogy put the physical supporting pieces in place.

VOICE BACKBONE PROTOCOLS USED AND THEIR COMMUNICATIONS STACK RELATIONSHIP / PROTOCOLOS DE VOZ UTILIZADOS Y SU INTERRELACION CON EL STACK DE COMUNICACIONES

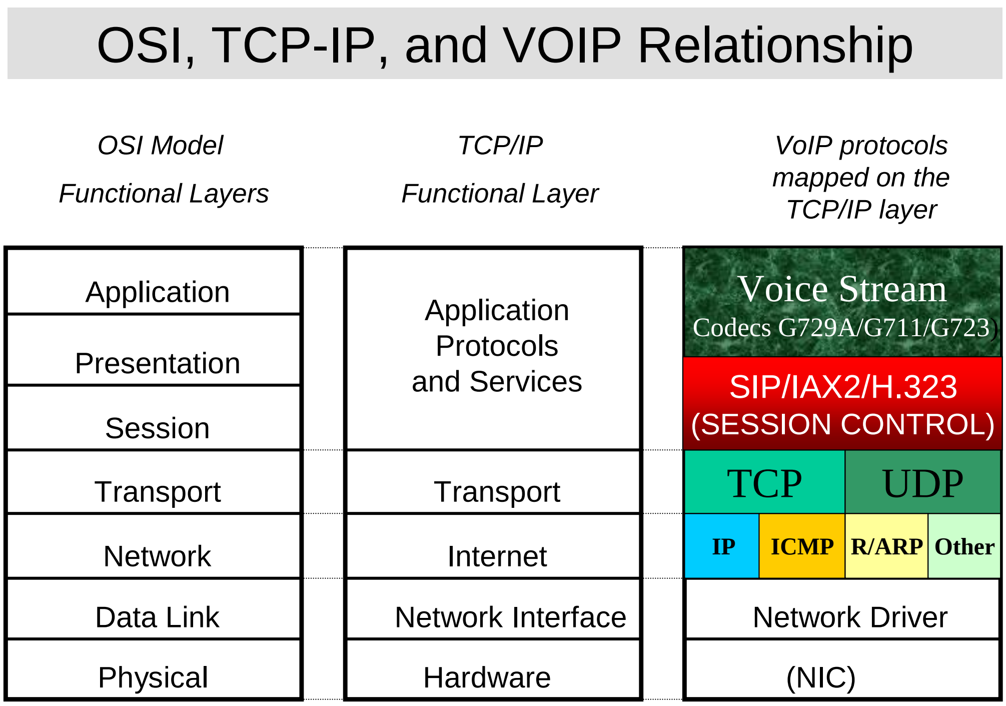


Figure 4-7: IPT/VOIP Basic Protocols

Figure 4-7 outlines the OSI, TCP-IP and VOIP Relationships. What this means is that by looking at the figure we can quickly distinguish and associate the different layers of network, transport and application levels to determine where the Voice Over IP protocols come up and in which layer they do operate. Our software layer, introduced and implemented in previous chapters only deals with the presentation and application level of the seven layers of the OSI stack. IP Telephony protocols such as SIP, H.323, IAX2 and other session control protocols allow initiating and exchanging telephony-related voice circuits' information, therefore establishing calls from point to point. SIP stands for Session Initiation Protocol, H.323 is a CCITT standard used for telephony signalling and call establishment between switching systems using IP networks, and finally IAX2 is a proprietary Asterisk (open source PBX software) protocol fulfilling the same functions as its SIP and H.323 counterparts.

4.2.6 Hardware Equipment in Use supporting the platform

Some of the hardware equipment in use to implement the physical part of this dissertation is the following:

- *Cisco* Routers and Access Switches.
- *Asterisk* open source PBX software acting as calls gatekeeper and controller.
- *Voipswitch* (VPS) Class-5 Soft switches acting as the platform front ends.
- *Digium* TDM E1 TDM cards used to provide the connectivity with the TDM world.
- *Dell* Servers, basically telco grade-level rackable 1U servers in charge of housing our software and application level pieces.

All the above pieces of equipment would be useless without our OSS platform running on top of them which acts as the Network Intelligence and controlling engine of the whole system. By gluing all the pieces together the BARA OSS applications make sure everything runs smoothly and an under controlled and supervised manner.

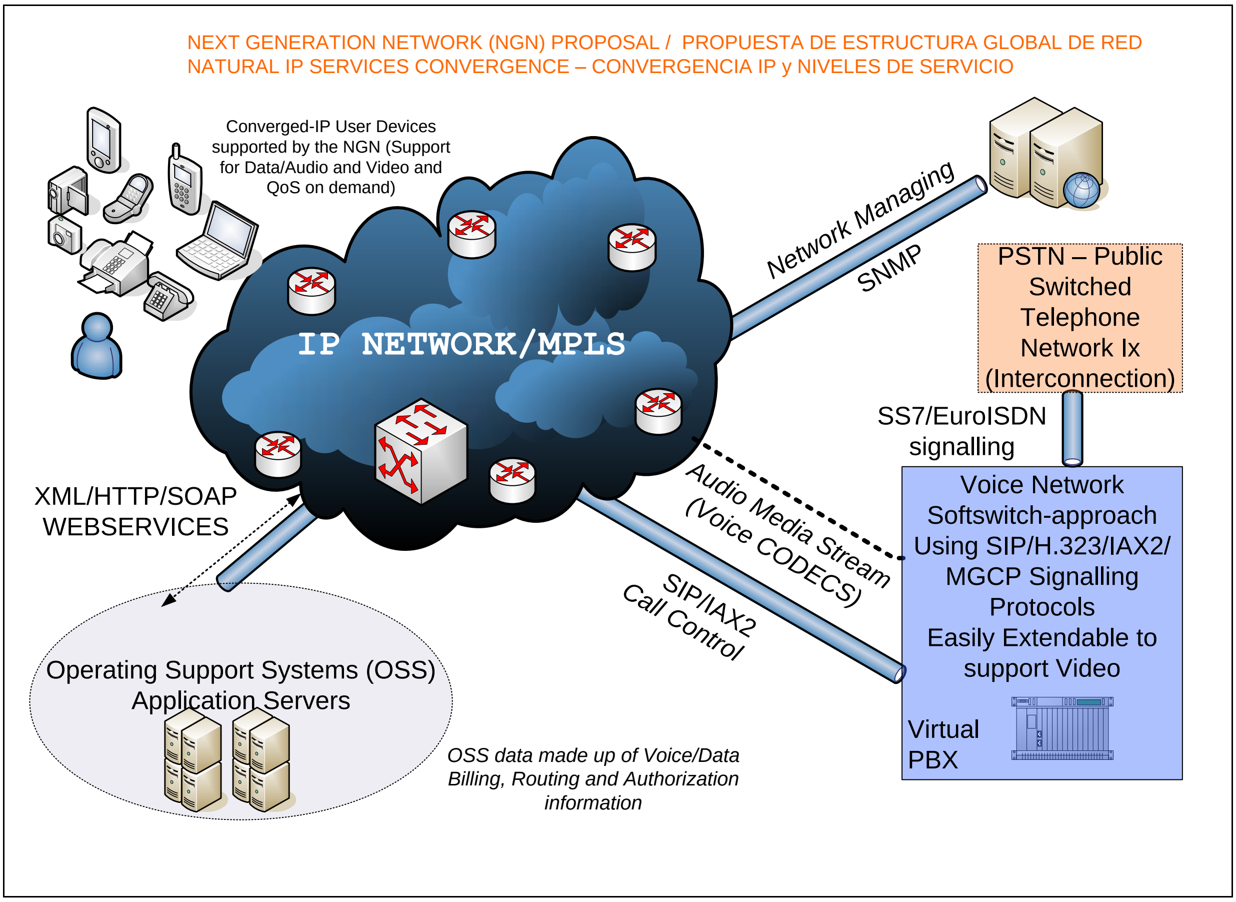


Figure 4-8: Next Generation Network (NGN) Systems Approach

Now, we shall look into all the components we did need to roll out in order to build the production system. The author opted to follow a NGN – Next Generation Network approach to design the way the underlying network and telephone signalling components would interact among them.

The first step in any network topology design is to clearly identify the components which are needed and how they shall interconnect and interact between them. In our case, and by following an own adaptation of some NGN ideas, the author based his production network design on a system mixing together network level, application level and call establishment and signalling systems. Figure 4-8 clearly outlines, first of all, the Operating System and Support Application Servers, the ones that shall control the operation of the platform. Then the network backbone is based on a universally available All-IP network as the interconnection among the local systems use IP to communicate. Also the application and presentation protocols in use are from the IP universe as this is a VoIP-based design.

User and remote end-client or third-party IP switches and telephony-aware devices shall come in from the IP cloud of our architecture. Our IP cloud must follow the general rule of providing Quality of Service- QoS - features in order to assure that voice is prioritized whilst flowing throughout our internal backbone. Moreover, we must ensure that when trunking with third-party Telephone Companies for the purpose of voice traffic exchange.

Thus, the concepts and general ideas are clear: an IP Network with QoS, an OSS platform managing the system in a distributed manner, and any kind of soft switches and IPT-capable devices to deal with the end-to-end transmission of IPT streams using the mentioned SIP, IAX2, H323 or other possible IPT protocols supported by our abstraction OSS layer. Finally an SNMP – Simple Network Managing Protocol – monitoring server or platform to capture and generate alarms on quality issues, which shall also communicate with our OSS platform in order to pass it the adequate messages depending on the Service level defined by each application service.

Our OSS application, as we know from the previous chapter, internally speaks HTTP, SOAP and it exchanges information using the XML standard. Webservices are the key to extend this platform in this future as we shall briefly outline in the ending chapters of this dissertation when bringing up the possible advances, enhancements and future of the platform.

The magic of this Telephony IP-World is that any old, current and future device can be supported; this ranges from old telephone handsets using IP gateways, new IP phones, totally new IP telephones even supporting Video streams, PDA, mobile phones and every device whose output can be converted to IP and its audio stream transmitted on the application layer on top of IP.

We are aware that the OSS module of the architecture basically shall initially cover the Billing, Authentication and Routing of the calls established by the IP end-devices or IP-switches interconnected to the system. As a whole, this “cloud” allows for voice traffic to be exchanged among heterogeneous sources of telephony content. From the audio world of the signal processing field, it is important to remind the reader the concept of *codec,* which can be defined as the format or defined framing that an audio stream is digitalized, modified and turned into a transmittable format. Different codec formats exists from the mentioned PCM, also known as G.711 codec, G.729, G.723a, GSM – widely known or at least used by all of us when making a telephone call using a mobile handset, and so on. New codecs are developed as time goes by as innovation in this field never stops. Only important to bear in mind is that on top of the application level IPT protocols, the negotiated codec specifies the way the actual audio stream is flowing throughout the network from point to point. Dozens of books can be written about the different subjects commented so far. Each of the aspects involved in this project could itself have its own specific project allocated as many variables and technologies come together to be able to implement an IPT Backbone.

We shall now highlight one of the modules drawn in Figure 4-8: the PSTN Interconnection Gateway. PSTN stands for Public Switched Telephone Network as we are aware. This is basically the universal telephone network available worldwide. Anyone wanting to be able to make or receive a call from a PSTN user must first interconnect with the network at one of its edge points. The IPT world is still too new to contain a considerable large number of telephony users; in human words this means the following: there are not a large number of people using IP networks to initiate or receive their telephone calls; the number of these is quickly extending on a daily basis and one day the PSTN shall cover and integrate the IP world into itself. But nowadays, in 2008, there are more than 2 billion telephones lines in the world that rely on the PSTN and obviously people in these locations need to be reached by an IPT platform as otherwise communication would stay inside an intranet.

Interconnection with the Digital Telephone Hierarchies in use by the large telephone companies is therefore a requirement for our platform; without it none of our users of either our plain telephone service or of our additional voice services running on top of it would not be able to reach external users sitting on the universal PSTN.

4.2.7 Interconnection Agreements – PSTN Side

The author of this thesis therefore did also have to establish interconnection contracts and agreements with both National and International Carriers in order to connect this dissertation platform with the real world out there. Among others, the interconnections that were made, and that took a few months to be negotiated, provisioned, configured and put into production were the following:

- ONO - National Carrier
- COLT – Cable Of London Telecom, interconnecting with their Voice backbone in the Barcelona, Spain POP – Point Of Presence
- Verizon/MCI – American Carriers, interconnecting also with them on their BCN POP.
- Orange/France Telecom/ALPI, again, interconnecting with them at their BCN POP.

Never-ending or better defined as very long-timely negotiations are needed to interconnect with large telephone carriers. From economic factors, to interfaces definitions, to equipment installation, tuning and configuration is required to do so. Also, humbly, quite a bit of patience and large amounts of motivation is advised as dealing with long-established Telephone Companies which is not a straightforward process. Furthermore; even a National Telecommunications Operator License is needed. In Spain the process of requesting and being assigned one Carrier License is delegated to the Spanish CMT – *Comisión del Mercado de las Telecomunicaciones* -. A serious part of this dissertation was to deal with them as otherwise the implementation of these ideas would not have seen any real materialization.

On a second phase the author also interconnected this dissertation platform with the following carriers:

- Telefónica de España
- British Telecom
- Teleglobe

And also negotiations with the following operators are underway at this time of writing:

- Jazztel España in Spain
- AT&T in Miami, FL, U.S.A.

We introduced the Digital Hierarchies in use by the large telephone companies so that we could at this moment bring all our infrastructure pieces together. Now, the reader of this dissertation shall understand why the TDM and Digital Hierarchies Gateways are needed for our platform to talk to the telephony world out there. Without interconnecting to legacy switches, legacy voice networks and without talking to the large telephone companies carrying international and national traffic it is impossible to deploy any production business-like service.

This is why a deep understanding of these technologies was needed to first understand the scope of this work and later on be able to bring this idea into the reality that it is now: a commercial platform interconnected with different telephone companies and exchanging traffic with them, moving to the next phase of actually being able to sell international traffic to the big telephone companies as well, moving from a client to a provider of services, something impossible to imagine decades ago. In Figure 4-9 we define the nomenclature first and then we list our phase-1 interconnections with different commercial telephone carriers.

We put in place a heterogeneous range of telecommunications equipment for each single interconnection carried out. For some telephone companies, E1 interfaces were used; for others, more advanced telephone companies, we were able to use direct H.323 or SIP IP trunks among others. And for some of the phase 2 (our future interconnections as we start to be in need for larger and more capacity circuits) probably we shall have to interconnect using SS7 – System Signalling Number 7 -, a CCITT standard in use to arrange signalling information among telco circuits when in trunking mode.

As far as the interconnection with the PSTN is concerned, we have introduced the basics to the reader of this dissertation. Further details exist and, again, if described or even outlined, they shall take a large amount of documentation as work includes technical interconnection details that covers several CCITT standards.

We shall leave that information as out of the scope for this work as we do not want to get so much into detail. Worth saying then is, that interconnection is done using legacy circuits and technology and the new All-IP paradigm or *network convergence* shall take a few more decades to arrive. Intrinsically interrelated to this concept is our communications backbone: we have designed our system using IP technology but then converting legacy systems by the means of protocol gateways to and from this IP world. By doing this, we have merged both network and cohabited our scenario with the possibility of making calls on each side of the communication channels either on the IP or PSTN worlds.

Our platform backbone has to be: concise, reliable, and scalable with a high grade of resilience. For the physical deployment of the POP – Point Of Presence – site where this dissertation system was to be rolled out, a secure and reliable location was selected in Barcelona, Spain. The place where the hardware and network equipment was to be installed was in Carrierhouse, Telvent premises in the outskirts of Barcelona, in the Zona Franca area, in the same building where carriers like Orange, Jazztel, AL-PI, Telefónica and so on also have their transmission and communications equipment, therefore in the author’s opinion a good site to host the physical systems involved in this project.

100% uptime is a requirement in our mission-critical production environment. Telephone Networks inherently need to support a very high grade of resilience, mainly duplicating or having multiple instances of the basic systems available across the system. A Telephone Backbone cannot afford to be out of service as the service that is giving is a universally available one. As such, users have got used to an almost full one hundred percentage uptime and this is a standard in the Telco world. Bottom line: any new system aspiring to become a player in this niche and very difficult specific world need to have the same SLA constrains and fulfil them in an adequate and constant way.

This requirement was the one the author followed when choosing the providers of IP connectivity for the OSS platform. A single provider was not enough as it would not therefore be compliant with the resilience requirement. As a result, and after many negotiations, technical and financial analysis, the platform was interconnected at the IP level with the following carriers:

- Teleglobe, an international Carrier provider also of IP international traffic peering.
- Intelideas, a Spanish ISP
- COLT – Cable Of London Telecom, providing us with also IP traffic peering capabilities.
- British Telecom, providing the building inter-access switches fibre links

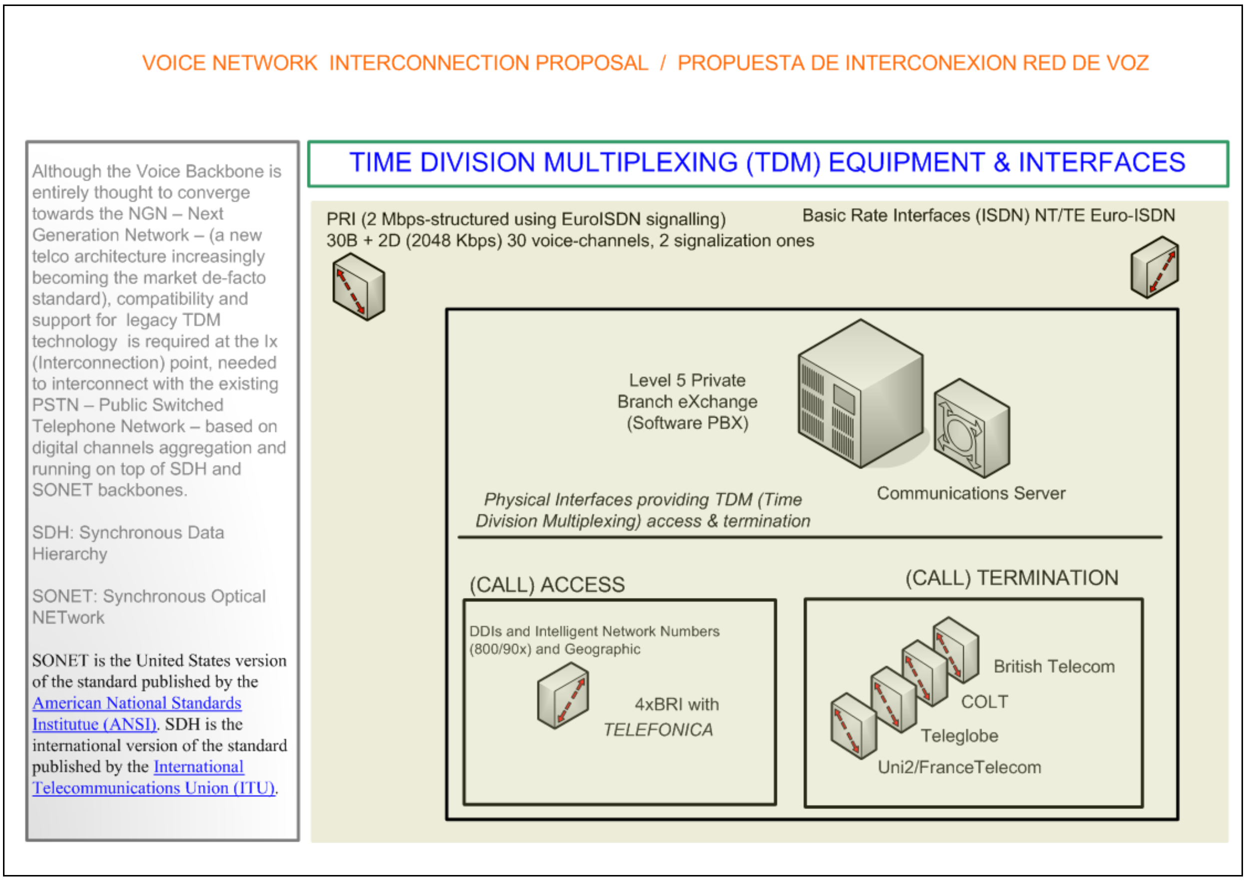


Figure 4-9: Platform Interconnections

4.2.8 Interconnection Agreements – IP Backbone Side

Figure 4-10 summarizes the present (or at least the phase 1 as we are already evolving quickly and many changes happen on a daily basis) IP International Network Connectivity: i.e.: the IP backbones providing us Internet connectivity with the rest of the Autonomous Systems on the Internet.

BGP-4, or *Border Gateway Protocol* 4, is the one configured and in use in the platform´ routers to do the routing with other Autonomous Systems on the Internet. What this means in down-to-earth words is that this mechanism takes care of making sure our servers, and IPT equipment is almost reachable from our users, clients and peers accessing from the International or National side.

One of the platform services allows for the trading of voice traffic; this means our platform buys and sells voice channels or *voice routes* from/to other carriers. Obviously as a client can also be a large carrier and consequently capable of sending a large amount of simultaneous calls and due to the fact that the benefit of the service depends on the amount of calls dispatched every day; one can deduce that the uptime of the system also contributes to the commercial viability of the system as it generates revenue due to the service it is performing.

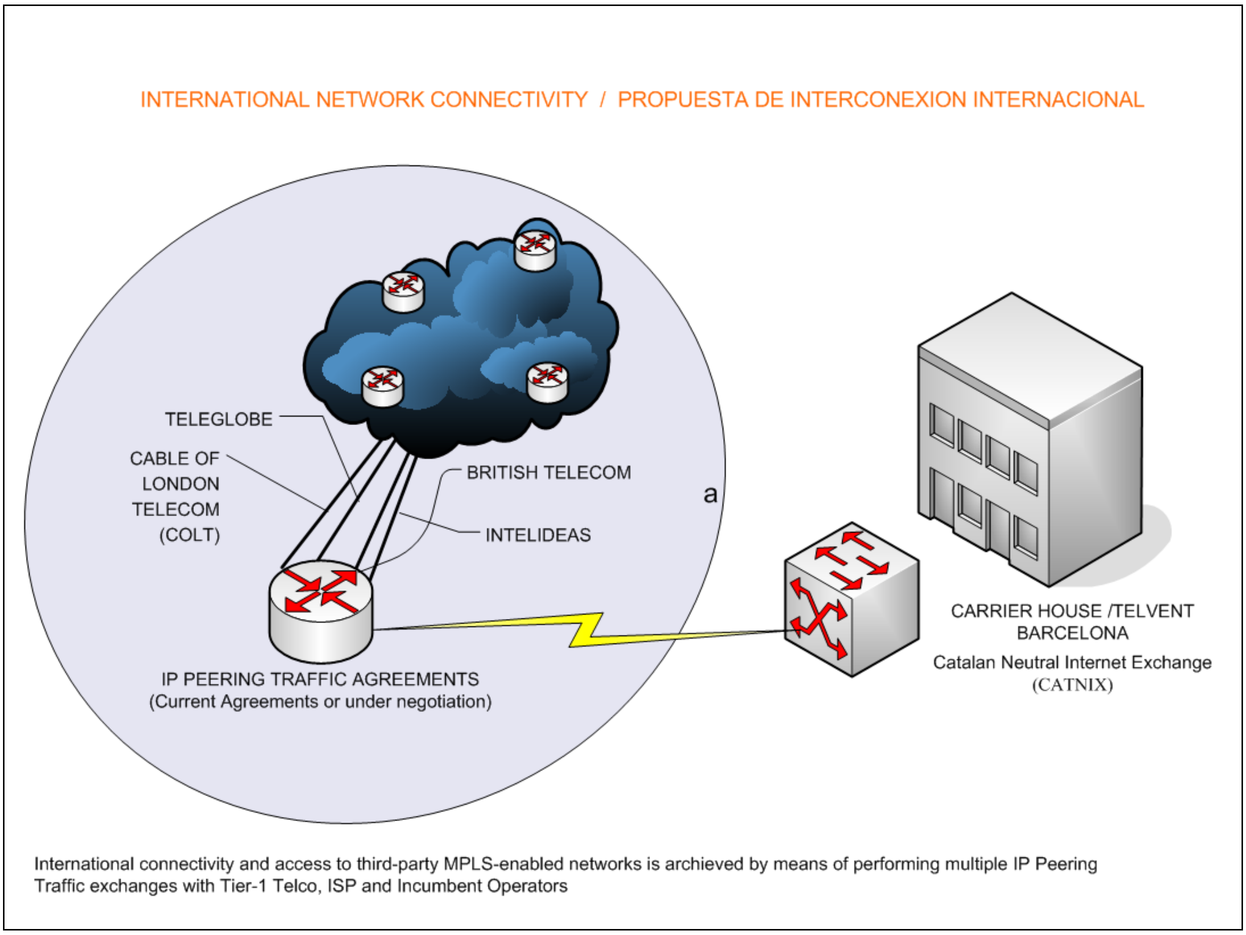


Figure 4-10: International Network Connectivity

4.2.9 Platform Hardware Support Systems – Topology Layout

As this system is currently launched as a commercial one and associated to the author's incorporated company and, also due to security concerns, specific detailed deployment information is regarded as non-totally public; but regardless of this the generic topology design is just an abstract idea whose implementation is the foundation for supporting this dissertation voice OSS platform. Our Physically deployed in systems scenario is depicted in Figure 4-11. We utilize a load-balancing layer-four load-balancing service to relay voice establishment requests and other signalization tasks to our registration servers behind our balancing server. This was first implemented using a DNS-round robin technique and later on implemented in a hardware-way by configuring the layer-4 balancing in a Cisco Router equipped with the appropriate routing operating system.

The DMZ – Demilitarized Zone – is the security perimeter acting as a security and protection barrier between the outside world and our backend voice systems.

Our backend systems are the server in charge of hosting our OSS platform, implementing web servers, database services, directory services and other support ones.

Finally, we need to mention the very important and critical gateways in charge of handling the establishment and signalisation of calls and its conversion from the TDM world when appropriate.

By balancing all the service requests we make sure the platform inherits the level of resilience that we need to provide a mission critical service to the users of the telephone service.

Figure 4-11 depicts the abstraction of the infrastructure topology and functions of the servers in place. Real implementation varies as some of the services at the end of the day may be running alongside others in one or multiple physical servers, sometimes also in a virtualized machine mode where more than one operating system is running in the same server, therefore allowing for a better use of the hardware resources.

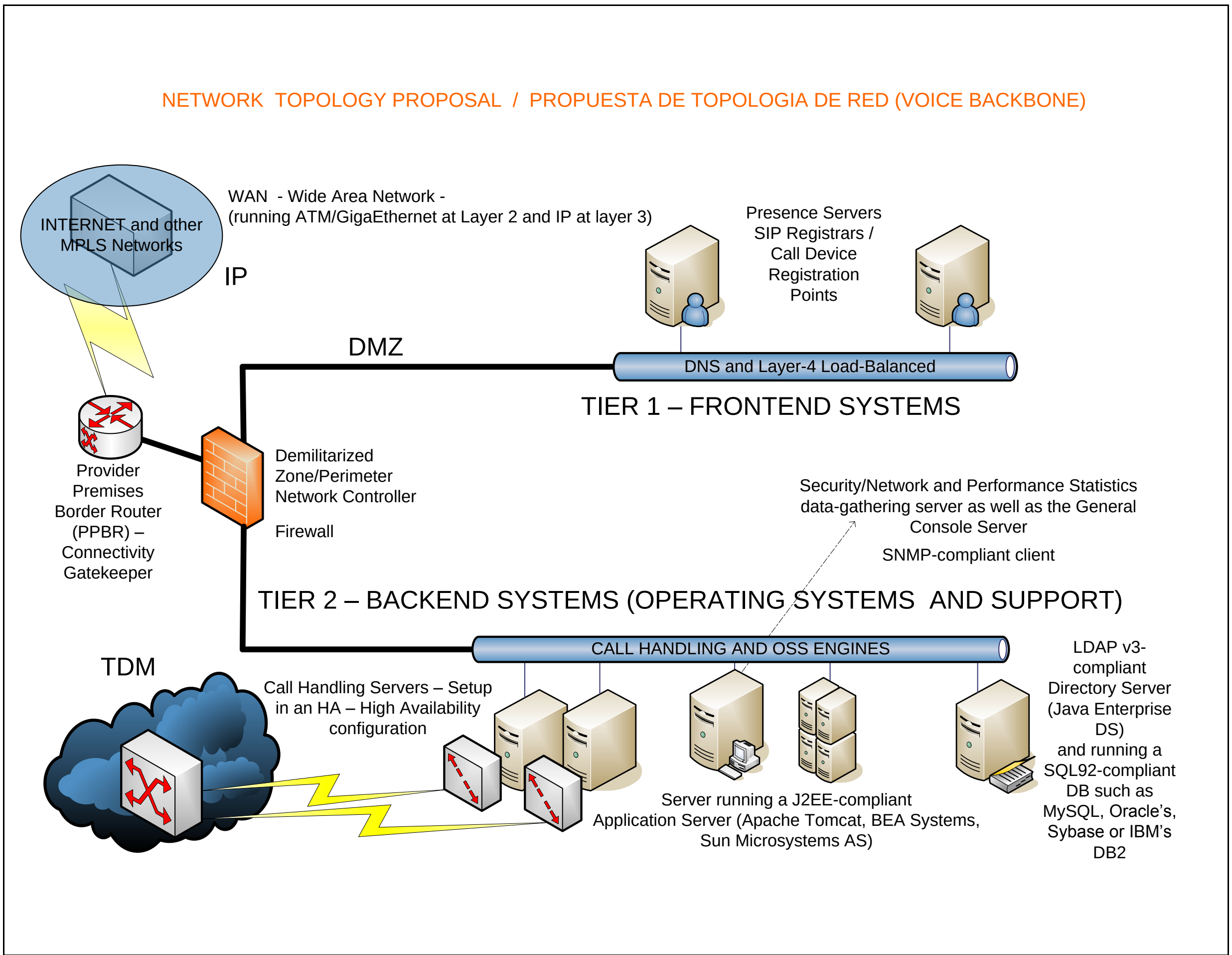


Figure 4-11: Network Topology for the Platform Voice backbone

Last but not least, to mention that the OSS platform abstracts many of the underlying functions. Worth mentioning is the database service in our software layer as BARA – our OSS application – supports various third-party databases such as MySQL, Oracle, LDAP-V3 directory services and others such as IBM DB2's and Java DB. The only condition is to implement the associated plugin or Abstract Interface for the required database. That is why multiple database servers running different database instances are also depicted in Figure 4-11.

4.2.10 Actual Hardware Deployed

All the concepts, ideas, technology backgrounds, software design and subsequent developments finally materialized on a real production scenario running on different hardware and located in a physical location. For the sake of introduction and educational purposes, we outline on

Figure 4-12 the systems map of our hardware deployment as it was during the roll out of the initial systems. The system inventory and equipment is a living creature as new systems keep being added on a periodical basis especially to provision new circuits and interconnections as well as providing further resilience to the software layer when needed.

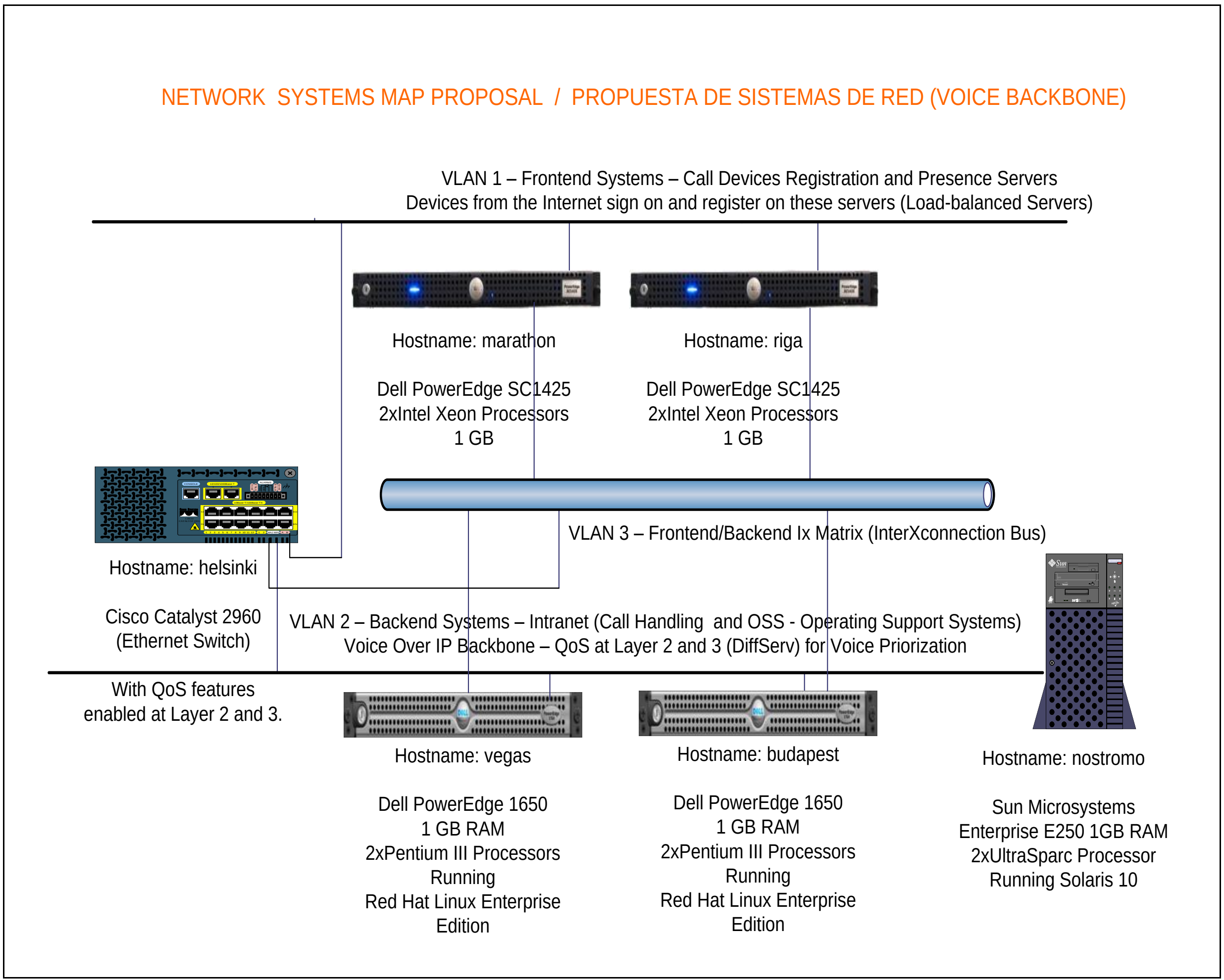


Figure 4-12: Network Systems Map

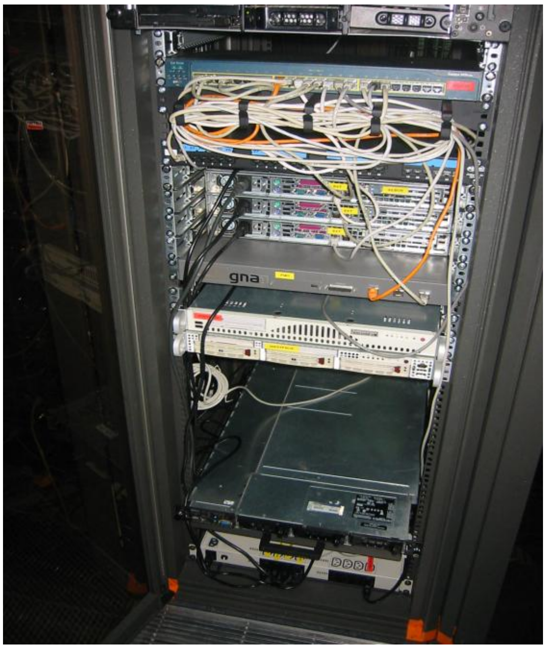
Above we can observe the six initial servers that were installed to give service to the whole platform and hosting all the dissertation services and ideas as well as serving as an interface between the IPT and the TDM world. These servers run on a 24h shift, powered by two redundant power supplies, double Ethernet interfaces, and connected through access switches to two Cisco catalyst servers not outlined in this picture but acting as fault-tolerant network routers to make sure network connectivity is always up and running at it is best level.

4.2.11 Production Day: The Idea Becomes a Reality

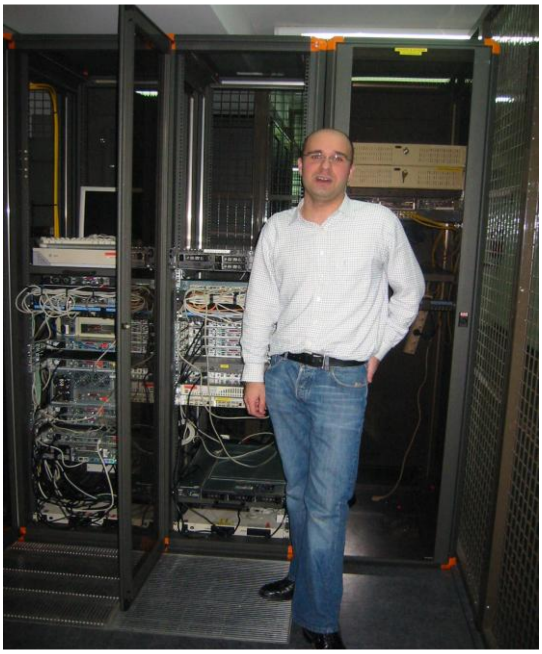
The work presented in this thesis is not only a theory in the mind of the author; all the work introduced along the previous chapters and sections materialized months ago when the platform was connected, the software installed and the services defined and explained in this thesis started to operate. So far, more than 1 million of telephone calls have passed through the system, and more than 20 million of minutes have crossed the boundaries of this dissertation system. The number of voice minutes; i.e.: number of minutes that the calls last and real users talk on the phone is doubling every month and wholesale traffic is also looking like it is growing in a non-linear way. Negotiation with foreign carriers and wholesale traffic aggregators are under way with the goal of interconnecting with major players to highly increase the number of simultaneous calls processed by the platform described in this dissertation.

Figure 4-13 and Figure 4-14 show the photos taken during the configuration of the platform on the first 'production' day when it was switched on. The photo is only included to backup the concept of the large amount of work that this dissertation did take, in terms of design, hardware installation, networking and so on. But finally it did pay off as the system is holding to its promises and acting as a small telephone company as the original idea was.

SYSTEM SETUP DAY: MISE-EN-PRODUCTION EVENT PHOTOS DAY (1/ 2)



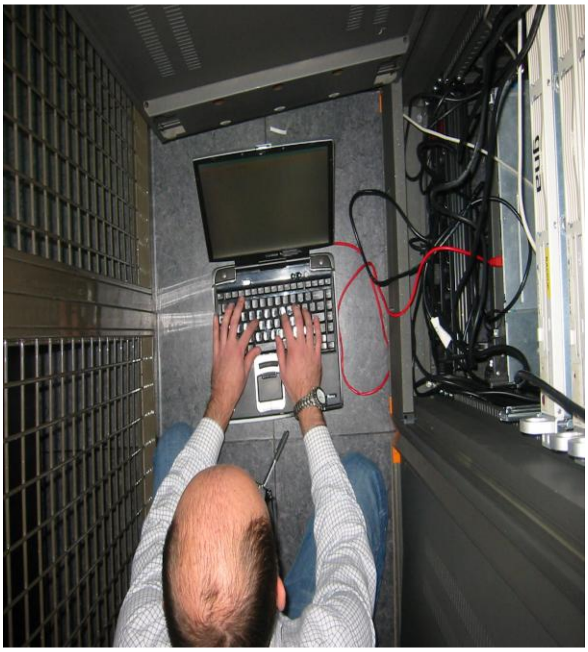
Main systems rear-closet prespective: Servers are configured and physically installed.



First minute of systems booting up and staying in production from that precise moment onwards.

Figure 4-13: Production Day System Go-LIVE Photos 1 / 2

SYSTEM SETUP DAY: MISE-EN-PRODUCTION EVENT PHOTOS DAY (2/ 2)
CONFIGURACION DEL SISTEMA EN PRODUCCION: FOTOS DEL EVENTO



Attaching my laptop to the servers for local service configuration.

First production server (vegas.ipsquare.net) plugged into production on March, 14th 2005.

Figure 4-14: Production Day System GO-LIVE. Photos 2/2

5 Actual Services Running on the Developed Platform

Real solutions are for real-world problems. The platform conceptually introduced and developed in this dissertation, would remain just an idea if it did not support real-world requirements in the form of commercially available services. Since the very beginning of this project, it was decided that the aim of the platform should be to start immediately serving its purpose of supporting commercial services. Among the wanted properties sketched out during the initial conception phase, the remarkable ones were scalability, easiness of rollout and flexibility of supporting heterogeneous services. These three characteristics can be identified throughout the next paragraphs as we introduce and describe the services we have deployed in a production environment controlled, supported and supervised by our platform.

These services include an All-IP VoIP End-User devices service- pure VOIP over the open Internet -, a Calling-Card platform also running within the platform scope and finally a PINLESS-like system running on both landlines and GSM/3G mobile handsets. All three services running on the underlying OSS, we developed in this dissertation, and therefore proving that the original goal of implementing our own Telephone Company-like Operating System and Support System has been achieved. As these run on a Voice-Over IP – VoIP – backbone, we fulfilled the design idea with a real implementation thus achieving our ambition of deploying Telco-services with fewer needs in terms of equipment, software and financial resources.

5.1.1 End-User Access Technologies in the 21st Century

We shall walk the reader through these services in the next paragraphs. Prior to this, an introduction to the current state-of-the-art in IP access connectivity is to be introduced. In the early 21st century the broadband connectivity panorama in terms of access networks providing high speed connectivity can be summarized as shown in Figure 5-1.

The Access Methods to IP Broadband resources are summarized in this figure. These cover more than 70% of the current developed-countries Internet/IP technologies. These are the technologies that the end-users utilize to connect to the Internet, to IP networks and consequently to gain access to the world of VoIP and to the magic of being able to carry out their voice conversations on top of the IP-converged networks.

We have chosen our platform to use an IP-All Backbone to act as the underlying layer putting all the pieces together, thus for this to work each user of the system consequently has to have the ability to access this IP layer. By using any of the mechanisms available and depicted in Figure 5-1 the user can access the services supported by our platform. Initially the very straightforward first-application that comes to mind is to allow the users to use Voice Over IP devices: including software soft phones, VoIP IP Telephones, PSTN-VoIP adapters – such as adapters for conventional phones, faxes, PBX and so into the world of IPT – Internet Protocol Telephony -.By doing so, we encourage the use of the Internet and private IP networks – for instance, any corporate VPN network used in large corporations worldwide -, to carry voice traffic in addition to the old-classical data-only services.

Nowadays, IP-Convergence translates into the fact that not only the Internet has become a common place for devices to meet but it also has allowed all-kind of companies to connect their branch office together in a mess architecture on top of these IP networks and use them to run both their data and voice services on top of them: by doing this, an automatic financial benefit is accomplished as the return of investment in the networking equipment now becomes faster as it automatically starts supporting data and voice networks. By migrating old-voice networks to IP, any kind of organization that used to rely on two different networks, one for data and one for voice, can quickly realize the benefits of running only one network for all their services. This is the magic of an

All-IP Network or, what is becoming known as the Convergence of Networks into the IP World. It is on top of this very simple idea that the author of this dissertation started to develop his ideas. Without the convergence of the data and voice networks into a unique common world this would not have been possible at all. By all accounts this is a great benefit achieved in the 21st century as even the big Telephone Companies; they do all have plans to reach a 100% All-IP Converged networks within a 20 years period. The networks of the future shall all run on IP backbones and the exploited service in this dissertation is the property of being able to carry voice traffic on top of IP. This work is the living proof of the synergies and advantages of this convergence period we live on.

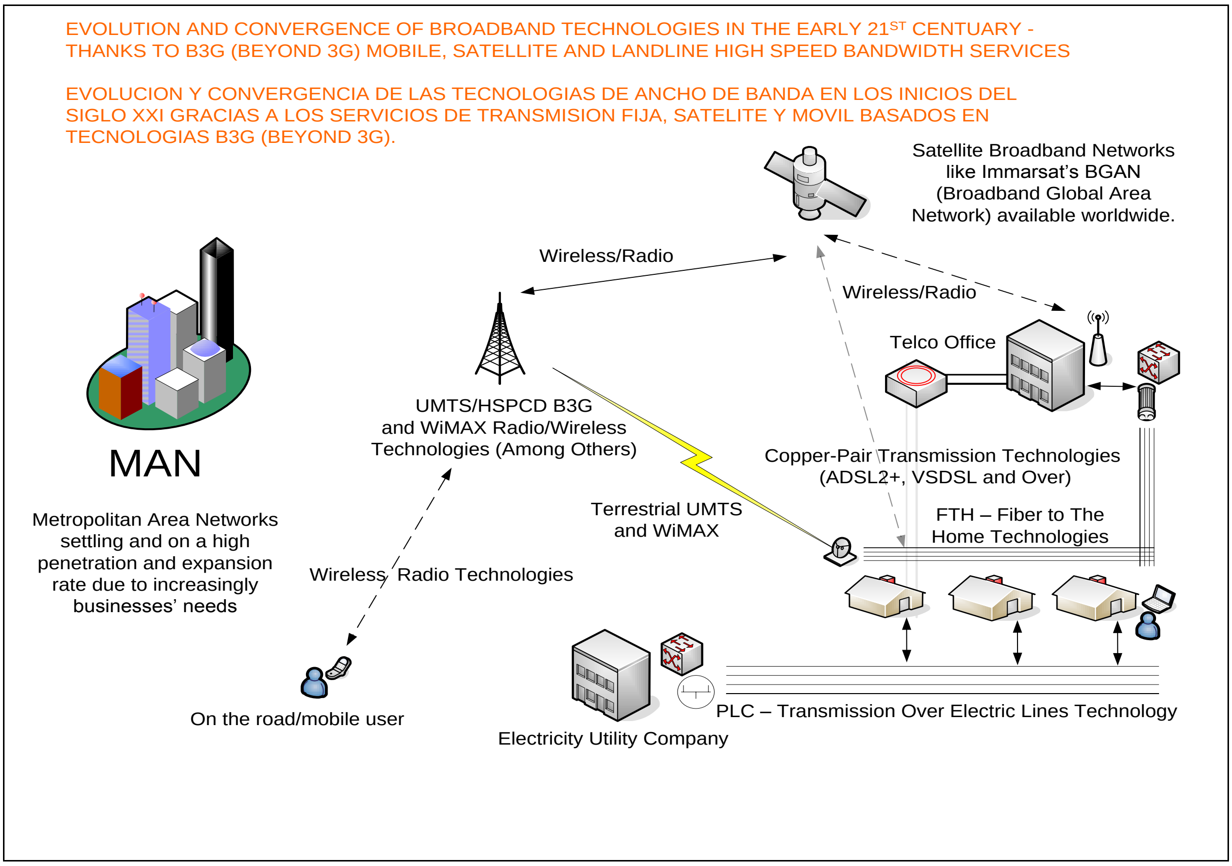


Figure 5-1: Broadband Technologies All-IP Convergence

5.2 An IPT controlling platform: VOIP Devices and Soft-Switches intelligence (ITSP Deployment)

The first service deployed under the platform´s supervision was an ITSP – Internet Telephony Service Provider -. This is the analog of an ISP but providing IPT services for the users: i.e.: any client using a VoIP device can register to the ITSP and perform telephony functions such as calling, receiving calls, call forwarding, voice mail services, access to IVR and other voice-related applications. Among the end-user devices we can mention Cisco IPT telephones, Linksys VoIP phones, Linksys Analog-VOIP adapters like the Linksys PAP2, Cisco 8xxx phone series and any device supporting registration against a VoIP service. These including SIP or H323 enabled devices. The range of user devices which can use the ITSP services includes single-user devices, up to powerful PBX or even Class-5 Soft Switches. Both end-user devices and software evolve as time goes by but the platform is supposed to be device-independent in the chase of thriving to provide a complete supporting solution.

In an initial beta phase, carried out in the beginning of the platform rolled out, fifty users were given access to the service, later on; the service moved to its commercial milestone and started supporting a large number of offices and users located worldwide. Beta-Users were located in different locations like Hungary, the U.K., U.S.A, Paris and Barcelona. At the end of the Beta phase the system proved to be stable and ready for a real rollout. The nice thing of having an OSS in our backend systems supporting us is that it becomes very natural to keep on adding users to the service. This is how the ITSP service evolved and it now supports up to 50 other Class-5 Soft Switches Peers, or interconnections. Translating this into human numbers we shall say that the ITSP part of the platform is daily performing more than 45k calls, and through it more than 200.000 minutes of voice minutes are processed. It handles voice traffic from places like Spain, Hong Kong, Cuba and hundreds of other destinations and it is interconnected with dozens of other IP carriers for voice traffic exchange. The statistics show that voice traffic keeps on growing monthly on an exponential rate, therefore very soon the system is expected to process 1 million voice minutes every month.

We moved from supporting an initial number of fifty test users to giving service to other ITSP, exchanging traffic with other carriers and having dozens of different TDM and IPT routes for each destination that the platform is serving. For the reader to have a physical idea of the magnitude of the systems supported, we encourage the reader to think of it as a software layer bringing together thousands of telephone devices from multiple parts of the earth. Hundreds of voice conversations are carried out using the platform which is controlling the telephony engine part. It can be thought as an engine in charge of making use of the IP backbone plus a layer of software that together oversees the behavior of a modern small telephone company.

Figure 5-2: Example of VOIP Devices available in the Market

The evolution in the development of VoIP end-user devices is too rapid to keep track of all the supported and available devices in the IPT Market that they continuously get developed and commercially launched.

5.2.1 Common Platform Support Attributes

The platform defines and internally gives support for some objects aimed to be useful for the services running and defined around the system. The very basic pillars of these objects shall be defined in the next sections.

5.2.2 User Accounts

An object encompassing and wrapping together the abstraction of a system user includes information about the user´s privileges, access and services quotas and thresholds, as well as reaction to service events and policies. i.e.: the platform supports validation, authentication and nomadic user presence from a device and geographic independent location.

5.2.3 User Groups

Groups of users can be tied up together and associated services policies defined and associated to them. Users groups uses are, among others, to put billing information together, to be able to define closed user groups as internal voice extensions, special privilege users and pre-authenticated ones. User Groups allows us to associate a set of users to one of multiple systems, therefore restricting some services to these or alternatively, very narrowly, define which entities can access which services and under which circumstances and environment.

For example, in the following to be introduced PINLESS CLI-authentication service, some users get granted their access to the system by identifying and consequently authenticating them given their origin calling number. By using user groups to define the list and permission of these services members, the system is capable of distinguishing the voice capabilities given the user´s Direct Inward Dialing Number and Caller Line Identification. Then, wrapping these parameters alongside the group information, we are able to split the users into closed user groups. A consequence of this is that we are able to associate, for example, different default language, system voiceovers, different balance and credit information to each granular user. This is the conclusion of the previously introduced concepts and design as seen in the previous chapters of this dissertation.

5.2.4 User Associations

Each user has got an association or link in our platform objects repository, therefore each user belongs to either a default user class or a customized service user class. This clearly identifies the permissions and services a user is able to access. By having a multiple amount of different service rates, class of service, and other parameters this gives us the advantage of being able to put these heterogeneous objects together to build a new service in a smooth and non time-consuming way.

5.2.5 Rates and Authentications Levels

The current system has got about a hundred rates loaded into the database engine tables. As the platform supports granularity-by-service, this means that a service owner is able to define his/her own set of rates and change them and/or manage them when required. Authentication is performed using multiple attributes such as user-number, Caller Line Identifier, end-user device network identifier, mobile SIM card and so on. Regardless of the integrated authentication class acting at each moment, the platform abstractly is able to authenticate a given user and grant him/her the session information depending on the authentication level or policy in place.

In down-to-earth words, the above means: we can authenticate users using different access mediums: either user calling from a VoIP Telephone, a user calling from a mobile handset from a GSM or UMTS network, or even a legacy PSTN client using any telephone in the world to get access to one of our platform services. By abstracting authentication and defining service policies what we have achieved is to very clearly mark the separation line between access network and abstract service. A good platform does not have to be very tightly bound with the specific details of the technology that it supports, but rather it has to rely on abstraction objects giving it the capacity to always having the capability of supporting new services despite the technology evolving or their specific implementation details changing. By using a good design this level of abstraction turns out to pay off as it enables us to bring telecommunication services to the world in a promptly way.

5.3 A Pre-Paid Voice Traffic Platform

The second real-world example of a one service running of top of this developed platform is the Pre-Paid Voice Traffic Platform, also referred classically as a Pre-Paid Calling Card or PIN-based Platform. By printing, distributing and selling calling cards in POS – Point Of Sales – locations, a commercial system based on selling these cards and giving the buyers users to either national or international calls is introduced. A commercial service was rolled out with this service description in mind. So far, more than 20.000 calling cards were distributed and sold. An associated estimated 1 million calls were made in a 1 year time interval and customers like the Barcelona Turisme government entity commercially did take care of selling these cards to the public. Some of the calling card designs launched as products can be seen in Figure 5-3.

Since the early 1970 Calling Cards platforms and software have been available. This is one of the very classical services that almost any telecommunications company in the world has in its portfolio. The author deployed and launched this service using the described platform though instead of relying in very expensive and legacy-software. As the author designed a system with the aim of supporting as many voice systems as possible this became obvious to do, too and therefore a visual design of some calling cards was made and the service was commercially launched. Since then, the platform has supported it in a stable way and fulfilling its purpose correctly by also giving support for an IVR module, multi-language and multiple-rate support. Some of these concepts and others shall now be described in the next paragraphs. Figure 5-4 shows some of the artwork designs used for the commercial launch of the calling card service in Barcelona, Spain.

Figure 5-3: Commercial Calling Cards Marketed

The Calling Card service allows any person buying one to make international calls from any landline, mobile phone and public booths. The difference in this case is that the underlying system is the one the author has designed and implemented instead of relying on other third-party more expensive telecommunications software. Thus, one of the aims of this project of saving in software costs has also been achieved. By using our own platform to deploy our own services we become independent from very costly software.

Figure 5-4: Calling Card Artwork/Designs

Users or, in this case, most appropriate to refer them as, *clients* as this is inherently a very commercial service, buy their PIN tokens cards in any given POS. Later on, they scratch the back of the card and through that action have access to a PIN code. Then, with that code, the client can use any telephone in the world to make calls worldwide. The only thing he/she needs to do is to key in the PIN and the system grants him/her with a service amount of voice minutes to be used during the call session. As the platform supports multiple DID – Direct Inward Dialing – numbers, a central or distributed platform can give support to thousands of Calling Card services regardless of their location. Besides, we implemented a localization multi-lingual capability that translates into the fact that also regardless of the client´s language, country of origin and actual location, and the platform is able to interact with him/her through an IVR – Interactive Verbose Response – system that takes care of any language and special location requirements.

The author incorporated a company in Spain called *IP Square Communication, S.L,* ,which requested a telecommunications license from the Spanish Telecommunications Agency in order to be legally eligible to distribute and resell telecommunications services such as the one attached to deploying a calling card service like the one advertised in Figure 5-5. Every real commercial service requires a strong sales campaign and marketing support material as well as product lists, constant updating and, generally speaking, a considerable amount of both human and procedural resources to keep track of sales, advertisement, marketing, distribution and so on. All this had to be developed alongside the launch of the calling card services to market.

Figure 5-5: Calling Card Marketing-support material

Figure 5-5 shows some of the designs commercially being used, whereas Figure 5-6 shows a brochure sample of the advertisement of the calling cards used in the POS – Point Of Sales – locations.

Figure 5-6: Calling Cards Designs

Figure 5-7: Marketing Sample Material for Calling Cards

5.3.1 XML-Based

All the internal communication going on in our platform is based on the XML Bus architecture introduced earlier on. Billing, Routing and Authentications events are described in XML and passed from and to the modules involved in any given service running. Thus, again, the platform acts as the joint between communication and service objects/attributes with a policy using implemented objects defined for each production service.

5.3.2 Network Intelligence Features.

The platform supports a wide range of Network Intelligence features, these defined as the functionality that is able to act differently depending on the access network, end-device, or service-specific rules in place. Remarkable features in this category supported by the platform are: different behavior by DID – Direct Inward Dialing -, also different behavior depending on the user device, user language, user location and generally depending on any object we want to act upon. We would like to highlight again that the underlying technology is not a key-point here, it does not matter if the service is running on top of a TDM backbone, VoIP one, IP direct trunks or any other voice-bearer. What is important and gives its greatest advantage to the platform is, again, the degree of abstraction that the platform is capable of conducting. We list some of these features in the following subsections. We enclose these as an introduction and practical examples of the platform features.

5.3.3 DID and Multi-DID support.

To fulfill the needs of many services that still rely on the legacy PSTN, the OSS platform, developed in this dissertation work, needs to be able to support some of the features of the public network. Remarkable is the need to be able to distinguish calls coming from different DID – Direct Inward Number – By providing support for different services running on top of different physical circuit and logical telephone numbers, the platform allows to implement multi-functional and customized services accessible by using different system entry-points. A very clear and quick example of this is, for instance, a service with different associated PSTN numbers, one for each language that the service supports, therefore whenever the user dials the desired language DID, the system passes the language variable to the OSS platform and the service session language is set to the requested language. Currently as the world has become very global this is a very demanding requirement in every single telecommunications-supporting platform available in the market. The platform outlined here has been designed from the very early stages with this need in mind.

5.3.4 Multi-Lingua features.

Currently, the introduced Calling Card service was launched with support for English and Spanish voice-overs (languages). The commercial service is running with support for these two international languages. Internally, the platform's IVR accepts multiple languages activated on request by the platform service objects.

In the services running at the present, multi-lingual support is omnipresent. In the case of our ITSP service, any pure IPT user can access their account information in the preselected language. For instance, one user can be told his/her remaining service balance in the language he/she specifies, and also all the call establishment and informational/error voice-overs are given in the user´s language, too. International Telecommunication Companies record and give the users´ notifications in their country language. The author did keep this in mind and therefore the implementation of the platform accepts and, at all times, gives the user the information in his/her language. This allows future services to be developed regardless of the geographical location. This requirement was taken into account and all the services in place and introduced in this project honor this rule accordingly.

The author shall like to highlight the large amount of time required to implement this module itself. It was time-consuming as for every language supported, not only the programming has to deal with the order of the language-specific words and syntax correct order to play the words in a credible-way but also in order to acquire human voice-overs a music-studio kind of recording work is needed. For this platform´s IVR language acquisition, the author located two persons: one Spanish-speaking native and another with English-language skills. Thus, support for these two languages was possible but not before spending half a day in a music studio recording the various voice-overs needed by any language set of messages. In fact, an initial 150 voice-overs were recorded, trying to foresee any possible Telephone Network error and information messages needed to be played back to the users.

A professionally-conditioned audio-studio in the city of Barcelona was used for the recording sessions, using a noise-isolated room and special recording equipment along with an audio processing expert to achieve the digitalized IVR voice over that, later on, would be integrated into this dissertation´s work platform.

5.3.5 Multi Origin capable

In order to be able to support a wide range of current and future services the origin of the voice call needs to be abstracted. In a perfect environment the call source should not matter as long as the call has been entered into the dispatching engine of the platform. Thus, the platform allows for multi-origin capable functionalities: in plain words this means that despite of the user´s device that the call was originated from, it should flow in the system smoothly. Having said that, some device-dependent features still exist as some devices allow for some network functionalities that others do not; that is why the multi-origin capable approach was an important one to consider in the development of this platform.

Examples of this requirement can be seen in the applications or services we deployed using the developed platform. For instance, users accessing the system using IPT devices are automatically detected and extended session attributes and services associated. This translates into the fact that one user calling, using for example a Cisco-enabled Video Telephone, shall be able to make video-calls in addition to pure voice calls, whereas a user accessing our system using a normal handset from his/her land line shall not possess such functionality. This multi-origin feature also is independent on the access network, as for example one user making a call from a UMTS/3G mobile handset who might also be enabled or authorized to make a video conference even if the remote end is using a totally IP-All device like a Cisco System video telephone. This means that the platform treats streams in an independent way regardless of the underlying transport being used. It checks for supported services on each layer and it escalates this to the upper network controlling service. In this side of operations, clearly we can assert that by having asked to support multi-origin devices, a total convergence between different access networks and devices is achieved.

This feature inherently supports Billing, Routing and Authentication services to be run on the user´s device regardless of their type and location. A total abstraction versus end-device dependency is therefore accomplished.

5.3.6 IVR Design, features and customizations.

One of the modules highly tight to our platform is the IVR: Interactive Verbose Responsive system. An IVR is an automated software piece that implements voice-over functionality; that is, it guides the user using voice-overs – recorded audio messages – from a human operator. By combining these messages and making questions to the calling user, the system is able to interact with the user in a human-kind of way, by requesting or making questions to the user, asking him/her to select options and implementing any possible service by using this interaction capability along with voice recordings.

The author will not go into further implementation details of the IVR system but, sufficed it to say, that it is one of the modules that took longer to implement as it is used by many of the other modules and services running on the platform. As we deal with telephone calls and phone error and informational messages, this consequently requires us, at all times, to interact with this module to transmit human messages to the users.

For pure reference we shall inform the reader that the IVR was implemented using the Perl language, using an object design approach and storing the user attributes and functionality in XML classes/messages. The Perl module parses these messages and passes them to the control module of the IVR than then it plays the voice-overs to the user using the language specified within the associated XML object. To fulfill this need, an additional XML Schema was also introduced in order to allow for the module to interact with its counterparts. The IVR is used in different services along the line: users using IPT devices get all the information played back to them, users of the Calling Card services are asked to select their preferred languages to interact with them, and users of the PIN-less nomadic mobile service – introduced very shortly in the next section, rely on the IVR to have their access authenticated in case of their CLI – Calling Line Identified – number not already provisioned in the system service.

5.3.7 Collection of Usage Data

This concept comprises a whole lot of data in terms of a user´s session parameters: all the information created during a user presence in the system service. Quite a large number of variables are associated to calls made or received by the platform´s users. Either accounting, billing, end-device and session records are collected and fed into the OSS platform, converted to the internal needed XML messages format so that they can flow among the modules involved in the different processes stages at all times.

For each implemented services XML Schemas exist as they define and support the service-dependent attributes needed to take decision on parameters such as user quota, user balance, time allowed in a call session, language in use, transport end-to-point session capabilities and so on.

5.4 A PIN-less Nomadic Service

5.4.1 AS-IS Situation

It is of common knowledge that Telecommunications Services started to become widely available to the masses in the late 1990s. The Internet revolution first brought us IP connectivity in our universities, research centers, public libraries, and at people-gathering points like airports, train stations, congress halls and other meeting places. The tendency then moved on and it extended to more public places ranging from coffee stores to restaurants, bars, buildings and so on. Finally, it developed even further and it did get its way into our households as the Telephone Companies laid down either fiber optic cables or adapted their old copper with new equipment to bring high-speed connectivity to the neighborhoods; consequently down to our houses.

In less than 20 years we moved from an environment where only high-skilled people used the Internet in such places as research centers, defense organizations or in educational environments, to a situation where most of the developed-countries population has access to the Internet, sometimes with access percentages of up to 90% in some highly-connected countries. This was a revolution we lived and a revolution that shall stay with us for the years to come. Events like this are only lived once in life and we did see it developed and materialized. They were interesting and amazing times and they developed a change in the society of knowledge. For this, the humans need to be grateful.

The Internet uprising has been highly remarkable in front of our eyes for sure. Having said that and what keeps astonishing the author´s mind and soul is the newest even faster and more compelling ongoing revolution: the mobile one. These days the penetration index of people owning a mobile handset is, in some countries, higher than one; meaning that people start carrying, in some cases, even more than one mobile telephone with them. This is not only remarkable but an unstoppable trend as more and more mobile gadgets tend to become part of our daily lives: mobile phones, PDA, *iPhone* (iPod+Mobile phone together), embedded mobile-enabled chipsets in our laptop computers, and more and more gadgets that keep on surprising us with their always more surprising features. Professionals are already big consumers of these devices and the younger generations are following the same trend even at a higher rate as the devices they acquire get mobile functionalities added.

Some people may disagree on the necessity for others to be always reachable and carry mobile phone with them; regardless of the opinion, the fact is that we are now in a mobile decade and the mobile handset is the preferred-device to access both voice and Internet content. Providers and Network Operators have allied and worked together to enable these devices to access any kind of content available on the internet; in addition to continue to support the universal voice telephone service.

Nobody can fight against trends and evolution as when they occur they are unstoppable. The first and second decades of the 21st century shall bring mobile access to every part of the globe. Once that occurs and the OSS systems converge to an IP world, we will have reached the paradigm of having a single network; one running multiple services and content but using the same overall technology and control systems. Once humans reach that moment, the network itself will become a *commodity*, like the water, electricity and gas networks are: vital for the humans' subsistence. In that precise moment a final step on the Internet, or *global network* will have been reached.

Our sons shall ask us how it was before the Internet and how it was before household and world devices interacted to each other regardless of position, language, country or identity. Information shall be available from any place, at any time, using any device impossible to imagine now but surely incredible when looking maybe twenty years ahead. We already belong to a generation that did know the time before and the time after the Internet and that witnessed how mobile technologies landed in our lives.

In the next section we shall adapt our developed platform to support the mobile or *nomadic* users, defined as users that can use any device: mobile phone, home landline or any public telephone to log into our platform to use our telephone services. Every evolution has got a beginning.

5.4.2 Mobile Users using mobile-handsets for roaming and international use

In 2008 Roaming Charges still apply. The occidental world mobile networks are widely available in any country where we travel, live or happen to move from a place one day to quickly move to another one: *Mobility* is the key. Millions of people either for work or just to run their normal lives, travel from one country to another and cross boundaries, borders and geographical limits all the time; the world is interconnected almost everywhere in the globe now but this is mainly driven by commercial interests: the big Telephone Companies, or *Network Providers*, of this decade still believe that business comes first and they still charge inter-border telephone calls charges to their clients. As most of us know, when we carry our mobile phone abroad, an additional per-the-minute and connection fee is charged when we make or received calls. This charge is quite considerable and the Telecommunication Companies everywhere still believe in this revenue-generating custom. It is foreseen that, eventually, this unjustified fee shall be lifted as European and international agencies try to force the big Telecommunication companies to do so. However, it may still take some time and some political fight between companies, governments and entities involved; in the meantime the final user shall keep on being charged for this *Roaming* services.

In a self-evolving and business-driven world, companies continuously work to find new opportunities: the situation outlined in the previous lines also has compelled some companies to find work-arounds or solutions to fix the situation for their mobile customers. In other words: smaller or medium companies invent systems or scenarios valid to avoid charging their users with roaming charges; this in turn, makes them more compelling to their clients.

5.4.3 The PIN-LESS Service

The OSS platform developed in this dissertation work allows for telecommunication services to be easily developed and commercially launched. As requirements and business opportunities arise, these services acquire a reason-to-exist. A few months before this dissertation was even started, a company approached the author of this project to help them implement a service for mobile phones. This service should require authentication per CLI – Caller Line Identifier – and basically allow any user calling from either a mobile or landline telephone, to be able to bypass the local telephone company and log into a second-party telecommunication company operator, therefore granting access to clients to another carrier running services on top of the incumbent operator.

It is an easy to understand service; putting it in plain words: using our mobile handset we call a local number; then we automatically obtain a second dial tone and can proceed to dial the desired number; it is the same functionality that the Calling Card service supported by our platform with the remarkable difference that no PIN – Personal Identification Number – is needed to authenticate the user as the own mobile or landline phone number acts as the identification token as each PSTN number is unique to the assigned user; thus providing the capability for any user carrying or using a telephone to bypass his/her Network Operator.

The PIN-less, service known by this name as there is no need to have a PIN code, consequently the word ‘less’ applies, is consequently a service allowing any one travelling or living abroad for a period of time to call back home or call anywhere in the world just using his/her mobile or landline phone as the identification token.

Having sketched the service definition, the author proceeded to implement a new service object with this requirement in mind. It was not long before a new service was launched supporting the authentication of users per-CLI, and hence the PIN-less service became a reality.

5.4.4 Fixed landline-Users Roaming and Moving locations, Test Users

In the initial beta phase of the service, nearly 30 users were provisioned in the system. They immediately were able to use their mobile handsets to call abroad without any need to enter any pin codes from their telephones. As the roaming charges did not apply and the international calls were more affordable as they use VoIP cheaper routes instead of very expensive ones, the service has quickly evolved and turned into a reality. The customer ordering the service from the author has placed an order for one thousand student mobile telephone to be added in the system, plus later on about three thousand Spanish telephone lines to be provisioned, too.

5.4.5 Service End-Devices

Any telephone is supported by the service by interfacing with the user using our IVR module. Having said this, the author is now developing a interface for UMTS/3G handsets that will give them the functionality of interacting with the platform using a mobile web application, therefore making it usable from any mobile telephone out there. Specific support for Apple´s *iPhone* handset is foreseen.

5.4.6 Inherited features from the System´s Supported capabilities.

Natural inherited entities are the platform´s Billing, Authentication and Routing modules from the core supporting this service. There was no need to design them or implement them again as they already belonged to our OSS platform core. This proves us that the design can really implement new services and commercially launch them in a prompt and competitive way. The entire infrastructure commented and walked through in previous chapters is used to support this service: from the BARA modules, to the TDM circuit provisioning, routers configuration and supporting hardware servers, everything developed for this dissertation work is recycled to sustain this new service.

The initial service users are international students living and studying in Barcelona, Spain. These users are given a national mobile provider SIM-cards for their mobile telephones and when their telephones are handed to them they are advised to store their contacts using the service short-dialed, or *prefix*, so that their phones dial this code before dialing the desired number to call. As this is automatic and any Nokia and, most of the generic mobile phones available in the market have got this functionality, the service can be used by all the students to make their international calls without having to hit very expensive telephone costs.

The OSS platform automatically takes care of authenticating these users, grant them service to the calling features and bill them either post-paid or pre-paid as specified by the service owner company. The entire service is provisioned, monitored, control and driven by our platform. No third-party software is required as the entire functions are supported by us.

5.4.7 Requested Functionality associated to this new service

A CRM – Customer Relationship Management – system has been asked by the client ordering this service. To fulfill this request, in the next months the OSS platform shall bring in a new module consisting of a Web-accessible interface to manage the provisioning of users, associated rates, prices, user limits and so on. Basically, a tool running on top of our platform to allow the end client to control his service itself instead of accessing the system directly as we have been doing for the other services running on it. Once this CRM is launched, we shall be in the position to resell custom-made services to all kind of companies requiring telecommunication services for their mobile workforce and nomadic users.

5.4.8 Commercial Viability

Every telecommunication service generates revenue; every single one as in the author´s opinion the Telecommunication market is one of the most rapidly and business-creating opportunities existing nowadays. Additional circuits have been ordered to keep up with the telephone channels that this service is requiring. The viability of the service is clear and the author hopes to have the second phase of the service in production in about one month time as this is a new service.

Once this point is reached, more than three thousand users shall become daily users of the platform. These numbers only for the PIN-less service, as the other services introduced throughout this dissertation have been commercially working for a while already and they also have their own user base.

6 Conclusions and Future Work

Every project begins with an idea and ideas come to life triggered by human, social or business needs. Some of us believe that the ideas that thrive in this world are the ones driven by a strong belief and motivation. The author of this dissertation was always passionate and thrilled by the world of Telecommunications, since his childhood, and he did always had the dream of, one day, being able to become part of it by playing some kind of role within this exciting engineering field. By combining the author´s interest in the fields of Computer Science and Telecommunications, decades later this goal has been achieved. It has taken a considerable amount of time and work to execute this target, but the author is grateful to the evolution of technology that has made it possible. In this dissertation an approach to building a Telecommunications Company from scratch has been introduced and put into practice, proving that even the more remote objectives in our lives can be realized if human beings really want to accomplish them.

The very heart of the aim of this dissertation was to build up a system able to act as a small Telephone Company processing and providing the basic telephone service in addition to the Telecommunication Services demanded in the decade we live in. Prior to starting the materialization of this dream nothing existed, and months later, after this dissertation has been concluded, a system processing daily thousands of telephone calls exists and keeps on growing adding new services on top of it. The good thing about achieved goals and about dreams in particular is that when they become real the person who is lucky enough to see them happening is filled up with joy and satisfaction. This is the case of the author of this dissertation work and he hopes his contribution can be transmitted and serve others to learn and also make their dreams true.

6.1 Goals Achieved: Level of Original Targets Achieved

The initial requirements introduced in the first chapter of this project were clear: to design and implement a system capable of supporting the infrastructure of a Telecommunications Company providing telephone calls and additional services to its users.

Along this dissertation we have designed the several modules required to make these requirements become real. By first making a proper design and, later on, implementing exactly what we originally had in mind we believe our initial aims has been accomplished completely. Our aim was a humble one and there was no need to highlight or make a remarkable system, we just wanted to be able to implement a real system, first an existing and working one, and later on, in the second phase of development to apply our knowledge and learnt lessons during the Computer Science Engineering Career and to use these to convert these requirements into a real system.

Nowadays, a system fulfilling our initial specifications exists; it does work in a 24h shift serving and allowing people to talk to every part of the known earth, processing their calls as any other Telephone Company does. This is more than an accomplishment of what we outlined in the beginning of this dissertation work. Technically speaking we did want to implement an OSS – Operating Systems and Support – platform. That would be the engineering name for the function we have implemented. However, what was hidden underneath is that the real goal was to prove ourselves that a platform supporting the same services than a Telephone Company does can be achieved using a humble amount of resources, software – nothing else that people´s time and ideas -, and hours working on the project.

This has been our first attempt to achieve these goals. We have realized them and the author´s idea is to work forward and continue expanding this work by implementing new services as they come by. The author is so motivated that he is now trying to develop a real company around this platform; a company with real people, a company with real and different roles, a company interacting with other established international Telecommunication Companies and slowly but steady the work done in this dissertation has not only being implemented in a real entity but it is also contributing to the development of, hopefully, a new player and known company within the field of Telecommunications. The company incorporated to support this platform´s commercial live is now interacting and having international clients; it is just the beginning in the author´s opinion but the platform is starting to serve the needs of companies in the United States, France, Ukraine, Russia, Bolivia, Ecuador, China, and more and more clients are added to the portfolio of interested companies wanting to use the service that this dissertation´s work has made possible. By implementing the needs we had in the beginning and doing it in a stable and scalable way, we are now in the condition of acting as a living company with its own dreams and ambitions. This is possible as we work hard to really implement what we needed to fulfill this target.

6.2 Conclusions

One thing brings to another. The dissertation goal was to come up with the software, hardware and right binding concepts to build up a small Telecommunications Company. The Voice Over IP technology allowed us to see this goal achieved faster as it made it more achievable in terms of investment and tools development. As time goes by, though, human individuals learn, mature and get to the point of being able to group concepts, systems and criteria in higher layers of abstraction, therefore what started with a pure engineering approach and goal of creating a whole platform supporting a system able to carry voice from one place to another, and properly managing this service and allow for further Telecommunications to run as well, has evolved also in parallel in another non-engineering scenario: an enterprise or business company. The author started this as a mere idea but now he has come across that in order to continue developing this, he has to make it grow and not only in the technical side anymore but in the business-side of it. Each platform servers a purpose and this developed platform serve the people´s needs to talk on the phone, see each other and in general to communicate with each other. But having said that, the platform is inherently supporting a business itself as the services and core design was meant from day one to bill for the services, therefore making it a company since its very first day of operation.

From the previous paragraph, a dilemma arises as the platform is now acting and supporting a business. The author of this dissertation studied to, one day, become an engineer. His main motivation was to become one but, by chasing this, he has now had to learn the rules and operation of a real company. The reason for this is that the magnitude of the services and associated human work required to run a Telecommunications Company requires resources in terms of people working for the company, selling the products – the services supported by this dissertation platform -, paying the suppliers – the other telecommunication companies providing services to our own company -, coordinating with the financial people to validate inbound and outbound transactions and so on. In other words: the author has now ventured himself, as he is challenged and thrilled, into the next goal of maintaining his dream and making it grow day by day. This means having to build and extend a company in order to see the platform fulfilling the needs for which it was developed and for which this dissertation was written. Because of this, the dissertation´s author has immersed himself in the world of Telecommunications by incorporating two different companies: one is called *IP Square Communications* and it deals with calling cards, ITSP Services and services aimed to small and mid-sized companies to cover their telephony needs whereas a new one, recently incorporated company, called *Bluesense Communications* deals with voice wholesale traffic acting as a clearing-house and as a broker trading international voice-minutes.

Consequently, now this dissertation work is seeing his conceived ideas made real by means of running associated to an incorporated Telecommunication Company itself, therefore the dream was achieved: not only we wanted to be able to support such goal but, at the very end, we had to legally and commercially create one to really seeing it happening. Ideas, sometimes, become real and we are grateful that is has been like that in this case. Only time shall tell how this dissertation work evolves and survives in the very competitive world of the big Telecommunication Companies.

6.3 Future Work

This is a complex and large development; it includes software and hardware systems and that makes it difficult to maintain by its inherent nature. That is why future work should and shall consist of documenting every single platform module, and hardware system in place. This is needed in order not to lose focus on the platform development but also to one day achieve the extra goal of being able to convert the platform in a product itself to resell it to other companies interested in the services this platform provides.

A platform roadmap should also include video and rich-content integration by means of supporting the new devices and services that shall land in our lives in the next years to come. The platform is ready to cope with it and its modular and extendable design makes it adaptable to fulfill these needs. The platform´s author has always being very open in what refers to the platform future development. Related to this some of the next functionality or new *nice-to-have* features are listed below:

- An all-IP world: Support for the next services on the roadmap: video, video and video: design an appropriate scalable service to manage handling and billing of video streams.

- Implement a Web Portal Interface to ease the administration and daily basis of the platform configuration, monitoring and services. Integrate the current platform into a distributed service-based one like the already design allows to do.

- Implementation of Video Gateways to B3G Mobile Networks: allow the platform to be able to convert video streams from IPT Devices to/from Mobile handsets using UMTS/3G Video codecs.

- Business Model Requirements: Support for Integration with Third-Party Systems: implement functionality bringing up the capability of talking to external CRM and ERP systems, such as Siebel, SAP, Peoplesoft, and so on.

- Work on an automatic procedure to exchange billing data with resellers, carriers and traffic aggregators to encourage them to create business around the actual voice platform. Try to convince other players to use and be bound by our platform.

- Need for a CRM – Customer Relationship Management – frontend to increase the accessibility and usability of the platform. Besides, such a tool is needed to bring into the system a commercially trained team of customer support representatives.

The listed features are not only a summary of the capabilities that the author would like to see supported and implemented in this dissertation platform but, more important, a statement to witness the further development of the platform and a materialization of the roadmap associated to it. In other words: the platform shall evolve and work does not finish in this dissertation but it shall continue and hopefully evolve to the point of one day supporting and coping with very large Telecommunications needs.

Time will tell.

7 Bibliography

Bartee, Thomas C., *ISDN, DECnet and SNA Communications*, Howard W. Sams & Company, U.S.A, 1989

Gruber, Martin, *SQL: Instant Reference*, Sybex, U.S.A, 1993

Date, C.J, *An Introduction to Database Systems*, Addison-Wesley Plubishing Company, U.S.A, August 1995

Gamma, E., Helm, R., Johnson R.,Vlissides J.,*Design Patterns: elements of Reusable Object-Oriented Software*, Addison-Wesley P lishing Company, U.S.A, October 1997

Martin, James and Lebel Joe, *TCP/IP Networking, Architecture, Administration and Programming*, Prentice-Hall ,U.S.A. 1994

Oppenheim, Alan V., Willsky, Alan S., Young, Ian T., *Signals and Systems,* Prentice-Hall, U.S.A, 1983

Orfali R, and Harkey D, *Client/Server Programming with JAVA and CORBA*, John Wiley & Sons Inc, U.S.A, 1998

Orfali R, Harkey D and Edwards, J, *Instant CORBA*, John Wiley & Sons Inc, U.S.A, 1997

Pressman, Roger S, *Ingeniería del Software: Un Enfoque Práctico*, McGraw-Hill, U.S.A, 1993

Serra, Ignasi and Vilanova Ramon, *Tractament del Senyal*, Manuals de la Universitat Autònoma de Barcelona, Bellaterra, 1999

Stevens, W. Richard, *Advanced Programming in the UNIX Environment,* Addison-Wesley P lishing Company, U.S.A, March 2000

Stevens, W. Richard, *TCP/IP Illustrated Volume 1: the Protocols*, Addison-Wesley P lishing Company, U.S.A, February 2000

Wendell Odom, *CCNA Intro and CCNA ICND: Exam Certification Guide*, Cisco Press, U.S.A, 2004.

8 Glossary of terms and abbreviations

The following terms and nomenclature are used throughout this document. We list those below as a reference to the document reader.

Mnemonic	Description
ACD	Automated Call Distribution
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
BRI	S0-ISDN Interface PSTN
CAS	Centralized Attendant Services
CCITT	Comite Consultatif International de Telegraphique et Telephonique (International Telegraph and Telephone Consultative Committee), now ITU-T
CORBA	Common Object Request Broker
CRM	Customer Relationship Management
CS	Commercial Specialization
CTI	Computer Telephony Integration
DID	Direct Inward Dialing
DGR	Data Gathering Report
ETSE	Escola Tècnica Superior d'Enginyeria
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IDE	Integrated Development Environment
IP	Internet Protocol
IPT	Internet Protocol Telephony
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
IVR	Interactive Verbal Response
ITU	International Telecommunications Union
ITSP	Internet Telephony Service Provider
IVR	Interaction Voice Response
LAN	Local Area Network
LCM	Life Cycle Management
MPLS	Multi Protocol Label Switching
OSI	Open Standards Institute
PBX	Private Branch Exchange
PDA	Personal Digital Assistance
PDF	Portable Data Format
PERL	Practical Extraction and Report Language
PIN	Personal Identification Number
PRI	Primary Rate Interface S2M-ISDN Interface PSTN
PSTN	Public Switched Telephone Network

QOS	Quality Of Service
SDH	Synchronous Digital Hierarchy
SLA	Service Level Agreements
SSL	Secure Sockets Layer
SOAP	Simple Object Access Protocol
SONET	Synchronous Optical NETwork
SQL	Structured Query Language
TDM	Time Division Multiplexing
TLS	Transport Layer Secuity
UAB	Universitat Autònoma de Barcelona
UML	Unified Modelling Language
VOIP	Voice Over IP
WAN	Wide Area Network
Workbook	Workbook means a detailed description about the Solution including the IPT Design for Office and Call Centre Design, the Reporting and the required Adjuncts
XSLT	eXtensible StyleSheet Transformations
XHTML	eXtensible Hypertext Markup Language
XML	Extensible Markup Language
WAP	Wireless Application Protocol
WML	Wireless Markup Language

9 Table of figures

SIGNATURA DE L'AUTOR/FIRMA DEL AUTOR/AUTHOR'S SIGNATURE

Esteve Vilardell Cànovas

Signat/Firmado/Signed: ....
Barcelona, 11/06/2008

RESUM

En aquesta memòria l'autor, fent servir un enfoc modern, redissenya i implementa la plataforma que una empresa de Telecomunicacions del segle 21 necessita per poder donar serveis de telefonia i comunicacions als seus usuaris i clients.

Al llarg d'aquesta exposició es condueix al lector des d'una fase inicial de disseny fins a la implementació i posada en producció del sistema final desenvolupat, centrant-nos en solucionar les necessitats actuals que això implica. Aquesta memòria cubreix el software, hardware i els processos de negoci associats al repte de fer realitat aquest objectiu, i presenta al lector les múltiples tecnologies emprades per aconseguir-ho, fent emfàsi en la convergència actual de xarxes cap al concepte de xarxes IP i basant-se en aquesta tendència i utilitzant aquesta tecnologia de Veu Sobre IP per donar forma a la plataforma que finalment, de forma pràctica, es posa en producció.

RESUMEN

En esta memoria el autor se vale de un enfoque moderno para rediseñar e implementar la plataforma que una empresa de Telecomunicaciones del siglo 21 necesita para ser capaz de dar servicios de telefonía y comunicaciones a sus usuarios y clientes.

A lo largo de esta exposición se conduce al lector desde el diseño inicial hasta la implantación y puesta en producción final del sistema desarrollado, centrándonos en solucionar las necesidades actuales que esto conlleva. Esta memoria cubre el software, hardware y los procesos de negocio asociados al reto de hacer realidad este objetivo y presenta al lector las múltiples tecnologías utilizadas para hacerla realidad, haciendo énfasis en la convergencia actual de redes hacia el concepto de redes IP y basándose en éste para utilizar la tecnología de Voz sobre IP, para dar forma a la plataforma que finalmente se pone en producción de forma práctica.

ABSTRACT

In this dissertation the author takes a modern approach to redesign and implement the platform needed by a 21st century Telecommunications Company to provide both basic and advanced telecommunication services to its users and clients.

This essay walks the reader from the initial design phase to the final implementation and deployment of the developed system by focusing on real-world practical needs. This dissertation covers the software, hardware and business processes encompassing the challenges associated to this task and it introduces the reader to the multiple technologies used to develop it, emphasizing the present convergence of networks towards an All-IP world and strongly relying on Voice Over IP technology to shape the platform and bring it to live.