TECHNISCHE UNIVERSITÄT BERLIN

School IV - Electrical Engineering and Computer Science
Department of Telecommunication Systems
Chair for Next Generation Networks (AV)

Design and Implementation Aspects of Open Source Next Generation Networks (NGN) Test-bed Software Toolkits

- Engineering Doctorate Dissertation -

Author: Dipl.-Eng. Dragos Vingarzan

Scientifically Defended in Berlin, on the 27th of November, 2013
D 83

Fraunhofer
FOKUS

R&D in cooperation with Fraunhofer Institute for Open Communication Systems

TECHNISCHE UNIVERSITÄT BERLIN

Design and Implementation Aspects of Open Source Next Generation Networks (NGN) Test-bed Software Toolkits

vorgelegt von
Dipl.-Ing.
Dragoș Vingărzan
aus Cugir/Rumänien

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender :	Prof. Dr. Axel Küpper
Gutachter :	Prof. Dr. Thomas Magedanz
Gutachter :	Prof. Dr. Klaus David
Gutachter :	Prof. Dr. Nöel Crespi
Gutachter :	Prof. Dr. Paolo Bellavista

Tag der wissenschaftlichen Aussprache: 27.11.2013

Berlin, 2013
D 83

This work represents my engineering doctorate dissertation, defended on the 27th of November 2013 in Berlin.
The academic requirements have been fulfilled at the Technische Universität Berlin, within School IV - Elektrotechnik und Informatik - Chair for Next Generation Networks.
The practical R&D has been performed in cooperation with the Fraunhofer Institute for Open Communication Systems (FOKUS). The resulting Open Source IMS Core project is a community project, for which the source code copyrights belong in large parts to Fraunhofer FOKUS.

www.tu-berlin.de
www.av.tu-berlin.de
vingarzan@gmail.com

www.fokus.fraunhofer.de/go/ngni
www.openimscore.org
dragos.vingarzan@fokus.fraunhofer.de

Impressum

Druck und Verlag: epubli GmbH, Berlin, www.epubli.de
ISBN 978-3-8442-7788-3
Zugl.: Berlin, Technische Universität, Diss., 2013

Abstract

Information and Communication Technologies provide for a long time already the backbone of telecommunication networks, such that communication services represent an elementary foundation of today's globally connected economy. The telecommunication landscape has experienced dramatic transformations through the convergence of the Telecommunications and the Internet worlds. The previously closed telecommunication domain is currently transforming itself through the so-called Next Generation Network (NGN) evolution into a highly dynamic multi-service infrastructure, supporting rich multimedia applications, as well as providing comprehensive support for various access technologies.

The control layer of such NGNs is then of paramount importance, as representing the convergent mediator between access and services. The use and the optimization of the IP-Multimedia Subsystem (IMS) was researched and considered in this domain for many years now, such that today it represents the world-wide recognized control platform for fixed and mobile NGNs.

Research on protocols and services for such NGN architectures, due to the convergence of technologies, applications and business models, as well as for enabling highly dynamic and short innovation cycles, is highly complex and requires early access to vendor independent - yet close to real life systems - validation environments, the so-called open technology test-beds.

The present thesis describes the extensive research of the author over the last nine years in the field of open NGN test-beds. It focuses on the design, development and deployment of the Open Source IMS Core project, which represents since years the foundation of numerous NGN test-beds and countless NGN Research & Development projects in the academia as well as the industry domain around the globe. A major emphasis is given for ensuring flexibility, performance, reference functionality and inter-operability, as well as satisfying elementary design principles of such test-bed toolkits.

The study also describes and evaluates the use of Open Source principles, highlighting the advantages of using such an approach in regard to the creation, impact and sustainability of a global OpenIMSCore research community.

Moreover, the work documents that the essential design principles and methodology employed can be reused in a generic way to create test-bed toolkits in other technology domains. This is shown by introducing the Open Evolved Packet Core (OpenEPC) project, which provides for seamless integration of different mobile broadband technologies.

Zusammenfassung

Informations- und Kommunikationstechnologien bilden seit langem das immer wichtiger werdende Rückgrat der weltweiten Wirtschaft und Telekommunikation, in der speziell Telekommunikationsnetze und -dienste einen elementaren Anteil tragen. Durch die Konvergenz von Telekommunikations- und Internettechnologien hat sich die Telekommunikationslandschaft in der letzten Dekade drastisch verändert. Bislang geschlossene Telekommunikationsumgebungen haben sich im Wandel zum sogenannten Next Generation Network (NGN) hinsichtlich unterstützter Zugangsnetztechnologien und angebotener multimedialer Anwendungen sowie der eingesetzten Protokolle und Dienste zu komplexen, hochdynamischen, Multi-Service Infrastrukturen gewandelt.

Die Kontrollschicht solcher NGNs ist dabei von übergeordneter Bedeutung, da diese zwischen den Zugangsnetzen und den Anwendungen sitzt. Der Einsatz und die Optimierung des IP-Multimedia Subsystem (IMS) wurde in diesem Kontext Jahrelang erforscht und diskutiert und es repräsentiert heute die weltweit anerkannte Kontrollplattform für feste und mobile Telekommunikationsnetze.

Die Forschung an Protokollen und Diensten in diesen NGN Umgebungen ist aufgrund der Konvergenz von Technologien, Anwendungen und Business Modellen sowie der hohen Dynamik aber kurzen Innovationszyklen hochkomplex. Der frühzeitigen Zugang zu herstellerunabhängigen – aber dicht an der Produktwelt angelehnten - Validierungsinfrastrukturen, sogenannten offenen Technologietest-beds, kurz Test-beds, ist daher für Forschungs- und Entwicklungsabteilungen unerlässlich

Die vorliegende Dissertation beschreibt die umfangreiche Forschungsarbeit des Autors auf dem Gebiet der offenen NGN Test-beds über die letzten neun Jahre und konzentriert sich dabei auf Entwurf, Entwicklung und Bereitstellung des Open Source IMS Core Projekt, das seit Jahren die Grundlage für eine Vielzahl von NGN Test-beds und zahllose NGN Forschungs- und Entwicklungsprojekte im akademischen als auch Industrienahen Umfeld rund um den Globus darstellt. Dabei wird ein großer Schwerpunkt auf die Anforderungen hinsichtlich Flexibilität, Leistung, Funktionalitätsumfang und Interoperabilität, sowie elementare Designprinzipien von Test-bedwerkzeugen gelegt.

Die Arbeit beschreibt und bewertet darüberhinaus den Einsatz von Open Source Prinzipien und veranschaulicht die Vorteile dieses Ansatzes hinsichtlich Einfluss und Nachhaltigkeit der Forschung anhand des Aufbaus einer globalen Open Source IMS Core (OpenIMSCore) Forschungs-Community.

Außerdem veranschaulicht die Arbeit zum Ende die Wiederverwendbarkeit der wesentlichen angewendeten Designprinzipien an anderen maßgeblich durch den Autor entwickelten Test-bed Werkzeugen, insbesondere dem Open Evolved Packet Core (OpenEPC) für die nahtlose Integration verschiedener Breitbandnetztechnologien.

Acknowledgments

"We could change the world if the gods would give us the source code!"
– Random Internet guy

Here is what an agnostic computer and software engineer specialized in telecommunications came up with.

This thesis was written between 2008 and 2013, based on work at the Fraunhofer Institute for Open Communication Systems (FOKUS) in Berlin, starting back in 2004.

First of all I would like to thank Professor Magedanz for his supervision and especially for the big-thinking. What a crazily exciting and successful ride the Open IMS/EPC Playgrounds have been! And this is just the end of the beginning.

Second I must thank Peter Weik as bringing to life such major projects is a team sport. Thank you for teaching the Swabian rigor and calm to a hot-blooded programmer. Also his help spotting errors and making this dissertation more readable are highly appreciated.

Also credits must be assigned to Dorgham Sisalem for teaching me how to wrangle open source software.

Then of course many, many thanks to my team, colleagues and also students that I supervised, without whom spinning these projects to world-wide scale and recognition would have not been possible. In alphabetical order: Alberto Diez Albaladejo, Shengyao Chen, Marius-Iulian Corici, Bogdan-Vasile Harjoc, Alexandru Ilie, Ancuta-Andreea Onofrei, Bogdan Pintea, Florin Popescu, Adrian-Daniel Popescu and many more.

Finally I would like to thank my children, wife, parents and in-laws for giving me the time to write this thesis.

Berlin, January 21, 2014
Dragoș Vingărzan

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

At the beginning of this century the telecommunication industry stands at technological cross-roads. The current communication networks are now available virtually everywhere and almost to everyone in the world, which means that they are more and more regarded and accepted as part of the basic human needs [18]. The services offered over these networks however have remained almost the same for decades[1], although in parallel the Internet world has taken huge leaps in terms of introducing new communication concepts and revolutionary services. To remain relevant in the telecommunication business, the current Telecommunication Service Providers (TSPs) will have to constantly evolve their communication networks in order to keep the pace and even improve on the services provided by the Internet, or risk being reduced to simple bit-movers for the transport of abstract Internet data. Today's architectural evolution targeting these changes is referred as Next Generation Networks (NGNs).

Towards fulfilling the NGN [19] requirements, technology sets like IP-Multimedia Subsystem (IMS) [20] or Evolved Packet Core (EPC) [21] represent (at the moment of the writing of this thesis) evolutionary steps for the traditional TSPs towards the Future Internet [22] concepts. These new platforms aim to optimize the core networks from a cost efficiency perspective, while also striving to offer a rich and future-proof environment for fostering the Internet innovation of services.

The telecommunication domain was traditionally a rather closed one. At the beginning of this work the pre-NGN telecommunication networks relied heavily on closed architectures and a simplistic value chain. Although today's new concepts

[1] Most prominently telephony

like Software-Defined Networking (SDN) are indicating that the evolutionary path towards open and software-driven architectures is the correct one, back when IMS was introduced the possibility of enabling complex value chains inspired from the Internet world was a very daunting question for the leading engineers of the domain, requiring very risky decisions to be taken largely based on theoretical evaluations.

The Internet world has experienced a tremendous increase in value and popularity over the last decades. Its capacity of sparkling innovation through its open model is remarkable. From the services perspective, the Internet has proved to be at the forefront of innovation, with current concepts like Web 2.0 flourishing and enabling interesting business models for servicing the communication needs of the new generation.

From a technical perspective, the Internet protocols continue to have such a successful evolution and have matured so well in a relatively short time, that they have managed to overtake in several performance indicators the traditional telecommunication counterparts. Recognizing this, the telecommunication industry has started a shift in its approach from the old ways of using highly specialized, strict protocols and closed interfaces towards reusing the generalized and more flexible IP protocols, while embracing open interfacing towards applications and services, but also internally between Core Network (CN) functional components. And perhaps even more visible today is the opening of said interfaces to the public, as well as the creation and fostering of developer communities, which provides a significant new level of openness and opportunity.

The NGN convergence represents then a major transition and evolution of the Telecommunication world. And this will not only be limited to the technological aspects, but additional major changes in business models, markets and also standardization process have been started and will take place in the near future.

Considering such a broad change at all these critical and major levels, it can be inferred that an entire industry is reinventing itself. From this, the sheer size complexity emerges, together with the need for extensible technology test-bed enablers. Such tool-kits allowing to prototype fast and efficiently new ideas beyond simulation and emulation, would satisfy the hunger for high dynamism of the NGN environment.

Such an evolutionary change represents a major undertaking then and will require significant efforts in research, design, standardization, implementation, deployment and so on. Not to underestimate are also the major differences between different domains at Access and Application layers, which should be mediated by the middle Control layer provided by IMS. Fortunately, with the first architectural design steps completed, the IMS architecture as a solid CN instantiation for NGNs control emerges and this will be targeted as a common control platform and reference model [11].

With a defined architecture a next step, towards mastering the complexity, is to perform early trials of the technology, which will then also invoke evaluations of the service concepts and business feasibility and as such span on multiple levels and domains. While many such tasks can be realized through theoretical models and

evaluations, test-beds with prototypes and toolkits would be essential for discovering early issues as well as providing solutions and building a healthy feedback cycle, critical for the overall architectural success.

These prototypes must also be close enough to real-life implementations, such that realistic behavior can be analyzed. In this direction simple emulations of functionality are certainly not satisfactory, but the typical performance-seeking patterns of the Telecommunication Equipment Manufacturers (TEMs) must be followed. The author, as a Computer Engineering graduate with a major in low-level software development and networking, believed that a `C` implementation would be the proper choice, even as the cost and complexity would be much higher than an alternative Rapid Application Development (RAD) approach for example on a `Java` basis. Real-life signalling routing and processing equipments are highly tuned network elements enabling flawless inner-working of the operator's CN, hence only similar implementations and approaches would yield similar results.

Following on the topic of complexity associated with the introduction of NGN architectures, **the first traction point** of the thesis will be represented by **identifying the key design principles and then analysing the main implementation challenges** for realising such test-bed toolkits.

Additionally, as the target is research, the implementation should provide a solid degree of openness towards its users. Experimenters are often in the need of performing complex "what-if" trials, which require not only an external re-configuration of a test-bed, but also deep changes in the fabric of the experimental prototypes. Hence direct control of the implementation through access to source code is not only beneficial, but key for enabling a reference toolkit. Open Source principles then would provide a boost to experimenters by first allowing deep re-purposing as required by bold approaches, but also encouraging researchers to exchange tools and ideas.

Establishing a singular test-bed is of course valuable, yet seriously limiting the beneficiaries to a lucky few. Rather publishing an Open Source project as a toolkit for test-bed will allow a much broader research base to establish numerous such laboratories and to experiment with NGN in many more ways than possible in a centralized way. Here Open Source would provide the second boost, by exploding the experimenters base to an industry-wide reach.

During the NGN pioneering years, when the migration towards all-IP concepts started, an acute need for practical experimentation emerged. However, such a task for the very complex Core Networks (CNs) is certainly far from trivial. The author was one of the first such engineers to start prototyping the new concepts and to trial their feasibility in realistic environments by starting to considering the power of Open Source as a catalyst for compensating the costs of establishing such toolkits. As in today's world the result of the work exposed here are available and largely employed world-wide by NGN experimenters, the use of Open Source for driving innovation in the telecommunication is a feasible option, one can directly say that these targets have been achieved and the evolution from simple value chains with closed tooling to a complex one with open architectures is reality and also a base

on which even more advanced concepts are starting to take shape.

Not to forget are also the benefits of the initial functionality set provided, which would give researchers a head start and give them more time and resources to concentrate rather on their immediate and particular needs. Instead of replicating investments into the same core functionality elements, each experimenter can reuse and improve on a critical mass of functionality, common between various experiments, again as an added advantage provided by an Open Source model.

At the beginning of this work, when IMS started to be defined, there were no such toolkits available for NGNs. Several initiatives were started, yet none was ambitious enough to include in scope a significant part of the architecture, neither was any going directly for the core scopes and targets of the Telecom domain.

What proved to be viable though were projects in the Voice over IP (VoIP) space. There a mini-revolution of telephony-replacing services was sparked by many successful and highly innovating new start-up companies. While such VoIP services were initially in high demand from the Over-The-Top (OTT) and Internet Service Provider (ISP) actors in the Internet domain, soon after these started to be used also in the Telecom domain, at first as effective means of reducing costs in less sensitive sub-systems. Such projects were in most cases taking advantage of the openness in the Internet world, by using publicly reviewed and improved technologies, as well as the Open Source models for sharing development and testing costs.

As these first VoIP prototypes showed that Telecom traditional services can be realized with such open source Internet world tools, **the second traction point** of this thesis is represented by **reusing and improving upon Internet Open Source projects for the purpose of providing reference NGN toolkits in the test-bed environment**.

Feasibility of such an undertaking is to be achieved, in spite of the major functionality scope extension to an entire NGN CN architecture as IMS, by introducing limitations on other Key Performance Indicators (KPIs) to reduced test and trial environments values and exclusion of real-life deployment and exploitation requirements.

Considering targeted audience, traditionally the academia had very limited access and impact on the proprietary architecture of the Telecommunication domain. The high costs required for realizing suitable hardware and software systems was always prohibitive for any other entity outside the major operators. Neither was the design or standardization process open for academia participation or any other external review, with many essential parts of the technologies being regarded as well guarded secrets. The net result then is that the academic curriculum on these architectures was dry and could not provide enough insight or contribution paths for the educational domain towards the telecommunication world.

Independent and open toolkits have though the potential of filling this gap. Low cost prototypes would provide to young students the opportunity of trialing the technologies themselves, while they are still in their educational phase, providing better preparation for their future as industry contributors. As a next step, arming the students with the additional enabler of modifying and trying their own ideas,

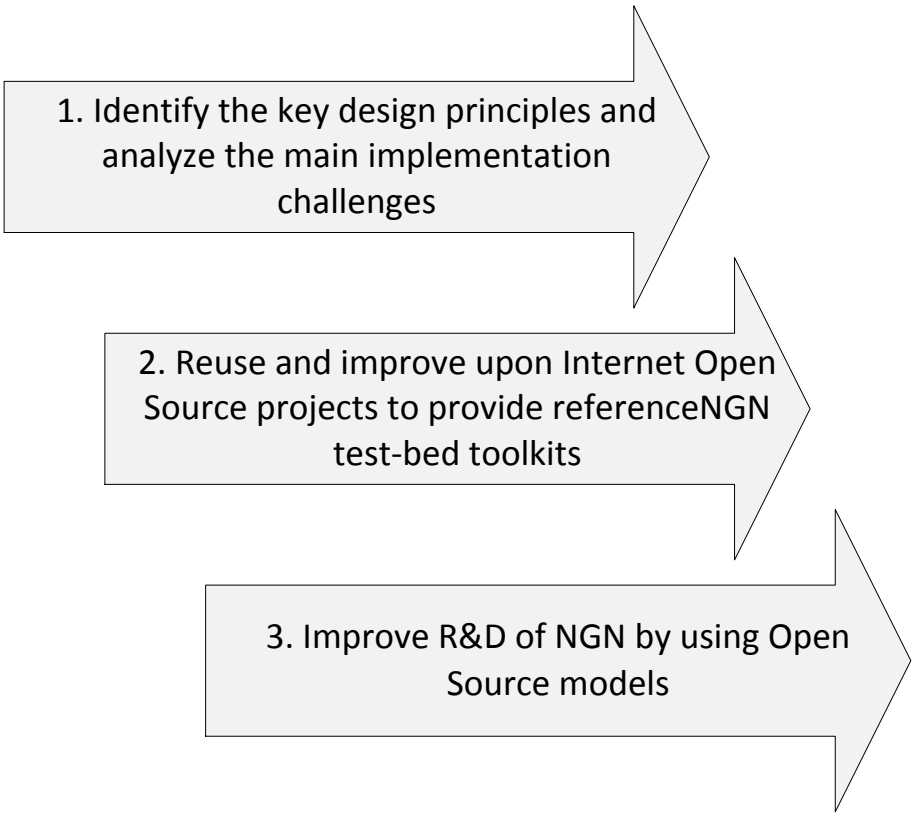


Figure 1.1: Main Directions of the Dissertation

would represent a significant boost to innovation. Hence **the third traction point** will seek **if an Open Source model would bring benefits for the Research and Development (R&D) of NGN**.

As indirect targets, the work will allow for answering a few secondary questions, yet these will not be key driving points, rather additional results. For example, to validate if the principal targets were achieved, a simple methodology to evaluate the usefulness of test-bed toolkits is required. Then a question of relevance in regard to real life systems would have to be answered, which would indicate if such prototypes, as the ones realized in this work, can be considered representative in their behavior and performance in respect to future deployments. And last but not least, the openness of the project would have the potential of establishing an innovation bridge between a previously closed telecommunication domain and the academic world.

1.2 Scientific Challenge, Target & Major Keyword Definition

The **principal scientific challenge** for the evolution of telecommunication networks towards NGN concept is represented by the combination and evolution of the telecommunication's domain traditional and highly valuable characteristics (security, performance, reliability, scalability) with the beneficial traits of the Internet domain (flexibility, simplicity, openness, nimbleness and lower costs). Further, this evolution must be made into a singular and ubiquitous environment, which should be both legacy integrated and open towards future service concepts.

Target: the present thesis will present research in the field of NGNs, limited to Test-bed environments, which leverages the power of Open Source to overcome shortcomings in innovation pace in the Telecommunication Industry. The work will center around the establishment of a powerful Open Source NGN toolkit for R&D, as catalyst for new ideas, as validation tool and as bridge between the academia and the industry.

Although to the astute NGN specialist the thesis should not present difficulties in understanding, the domain is as such riddled with acronyms (see the List of Acronyms section), concepts and models that even this introduction would pose difficulties to the non-specialist reader. Before introducing extensively the domain and used concepts in the following chapter, a series of general terms and keywords will be briefly introduced here as reading keys.

The term **Next Generation Network (NGN)** is used today to indicate a type of communication network used in the telecommunication industry to provide services to subscribers by leveraging the communication protocols and paradigms from the Internet world. One of the best definition is provided by the International Telecommunication Union (ITU) - Telecommunication Standardization Sector (ITU-T) [19]:

> *"A Next Generation Network (NGN) is a packet-based network able to provide services including Telecommunication Services and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent from underlying transport-related technologies. It offers unrestricted access by users to different service providers. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users."*

IMS was at the beginning of this work one potential architectural variant for realizing the NGN targets. Its main advantage over other theoretical options is the use of all-IP technologies and concepts, as practical means to provide the Internet world benefits. By achieving additional targets like for example the Fixed Mobile Convergence (FMC) between the two main branches of the telecommunication industry, but also by being the architecture of choice today for the deployment of telephony services on the latest generation of mobile networks (Voice over LTE (VoLTE)) as well as the latest service delivery platform (RCS enhanced (RCS-e)), towards the completion of this work the IMS is the clear architectural winner for the NGN-Control scopes, convergently allowing for Access and Applications from various domains, hence it can be stated that:

$$\mathbf{NGN}(Control) = \mathbf{IMS}$$

However, a big differentiation point between IMS and an Internet world VoIP and services platform would be that IMS comes with built-in Telecom characteristics. Hence the fabric of IMS is woven with high-grade security, reliability and performance threads. Unlike for the best-effort Internet services, Telecom NGN architectures are subject to high standards of performance, being able to provide predictable and always available services, while also ensuring high capacity and security for its subscribers and their data. From this results the critical challenge for the IMS architecture, of realizing complex service platforms similar to the innovatory best-effort OTT offerings of the Internet, yet at the high standards of the Telecom world.

Telecommunication Test-beds refers to the laboratory and trial environment organized by today's Telecommunication Service Providers (TSPs), Telecommunication Equipment Manufacturers (TEMs) R&D departments, universities as well as other research institutes around the world, in order to test the newest generation of communication architectures, concepts and equipment.

One of the most referenced definition of a **test-bed** is provided in [23]

> *"An environment containing the hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test."*

R&D is defined in [24] as:

> *"Research and Experimental Development comprise creative work undertaken on a systematic basis in order to increase the stock of knowledge, including knowledge of man, culture and society, and the use of this stock of knowledge to devise new applications."*

Open Source refers to the software licensing and development models in computer engineering where the source code used to generate the executable program binaries is provided[2] to other developers. This creates a different business environment and model than the traditional one, where know-how as well as testing and development costs are shared between multiple parties. Everyone enjoys extensive freedoms of expression such that innovation is free to flourish, even from small stake-holders in major projects, without the typical borders of large structures[3]. The Open Source model has witnessed a broad adoption and also contributed significantly to the evolution of the Internet. Also in the telecommunication industry it is used more and more as an effective method of both improving the TSP's agility as well as for lowering the Total Cost of Ownership (TCO) of their communication networks.

1.3 Scope

The immediate scientific scope of the work will be the satisfaction of the stringent demand for practical experimentation, hence test-beds, for the emerging NGN technologies and concepts. Two principal domain scopes are to be followed:

- the use within academia, for bringing NGN into the higher education curriculum, by enabling student level experimentation and education on the cutting edge of telecommunication domain technologies. Major universities should be enabled with such test-beds in order to both teach as well as innovate new concepts.

- the use within telecommunication operators, as vendor independent tool-kits, enabling pre-deployment experimentation for verifying feasibility of architectures and standards-based functional elements, understanding, identifying and eliminating the technical challenges, as well as enabling a quick path for innovation by adding a short-loop idea-to-prototype-to-product alternative path to the established yet slower cascade standardization model.

Telecommunication systems span today at a world-wide scale and are virtually used in all human activity. Presenting then a comprehensive taxonomy is a complex task in itself. For the purposes here, the representation axis is a technical one related to the communication area and domain, such that the first level can be split, as indicated in Figure 1.2, into *Access*, *Transport*, *Control* and *Applications*.

Critical is also the scoping towards high-reliability communication. While the Internet OTT applications of today do not usually enable this, the Telecom domain is entirely different. For example the Circuit Switched (CS) services are regarded as a gold standard for reliability and even recent networks that require utmost

[2] Most of the time without costs

[3] An extensive and very good introduction to these models is presented by Eric S. Raymond in his *The Cathedral & the Bazaar* [25]. The closed monolithic system of developing software is compared there with the open contribution model of the Open Source communities.

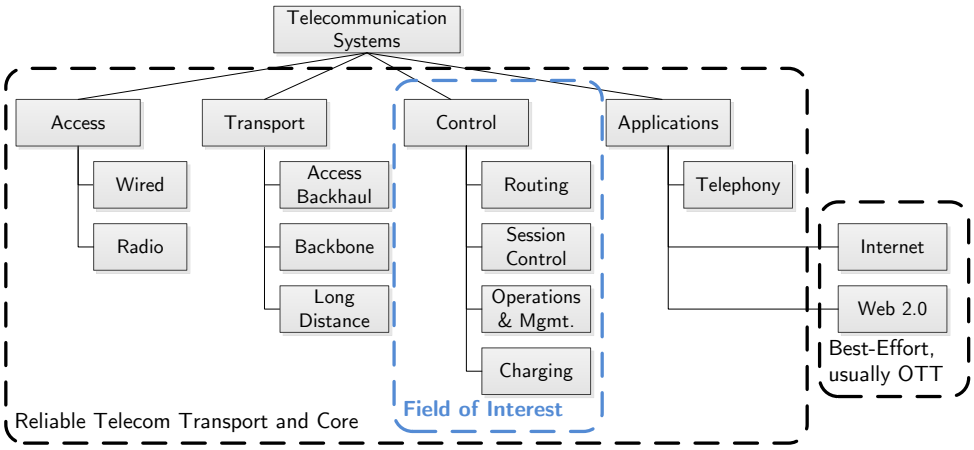


Figure 1.2: Major Aspects of today's Telecommunication Systems

reliability (e.g. GSM for Railways (GSM-R)) are still predominantly relying on legacy CS for safety reasons, even as Packet Switched (PS) networks certainly are more efficient in providing performance. CS services appeared first and were based on simpler concepts of establishing dedicated communication paths between endpoints. As data transfers increased in volume, yet of course exhibit a different load characteristic than voice communication, PS became more attractive as providing better efficiency and use of resources. The PS services have fueled an explosion of richer multimedia communication and enabled best-effort applications like Instant Messaging (IM) or VoIP, yet these applications and use of PS are not in scope here. Rather the end-to-end Quality of Service (QoS) and security enabled and hence reliable communication is in scope, of course extending the offerings of the OTT applications, yet providing the additional high standards for services of the traditional Telecom operators.

Based on the telecommunication systems taxonomy, various networks and operators specialize and emphasize on various domains, like for example the Telecommunication world being oriented towards providing *Access*, *Transport* and *Control*, yet relatively limited in the *Applications* space. On the other extreme, Service Providers on the Internet would concentrate first on the *Applications* and then the *Control* domains.

The scope of the present thesis is defined by the IMS as a CN control architecture, which in the presented taxonomy is positioned in the *Control* domain. This will be in effect a convergence middle-ground between the traditional Telecommunication and Internet worlds.

The IMS platform itself also pushes requirements and designs to both the *Access* and *Applications* domains. Yet the architectural design is merely indicating inter-

facing points, but not what are or how would the systems in these related domains work. Accordingly, the thesis scope will be clearly delimited to the main *Control* domain.

Considering then also the motivation which relates to test-bed targets, subsystems like Operations & Management will not be addressed as these would typically only come into play later on in the NGN adoption phases, once real-life deployments would begin.

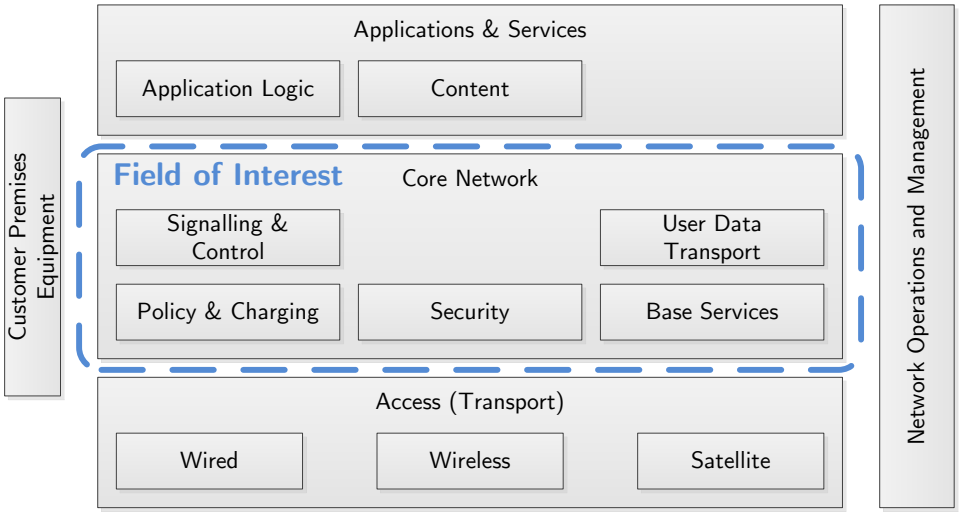


Figure 1.3: Control Aspects within a Converged NGN Reference Architecture

Figure 1.3 puts into focus the Core Network (CN) functionality requirements for NGN architectures like IMS. A typical triple-layer domain is considered, where the CN has the role of providing control capabilities, between the Access & Transport domain (encompassing various wired and wireless access and transport technologies) and the Applications & Services one (where the various communication and control building blocks are used to create complex communication scenarios). The main functional elements of the Core Network are: the signaling routing and processing; the transport, policing and charging of the user data; the security monitoring and enforcement capabilities; the base services, which provide the basic building blocks for complex applications.

As the IMS architecture has been historically started and driven through its major features by 3GPP, the mobile domain has the most important impact on this evolution. Of course, the scope will not be limited to the mobile domain but regarded in a converged networks view. Yet as depicted in Figure 1.4, the mobile domain has always been the driving force and as such will be primarily addressed.

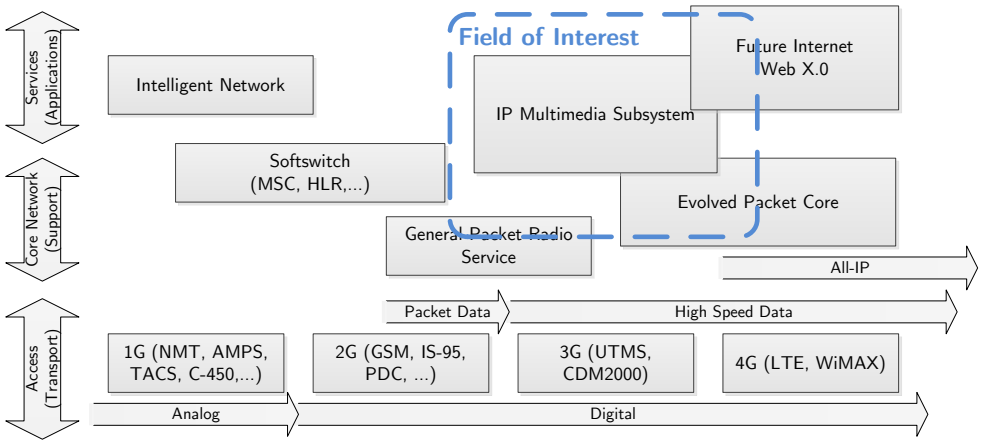


Figure 1.4: Evolution of Control Platforms in Regard to Mobile Systems Evolution

To continue the scope definition, the thesis will concentrate on the IMS architecture and the all-IP convergence with the Internet domain, while also taking a brief look at future NGN complementary architectures like EPC for validation purposes.

Although for the Internet world Open Source has been proved as functional part of its fast evolution mechanisms, in the telecommunication business it made only small steps so far. The thesis will build on the believe that there is big evolution acceleration potential for introducing Open Source directly at the core of future telecommunication networks. The first result presented will be that of the OpenIMSCore project [26], which aimed at accelerating the standardization, evolution and adoption of IMS.

The evolution of telecommunication architectures, concepts and protocols mixed with the boost provided by Open Source, all placed on a common industry/academia ground, is the focal point here. The presentation will follow the prototypical implementation of the NGN architectures into their integration with the Internet world. The principal motivation will be to satisfy the requirement for the open, extensible R&D infrastructure which is much needed in order to validate the new technologies and accelerate their adoptions.

The implicit targets to be used as a vehicle for thesis validation, will be represented by the successful design and implementation of the Open Source IMS Core (OpenIMSCore) as a toolkit providing much-needed NGN infrastructure in the form of comprehensive CN prototypes for IMS. As a secondary implicit target, an additional effort will be made to ensure future re-usability and long exploitation life for the software framework designed and implemented as part of the primary targets.

The introduction of such Open Source projects will also represent a significant opportunity for the academia and research communities to enjoy a low entry-cost for trialing the next generation of telecommunication architectures. In the past access to these technologies was (and still is currently) severely limited due to the closed nature of the TEMs business models. For example, as it was very difficult for students to interact with equipment from the Intelligent Networks (IN) generation, innovation from the academia failed to materialize.

To summarize, this dissertation will present the concepts of establishing, running and exploiting such Open Source NGN toolkits in universities or research institutes. The work will highlight the relevant aspects of developing such projects, with insights into the licensing models, community fostering, industry and academic cooperation. The primary target will be that of accelerating the NGN technology adoption by providing open and solid R&D infrastructure. The secondary target is to establish a blueprint for introducing open source in a previously closed domain, as a bridge between the academic world and the established industry, meant for fostering evolution of the domain. Although the work will have an important level of detail into the NGN evolution of the telecommunication industry, the concepts are general enough not to be limited to it.

1.4 Key Questions Addressed by the Dissertation

While the first section of this chapter briefly introduced the motivation of the present work, the second one limited the work domain creating the focus area. At this point the key scientific and technical aspects served by the work can be defined, as directly derived from the points enumerated in the motivation.

The key questions of dissertation are then split into 3 principal ones and 3 secondary targets.

1.4.1 Principal Questions

Q1. What are the key design principles and implementation challenges for an NGN test-bed toolkit?

Identifying the forces that would shape the targeted test-bed tool-kits is essential to a correct execution and then by extension to the resulting toolkits. Hence a proper analysis of the initial situation will be required, followed by careful design as a composition between state-of-the-art and requirements, and then the proper execution of the design plan as to maintain in the implementation a good representation of the design goals.

This question defines in effect the principal engineering process, which will combine scientific and practical knowledge to first design and then implement a solution to the existing problem of the NGN introduction and evolution.

Q2. Is the Open Source model feasible for such NGN prototype implementations?

As the motivation and scope definitions indicate, the target at hand is a very complex and risky one. Hence a proper strategy will be required in order to master the problem and reach a successful outcome. Open Source will be taken then as the foundation of this strategy, the value-multiplying catalyst, which will ease on the difficulty of the problem and act as a lever for reaching the major goals.

The approach is also novel as, even though operators have used Open Source software before (mostly as cost-reducing mechanisms), still the OpenIMSCore project was in fact the first to target the implementation of a comprehensive CN functionality set as Open Source. Successfully reaching the results will provide as secondary result a validation that Open Source is a serious alternative even for the most critical parts of the Telecom domain.

Q3. Does an Open Source model bring benefits for the R&D of NGN?

Although directly Open Source is used as a strategy to ensure success, in itself it has the potential of enabling more openness in the R&D process of evolving telecommunication networks. The current cascade standardization models are largely denounced for their lack of immediate feed-back loops and their relative closed approach in regard to the academia. Hence Open Source, through its freedoms of expression, will provide here for the additional opportunity to allow a much leaner evolution of the architecture, directly driven by a shorter feedback loop through practical experimentation and empowering the academia with the tools to make an impact on the evolution of these networks.

1.4.2 Secondary or Indirect Questions

Q4. Testbed scalability and Performance: can small scale testbed results be used for real networks?

The scope of this question is to follow on the relevance of the results. While the experimentation with such prototypes would allow for results to be produced, a solid analysis must follow on verifying if those results are correct in the sense that they will verify later on, during real-life deployments of IMS as the NGN solution.

Answering to this question will validate the initial assumption that practical early experimentation on the architecture will help accelerate its adoption in real-life.

Q5. Can innovation be fostered by bridging the gap between industry and academia with independent and affordable (eventually reference) prototypes?

As legacy telecommunication domain equipments were notoriously closed, coupled with the high complexity of the domain as resulted from having to make-do on stringent requirements with rather limited hardware resources, the academia was mostly left-out of the innovation cycle, at least when compared to its impact on the Internet domain. The proposed prototypes will have the potential of empowering the academia with the tools required to foster innovation and push to the industry

new radical concepts, similar to how they are pushed today in the fast-paced Internet globally connected world. If successful, the model can be re-used for future prototyping of even more advanced network architectures.

Q6. How to evaluate and quantify the usefulness of the resulting toolkit for test-beds?

The present question goes towards defining the success criteria for the resulting implementation. As the approach is novel, also novel evaluation principles are required. The validation parts of the dissertation will attempt to provide also the tooling for quantifying the resulting toolkit.

1.5 Methodology

An overview of the model which will be used in the thesis is presented in Figure 1.5. For the practical realization of the implicit targets, there will be 3 main input domains:

1. *Influences* as arriving from the current state-of-the-art and future trends in the telecommunication domain.

   One of the most prominent influence is represented by the current convergence of the Telecommunication domain traditional architectures and the Internet world. Steamed primarily by the movement in the telecommunication standards towards the adoption of all-IP protocol stacks and similar models, the current transition from highly specialized architectures towards common concepts and re-usability, holds high promises for cost reductions as well as much improved agility for the current major TSPs.

   Border lines in the market between network operators and service providers are already very difficult to define, as provided services and their delivery become more and more ubiquitous. Accordingly, the scopes will have to take into consideration influences and requirements from both sides: traditional operators and the new service providers.

   Traditionally the academic community had relatively limited or isolated in time influences on the evolution of the communication networks. On one side this is natural as with high costs for establishing communication networks comes also a need for a long-lasting serving infrastructure, allowing for optimizations only at long intervals. On the other, the Internet world proves that communities, like the ones based on Open Source models, are also capable of self-organization and evolution on a much faster pace. The Computer Engineering and Computer Networks faculties around the world seem to have a higher and more direct influence on the Internet evolution with freely spawning prototypes and research, than their Telecommunication System faculties on the traditional operator evolution. In the next chapters the work will build

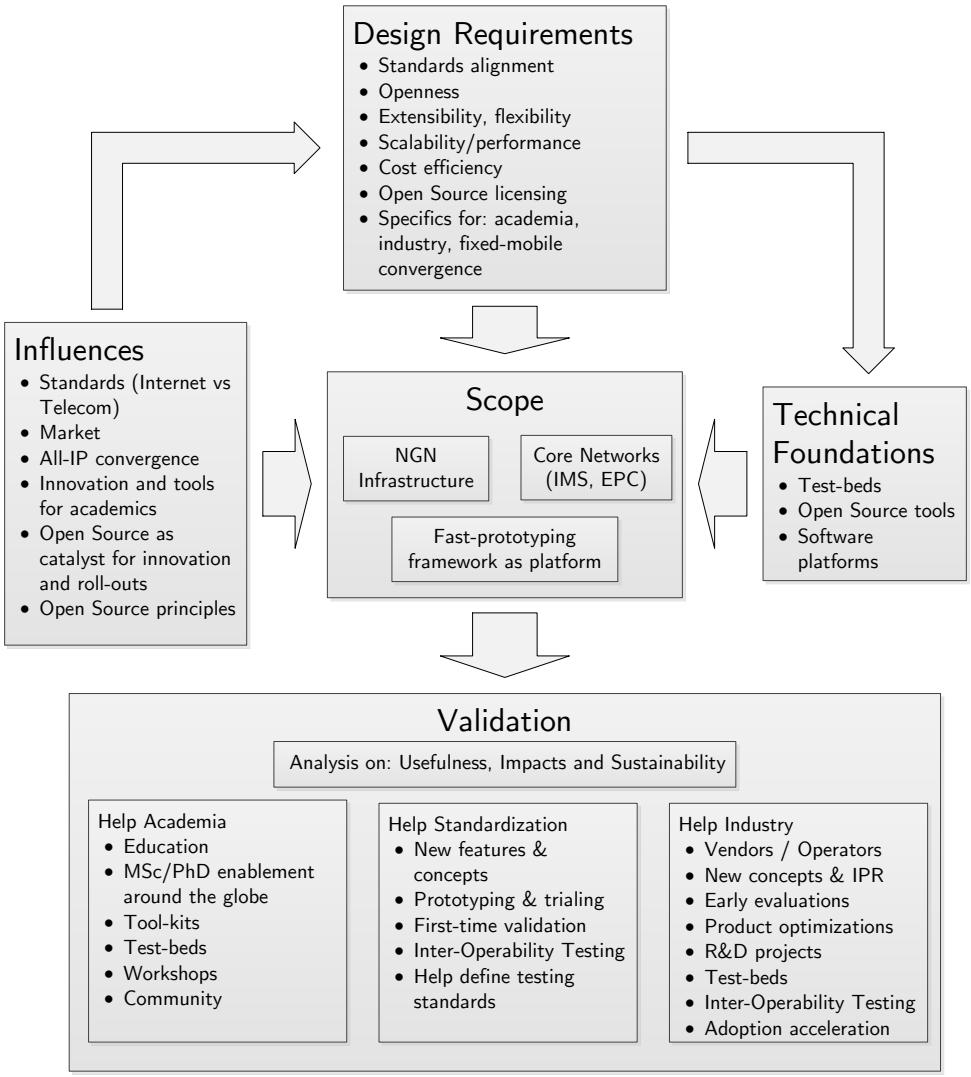


Figure 1.5: Overview of Used Methodology

upon the idea of using Internet world models (like Open Source) in the traditional telecommunication domain, such that at least the first evolutionary steps for development and introduction of novel architecture could be significantly improved and accelerated by empowering also the academia to have a direct participation.

Ultimately, as Open Source principles will be embraced, the respective models and peculiarities will be taken into account while developing the scope, such that healthy communities will be created, through which solid validation will

occur.

2. *Design Requirements* as to ensure successful execution and predictable results.

   Current CN architectures have a high tendency to converge. This trend is natural as networks span across borders and continents and there is hardly space for significantly specialized local solutions in a globally connected world. The number of combinations for inter-working between different network types also negatively contributes to the cost of networking equipment, further strengthening the technological convergence trend.

   Considering these trends, the scope will be targeting only globally recognized solutions, which span widely also on the transport as well as service axes. Alignment with the existing and future standards and specifications would be critical first to the identification and adoption inside communities of the projects, later also to the overall success. Here a *flag-in-the-ground* approach will be taken, where even if full compliance is not economically feasible, the prototypes will align as much as possible to the latest standardized concepts. The flag will indicate the standards alignment level, while the actual prototypes will grow progressively with the most important features coming in first and the least important deferred for later or even never.

   A long practiced strategy (or pitfall, depending on where one stands) in the telecommunication sector is that of vendor lock-in[4]. Against this the Operators are demanding functional split in equipment and standard interfaces[5] between them. In this context, the project will have to align to a certain "openness" standard, such that the utmost importance will be given to ensuring alignment to the standards in such a way as to improve interoperability to the maximum. If in certain situations this will not be feasible, or if new concepts which are not yet standardized will be implemented, modifications will be documented and always disclosed. Without this provision, the results will not enjoy a broad acceptance and adoption as interoperability is critical both at the inter-domain interfaces, but also inside each CN architecture.

   Next a series of practical requirements will be considered. First of all, decent performance will have to be provided, or else any tests would not be comparable or significant in regard to real-life exploitation. Coupled with the cost efficiency, performance should be sufficient to service from simple test-beds requirements[6] up to realistic field-trials[7]. To limit though the costs, real-life

[4]Through vendor lock-in, the TEM introduce proprietary modifications to their products, motivated most of the time by the need to cut corners in the quest to reduce costs. While standardizing these modifications would also induce extra costs, they are most often kept proprietary, with the added side-effect that the Operators will eventually be locked into the respective TEM offerings, which are the only ones providing flawless and guaranteed interoperability. The negativity comes from the reluctance of the TEMs to disclose the proprietary changes, as an effective way of reducing market competition. Breaking out of a lock-in often requires a major network overhauling.

[5]Interfaces are commonly referred to as "reference points"

[6]Typically single-digit range for number of client devices, sessions, applications

[7]Operators normally conduct pre-launch friendly-user-trials with up to several thousand sub-

performance will not be targeted, such that high performance coupled with high-availability and security will be considered out of scope.

Following on the cost limitation, the realized prototypes will target commonly available hardware and software platforms. As much as possible the functional components will not rely on any proprietary or rare hardware, neither would it involve highly specialized and expensive licensed software. Rather, Open Source tools and platforms would be used as much as possible.

As a good software development practice, the prototypes will be realized with flexibility and extensibility in mind, within modular frameworks. This will ensure that changes and adaptations would not be expensive, while also major re-purposing would be feasible at the end of each project's lifetime.

To round the design requirements, a series of specific adaptations would be required for each targeted party category.

3. *Technical Foundations* which ensure that the implicit targets are based on and building upon solid grounds.

   The last input domain will take into consideration requirements coming from the targeted usage environments as well as what is technically feasible and what are the currently available tools and platforms to be used as foundations. The technical solutions would be chosen primarily on the influences and design requirements above, but then filtered through the practicality factor of the technical foundations.

The *Influences* together with the *Technical Foundations* will represent the external input factors and as such introduced through the State-of-the-Art. The *Design Requirements* will be internal parameters, to be established in order to achieve the planned targets. The subjects of the *Technical Foundations* will be then of course chosen also based on the *Design Requirements*, which in turn will be partially affected by the *Influences*.

The Scope, or otherwise said the implicit targets which stem from the 3 input domain will be represented by the several R&D major projects which will prototype the IMS and EPC CN architectures and as such serve as building blocks and NGN critical infrastructure for validation of the transport/access and application domain in the overall system architectures.

Validation, in order to properly evaluate if the proposed targets have been achieved, will be clearly split based on the targeted audience type:

- *Helping the academia* will be the first and most important criteria to evaluate the success of the work. The thesis will follow on whether the implicit targets in scope can be effectively used in the process of teaching as well as whether they are flexible and open enough to provide toolkits and starting points for future advances in the domain. Results will be judged by considering their

scribers

adoption in academic organizations as well as their influence on specialist conferences, workshop and communities.

- *Helping and influencing the standardization* will also be considered as a validation criteria. The thesis will highlight a series of contributions, first to standards validation, but also to the creation and acceleration of new standards. Also here the use of the implicit targets as proof-of-concept tools for new ideas will be taken into account.

- *Helping the industry* will be additionally considered as a metric of relevance for the performed research, in regard to real-life requirements. The industry will be used as the principal financing factor for the implementation, ensuring as such that the requirements are realistic and the results satisfy the ultimate goals for the telecommunication architectural evolution.

The thesis will start with the introduction of existing solid Open Source toolkits from where NGN prototyping can begin, like the SER [27], which is used even in large real-life deployments and represents a reference for signaling functionality and performance. The main part will follow on the Design and Implementation of the OpenIMSCore project, as a specialized development on top of SER. The validation sections will then round the IMS prototyping as a NGN toolkit, with results and proofs of initial scope achievement.

As an added target, an incremental approach will be studied, targeting the EPC newly introduced IP connectivity platform in the mobile domain. Research projects as well as evolution in network architectures are overlapping and the model will take advantage of this, by overlapping also the development phases. Having a long lasting and parallel evolving prototyping framework will be beneficial for ensuring test-bed transformations in line with the NGN evolutions.

Putting it all together, it is important to note that this thesis will not only concentrate on the design and implementation of the CN prototypes for test-bed purposes, but these are implicit targets. An equally important topic is that of using Open Source tools as effective means for prototyping NGNs, through added innovation fostering and involvement of the academia.

1.6 Major Achievements

These key questions are the principal goals of the dissertation. They represent the technical issues at hand, hence defining the problem to be solved.

The answers to these questions would provide the scientific contributions, as mostly presented in the results analysis in Section 7.2 – Validation through the Initially Proposed Targets.

The main chapters of the work will present then the proposed model, as derived from requirements applied on state-of-the-art and driven by the overall goals. The practical realization is provided as an open prototype implementation, on which experiments can be performed.

Going a bit ahead of the normal flow of the dissertation, the final chapters of this work for validation and summary will provide proof on the scopes defined here, as the resulting OpenIMSCore has enabled years of education in major universities around the world (e.g. Technische Universität Berlin, University of Cape Town South Africa, University of New Hampshire, the Deutscher Akademischer Austausch Dienst / German Academic Exchange Service (DAAD) UNIFI projects and so on) and has also been adopted as a **reference for NGN test-bed toolkits** by both the academia and the industry, especially as virtually all major operators (Deutsche Telekom/T-Labs, Vodafone, Telefonica/O2, NTT, Telkom Indonesia, etc.) are experimenting on new services or even inter-operability with it. Even further, the project itself gathered a vivid Open Source community, with a rich mix of academia and industry innovators being the primary users.

As pre-proof of the academical achievements and impacts, besides empowering academic education and innovation through the open prototypes, the author has (co-)authored 36 scientific publications on NGN[8], contributed significantly to 4 standards covering IMS evaluation and testing, innovated through multiple new patents and also held numerous invited conference presentations and talks around the world. Also worth noting would be the establishment of a recognized academic workshop for Open NGN and IMS Test-beds (ONIT), hosted yearly within high profile Institute of Electrical and Electronics Engineers (IEEE) and Institute for Computer Sciences, Social Informatics and Telecommunications (ICST) conferences. Extensive details are provided in Section 8.2 – Publications, Contributions to Standards and Presentations.

1.7 Overview

From a structural perspective, the thesis will continue with Chapter 2 – State of the Art, where the major NGN architectural concepts and technologies will be introduced. Also special attention will be given to the Open Source models, as they form an integral part of the scope.

To set the input parameters, Chapter 3 will analyze and define parameters for the design, implementation as well as evaluation of the NGN toolkits in scope. Directly drawing from the State of the Art and these requirements, Chapter 4 will compose the design matrix, which serves as a North-Star for directing the project. Following, Chapter 5 creates the blueprint for the project by exposing the design decisions. To round up the remaining implementation process, Chapter 6 presents the realization of the OpenIMSCore project, by following the specification presented in the preceding chapter.

Chapter 7 – Validation will go back to the first parts and put OpenIMSCore into perspective as related to the initial scopes and targets. Validation will continue by presenting real-life use-cases and success stories, demonstrating as such that the result is a successful NGN toolkit. One of the most important aspect of the valida-

[8] 14 of them being directly on the IMS immediate scope.

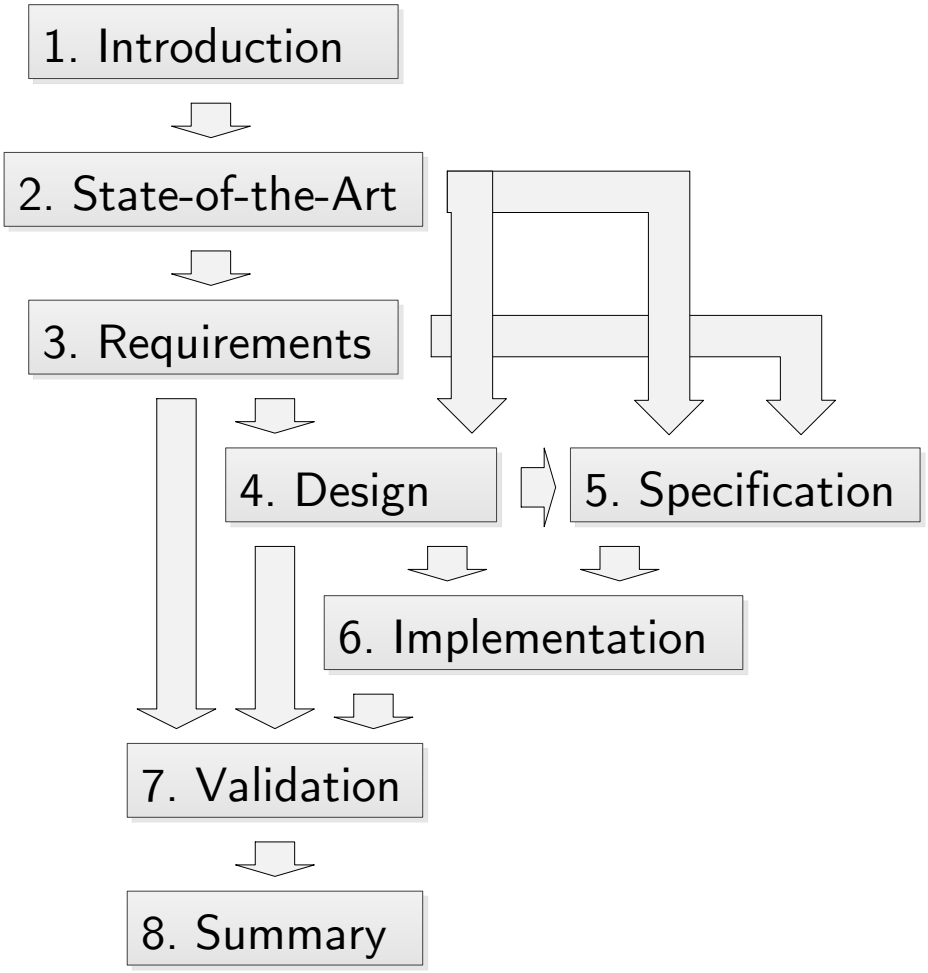


Figure 1.6: Structural Overview of the Dissertation

tion is that even after almost 10 years since its inception, the OpenIMSCore project is still a relevant element of the R&D test-beds. By simply providing integrations with the EPC connectivity platform and architecture, the project remains at the core of current VoLTE experimentation to bring telephony in the Long Term Evolution (LTE) market, as well as to finally transition it to all-IP platforms. The last chapter of Summary & Outlook closes the dissertation with a view on the future research and application of the learn experiences.

While the Table of Contents, List of Figures and List of Tables precede the main contents, the Bibliography, the List of Acronyms follow after. Appendices will be presented last, including Author's Publications, Contributions to Standards and

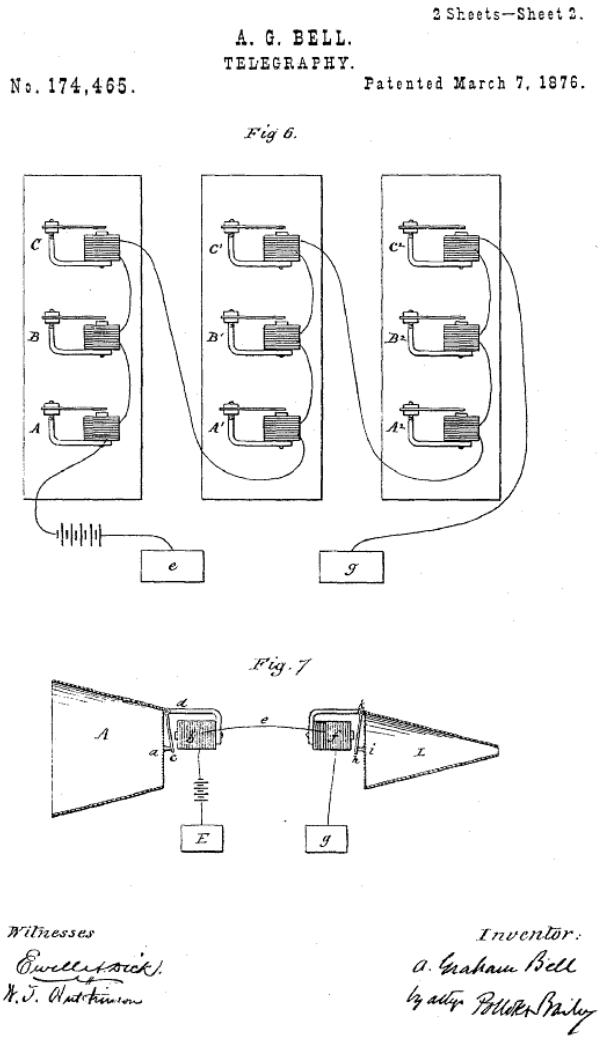
Presentations which comes to support the work and claims.

Chapter 2

State of the Art

2.1 Telecommunication Concepts and Technologies

To understand the driving forces which shape today's telecommunications industry, a short look at the historical evolution of telephone communication since the 19th century is required.


2 Sheets—Sheet 2.

A. G. BELL.

TELEGRAPHY.

No. 174,465. Patented March 7, 1876.

Fig 6.

Fig. 7

Witnesses

Ewell Stock.

H. J. Hutchinson

Inventor.

A. Graham Bell

by Atty Pollock & Bailey

Figure 2.1: Patent US 174,465, Regarded as the Invention of the Telephone

Since its invention in 1870s by Alexander Graham Bell [28], the telephone system

and the associated communication networks have been continuously evolving. Until the 1950s the main focus has been on improving the technologies such that voice communication could be performed over increasingly longer distances, with better quality, while also continuously reducing costs.

Then, as summarized by R.L. Bennett in [1], in the 1950s, fostered by the cost reductions, connectivity started to become attractive also to specialized communities, which had requirements well beyond the offered telephony facilities at that time. While in the 1960s and 1970s such new services have been introduced, the lack of proper technologies delayed their adoption until the 1980s, when the advances in computing and space industries had been made available also to the general public and the communication industries.

In his article from 1993 Bennett proposes and encourages the IN concepts. Earlier advances and architectural changes evolved telephony switching systems to common control, in order to no longer keep blocked the call establishment resources during the call. Similarly yet a major step further, the 1990s transformation was to be driven by the shift of the switch data and procedures into a split-architecture between the interconnection layer, providing the basic connectivity features, and the services control layer, which interfaces with and provides the communication services. Bennett summarized this trend and change as:

> *"In summary, the telecommunications industry, which has been interconnection-driven, will, in the future, be service driven!"[1]*

This model for the "Network of the future" (see Figure 2.2) was first instantiated as the IN network evolution, which unfortunately was too much plagued by a closed architecture which significantly slowed its evolution. A much more open model, yet based very much on the same architectural concepts, was created in the 2000s, as a combination and alignment with the Internet world and communication models, as the IMS architecture, which is in fact the topic of this dissertation.

Following Chapter 1, this chapter will introduce the concepts and technologies of interest, limiting of course to the relevant ones in scope. This state-of-the-art targets merely an overview presentation, highlighting the most important architectural designs, without delving too much into the gargantuan complexity of the field.

2.1.1 First NGN: Soft-switching and Intelligent Networks (IN)

To understand the main topic of IMS, an overview of the architecture from which it evolved is necessary. The Intelligent Networks (IN) architecture pioneered the "Universal Service Platform" concepts in the telecommunication industry, where a middleware platform was introduced between the network with its resources on one side and the services with the respective environments on the other. This important decoupling allowed for the first time services and networks to evolve freely and independently. Important savings were to be made as no longer very expensive service changes were mandatory once a new network technology was to be introduced. Neither would networks need to be re-engineered or upgraded, in

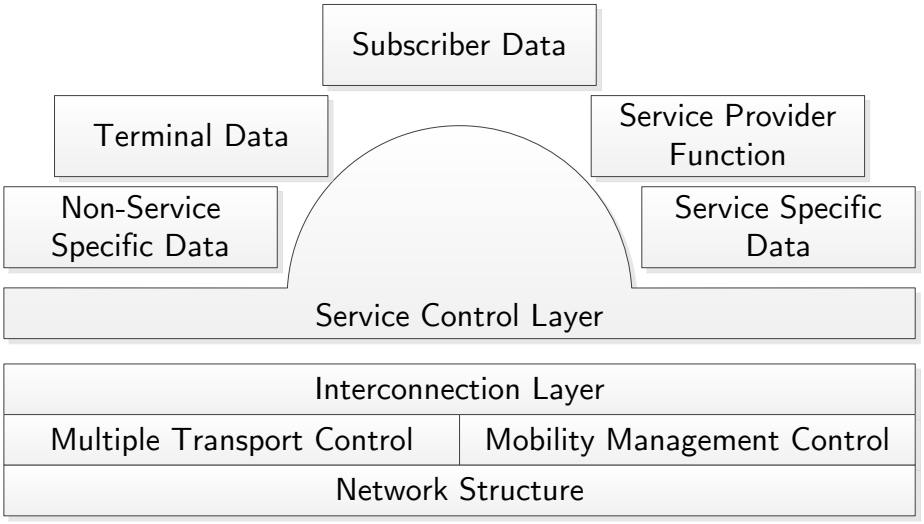


Figure 2.2: Network of the Future: Components (from [1])

very extensive and time-consuming operations, to support the introduction of a new service.

The IN concepts can be tracked back to February 1985 [29], when the Regional Bell Operating Companies have started an initiative with the following main scopes:

1. allow for fast introduction of new services;
2. define clear interfaces between components, such that operators would have a good choice of vendors and equipment;
3. allow 3rd parties to provide new innovative services over the typical operator networks.

This simplistic view is probably one of the best indications on what an IN was expected to provide, on the 3 levels: fast new services; competition and interoperability between various equipment manufacturers; allow external parties to develop specialized services as they would require them.

The IN paradigm was built firstly on the concepts and capabilities of the existing telephony signaling protocols at that time, Signaling System # 7 (SS7). The concepts called for a split in architecture, such that the call handling procedures (happening in the so called Service Switching Points (SSPs)) would be "hooked", call processing would be suspended and signaling forwarded through the Service Control Points (SCPs) components, which would add intelligence to the processing, as suitable for each advanced service provided.

2.1.1.1 The SS7 Signaling Protocols

From a historical point of view, signaling has started as simple supervisory indication on the status of equipment and network elements, such as on/off-hook to indicate the available or busy status, or pulses to transmit a telephone number, which allowed for the first automatic controls of telephony networks. At beginning these happened "in-band"[1] and used pulses or frequency techniques, with the obvious limitations of being very limited, slow, vulnerable to attacks and obviously sharing the channel with the actual communication [30].

The next evolution in signaling was provided by the move to "out-of-band" methods, where special circuits are to be reserved just for signaling and for managing the other communication circuits. This Common-Channel Signaling (CCS) method is in fact one of the main innovations which spurred the introduction of the Common-Channel Signaling System # 6 (CCS6) set of technologies [31]. Although taking advantage for the first time of the electronic processors, CCS6 was of course severely limited by the technical limitations and digital rates available in the early 1970s. As such, the protocols had to be designed and optimized in a monolithic manner [30].

The Comité Consultatif International Téléphonique et Télégraphique (CCITT) (later renamed to ITU-T) SS7 [32] new layered model was then entirely based on the OSI 7-layer model [33]. Allowed by the technological advances, a layered model provided the much desired opportunity of splitting and customizing protocols based on their action points, as highlighted in Figure 2.3

The first 3 layers, the Message Transfer Part (MTP) are responsible for the delivery of messages on the first 3 OSI layers, ensuring reliability, sequencing and de-duplication, as well as detection of data-path problems, such that signaling flows can be redirected immediately over backup links.

Level 1 of MTP is commonly referred to as "Signaling Data Link Function" [34], represents the OSI Physical Layer and provides transmission path composed of 2 signaling channels operating at the same rate, in opposite direction, such that the communication channel is bidirectional. Typical values depend on the regional and evolution standard and while they started at values of 56 or 64 Kilo Bits per Second (1,000 bits / second) (Kbps) (American National Standards Institute (ANSI) or CCITT), nowadays typical speeds are of 1.544 Mega Bits per Second (1,000,000 bits / second) (Mbps) or 2.048 Mbps (also known as T1 or E1) [30].

Level 2 of MTP is referred to as "Signaling Link Function" [35] and corresponds to the OSI Data Link Layer. Although this layer is similar to other bit-oriented link protocols like High-Level Data Link Control (HDLC) or Synchronous Data Link Control (SDLC), important considerations have been taken due to performance requirements for signaling, especially for avoidance of lost messages and for keeping the delays to a minimum [30]. Messages are transmitted as signaling units of variable lengths. The protocol provides error detection and correction mechanism, as well as retransmissions of unacknowledged messages and flow control.

Level 3 of MTP is referred to as "Signaling Network Function Level" [36] and

[1] As in transmitted on the same channel as the main communication

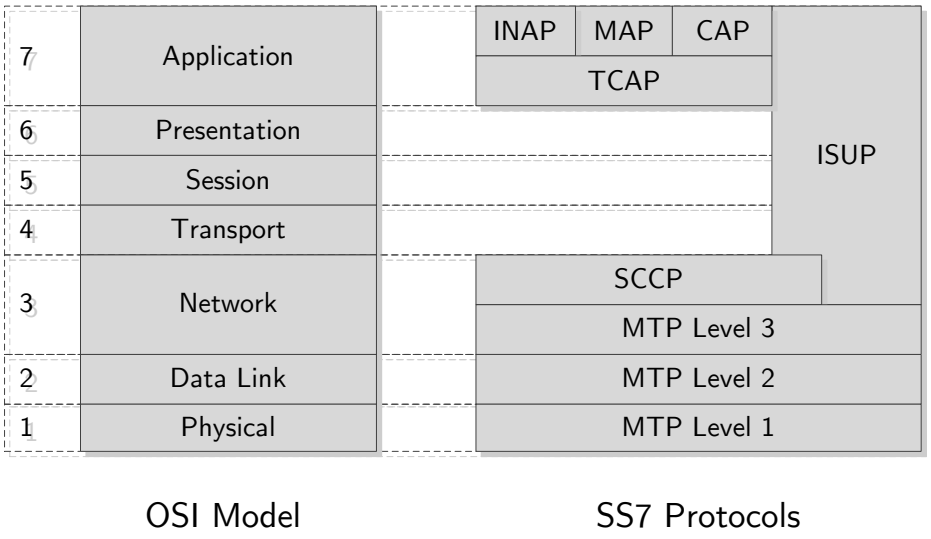


Figure 2.3: The SS7 Protocols Compared to the OSI Model

provides the procedures for transferring messages between Signaling Points (as with system which provide the MTP1 and MTP2 levels), ensuring message routing and reliability of this operation. Addressing and routing mechanisms are introduced at this level, such that messages are reliably transmitted between SEPs, by using nodes which double as message routing entities, Signaling Transfer Points (STPs).

An important aspect of the MTP protocols is that IETF provides adaptations and replacements for transport over IP networks [37, 38, 39, 40]. These variants, commonly referred to as Signaling Transport (SIGTRAN)[2], make the SS7 higher protocols usable also over IP networks, smoothing their evolution path.

On top of the MTP levels, for services requiring additional features, the Signaling Connection Control Part (SCCP) [41] protocol provides extra addressing and routing capabilities, like for example the opportunity of using addresses like dialed digits, which normally would not reflect on the immediate STP network and routing topology. SCCP also provides for 5 classes of protocol connections, ranging from connectionless up to error-recovered and flow-controlled connection oriented ones.

Making use of the transport protocol from the SS7 stack, OSI Application level protocols like ISDN User Part (ISUP) [42] then are able to handle the actual call set-up operations.

For more complex operations, the Transaction Capabilities Application Part

[2]From the IETF workgroup with the same name, http://tools.ietf.org/wg/sigtran/

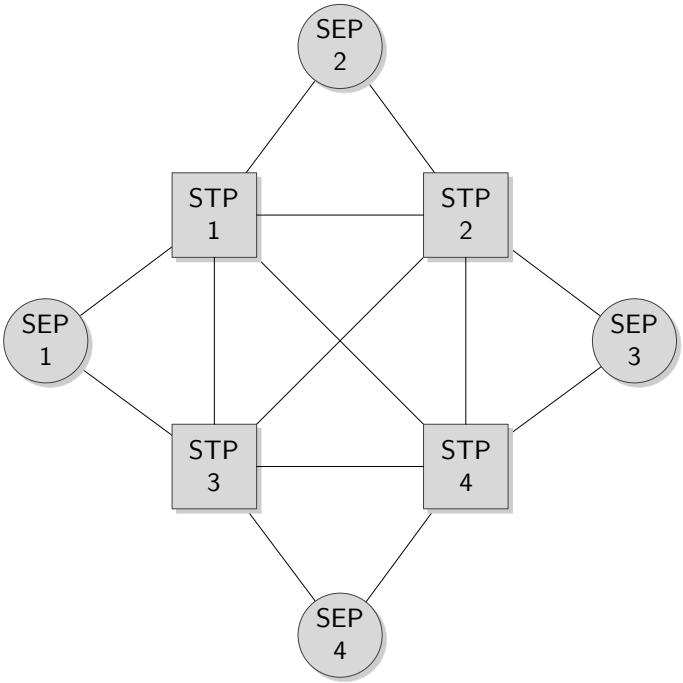


Figure 2.4: A SS7 Signaling Network Example (Note the Full Link Redundancy; Signaling End Point (SEP), Signaling Transfer Point (STP))

(TCAP) [43] provides transaction-like facilities. Operations are defined in a dialog manner, where primitives are sent sequentially after the operation initiation. The dialog is completed by either explicit finalization, abortion from the originating side or cancellation on the terminating side.

The top of the SS7 protocol stack is represented by protocols like Intelligent Network Application Part (INAP), Mobile Application Part (MAP) or Customized Applications for Mobile Networks Enhanced Logic (CAMEL) Application Part (CAP). INAP for example performs the IN service orchestration, providing the application/service logic. MAP provides extensions to the regular fixed network capabilities, required in Global System for Mobile Communications (GSM) and Universal Mobile Telecommunications System (UMTS) mobile core networks, as for example location management, call handling, Short Messaging Service (SMS) and so on. An interesting protocol is then also CAP [44], which provides the capabilities for CAMEL service creation environments, targeting mobile services beyond the base GSM/UMTS applications, to enable third party service design.

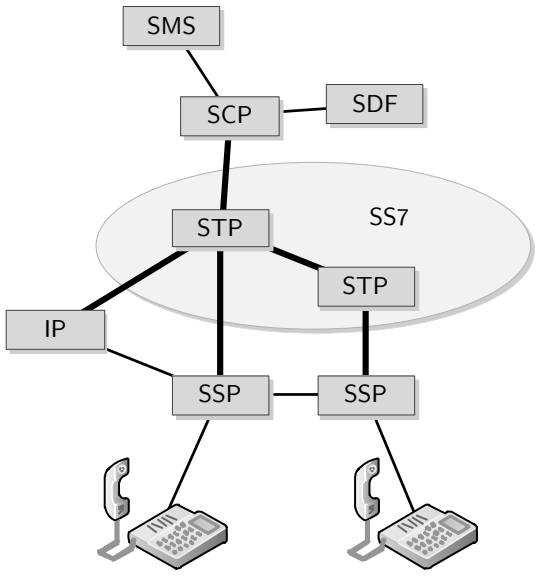


Figure 2.5: Basic IN Architecture [2], [3] (Intelligent Peripheral (IP), Service Control Point (SCP), Service Data Function (SDF), Service Management System (SMS), Service Switching Point (SSP), Signaling Transfer Point (STP))

2.1.1.2 The IN Architecture and Evolution in Capability Sets

The IN architecture bases itself on a functional and evolutionary split in the telecommunication networks between the telephony switches (exchanges), referred to as the SSP (which provide the base interconnection facilities) and the additional service providing functions, referred to as the SCP. In this architecture, the IN platform represents a middleware between services (as in applications) and networks (as in communication resources), ensuring on both sides service- respectively network-independence, which allows then for the targeted decoupled evolution.

IN networks then are based on the SS7 protocols and concepts. The legacy SSPs require upgrades for use in IN to SS7 capabilities, such that they can be interconnected with the SCPs as well as for traffic routing and exchanges purposes with other operators and network domains. The upgrades also included capabilities to interrupt the progress of normal call procedures in the exchanges and to forward call related information towards the service points, interleaving as such the legacy procedure with various hooks and signaling modification procedures which allow for the definition of new communication applications, through reuse and orchestrations of the basic call management procedural building blocks.

Additional functions are then brought in the general architecture, like for example the IP for providing reusable intelligent functionality as announcements, text-to-speech, speech-to-text and so on. On the services side, data storage functions are represented by SDF while the management and service orchestration functionality is provided by the SMS. As depicted in Figure 2.5, routing of the SS7 signaling is then performed as usual in an SS7 system, through STPs.

The evolution of the IN architecture happened in Capability Sets, which are architectural blueprints and service definitions. Each new version brought incremental upgrades to the existing services as well as definitions of new functions and new services. While most of the features pertained to advanced telephony services like the ones listed in the next paragraph, the road-maps envisioned the use also well beyond telephony, into providing mobility, data/multimedia services and even Internet services.

The IN Capabilities Set 1 (CS-1) [45], published around 1993, was the first subset of IN functionality to be defined. For the sake of simplicity on the control plane, it contained definitions of only single-ended and single-point-of-control services. Architecture-wise, this meant that at this point the IN network architecture remained fairly simple in nature. To form a rough overview of what CS-1 provides, the list of services in scope follows:

- Abbreviated Dialing (ABD);
- Account Card Dialing (ACC);
- Automatic Alternative Billing (AAB);
- Call Distribution (CD);
- Call Forwarding (CF);
- Call Rerouting Distribution (CRD);
- Completion of Call to Busy Subscriber (CCBS) [3];
- Conference Calling (CON) [3];
- Credit Card Calling (CCC);
- Destination Call Routing (DCR);
- Follow-me Diversion (FMD);
- Freephone (FPH);
- Malicious Call Identification (MCI);
- Mass Calling (MAS);

[3] Only partially supported in CS-1

- Originating Call Screening (OCS);
- Premium Rate (PRM);
- Security Screening (SEC);
- Selective Call Forward on Busy/Don't Answer (SCF);
- Split Charging (SPL);
- Tele-voting (VOT);
- Terminating Call Screening (TCS);
- Universal Access Number (UAN);
- Universal Personal Telecommunications (UPT);
- User-defined Routing (UDR);
- Virtual Private Network (VPN) [4].

It is important to note that these initial services of CS-1 were meant as building blocks for the future more advanced services. Simple in nature, they provided the Service Independent Building Blocks (SIBs) as a base for the next evolution, which followed into the IN Capabilities Set 2 (CS-2). [45]

The 2nd IN major iteration [46], published around 1997, was a super-set of the previous CS-1 and provided mainly 3 categories of services:

- Telecommunication Services:
  - Inter-network Freephone (IFPH);
  - Call Transfer (CT);
  - Inter-network Premium Rate (IPRM);
  - Call Waiting (CW);
  - Inter-network Mass Calling (IMAS);
  - Hot Line (HOT);
  - Inter-network Tele-voting (IVOT);
  - Multimedia (MMD) [5];
  - Global Virtual Network Service (GVNS);
  - Terminating Key Code Screening (TKCS) [5];
  - Completion of Call to Busy Subscriber (CCBS) [5];
  - Message Store and Forward (MSF);

[4] For telephony only, creating virtual Private Automatic Branch Exchange (PABX) like environments

[5] Only partially supported in CS-2

  - Conference Calling (CONF);
  - International Telecommunication Charge Card (ITCC) [5];
  - Call Hold (HOLD) Mobility services (UPT) [5].

- Service Management:
  - Service Customization Services;
  - Service Control Services;
  - Service Monitoring Services;
  - Other Management Services.

- Service Creation:
  - Service Specification Services;
  - Service Development Services;
  - Service Deployment Services;
  - Multiple Service Management Points Support;
  - Service Creation Management Services;

With CS-2, the network architectural changes were more profound. From a logical perspective, CS-2 introduced several new layers in the architecture, such as the Physical Plane where the SSP and SCP function reside, the Distributed Functional Plane where the basic call state model is being processed, the Global Functional Plane where the SIB introduced in CS-1 and now extended were provided and last the Service Plane where the overall service orchestration, mobility, user interactions and inter-networking were provided. [46]

IN Capabilities Set 3 (CS-3), published around 1999, besides the incremental enhancements and evolutions on CS-2, brings as main new features the Multiple Points of Control concept, which allowed for more than one service logic to apply on the same call signaling parts, independently from each-other, by employing re-triggering. Control of these new interactions was introduced through a Feature Interaction Manager, while combining multiple service logic was also possible, in a controlled manner. Other notable enhancements were the added interoperability with Integrated Services Digital Network (ISDN) basic and supplementary services, the support and provision for Number Portability and additional new features for User-to-Service Interactions. Of interest was also, for the first time in IN, the addressing, at least at a minimal level, of the interactions in regard to IP network services and applications. [47]

In IN Capabilities Set 4 (CS-4), published around 2001, the evolution of CS-3 continued. However, the new feature of interest here was the trend and intention of providing IN support for VoIP. In a sense, this represented a broad opening towards interactions with the IP networks, especially in audio/video communication. Addressed were the interactions with H.323 [48] Gatekeepers, SIP [49] Proxy and H.248 [50, 51, 52] Call Servers. [53]

2.1.1.3 CAMEL - the IN Customization for the Mobile Domain

The IN concepts started and evolved in parallel with the introduction of GSM. As ITU-T concentrated mainly on fixed networks (Public Switched Telephone Network (PSTN) and ISDN), there was an acute need to similarly support the newer and more advanced services provided by the flourishing mobile networks. Besides the obvious shortage of support for mobility and roaming, the IN standards also lacked in the areas of charging and standardization of the triggering methods [54]. 3GPP, as an international umbrella for multiple regional standardization bodies, had taken upon the task of adapting and enhancing the ITU-T specified IN concepts for the mobile domain.

The Customized Applications for Mobile Networks Enhanced Logic (CAMEL), as specified by 3GPP, aligned and followed with the IN evolution as described in the previous sub-chapter in Capabilities Sets, customizing the original ITU-T specifications for the mobile domain.

The first phase [55] was published as part of 3GPP Release 96 and followed on CS-1 by providing the description of the basic services as building blocks. The second phase followed through with more advanced features as part of the 3GPP Release 97 and Release 98. While the first 2 phases where based on 2nd Generation Wireless Telephone Technology (GSM) (2G) mobile networks, with the introduction of 3rd Generation Mobile Telecommunications (UMTS, CDMA2000) (3G) the 3rd phase of CAMEL also followed in 3GPP Release 99 and Release 4 [56], adding capabilities for UMTS.

The fourth phase continued the evolution in 3GPP Release 5 with the notable recognition and interactions with the upcoming service control and core network architecture, IMS.

IMS was considered at its first introduction in 3GPP Release 5 to be an evolution and replacement for the aging IN and Soft-switch models, towards an all-over-IP evolution. However, even as IMS was initially thought to establish itself in the market together with the 3G UMTS mobile networks evolution phase, the radical changes that it required delayed its introduction and broad adoption well into the 4th Generation Mobile Telecommunications (LTE, WiMAX) (4G) phase, when CS alternatives were no longer considered or available with network operators, hence an evolution to VoIP is mandatory.

This difficult adoption of the all-IP concepts translates then into the broad deployment of CAMEL models and applications at the core of the majority of mobile operators today, with excellence points within the booming of services like pre-paid[6] (which rely heavily on demanding online charging models), roaming or messaging services like SMS or Multimedia Messaging Service (MMS).

[6]Pre-paid is charging model for services with a light relation between subscriber and network operator. The customer acquires credit previously to the use of the service. The operator maintains a live balance of the customer's credit by charging immediately services. Upon exhaustion of credit, access to services is denied until a subsequent top-up. While requiring significantly more complex service delivery systems, the system offers simplicity on service contracting, which made it very attractive to young people or in developing countries.

2.1.1.4 Evolution of Services and Service Platforms in IN

Together with the evolution of IN and CAMEL comes also the evolution of services. Of course, the IN itself provides many telecommunication services. However, for a free evolution of telecommunication applications, decoupled from the network operators and directly driven by the service demand and the application users, the IN architecture has been enhanced with interfaces towards the service building blocks. These interfaces are represented by various Application Programming Interfaces (APIs) which export the capabilities and functionality of telecommunication networks to the service platform evolving in the "dot-com" boom of Internet services in the late 1990s .

The service platform trend here sought to change the norm of introducing expensive new architecture, components and overhauls of telecommunication networks once new services were introduced, with the publications of APIs and introduction of either API executing nodes or gateways towards standard IN functionality.

One of the first push was done by Sun with the Java technologies in the Java APIs for IN (JAIN) activities [57, 58]. This was aimed towards providing adaptations and facilities for Java developers to integrate their applications with SS7 capabilities as well as with the IN provided services like for example call control. The effort consisted mainly in the definition of Service Creation Environment (SCE) and Service Logic Execution Environment (SLEE). The SCE was to support many services, 3rd party independent building blocks, service logic portability and independence in service development from the network. The SLEE represented the architectural blueprint of the functional components which were to execute and provide the developed services in a telecommunication network, such that this could happen dynamically with modularity, independent of network architecture and topology and in a scalable manner.

Also very important are the Open Service Access (OSA)/Parlay APIs, published by The Parlay Group, through their cooperation with 3GPP and ETSI. From a technical perspective, OSA/Parlay [59] provided the API specifications as mappings to Common Object Request Broker Architecture (CORBA) and Simple Object Access Protocol (SOAP) web services, which clearly defined the interfaces between services and the IN networking environment, without actually providing a reference implementation or restricting how the applications would be realized.

Although powerful, the OSA/Parlay API brought with it also a significant complexity. To ease this, Parlay X [60] has been created, as a web services only and simplified API. This architecture has been deployed with operators, starting some of the first developer communities specifically targeting implementation of new telecommunication services from a 3rd party perspective.

Following on, Parlay X has evolved and further aligned with the service concepts in the Internet world. The Parlay X legacy can now (year 2013) be found in the OneAPI effort of GSM Association (GSMA) [61]. Notable advances are visible here in the area of streamlining and simplifying the access to the network operator's provided facilities and the adoption of the new trends in APIs, like for example

REST [62] interfaces, which further eases the developer's adoption[7].

2.1.1.5 Evolution of IN and Shortcomings

Undeniably, the IN architecture, as well as the multiple technologies which it has spawned during its evolution, have a major impact on today's telecommunication networks. Virtually all mobile network operators in the world base their GSM and UMTS networks successfully on this architecture and even more, provide services over roaming agreements at a world-wide scale.

There are though also major shortcomings which have become evident, such that the industry has decided for a major overhauling and evolution with an all-IP architecture, the IMS.

First and the most obvious was the impressive complexity of the SS7 and INAP protocol stacks. For sure one must not disregard the high demands and requirements on today's in-production IN networking equipment, yet the success of the IP based networks is too tremendous to ignore. It is a recognized fact that when operators wish to cut costs on certain data communication links, using IP equipment is a good and easy solution, through large economies of scale, commoditization and seamless interoperability. Looking back at the initial objectives of IN, this was also one of the key motivators.

Then the high complexity of IN translated into only a small number of manufacturers being able to provide equipment at the needed compatibility and quality levels. The IN architecture is hard to comprehend and to master for developers, as demonstrated very clearly through the simplification trend of the offered APIs over time.

One other shortcoming was due to parallel standards: for North America the Advanced Intelligent Network (AIN) by Bellcore/Telcordia and for Europe the INAP by ETSI. While they share the same ITU-T roots and are similar from a functional perspective, they are essentially incompatible at the protocol and encoding levels.

The existence of many and sometimes even parallel standards meant that in practice equipment would rarely implement the full set of standards specified features. This meant that in reality deployments had incomplete coverage of features and were then affected also by interoperability issues. Coupled with the fact that telephony switches had to be upgraded to support interfacing with IN, which again were not always fully implemented [64], meant that the initially targeted vendor-lock prevention of IN was in reality present with the opposite effect.

In the booming mobile domain, where operators enjoyed consistently high revenues, such expensive architectures could still afford a healthy evolution. Yet other domains like fixed or broadband networks, where the Average Revenue Per User (ARPU) is much lower, have seen only limited advances and evolutions, being practically reduced to flat-rate basic services without much value-added advantages

[7] "Amazon has both SOAP and REST interfaces to their web services, and 85% of their usage is of the REST interface" [63]

and used only for their transport capabilities as access solutions for the Internet services evolution.

To conclude, IN systems are often regarded as very expensive, complex and demanding architectures. A simplified model based on the IP concepts and protocols was required for the evolution of telecommunication networks not only in the mobile but also again in the fixed and broadband domains.

2.1.2 IP Network Principles

The evolution of telecommunication networks and concepts around the year 2000 was attracted towards the successful and broadening adoption of Internet services on a world-wide scale. Of course the cutting edge IN concepts were already starting to play along with and reuse the IP concepts and networks. These concepts and in fact the Internet itself, as a global communication network, even although based on and reusing the standard telecommunication networks for data transport, had a parallel evolution of concepts and architectures. This section seeks to expose the core concepts and architectural driving forces beyond the Internet world, from a technical protocol view.

2.1.2.1 TCP/IP and the Birth of a World-wide Network

From the historic perspective, data communication can be dated back to the telegraph or the first telex machine. A significant push here was started in the 1960s, when under the influence of J. C. R. Licklider publications on the man-computer symbiosis and communication [65, 66], the first PS network, Advanced Research Projects Agency Network (ARPANET) became operational. As it was the first to differentiate from the CS concepts, of course it was using at its beginnings CS to establish links between sites over regular voice lines with modems, yet it was providing a communication network where at any moment any connected host could send datagrams (data fragments) to any other host without reserving a dedicated circuit.

ARPANET was constructed by linking Interface Message Processors (IMPs)[8] with modems over long distance leased lines. Each IMPs was then connected to a local computer, allowing it to communicate with the other computers at remote locations. The initial protocol was Network Control Program (NCP) [67], which although dating before the OSI model [33], still has the physical, data link and network layers well separated.

In the context of ARPANET work continued as founded by the Defense Advanced Research Projects Agency (DARPA) programs in the 1970s, with the notable contributions of Vinton G. Cerf and Robert E. Kahn, which proposed in [68] "A Protocol for Packet Network Intercommunication", the first stepping stone of the Transmission Control Protocol/Internet Protocol, the Internet Protocol Suite (TCP/IP)[9].

[8] Gateways, precursors of today's routers

[9] The Transmission Control Program, later known as the Transmission Control Protocol, was

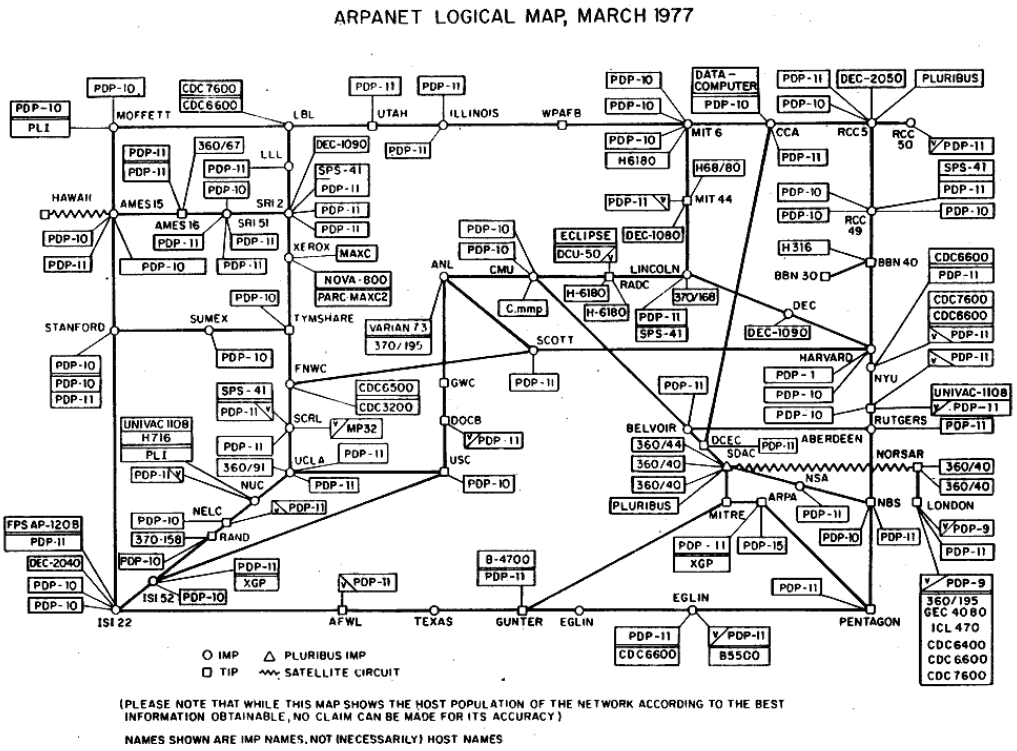


Figure 2.6: The ARPANET Logical Map, March 1977 [4]

Vinton proposed a Transmission Control Program, as the transport means for data between different types of networks, by using hosts (as today's networking end-points, network gateways) connected to gateways (as today's routers, ARPANET's IMPs). Here are some of these initial concepts:

- addresses, which would contain a network identification, Transmission Control Identifier and port;
- individual programs/processes running on hosts which would individually communicate with each-other, identified by ports;
- multiplexing and de-multiplexing of data from/to different processes/ports;
- fragmentation of data into segments as to overcome the issue of different network types with different maximum segment sizes;

in effect proposed before the Internet Protocol, as the reliable means to enable communication for multiple processes running on different hosts located in networks running different protocols. The Internet Protocol itself was then fully specified and, although transporting TCP, IP continued to be regarded as the basic unreliable protocol for communication between different networks. Hence, the TCP/IP naming was then used as a reference to the Internet Protocol Suite, even though from a logical and network layering perspective the naming might appear reversed.

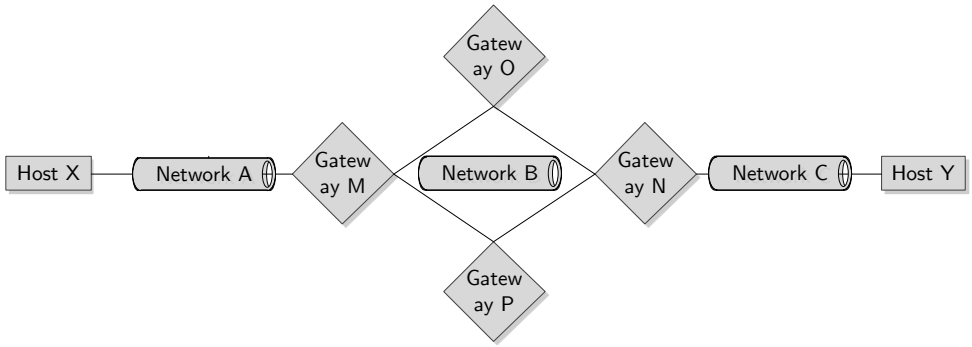


Figure 2.7: Initial Transport Control Program Inter-Network Model

- reassembly and sequencing of data fragments;
- reliability of data transmission by retransmissions, acknowledgements and detection of duplicates;
- introduction of a window strategy for implementing the above features;
- definition of the logical communication connection as an association of remote processes in a connection-free PS environment.

Over the next years, with practical experimentation and refinements, these initial concepts evolved into the de-facto standards of the Internet: the Internet Protocol (IP) [5], the Transmission Control Protocol (TCP) [7], the User Datagram Protocol (UDP) [6] and so on. At the beginning of 1983, the TCP/IP replaced the aging NCP as the core protocol for ARPANET, marking the practical start of the Internet [69].

2.1.2.2 The Internet Protocol (IP) from a Technical Perspective

The IP evolved over the years from the experimental protocol for exchanging messages in the ARPANET environment, to become the fabric itself of the world-wide Internet. The version which is in use today, and in fact also the first version used on a wide scale, is referred to as IPv4 [5].

IP is a simplistic protocol in its definition. Its use now at the world-wide scale is of course not trivial. The operations which it can compose in applications and especially the dynamic routing of packets are quite complex, yet in itself the protocol provides only a set of basic and simple operations, mainly addressing and fragmentation.

Its scope is to allow for delivery of datagrams sent from a source to a destination, over a series of inter-connected networks. This delivery is based on best-effort

principles, without provision for reliability, congestion control, sequencing and so on, which are to be provided by higher level protocols and application (e.g. TCP). Also at the basic IP level, there are no mechanisms for correlating between datagrams[10], or for creating virtual connections or logical circuits. This includes a lack of acknowledgements and even the error control is limited to a checksum on the header only and not the datagram itself. Neither does IP provide multiplexing and identification of datagrams at the host process levels. [5]

Additional mechanisms for enhancing IP functionality exist and are present as different protocols, in a modular way (e.g. Internet Control Message Protocol (ICMP) [70]). A layered model is used here, where basic protocols like TCP or UDP build upon IP, which in turn can be used to build upon other protocols and applications.

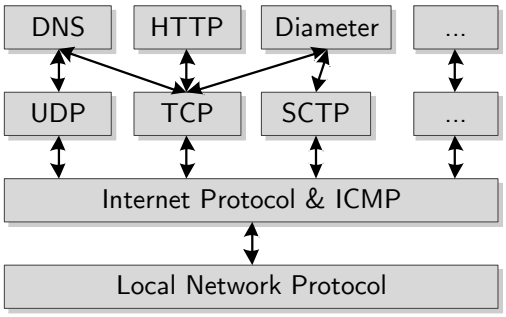


Figure 2.8: IP Protocol Model [5]

The operational model is based on the transport of datagrams[11] from the source host to the destination host. On the path, a series of gateways need only to understand the IP itself to route the packets towards the destination, such that the **applications are provided in an end-to-end manner**. Below IP, each host and gateway uses whatever protocol is employed on the respective transport network: Ethernet, Asynchronous Transfer Mode (ATM), Point-to-Point Protocol (PPP), WLAN (WiFi), etc.

From the basic operations provided by IP, first addressing is represented in IPv4 as a fixed 32-bit address format, which distinguishes between a prefix (representing the address of a certain network) and the remaining suffix (representing the local address of the host on that particular network). Routing itself is a complex operation, which in many situation relies on many other protocols to dynamically adapt routes to changing network conditions like link outages, congestion, QoS parameters

[10] Other than for fragmentation purposes

[11] A datagram is an IP packet, a data segment with the length a multiple of 8 bits, composed of header and data segment.

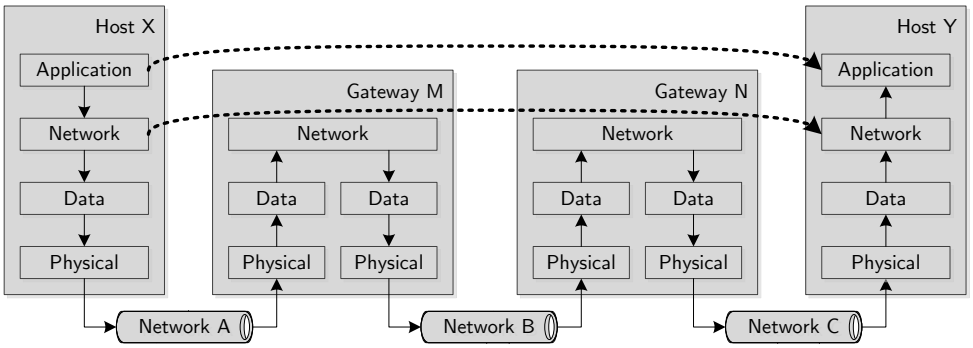


Figure 2.9: IP Conceptual Layered Model

and so on.

The second basic operation provided is that of fragmentation. This is due to the fact that, while transporting the datagram over various types of networks, the Maximum Transmission Unit (MTU) varies and it is rather difficult to discover it end-to-end or keep its value constant as an application logical circuit might be routed over multiple paths. For fragmentation, every gateway has the capability of splitting the original datagram into multiple parts and transmitting them individually. The end host (or even intermediary gateways in some situations) must be able to reassemble the original datagram, even as sequencing is not guaranteed and parts might arrive in a different order. In case of transmission errors, as there are no acknowledgements and retransmission mechanisms at the IP level, an error amplification is experienced, as the loss of just one fragment will result in the discard of the full datagram.

The IP packet format consists of a variable length header followed by the datagram to be send from the source host to the destination. The header itself has first a fixed part, which defines the minimum header length of 20 bytes, followed by a variable length options part, which are suitable for transporting additional optional parameters, raising the maximum header length to 64 bytes, in increments of 4 bytes.

A brief explanation of the header fields follows:

1. Version - for IPv4 this field contains the value 4.

2. IHL - Internet Header Length, represented in units of 4 bytes, after which the transported data follows.

3. Type Of Service - includes Precedence as well as indicators on whether to optimize the packet transport for Delay, Throughput, respectively Reliability.

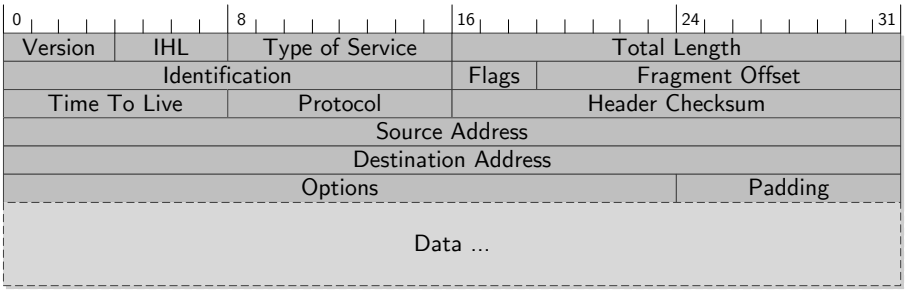


Figure 2.10: IPv4 Header Format [5]

4. Total Length - on 16 bits, allowing for a maximum packet length of 65,535 bytes. In practice, this length is much higher than what the underlying networks support. A lower limit is set at 576 bytes, which is the minimum datagram any host must be able to accept, yet through fragmentation, this can arrive as multiple datagrams. Typical values are around the 1400-1500 range, which matches well some of the most used underlying protocols as of today (2013), without fragmentation.

5. Identification - to help with reassembly of fragmented datagrams. When fragmenting, this field is copied verbatim in all fragments of the original datagram.

6. Flags - for indicating whether the present datagram may be fragmented or discarded if the network cannot transport such large datagrams, as well as the last fragment of a segmented datagram.

7. Fragment Offset - indicating the relative position of the current datagram in the un-fragmented datagram; this is relevant as fragments are not necessarily delivered in order.

8. Time to Live - an indicator in seconds as a datagram expiration, after which it should be discarded if not yet delivered. As each gateway and host must decrement this counter by at least 1 second, it is also used as a maximum hop limit.

9. Protocol - an indicator on what the next layer protocol is, for which this datagrams transports data.

10. Header Checksum - as a simple verification on the error free transmission of the datagram; while the header changes over each gateway on at least the Time to Live field, this is applied on a hop-by-hop basis.

11. Source and Destination Addresses.

12. Options - a set of indicators for security, routing, stream identification or time-stamping.

Building on top of IP, User Datagram Protocol (UDP) [6] adds a simple data structure. A header precedes the actual UDP transported datagram and adds support for multiplexing on the source and destination hosts, such that different processes can associate themselves temporarily to ports. The port is represented by a 16 bit value. UDP also adds protection in the form of a checksum to the actual transported data.

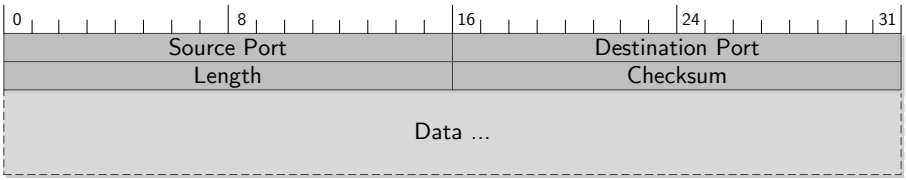


Figure 2.11: UDP Header Format [6]

It is not much that UDP adds to the basic IP datagrams, yet these small additions transform it into the protocol of choice for implementing many of the Internet protocols which do not require features like reliability or in-sequence delivery of datagrams (e.g. Domain Name System (DNS), Real-time Transport Protocol (RTP), Trivial File Transfer Protocol (TFTP)).

For reliable transmissions, Transmission Control Protocol (TCP) is currently the most used protocol. This is in fact the practical evolution of the first concepts for the Transmission Control Program in [68].

The TCP functionality can be summarized as:

- Basic Data Transfer - as continuous streams of bytes, in each direction; the order of the octets is of course recovered during in-sequence delivery of data to the upper layers on the destination host, as received from the upper layers on the source host.

- Reliability - data transfers are acknowledged, such that lost fragments can be retransmitted, duplicates discarded and the data sequence maintained; data segments are protected against errors with checksums.

- Flow Control - such that the receiving end can control the amount of data that the sender is able to deliver over the connection.

- Multiplexing - of multiple processes on the end-point hosts, with a similar port system as in UDP.

- Connections - as a mechanism to provide the reliability features above. Connections are established, maintained and terminated, with the information of the upper layer on their status.

- Precedence and Security - as an optional feature available for the upper layers, with default value and recommendations on their implementation.

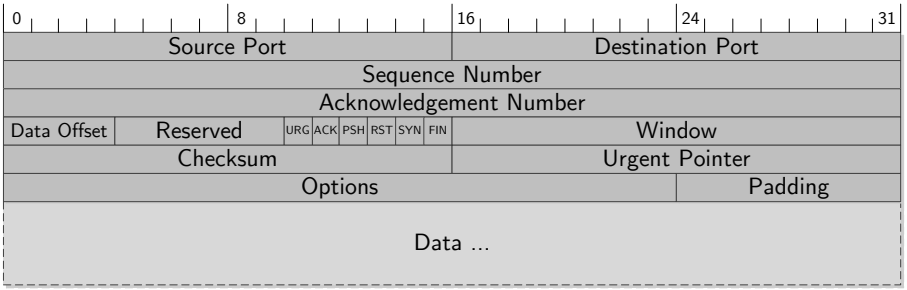


Figure 2.12: TCP Header Format [7]

TCP uses a window mechanism to ensure reliability of data communication. Although large scale and large capacity deployments have uncovered flaws and areas for optimization, TCP is the protocol of choice for a simple and reliable stream-like data connection between two end-points, used with some of the most important upper layer protocols today (e.g. Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP)).

Of special interest in the telecommunication domain is the Stream Control Transmission Protocol (SCTP) protocol [71, 40]. It provides reliability features similar to TCP, but it also extends on them by including message boundaries, such that transmitted data can be reconstructed not only as a continuous byte stream. As its name implies, a SCTP connection, although still of a unicast type between just 2 hosts, can further multiplex more than just one stream of data, as it was the case of TCP, in such a manner that losses or big fragments in one stream will only block temporarily the respective stream, while messages in others continue to be delivered.

To provide high reliability features, as demanded in some of the carrier communication situations, SCTP supports also multi-homing, by which one end-point can use multiple IP addresses, connected to different networks. In case of a network failure, the connection will not be interrupted, but it will continue over the remaining active network interfaces[12], similar to high-availability protocols from the IN domain.

[12]Load balancing of messages through multi-homing is not yet part of the protocol

2.1.2.3 Evolution to IPv6

Without doubt, the most important limitation of IPv4 is today its addressing space. During broad adoption of mobile communication and the always-on devices, as well as due to the increased penetration of the Internet not only geographically, but also on a society basis, and not forgetting the increased use of sensors, actuators and the associated Machine Type Communication (MTC), have almost exhausted the unallocated IPv4 address pools[13].

To track back on the cause for this exhaustion, back in 1974, Vinton G. Cerf, while proposing TCP stated:

> "The choice for network identification (8 bits) allows up to 256 distinct networks. This size seems sufficient for foreseeable future." [68]

Obviously, he did not expect back then the adoption of TCP/IP at such a broad scale and its use not only on the interconnection of networks, but also directly on virtually all data hosts today. Later on, as the first IPv4 standards were realized, he took an executive decision on fixing the address length to a fixed length of 32 bits, allowing for 2^{32} theoretical addresses, out of which in effect about 3.7 billion addresses are usable in practice [8]. This initial decision was mostly made as the believe was of course that IPv4 was only an experiment and in practice this would have been more than enough for such purposes, with an extension happening in due time before the network would grow too much.

While this number seems big even for a world-wide scale, in practice the consumption was quite avid and the number of available addresses to be allocated has been on a continuous descent, such that an exhaustion of unallocated large blocks has already happened in 2011 [14].

The IPv6 [9], as an evolutionary step from IPv4, has an address formed of 128 bits, elevating the address starvation. In a sense then, the IPv6 is the in-production approach to IP, where after experiments and lessons learned, an improved protocol is offered. Only that, the precursor IPv4 was and still is so successful that its replacement is quite a difficult task and is still relevant today, still being more important and used in 2012 than IPv6.

With the new address space, newer and more reliable network configurations have been introduced, like the Stateless Address Auto Configuration (SLAAC).

Besides the significantly expanded addressing space, IPv6 improvements are found in many other areas. For one, the header format has been much simplified towards reducing the processing costs in the network gateways and easing the routing services. The packet format is now more flexible, such that multiple headers and options can be attached. Of relevance is also the progress on definitions

[13] While as of 2012 the IANA address pools have all been allocated, still regional registries and ISPs still have unused or reusable addresses.

[14] The last 2 Internet Assigned Numbers Authority (IANA) unused /8 large network address blocks (approximately 16.7 million addresses) have been allocated on 31st of January 2011. Allocation will continue, yet in significantly smaller blocks.

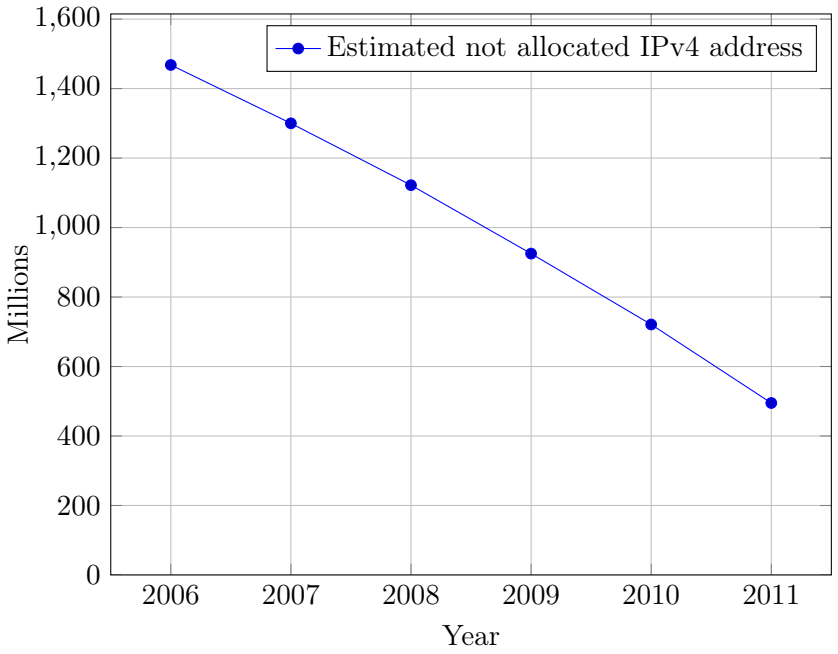


Figure 2.13: Decline of Available IPv4 Addresses [8]

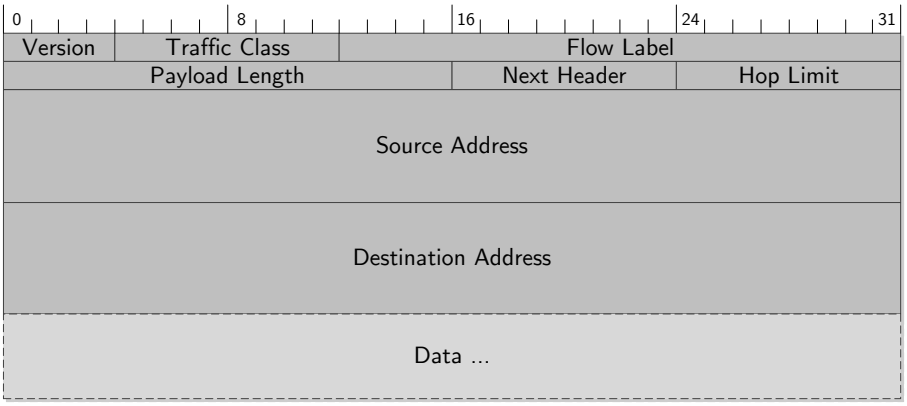


Figure 2.14: IPv6 Header Format [9]

supporting QoS processing of packets, with the introduction of a 20-bit Flow Label, to help with matching individual datagrams to data streams for the purposes of applying a certain routing profile during message routing and delivery.

Topics as data authentication, ciphering and confidentiality, or even mobile IP routing, previously specified as extension, are now part of the base protocol and can

be provided regardless of the application. Multi-casting, which although functional in IPv4, was very complex both for technical but also logistical reasons, is now part of IPv6 core functionality. So is an added capability of any-casting, removing the need for lower-layer hacks and aids in routing packets towards one of the closest destination hosts in a group.

Datagram fragmentation has been redesigned. This is now to be coordinated with the upper layers and should instead happen only in the source host and not on the gateways. For this to be feasible, an additional mechanism for discovery of the minimum link MTU on the path has been introduced [72]. The minimum IPv6 datagram length has been increased to 1280. While the length parameters have been maintained, support for much larger datagrams, known as "jumbograms", has been introduced as extension, bringing the current payload limit to 2^{32} bytes [73].

The adoption of IPv6 continues today, with more and more major service providers offering dual-stack access to their services and access providers deploying it increasingly to customer premises in response to the need of Network Address Translation (NAT) workarounds for IPv4. Gateways and translators are also available, such that IPv4 can and probably will continue to be provided side-by-side with IPv6 for many years to come.

2.1.2.4 End-to-end Principles and a Complexity in the Internet World

While looking at the Internet architectural principles, one of the first and most visible characteristic is its liberty and freedom of evolution. The Internet is not built on a set of carefully designed plans and rules established by a committee of experts, as it is the case within the telecommunication domain. It has a rather different model, which calls for ideas to be implemented first as a best-effort experiments, then rapidly turning into usable components. As these get refined and once multiple alternative implementations for the same concepts and services appear, clarification standards are created to ensure present and future interoperability.

In such an environment then experience and the following of a set of good practice rules is very important, yet also disruptive ideas are encouraged, as each solution is at most a compromise between simplicity and complexity, between cost and benefits, between innovation and legacy.

The Internet is not controlled centrally or owned by anyone, rather it is an exchange of data between different networks, which even if some parts would part ways or disconnect, would still function as a whole. The lack of a functionality-critical centralized control then translates into it being driven by direct experiences on real implementations, rather than being shaped by some limiting architectural constraints.

The Internet does not really have a well-defined architecture, but more of a tradition:

> "The goal is connectivity, the tool is the Internet protocol and the intelligence is end-to-end rather than hidden in the network." [74]

Then looking at the Internet Protocol as a convergence point, arguing that at the network level in the OSI model there should be only one protocol, IP is used on a global scale by all participants. Of course, as a network to connect other networks, the edge ones are not included here. While the single protocol argument might seem attractive as to simplify the global network, in practice this is nearly impossible, considering for one the evolutionary requirements of the Internet Protocol itself, stemming from the changes in requirements.

On the lower protocol layers, the rapid changes of transmission technologies have translated in each providing transport capabilities for IP. It is the case today that these layers are quite well hidden and their influence is not really a point to be considered, with the major exception of course of the back-bone interconnections. Heterogeneity is consider a key factor to follow, as computer architectures, transmission speeds and applications have much higher rate of change than what would be feasible at network layers on the globally interconnected scale.

On the upper protocol layers, the general trend is for end-to-end protocols providing end-to-end applications. The success of the end-to-end concepts arises first with the acceptance of the fact that any networks, no matter how well designed, will eventually experience failures and errors in transmissions. Then if one does not rely entirely on the error-free capabilities of the lower layers, neither expecting that these would keep state of the communication links, end-to-end protocols can focus on maintaining the state just in the end-points and on using the network layer only to exchange information.

End-to-end architectures have been adopted and used by most of the applications in use today. They do not require significant changes in the lower layers when introduced, neither would they be dramatically affected by changing network conditions. Upgrades then can be simply enabled by updating just the endpoints which would require it, with the fabric remaining mostly the same. And overall then, the application state is resilient over the network failures, as long as the state survives in the end-points.

When put into perspective then the model here resembles a "hourglass" [75], where the IP layer remains as a minimalist one at the core of the Internet, which should be as simple as possible, while the complexity is to be found at the edges[15]. This is of course a rather opposite approach for example to the PSTN model, where almost all the complexity was found at the core, in the Class 5 telephony switches, while the edge was provided by rather simple and dumb terminals capable only of simple operations (e.g. voice transmission, simple key input), relying entirely on the core switches for providing all the supplementary services (e.g. call waiting, line identification, etc.). In hindsight, of course, there are many technological factors which forced the PSTN model. As now these have been largely solved, future networks can allow for the edge-intelligence model (e.g. today's smart-phones contain much more intelligence and networking capabilities than the routers and gateways

[15] Of course, there is also state to be kept on the middle back-bone, like for example the routing state. This can be however ignored on the end-points, such that state interactions between core and edge are only rarely of interest and can be avoided by design

of the past).

Simplicity then is the most important design principle to follow for the Internet core. For one, the network complexity does not anymore enjoy linearity in scalability and costs. There are many causes for the complexity escalation, out of which some are enumerated next [76]:

a) amplifications of small fluctuations at small scale into significant effects at large scale;

b) coupling on both horizontal and vertical axes between various communication layers introduce non-linearity at large scale; these could range from resonant effects to cascading failures;

c) increasingly software-dependent systems do not enjoy the same Moore's law simplification and cost reductions as hardware-bound systems, such that the new networks have different models for Operating Expenditure (OPEX) and Capital Expenditures (CAPEX)[16];

d) with the transition from CS to PS, contrary to the popular myth of improved efficiency, in fact the effective network utilization decreases[17] [76];

e) difficulty of ensuring reliability, as the optimization windows is quite low[18] [76].

Accordingly, successful systems in the Internet world have a hard requirement to maintain their simplicity to ensure that they could be reliably exploited on large scales.

The initial concepts of simplicity in the Internet is most often regarded as the Keep it Simple Stupid (KISS), meaning that solutions are to be preferred mostly by their least problematic characteristics. They do not always have to work entirely, neither must they be perfect, as long as the targeted communication is achieved in a simple, cost-effective and scalable manner. Manual parameterization is to be avoided as much as possible and should be done rather dynamically if the increased complexity is reasonable, or by configuration. Also solutions are required to be strict when sending and tolerant while receiving, or even quietly discarding unrecognized messages, such that interoperability would happen loosely and not require strict specifications and standards. [74]

With regard to external issues to the protocols and applications, the best practice guidelines recommend that patented technologies should be avoided, as to not hinder

[16] The software complexity must not be underestimated. Internet routers are largely more complex on software than telephony switches, with the added liabilities of software bugs, maintainability and reliability.

[17] IP networks are largely over-provisioned because: the nature of the traffic is asymmetric and bursty, while the data and physical links are symmetric and fixed; traffic growth is difficult to predict; a model of using a 1:1 over-provisioning (below 50% utilization) is typically used, but not as a future-proof extension, but as a wide-spread protection mechanism

[18] About 40% of unplanned outages in IP networks are due to human errors and another 40% due to application errors, leaving only 20% to optimize at the network level [77]

the liberties and freedoms of use. If however better, yet patented technologies exists, or there are concerns regarding legal issue (e.g. export laws), these technologies should still be used while reasonable terms are still available and eventually the technologies will be reproduced even over such legal borders. [74]

Regarding security, the Internet principles recommend that this would be provided by carriers at the network and transport level, yet as this is not mandatory, then always applications should provide their own level of protection for privacy and authenticity. For flexibility, multiple security algorithm choices are to be offered, in a manner which will choose at least one as mandatory, to guarantee basic interoperability, but let the opportunity available for other mechanisms to be used in an inter-changeable way.

While modularity in design is recommended, recent experiences show that in many cases protocol layering and even optimizations are harmful. Accordingly, the OSI model [78] is no longer the guiding principle, but higher level protocols are encouraged to take into account lower layer characteristics and to minimize the number of layers as to prevent unnecessary complexity (see for example [76] - section 3).

Especially with the advent of wireless networks and their associated optimizations, cross-layer design as short-term optimization has been found to be beneficial, as long as they are kept within control and the short-term gains are not negating the long-term ones provided by modularity [79]. Also in the Internet world, several protocols make successful use of by-passing several layers for optimization purposes [80]. Optimizations and also additions of multiple features are also considered harmful in many cases, for the same reasons of violating the simplicity principles.

To conclude, the only constant principle of the Internet remains its driving community force with its willingness to constantly adapt, optimize and evolve its principles and the used technologies, rather than following the typical telecommunication operator approach of large architectural redesigns and adaptations.

The Internet model, as shaped by its standardization, is then practically driven by its community, through a model which requires first practical implementations as trials for new concepts, before a standard is written, after which adoption happens naturally as needed and pushed forward in a peer-like environment.

2.1.3 Unified Control: NGN and IMS

In this section the evolution of telecommunication networks and concepts is presented, from the IN roots to the current NGN all-IP transition. Especially detailed of course here is the IMS architecture, which constitutes also the targeted architecture of the project analysed in this dissertation.

2.1.3.1 NGN: Convergence of IN, GPRS and VoIP in Standards

At the beginning of this century the telecommunication industry stands at technological cross-roads. The current communication networks are now virtually available

everywhere and almost to everyone in the world, which means that they are more and more regarded and accepted as part of the basic human needs. The services offered over these networks however have remained almost the same for decades[19], although in parallel the Internet world has taken huge leaps in terms of introducing new communication concepts and revolutionary services. To remain relevant in the telecommunication business, the current operators will constantly need to evolve their communication networks in order to keep-up with the pace and even improve the services provided to end-users over the Internet. Otherwise they risk being reduced to simple bit-movers for the transparent transport of Internet data.

On the horizon new communication paradigms are already forming on top of the Internet platforms. Initially thought as a communication network for exchanging data between networks of computers, in the recent two decades the Internet grew beyond this initial scope and became *the ubiquitous* platform for communication between people. Through services like e-mail, instant messaging or Twitter, the Internet provides more efficient alternatives for sharing information to the classical postal system, telephony or fax. Evolutions in networking technologies have improved the capabilities of the IP networks such that today the Internet is virtually capable of carrying all the telecommunication needs of legacy networks, while also providing services like voice, television or other forms of entertainment. Besides the Big-"I"-Internet[20], the same set of IP technologies is successfully used in various other networks with stringent requirement for security, making the IP a de-facto standard in all of today's telecommunications.

Another side of the Internet evolution is that the services no longer require a long time interval between the need[21] and availability. The presence of established underlying communication layers coupled with the freedom of choice, low-costs and equality of service providers has made the Internet a booming environment for the creation of new services. The creation and introduction of new communication paradigms has been practically decoupled from the physical requirements, such that new services are currently researched and developed based on future expectations and not just on direct paying customer requests. This of course can be a dangerous environment if not properly understood, as the "dot-com" bubble exposed, yet the technical results are superior to those offered through legacy service providers and operators.

Considering the clear benefits of the Internet services, each current TSPs is faced with two options: either recognize the Internet as the single future platform for communication and become so-called bit-movers for supporting its basic need of abstract data exchange; or align their networks into Internet-like ones and start offering Internet-type of services. Of course, the reverse alternative for the operators

[19] Most prominent service remains telephony.

[20] The word Internet is spelled with both capital "I" and lower-case "i". The capitalization is used to indicate that the term refers to the current TCP/IP network globally interconnected and publicly available. The lower-case term is used to refer to a network using the same technologies, but not necessarily connected to the global network, or abiding by its regulations.

[21] As in customer demand.

to introduce a plain Internet competitor has little value as only insignificant niche customers would actually benefit from moving away from the current Internet and its content to a probably disconnected environment. But reusing, extending and evolving the current Internet platform through telecommunication industry specific advantages like reliability, five-nines[22] availability, security, quality of service, billing and so on is of real value.

The chosen name for this evolutionary merge of the legacy telecommunication network, represented by the IN set of technologies towards, the Internet Protocols and Services is referred to as the migration to the Next Generation Networks. This is however a more or less abstract concept, as there are multiple architectural options and feature-set versions which qualify.

More formally, the NGN network working description has been formalized by ITU-T as follows [19]:

> *"A Next Generation Network (NGN) is a packet-based network able to provide services including Telecommunication Services and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent from underlying transport-related technologies. It offers unrestricted access by users to different service providers. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users.*
>
> *The NGN is characterized by the following fundamental aspects:*
>
> - *Packet-based transfer*
> - *Separation of control functions among bearer capabilities, call/session, and application/service*
> - *Decoupling of service provision from network, and provision of open interfaces*
> - *Support for a wide range of services, applications and mechanisms based on service building blocks (including real-time/streaming/non-real-time services and multimedia)*
> - *Broadband capabilities with end-to-end QoS and transparency*
> - *Interworking with legacy networks via open interfaces*
> - *Generalized mobility*
> - *Unrestricted access by users to different service providers*
> - *A variety of identification schemes which can be resolved to IP addresses for the purposes of routing in IP networks*
> - *Unified service characteristics for the same service as perceived by the user*
> - *Converged services between Fixed/Mobile*

[22] Refers to a service availability of 99.999%, as in below 5 minutes of service disruption per year

- *Independence of service-related functions from underlying transport technologies*
- *Compliant with all Regulatory requirements, for example concerning emergency communications and security/privacy, etc."*

The term was formally defined in the current form by ITU-T in 2004, yet it seems to have appeared many years before that in popular language to describe the evolution towards all-IP architectures. The first concepts called for the use of IP and Multiprotocol Label Switching (MPLS) for transport and H.323 [48] for signaling, later on replaced with SIP [49].

2.1.3.2 The 3GPP and TISPAN NGN Solutions

The IMS architecture has been specifically designed and realized as an NGN. Also as this architecture was embraced and extended by multiple standard organizations besides the starting point in 3GPP, it is now regarded as the reference architecture for NGN.

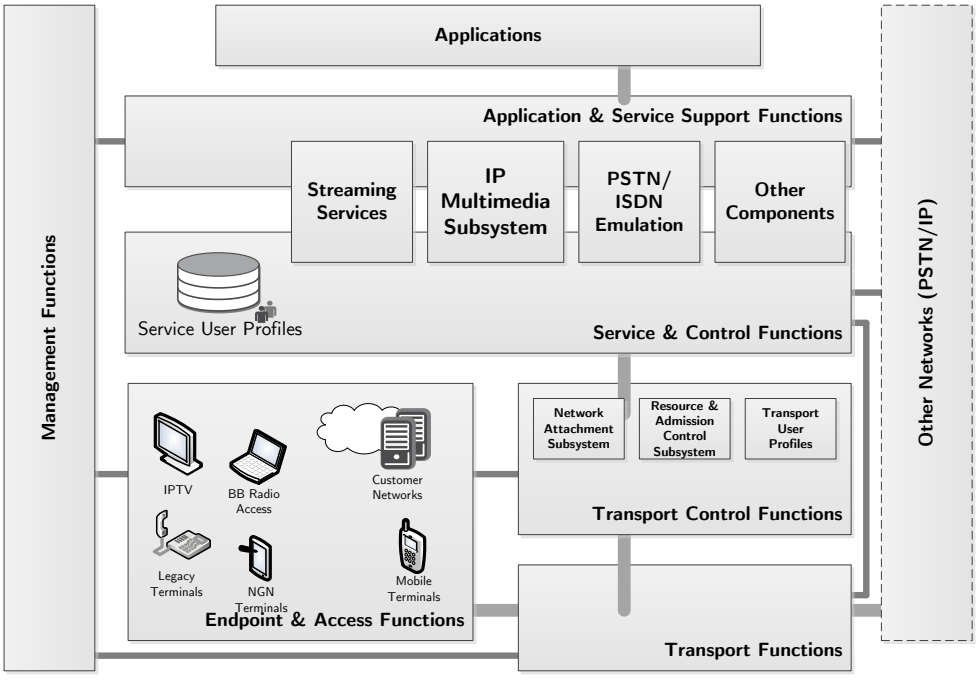


Figure 2.15: The 3GPP and TISPAN NGN Architecture [10]

More extensive formalization for the NGN comes from TISPAN, in the form of

their NGN Releases [10]. NGN Release 1 was finalized in December of 2005 and provided the "robust and open standards that industry required for the development, testing and implementation of the first generation of NGN systems" [10][23]. The architecture adopted the previously 3GPP standard IMS architecture for SIP-based applications, but also included further subsystem for non-SIP applications. A big part of the work scope has been towards harmonizing the wireless and wire-line networks, result which has actually been transmitted back to 3GPP in early 2008 for standardization with all other involved bodies into a "Common-IMS" platform.

NGN Release 2 was finalized at the beginning of 2008 and was concentrating on adding additional components like IMS and non-IMS based IP Television (IPTV), Home Networks and devices, as well as NGN interconnect with Corporate Networks.

Currently TISPAN is working on Release 3 which targets IPTV enhancements, IP network interconnection, NGN security enhancements, QoS with overload control and other NGN requirements.

2.1.3.3 IMS, EPC and NGN Solutions

Many efforts have been spent on defining an instantiation of the NGN concept and so far the most recognized and accepted model is that of the IP-Multimedia Subsystem (IMS). Initially started as an evolution of the UMTS standards in 3GPP standardization body in an attempt to replace the legacy CS-based telephony with a PS-only all-IP alternative, the core concepts have quickly been adopted by many other standardization bodies and extended to many other types of communications and services. Although not a complete solution to the NGN set of requirements, to date, IMS is the most recognized path for migration of the current telephony services to the NGN, as the most non-abstract and advanced solution that currently exists.

Another concept that recently arises from the specific domain of the NGNs is that of the Evolved Packet Core (EPC). This constitutes itself as complement for IMS, but can also be considered as standalone architecture, providing an IP-connectivity layer between various radio access technologies and generic IP services. This architecture on one side covers many of the short-comings in the IMS interactions with the access stratum and can be used as such in composition with IMS, but also comes in as a lighter CN architecture allowing direct OTT pure IP applications to be improved by the operator with specific advantages in the domains of QoS, mobility or security. EPC will be presented after the IMS overview, in Section 2.1.3.6.

The NGN architectures bring with them an extensive set of communication protocols, as briefly summarized in Figure 2.17. For the purposes of this dissertation, the most relevant ones will be highlighted further on in Section 2.1.4.

[23] In retrospect one has to observe that NGN Release 1, as often is the case with many first versions of 3GPP architectures, was more a flag-in-the-ground than a true upgrade of equipment in exploitations. Subsequent versions have matured and completed the requirements, allowing for true products to appear.

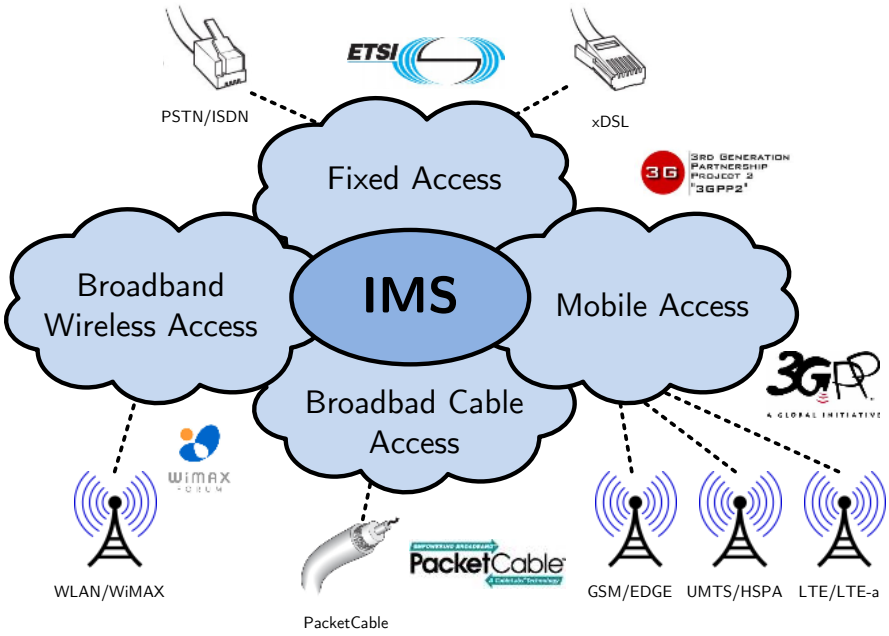


Figure 2.16: IMS as a Common Standard for NGN Architectures [11]

2.1.3.4 The IP-Multimedia Subsystem (IMS)

The current networking environment presents us with a multitude of services which are each based on different technologies and have completely different implementations. For example, the SMS service, although similar in concepts to MMS or Instant Messaging, has a completely different technical realization. Yet the users would like to combine their services in more convenient ways. From the user's perspective many of these services cover the same communication needs, yet technically they are hardly even inter-operable (without additional translation gateways), which in turn seriously segments the user communities and limits the combinatory usage.

Then there are a multitude of Access Network (AN) solutions, each different in authentication methods, security and provided services. Each radio network evolution implies a long series of steps before reasonable service is attained and even so, not all services can be properly provided or take into account each individual AN capabilities. This is highly uncomfortable for the user as one has to continuously manually switch and adapt between these different networks as one sees fit based on network coverage, bandwidth, costs, security, quality and so on. Instead, the users prefer to use ubiquitous services which work always, from anywhere, on any device and in the same manner.

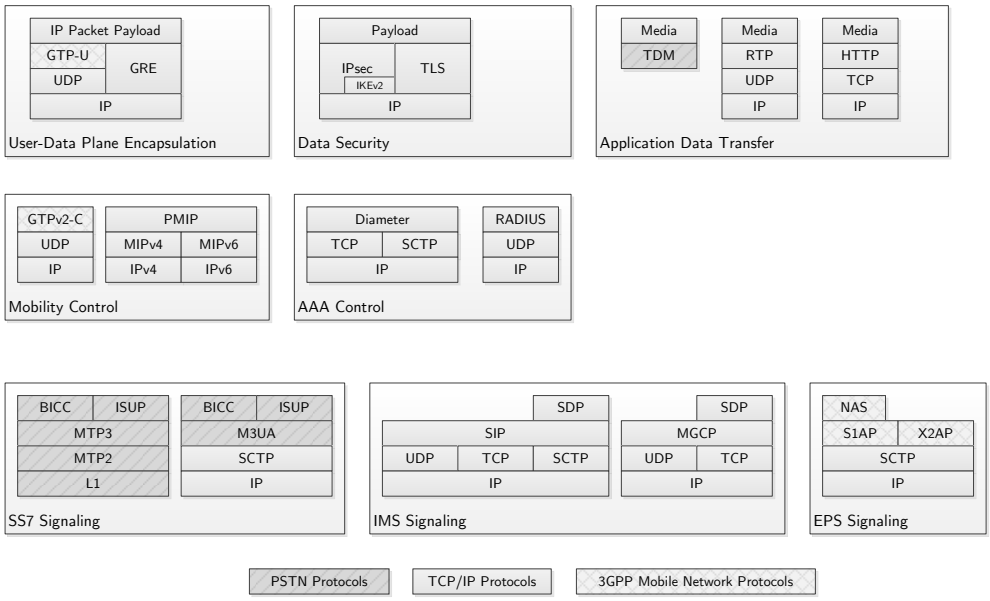


Figure 2.17: A Taxonomy of the NGN Protocol Stacks

Introducing a new service in the current telecommunication industry requires a complete vertical integration. This is because the currently deployed architectures are not flexible enough nor do they have the required interfaces to allow the integration of new Internet services at the pace that the market demands them. Also introducing new access technologies is even harder to perform as each existing service must be extended in this vertical silo mode.

The initial concepts of IMS, as initiated by 3GPP in Release 5 and then further refined, were based on a 3-layer architecture targeted at creating a horizontal model for adding new access networks and future services with minimum re-investments.

At its core, switching nodes offer signaling routing functionality as well as a series of universally common services like authentication, security, location, session set-ups, message exchanges and so on. Below this layer, a multitude of AN technologies can be employed. Each access solution will provide its specific capabilities to the User Equipment and will interface with the CN through standard interfaces, such that the introduction of future radio advances will not require major redesigns above this layer.

To the north, the core layer exports flexible interfaces for triggering powerful services by filtering and routing signaling towards generic Application Servers. Again these interfaces have been designed as extensible as possible, in order to allow effective definition and introduction of future services.

Having such a flexible architecture is a major advantage towards reducing the

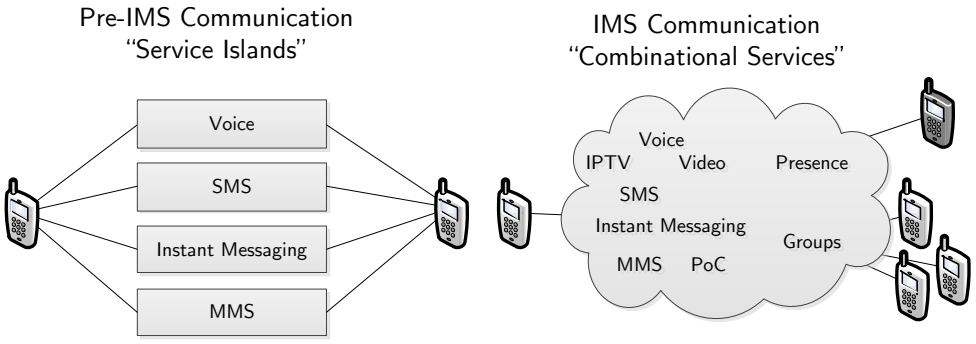


Figure 2.18: "Service Islands" vs. "Combinational Services"

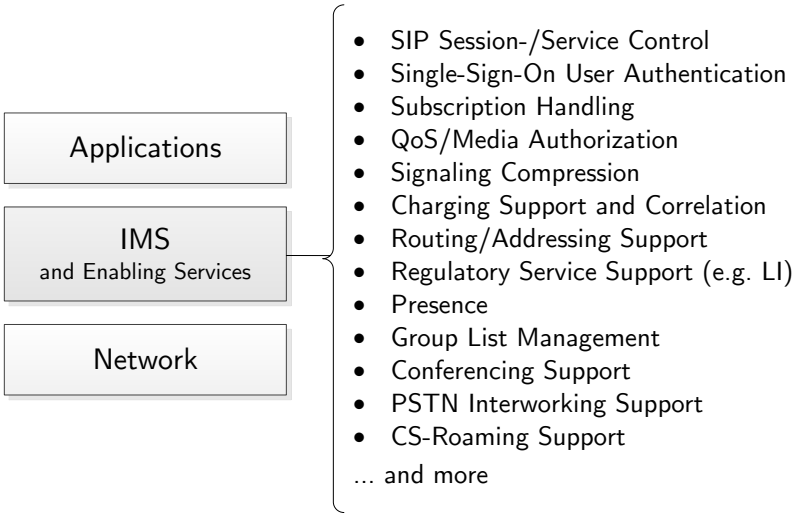


Figure 2.19: The IMS 3-Layers Architecture

CAPEX linked with the introduction of any new service or access technology, while also providing benefits on the OPEX through extensive reuse of openly standardized core components.

Initially IMS has been received with a lot of skepticism [81]. First of all, it came together with too many buzzwords and unrealistic promises, while still being only partially standardized, with many functional, security and quality issues. This and

the entirely new architecture caused an initial rejection in the telecommunication business and resulted in it being regarded as many years away from reality. Then also the ISPs initially rejected its model, as it was perceived with worries for a walled-garden architecture, which could seriously affect the network neutrality concepts in the view of their predominant Internet services users.

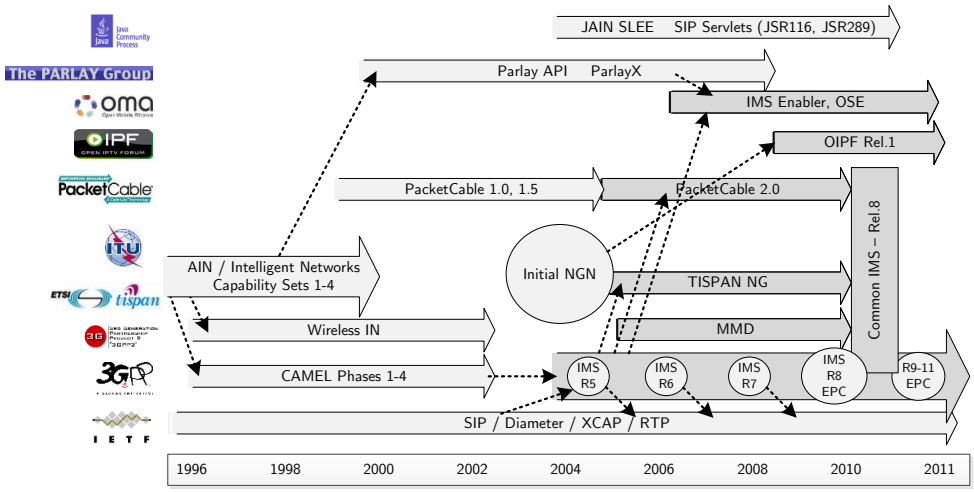


Figure 2.20: The IMS Standardization History

Although adopted by most of the standardization bodies and organizations, the IMS architecture has grown in maturity of standards, but has had so far a slow start and emergence into the currently deployed networks. On one side the architecture brings a radical change and as such it is normal to have a slow introduction in order to preserve a high Quality of Experience (QoE) for the users. And unfortunately much has been waged on the invention of a "killer application" to sparkle the initial adoption, as the World-Wide-Web did for the Internet. This is still to materialize. It was by no means that there was a lack of demand or a market saturation, as in parallel the Internet has just went through the Web 2.0 revolution. The high costs and partially walled-garden architecture has made IMS somehow unattractive to the Internet innovators, which still do not have enough incentives to embrace the IMS architectures as a base for their services.

On the positive sides though, the IMS managed to merge the previously largely digressing fixed and mobile core networks. The FMC concepts established IMS as common CN architecture between these networks by practically adopting the same architecture in the specific standardization bodies[24]. And even further, as of 2013,

[24]3GPP and 3rd Generation Partnership Project 2 (3GPP2) for the mobile domain, TISPAN for the fixed Digital Subscriber Line (DSL) domain, Cable Television Laboratories, Inc. (CableLabs)

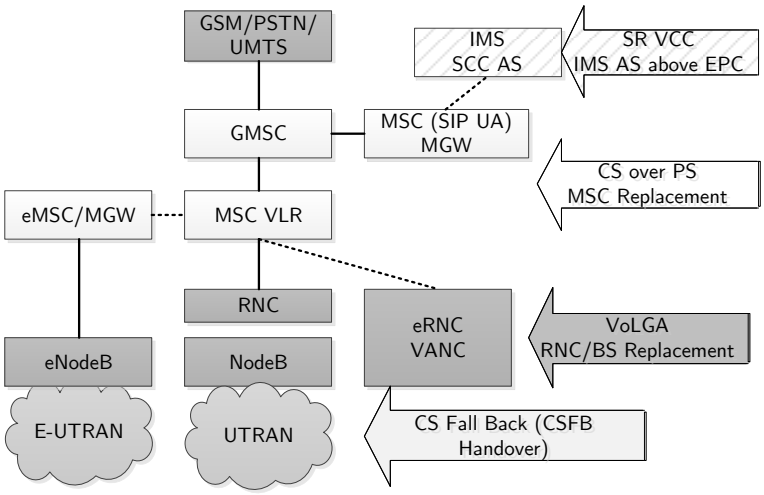


Figure 2.21: VoLTE Options

IMS is set to get back in the spotlight, as a solid solution for VoLTE. As depicted in Figure 2.21, while multiple alternatives to replace the CS-voice service in the LTE PS-only environment have been explored, IMS is in a winning position.

From a functional perspective, the IMS Layer, as presented in Figure 2.22, is composed of a set of components communicating through standard interfaces. For reference, a complete list of IMS relevant functions and interfaces, compiled from various 3GPP standards, is presented in Figure B.2. While an exhaustive presentation is beyond the scopes here, a short overview of the most relevant functional elements in scope here is presented in the next section. A more detailed analysis of the main procedures and mechanism is deferred then to the specification section in Chapter 5. In standards, the most important information source is the [20] standard, which describes the main IMS architecture.

2.1.3.5 IMS CN Elements

From Figure 2.22, the Service/Application Layer and the Transport Layer are out of scope in the IMS context, as only the IMS Layer (also known as the Session Control Layer) itself is being approached in the further presented Open Source IMS Core (OpenIMSCore) project, as the scope is limited to CN. The main components in this IMS Layer are the Call Session Control Functions (CSCFs) and the HSS.

It must be also mentioned that these are merely standardized functions which in reality could be implemented in various parts or combined logical components, as required and found suitable by implementation and exploitation requirements. The

for the fixed Community Access Television / Cable TV (CATV) domain

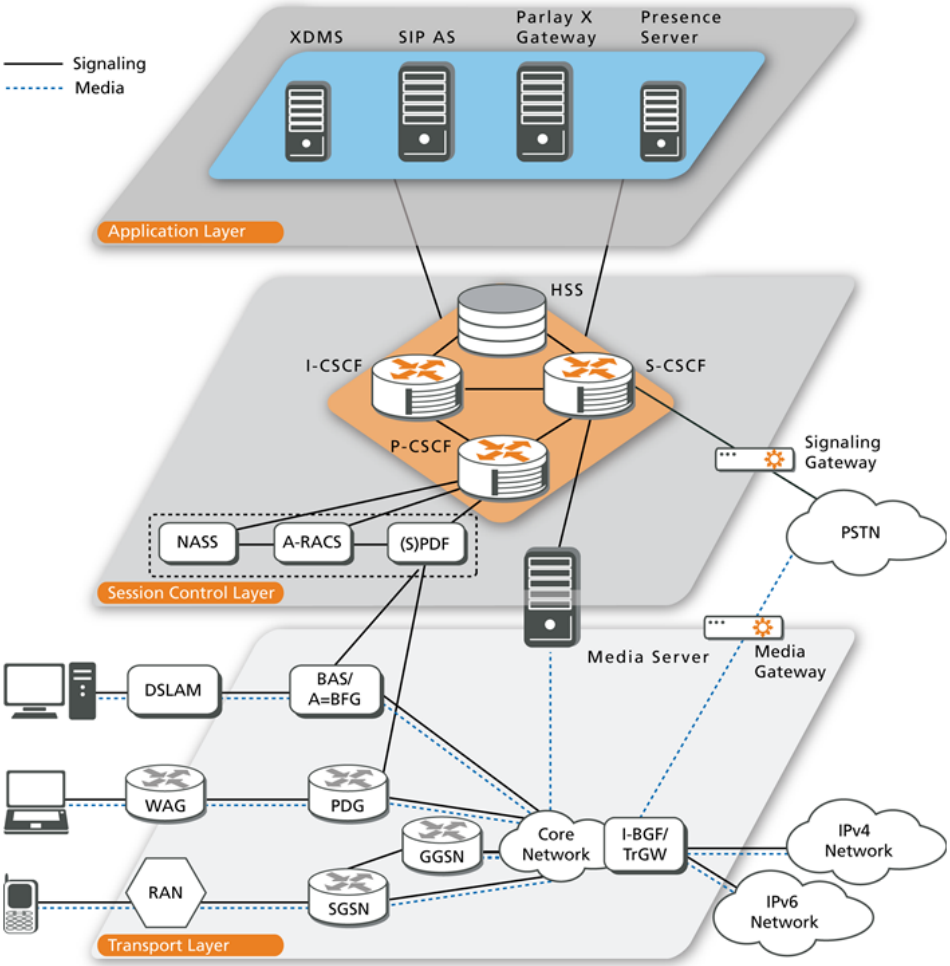


Figure 2.22: IMS Functions

important thing to keep in mind is that these functions are communicating through standard interfaces[25], which should be always exposed for the sake of interoperability, and as such they are named "reference points".

The main signaling protocol in IMS is SIP. Additionally, to help with AAA operations inside secure domains, Diameter is employed.

Proxy CSCF (P-CSCF)

The Proxy CSCF (P-CSCF), depicted in Figure 2.23 represents the entry point, on the User-to-Network Interface (UNI) interface, to the IMS domain. Through the Gm interface IMS-UEs connect to the CN. In essence it has the role of a security gateway for signaling, implementing a specialized and standard defined Session Border Controller (SBC) for protecting both the CN from rogue UEs as well

[25] In 3GPP parlance, standard interfaces are referred to as "reference points"

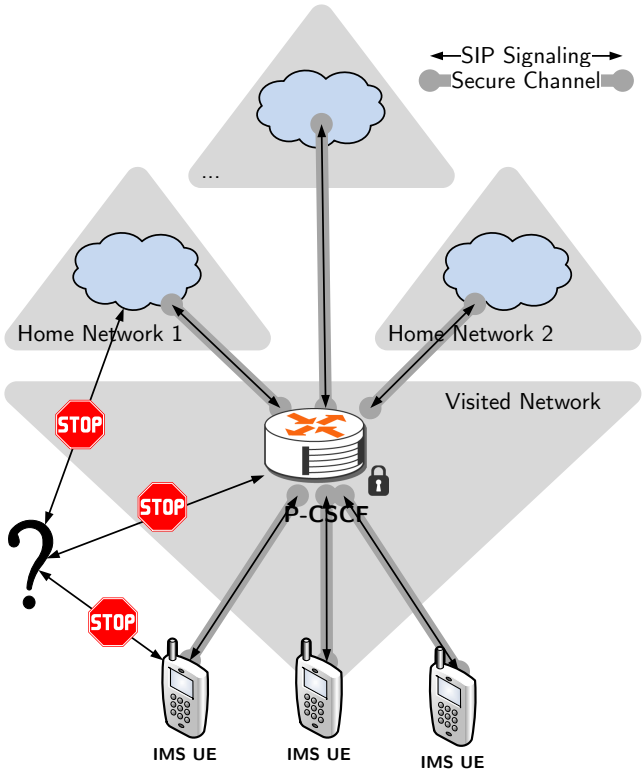


Figure 2.23: IMS Proxy CSCF (P-CSCF)

as the UE from malicious signaling.

As to ensure its effectiveness, the client side elements should only accepts signaling from the P-CSCF and otherwise reject anything else. The Gm interface is secured during the initial registration and authentication procedures with either IPsec or TLS. During these procedures and especially during the network-based authentication ones, the P-CSCF plays a critical role in ensuring security of the entire IMS CN, as it is also interconnected with the rest of the IMS functional elements.

This interconnection is based on a chain-of-trust, established between various operators and security domains. The P-CSCF, as the end-point of the secure Gm communication channel, vouches for the UE identity and security. To do this it follows on the registration procedures first and saves locally the needed information in a reversed[26] registrar structure. On regular signaling then the P-CSCF checks all the inbound messages from clients for correctness, both on used identities, as well as on various message routing relevant information. On incorrect signaling, based

[26] A regular registrar, as found for example on the S-CSCF would associate from Address of Record (AoR) towards Contact addresses. On the P-CSCF the mapping is required in the reversed order from an indexing perspective, as the client presents the Contact information and the P-CSCF must check its correctness.

on operator policy, it either rejects signaling or fixes it before forwarding towards the secure domain.

In the generalized case the P-CSCF is situated in a visited network, representing a roaming situation for the UE. This information is added to the signaling, such that home networks can verify and enforce roaming policies. The connection on the Mw interface between the P-CSCF and the CSCFs in the home domains is also to be secured, yet the security there is not per-UE as on Gm, but on the entire connection and is to be realized with standard IP security mechanisms (e.g. IPsec).

The P-CSCF, due to its position close to the Radio Access Network (RAN), has the role of an authorization gateway. It derives from SIP and Session Description Protocol (SDP) signaling for gating, QoS and charging and sends them towards a Policy and Charging Rules Function (PCRF) element, in charge of processing these requests based on the operator policies and rules, pushing them towards the connectivity nodes and eventually ensuring their enforcement. The communication channel is bidirectional, allowing the P-CSCF to react to events from the connectivity layer (e.g. loss of bearer, Radio Access Technology (RAT) change and so on).

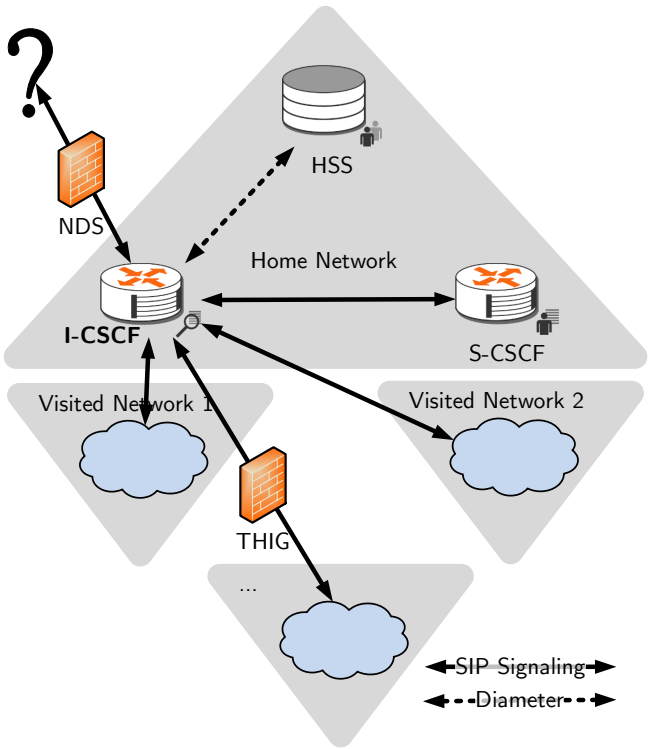


Figure 2.24: IMS Interrogating CSCF (I-CSCF)

Interrogating CSCF (I-CSCF)

The I-CSCF, depicted in Figure 2.24 has the role of a directory, as an entry point within a domain. It implements the function of a load-balancer, by allowing the subsequent processing elements to scale naturally through segmentation of the processing context in sets of UE-associated elements, served on multiple processing nodes.

To achieve its directory-like functionality, the I-CSCF queries the domain-global databases in the HSS, over the Cx interface, for where signaling of the respective UE is currently being served, or for required and optional capabilities for the respective UE in order to make a decision on routing new signaling sessions. Advanced load-balancing features can be built within this element.

Historically the I-CSCF was also regarded as the first functional element on the entry path inside a security domain. As to protect the home network from attacks and information leakage, NDS and Topology Hiding Internetwork Gateway (THIG) features were specified as to be applied on all inbound and outbound signaling. In the recent standardization releases, as the complexity grew and also while the same security had to be applied on the visited networks side, this has been extracted as a separate functional element, the Interconnection Border Control Function (IBCF).

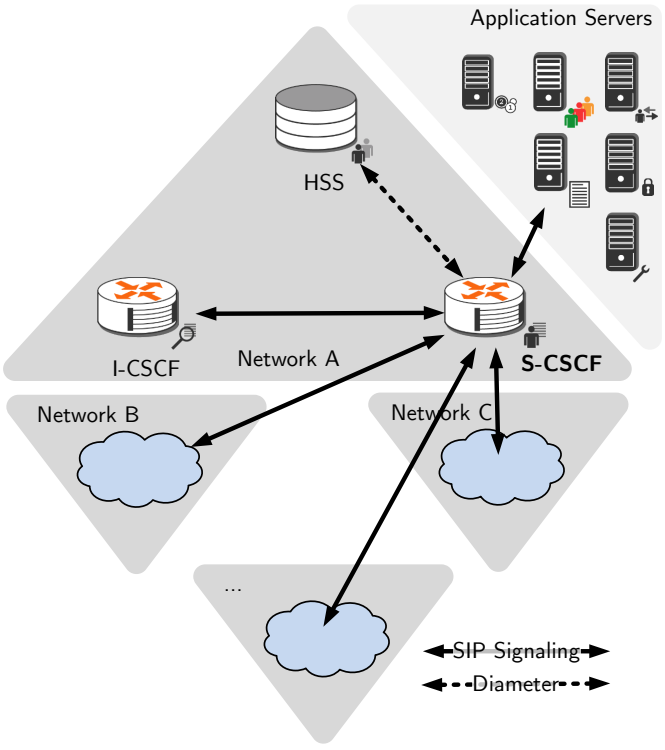


Figure 2.25: IMS Serving CSCF (S-CSCF)

Serving CSCF (S-CSCF)

The S-CSCF, depicted in Figure 2.25, is the workhorse element of the IMS processing architecture. Besides the registrar role and its involvement during authentication, it functions as the main CN signaling routing function. Beside the standard functional elements which have to be triggered while processing signaling (e.g. Media Gateway Control Function (MGCF), Breakout Gateway Control Function (BGCF), Media Processing Function Controller (MRFC) and so on), the S-CSCF also enables one of the corner-stone features of the IMS architecture, the capability to trigger services on a subscriber-by-subscriber customized basis.

Service Profiles are downloaded from the HSS during the first signaling procedures of a client device, over the Cx interface. The S-CSCF also associates itself in the HSS global database with the processing context of the user. Upon subsequent signaling, the S-CSCF filters and verifies for Trigger Point matches. In case of successful match the signaling is routed through the AS associated with the respective service. Multiple services can be triggered sequentially for a single message.

The S-CSCF is also in charge of managing for each user its dialog sessions and its registration status, such that notifications and actions can be taken on user generated as well as administrative events as to keep the context in order.

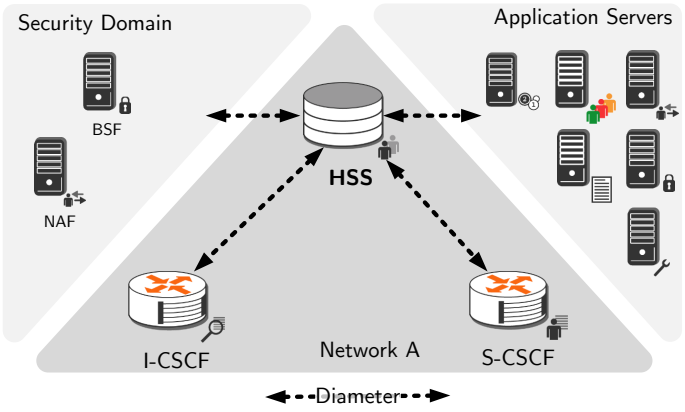


Figure 2.26: IMS Home Subscriber Server (HSS)

Home Subscriber Server (HSS)

The HSS, depicted in Figure 2.26 is in effect the Home Location Register (HLR) evolution, supporting IP services. It provides database-like facilities, for storing and interrogating subscriber and service profiles, generic services information, authentication information, global context location information and more. The I-CSCF uses its facilities by employing the Cx Diameter interface to query the database. The S-CSCF updates this information and downloads Service Profiles. Services employ the Sh interface to access the same information as well as to enable them

with generic database functionality, consolidating as such the information storage facilities in the CN.

Usually realized with back-end Database Management Systems (DBMSs), the HSS would feature a customized request processing engine on each of its interfaces. There the access rule as well as logical operations are implemented, such that the CN and service access to the raw data is properly policed. All such interfaces use Diameter as the communication protocol.

2.1.3.6 The Evolved Packet Core (EPC)

From 3GPP Release 8, as part of the System Architecture Evolution (SAE) [21] concepts, another lighter CN architecture has emerged as an evolution of the IMS Transport Layer[27]. The EPC is not a replacement for the IMS CN architecture, but more of a complement and also a lighter alternative.

The basic service concepts have been relaxed in the sense that from the perspective of the transported application, the hard-requirement for SIP as the signaling protocol has been lifted. The EPC architecture assumes the service to be a generic IP application. Consequently, the Internet domain services can be directly used on the EPC architecture.

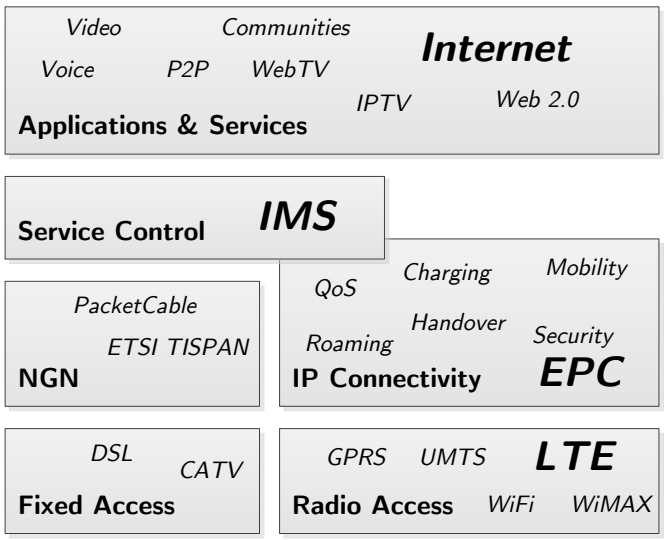


Figure 2.27: EPC Positioning in the Telecommunication World

Of course, IMS is using the EPC as a well-defined Connectivity architecture. Standard interfaces are in place to provide IMS services with the full set of EPC features and advantages.

[27] See also Figure 2.20 for where and when the EPC architecture appeared

EPC then in itself, as depicted in Figure 2.27, represents an IP Connectivity Layer between the Radio Access and the Service Control/Application & Services strata. Its value is in providing a functionality set including for example QoS, Mobility, Handovers, Always Best Connected (ABC) [82], Security and Charging in an uniform manner over various access technologies.

To take these features in order:

- QoS is provided by managing various RATs such that the requirements are enforced on the resource-limited radio spectrum, while also events and updates are being pushed back up towards the applications.

- Mobility and Handovers are provided by orchestrating the various ANs and maintaining various tunnels with specific Access Network Gateways (ANGws) in a seamless manner, which would hide the complexity of individual intra- and inter-system handovers from the application layer.

- ABC is achieved through the introduction of the Access Network Discovery and Selection Function (ANDSF) [83], which acts as an Access Networks Coverage Map and Inter-System Mobility Policies repository and is in direct contact with the UE. Through this, the operator can push handover policies which would implement both an ABC concept and a good AN selection and load balancing.

- The Security aspect is enabled by using consistently strong authentication and encryption methods and algorithms, in all the ANs. Even when using non-3GPP technologies like WiFi, strong authentication as Extensible Authentication Protocol (EAP)-AKA, IPsec encryption and integrity protection is specified and available for use.

- Charging is provided by linking with the Applications and Services on a common mechanism with QoS and then connecting the EPC Gateways, capable of reporting traffic metering values, to Online and Offline Charging Systems.

The EPC features described are realized by enabling bi-directional information exchanges between the IP Connectivity Layer and the Radio Access Layer, in a tight integration and with well-defined mechanisms for each individual AN technology. It is by design foreseen that, as the Radio Access will evolve and new radio technologies will be introduced, the integration efforts will be limited to the IP Connectivity level.

Then to enable the use of the described features, a bi-directional information exchange path is opened between the IP Connectivity Layer and the Application & Services Layers. The Applications can push QoS and Charging Rules describing their IP communication needs. The EPC will respond with actually committed rules, adapted to the current AN situation and the respective Subscriber Profiles. Additionally, on events and changes, these are also pushed up towards the Applications, so that the services can immediately react and adapt to the updated network situation.

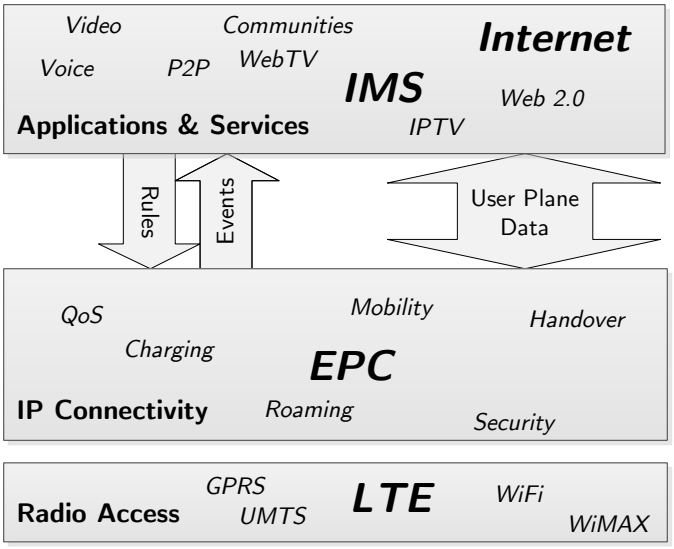


Figure 2.28: Interaction in the EPC Architecture

From a functional perspective, several sub-systems can be distinguished in the EPC architecture:

- A Core Network Mobility group of functions is in charge of enabling the Mobility and Handover functionality.
- A Policy and Charging Control, evolved from the concepts just started with IMS, integrates with the Core Network Mobility to provide QoS and Charging.
- Additionally the same HSS as in IMS is found here, only with different interfaces
- and the ABC enabler, the ANDSF.

2.1.4 SIP and Diameter as Foundations

Before diving into the next chapters, it is necessary at this point to briefly introduce the main protocols used in the CN of NGNs.

With the migration to IP technologies, to replace the SS7 [32] protocol for signaling, SIP has been chosen as the main signaling protocol in IMS. As it will be presented next, SIP has the right characteristics in order to ensure a long-term flexibility and extensibility of future CNs. Diameter [84] is then also used largely for the internal network interfaces and will also be presented below. Many other protocols are also introduced in the new architectures, yet they will not all be approached as they play a relatively minor role in the targeted IMS CN implementation.

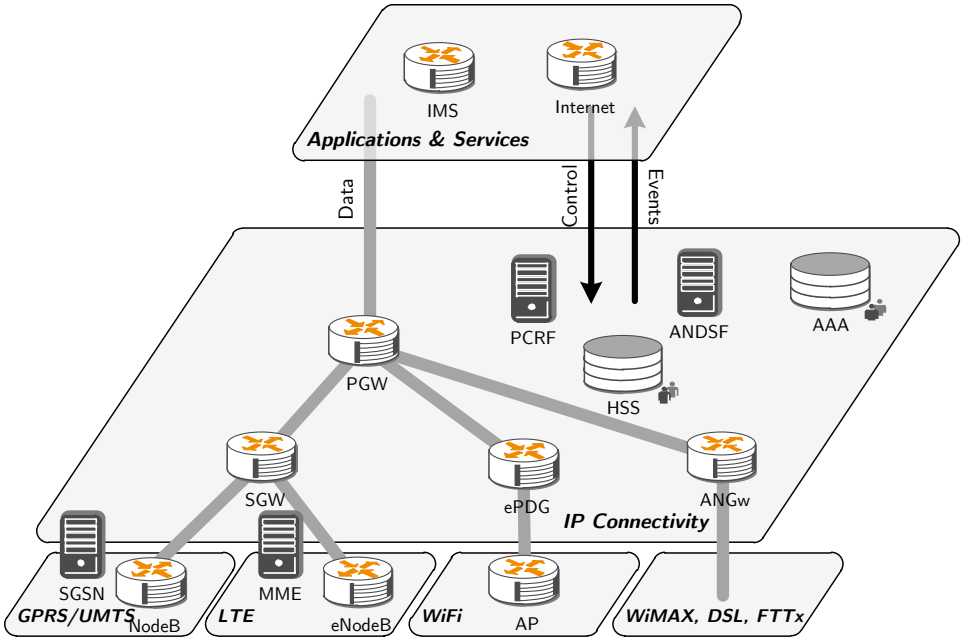


Figure 2.29: The EPC Functional Components

It has to be noted that for most of the protocols[28], 3GPP and ETSI have opted to either use existing IETF protocols or to extend and complement those protocol specifications, by contributing back to IETF. From this perspective, it can be observed that 3GPP, ETSI, 3GPP2, CableLabs and other telecommunications domain standardization bodies are currently limiting themselves to producing standards on how to use the IP protocols, but the protocol specification itself is left under the supervision of the Internet standardization bodies (i.e. IETF).

2.1.4.1 The Session Initiation Protocol (SIP)

SIP [49] started as a concept around 1995 in the IETF Multiparty Multimedia Session Control Working Group (mmusic) [85]. The initial purpose was to provide a signaling protocol for establishing multimedia session, which more precisely would materialize first as an alternative protocol for VoIP to ITU-T's H.323 [48]. Unlike H.323, which was based on traditional telephone recommendations and concepts, SIP was, since its beginning, strongly anchored in the Internet way of thinking. This meant that it was favored by the "dot-coms" and eventually managed to successfully and completely replace H.323.

[28]With notable exceptions only in the area of 3GPP RAN, which still remain highly specific for the respective 3GPP technologies.

It is important to note that SIP is an end-to-end, client-server session signaling protocol. It is not a transport protocol[29], QoS reservation protocol[30] or a Gateway Control protocol[31]. SIP is meant to provide services as presence, mobility, session set-up/management/tear-down, state changes and so on.

Historically, the first version of the protocol [88] appeared in March of 1999. Later that year, a separate IETF Working Group (WG) has been formed for SIP and worked continued. A much more refined version, which is in use today, appeared in June of 2002 [49][32].

Currently the work continues into multiple groups, with each concentrating on specific topics and providing additions and extensions to the main protocol specification. Due to its built-in extensibility, these additions are provided as separate specifications, which do not change the core functionality of the protocol.

IETF Working Group	Topics
mmusic	Origin of SIP, SDP extension, Session Description and Capability Negotiation (SDPng)
sip	SIP core specification maintenance, protocol extensions
sipping	Requirements for SIP, specific SIP application services
simple	Instant messaging and presence extensions
p2psip	Peer-to-Peer SIP
iptel	IP Telephony
enum	Telephone Number Mapping
sigtran	Signaling Transport

Table 2.1: A Short List of IETF Working Groups Related to SIP

From a message format perspective, SIP is very similar to HTTP [89]. It keeps the same textual format, making the protocol easy to debug and understand without complex protocol dissectors. Also a similar request/response model is kept.

The SIP messages can be logically split in 3 parts:

- **The first line**, which in the case of the request contains a `Method`, a target address commonly referred to as `Request-Uniform Resource Identifier (URI)` and finally the protocol identifier/version. In responses this first line differs slightly as the first element is the protocol identifier/version, next comes a `Status-Code` in a machine-standard numeric format and then a human-readable `Reason-Phrase`.

[29] Even as it is transporting in payloads other protocols like SDP.

[30] QoS reservation is typically negotiated using the Diameter protocol.

[31] Media Gateway Control Protocol (MGCP) [86, 87] is the protocol of choice for Gateway Control.

[32] Comparatively SIP 2.0 specified in RFC3261 [49] has 269 pages as compared to SIP 1.0 specified in RFC2543 [88] within 153 pages.

```
INVITE sip:peter@open-ims.org SIP/2.0
Via: SIP/2.0/UDP 192.168.0.1;branch=z9hG4bK63672474
Route:  <sip:orig@scscf.open-ims.org>
Call-ID: 2043876218@192.168.0.1
CSeq: 3586 INVITE
From: "Dragos Vingarzan" <sip:dragos@open-ims.org>;tag=406B53C
To: "Peter Weik" <sip:peter@open-ims.org>
Subject: sip:dragos@open-ims.org
Content-Type: application/sdp
Content-Length: 187
Contact: "Dragos Vingarzan" <sip:dragos@192.168.0.1;transport=udp>

v=0
o=username 0 0 IN IP4 192.168.0.1
...
```

```
SIP/2.0 200 OK Let's have a chat
Via: SIP/2.0/UDP 192.168.0.1;branch=z9hG4bK63672474
Call-ID: 2043876218@192.168.0.1
CSeq: 3586 INVITE
From: "Dragos Vingarzan" <sip:dragos@open-ims.org>;tag=406B53C
To: "Peter Weik" <sip:peter@open-ims.org>;tag=5F013788
Content-Type: application/sdp
Content-Length: 191
Contact: "Peter Weik" <sip:peter@192.168.0.2:5062;transport=udp>

v=0
o=username 0 0 IN IP4 192.168.0.2
...
```

Figure 2.30: SIP Message Example

- **A series of message headers**, in the format of header name and values. Some headers can have their values grouped into one, with values separated by commas. The order of header appearance is not relevant between headers of different types, but in most situation relevant for headers of the same type or multiple values grouped in single headers.
- **The message body**, where optional message payloads can be included. The most commonly payload is SDP [90, 91, 92] and multiple bodies (multi-part) can be packed within a single message.

The SIP `Method` identifies the type of the operation requested. Table 2.2 lists

SIP Method	Purpose	Defined in
`REGISTER`	allows Clients to register their current location (one or more addresses)	[49]
`INVITE`	used to establish a call	[49]
`ACK`	is sent by a client to confirm that it has received a final response from a server, such as `200 OK`	[49]
`BYE`	sent either by the calling agent or by the caller agent to close a call (in dialog context with `INVITE`)	[49]
`CANCEL`	sent either by the calling agent or by the caller agent to abort a call establishment or other SIP transaction	[49]
`OPTIONS`	allows clients to learn a servers capabilities; the server will send back a list of the methods it supports	[49]
`SUBSCRIBE`	starts or stops session or user supervision (event monitoring)	[93]
`NOTIFY`	informs subscribed entity about occurred events (in dialog context with `SUBSCRIBE`)	[93]
`PUBLISH`	enables an entity to push event information	[94]
`MESSAGE`	allows to send an instant message	[95]
`REFER`	informs a recipient to contact a dedicated SIP user	[96]
`PRACK`	acknowledgement of provisional responses	[97]
`UPDATE`	change of media (SDP) during session setup (in dialog context with `INVITE`)	[98]
`INFO`	exchange of application layer information	[99]

Table 2.2: A List of Common SIP Methods

the most used methods today.

The SIP `Status-Code` is standardized in the style of HTTP, with similar 3-digits codes. Although many codes are specified for standardization and inter-operability, there are enough values left for proprietary or new applications. The codes from 100 to 199 are indicating provisional responses, while codes above or equal to 200 indicate final responses. When considering SIP transactions, for each request one User Agent Server (UAS) can send an arbitrary number of provisional responses, as required by the service, followed by a single final response which completes the transaction. The `Status-Codes` are grouped into classes and several examples are presented in Table 2.3.

All requests are supposed to be responded with a final response message, with the exception of the request with the `Method ACK`. This special request is sent in order to acknowledge the receipt of a final response to a `INVITE` request.

From an addressing perspective, the protocol is very flexible accepting a multitude of formats. The most used ones are the SIP-URI [100], the telephone (tel)-URI [101] and the Uniform Resource Name (URN) [102].

Status-Codes	Description	Examples
1xx	Informational – Request received, continuing to process request	100 Trying 180 Ringing 181 Call is Being Forwarded 183 Session Progress
2xx	Success – Action was successfully received, understood and accepted	200 OK
3xx	Redirection – Further action needs to be taken in order to complete the request	300 Multiple Choices 302 Moved Temporarily
4xx	Client Error – Request contains bad syntax or cannot be fulfilled at this server	401 Unauthorized 408 Request Timeout
5xx	Server Error – Server failed to fulfill an apparently valid request	503 Service Unavailable 505 Version Not Supported
6xx	Global Failure – Request is invalid at any server	600 Busy Everywhere 603 Decline

Table 2.3: A List of SIP Status Codes

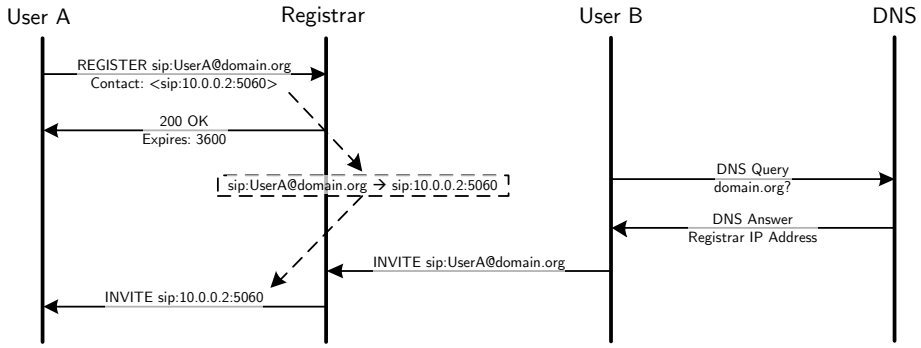


Figure 2.31: Basic Functionality of the SIP Registrar Service

To route requests, SIP uses upfront addresses which contain DNS information and as such are globally routable. Once signaling approaches the targeted endpoint, these are translated to locally routable information, like IP addresses. Related to this addressing scheme, a mechanism is defined to provide mobility for mobile devices: endpoints "register" (at regular intervals or on changes) their current locally

routable information to registrars, which keep lists of associations with regard to the globally routable addresses. The registrar servers are in turn found by performing DNS queries.

Responses are routed based on a stack of `Via` headers, which record the path of the corresponding requests. The path of the response will then follow in reverse the path of the request, with the possible exception of skipping stateless proxies, which have not pushed their addresses in the respective stack.

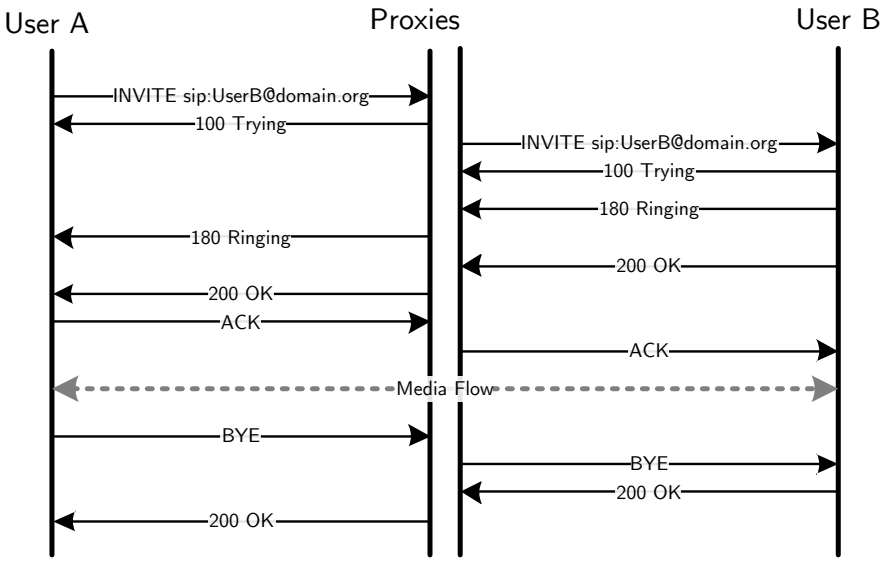


Figure 2.32: Signaling Flows for a SIP Session

To save from the effort of rediscovering a signaling path for requests which would be contextually grouped, a dialog concept is defined. After the initial transaction a stack of `Record-Route` headers are saved and later used as `Route` headers to easily re-route subsequent requests through the same proxy nodes. The mechanism is further enhanced for additionally saving originating [103] and terminating leg [103] routing paths based on the operator's policy.

The SIP specification [49] defines one dialog type, `INVITE`, for the purpose of establishing multimedia sessions. Additionally so far only one other type has been defined, the `SUBSCRIBE/NOTIFY` [93] dialog, which allows for an event based notification system.

From a functional perspective, unlike HTTP, SIP peers can act as both clients and servers in different transactions[33]. Another very important difference is repre-

[33] User Agent Client (UAC) and UAS modes

sented by the optional use of UDP for transport, besides TCP and SCTP [104]. With UDP, a much lower latency can be achieved, alas due to the inherent packet-loss the protocol complexity rises significantly with the introduction of retransmissions and acknowledgements through transaction state machines. Support for IPv4 as well as IPv6 is provided implicitly as the protocol does not have any problematic restrictions or limitations related to the TCP/IP transport layer.

An important feature of the protocol is its extensibility. There are of course many operations defined currently and it is relatively easy to conceive a wide range of services by recombining the existing blocks. However, it has to be noted that the protocol provides for even further extensibility as at almost all levels future extensions are allowed without breaking the existing functionality and also even without requiring expensive software upgrades in existing SIP network architectures. Here are some of the examples of where the protocol can be easily extended:

- New `Methods` can be freely added besides the currently defined ones.
- New `Header` types can be defined and used.
- New `Status-Codes` can be standardized or even used without standardization to indicate proprietary sub-cases in the main classes.
- New `Dialogs` can be defined[34].
- New `URI` types can be used[35].

Over the years and especially with the protocol use as the main signaling method in IMS, SIP has been significantly improved and tested, making it today the reference standard for VoIP.

2.1.4.2 The Diameter Protocol

The concepts of AAA call for a client/server split between systems, where the server stores in databases local user information or gather accounting information. The clients are represented by remote system which need to share that common repository in order to perform user authentication, authorization of resource use as well as accounting for charging purposes.

The history of the AAA protocols can be traced back to the 1950's/1960's first "login" mechanisms. In multi-system / multi-user environments a centralized system for providing user authentication and resource accounting was required. One of the first standardized and formalized protocols dedicated for these purposes was Terminal Access Controller Access Control System (TACACS) [105].

In 1997[36], a critical milestone is achieved with the IETF standardization of Remote Authentication Dial In User Service (RADIUS) [106, 107, 108, 109]. This

[34] Here for state reasons, dialog aware SIP proxies would need to be upgraded
[35] Poorly implemented protocol parsers might break
[36] First concept appearance was as early as 1991

was designed as an extensible protocol, with multiple additional standards adding functionality to it. Using UDP as a transport protocol and a client/server paradigm, it became the de-facto standard for AAA and today is largely used as the network-side protocol to allow edge access devices to authenticate remote users[37]

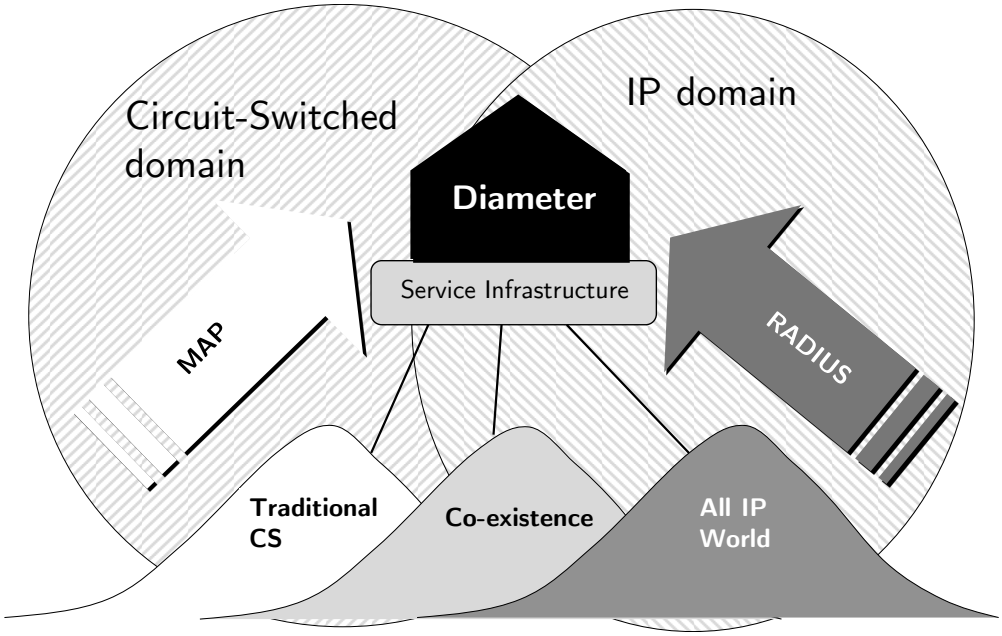


Figure 2.33: Convergence of AAA Protocols Between the CS and IP Domains

As early as 1998, a new working group was formed targeting improvement on the RADIUS protocol. The result was standardized as Diameter[38] [84]. The drafts matured around 2003 and the message format was similar such that backwards compatibility can be achieved through translation agents. The following major changes have been included:

- Use of TCP and SCTP for reliable transport.
- Exchange of capabilities and connection management.
- Allow for larger messages and information elements.
- Allow for vendor specific messages and information elements.

[37] E.g. in WiFi authentication with WiFi Protected Access (WPA)-EAP, PPP tunnel authentications, authorize Packet Data Protocol (PDP) contexts in GPRS networks, authenticate mobile users in CDMA 2000 - first 3G technology deployed (CDMA2000), roaming scenarios, etc.

[38] Even though the name of the protocol is often spelled as DIAMETER, there is no acronym behind it. The name was in fact chosen as a word joke: "Diameter is twice the RADIUS".

- Bi-directional requests, as each peer can act as both client and server.
- Introduction of Agent roles (e.g. proxy, relay, redirect & translation, routing).
- Base protocol state machines for Authorization and Accounting.
- Explicit modular structure for extensibility.

As shown in Figure 2.33, the Diameter protocol was also regarded as a convergence and replacement protocol between the existing mobile operator domain's MAP SS7 protocol and the RADIUS protocol in the IP domain.

Looking deeper at the protocol, the message encoding uses a binary format, in order to minimize message sizes and as such accelerate performance. Unlike 3GPP binary protocols though, the Diameter protocol keeps a slightly more wasteful structure, yet provides for easy extensibility and interoperability as all the information elements have basic standard structures, which would not require costly protocol stack adaptations as future extensions would be added.

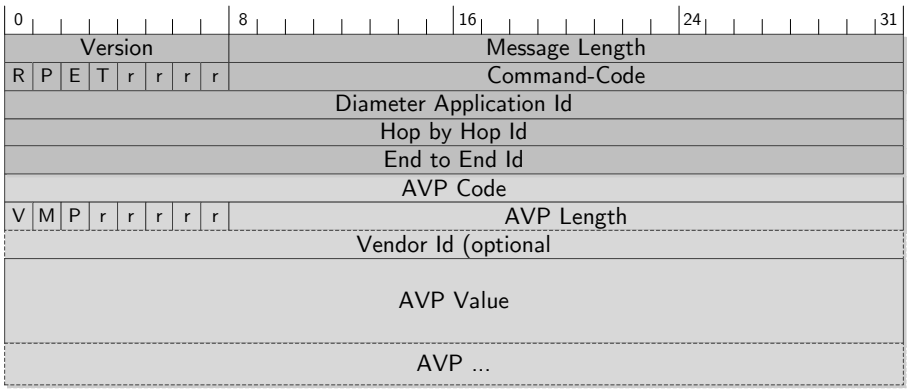


Figure 2.34: Diameter Message Format

Each Diameter message is formed from the following elements:

- `Version` byte; currently only version 1 is defined.
- `Length` indicating the total length of the message [39].
- `Flags` indicating whether the message is a request or reply, message is proxiable, contains a protocol error or it is a potential retransmission.
- `Command-Code` indicating the type of the command request or command response.

[39] Up to 16 Mega Bytes (1,048,576 bytes) (MB) long messages are allowed

- `Application-Id` which defines the domain of the command, such that a reuse of `Command-Codes` is feasible[40].
- `Hop-by-Hop` identifier, which provide unique identification of message for routing purposes.
- `End-to-End` identifier, which provide unique transaction identification.
- `AVPs` which carry the main payloads of the protocol. Each AVP in turn contains the following information:
  - `AVP-Code` used to indicate the type of the AVP[41].
  - `AVP-Flags` which tell if an optional `Vendor-Id` is present, whether the receiving node must understand this AVP in order to process it and whether encryption is needed for end-to-end communication.
  - An optional `Vendor-Id` used to specify the domain for which the `AVP-Code` applies. Similar to the `Command-Code` and the `Application-Id`, this mechanism is used to provide a wider value space for `AVP-Codes`[42].
  - *AVP Value* containing the actual payload data. The format of the data is specific for each AVP.

    The base protocol defines the following basic data formats:

    * `OctetString`
    * `Integer32`
    * `Integer64`
    * `Unsigned32`
    * `Unsigned64`
    * `Float32`
    * `Float64`
    * `Grouped`

    Additionally, the following derived data formats are specified in the same base protocol:

    * `Address`
    * `Time`
    * `UTF8String`
    * `DiameterIdentity`
    * `DiameterURI`

[40] In RADIUS the representation of the `Command-Code` was a real issue as all possible values have been quickly used and nodes had to be specifically configured for individual applications to avoid confusion on value reuse.

[41] This is not the format for the data inside, just an identifier based on which one can retrieve from a standard the actual data format

[42] Vendors can apply for identifier allocation at IANA and then proceed to define their own proprietary AVPs

* Enumerated
* IPFilterRule
* QoSFilterRule

More data formats can be defined by vendors or standardized as required.

An important AVP format must be highlighted: the Grouped AVP. This allows for the inclusion of a list of AVPs into a single one, transforming the simple linear structure of the AVP list in a Diameter message into a more flexible tree-like structure. This allows for contextual use of the AVPs. However it must be noted that unfortunately no special AVP-Flag has been allocated to indicate this AVP value format and as such protocol stacks are not able to fully unravel the entire tree-like structure without knowledge on whether each AVP-Code used corresponds to the Grouped value format or not.

AVPs with the same AVP-Type can appear in a message or a sub-grouped AVP. The order of the AVPs is not specified as being hardly enforced, although the standard does indicate that a proxy or relay-agent should not change this order. In practical terms, many implementers actually took the order as strict and as such it is a good practice to abide by, but not rely on it.

Diameter messages are logically grouped into transactions, each containing exactly one request and one response. As reliability is ensured at the transport level, no special reliability on transactional level is specified and it is left up to individual stack implementations on how to address time-outs in request/response transactional processing.

Responses usually include at least a Result-Code AVP or, in case a vendor or application specific response is required, an Experimental-Result-Code[43] AVP, which indicates the outcome of the request. Similar to the SIP Status-Code, classes of codes are used and a few examples of the values standardized for interoperability are presented in Table 2.4

Although RADIUS still has a very solid footprint in today's real life deployments and as such is very difficult to replace, Diameter is starting to make inroads with the use at the core of NGN architecture like IMS and EPC. For exemplification, a short list of defined applications currently making use of Diameter is presented in Table 2.5.

2.2 Open Source Principles and Basics

2.2.1 Open Source in General

2.2.1.1 Free/Open Source Principles and Software

The term of "Free Software" was probably first defined in February 1986 by the GNU's Not Unix! (GNU) Project [115] in their GNU's Bulletin [116]. The most important aspect from the original definition was [116]:

[43] Includes also a Vendor-Id AVP and is Grouped, allowing for vendor specific result codes

Class	Description	Examples
1xxx	Informational	1001 DIAMETER MULTI ROUND AUTH
2xxx	Success	2001 DIAMETER SUCCESS 2002 DIAMETER LIMITED SUCCESS
3xxx	Protocol Errors	3001 DIAMETER COMMAND UNSUPPORTED 3002 DIAMETER UNABLE TO DELIVER 3003 DIAMETER REALM NOT SERVED 3004 DIAMETER TOO BUSY 3005 DIAMETER LOOP DETECTED
4xxx	Transient Failures	4001 DIAMETER AUTHENTICATION REJECTED
5xxx	Permanent Failures	5001 DIAMETER AVP UNSUPPORTED 5002 DIAMETER UNKNOWN SESSION ID 5003 DIAMETER AUTHORIZATION REJECTED 5004 DIAMETER INVALID AVP VALUE 5005 DIAMETER MISSING AVP 5012 DIAMETER UNABLE TO COMPLY

Table 2.4: A List of Diameter Result-Code AVP Values

Application	Defined in
Mobile IPv4 Application	[110]
NAS Application	[111]
Diameter Credit Control Application	[112]
EAP Application	[113]
SIP Application	[114]

Table 2.5: A List of Diameter Applications

> *"The word «free» in our name does not refer to price; it refers to freedom. First, the freedom to copy a program and redistribute it to your neighbors, so that they can use it as well as you. Second, the freedom to change a program, so that you can control it instead of it controlling you; for this, the source code must be made available to you."*

This was then a manifest, a pledge for developers to participate in projects where their work would be liberated from the norm copyright and distribution constraints of that time. The GNU project's target was not to provide a gratuitous software package, but beyond that to have an unrestricted distribution, modification or usage of that software. To clarify the rights the modern definition is composed of 4 essential freedoms that the program's users have [117]:

- *"The freedom to run the program, for any purpose (freedom 0).*
- *The freedom to study how the program works, and change it to make it do what you wish (freedom 1). Access to the source code is a precondition for this.*

- *The freedom to redistribute copies so you can help your neighbor (freedom 2).*
- *The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this."*

Freedom 0 enforces that no entity can restrict the usage of free software, such that anyone can use it as they see fit. Freedom 1 adds the liberty of modifying the program and imposes unencumbered access to the source code for this purpose. This ensures that all users can make the program perform for their own purposes, unrestricted. Also it adds an important quality, often overseen, that of the liberty to analyze the source code to learn from it and also to ensure that it has not been tainted against your interests. Freedom 2 adds the distribution right unrestricted by any intellectual property copyrights. Freedom 3 adds the right to distribute also the program with your modifications, as such enabling the free program's evolution as well as the community cooperation models.

This definition has later been adapted and evolved by Bruce Perens in the Debian Free Software Guidelines [118]. This then became "The Open Source Definition" [119], where the term "free" was replaced with "open-source".

This difference between free and open is both trivial and complex. On one side, one must understand that the term "free" does not mean "without cost", but it rather has the meaning for freedom. The common stance is that programs might not be free like in "free beer", but more like in "freedom of speech", the advantage being more for a matter of liberty than price.

In reality, even though the software is free from a cost perspective[44], one must not make the confusion that using it would not involve any costs. The price would be in most cases similar to that of using commercial programs, minus the cost of the license. The word "free" is not used as in "free lunch". Free software might not have an upfront cost and this means that since nobody is receiving money for a license from a licensee, then the licensed party has to provision for its own support, running and any other additional costs. Whether one needs documentation, installation or use help, testing or modifications, there are always costs involved, even if they are not necessarily of a financial nature.

The major advantage of Open Source is then in the freedom space, as for example having the benefit of not being restricted in use by commercial license terms or the option to learn, modify and improve the program to one's own requirements and further distribute it freely. Also it must be considered that the total costs (TCO) of using Open Source are in practice shared between the community members.

[44]Costs can be involved and are actually encouraged; yet as the program's user has the right to unrestricted distribution, according to some of the licenses, the licensing cost is eliminated.

2.2.1.2 The Open Source Communities

The freedom associated with these concepts radically changed the way that software is developed within the alternative Open Source paradigm. Unlike the closed source projects, the Open Source ones are not restricting the community of developers to a closely controlled domain, as a non-open source would typically restrict to a single company, with an own well-defined agenda.

In contrast, with Open Source projects large communities are naturally formed. These do not have entry barriers and also do not restrict themselves to just developers. Instead also non-developing users and topic experts are welcomed as they provide invaluable feedback and insight to the developers, as well as testing and support. This feedback loop is also present in the closed projects, yet here it is much shorter and much more effective as everyone has the possibility to influence the project directly and has the freedom to further adapt it to his/hers particular requirements.

The size of these communities is very important and critical to the success of the project. The reasoning is that more eyes will easily spot bugs, will implement more features, will test more cases and also the contributions would be much richer.

Large Open Source communities are often organized in loose hierarchies, based on merit. One would advance in this hierarchy if his/hers contributions are recognized as positive and would be demoted in case he/she is acting against the interests of the community. Modesty is considered to be an important quality of a good developer. As such, in most cases decisions do not need to be enforced, but often are let at the latitude of major contributors, which are recognized by the large community as leaders due to their high level of commitment to the projects.

Healthy Open Source communities do not make restrictions on who contributes, how to do or what gets into the project. In case of severe disagreements, the open source projects have always the "fork" option, where the communities get split and also could later re-unite. Although not ideal, this mechanism is critical:

- as to ensure a healthy development road-map which would satisfy various targets;
- as to re-establish the purpose of the projects as their context evolves, against stationary forces;
- as to repair various digressions from the healthy community principles.

For example, it is quite common that some users rely on existing functionality and would reject new features as they would indubitably bring with them new bugs. For others though the benefits of the new features would overcome the problems cause by new issues. In this situation, the community could be simply split based on the immediate interest into following two new forked projects: a traditional/stable one and a new target/release one. It is also often the case that once the next release matures, the stable teams would embrace it and the community would be rejoined. In effect, major communities employ this mechanism as part of their regular release

strategy, without being sparked by conflicts, but as a mechanism of evolving for example projects which require utmost stability.

2.2.1.3 The Open Source Licenses

Although the initial statements about free/open source software are simple in nature, the realistic development environments have given birth to a plethora of license models and licenses. As almost all of them abide by the Open Source principles, many are accepted as such. The Free Software Foundation and the Open Source Initiative both maintain lists of approved such licenses, [120] respectively [121].

A deep analysis of the individual licenses is beyond the scope of this dissertation. However there are several major types and concepts of open source licenses which are worth a brief discussion:

1. **Public Domain**: applies typically to software that does not have any restrictions. This is for example the case when a copyright is not in place or it has expired.

2. **Berkley Software Distribution (BSD)-style licenses**: are permissive licenses which have no restrictions on distribution. The difference from the Public Domain licenses is that this retains a copyright notice as a measure of disclaiming warranty and to indicate the license status. Some other similar licenses, like the Apache-style ones, demand that the original notices are carried over. The scope of these licenses is to provide the full liberty for the involved parties such that they can choose to use the software also in closed source projects and hence maximize the code's usability.

3. **Copy-left licenses**: These licenses often have the restriction that all redistributions must maintain the same license and unlike the BSD-style licenses, closed-source-only redistribution of modifications or derivations is not allowed. The most known is the GNU General Public License (GPL) [122]. The copy-left licenses have the scope of spreading and protecting open source by "infecting" contributions and forcing them to stay Open Source. This fact though makes the license also controversial, as some argue that it is not truly a free licenses as the redistribution is restricted, while others argue that it is exactly this restriction that keeps the community healthy as progress must be shared back and not closed.

4. **Pseudo-open-source licenses**[45]: These licenses claim to be open source, yet they include clauses which for example restrict the essential freedoms of the program's user. Often commercial companies employ such licenses to show the source code, as a sign of good faith or as to facilitate understanding of internal mechanisms. In some cases the restrictions are minor, introduced

[45] This license category is listed here just to show that there are situation when the source code would seem open as in one can read it, yet the license is not aligned with the open source concepts and hence fare very badly on the liberty plane.

by mistake and later removed on contact with one of the free/open source software groups. However, in most cases there are serious limitations which are definitely not in the spirit of the free/open source movements, but designed such that competing parties would not be able to use the source code to gain any relative advantage.

2.2.1.4 Business Models

Open Source is a functional part of the Internet's evolution mechanism. Source code sharing has drastically improved the standardization and evolution of new services as virtually all the participants can contribute and improve the environment, each sharing the costs and the benefits, each capable of optimizing the product for their own needs. Large number of users and contributors means that also the quality of the code has improved, with issues like security clearly even better covered than in the alternative closed source models[46]. It is actually very common that code written tens of years ago is still successfully and reliably used in production today even on the most modern computing architectures, as it was proved to be correct and stable. Although the benefits of Open Source are far larger in the freedom space than in the financial aspect, reducing the TCO is another critical point that aided in the agile evolution and expansion of the Internet.

Studies looking for example at the Operating Systems used in the top 1 million web servers show that in at the end of 2010, more than 30% [123] were running the Linux [124] Open Source operating system kernel. To indicate also some extremes, at the end of 2010 Linux was used in 91.8% of the TOP500 super-computers [125] while also making serious inroads into the mobile device platforms, with 81% of the smart-phone market (Android [126] and Linux) [127]. These numbers show that the Open Source is widely used on a very large number of applications, from the ones demanding utmost performance and stability to small mobile devices. As a result it is safe to conclude that Open Source software like the Linux kernel has successfully passed critical tests in demonstrating its readiness to act as one of the premier Operating Systems.

In the recent years, the ISPs have experienced the benefits of Open Source through the use of projects like the aforementioned Linux kernel, but also through projects like the Apache HTTP Server [128] or the MySQL database system [129], which became milestones in building the Internet services. In fact, using Open Source software is now a daily practice, especially as, in order to be lean competitors, the ISP companies employ software engineers and developers on a large scale. These companies routinely release new features even on a daily basis and as such they need to employ reliable and good software packages.

In fact, with large number of software developers involved, it is only rarely that the Internet companies would need to reach-out for outside-built software and

[46] Open Source projects are force to provide security by design, with very well scrutinized implementations, while close source project often employ security-by-obscurity

tools[47]. Instead, it is more economical and better customized if the problems are solved internally. And once these problems are solved, many results are being shared through Open Source.

This process might seem counter-intuitive – why would a commercial company share its results with its competitors for free? In reality this is one of the most important benefits of Open Source.

Let us consider a hypothetical example, where any names and subjects are purely fictional. *Company X* needs to troubleshoot a networking problem in its new VoIP network:

a) For this it needs a protocol dissector which will inspect and analyze the network traffic. To solve the problem quickly and efficiently, developers from *Company X* write a VoIP SIP protocol dissector for the Wireshark [130] Open Source protocol analyzer. In this process, *Company X* avoided having to invest into the development and debugging of a complete network traffic capturing, filtering and dissecting tool, but reused the existing ones from the Wireshark project. The problem has been solved quickly and now *Company X* also has a tool that it understands and can in the future use and extend to solve new problems.

b) *Company X* contributes the SIP protocol dissector to the Wireshark Open Source community and this is being integrated in the main development tree. At this point there are several further and immediate advantages:

   1. *Company X* does not need to keep an internal development tree of the Wireshark with its internal SIP protocol dissector, as Wireshark now includes it in its standard distribution. *Company X* will start to benefit then from the recent Wireshark progress, without having to invest into keeping its internal development tree in sync.
   2. The community contributes back with fixes and comments on the new dissector, directly improving *Company X*'s contributions and main topic of interest.
   3. *Company X* asserts its know-how as VoIP specialists in the Wireshark community, which ultimately is formed by numerous developers active in the VoIP field.

c) Another *Company Y*, has access to the new SIP dissector and it uses it also to debug its network. It notices though that from the SIP interpreted messages, it could extract the necessary information to be able to follow on the SIP included SDP and following RTP media streams and debug audio quality issues. As such, it writes a new module, reusing the functionality contributed by *Company X* and providing media stream extraction and statistics. Understanding the value of contributions in Open Source, also *Company Y* also contributes back its additions.

[47] As software platforms become ubiquitous, specialized hardware is limited to critical and specific applications; as such the typical "tools" are now software programs and the hardware platform is uniform.

d) *Company X* can now also use the improvements made by *Company Y* and as such it can better use Wireshark as a tool for debugging VoIP networks.

Of course, this example is simple and not aligned with any real facts regarding the SIP, SDP and RTP dissectors from Wireshark. However it illustrates how the Internet services companies cooperate to share their costs.[48,49]

This rather long example should serve as a model of the Open Source contributions and benefits cycle. One has to notice that although progress has been shared with the potential competition, many advantages come from it. The model itself has no value without sharing as the changes will have only limited validation and also the system would reward back the contributors only on a short term, as by reusing some free open source parts saved some costs.

Regarding costs, the example above gives the impression of a "free" as in no financial costs. This certainly is not true, as both *Company X* and *Company Y* invested development time. Even if *Company Y* would have not contributed back with a new feature, many similar users of *Comapny X*'s contribution would've sent back at least their feedback, which is just as valuable in validating that the tool is correct.

Getting back to the ISPs, they are using the Open Source model to their benefits. Of course each company would internally decide on where to draw their contribution lines based on own interests and it is not uncommon for them to have developments based on Open Source which they would never or only later release back to the community. In most cases, these are related to own principle advantages and values, which are best kept guarded for a while. When these advantages would not be any more so differentiating, the contributions line would be redrawn as to take advantage of the Open Source community advantages.

And of course, ISPs would still use a lot of closed-source components. Overall though, the next chapter will seek to clarify why the Telecommunications Domain

[48] At this point the reader would wonder then how can then *Company X* differentiate itself from *Company Y*, when they practically have the same tools available. The fact is though that on one side these tools were not what primarily made *Company X* or *Company Y* successful. Of course, using the best tools to improve the QoE for its customers would give an edge to both companies as by actively investing here have improved their services. On the other hand it is probably not the tools but how they are used that would make a company succeed.

[49] Also another point would be that there are lots of commercial companies providing alternative traffic monitoring solutions, which would have been usable. If the problem is particular and not in scope for *Company X* developments, this would be obviously the better solution. But even though in some cases this could have been a shorter and easier path to solve the immediate problem, there are several disadvantages. First, if the protocols are new and they are new in a large amount of situations where innovative products are offered, such tools might have not been already available at all. Then the costs would have included also the traffic capturing tools and future upgrades would also not be free like the ones from community-cost-sharing *Company Y*. There would be only limited customization to *Company X*'s environment and needs, with all such changes being subject to the interests of the commercial tool developer and vendor. *Company X* would've not asserted itself in the community as specialists; neither would they have gained the experienced specialists understanding the subtleties of the protocols. Overall, the difference would be that *Company X* would've just bought a black-box tool, instead of investing into its own development.

sees much lower Open Source penetration today.

2.2.2 Open Source in Telecoms

As opposed to the Internet world and market, traditionally the telecommunication business has been a specialist environment, where only a chosen few companies had the extensive know-how required and the financial power to act as TEMs. Unlike the Internet domain where "best-effort"[50] performance is used widely, in telecommunications the quality standards align with the "five-nines" high availability Service Level Agreements (SLAs) and are even strictly regulated by national and recently also international authorities.

In this situation, operators often require insurance that their networks would function within parameters as deviations would be very costly. Such a thing cannot be provided directly by an Open Source community as anyway their nature would in most cases disclaim any warranty. Producing such equipment, which is actually in most cases also used on the critical Internet infrastructural parts like its backbone, is then a complicated matter which involves high costs.

On the topic of these critical CNs, in the current architecture migration environment, it is of utmost importance that operators have the opportunity to trial early the new concepts and service opportunities such that feedback would be sent to the standardization bodies as early as possible. But the TEMs would of course demand that their R&D efforts would be rewarded regardless of the market changes. This would work if the operator is certain that it can turn the big investment into profit in the near future, but otherwise the risk is too high and a potentially blocked state is reached. This reactive loop between the customer-needs, network operators, equipment vendors, standardization bodies, then again network operators and customers is quite long and plagued by individual bias.

The Internet domain manages though to avoid it. Its CN is in a continuous modification as protocols arise and are obsoleted very often in comparison. The principles call for a different model, where the concepts would be first trialed without standardization in real world use, feedback is gathered and the technology recognized as a standard, such that further implementations would be compatible. Practically for every Internet protocol there are Open Source implementations, which help study, debug and improve the new concepts. Of course, the critical infrastructure is often a case of proprietary optimizations, yet the existence of open implementations before and after the ones to be used in the critical infrastructure is probably essential to the Internet evolutionary speed.

[50] The term is not strongly defined. A best effort delivery system is similar to the postal service when a packet is not scheduled to be transported in advance and as such its exact delivery date is not known, cannot be guaranteed and there are no repercussions on delays. IP routers typically implement such a system for the sake of simplicity, although today's advancement seek to change this in order to maximized network utilization and the quality of services. A QoS-ed network cannot function over such a best-effort network and solutions are provided at the cost of IP router complexity.

Besides the CN critical infrastructures of the operators, the service space must not be neglected, where the operators are in similar positions if not even in direct competition with the Internet world. It is then logical that the service space problems should be regarded similarly if one expects the operators to be competitive.

For Internet services, "best-effort" performance concepts are in most cases sufficient. Considering that many such services have a temporal sweet-spot of exploitation spanning only a few years or even months, using expensive and inflexible equipment for these types of services would practically take away the competitive price advantages. This resulted in the Service Providers in the Internet world using in high concentrations in-house development, which allowed for lean business models, such as the perpetual-beta of Google® or the almost hourly release interval of Flickr®[51]. And there are clear benefits of these lean models, as for the given example Google® practically outsourced part of testing to its users and Flickr® is able to better compete by having the latest features available immediately. The given examples do not necessarily apply direct to the use of Open Source, yet it shows the shift in operation model for Service Providers, which is the most important new role of telecommunication operators in an NGN environment.

The link between Open Source and an in-house development model is that switching to an in-house development model is greatly aided by using Open Source, as the development costs are practically shared. Of course, in order to maintain its edge, each Service Provider would keep its main advantages hidden from their competition. However, the commoditized components are far more economically maintained when this effort is shared between the benefiting parties.

To this extent the operators are commonly plagues by vendor-lock issues, even though there are clear standards for the interfaces. Once an interoperability issue is discovered between an existing and a new equipment, the operator has rarely a chance to solve it on its own and could be kept hostage by vendor interests of denying interfacing for example with cheaper competing equipment.

In the context of future NGNs, the added value of the Service Provider would most likely reside in the provided services, the end-to-end user experience and the area of coverage. All these edge added value advantages reside mainly in the Service and AN Layers, with the CN layer being common. This is further emphasized by the fact that the CN has to be a common one for architectural reasons. As such, it is conceivable that the NGNs will in the future employ similar development models as the ones found in the Internet service space and as such also use extensively Open Source software.

2.2.3 Open standards, interfaces and the impact on R&D and academia

Another topic that differentiates between the Internet world and the telecommunication legacy business is the standardization process.

[51] The software release changed from beta to gamma, which means that the software updates are continually rolled and validated by the users

Traditionally, when a new service was demanded by customers, the operators would first turn to the standardization bodies with requirements. Those organizations would attempt to find a technical solution and write specifications, which would be passed to equipment manufacturers as blueprints. Next the manufacturers would provide their products for integration within the live networks.

This model, although in use for a long time and providing very reliable technical solutions, has its inherent flaws that can be already observed: long convergence duration, no direct feedback from customers, biased by the operators and so on. Notable for this model are organizations like ITU-T or ETSI.

On the other hand, for the Internet world, a different and de-centralized model has emerged. There requirements are immediately fulfilled with practical solutions deployed on the spot to customers and only later, when the best options have emerged and have been trialed, standards are required to ensure interoperability, for example, between different implementation of similar concepts. As such, the solutions are firstly trialed out and standardization begins after a first round of customer feedback, which establishes the solution as a viable one.

In the Internet world then the standardization process in itself is an Open one, where the proposals are published for public review and only accepted after a consensus has been reached. Additionally, all Intellectual Property Rights (IPR) must be claimed during the draft phase of the standard and the claimers are compelled to provide for anyone fair, reciprocal and non-discriminatory rights on using the described technologies. This model, used for example by IETF, ensures a better, more open and fluid standardization process which enable a more alert technological progress than the previous one.

As it can be observed, the two approaches are quite different. A brief analysis is presented in Table 2.6.

Even before, but most notably with the standardization of NGNs, the telecommunication traditional standardization bodies have recognized the benefits of the open Internet model and have improved their procedures. Furthermore, the Internet generated IP specifications for protocols and interfaces are now largely used in NGN standards as references, in a model that adds value by indicating a set of rules on how to use those specifications to ensure well defined quality, performance and security levels.

Open standards are very important in order to ensure a healthy competition between different players, as the progress cannot be locked anymore through vendor-lock tactics, but value is driven purely by market demand and R&D. Then the open standards can actively stimulate development and progress by reducing the digital divide between different markets, as the basic know-how is no longer requiring a high entrance fee. Finally, a standard that has been exposed for public review is inherently safer in exploitation: even though the inner workings are exposed, security by design is better than security by obscurity.

Open standards are not only breaking commercial barriers, but also educational ones, as progress and contributions are open to anyone. Considering the academia output of graduates ready to enter the industry, open standards and tools are a

Characteristic	Telecommunication Domain	Internet Domain
Motivation	Financial chain, on customer demand, expects a direct revenue from the customer and a direct reward of the vendors	Community cost sharing, for the purpose of interoperability between various implementations
Innovation	Indirect, on request, filtered through commercial considerations first	Directly innovation driven as added value
Driving Force	Based on customer demand, high power due to financial reasons, but few contributors involved	Based on practicality reasons, low individual power, but many contributors involved in very large communities
Influences	Closed evolution influenced by operators and vendors, indirectly by customers	Open evolution, the entire community can contribute and influence the concepts
Evolution	Concepts can evolve, but typically only over entire iterations of the process	Continuous evolution, as standards are being amended and improved
Number of standards	Fewer, generations clearly delimited and individually exploited	Many, relevance indicated by use
Time-to-market	Very long, as the concepts need to be first developed, then standardized, implemented and deployed	Even zero, as the concept is first available, then standardized

Table 2.6: Differences between the Telecommunication and Internet Domains Standardization

big source of education and as such significantly increase the technical value of the new work-force. Without open standards, the telecommunication businesses must invest additionally into the education of their new employees, without being able to find highly skilled and ready-to-work personnel. Open source tools and prototype further complement the open standards by providing un-biased hands-on experience to the work force as well as to the academia to perform practical experimentation and provide academic advice and feedback.

2.2.4 Open Source Foundation Tools

This section seeks to provide an introduction to the foundation software which was primarily used in the realization of the dissertation topic, mainly the OpenIMSCore project. Special attention will be awarded to the SIP proxy platform, while the additional tools will be just briefly introduced.

2.2.4.1 The SIP Express Router (SER)

With the target system being an IMS CN, a wide range of functionality must be covered. The main feature of interest here is of course the capabilities of routing and processing the signaling protocol SIP. Back in 2004, when the first initial experiments on IMS were started at Fraunhofer FOKUS, they were based on a rising Open Source project, the SIP Express Router (SER) [27].

SER was started in 2001 by a small group of researchers with experimentation roots dating back to 1995, group which included Prof. Henning Schulzrinne, one of the co-authors of the SIP specifications [88, 49]. One year later, the project is launched publicly under a GPL license, which broadened significantly the contributors space, this no longer being limited to just the Fraunhofer researchers and graduating students. Over the next 2 years SER was adopted not only by researchers, but also by many ISPs and commercial companies, which use it even commercially up to today, as a free and flexible platform for providing the increasingly attractive VoIP services. A community effort started to emerge as the *iptel.org* laboratory, which provided an open platform for working and interfacing with SIP [131].

Over the next years SER continued it evolution both in the Open Source space, but mostly it broke away from its research roots into commercial companies, which could better offer carrier-grade services to the booming user community. The project itself has been even forked and un-forked several times as different groups had different requirements in the feature-rich/stability domains, such that over the years the common base was customized under different names: OpenSER, OpenSIPS, SER-IMS[52], Kamailio or SIP-Router.

A notable achievement of the SER project is that, beyond being recognized as defining the standards for flexibility and performance for SIP proxies and servers, it is also solid and stable enough to constitute a serious alternative to commercial equipment vendors in the carrier-grade telephony space. For example, 1&1 Internet AG, one of the major ISP in Germany and one of the leading web hosting companies in the world, currently uses Kamailio at the back-end for providing reliable telephony services for its multi-million telephony subscriber base with VoIP [132].

From a technical perspective SER is a `C`-based software with a partly standard modular architecture, where individual features can be enabled dynamically at runtime by configuring and loading functionality libraries. Yet besides these, there are 2 main driving factors which shape it special traits.

1. Deeply integrated and optimized SIP protocol functionality, without a clear separation of the protocol stack.

   The signaling processing design has been from the start anchored on the requirements of ultimate performance. With a text-based message format similar to HTTP or SMTP, SIP was unlike the typical SS7 signaling protocols, in the sense that even a protocol parser has entirely different processing characteristics, not

[52] Technical name for the frozen SER base plus the customizations which constitute the OpenIMSCore project

to mention its openness and flexibility which constitutes a significant difficulty in maintaining and upgrading the protocol stacks.

The SER designers went for full optimizations from the first design days, targeting to obtain the best performance possible, as a benchmark and reference for SIP signaling processing. Starting with message parsing, 32-bit optimized techniques have been used, which enabled very fast analysis of the message tokens, with groups of 4-bytes being consumed at once in just a few CPU cycles[53].

From a processing perspective, the model was designed around scalability, with parallel processing units enabled to process individual messages side-by-side, avoiding also as much as possible blocking and waiting due to Inter Process Communication (IPC). An early assumption on the SIP traffic using almost exclusively UDP as transport caused an unfortunate design choice of a multi-process model. When considering also the TCP transport, with its peculiarities in having to maintain the transactional association on a connectivity basis, a multi-threaded model would have been arguably better. The chosen multi-process model has been though deeply optimized especially in the IPC regard, with easy-to-use shared memory and very low latency locking mechanisms.

To keep it aligned with the carrier-grade demands, the memory space and allocation is implemented in a limited manner, where unlike the standard model of allowing a virtually unlimited memory space, memory pools are pre-allocated. This prevents swapping and an unpredictable behavior, while also further optimizing for performance. Additional mechanism have been built to aid with debugging memory related bugs and leaks, such that development cycles were greately optimized.

2. Extensive flexibility in configuration and exploitation provided by a signaling routing script as part of it configuration files.

   While building such a SIP proxy engine, an important difficulty arises from the open space of functionality. Besides the basic operations, it is not clear or even possible to program all the possible scenarios. What exact functionality would the respective node provide is rarely known from the beginning and in many cases it is actually that this will only be known when exploitation starts, with many changes and adaptations required on-demand.

   To solve this problem elegantly and without requiring that network administrators would need to "program" the functionality directly in `C`, SER exports a simplified language for defining routing scripts. This allows for a specification of the message processing steps in a simplified language, as part of the configuration. This message routing scripts is compiled at run-time for optimization purposes and executed step-by-step for every message received.

   The routing script splits then definition of the processing logic in 2 main parts:

[53] The parser uses `switch` blocks to identify tokens in messages, by using as `case` values all possible combinations of 4-byte character groups. While very fast, this strategy entails rather hard to maintain and extend code as the SER platform had the tokens and cases hard-coded.

a) one that is to be optimized, implemented directly as C-code in the core or later on in additional modules by software developers;

b) and another as an orchestration of the basic building blocks operations, left as a process control method to the network administrator in the configuration files.

The benefits are tremendous, as SER is then not a specific purpose tool, but its actual behavior is very easily changed by the operator before putting it into service. In practice this translates to many operators using it even besides other SIP proxies and servers, many times as a Swiss-army-knife, to enhance their functionality and to manage the incompatibilities and shortcomings between and of various equipment and their SIP protocol stacks.

SER, as directly targeting the implementation of SIP proxies, represented a prime choice for realizing the IMS CSCFs. The initial motivator in the work presented here was the need to trial for the first time the service triggering capabilities proposed with the IMS S-CSCF first concepts, as they started to emerge in the 3GPP Release 5. This first step was undertaken in the author's Diploma Thesis, back in 2005 [133]. Following on and also considering the reference in SIP processing performance which SER was setting, a group of companies led by Intel® and Fraunhofer FOKUS sponsored a Special Interest Group (SIG) and research effort into prototyping IMS for the target of understanding and benchmarking its performance. OpenIMSCore, was the result of these combined efforts, based mainly on the SER platform.

Even today, SER and its related forks represents a continuously evolving open reference for providing efficient SIP signaling processing in CNs.

2.2.4.2 The MySQL Database

Also part of the IMS CN is the HSS, which represents primarily a database of subscriber profiles, with an interfacing layer over Diameter. Choosing then a good and powerful DBMS is key in the final performance and flexibility of the HSS. Replacing it later with another solution is also feasible, once a standard database query language is to be used, for example Structured Query Language (SQL).

MySQL [129] is then a primary choice, as the most used such open source Relational DBMS (RDBMS). Arguably telecommunication carrier-grade systems have favored in the past commercial solutions, yet lately the performance and ease of use provided by MySQL have enabled large scale uses, at least for the 2nd grade background system as an effective means of cutting costs. MySQL has also been adopted by large operations on the Internet, which prove its readiness to serve even at large scales, for example running main databases for Facebook or Twitter.

MySQL has been developed in C and C++ and is in fact one of the Open Source projects with the largest developer base. As its name implies, the database interrogation is performed by using SQL statements. There are many adapters and

libraries available for access to the database, such that integrations are not an issue regardless of the client platform.

As it comes as part of almost all Linux distributions today, managing this backend is very easy, letting developers free to concentrate on the development of frontend applications.

MySQL has been started around 1994 and is executed in a dual-licensing model: it is available freely under a GPL license, yet its copyrights are owned by a commercial company which offers it also under other proprietary licenses in paid commercial model. MySQL AB, the commercial company behind it, has been acquired by Sun® microsystems and then Oracle®, which raised community concerns on the continuation of the GPL track, especially due to obvious conflicts of interest, as the corporate owners had parallel closed source offerings, with which MySQL was in direct competition. While currently the GPL licensing is maintained, as a natural reaction of the communities, many forks have appeared, not only as special purpose adaptations, but also as a direct 1-to-1 compatibility to the original to elevate uncertainty in the Open Source space.

2.2.4.3 The Apache Web Server, PHP and the Apache Tomcat

To provide the means to provision the IMS CN parameters and operations in a facile manner and especially to provide the human provisioning facility for the HSS stored subscriber profiles, a GUI application was required.

Again as a logical choice, the most used Open Source web server has been adopted. The Apache Web Server [128] is a `C`-based web server developed by the Apache Software Foundation, well-known for its role as a powerful driving engine of the World Wide Web (WWW) initial expansion. Its structure, which also inspired SER, is a modular one, with a core providing the main functionality and additional modules adding capabilities. It features high-performance and is particularly adaptable to many platforms, which contributed to its adoption on more than half of today's web servers in the Internet [134].

For adding logic to a web-based user interface, a platform is required. A simple yet very powerful such environment and programming language is offered by PHP [135]. With it, one can simply embed PHP code into `HTML` and this will be executed by a module in a web server like Apache, upon every client requesting the respective page, which results in a dynamic page being served to the user.

PHP started as a much more flexible replacement for Common Gateway Interface (CGI), the name acronym originates as Personal Home Page/Forms Interpreter or Personal Home Page Tools. It evolved dramatically as one of the main choices for dynamic pages in the early days of the WWW, into a general-purpose scripting language, targeting web usage, with good database support and multi-platform capabilities but usable in fact for many other purposes, as for example stand-along exploitation.

While the Apache/PHP combination provides a very powerful platform, the need came in the OpenIMSCore project for a built-in rather than separate interface

serving facility. On one side this was to ease the deployment as to not require the separate configuration and provisioning of a web server. On the other hand interfacing on HSS is not sufficient between the provisioning interface and the DBMS, but also certain operations would require direct interactions with the HSS engine. These could either be performed through the DBMS, which would be rather complex, or directly between the interface and the HSS.

As the HSS was built standalone as a `Java` application, the Open Source Apache Tomcat [136] has been chosen as the web server which could be unitary executed together with the HSS `Java` application.

The Apache Tomcat is different from the Apache Web Server, as it is entirely written in `Java` and its purpose is to support web pages which rely on `Java Servlets` and `Java Server Pages` to serve dynamic web pages. In the situation in scope here, `Java Servlets` have been developed as part of the HSS provisioning and operational interface.

2.3 Related Work and Toolkits

Around the start of the OpenIMSCore, in 2004-2005, there were very few specific IMS implementation projects, in either open source or commercial space. Considering that the concepts only just appeared in 3GPP Release 5, this was normal as typically even first experimental products appeared with a delay of about 1-2 releases[54].

Most notable for experimentation related to IMS was in fact the same SER platform. As an Open Source platform this allowed many researchers to realized simple IMS-like prototypes. While on one side it was easy to add simple behavior through the routing script, the complexity increased to barely manageable levels when using it exclusively to provide functionality.

One such notable effort was started and successfully implemented a significant amount of IMS functionality at the PT Inovação, a research branch of Portugal Telecom, by João Filipe Plácido and Luis Silva. They have recognized though that such an effort was complex and only limited in success, as the scripting capabilities of SER were hardly sufficient for achieving a good compliance level to the standards which required even the use of new protocols. With the release of the OpenIMSCore as Open Source, their project has been discontinued and the developers have joined the Open Source community.

Continuing on the SER platform, in a sense, most of the carrier-grade VoIP deployments were following on similar concepts as the IMS ones. All such deployments featured first a security gateway, SBC, which firewalled a CN from unauthorized signaling, similar to the P-CSCF in IMS. In the CN processing was distributed by light load balancers similar to I-CSCFs, to SIP registrars and other signaling processors similar to the S-CSCFs. The registrars segment the subscriber base as to provide

[54]First Release 5 pre-products appeared around the standardization period for Release 7, when Release 5 definition was stable enough and well understood.

scalability of performance. RADIUS servers provided authentication and accounting services, similar to the HSS concepts.

Such proprietary SIP CN architectures were used by most of the VoIP providers and unfortunately were far from being standardized or cleanly designed. Each solution had to struggle and find for itself the needed performance and interoperability standards, in many cases rediscovering the same performance bottlenecks and reaching similar, yet incompatible solutions.

Studying such solutions was an important initial step for designing the first IMS prototype. Many parts could be almost entirely reused, yet most required a re-design to fully align with the IMS requirements. For example, the SER registrar functionality, although having the base functionality, did not initially support registration event notifications. Or the initially mandatory AKA authentication mechanism was never implemented before. The RADIUS modules provided a good start-up point for the similar Diameter message formats, yet integrations of a full-blown Diameter stack required features which were not yet supported even by the SER core functionality, as its messages are transported over a reliable TCP or SCTP connection, maintained by a complex protocol state. These differences are significant enough to influence the performance characteristics of IMS versus such proprietary carrier-grade VoIP architectures.

An important issue was the lack of clients and even any implementations for that matter, for the newly mandated by 3GPP Digest-AKA-MD5 authentication algorithm. It can be argued that the simpler and widely spread Digest-MD5 algorithm was usable, although id did not provide all the new features that AKA did, as for example reciprocal authentication or generation of ciphering and integrity keys for further protection of the signaling on the potentially unprotected UNI.

AKA was of course used already in performing the authentication of mobile terminals in UMTS networks. Yet there was no open implementation which could be used to validate a new implementation. The OpenIMSCore one was then the first to be offered to the community as part of the HSS Authentication Center (AuC) implementation and was quickly adopted in clients. It was first ported to KPhone [137], then to the IMS Communicator [138] and other newly spawned IMS clients as well as test tools like SIPp [139]. An interesting effect was created by an error in the initial implementation, which although re-implemented for other purposes and even programming languages went for years unobserved. This showed that open source projects are biased in correctness by problematic code being studied instead of standards and reused verbatim in other implementations.

As the OpenIMSCore captured the attention of most developers, most of them chose to contribute directly to this project, with a peak between 2006 and 2009. Other projects have also started in parallel or have spawned and reused the experience gained. The HOTARU Open Source IMS project [140] started as an effort to provide a similar, open source, but non-GPL platform in Japan. Besides aiding in interoperability testing, there is little information or code still available from this project.

Important efforts and outcomes of the OpenIMSCore are also the redesigns and

incorporation of parts and concepts into the main development trunk of Kamailio (fork of SER), towards offering IMS features in a carrier-grade VoIP deployment based on the SIP-Router project [141]. The project leader, Carsten Bock, has also announced that work is in progress for a rewrite of the HSS targeting performance and reliability [142]. Although still in testing phases, this effort represents an important outcome of the OpenIMSCore project, which was mainly focused on the research aspects, by providing carrier-grade exploitable Open Source alternatives.

The same SER basis has also spawned commercial IMS CN solutions and products, as part of the Tekelec SIP Signaling Router and CSCFs solutions [143]. As these are closed source solution, they cannot be further characterized, other than their intended purpose of providing NGN and IMS carrier-grade CN equipment.

Although more or less unrelated, several other Open Source projects should be mentioned here, as they have a major impact on the telephony domain. First, perhaps the most known such project, Asterisk [144], is a software implementation of the Private Branch Exchange (PBX), providing many advanced telephony features like conferencing, voice mail, Interactive Voice Response (IVR), call routing. It supports many VoIP protocols besides SIP, yet of course it is not designed to provide a CN service, but more of a complete suite of small-office telephony facilities. FreeSWITCH [145] is another such alternative, which targets different scalability as a soft-switch replacement, towards providing similar telephony services.

Chapter 3

Requirements for an Open Source IMS Toolkit

The present chapter seeks to establish the most relevant requirements for NGN test-beds. Going first through the obvious properties of functionality and performance, the analysis looks then at testing directions as benchmarking and interoperability. Conformance or standards compliance will be excluded entirely and not used as requirements as these are not critical for proof-of-concept prototyping when in the first phases of test-bed establishment. Cost-based considerations will then complete the requirements, ensuring that the volume of requirements and their associated levels will be kept under control.

3.1 Critical Mass of Functionality

The dissertation subject targets the establishment of test-beds infrastructure. A good coverage of functionality for the NGN CN components should be considered, such that relevance and comparability of realized prototypes and tests would be achieved in regard to full-blown products acting in real-life exploitations. Then there are also factors which can reasonably limit the requirements, keeping the costs of establishing such prototypes within reasonable limits.

First of all, before going through a brief analysis on functionality versus costs of implementation, it has to be noted that generally the costs would be composed of two roughly defined parts:

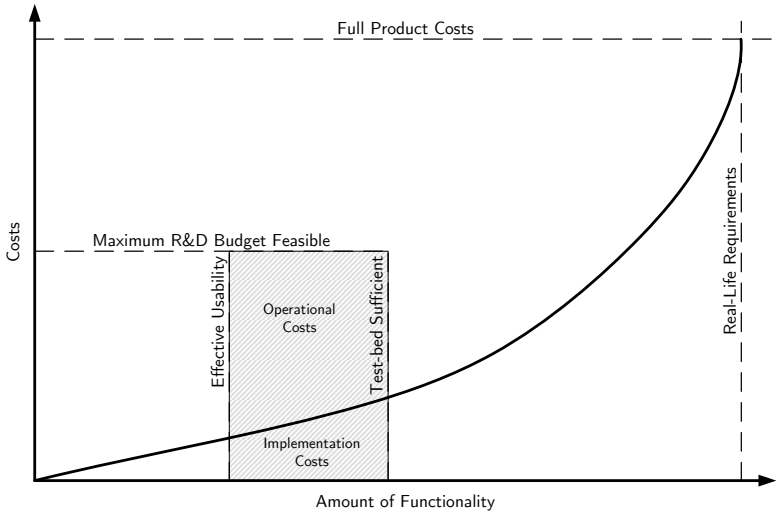


Figure 3.1: Hypothetical Economical Analysis for Test-bed Functional Components

1. a prototype implementation part
2. and an operational part.

It is obvious then that as more is invested into improving the implementation, the easier it becomes to use it within experiments and/or real-life exploitations, reducing the operational costs. In Figure 4.1, the implementation costs would be considered the area under the implementation cost curve and the operational one above. For the discussion here though, please note that the scale or relative values are of course not relevant and as such the illustration or the analysis does not follow on this aspect.

Considering a full-blown product, which ideally would target all the functional aspects as specified by the standards (or at least cover all functionality which would be effectively required for successful real-life exploitation), this system would define the upper costs extremity required for implementation, validation and use, indicated in Figure 4.1 as *Full Product Costs*. Included within these total costs are also the R&D related ones, as for example of interest here, the ones required to establish technological prototypes for trials within test-beds.

However, test-bed purposes are much different than real-life exploitations. Lack or limitation of real-life security, high-availability or performance have of course an

important influence on the test-bed results, yet their costs often do not cover or motivate the benefits of full-blown implementations within the limited purposes.

Hence an intermediary functionality domain, left of the *Real-Life Requirements* can be defined as of interest for test-bed purposes:

- towards the minimum side of required functionality a threshold can be identified, represented by the *Effective Usability* level, or the minimal functionality set required from a prototype as reduced as possible which would still be able to satisfy at least some of the test-bed purposes;
- towards the upper functionality side a similar threshold, the *Test-bed Sufficient* level, would represent the boundary after which extra features would not have a significant impact on results, whether they would be present or not.

In-between these two roughly defined lower and upper functionality limits (left and right on the horizontal axis in Figure 4.1) would be the *Prototype Functionality Ideal Area for Test-beds*.

As the targets here are set to provide NGN infrastructure for the IMS architecture, the following analysis will concentrate on identifying the specific functionality topics for IMS only, which would reasonably fit between the thresholds defined above and as such satisfy test-bed requirements within test-bed budgets.

3.2 Minimal Functionality Requirements for an IMS CN Prototype

The IMS CN Architecture calls for the use of 3 types of CSCFs and the HSS[1]. The CSCFs[2,3,4] have the role of routing and processing SIP signaling at the intermediate infrastructure layer between the AN and the IMS ASs. The HSS would act as a database entity and provide the CSCFs as well as the ASs with the necessary subscriber profile information. These 4 components, as it will be more clearly outlined in the following presentation of basic features, are always required to provide even the most basic functionality.

The following list attempts to define then the IMS test-bed sweet-spot from the functional perspective:

- Basic SIP signaling routing as indicated in [146] and specified for SIP in [49].
- Authentication of IMS UEs and registration operations, as a prerequisite for basic security and IMS signaling routing and processing.

[1]For a comprehensive architecture, please see Figure B.2

[2]The P-CSCF has the role of a security gateway (SBC functionality) for the UNI

[3]The I-CSCF acts mostly as a routing and directory proxy, by reusing global mobility information from the HSS

[4]The S-CSCF is the workhorse of processing signaling, providing the basic services as well as filtering and triggering the AS

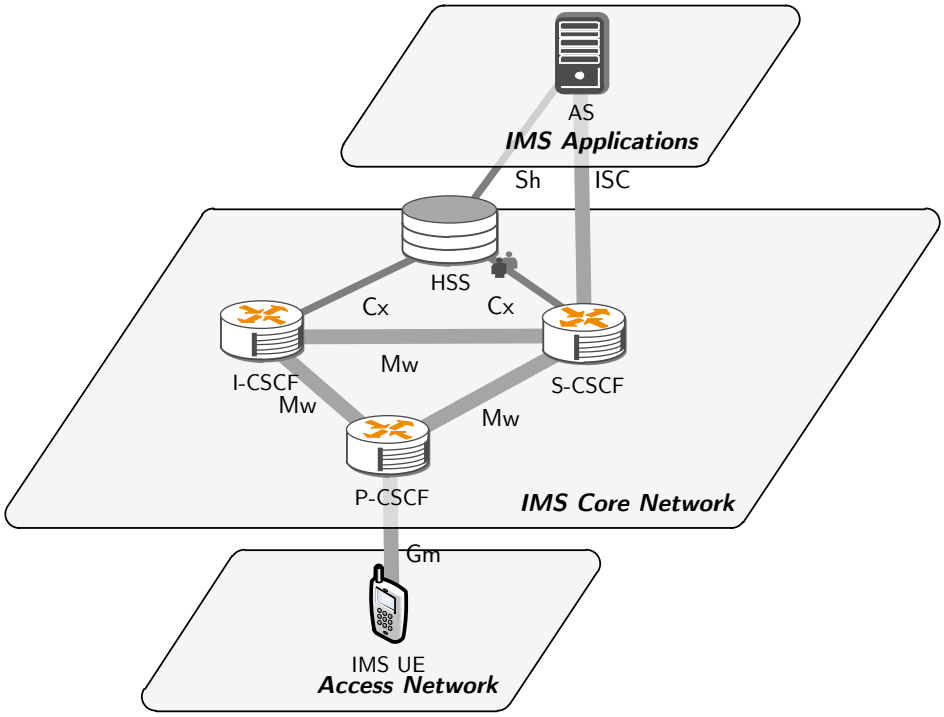


Figure 3.2: IMS Architecture for Basic Scenarios

- Routing of SIP signaling on the CSCF ⟷ CSCF (SIP) (Mw) interface between the CSCFs.
- Use of HSS as a central repository for IMS UE first-level information, by saving and using the UE's registration status and currently assigned S-CSCF address.
- Download of Subscription Profile information from the HSS to the S-CSCF and its use for AS triggering purposes[5].
- Export of the HSS information towards the AS.
- State management for UE registrations as well as ongoing dialogs.

An additional list of nice-to-have optional features follows, as a compilation of non-essential, but still test-bed beneficial functionality:

- Support for AKA security methods as indicated by 3GPP for the mobile domain in [147, 148, 149, 150]

[5] SIP signaling is to be filtered through the iFC and matching initial requests are to be forwarded towards AS.

- IPsec and TLS security for encryption of signaling on the UE ⟷ P-CSCF (SIP) (Gm) UNI interface.
- AAA interface between P-CSCF and ANs for QoS reservation as well as notification on transport network events[6].
- NDS on the Network-to-Network Interface (NNI).
- Advanced AS features, as for example Dynamic Service Activation (DSAI).
- Emergency CSCF (E-CSCF) functionality for emergency scenarios.
- MGCF and Media Gateway (MGW) for basic media processing functionality.
- BGCF and IBCF for interconnections with legacy PSTN and IMS local or roaming domains.

All IMS CN interactions are defined as procedures in [146], such that all the above indicated functionality can be referenced as procedures in this TS.

From an interface perspective, the reference points between the principal IMS CN components required to provide the above basic functionality set (excluding the additional features) are:

- Mw between the CSCFs.
- Gm between the UE and the CN.
- ISC between the S-CSCF and the AS.
- Cx between the I-CSCF, the S-CSCF and the HSS.
- Sh between the AS and the HSS.

3.3 Performance and Carrier-grade Features

The *Carrier-grade* attribute refers here to the capability of a given system to satisfy the highest requirements which arise from real-life deployment and exploitation. This typically refers to the base-level performance expected to be provided by a solid infrastructure and encompasses a broad range of KPIs like capacity, processing latency, security, availability, correctness, etc.

A good number of very important requirements for full-blown products are eliminated or reduced once test-bed environment are considered. Following, a series of key characteristics are listed, each with considerations on feasibility and acceptable levels for the limited scope of test-beds.

[6] E.g. Loss of radio connectivity.

3.3.1 Processing Performance and Scalability

Real-life deployed products have to be implemented in such a way that solid performance could be provided within reasonable costs, with near-linear scalability.

Considering the number of subscribers advertised by TSPs, the highest figures are found in the mobile domain, with more than 6.8 billion mobile cellular subscriptions[7] (96% of the world's population) estimated at the end of 2013 [18]. These subscribers are divided between multiple operators, which even if present on multiple markets, still split their networks on country-sized operational and management domains. Each such typical size national network ranges in number of subscribers between several millions in small-scale deployments, found normally in small countries, up to the more than 759 million subscribers, as advertised by China Mobile Ltd. [151].

Of course such networks would be further split through network fragmentation, traffic distribution and scalability mechanisms, yet the number should set a rough estimation of the scale that real-life networks are set to serve. Accordingly, scales of several million subscribers serviced on aggregation points should be considered.

However, for typical R&D test-beds, the vast majority of functional tests involve less than 10 subscribers (most just a couple or even a single one). Architecture evaluation from a capacity perspective would of course scale up to the aforementioned million-scale magnitude, yet these would not be the norm. Nor would such activities be representative unless testing full-blown products at full capacity, testing normally performed much later than the technology R&D phase, into the actual deployment phases.

The upper limit for capacity can then be safely set at the limits of pre-deployment field-trials, which due to logistical and management costs are rising more sharply than any other costs, would top-out at an order of magnitude of the hundreds or thousands of subscribers.

To conclude, a telecommunications R&D prototype would be considered capacity-sufficient if it can provide enough performance and scalability, within reasonable costs[8], to support subscriber bases of up to thousands of users.

A second critical performance criteria after capacity is the processing latency introduced by the system while processing and passing traffic through the infrastructure. Limits for this metric are typically provided by the direct Use-Case and Scenario requirements[9]. Additional indicators can be found indirectly from the performance of the surrounding networking layers.

Putting then Performance into perspective with regard to Functionality, there is

[7] Obviously a marketing figure, estimating the total number of provisioned Subscriber Identity Module (SIM) cards, not the number of subscribers which have to be handled at any given time.

[8] The term "reasonable" here should be considered as providing satisfactory scalability, such that adding capacity in the upper performance domain would not exponentially increase the costs of the infrastructure

[9] E.g. A reasonable IMS call set-up delay would be considered in regard to the legacy system performance it is replacing; in practical terms for IMS, same or below the recommended ISDN telephony call set-up latency.

a direct correlation. A partial functionality system, like a prototype in scope here for test-beds, only executes a subset of procedures and operations in comparison with a full implementation. When considering then the overall performance of such systems it should be taken into account that this would be better as compared to a real-life situation. For this matter, a recommendation for better performance is then derived, such that it would be preferable for a test-bed system to provide better than real-life expected performance.

Ultimately the performance of a system is highly dependent on the design and implementation choices made, up to around the point of maximum achievable performance within reasonable optimization budgets. After this software related performance edge, the system limitations would be regarded as pertaining to the used hardware platforms.

Today's hardware platforms tend to optimize performance by employing highly parallel processing architectures. Even as traditional telecommunication specialized hardware use such architectures for a long time, the main-stream x86 architectures are catching-up in number of processing pipes, while also enjoying lower costs through economies of scale and built-in optimizations for security algorithms, power efficiency and easy re-purposing[10].

Optimizations and deep integrations in regard to the hardware architectures are then an important criteria for performance. Yet once performance closes again on the maximum capabilities of the hardware, the next strategy step in increasing it would go towards scalability mechanisms.

The CN architectures in scope have been designed from their inception with scalability in mind. There are multiple levels of implicit architectural scalability, which allow for fragmentation of the network infrastructure on criteria like subscriber location, subscription type, service domain and so on. The result is that the individual functional nodes can be instantiated multiple times as required, through a built-in architectural scalability.

Each type of node is expected to have different capacity and scalability parameters based on the different types of processing loads it serves, such that the overall network topology would not be a multiply-all case. To exemplify for IMS:

- the P-CSCF scalability is ruled by its location close to the AN and the scalability of security association mechanisms; if collocated with the actual radio base stations, restrictions on power consumption and efficiency would also become relevant;
- the I-CSCF has a good performance as not tied to individual subscribers and the most likely limitation factor is the number of SIP transactions which can be processed per second; multiple instances can provide effective live load-balancing;
- the S-CSCF scalability is ruled by the number of subscribers that each instance needs to serve as well as how heavy the respective subscribers' profiles are in

[10] See for example trends similar to the ones in Software Defined Radio (SDR)

term of services which need to be triggered; hence the scalability here is not only of a processing load nature, but also dependent on the information caching capacity available within reasonable performance levels;

- the HSS is a light-weight front-end to a database back-end; accordingly its scalability would be almost entirely ruled by that of the back-end DBMS, with all the associated and well-known performance and scalability profiles.

With the implicit architectural scalability presented above, further scalability, if required, can be simply obtained by functional design. Hence the additional need for extra capabilities in terms of scalability is not of critical concern.

3.3.2 Stability from the Perspective of High-Availability

TEMs normally design and engineer telecommunication equipment to very high standards of availability, as requested by TSPs, which in turn have to abide by strict SLAs. The five-nines rule is used here very often as a service stability and availability reference. Considering that the resulting very low allowed downtime would not only be counted during service break-downs due to equipment faults, but also during network or service reconfigurations (which would simply not be feasible without service interruptions), such equipment is engineered from the perspective of providing at any moment the service, even in some of the most extreme situations like faulty hardware or software conditions.

When considering the stability under high traffic volumes, the real-life equipment should be engineered in such a way as to still fully service a reasonable[11] amount of subscribers below the declared maximum capacity, while rejecting or ignoring the traffic from the rest [12, 152, 153].

Practical tests show though that, unlike in legacy SS7 systems, NGN architectures using SIP would be harder to control on such overload situation, mainly due to its increased software complexity. Judging by results from [154], systems which would not be properly engineered might be plagued by high-stress issues and Denial of Service (DoS) attack vulnerabilities. The problems here lie for example even within the implementation of the basic SIP protocol stack, which might consume significant processing resources and as such hit early an upper performance limit. Then even a policy of reject all incoming traffic will be impossible while the decoding and identification of incoming messages would require more processing than what is available.

Of course, all the discussion above is for the real-life exploitation situation. For test-beds, most requirements relax significantly or are entirely not relevant.

- Five-nines availability – not relevant at all. A test-bed prototype would provide adequate stability if it can run stable through the targeted tests, replicating the results within measurable and tolerable performance deviations.

[11] For example 90% of full capacity available under any high stress situation.

- Availability during reconfigurations – not relevant for test-beds. Real-life exploitations must provide facilities for network re-configurations or equipment replacement without negative influences on the provided services. Yet in test-beds, this is not the case and there is no such hard requirement. Even in large-scale field trials, one can identify low/zero-traffic periods of time when such maintenance can be performed with minimal service interruptions. Also in trials it is more practical to plan down-time than to further engineer prototypes with high-availability features.
- Availability during high-stress situations – only relevant when actually doing benchmark testing. Otherwise for test-beds the costs for adding more capacity are typically under those required for engineering solid overload protection in the prototypes.

3.3.3 Security

First of all, a split of the security domains must be done, in order to provide better focus and analysis on each direction:

1. Functional Security
   a) UE Authentication
   b) UNI Security
   c) NNI Security (inter-domain)
2. Node/Operations and Management (O&M) Security
3. Attack Protection and Prevention

From this list, only the first point and its sub-points will be of relevance (1.). The rest can be safely disregarded as test-beds are closed and secure environments, normally not opened to access over public infrastructure.

The functional security is in many situation critical for realizing various close-to-reality test-bed experiments. Otherwise enabling such features might be required by the need for realistic performance evaluations, including all the processing power expended during these security operations.

Functional security – UE Authentication should be considered as the full set of security operations (not limited to authentication) which must be fulfilled in order to allow further regular functionality. Typical such operations would be client identification, authentication and key exchanges.

Functional security – UNI Security addresses the rest of mechanisms required for securing the interface between the CN and the client side devices. Most often used are the encryption and integrity protection mechanisms for subscriber signaling and data. Worth noting also are the mechanisms to prevent rogue UEs from causing damage, by enforcing sanity as well as signaling correctness and eventual isolation, on signaling as well as User-Plane (UP) data traffic.

Functional security – NNI Security would be of concern for the inter-domain interfaces. While exchanging signaling and data, each domain should first of all protect itself by: hiding internally sensitive data, securing communication with encryption and integrity protection and also ensuring that only correct and acceptable signaling and UP data enters its security domain.

3.4 Performance Benchmarking as architecture evaluation

An important point to address for new technologies is that of evaluating the performance criteria. Absolute or reference numbers are not in scope here, as obviously neither test-bed prototypes, nor the test-bed environments themselves offer truly representative platforms in regard to real-life exploitation. What is in scope here is in fact the architectural performance in factors as for example: the performance indicators' magnitude, the scalability characteristics of the architecture as a whole as well as of individual functional elements, or the relative processing costs of different operations.

To produce relevant results prototypes must be first of all designed with alignment to the concepts used in carrier-grade products. Most important in this direction of relevance would be the optimization of the prototypes following the realistic exploitation situations.

For one example, the signaling processing functions should be designed and developed following the current trends in the Information Technology (IT) industry, with multi-core parallel processing on much more uniform computing platforms or even cloud computing, taking full advantage of the advantages provided by the economies of scale and the Network Function Virtualization (NFV) trends as important methods of improving utilization and efficiency.

As a second example, implementations even as prototypes should take into account the realistic traffic models and optimize in that direction. It is a common mistake to measure performance of a system by charging it with repetitive operations, like the number of call set-ups it can successfully service in a time interval, without actually factoring in the number of subscribers which are to be used in such tests. A better test would ensure that, in this particular situation, a similarly very low ratio of call set-ups per interval per subscriber will be tested, as observed in a real-life situation, hence benchmarking the system in the realistic traffic situation for which it was designed and optimized.

For benchmarking to be relevant, prototype systems must implement as much as possible from the full specification procedures. This includes also the use of all specified communication protocols and, as much as possible, also the respective security mechanisms. Omitting or short-cutting on the protocols or security procedures here would eliminate from the benchmark their actual performance profile, producing irrelevant results.

Besides the performance evaluations as rough indicators for the architectural

feasibility, such test-bed benchmarks are used to identify potential bottlenecks and issues in the architectures, functional elements, interfaces or communication protocols. This further will present the opportunity of enhancing the prototypes with new concepts as to improve on such limitations and then contribute back to the standards with key modifications for significant performance improvements.

To round up, performance evaluations are not only useful for the purposes exposed before, but can also be used to spot faulty implementations, which normally would not be exposed at low traffic rates and minimal load levels, or without the parallel influences of multiple factors.

As a conclusion towards requirements, as derived from the arguments above, test-bed prototypes should implement as much as possible from the relevant functional procedures, interfaces and communication protocols. These implementations should follow software designs aligned with the realistic exploitation scenarios, even if the resulting components would never be used to serve in-production networks.

In the scopes of the 3GPP NGN evolving architectures, a point of relevance is that, unlike in the Internet world where standards and specifications are created only and after practical implementations exist, the telecommunication standards are written before and by design take a good measure of care to mostly standardize interfaces as part of the procedures, but not restrict on how the actual functional node implementation will be realized. This difference is introduced on purpose as to allow competing equipment manufacturers to individually optimize their products, to better position them on the market.

This telecommunication domain standardization norm then introduces an aditional discrepancy on the performance levels of initial prototypes versus the eventual commercial counterparts. As it would be unfeasible to optimize for ultimate performance, a best-effort approach would be to prototype by using state-of-the-art paradigms as standard approaches and as much as possible reference solutions and algorithms. For example, when designing the HSS function, although many of the telecommunication applications use highly proprietary and optimize DBMS solutions, a best-effort approach is to choose MySQL as a well-known reference model. This does not provide the ultimate performance, yet it is well-known, widely used and its performance well understood even in comparison to other solutions, such that optimized performance conclusions could be derived with ease from it as a reference.

3.5 IOT and Alignment to Standards

For all test-bed components interoperability is a critical aspect, in the absence of which any other practical tests would turn into time-consuming debugging and patching activities. Once this is mastered, it is easy to obtain a lean prototype, which although not implementing full standards, can be either configured, provisioned or quickly modified to interface successfully with other functional components. This could happen for the purposes of realizing a higher level more complex architec-

ture, or for simply evaluating the capabilities and characteristic of the additional component.

Practical experiences indicate that in order to ensure a good interoperability level it is very important to understand the overall principles and the chosen procedural solutions, such that during the design and implementation of prototypes those concepts will be the primary driving forces and not the implementation of individual procedures or interaction routines. The overall principles are in general more stable in time over subsequent versions of standards hence directly following the ultimate goals, while the chosen procedures and their specification are often influenced by indirect factors as implementation criteria, performance considerations, security constraints and so on. And these last influences are the ones which typically change more often.

Keeping the overall goals and principles in mind while designing the test-bed prototypes is key. One should not fall into the trap of detailing too much on the peripheral parts of the functional components, but should dedicate also a good part of the effort to realizing a clean and flexible core engine.

These principles would seem a bit counter-intuitive especially for the purposes of interoperability, which depend in the first place on the edge interfaces. Yet they are not so when considering that standards, in their evolution, would rarely change the core principles, but more often adaptations of procedures and additional operations would be operated, in order to improve the original specifications.

A good understanding of the architecture and functional principles is key. Of course, the procedural details are not to be disregarded, as after all, interoperability would start with the lower layers and proceed upwards. But it is critical that the prototype implementations would provide clean and flexible cores, to be complemented and customized in a simplistic manner.

Regarding the lower layers, special care should be taken during design and implementation of the protocol stacks. To ensure a good interoperability, the protocols should be implemented as complete as possible, such that in case of interoperability issues, resolutions would not require also changes in the protocol stacks, which are time consuming and expensive as often interoperability with previously tested systems must also be maintained over subsequent changes. Besides, many of the interoperability problems are caused by failure to align with protocol specifications, which will cause faults in the communication, even though the principle procedures are executed correctly. Following on the IP development concepts, the protocol stacks should be forgiving in what they receive and strict in what they send, while implementing always at least the mandatory features and as much of the optional ones.

An option here is also the reuse of protocol stacks and communication libraries from other projects, which would have the benefit of providing already tested and proved solutions. However, most often, such libraries would have specific requirements or software architectures, often not fitting or impeding the natural development of the new component's core functionality.

To conclude, two main requirements are identified for ensuring a good interop-

erability of the prototypes in scope. First the implementation should be based on well understood architectural principles, after careful examination of the most recent complete specifications and standards, as well as the original and evolving functional requirements and targets. Then also the protocol stacks should implement upfront as fully as possible the communication protocols.

3.6 Cost-efficient

As also exposed in the beginning of this chapter, considerations on the cost of implementation, debugging and exploitation are critical, as the scope is to realize prototypes which resemble as closely as possible real-life products, yet would fit within much smaller R&D initial budgets.

When approaching functional components, the important procedures should be identified and only then chosen for actual implementation. Normally standards specify a set of procedures and operations which are as complete as possible and cover the highest reasonable amount of possible states and event. Yet when considering test-bed purposes, many such procedures are of rare occurrence and as such very limited value.

Accordingly, a subset of the actual procedures should be targeted for prototyping, with the rest kept in an "ignore" state, silently discarded or answered with dummy responses. The reasoning is that often 80% of the most used operations can be easily implemented with costs of, for the sake of the argument, 20% of a full implementation. Then implementing the rest of 20% does not bring a sufficiently high benefit for experimentation to motivate the 80% remaining costs[12].

In the category of these non-mainstream features are often exceptions or at-the-limit procedures, which do happen in real-life exploitations, yet in the laboratory can either be avoided by simple constraints on the test parameters, or the negative effects can be overlooked in friendly-trial environments by raising awareness on the limitations to the testers.

Similarly to reductions in functionality, reductions in performance can be operated following the same reasoning and principles. A "good-enough", or sufficient performance level limits the upper cost significantly, in both hardware, software and operational regards.

Besides limiting the functional aspects of the prototypes, cost savings can be obtained from reusing existing tools. This practice is commonly found in the Internet world as for example by using Open Source projects as building blocks, where tooling is regarded as a common and shared effort. Enhancing an existing tool to specific needs is often less expensive than developing such specialized purpose components from scratch for the single scope of prototyping a specific functional component.

[12] The Pareto Principle [155] is adopted and particularized here, from the generalized formulation that 20% of the causes generate 80% of the effects.

3.7 Openness

Unlike commercial products, test-bed prototypes do not gain any advantages from acting as black-boxes. Quite to the opposite, good prototypes should on one hand offer a clear view on how they are implemented, such that on testing conclusions it should be possible to decide on changes and optimizations. On the other hand prototypes would benefit from exposing as much as possible information in a standard and open manner, unhindered by proprietary tools or security requirements.

The openness requirements start with exposed communication interfaces, which must be as much as possible aligned to public standards, or otherwise clearly specified, such that interoperability will not be negatively influenced by any barriers, artificial or intrinsic.

Then prototypes should be easy to steer by flexible reconfiguration, functionality changes and mash-ups. All parts that could and should be parametrized must enjoy this facility and all the configuration options should be documented well. Also the software architecture should be properly described and the functionality should be split into individual modules in such a flexible manner as to allow easy re-combinations in order to modify the overall system functionality in the future.

Another good test-bed trialing tool is found within the verbosity of the prototypes, which not only exposes their state, but also provide simple and clarifying explanation on events and state changes. This can be achieved most simply by integrating standard debugging features. But also by exporting internal functionality triggering points, which can significantly aid in understanding issues and providing simple trial-like resolutions, before more consistent and well-educated fixes will be available.

While a debug log is easy to read and comprehend by the original developer, this is not true for other persons, or even for the same one after a long disconnection period from the subject. Then the debugging facilities should be presented in a clear manner, through advanced facilities like graphical/web human readable interfaces or interactive consoles.

After arguing for a reduction of costs, one might argue that such advanced openness in configuration and execution through debugging facilities does not motivate itself in the limited scopes of test-beds. However, one must also consider that with fast-prototyping comes also a high probability of introducing mistakes. A systematized plan which includes also these openness concepts would have of course an initial price, yet it will repay itself by providing fast and predictable issue identification and correction capabilities.

3.8 Relevance; the “Reference” Status

An important characteristic which should be provided by a test-bed prototype is that of representing a recognized reference of functionality. One must not assume that, by the simple procedure of implementing a standard or specification, a functional reference is obtained.

If standards would be complete then their implementation could be automatized. Yet this is not the case as such exhaustive standardizations and implementations are neither feasible nor desired. A set of decisions are taken during the design and implementation phases to cover this lacunae, which will shape the prototypes through the personal view and understandings of their authors. Exposing then the result through academic dissemination and testing against and with similar parallel implementation is an important step towards obtaining validation in regard to the implemented standards.

A sort of reference level and status must be sought for the test-bed prototypes, through acceptance from the wider research communities. The benefits are that results obtained, of course upon validation of testing methods, would also be accepted as reference.

As part of this process the authors should request and listen for feedback, then take it into account and improve the models accordingly. In Open Source projects this is part of the community models, where contributions are not limited to software development in the form of patches, but also in the form of user experiences, reports of mishaps and unexpected behavior and suggestions for improvements.

The Open Source project models also exhibit an interesting behavior which can be exploited here: when several projects compete, typically the best approach will manage to capture the attention of most of the developers, which will, in a cascading loop, improve even more the project and as such attract more developers even from competing projects, which will eventually starve-out. If in the ideal case, the prototyping project manages to capture the critical mass of attention and acceptance, then it should strive to maintain it in order to maximize the outcomes. This mechanism is important here as with acceptance as a best-of-breed project, comes also an implicit "reference" status.

3.9 Summary of Requirements

As a conclusion to the present chapter, a summary of the most relevant requirements is presented below.

- Functionality
  - Minimal functionality – basic scenarios
  - Sufficient functionality for test-beds – main-stream scenarios, without the low rate of occurrence exceptions
  - Specific functionality for the targeted CN architectures
- Performance
  - Capacity – up to thousands of users
  - Latency – similar to legacy telephony networks
  - Hardware platform optimizations – to be considered for the future

  - Scalability – implicit support in the architecture

- Stability
  - High-Availability – not required
  - Availability during reconfigurations – not required mostly due to lower, less-critical traffic within test-beds
  - Overload protection – not required, more effective to rather add capacity

- Security
  - Functional security – as much as required from functional and performance representative tests
  - Node/O&M security – not required
  - Attack prevention and protection – not required

- Benchmarking
  - As complete as possible implementation of functional procedures, interfacing and communication protocols
  - Software design aligned with realistic exploitation patterns
  - Use state-of-the-art algorithms and components, with well-known performance characteristic, without additional optimizations

- Interoperability and Standards alignment
  - Comprehend well the architecture and principles and apply them as the main driving forces in the design phase
  - Have as complete as possible protocol stacks.

- Cost
  - Limit the implementation scope as to significantly reduce costs
  - Impose upper bound limits on provided performance
  - Reuse available tools by enhancing them rather than building specialized ones

- Openness
  - Standards aligned reference points
  - Open configuration and well described modular architecture
  - Advanced debugging facilities for correcting problems as well as exposing the inner workings

- Relevance

- Strive for the "reference" status as community acceptance indicator
- Join communities around the common goal

This list of requirements will be used in the further chapters of the dissertation first as guiding principles for the design and then the implementation of NGN prototypes for IMS architectures. In the later parts these are revisited as to validate the results. There the resulting prototypes will be evaluated on how well each of the individual points before have been satisfied. While the list is probably not complete, still the target is to have a comprehensively solid implementation and results.

Chapter 4

Design of the Open Source IMS Core

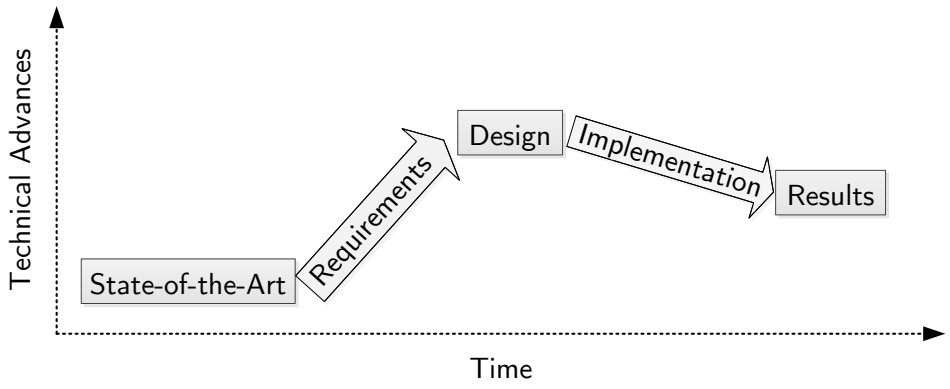


Figure 4.1: State-of-the-Art + Requirements ⇒ Design

Chapter 2 – State of the Art introduced the IMS targeted architecture. It has also provided enough historical insight into the evolution of telecommunication architectures up to the IMS point, such that the general principles should be clear here. Then Chapter 3 – Requirements for an Open Source IMS Toolkit has listed and motivated all the requirements for a CN prototype in the context of test-beds.

The present chapter will draw the blueprint for building an IMS CN prototype. Its design is a combination of the IMS standards, as a reference for how carrier-

grade products would be implemented, and the limitations offered and imposed by reducing the exploitation scope to test-bed environments.

The process of deriving this design follows logically from analysis of the State-of-the-Art tools in regard to the established Requirements. Several paths are individually analyzed in order to find the one with the best benefits, while also maintaining a reasonable costs limit for the users of the software.

4.1 Design Matrix: State-of-the-Art(Tools) versus Requirements

To formalize the evaluation, a design matrix is presented here. This has the form of a table with the rows as the short list from the previously defined requirements. The columns are represented by a series of different approaches at implementing the targeted IMS CN functional components, as major design choices. At the intersection of each column and row indicators are provided for how well or bad the respective row requirement can be satisfied by the respective column approach.

Choosing metrics for comparison would be beneficial for providing the scale of the advantages, respectively penalties, that each approach would entail. Unfortunately each of the requirements has individual metrics, which means that they would be impossible to combine without a lengthy and arguable definition of relative weights. Considering this fact, a simpler method is chosen for the analysis, where each matrix indicator will possibly have one of the following 3 values:

$\surd$ – indicating a good fulfillment of the characteristic;

Ø – indicating a neutral or non applicable value as either arguable or hard to measure;

$\times$ – indicating a bad fulfillment of the characteristics.

	Tools		
Requirements	Approach A	Approach B	Approach C
Capacity	$\times$	$\surd$	$\surd$
Scalability	$\times$	Ø	$\surd$
Openness	Ø	$\surd$	$\surd$
...	...	...	...
Combined Results	-2	$+2$	$+3$

Table 4.1: Example of the Design Matrix, to Expose the Design Methodology

At the end of the matrix a row will simply summarize the number of negative, neutral, respectively positive values, giving each a value without weight of 1. The

implementation path will be represented by the proposed design choice with the highest composed score.

The analysis and motivation for each value in the design matrix will be exposed and discussed in the next section. The resulting matrix is to be found after this motivation, as a conclusion to the design chapter. In order to clarify on the used methodology, an example of the design matrix is provided in advance in Table 4.1.

4.2 Multiple Choice Approach Model

The design methodology also includes a comparative analysis of several initially proposed approaches at implementing the prototypes in scope. Multiple options arise from different choices available for implementation in software, as well as from various platforms to be used as accelerating starting points. This section aims to constitute an A-B-C comparative choice model: upon analysis of each major option in regard to the previously established requirements, the best will be chosen to be implemented and trialed.

4.2.1 General Considerations for Limiting the Design Choice Space

First to consider would be the targeted hardware and software platforms.

Following the trends towards uniformity of computing platforms, as well as the ones for virtualization and cloud processing, a logical choice for hardware is represented by x86 platforms. Other competing architectures like MIPS, PowerPC or SPARC, as extensively used in the past for carrier-grade equipment in the IN architectures, would be potential candidates here too. Yet there are no compelling arguments to still motivate them today over Intel® 80x86 Instruction Set Architecture (x86). Quite to the opposite, the x86 platforms are broadly and inexpensively available, making them a prime choice. Without support for x86, artificial barriers would be raised.

In regard to Operating Systems, the prime choice as resulting from the Open Source and re-usability requirements would be the GNU/Linux platforms and distributions. To further strengthen the choice, its Uniplexed Information and Computing System (UNIX) heritage makes it a prime option for networking applications, as such platforms are at the core of today's Internet equipment and services. Although demanding in the matter of predictable response time models and latency, the SIP and IMS procedures would not require Real-Time facilities or such advanced computing platform features.

Both choices above in hardware respectively operating system platforms are not exclusive ones. In fact it is relatively easy to port a software packet to a different platform, or to use a different UNIX-like operating system. The prerequisite for such flexibility is represented by the use of standard software development practices and tools, as well as the avoidance of using proprietary features or optimizations. In practice many applications are easily cross-compiled for supporting multiple hardware architectures and GNU/Linux applications are easy to use also under other

operating systems like OS X[1]. As a notable exception, especially as most desktop computers today feature Microsoft® Windows™ environments, the significantly different architecture, especially in the networking domain, does not motivate a critical targeting of this operating systems family. As alternative for the most used desktop operating system, one could also employ virtualization on top it or other operating systems not directly targeted, to provide a virtual GNU/Linux environment with relatively low overhead.

Implementing from scratch an IMS CN system would most likely offer best results, as all the requirements can be properly addressed. However, starting from zero has much higher costs as also the base functionality would have to be implemented, as well as having to invest more efforts into the stabilization, debugging and validation of the new framework. An existing starting platform, although imposing limitations and most likely also an already defined underlying software architecture, has then the advantage of providing a significant level of confidence in at least the set of fundamental functionality, which would come mostly pre-implemented and pre-tested.

An unfortunate disadvantage in the telecommunication industry is caused by a slow adaptation of the academic domain to the latest advances and changes in the field. While for example the Internet world enjoys a large base of specialists well educated on the cutting-edge topics, which also have a solid background in Computer Science and Software Development, the vast majority of telecommunication college graduates have a similarly good experience in the Electrical Engineering, yet very limited Software Development skills. With this in mind a reasonable choice for the development platform on NGN component would be a simpler one, which would not necessarily require the high skills of very-specialized software developers to understand and to modify. The `Java` language and concepts are then well suited for these purposes, as they have much easier learning curve than the specialized Internet networking industry norms of `C` and `C++` developed functionality.

`Java` offers the added benefit of Rapid Application Development (RAD). The language itself is geared towards facile prototyping by allowing simple Object Oriented Programming (OOP) paradigms, managed virtual machine environments and extensive debugging and profiling capabilities built-in at its core. With these capabilities, `Java` application life cycles are often shorter, less demanding and less expensive than their `C++` equivalent counterparts.

However, `Java` platforms have also their limitations. The programs are not compiled and optimized for the target operating and hardware platforms, but the resulting binary code is executed by virtual machines. This level of abstraction adds significant penalties for latency, memory consumption and processing efficiency. Only very specialized Java Virtual Machines (JVMs) are capable of keeping these parameters under control, resulting in hardly predictable or good level of performance.

Then as a starting platform, the `C`-based SIP Express Router (SER) platform

[1] Also an UNIX operating system.

cannot be ignored. It is already a well-recognized standard for performance, as well as flexibility and represents a solid SIP Proxy/Server solution.

4.2.2 Proposal A - Java-based RAD

As mentioned before, a Java implementation should be considered for its friendliness towards not-so-experienced developers, but also as such an implementation could be driven at a fast pace. Libraries of functionality are available for almost any purpose and particularly here there are already reference SIP protocol implementations available (e.g. National Institute of Standards and Technology (NIST) JAIN SIP reference implementation [156]).

Such an implementation would be relatively easy to drive towards achieving a solid functionality set, well aligned with the standards.

Unfortunately as far as performance is concerned, there would be serious penalties caused by the lower efficiency of the JVM executing the Java binaries. Capacity wise each individual SIP session, dialog or registration will consume more than the optimal memory allocations, as Java is typically more wasteful in the memory management. Further limitations on the total amount of memory usable by an individual JVM would require the fragmentation of traffic processing and use of multiple instances to fully load powerful systems, which in turn again adds significant overhead and inefficiencies in processing costs.

The memory considerations above are made worse by the typical JVM memory garbage collection routines, which upon high memory usage would cause temporary interruptions of normal processing, causing as such erratic latency values. There are of course professional and highly optimized JVMs solutions which would avoid these issues, yet their costs and added complexity would impose additional artificial barriers.

To complete the performance considerations, low-level optimizations are hardly possible and overall internal component scalability would have a unsuitable curve. Architectural scalability is still feasible, by load balancing between multiple running instances, yet again still individual instance would have the significant JVM overhead to consider.

With the above functional and performance considerations, the benchmarking requirements are only satisfied in regard to functional completeness. In the other aspects realistic or relevant performance evaluations are most likely impossible to derive as even the ability to reproduce results with a good confidence interval could be threaten by erratic latency issues.

The ease of development and the extensive debug capabilities would allow for a stable environment as issues are easy to identify, isolate and resolve.

From the costs perspective, even as most of the features need to be developed almost from scratch, the low-entry barriers would keep implementation costs low for the standard functionality. Unfortunately achieving performance will be quite difficult and expensive in comparison. Re-usability should be average as there are only several protocol stacks already available and no functional SIP proxy modules

to be re-used by the project. It would be conceivable that, at least on the more critical SIP signaling levels, better implementations will most likely be based on lower-level C/C++ platforms for performance considerations, which would reduce the attractivity and re-usability of the Java-based developments here.

This Java approach will most likely fare very well on the openness requirements, due to the low barriers in development complexity, allowing the implementers and future users to concentrate rather on the configuration and use of IMS functionality, rather than wasting time on tedious programming details. However, these gains could be easily lost as the model might be considered too far from a full implementation to be relevant.

4.2.3 Proposal B - Script Controlled SER

The second proposal seeks on reducing of software development costs as much as possible. An orchestrations of operations would be theoretically possible from the SER routing script, used as the definition of signaling processing procedures.

SER and its already existing modules implement many operations which could then be configured to be performed as indicated in the IMS standards. Unfortunately without further developments, all these would be limited to an orchestration of existing SIP concepts and related protocols as used in VoIP deployments and defined in the IETF implemented specifications. The RAD characteristic is also here achieved as well as the ease-of-use for non-specialists.

Functionality wise this solution is unfortunately quite complex as the facilities provided by the scripting configuration are of course limited to the maximum functional space implemented by the existing SER modules. This would be then rather minimal and most likely not sufficient when considering alignments to the standards.

From the performance perspective, good figures can be potentially obtained. Yet again this would not be for sure close to the best which could be achieved as the added abstraction level by scripting of course adds at least a minimal overhead to processing, while also taking only limited advantage of already implemented general optimizations in the platform.

Benchmarking such an incomplete functionality system would most likely be plagued by relevance issues. The reasoning here is that it could be very easily argued that performance is not representative if the majority of procedures are not entirely followed. On the other hand, the already established SER as a SIP Proxy performance milestone would at least offer a rough insight into IMS processing requirements and performance characteristics.

Regarding stability this approach would fare very well, as the same base implementation is used already with success in carrier-grade environments.

The functional security is though nearly impossible to achieve without further development of existing and new modules, as the IMS mandatory procedures were mostly new and demanding in comparison with the state-of-the-art in the VoIP deployments.

Considering the implementation costs, these would be average. While on one

side the solution would fare very well on re-usability, the complexity of orchestrating functionality through general purpose scripting is not to be underestimated, especially as certain operations could potentially explode in complications for incrementally low functional benefits.

Regarding openness, the configuration capabilities would be very good, yet the lack of full features will introduce a negative acceptance criteria as the implementation would be in-between a proprietary SIP carrier grade platform and the similar yet standardized IMS platform. Still, a large part of the existing SER community members might consider it as a pragmatic compromise for providing IMS-like NGN capabilities at reasonable costs.

4.2.4 Proposal C - IMS modules for SER and Script Control

The third choice is an enhancement of the second one. Specialized IMS modules are to be developed, which would fill the missing features from the previous option with solid solutions. The configuration script based control should be maintained, as to keep the respective flexibility and the associated ease-of-use characteristics.

Development of additional modules requires though specialist understanding of `C` software development. The SER architecture is also not very well documented or easy to approach, further raising the development complexity and the entry-level for such developments.

Yet this approach would provide a good coverage of functionality as then all the protocol stacks as well as procedures required can be developed by directly following the standards and eliminating as much of the existing functional shortcomings. The functional requirements will then be well satisfied. Yet due to the more complex development, this will be slightly below the first `Java` approach, which could achieve more in this domain at comparable implementation effort levels.

Performance wise the system can be implemented and optimized close to conventional implementations, by following the same carrier-grade trialed model of the existing SER modules. By combining the best of performance in the low-level modules implementation with the flexibility and limited functionality exported to the script, low processing latencies are expected, with facile optimizations and good scalability characteristics.

With a good performance, but most importantly with a relevant and sound implementation, benchmarking requirements will be here satisfied at their best. The performance could be directly compared to future real-life exploitations, with slight limitations of course in relevance due to not 100% complete and aligned functionality levels.

Stability wise, the challenges should not be underestimated. The new modules, even if reusing as much as possible tested and proved code, would contain software bugs which are harder to identify and eliminate. Without a protection offered by a managed environment as JVMs, the gained run-time performance through elimination of overheads, would directly translate to increased costs for bringing the solution to a usable state.

The current approach would also be the most expensive one as a specialist implementation is required. However, while the costs for the initially usable version would be high, obtaining performance from the solution is then much simplified and in effect built-in.

From the standards alignment perspective the results are expected to be satisfactory to good, with the only limitation being here the implementation costs, which would most likely limit the coverage of at-the-limit rare cases.

The configuration aspects here are expected to be even better than the ones in the SER scripting solution, as one can take full advantage of the routing scripts, while also minimizing as hiding its complexity within the newly developed modules.

It is also conceivable that the present solution would be accepted by the IMS research community based on the two very important and strong traits: relevant functionality implemented aligned closely to the standards, coupled with relevant performance provided by the SER solid core.

In conclusion, it must be reiterated that the present approach is highly complex in its implementation phase, which drives the costs up and as such threatens its success. Nevertheless, as the performance and stability of SER has been proved already as suitable for the carrier-grade environment, the benefits would be major as this option has the bases of an already successful implementation and would potentially allow for a future transition into real-life use.

4.3 Design Matrix and Conclusions

Table 4.2 summarizes the individual discussions on the 3 proposed design alternatives. While it can be argued that the analysis is biased by the author's experience, the approach of using SER as a stable performance base, developing extra modules around and then still taking advantage of the flexibility provided by the configuration script, is the best approach to follow here. Still, each non-optimal solution should be individually eliminated before a final conclusion could be made.

The B approach of avoiding new modules and just riding on the existing functionality would be potentially a faster path towards the initial goals. Yet the newly introduced IMS concepts and mandatory protocol stacks seriously threat the functionality of the final result, which will hinder the relevance of the resulted platform as incomplete. Improving the results after a certain stage would in fact start to be increasingly difficult because of increasing complexity, with simplification attempts driving it then towards the C approach. Then the costs would simply be summarized between the simple B and the complete C approach. For this reasons, the B approach is discarded at this point in favor of the C approach.

The rest of the discussion will have to be split in two parts, as the intrinsic characteristics of the functional components differ highly: first the CSCFs and then the HSS.

The CSCFs provide SIP Proxy feature sets with clear functional separations, as to provide a good scalability of each type of CSCF, as well as well-recognized

Requirements	Tools		
	A - `Java` RAD	B - SER scripting	C - SER modules & scripting
Functionality			
- minimal	√	√	√
- sufficient	√	×	√
- specific	√	×	√
Performance			
- capacity	×	√	√
- latency	×	Ø	√
- hw./platform optimizations	×	Ø	√
- scalability	×	×	√
Benchmarking			
- functional completeness	√	×	Ø
- realistic performance	×	Ø	√
- relevant	×	√	√
Stability	√	√	×
Security			
- functional	√	×	√
Costs			
- implementation life-cycle	√	Ø	×
- for performance	×	Ø	√
- re-usability	Ø	√	√
Openness			
- standards aligned	√	×	√
- configuration	√	√	√
- debugging	√	Ø	Ø
Relevance			
- status/acceptance	×	Ø	√
- community	Ø	√	√
Combined Results	+2	+1	+14

Table 4.2: The Design Matrix, for the A-B-C Approaches

reference points between them, encouraging good interoperability.

Here the SER platform in approach C represents a better option than A as the SIP protocol and the base operations are already available, trialed and providing very good performance. Approach A, although better on the long term on the functionality aspect, would for sure neither have the expected performance, nor benefit from the already existing acceptance that SER enjoys.

Secondly for the HSS function, Approach C is an unnatural choice. Of course SER would satisfy the basic requirement of providing DBMS access, as well as shar-

ing the new Diameter protocol stack to be implemented. Yet the SER concepts are deeply anchored in the requirements for performance and flexibility in SIP signaling processing, while the HSS would in fact feature no such SIP interfaces or signaling processing.

A proprietary solution could also be envisioned, where the HSS functionality would be directly incorporated in the S-CSCF one, with direct access to the back-end DBMS. Yet this would impose the limitation of the severely un-aligned to the IMS standards and concepts, as well as negating the architectural security and scalability characteristics.

A standalone HSS design would be then the better path to follow, decoupled from the SER platform. While still a `C/C++` platform could be used here, potentially reusing as much as possible from the SER platform, still this would represent a new platform, together with all the associated high additional costs and limited satisfaction of requirements.

Approach A then comes back into picture as an elegant solution. The development is accelerated significantly through reuse of existing libraries as well as the extensive debugging capabilities and ease of development provided by the `Java` environment. Not to forget is also the good standards alignment which can be also provided with ease.

Performance wise the `Java` lack of pace and overhead penalties are to be taken into consideration. However, the HSS procedures call in fact for a rather light Diameter based orchestrations of operations on data stored most likely in a DBMS back-end. Then the performance limiting factor here is rather in this back-end and not in the light HSS orchestration layer. The DBMS is external to the planned HSS `Java` application, as provided for example by MySQL.

To conclude on the HSS design, approach A is the most compelling option with significant advantages in functionality over a `C/C++` approach and with performance limitations largely offset by the workhorse being in fact the back-end external to the HSS, the DBMS.

The Design Matrix and the reasoning on the 3 approaches analyzed lead to the following design conclusions for the implementation of an IMS prototype for test-beds:

1. The CSCFs are to be implemented as module extensions to the SER platform, with procedures and operations orchestrated by the configuration routing script.

2. The HSS is to be implemented as a standalone `Java` application, using an external database back-end.

Chapter 5

Specification of an IMS CN Prototype Implementation

The present chapter seeks to expose the implementation engineering phase of the OpenIMSCore project. It starts with a specification of the main functional features of an IMS CN prototype. These constitute blueprints for the functional features which are to be provided by the resulting prototype.

Before implementation aspects are to be exposed, a summary of the IMS functional features needs to be made. The purpose is to establish a set of principal targets to be realized. Here the IMS main building blocks are taken from the state-of-the-art descriptions, then adapted and reduced through requirements and the targeted constraints.

Only the principal functions are presented. Many sub-features or additional functional nodes are part of the implementation, yet they are not of critical importance or sufficient novelty to motivate an in depth specification and description here.

Each principal functionality topic is first described from an architectural and logical point of view. Processing, interactions and messaging flows follow, adding technical insight.

5.1 IMS Mobility Support

In the telecommunication domain providing "mobility" represents the technical capability of following a moving subscriber by providing (potentially uninterrupted) service even as the physical location and connectivity parameters of the subscriber's device changes, the terminal itself is substituted or the communication medium and the associated set of technologies change.

A critical concept here is the identity of the subscriber, or as seen from a technical perspective, the anchoring and identification concept, which identifies the individual communication endpoint provided with the mobility feature. The operational concept then deals with temporary mappings and following through the subscriber's reachable communication points towards his identity, such that at all times data can be delivered and received to and from the respective subscriber, without the application users at the other end of the communication path needing to be all continuously updated on the changes in data paths.

Historically this capability was provided first by human switch-board operators, which manually made connection based on their own knowledge on mappings between persons and physical locations of telephony devices. With automation the process greatly improved in precision, costs and speed. Yet the telephone numbers

and dial plans replaced the personal identification with an identity concept which was more prone to be processed by automatons. Telephone books were printed, as directories to be used by callers to find the their intended communication recipient.

Digitization and the following advances introduced by data processing systems have allowed operators to abstract from the location of the telephones and their associated numbers, introducing number portability. Mobile communication has further advanced the concepts by placing the subscriber identities physically in the hands of the users, through Subscriber Identity Module (SIM) cards, which are usable from any such compatible client access device. More critical though, mobile networks, although not from their beginnings, have introduced the physical mobility capabilities and advanced them as to provide seamless connectivity, even as the subscribers change their physical location.

Current advances follow on even more pervasive mobility, through cross access technology, by eliminating the boundaries between fixed, mobile or long range AN technologies. The telephone number is set to be replaced with more personal identity tokens, after the highly successful models from the Internet world[1]. This will completely eliminate the needs for operator-wide telephone directories, hence automating and improving even further mobility.

Mobility in IMS is then represented by the following functional capabilities of the platform:

a) to first identify subscribers in a secure manner[2];

b) to maintain mappings between their globally reachable identities and temporary network addresses indicating the temporary location of the respective subscriber;

c) to resolve identities into network addresses and to route and deliver data over various access networks, allowing information to be addressed, posted and delivered to the respective subscribers;

d) to allow re-mapping of subscriber identities to new network addresses seamlessly, with minimal or without service interruptions.

Each of these individual features will be detailed in the next sub-sections.

5.1.1 Subscriber Identification in IMS

In technical terms identities in IMS are represented through a series of identity concepts with similar representation, but differentiating between the identification purposes.

Top most, the logical identity of subscribers is referred to as IMS Public User Identity (IMPU). Represented as a Uniform Resource Identifier (URI), the most common representation would be that of SIP URI [100][3]. For legacy and technology

[1] E.g. the e-mail addressing scheme

[2] To prevent impersonation attacks subscribers must be authenticated and the communication must be secured.

[3] E.g. `sip:John.Doe@companyx.com`

migration purposes an identity can also be represented by a telephone number, in the form of TEL URI[4] [101]. Additional schemes can be of course envisioned and specified in the future, like for example the URN[5] [102] [6].

The IMPU itself does not transport mobility information, other than a mapping with the subscriber's domain.

Then identification concepts for services can use the same schemes, as after all, such services behave similar to actual subscribers in the communication procedures, with the notable exception of course of not really requiring services like mobility or authentication, as being provided from rather fix data-centers and usually trusted service platforms. This identity type is referred in IMS as Public Service Identity (PSI).

An important and defining characteristic of these public identities is that of them being globally routable. This means that when presented to a communication node, information about at least the next hop to be followed towards reaching the respective subscriber or service can be obtained. This concept then covers more than, for example, an IP network address as that would only be relevant to a certain network (or even location considering IPv4 NAT mechanisms). In case of SIP URI the Fully Qualified Domain Name (FQDN) part of the address is used as direct input for a DNS query mechanism. In case of TEL URI a DNS based mechanism, E.164 (Telephone) Number Mapping (ENUM) [157], for translating from numbering plans to operator FQDNs has been created and used on the Internet.

The next IMS identity concept of interest is the temporary network address of an IMS UE device. While the signaling protocol used is SIP, also here the respective concepts are used in the form of the SIP `Contact` information. The contained information can similarly be represented flexibly in many formats, the most used form being that of a SIP URI containing the IP address and port of the IMS client application. As the mobile devices moves between ANs, the IP address potentially changes, such that updates need to be provided[7]. Additional parameters which would help for purposes like identification or routing can be attached and transparently transported, saved and used.

The mobility procedures are obviously sensitive such that attacks must be prevented. Authentication is then an important and mandatory mechanism here. This is largely approached in a following chapter on security. Of interest at this point is

[4] E.g. `tel:+1-555-123-4567`

[5] E.g. `urn:service:sos.fire`

[6] This example for identification of a fire-fighting service number is interesting in the context of mobility, yet with a slightly different approach than the principal one in subscriber mobility. Here the feature of interest would be to provide to the caller the network address of the physically closer available service of this type to the current location of the caller itself, considering of course the caller dynamic mobility. Hence here the originating party location information is also used as a parameter in providing the terminating party network address.

[7] A notable case is that of current mobile networks. These provide in fact mobility at the IP address level in most situations of physical mobility through horizontal handovers (inter-cell), such that the actual IP address does not need to change. Still mobility is not (universally) covered in the case of vertical hand-overs (inter access technologies) or client device swapping.

the IMS Private User Identity (IMPI) which provides to the network the identity of the SIM to be used for authentication purposes.

Regular SIP VoIP exploitations use the same globally routable identifiers for both public subscriber identification as well as authentication purposes. There are though at least 2 important reasons for which a different identity should be used. First of all, the security of the authentication procedure is improved if the respective subscriber identification token is not necessarily public knowledge. Then the authentication mechanism is an operator based one, rather than a subscriber one. Of course subscriber password based authentication can be applied, yet this is much more insecure compared to the authentication token based approach, which can physically guarantee the safety of the authentication credentials, by never allowing read operations, but only generation and one-time use of authentication vectors further protected by mutual authentication[8]. Going back to the operator internal identities versus subscriber ones, especially in the mobile domain it has become common practice to pre-deliver SIM cards to vending points and customers, and only later, upon subscriber activation, to associate public identities to them, as an effective cost and time saving measure.

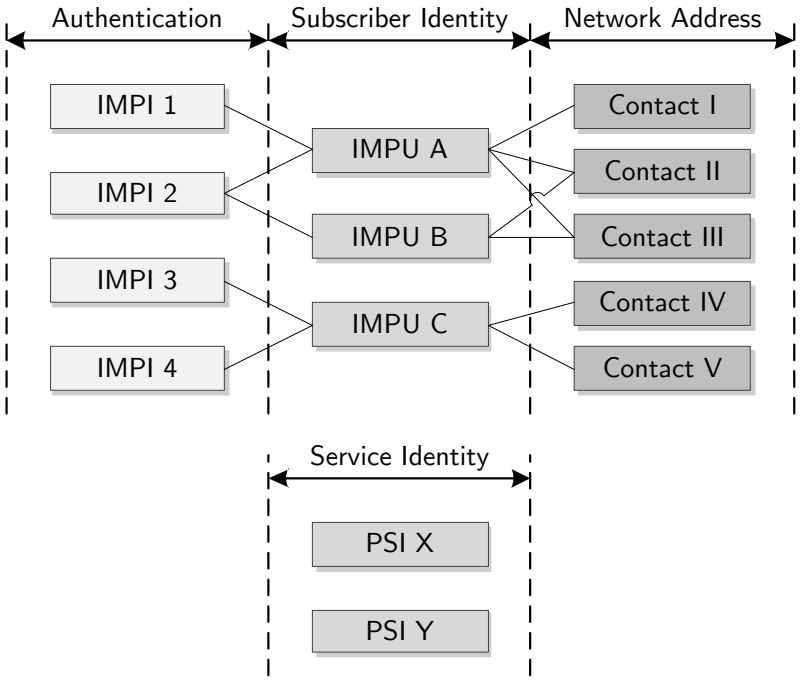


Figure 5.1: The Various IMS Identities and their Mappings

[8]Not only does the network authenticate the subscriber, but also the subscriber authenticates the network.

The IMPU identities are then central in IMS. These are administratively mapped to IMPI identities, allowing for strong and also flexible authentication procedures for globally routable IMPUs. Temporary *Contact* addresses are refreshed by the client devices and saved in registrars (provided by the respective home networks serving the subscribers) as associated to the IMPUs.

The signaling route is established first by following the path towards the home domain (or operator) based on information from the globally routable IMPU address. From there a last leg route is realized by using the last associated *Contact* information for the respective IMPU address, along with other associative signaling routing information.

Regarding the mapping relations between identities, the following points should be observed:

- one IMPI can be associated with multiple IMPUs for authentication purposes, on an administrative basis ⇒ multiple globally routable identities can be used from a single client device;
- one IMPU can be associated with multiple IMPIs, again on an administrative basis ⇒ multiple distinct client devices can authenticate and subsequently present and share the same globally routable identity;
- one IMPU can be associated with multiple *Contacts* ⇒ a single subscriber identity can be serviced by multiple client devices in parallel, requiring additional mechanisms for "forking" an initial single session into multiple ones, following different network paths and terminating in potentially different client devices;
- one *Contact* can be associated with multiple IMPUs ⇒ a single network address and port can potentially serve as a signaling point for multiple identities[9].

For the particular case of the PSIs, as these normally require neither security nor mobility operations, they do not need to be associated with real IMPI nor *Contact* identities. For uniformity reasons though, associations with virtual identities could still be made.

As a summary for the exposure of the various identities used in IMS, it can be concluded that there are 3 types based on their purpose and use model: global, authentication and network addresses. These are associated in many-to-many relations and accordingly associative lists would need to be maintained in the processing functions. Identification of services is simplified to global addresses.

5.1.2 Mapping Subscriber Identities to Network Addresses

The mapping of network addresses to globally routable identities contains multiple operations. These compose an overall scenario which allows for secure establishment

[9] As an implication here, then on originating signaling the network cannot make unequivocal conclusions on the originating identity by using only the originating network address as an indicator, but the client device should explicitly select one such identity on each originating message

of such relations, reliable maintenance as well as additional capabilities to inform other functional nodes and services upon the current location and status of a certain subscriber.

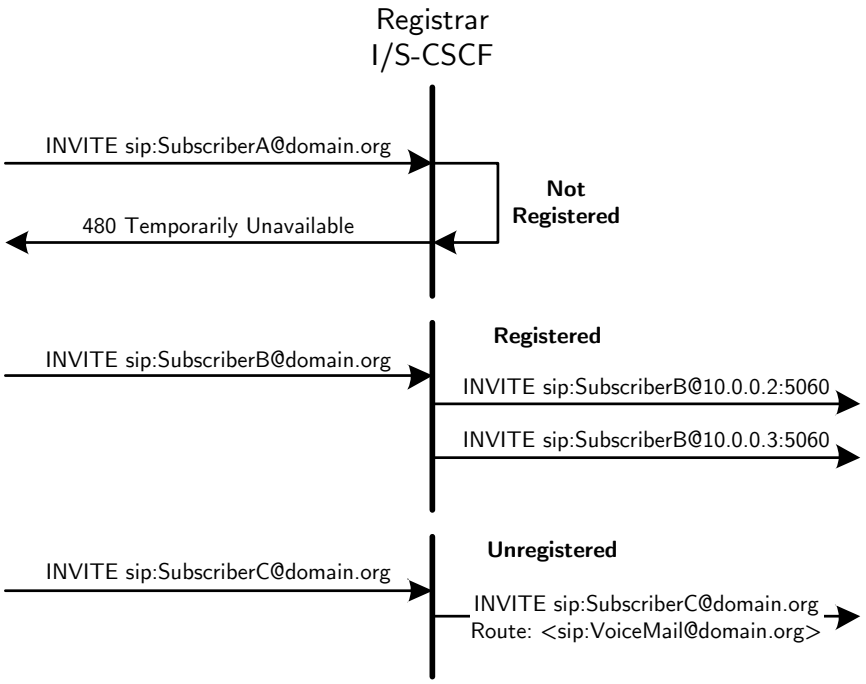


Figure 5.2: Exemplification of Signaling Flows for the 3 IMS Registration Statuses

As the status of the user has been mentioned, it is important to understand the logical model behind this. There are 3 possible situations (or states) for each subscriber identity:

1. *Not-Registered* - the IMPU is not active at the moment on any client device and as such signaling cannot be delivered. There are no mappings of IMPU to *Contact* network addresses.

2. *Registered* - the IMPU has one or more active client devices at the moment, which have registered their network addresses in the IMS registrar as routing destinations for terminating signaling.

3. *Unregistered* - the IMPU is not active at the moment on any device, yet there are services active for this subscriber identity which might be able to terminate

signaling for this subscriber[10]. The IMPU/PSI would be active in the domain registrars at the S-CSCFs, yet if the signaling is not entirely processed by ASs, its delivery will fail as there are no valid mappings to *Contact* network addresses.

Also it is important to note that as services do not require mobility functionality to support their operations, they do not need to be registered. As they will be in a reachable state still, the respective PSI identities will be in practical terms in a permanent *Unregistered* state. The procedures to place them into this state are the same as for normal IMPUs, where upon location requests the Service Profiles are checked for relevant indicators and signaling is redirected from the S-CSCFs to the ASs for processing.

5.1.2.1 Registering Contacts to IMPUs

The first step in the IMS mechanism for routing signaling while providing mobility services for IMS clients is the association of temporary network addresses, converging network location and routing information, with the globally routable identities used for addressing subscribers.

The main part of the procedure is not new, being in fact used in quite a similar manner in SIP based VoIP exploitations. Referred to as registration, the functional procedure entails the use of a SIP transaction, with the method `REGISTER`. The operation is originated by the client device upon initialization or any subsequent network interface changes or UE activity which entail a change in the network address and its additionally advertised parameters. The UE includes in this message `Contact` headers with the client device's local network addresses and parameters, as well as the IMPU which it wishes to associate with those addresses.

As all signaling in IMS, the `REGISTER` request will be sent over the Gm interface through a P-CSCF which was either discovered during the network attachment procedure, provisioned in the IMS SIM (ISIM) along the IMPU and other information, or otherwise configured or provisioned through other mechanisms.

The SIP request will have the *Request-URI* containing just the FQDN part of the URI[11]. The P-CSCF will employ DNS procedures to perform the FQDN based routing, respectively find the next hop network address, which services the destination network domain. The DNS system is to be provisioned with the correct information based on public and inter-operator agreements as it is the practice today with public domains on the Internet, respectively mobile operator's roaming agreements.

As to keep the procedure universal, the client device will be assumed to be roaming and the P-CSCF belonging to the respective visited network. Whether the user is in fact roaming or not does not impact the procedure at this point, other than in regard to underlying network security, considering procedures between the

[10] E.g. a voice mail service which would answer calls and record messages.
[11] If the URI contains one, otherwise the FQDN value as locally provisioned or derived.

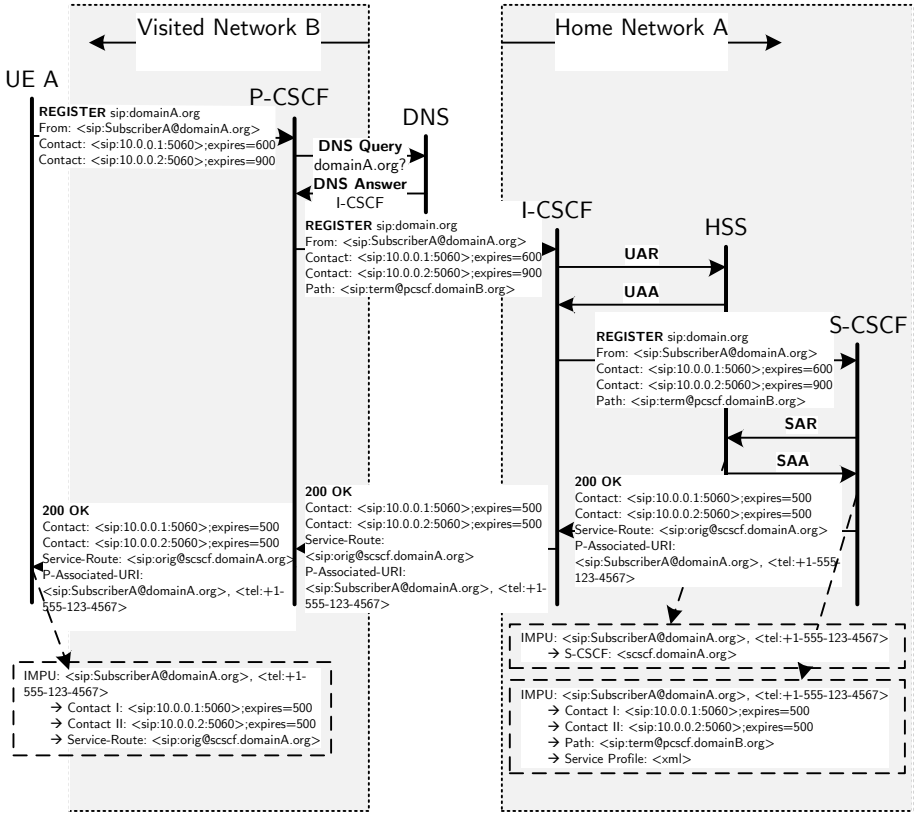


Figure 5.3: The Registration Procedure in IMS

visited network that the client device is currently using for network access and the home network which hosts and services the respective subscriber's applications.

The entry point in the home network is represented by an I-CSCF, which will be directly provisioned at the domain's advertised network addresses [12]. The I-CSCF as its name indicates, has the role of resolving the next routing decision to be made, that of locating the S-CSCF which either is currently assigned and serving the IMPU to be registered, or one that would be a good potential candidate for this.

The I-CSCF does not in fact hold the required information, but will enquire further the HSS, over the Cx Diameter Interface. The Diameter User-Authorization-Request (UAR) message is used to send to the HSS the IMPU, IMPI and the Visited Network identity[13]. The HSS will proceed to execute the *Cx-Query* and *Cx-Select-*

[12] For NNI security reasons the signaling might in fact proxy through additional security gateways, on either or both the visited network and home network sides. This intermediary point is not of critical functional concern here due to limitations in requirements.

[13] As filled by the P-CSCF in the `P-Visited-Network` header.

Pull functional level operations.

At this point the HSS will proceed to an early check if the IMPI/IMPU associations is a valid one, as well as whether the respective IMPU subscriber is allowed to roam in the currently used Visited Network[14], responding with failure error codes in case these prerequisites are not fulfilled

The HSS maintains the overall domain mappings between the IMPU identities and the S-CSCF FQDN where they are currently served, but not the mappings to the *Contact* network addresses, which have a more ephemeral status. In case one S-CSCF address is found then this is returned to the I-CSCF as an indication for where the signaling should be forwarded next. Otherwise, as in case of an initial registration, the HSS will either use internal mechanisms to select an S-CSCF instance, or will delay this decision to be made at and with I-CSCF internal mechanisms. Then a lists of mandatory and optional generic capability indications, relevant to this particular subscriber, to be used as input for the S-CSCF selection mechanism, could be optionally downloaded with the Diameter User-Authorization-Answer (UAA) message to the I-CSCF.

Next the I-CSCF will forward over the Mw interface the SIP `REGISTER` message to the indicated or selected S-CSCF. In case this is not responsive and there are still other choices which could be used, the I-CSCF will attempt to resend the `REGISTER` to the next S-CSCF.

The S-CSCF will continue the SIP message processing by first checking if the message has been received over a secure path. In case the path was not yet secured or the IMPU has not been already authenticated, the S-CSCF starts an authentication procedure. The authentication is not exposed here, but will be discussed within the security related scenarios.

The S-CSCF will then proceed to save and map the `Contact` header information in the local registrar. Diameter Server-Assignment-Request (SAR) is sent over the Cx interface to the HSS, indicating to the global subscriber database that the IMPU is to be associated with the procedure originating S-CSCF, as part of the *Cx-Put* functional level operation. The Server-Assignment-Answer (SAA) response will tell whether the association is acceptable and will provide the *Cx-Pull* function of downloading the Subscriber Service Profile to the S-CSCF. This Subscriber Profile contains complete information about the subscriber identities as well as the associated signaling filters to be verified and used as triggers for AS involvement in the subscriber's originated and terminated signaling.

Next the S-CSCF will answer to the `REGISTER` request, providing to the UE the result of the operation as well as the network routing policy (discussed in the following section). The signaling follows of course the route back through the I-CSCF, respectively P-CSCF, ensuring that also these entities are informed on the results.

To optimize the procedures, identities are grouped within the Subscriber Profile

[14]Roaming agreements are established between operators in order to provide service correlation as for example charging. These agreements are also employed to establish chains of trust with secure and reliable partners. Without such a guarantee, end-to-end IMS security features would not be feasible.

into sets of implicitly registered IMPUs. When responding to the UE for `REGISTER`, the S-CSCF will include this information in the `P-Associated-URI` [158] header, indicating that also the additional IMPUs have been implicitly registered together with the principal one, hence subsequent procedures for those are no longer necessary.

A special mechanism for polling the registration status is provided in the form of a `REGISTER` request which does not contain any `Contact` headers. The S-CSCF will interpret this as no changes to be applied, yet the response will be filled with all the currently stored mappings from the registrar. The purpose here is to provide a mechanism for the UE to manage invalid network addresses, which otherwise might have been left in hang-states after a UE or network failure.

In case an authentication procedure is required and triggered, the S-CSCF will respond with a `401 Unauthorized` failure to the first transaction. Then upon a new transaction containing additional valid authentication information, the S-CSCF will proceed and apply mapping and associated procedures.

5.1.2.2 Registration Validity, Expiration and Status Notifications

The registration mechanism includes a temporal validity parameter. This is used by the UE to indicate for each *Contact* network address a validity period, either as part of individual `Contact` header parameters, or generally for all contacts communicated, in the `Expires` header. In case no refreshes are received by the registrar within this interval, the mappings will be deleted at the expiration point. The S-CSCF has the opportunity to reject the validity period as being too short, or to indicate a shorter value to be applied, if the validity period is too long, both based on operator policy.

The mechanism so far, although used in many SIP exploitations has a major disadvantage. While the UE can at any time modify the registration status, the operation is only triggered from the UE side. Whenever events occur in the network which change for example the expiration of the registration, the UE will not be informed and potentially experience a black-out of incoming signaling, although its local status will indicate an active state.

Then it is mandatory for the UE to make use of the SIP event package for registration [159], immediately after a successful initial registration operation. A `SUBSCRIBE` request is to be sent by the UE with the `Event` header set to `reg`, subscribing practically the client device to its own identity registration status at the S-CSCF registrar. The S-CSCF will then provide first a full list of the active mappings, followed by incremental and immediate updates once influencing events happen in the core network.

This mechanism is also used by other entities, like the P-CSCF, which need to be kept updated on events and potentially take immediate action. One situation in which this information is valuable is for example to no longer allow signaling from the UE to enter the network upon de-registration. Also the HSS can request re-authentications, or the procedures can be used as effective mechanisms for CN re-configuration or maintenance procedures, where the registration procedure must

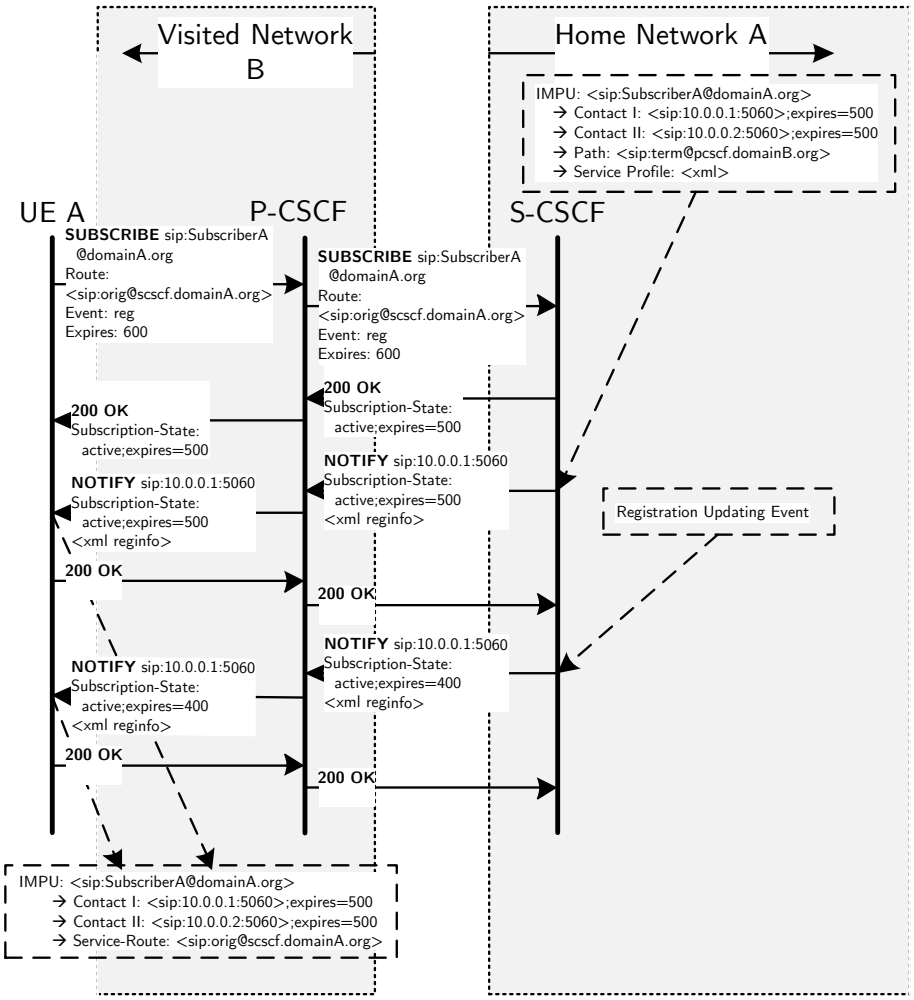


Figure 5.4: Subscription to the "reginfo" Package at the S-CSCF

be triggered[15]. However, due to its potential exposure of private information, the procedures must additionally verify that the subscribing entity is allowed to access this information.

[15] E.g. part of S-CSCF reallocation.

5.1.2.3 User and Administrative De-registration

To undo the identity bindings in the registrar, a similar operation is used as for registration, taking advantage of the expiration parameter and setting it to 0, as an indication for immediate time-out.

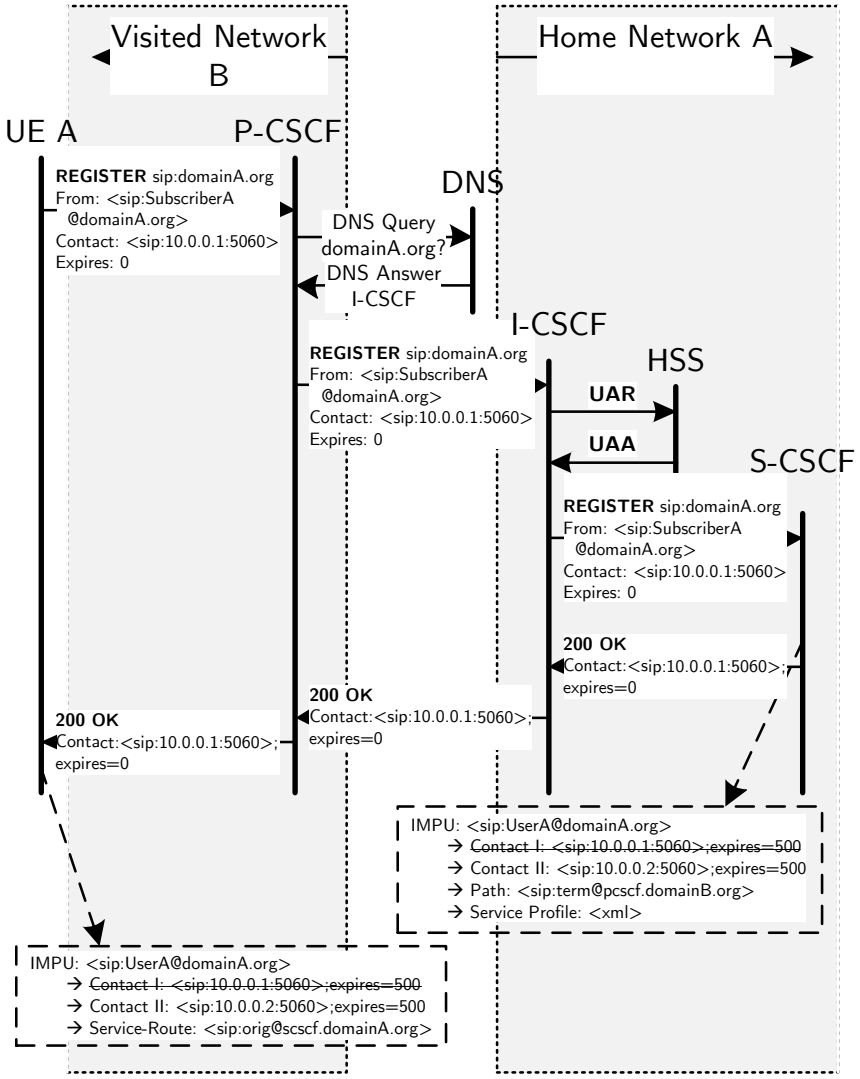


Figure 5.5: Partial De-registration of a Contact Address (No Server-(De)Assignment is performed as there are still contacts left)

When the de-registration is administrative and as such initiated on the net-

work side[16], the HSS can trigger it by using the Registration-Termination-Request (RTR)/Registration-Termination-Answer (RTA) procedure (functional level operation *Cx-Deregister*) on the Cx interface to the S-CSCF. The S-CSCF continues the procedure by expiring the mappings in the registrar. The UE as well as other interested entities will be informed through `NOTIFY` messages on the subscription sessions for the `reg` event.

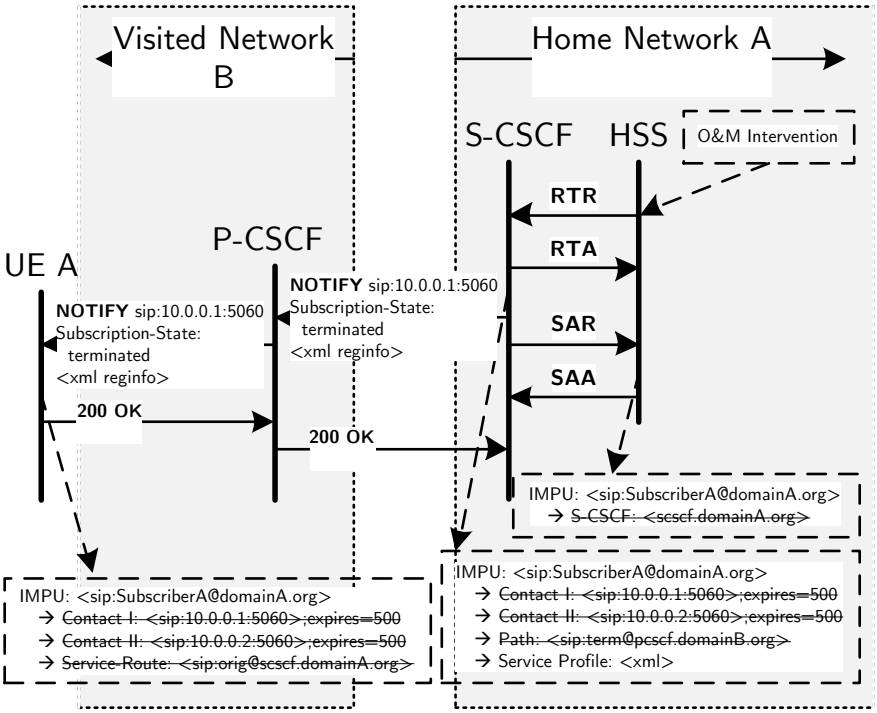


Figure 5.6: Administrative De-registration

Upon any such de-registration, when there are no more mappings for the respective IMPU on its local registrar and also there would be no services for the *Unregistered* state, such that a transition to the *Not-Registered* state is imminent, the S-CSCF will release its association with the IMPU at the HSS global location database, with the SAR/SAA procedure. From that point on, subsequent registrations will then potentially be served on other S-CSCF instances.

[16] E.g. in the O&M console

5.1.2.4 Summary

For the purposes of mapping subscriber identities to network addresses, all of the main IMS functional components are involved. The registration procedure is UE initiated and entails the use of expiration timers to control the operation. The S-CSCF, besides being the registrar where the identity mappings are saved, provides subscription based notifications to interested parties, sending immediate updates on registration status updates in order to avoid state de-synchronization. Administrative de-registration is also possible.

5.1.3 Locating Subscribers and Signaling Routing

Once the registration operation has completed the mapping operations between IMPUs and *Contacts*, the second part of the mobility operations can proceed. For any signaling that would need to be terminated towards an IMPU belonging to the home network, the I-CSCF, HSS and S-CSCF will proceed to route the respective message towards the subscriber's UE.

5.1.3.1 Location Procedure Overview

While signaling might originate in the same home network, for generalization it is considered that signaling originates in an external domain. The standard procedure is then to perform a DNS query in order to find the destination's home network. As mentioned before, the result will in fact be direct signaling to the I-CSCF.

The I-CSCF then initiates the functional level operation of *Cx-Location-Query* towards the HSS by sending the Diameter Location-Info-Request (LIR) message. Upon consulting its global location directory, the HSS will include in the Location-Info-Answer (LIA) message the address of the S-CSCF which is servicing currently the respective subscriber.

Next the SIP signaling message is forwarded to the respective S-CSCF, which applies first the service filtering procedures. After completion the S-CSCF will forward the message towards the UE by using the stored *Contact* network addresses.

Along with the *Contact* information that the S-CSCF registrar has stored, as indicated for example in Figure 5.3, additional information was saved which would help at this point. The `Path` header was filled by the P-CSCF during the registration operation. This value is now used as `Route` header to direct the signaling through the Visited Network's P-CSCF, currently servicing the UE. The final destination is filled by replacing the *Request-URI* with the saved *Contact* network address. As not to lose the important IMPU information (the destination UE could service multiple identities and it needs to know what was the actual destination identity), the original *Request-URI* value is saved in a `P-Called-Party-Id` header [158].

This procedure of using the `Path` header to route terminating signaling through the P-CSCF, as defined in [160], ensures a proper and secure routing of the packet towards the destination network. This is required as first of all it is not guaranteed that the Home Network would be able to actually route packets toward the UE

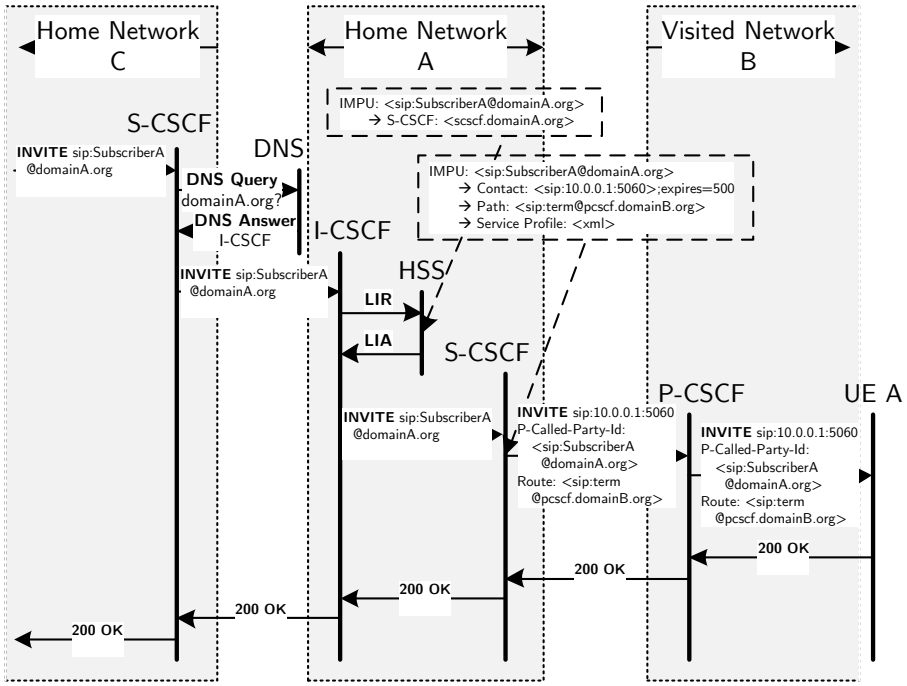


Figure 5.7: Terminating Leg Routing in IMS

address in the visited network[17]. Secondly, for security reasons, the UE must reject all signaling which does not arrive over the secure communication path established upon registration over the Visited Network's P-CSCF[18].

5.1.3.2 Forking to Multiple Contacts

In case more than one *Contact* address is provided for the same IMPU, the S-CSCF will proceed to "fork" the signaling. The procedure is the standard SIP [49] one and entails the creation of multiple sessions, individually targeting different *Contacts* and routing signaling according to the respective terminating route policies.

As provisional responses could potentially arrive for the multiple and different transactional legs, the originating party should be able at this point to distinguish and apply individual handling procedures.

This "forking" procedure realizes in practice a scenario in which multiple client devices, which have registered the same common identity, will start "ringing" in parallel when that identity is called. When one device answers, the communication

[17] E.g. the Visited Network uses NAT, or a different IP version network than the Home Network.
[18] More details on the security procedures are provided in the security related sections

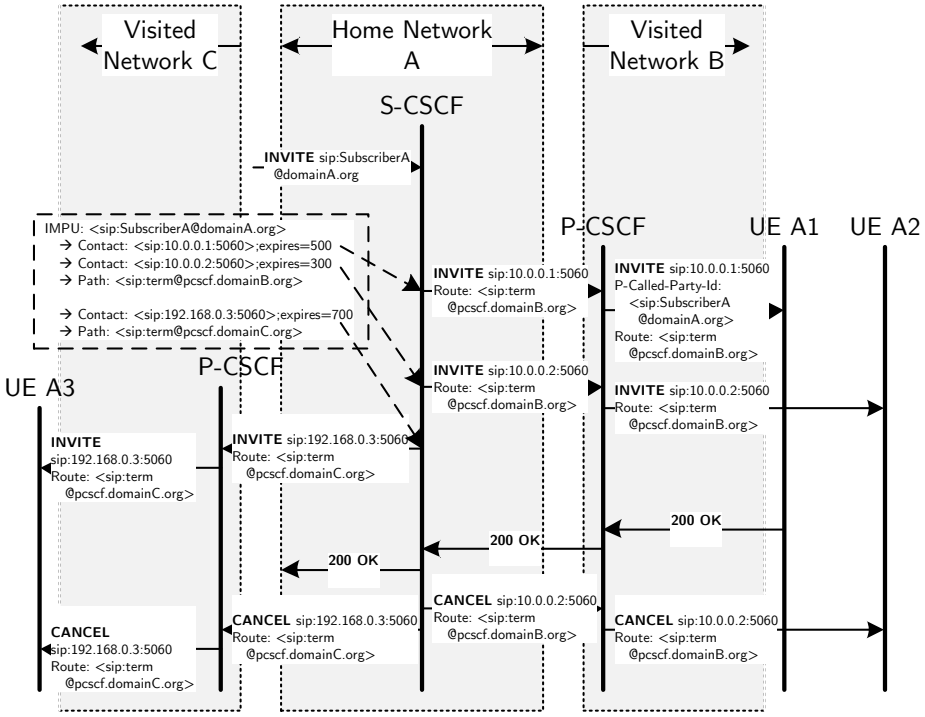


Figure 5.8: Forking to Multiple Contacts, UE A1 Answering

session is established and all other parallel pre-sessions will be explicitly aborted.

The S-CSCF is also in charge of sending `CANCEL` requests during these procedures, upon receiving a final response on one of the forked legs, to the other legs, effectively aborting the not-yet-answered transactions.

5.1.3.3 Originating Leg Routing Policy

The registration procedure provides through the `Path` header the mechanism to learn and make use of the terminating leg routing policy and path. The same registration procedure is also used for similar purposes, but in regard to the originating leg routing. This provides the facility for applying operator policies on how the UE originating signaling is to be routed, ensuring for example that all originating signaling is checked and processed correctly, as for example through a charging server.

As indicated in Figure 5.3, the S-CSCF fills in the `200 OK` answer to `REGISTER` one or more `Service-Route` headers. Obviously other proxy functions on the return path to the UE can proceed in a similar manner to complement these values.

The UE will save the `Service-Route` values locally and then will use them as

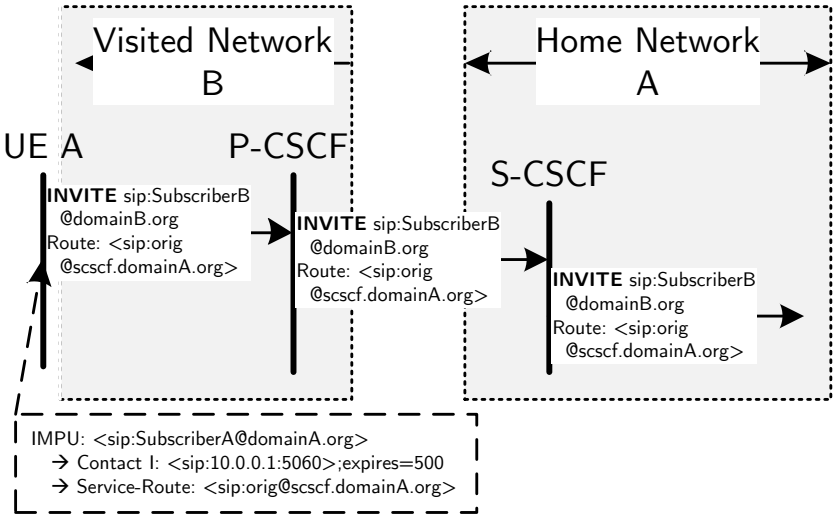


Figure 5.9: Using the Service Route for Originating Leg Routing Policy

`Route` headers on every standalone or initial dialog creating requests which it will originate. This will ensure for example that all such originating signaling will be routed through the Home Network's S-CSCF which currently services the user and would be required for filtering the signaling and triggering originating leg services[19].

5.1.3.4 Summary

The information saved and updated during the registration procedures is reused by the destination subscriber's Home Network I-CSCF, HSS and S-CSCF to compose the route for terminating signaling towards the subscriber device. The path follows first the S-CSCF where service triggering is applied. Then the signaling is potentially forwarded towards multiple destination network addresses. The signaling is further routed through the P-CSCF in the Visited Network and from there to the destination UE. In case of |forking", the S-CSCF is also in charge of canceling unanswered legs when the first final response is received.

The mobility provided is a coarse one in the sense that only the initial requests employ the above routing procedures. SIP transactional responses are routed by following back the path learned through the `Via` headers stack. Subsequent requests

[19]It is debatable still if the P-CSCF should explicitly insert its network address as a `Service-Route` header. While this will ensure the routing through the P-CSCF, this is actually a critical part of the trust chain operations and is to be enforced in any case by the security operations, hence the inclusion is a waste of network and processing resources.

are routed also by reusing the routing path learned during the initial transaction, yet the mobility of such session is the subject of the next subsection.

5.1.4 Session/Dialog Mobility

SIP dialogs are used to group context sharing transactions and to reuse an initially discovered routing path. The first request and its answer, which create the dialog, are used to learn the network routing path by allowing all proxy functions interested in the subsequent transactions to insert their own network address as `Record-Route` headers. Together with the network addresses of the originating and terminating end nodes, which are in fact transported in `Contact` headers, this will represent the dialog route. Subsequent requests will use this stack of learned network addresses as `Route` headers and the opposite endpoint network address as the *Request-URI*. Location procedures can then be skipped and the signaling is directly routed on a hop-by-hop basis towards the destination.

Upon mobility of the UE, the need arises for changing this signaling path while dialogs as multimedia sessions or subscriptions to notifications would be in progress. Potentially the *Contact* network address of the communication endpoint changes. Also the P-CSCF serving the UE could change, as this is part of the Visited Network's AN, while the UE could roam to another AN or even another Visited Network domain.

The SIP protocol specification [49] does not allow in fact the change of a recorded dialog route while the dialog is in progress. Only the `Contact` header values can be updated. This covers then only the IP address change, but not the P-CSCF change[20].

Updating the network address of the UE is performed as a re-invite (respectively re-subscribe for a subscription dialog) operation, where a subsequent request is sent on the original dialog, using the same `INVITE` method and indicating the new `Contact` header value to the other endpoint. For the generalized case, the more radical route change is of more interest to be discussed here as it covers also the potential route change.

As the dialog route cannot be changed, new dialogs are established and the associated application data streams are re-associated with new signaling sessions. Of course, the end-to-end applications would need to properly correlate and identify the session, as being handed-over between signaling dialogs, yet this is an end-to-end application dependent procedure and of no concern in regard to CN procedures and capabilities.

5.1.5 Summary on IMS Mobility

Mobility functionality is enabled by the described mechanisms, as the active network location of the user is followed through UE provided updates. This information is

[20] In practice, many of the current 3GPP mobile networks provide IP network mobility and do not change that often the IP address allocated to the UE.

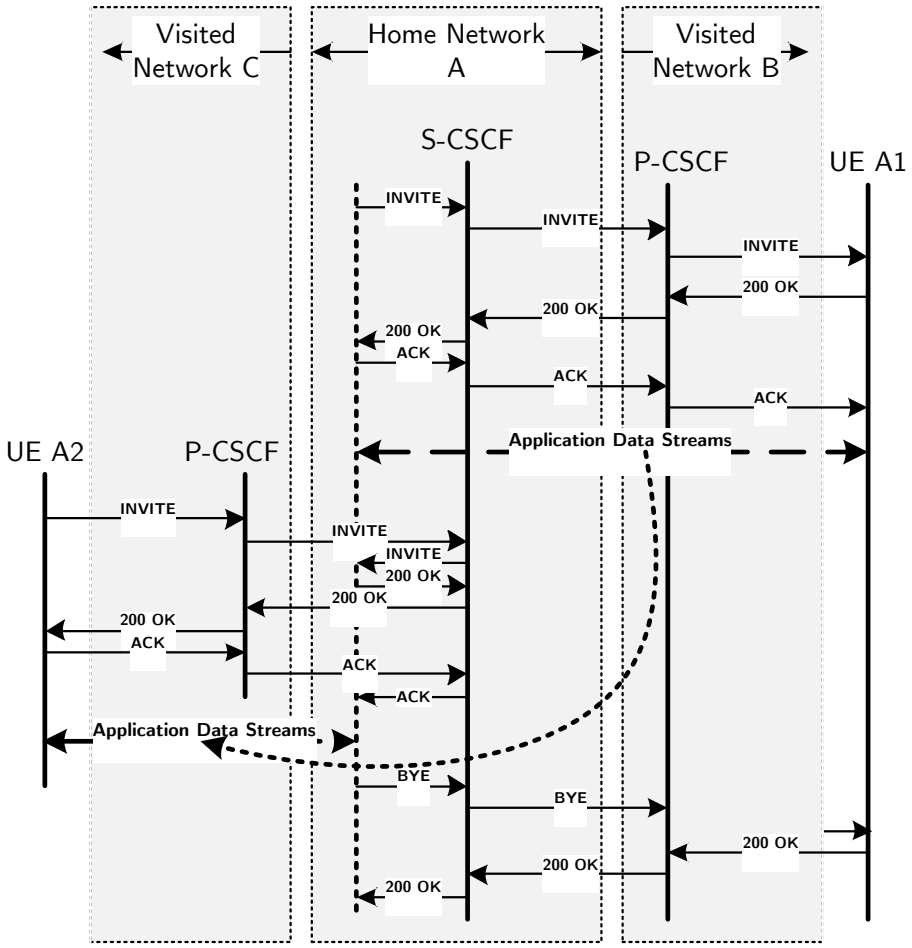


Figure 5.10: Explicit Dialog Handover between UE A1 and UE A2

then used to route signaling. Network terminating routing policies are also part of the operations, ensuring the correct signaling path is followed. For sessions, mobility is limited, requiring end-to-end applications to individually establish new signaling dialogs and migrate for example their active application data streams.

5.2 Security Operations

Within IMS the security operations are critical for a correct and carrier-grade functionality of the CN.

Historically the terminal devices, also referred to as Customer Premises Equip-

ment (CPE), have contained a relatively low level of intelligence and complexity, with most of it being actually concentrate in the core network. This kept also the complexity of the security architectures relatively low: most security procedures were operated and contained in the CN; the "socket-in-the-wall" was secured within the customer's premises and was within the customer's responsibility.

However, the NGN architecture complicates the issue as an important intelligence is actually deferred to the UE devices and equipment. Mobility requirements add yet another dimension by replacing the simplicity of the "socket-in-the-wall" previous solutions with the need to provide end-to-end security. This must not be underestimated as the use of common IP protocols and transmission equipment implies a zero-barrier for intercepting communication. Hence the transport channels must be secured accordingly.

In the security domain there are two main parts:

- signaling security – which should ensure that the signaling instructions are transmitted safely, without allowing for unintended interception and ensuring a solid end-to-end chain of trust between the various edge and core functional components;

- media security – which must ensure that also the application specific data is transported in a similar manner, such that interception is exclusively possible for legal purposes and the communication end-points can trust the transport channels.

Both of these domains are studied next, with an additional insight into the identity authentication procedures, forming as such the blueprint for implementing a secure test-bed prototype.

For the authentication mechanisms as well as the key agreement for cyphering and integrity protections described here, initially 3GPP called in Release 5 [146, 148] for an exclusive use of the AKA [149] mechanism and IPsec [161], very similar to the ones used in UMTS [147]. Later on, as the IMS concepts have started to be adopted by a larger telecommunication community, the requirements have been complemented first by standardization bodies like CableLabs or ETSI TISPAN, with less restrictive mechanisms[21] based on the established MD5 [162, 163] and TLS [164, 165]. These new additions have been adopted since the 3GPP Release 8 of the architectural standards, as part of the Common-IMS [11] efforts.

5.2.1 IMPU and CN Authentication

Encryption mechanisms for ensuring privacy and integrity of transported data are discussed in the following section. A requirement for these operations to effectively

[21] The AKA mechanisms would require a secure ISIM module, which typically is not deployed in fixed networks. The securing mechanisms are there usually provided by various options, often not interoperable, like proprietary CPEs with a stricter controlled O&M, internal procedures and security.

provide usable end-to-end trust chains is to ensure authentication of end-points. Without this, while the data is in itself protected from eavesdropping or malicious modifications, still it can not be trusted without ensuring the validity and authentication of the end-point's identity.

To summarize the point, it has to be noted that it is possible to establish security associations and to protect signaling, yet this does not provide a proper IMS communication channel without providing the additional authentication of used identities. Even after authentication, the edge functions in the chain of trust must enforce the use of only properly authenticated identities to further prevent impersonation attacks by exclusively using secure channels.

The authentication procedure are also defined to be, if possible, mutual, such that not only the identity used on the UE device is authenticated to the network, but also the network is authenticated to the UE, ensuring that rogue CN or Man-in-the-Middle (MITM) attacks are mitigated. The first mechanism to be studied, AKA implicitly includes and enforces the "mutual" authentication characteristic, while the second mechanism, TLS provides it through an asymmetric mechanism - MD5 for the UE to CN and TLS certificate verification and trust chains in the opposite direction.

Authenticating the UE identity is a mechanism designed to happen at the beginning of the signaling procedure, after network attachment. The identity used in public signaling, to be authenticated, is the IMPU. To ensure better security, this identity is only mapped to a set of private identities IMPI, which upon successful verification of commonly shared secret keys, are allowed then to take control of the IMPU.

5.2.1.1 AKA Authentication

Within the AKA [149, 148] procedure, the UE starts by sending a `REGISTER` request. The IMPU identity is included in the normal `From` header, while the IMPI is provided in a mostly empty `Authorization` header. This SIP message is routed just as indicated in the previous section detailing the mobility operations. On the S-CSCF, as the IMPU is not yet marked as authenticated, the HSS will be triggered through the *Cx-AV-Req* procedure with the Multimedia-Authentication-Request (MAR) message to generate one-time authentication vectors.

In this procedure the HSS will check if the IMPI $\Leftrightarrow$ IMPU association is a valid one. Then it will proceed to use the AuC to generate the requested vectors, by using the internally stored secret values for `K`[22], `OP`[23], `AMF`[24] and `SQN`[25,26]. As output a random challenge `RAND` is generated, as well as a network authentication token `AUTN`, an expected response `XRES` and cipher/integrity protection keys `CK`, respectively IK.

[22] Secret key
[23] Operator Secret
[24] Authentication Management Field
[25] Sequence Number
[26] The procedure to generate the authentication vectors are slightly out of topic here, but can be found in [147].

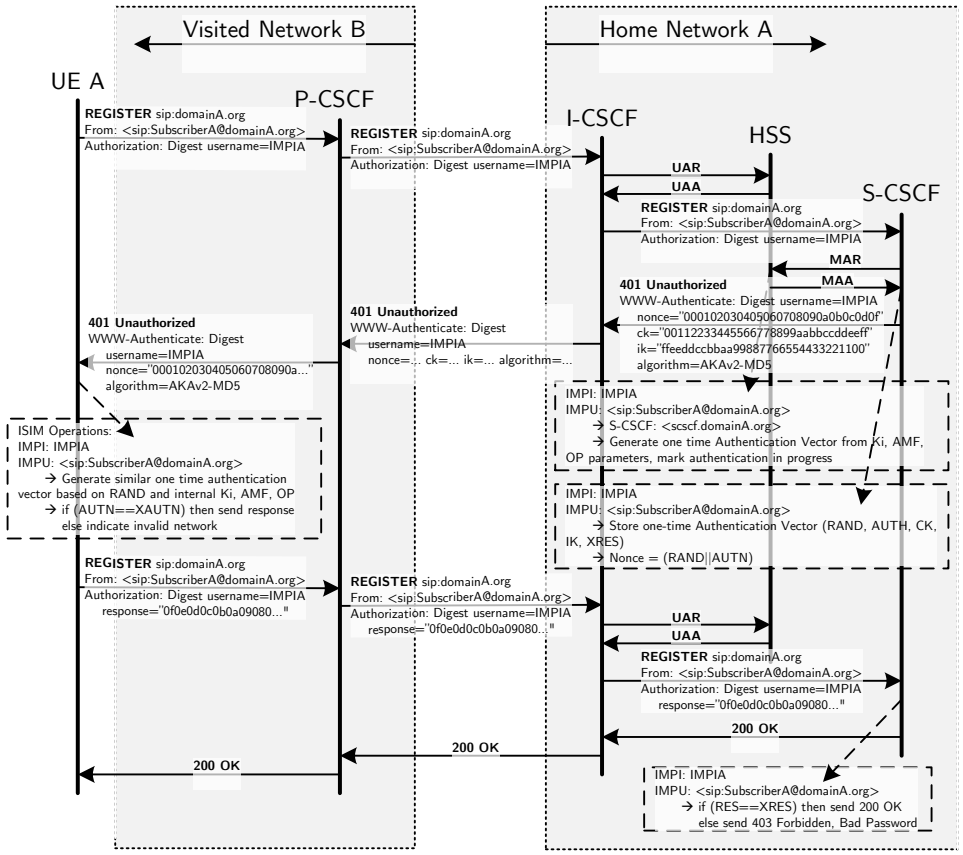


Figure 5.11: Authentication of the IMPU with AKA

When receiving the one time authentication vector(s) from the HSS, the S-CSCF proceeds to save the values and forward all but the `XRES` expected response in a SIP `401 Unauthorized` response message, within the `WWW-Authenticate` header. The P-CSCF will further remove from this header the `CK` and `IK` values while forwarding, as they are not to be sent over the potentially insecure Gm interface.

On the UE, the `RAND` and `AUTN` parameters are decoded from the `nonce` parameter in the `WWW-Authenticate` header and passed to the ISIM module for processing. This is provisioned with the same shared secrets as the AuC, yet its internal mechanism would not divulge the secrets. The SIM including the ISIM and other security modules includes the required logic to entirely process itself the authentication[27].

First the `AUTN` is verified, such that the ISIM can assert if the AuC in the network

[27] The `K` and `OP`/`OPc` keys are stored on the ISIM in such a physically protected manner that they can not be directly read. Only on correct (`RAND`||`AUTN`) pairs would the ISIM produce authentication materials as `RES`/`CK`/`IK` or `AUTS` (last one used for re-synchronizations of `SQN`).

knows the same secret as the local one. If this step fails, the ISIM will not produce a valid authentication response, but will instead indicate that the network is not authenticated and a MITM attack might be in progress.

Next, to prevent also replication attacks, the `AUTN` included `SQN` is verified if valid[28]. In an error case, a resynchronization procedure is initiated (not depicted in Figure 5.11, by sending a similar `REGISTER` request, yet including also the `AUTS` parameter in the `Authorization` header. From this, the AuC can both verify if the ISIM is the one provisioned with the right keys to perform the authentication, but also will extract the ISIM `SQN` value and store it. Another one-time authentication will be generated and the procedure will resume with the new values, synchronized now also on the `SQN`.

After successful execution of the verifications above, the ISIM will proceed to generate and output to the UE the values for `RES`, `CK` and `IK`. The `RES` is then sent in a following `REGISTER` transaction, in the `Authorization` header. The cipher and integrity keys are used locally for securing the signaling in further procedures.

When receiving the `REGISTER` request including the `RES` parameter, the S-CSCF will verify it against the locally stored `XRES` expected value. Based on the result of this operation, the S-CSCF will indicate either that the procedure has completed successfully or that the UE did not successfully manage to authenticate with the respective IMPI the included IMPU. With this, the authentication procedure is complete.

As it is exposed in a following sub-section, IPsec security associations are established in the middle of the `REGISTER` procedures exposed here, once the `CK` and `IK` values are correctly set on both the P-CSCF and UE sides of the Gm interface communication. Once such a channel is established, as the 2 functional entities could have only obtained same key values if the authentication is in fact successful, the P-CSCF can proceed to fill in the `Authorization` header the `integrity-protected` parameter, indicating that the message has been authenticated as received over a secure channel. This is of special added value for the re-registration scenarios, when new authentication vectors do not have to be generated, but the CN can continue to trust the already established security context.

5.2.1.2 MD5 Authentication

Digest authentication with MD5 [162, 163] is the main authentication method used typically for SIP. As such, it provides a good background technology to take advantage of already existing protocol stacks and equipment.

The procedures are largely the same as described before, yet as there are no elaborate AuC or ISIM facilities available, the input to the algorithm is provided by a simple secret shared key. Various standardization bodies have produced for IMS unfortunately different procedures and specifications for the MAR/Multimedia-Authentication-Answer (MAA) operation at the HSS. These affect the way that the

[28]The `SQN` is a monotonically increasing sequence number, which ensures that each authentication vector can only be used once.

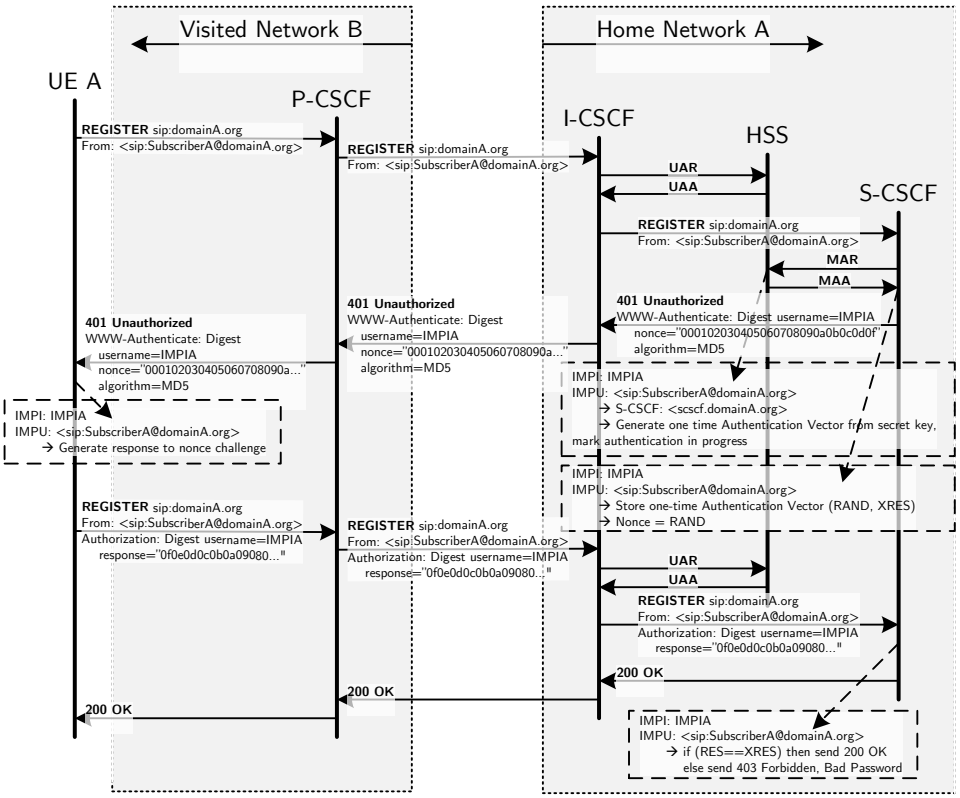


Figure 5.12: Authentication of the IMPU with MD5

secret key is handled, either kept protected in the HSS and authentication vector being sent, or sent to the S-CSCF for computations and verifications. Fortunately the differences are limited to the I-CSCF/S-CSCF $\longleftrightarrow$ HSS (Diameter) (Cx) interface, such that the SIP `REGISTER` procedures are always kept the same as the standard ones.

The MD5 authentication is inherently weaker as it does not provide features like protection against replication or MITM attacks. Unlike AKA, it also does not provide support for generating keying material, such that creation of security association is not possible without additional procedures. It is then often associated with TLS security associations[29], established previous to the first `REGISTER` message

[29]Standard SIP authentication often calls for subsequent authentication of each and every message, making the signaling quite heavy. IMS solves this by using the IPsec or TLS security associations. Once the security associations are deemed to be authenticated, subsequent messages using these channels no longer need to be individually authenticated, but are implicitly trusted if they decode properly.

exchange. By using certificates with verification through trust chains, the UE can also authenticate the network even before the authentication procedure starts.

On the opposite direction, it has to be noted that in this case of using MD5 authentication coupled with TLS, the `integrity-protected` procedures mentioned for AKA have to further consider that a successfully protected message is not to be considered authenticated before the IMPI authentication actually completes.

Then TLS requires the exclusive use of TCP based signaling transport, which fares worse on latency than UDP.

Due to its extensive use in fixed networks, the MD5 authentication can not be ignored and it is a required for any CN which is to claim FMC properties. Besides, ISIM modules are typically only available in mobile environments and even there, typical SIM cards are not Over-The-Air (OTA) upgradeable to support the new functionality, but need to be physically replaced.

5.2.1.3 Other Authentication Mechanisms

Besides the principal authentication procedures presented before, a few other alternatives have been standardized. These methods either take advantage of specific network environment or capabilities, or represent in-between solution for early deployments until full architectures and equipment would be available.

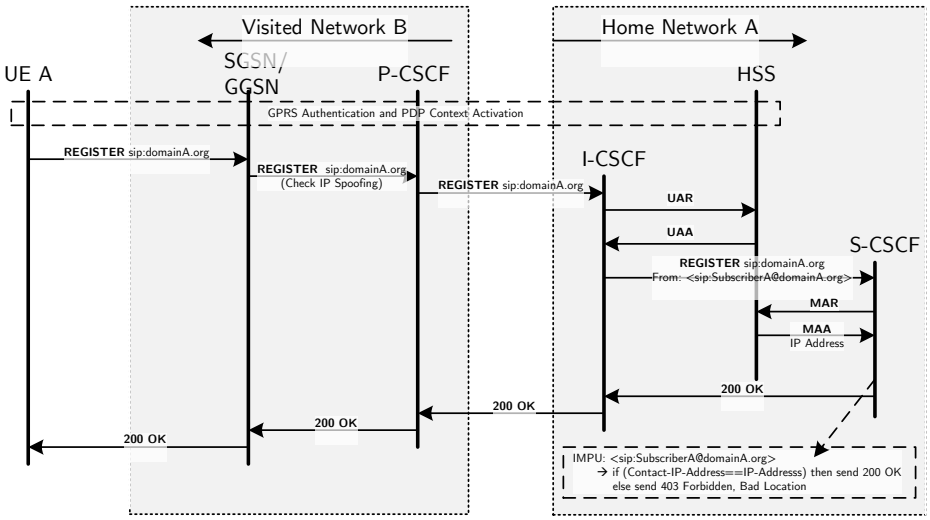


Figure 5.13: GPRS-IMS Bundled Authentication (GIBA)

The chronologically first such mechanism to be specified is the 3GPP Early-IMS authentication, also referred to as GIBA [148]. This uses a mapping between the IP address allocated to the UE and the IMPI. During GPRS (or UMTS) attachment the

network uses the SIM (respectively UMTS SIM (USIM) for UMTS or if avaialable for GPRS too) to provide authentication, which allows the HLR/HSS to create such a mapping. The RAN is then responsible for the validity of this mapping as well as to prevent other UEs from spoofing IP addresses.

In such environments, although limited in capabilities and to the mobile network environment, the `REGISTER` requests are automatically considered to be authenticated if they originate from the associated IP address. The procedures are similar, with the difference that the Cx MAR/MAA operation would return the currently authenticated IP address associated to that IMPI instead of authentication vectors. The S-CSCF can immediately identify if the `REGISTER` request originated from the authorized IP address and complete the process in a single transaction.

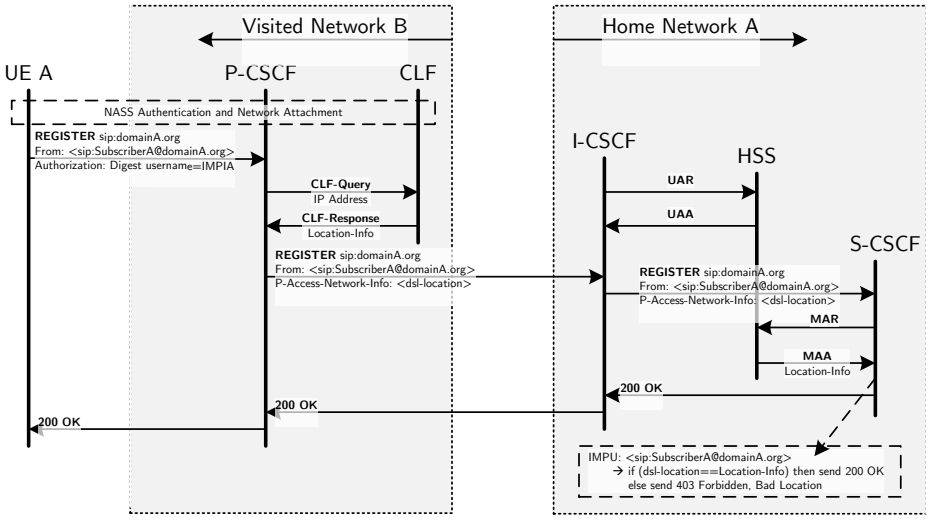


Figure 5.14: NASS-IMS Bundled Authentication

A similar mechanism, yet targeting fixed network deployments, is the NASS-IMS bundled authentication [148, 166]. This has the same requirements fir the access system to assert and protect IP addresses as associated to IMPI identities. Yet it is more elaborate in the sense that an additional security component is introduced in the architecture, the Connectivity Session and Repository Location Function (CLF). This communicates over the P-CSCF $\longleftrightarrow$ CLF (Diameter) (e2) interface with the P-CSCF and allows it to fill extra information in the `P-Access-Network-Info` header, as for example the DSL line identifier, as derived from the IP address of the UE. The HSS mechanisms will then use these enhanced identifiers to proceed in a similar single transaction `REGISTER` authentication.

The mechanisms presented here have an additional requirement as to provide the data transfer confidentiality and integrity protection implicitly, as the authentication

mechanisms themselves do not provide additional material to allow the use of the aforementioned IPsec or TLS procedures.

5.2.2 Signaling Protection

Signaling protection in IMS refers to the mechanisms which are used to ensure that the application and service procedures are performed always in a secure manner. The signaling is to be protected against eavesdropping from unintended entities by ciphering. Even more, specific parts of the messages have to be individually ciphered or hidden such that only the intended information would be disclosed for example at the interfaces between different operators or security domains. The used identities have to be authenticated centrally such that functional entities on the signaling path do not need to individually each perform authentication, but can use instead a chain-of-trust system. And to round-up, the messages must be also integrity protected, such that functional nodes would be able to indicate if the signaling has been tampered with.

As various CN domains represent also security "bubbles" based on which and how such signaling protection mechanisms are applied, the security mechanisms required here will be specified in two parts:

1. for the UNI between the UE and the CN;
2. for the NNI between CNs belonging to different domains.

5.2.2.1 User-to-Network Interface (UNI) Signaling Protection

The UNI exposes the CN directly to UE devices and potentially more hosts. Accordingly all signaling has to be thoroughly verified upon entering the security domain. The P-CSCF represents then an SBC deeply specialized in handling the UE devices and securing the CN according to the IMS security standards and considerations.

Upon registration the P-CSCF assists and establishes with the UEs Security Associations. These are in effect the cryptography contexts which allow for secure signaling exchanges between the CN and the terminal devices. As indicated in the authentication section, there are two main options currently in use for these Security Associations: IPsec when using AKA authentication; or TLS when using MD5 or one of the other less strong and not cryptography-material producing authentication algorithms.

For IPsec, the AKA procedure ensure that the P-CSCF receives the AuC generated `CK` and `IK` keys used for ciphering, respectively integrity protection of the messages. On the UE side the keys are generated locally by the ISIM and in case the AuC and the ISIM share the same input keys and parameters, for the same challenge `RAND`, both would generate the same values. Accordingly, no further signaling is required and both the UE and P-CSCF can immediately encrypt, decrypt, integrity protect and verify messages right away.

The nature of the IPsec Security Associations procedure [148] involves the use of IPsec [161] Encapsulating Security Payload (ESP) in transport mode. This implies

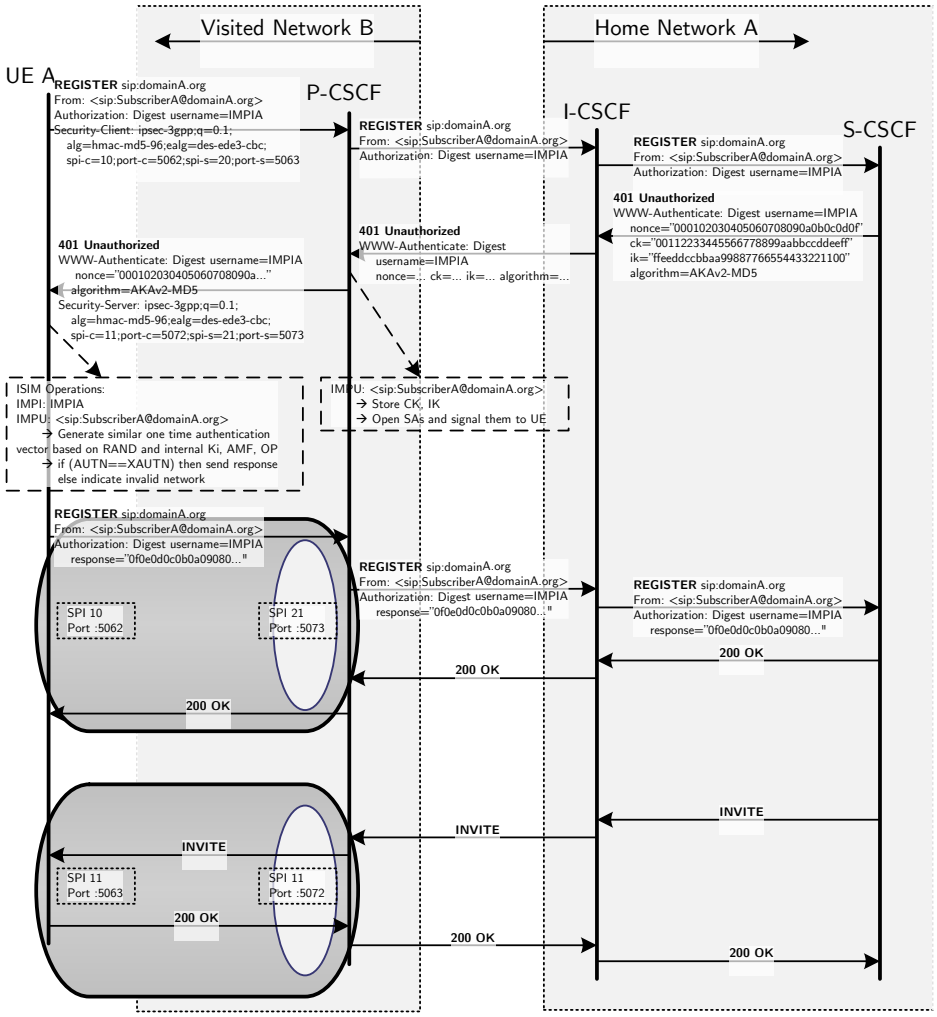


Figure 5.15: IPsec Security Associations

the allocation of 2 dedicate communication ports on each of the UE and P-CSCF sides: one for inbound signaling and one for outbound signaling. Each of these ports should be configured on the respective function to only allow correctly decrypted and integrity-protected messages, discarding all other.

For integrity protection, the HMAC-MD5-95 [167] and the HMAC-SHA-1-96 [168] authentication algorithms are specified [148] and can be used with keys derived from the Integrity Key (IK). For ciphering, the DES-EDE3-CBC [169] and the AES-CBC [169] encryption algorithms are specified and can be used with keys derived from the Cypher Key (CK). The procedure uses the mechanisms in [170] with a mechanism name of `ipsec-3gpp` to negotiate the algorithms to be used and to

exchange IP port and Security Parameter Index (SPI) information.

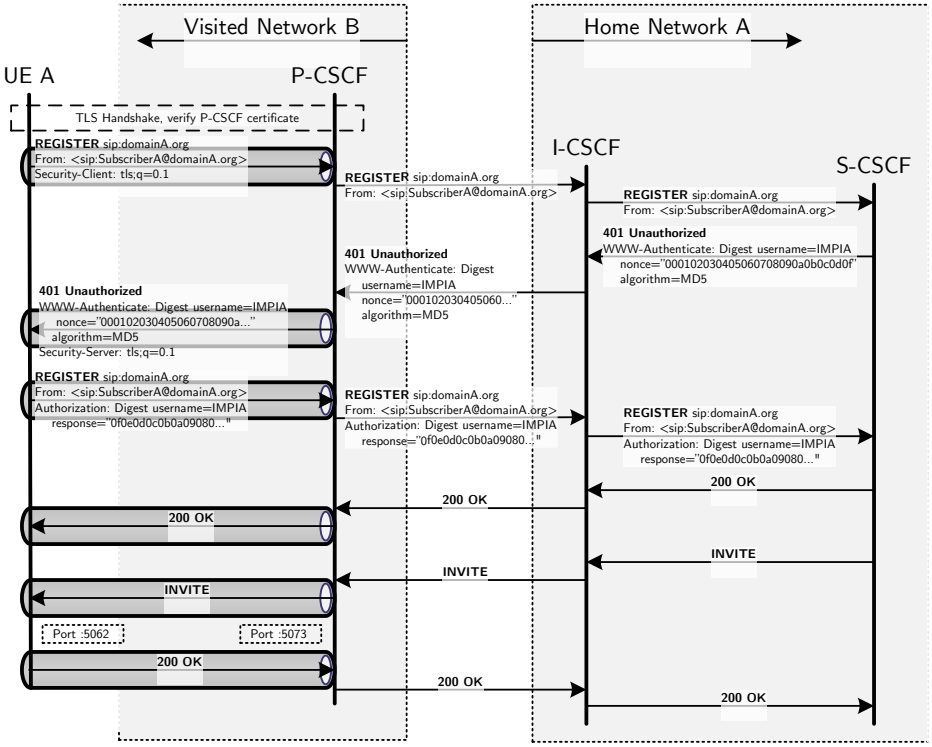


Figure 5.16: TLS Security Associations

When SIP Digest authentication is used instead of the AKA option, the lower latency, yet more readily available TLS at version 1.0 [164] mechanisms should be used, with extension from version 1.2 [165] and according to the provisions in [148]. The negotiation of the security algorithms and keys are to be executed at handshake, as per standard procedures. If the negotiation is not taking place before, but during the registration procedure, the same [170] procedures are to be used with the mechanism name of `tls`. The P-CSCF should present a server certificate, allowing the UE to authenticate its identity as part of an authorized IMS network trust chain.

The P-CSCF is then following on the signaling and will only allow messages through which are received over the established Security Associations, as well as the ones required to establish new ones.

Besides this, the P-CSCF will act as a signaling firewall, by enforcing only protocol correct messages to pass towards the CN. The SIP message headers are check for correctness and incorrect message are either rejected as unacceptable or corrected

and then forwarded. For example, the P-CSCF verifies if the correct `Via` headers stack has been filled in responses according to the ones in requests, ensuring that malicious responses would not bypass certain SIP proxies. Another example is that of dialogs and originating leg routing policies, where the `Record-Route`, respectively `Service-Route` headers learned are correctly used in `Route` headers. An overall logical function of the P-CSCF is that of ensuring that each Security Association is only used to originate signaling with identities authenticated previously, as to prevent impersonation attacks.

5.2.2.2 Network-to-Network Interface (NNI) Signaling Protection

Security of signaling at the interface between different networks or different security domains represents a different topic in its requirements, used mechanisms and solutions. Unlike in the UNI case, here the interface is typically between 2 operators which have a certain level of trust already established by mutual inter-connection or roaming agreements. Accordingly, the procedures here seek to enforce and verify those agreements, as well as to secure that communication between domains is not exploitable by 3rd parties, all at an aggregated level.

Standards wise the topic at hand is referred to as Network Domain Security (NDS) and the logical level functions are referred to as Interconnection Border Control Functions (IBCFs). These are typically grouped normally in pairs, facing each-other over a network link which is at its base potentially insecure, adding the features required for securing it.

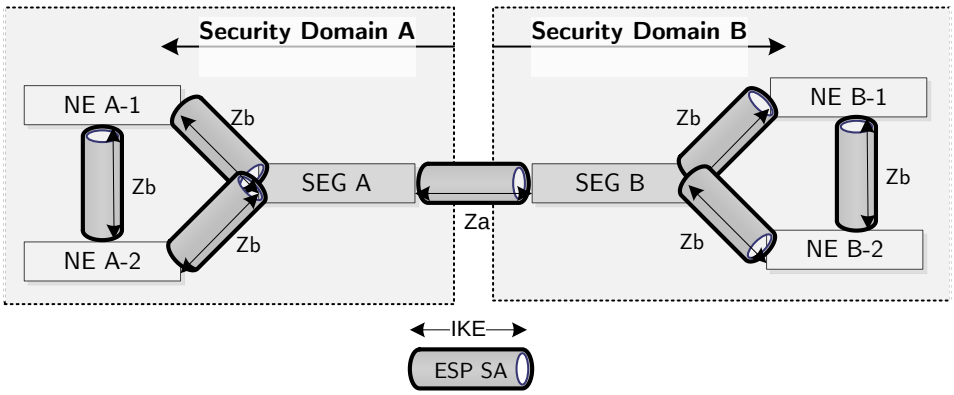


Figure 5.17: NDS IP Security with IPsec

Starting from the IP layer, the security of the network interfaces calls for the reuse of standard and established IP security mechanisms [171]. IPsec connections in tunnel mode are to be established between the various Security Gateways (SEGs)

and Network Entitys (NEs) acting in the IMS CN, as depicted in Figure 5.17. The management and distribution of the security keys are based on the standard IPsec Internet Key Exchange (IKE) system [172, 173, 174, 175]. The ESP encryption algorithms to be used are the ESP-NULL[30] the ESP-Data Encryption Standard (DES), which should be phased out in favor of the better and more secure ESP-ESP Triple DES (3DES) [169], as well as the AES-CBC [169]. For ESP authentication, the options are ESP-NULL[31], ESP-keyed-Hashing Message Authentication Code (MAC) (HMAC)-MD5 [167] and ESP-HMAC-Secure Hash Algorithm (SHA)-1 [168].

Being IP standard, the presented SEG $\longleftrightarrow$ SEG (IPsec) (Za) and NE $\longleftrightarrow$ SEG/NE (IPsec) (Zb) interfaces do not imply additional requirements. Their realization can be approached by using standard security tools available in the current operating system, hence this translates into a simple setup and configuration topic.

Above the IP transport, at a logical layer, the NNI security between domains addresses first of all the trust of identities. The previously mentioned chain of trust started at the UNI is to be continued also over NNI by handling identities in an IMS standardized manner. For this purpose a series of SIP headers, as defined in [176], are used to transport trusted identities and identifiers. Special handling is applied to these headers whenever they are passed over network domain boundaries.

The "privacy" headers are:

- `P-Asserted-Identity`
- `P-Access-Network-Info`
- `History-Info`
- `P-Asserted-Service`
- `Resource-Priority`
- `Reason`
- `P-Profile-Key`
- `P-Served-User`
- `P-Private-Network-Indication`
- `P-Early-Media`
- `cpc` and `oli` URI parameters
- `Feature-Caps`

Considering also that many of the SIP headers reveal important internal topology information (e.g. IP addresses, routing paths, parameters, etc.), a special security

[30] No encryption for links that do not need it
[31] No message integrity protection for links that do not need it

function, referred as Topology Hiding Internetwork Gateway (THIG) is used on the IBCFs. The original sensitive header contents are replaced with encrypted values places as parameters for the network address of the encrypting node. Message routing continues normally, such that once future related responses or other messages return to the encrypting nodes, these would be able to decrypt and replace back with the original values before forwarding towards the internal security domain.

The THIG headers to be protected are:

- `Via`
- `Route`
- `Record-Route`
- `Service-Route`
- `Path`
- other similar header which might reveal similar information and need to be hidden

A downside of the THIG procedures is that the size of the messages will increase quite significantly. Most often encryption methods would use padding as to hide entropy related information. Next the SIP header formats are text, additional encapsulation techniques have to be employed for translating from the binary outputs of encryption methods. An envisioned solution is to use, instead of encryption, reference tokens. Then the security of the information is improved as this will never leave even encrypted the origin security domain. Yet it adds the risk of losing the information if somehow the originating IBCF would lose the respective information and of course adds requirements for state maintenance on the security gateways.

Such an alternative solution is possible as overall the standard recommendations do not place hard-requirements on the procedures, besides that of ensuring basic interoperability. Interoperability, as well as correct functionality is ensured through 2 basic features: replacement of the original URIs with the URI of the IBCF and grouping of multiple sequential header values to be encrypted with a single one, yet of course maintaining the order and interleaving with non-encrypted header values. Hence any suitable data encryption mechanisms can be considered.

To close on the signaling security, the IMS-Application Level Gateway (ALG) is to be mentioned. This has the function of implementing translations between different network technologies or domains (e.g. IPv4/IPv6, very different and strict security domains). The functionality is part of the IBCF and acts as a SIP B2BUA, practically creating 2 different call legs and translating between the two at the application layer. This adds usability in a large number of scenarios, yet also requires numerous future upgrades as to keep up with the ever-changing application logic.

5.2.3 Media Security

Within the IMS architecture, the media delivery system is decoupled in large part from the signaling. UEs and applications directly negotiate media transport parameters and flows, typically through the use of SDP in the SIP signaling message bodies. Once this is completed though, the media can potentially flow directly between the end devices and applications, on the shortest path available[32].

Still, even though special media handling is not required and even more it can be argued that suitability of future applications would be adversely affected if media standardization would be enforced, it is of interest to at least mention here what mechanisms are recommended for the most common media.

The IMS media security architecture, as defined in [177] calls for the protection of RTP [178] transported media by use of Session Description Protocol Security Descriptions (SDES) [179] or Key Management Service (KMS) [180] coupled with Secure RTP (SRTP) [181]. Another common media is the Message Session Relay Protocol (MSRP) [182] and this is to be secured with TLS [164, 165].

5.2.4 Summary

The IMS security in scope for test-bed prototypes are mostly pertaining to signaling security. A critical point is the authentication of identities and signaling protection on the UNI, performed over 2 alternative mechanisms: the newly introduced for SIP signaling Digest-AKA coupled with the new IPsec Security Associations in mobile networks and the established Digest-MD5 coupled with the existing TLS protection mechanisms in fixed networks. Each has different traits, such that a single cover-all solution can not be easily obtained.

On the NNI, different security mechanisms are to be applied, as suitable for the interface between different network security domains or network technologies. These are encompassed in the NDS concepts: standard IP security tunnels with IPsec between operators, the use of standardized "privacy" headers to transport identities and identifiers, the THIG for hiding network topologies and the IMS-ALG to provide translations where needed.

Media security is also relevant, yet slightly out of scope when considering mostly the CN environment, as to keep it independent from a continuously and independently evolving applications space.

5.3 Session/Dialog Management

The dialog concept in SIP pertains to the grouping of several transactions within a context. The first request, referred to as the initial dialog creating request, is used

[32] Notable exceptions are: the legal interception procedures, which require also the monitoring of the media flows; trans-coding of media in case the end devices can not agree on a mutually usable encoding; end devices using different network protocols as IPv4/IPv6 or NAT and being unable to directly exchange media.

to discover and record a dialog route. Subsequent requests, coming from either side of the dialog, will then reuse the context and the recorded route, save important processing resources by not re-discovering it and also ensure that all signaling routers on the initial path are re-visited. There are only two types of dialogs defined so far, the `INVITE` dialog [49] used for multimedia sessions and the `SUBSCRIBE` [93] dialog used to provide a reusable notifications mechanism. Although more dialog types can be potentially defined in the future, for the current purposes of IMS, due to the implications in signaling processing which can not be foreseen, only these two are to be considered.

5.3.1 Establishment and Saving of Signaling Paths

In IMS the SIP dialogs are to be processed by the P-CSCF and S-CSCF, which are dialog stateful. These two CSCF functional node types need to follow on the initial dialog creating request and save an internal context. On subsequent dialog signaling the context is then reused. The dialogs can not remain open-ended as this will eventually cause exhaustion of resources on the respective CSCFs. Accordingly the procedures must always negotiate expiration timers as well as explicit dialog termination procedures.

During the dialog establishment signaling, both the P-CSCF and S-CSCF will follow on the `INVITE`, respectively `SUBSCRIBE`, transactions by saving the relevant dialog identifiers as well as the relevant routing information. Other signaling proxies can opt-in to be included in the subsequent signaling by adding themselves during this transaction in the `Record-Route` headers.

Subsequent requests are verified by the P-CSCF to ensure that the correct `Route` headers are used. In case of discrepancies, these are either fixed or the signaling is rejected as incorrect. The S-CSCF will route then also based on these headers, realizing the dialog route.

Both CSCF types will also need to keep track of dialog timers. These are updated based on specific header values for each dialog type, as presented subsequent messages. Additional mechanisms as [183] are used in the `INVITE` dialog, which was otherwise flawed by a lack of a built-in timer system. Subsequent signaling is here to be sent before dialog expiration, as to update all the signaling nodes involved that the session is still in progress. Upon implicit timer expiration all signaling nodes involved will discard the dialog context as stalled.

Explicit tear-down is of course the main mechanism to delete a dialog and to release the saved context information. The `INVITE` dialog is stopped by using the `BYE` SIP method, while in the `SUBSCRIBE` dialog the `Subscription-State` header is used to convey this information in either of the dialog's methods, `SUBSCRIBE` or `NOTIFY`. As upon dialog expiration, also in this case the entire context is released, such that eventual subsequent requests sent by misbehaving end-points will be rejected as orphan and hence invalidated.

5.3.2 Temporary Validations

While the dialogs are very important in saving processing resources by caching the routing paths, this signaling path is checked by the involved nodes. According to the SIP standard [49], a dialog's set of `Record-Route` headers can not be updated and has to remain the same for the entire duration of the dialog. If a communication endpoints does not correctly use the saved dialog path, this could be a sign of a potential attack and is to be prevented.

Additional validations concern the dialog expiration timers. The CSCFs have the option of changing the end-point requested values, as to indicate and/or enforce the operator's policies. Signaling belonging to expired or other no longer valid dialogs is to be explicitly rejected or silently discarded.

5.3.3 Summary

While not as complex as the registrar system, the session/dialog management procedures are critical in providing a reliable and secure communication environment. The P-CSCF and S-CSCF police the signaling and ensure proper message routing, as well as enforce the timely maintenance and eventual removal of stalled sessions/dialogs. Currently only two types of dialogs are defined, yet in the future, with new services and concepts, new dialogs and session models could be potentially defined. This will however entail the upgrade of the IMS CN components as to accordingly support and serve the new dialogs in a stateful, not just stateless manner.

5.4 ISC Interface to ASs

Service Triggering is one of the main architectural pillars of the IMS functionality. This enables the selection and activation of services dynamically as well as on a subscriber-by-subscriber basis. In technical terms, the service triggering is implemented by a signaling filtering and forwarding mechanism which analyses messages and redirects them as required by service triggers towards the AS implementing the service logic.

These mechanisms are implemented in the S-CSCF and the forwarding interface towards the AS domain is referred to as ISC.

5.4.1 Subscriber Profile Based Service Triggering

With IMS each subscriber is to be treated individually, such that even if services would be shared, each would enjoy a customized treatment and service space. The Subscriber Profiles are provisioned and stored in the HSS. Upon registration, the S-CSCF, as the signaling filtering point, downloads this information and stores it locally, as part of the SAR/SAA Diameter message exchange. Whenever updates are operated in the database, the HSS can send them to the S-CSCF associated with the respective subscriber in the previous procedures, with the Push-Profile-

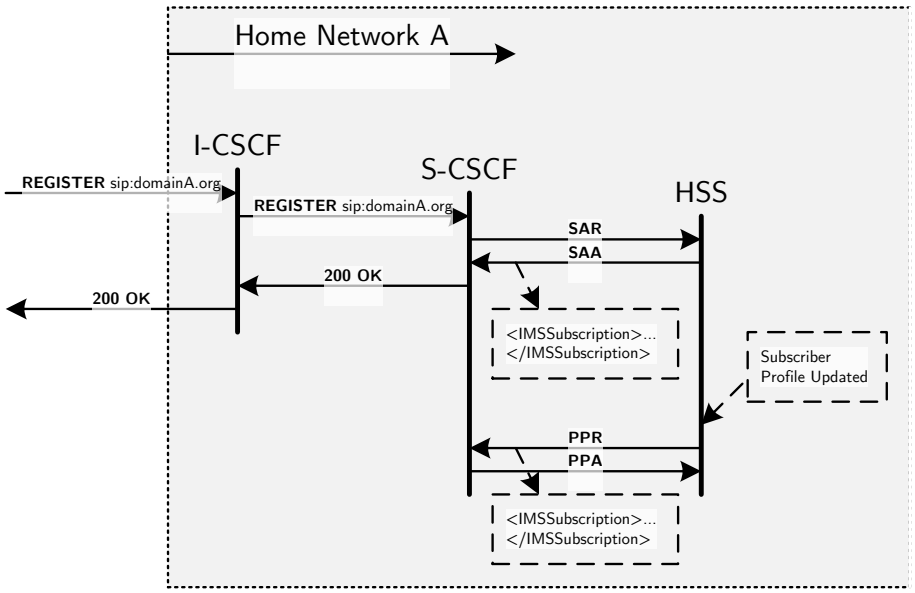


Figure 5.18: iFC Download from HSS to S-CSCF

Request (PPR)/Push-Profile-Answer (PPA) procedure, referred in standards [184] as *Cx-Put*.

The User Profile is downloaded from the HSS in an Extensible Markup Language (XML) encoded format. The data model is depicted in Figure 5.19 and is represented by an `IMS Subscription` which comprises of multiple `Service Profiles`, each for a sub-set of applicable `Public Identity` values. Each `Service Profile` (see Figure 5.20) contains the set of IMPUs to which it applies, and a `Core Network Service Authorization`, containing a string of service identifiers, followed by a set of `Initial Filter Criteria`, converging the filtering and forwarding information.

The `Initial Filter Criteria` (see Figure 5.21) is composed of a `Priority` ordered list of groups of Trigger Points, Application Server information and Profile Part Indicators. The `Trigger Points` represent the filtering information to be applied to the signaling and will be detailed in the next subsection. The `Application Server` is formed of a `Server Name`, relaying the SIP URI target for forwarding the signaling on matches; for cases when the AS does not respond, a `Default Handling` indicator is present here[33], along with additional `Service Info`. Finally, the `Profile Part Indicator` aids in distinguishing in services applied when the subscriber is an active state, or when the terminal is turned-off and unreachable.

[33]Indicates whether the signaling processing should continue or the session should be administratively aborted if the targeted AS is non-responsive

Figure 5.19: The IMS Subscription Data Model

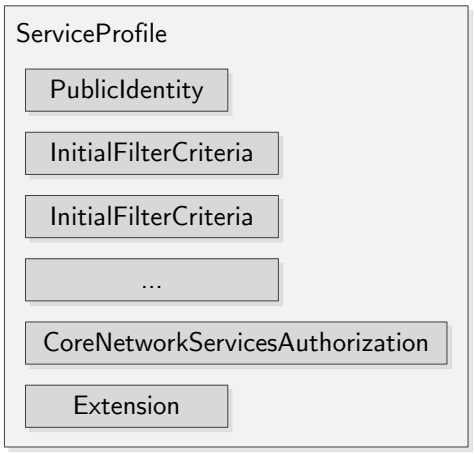


Figure 5.20: The Service Profile Data Model

To ease on the downloading operation, the `Initial Filter Criteria` which are common to multiple subscribers in the same form can be transported simply as identifiers, with provisioning for the full form as an additional on-the-side mechanism. This information is transported as a `Shared IFC Set ID` element.

5.4.2 Trigger Points and Filtering of Signaling

The Initial Filter Criteria (iFC) is the only filtering information present. There are no "Subsequent" similar concepts, this being just a standardization artifact,

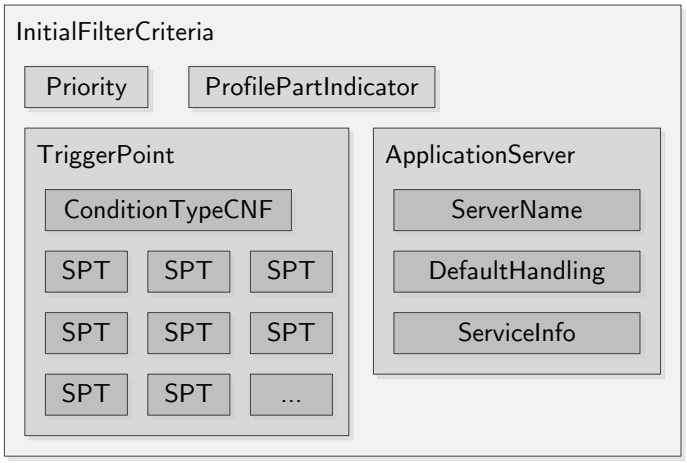


Figure 5.21: The Initial Filter Criteria (iFC) Data Model

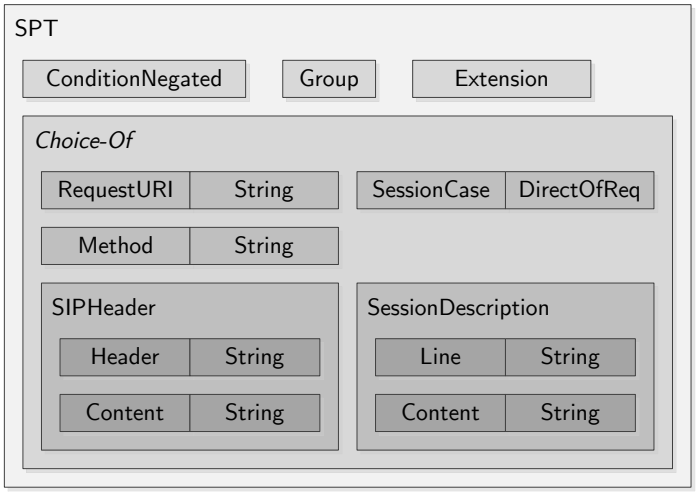


Figure 5.22: The Service Point Trigger (SPT) Data Model

carried over from IN, which was then rendered invalid by the adoption of SIP as the signaling protocol for IMS[34]. Accordingly, only the standalone SIP requests or the initial dialog creating ones can be filtered. SIP responses follow always the same route of transaction-stateful SIP proxies back as the request took, including the ASs

[34] SIP [49] compliance would be broken if subsequent dialog routes would be allowed to change.

which were visited on the path taken by the request. Subsequent requests in dialogs follow the dialog route, which can not be changed during the dialog. Any AS which needs to receive the subsequent requests of a certain dialog, must be triggered on the initial dialog creating request and then the AS will have to record itself in the dialog route, subscribing itself at such to all subsequent requests in that dialog.

The `Trigger Point` information in the `Initial Filter Criteria` element of the `Service Profile` is representd by a Boolean expression, with its atoms referred to as SPTs (see Figure 5.22). The atoms can be placed in either a Conjunctive Normal Form (CNF)[35] or Disjunctive Normal Form (DNF)[36].

The logical atoms to be evaluated are defined as being one of 5 distinct matching operations with parameters to be applied to the received signaling:

1. `RequestURI` equal to a string value;
2. `Method` equal to a string value;
3. `SIPHeader` present and/or matching a regular expression;
4. `SessionCase` being either originating or terminating; respectively from/to a registered/unregistered subscriber;
5. `SessionDescription` matching a line of the potential SDP payload to a regular expression

Each SPT can be negated and grouped to form the CNF or DNF expression, allowing as such for complex logical filters to be realized. An empty `Trigger Point` (containing no SPTs) is considered to always match and cause a consistent forward of the message to the associated AS. Figure 5.23 shows such an example for a simplified presence service triggering[37].

Each `Trigger Point` is checked against all standalone or initial dialog creating SIP request messages. In practice, a message is filtered potentially multiple times, for the originating, respectively terminating identity, in each step potentially being changed and forwarded with modifications.

As depicted in Figure 5.24, the filtering process is an iterative one, as a matching message can return back to the S-CSCF after being forwarded to an AS. Then filtering resumes, on the new messages, from the next `Trigger Point` in order of priority. Once all filtering has been completed and no more such matches are found, messages are routed forward normally, as per standard SIP [49] procedures, either towards other servers, or towards the terminating domain, or towards the P-CSCF serving the terminating UE.

[35] CNF: (A or B or C) and (D or E or not F)

[36] CNF: (A and B) or C or (not D and E and F)

[37] It has to be noted that the example is flawed for the actual purposes of a presence service. For correct functionality, the `SUBSCRIBE` request should be triggered only on the terminating side and the `PUBLISH` requests only on the originating side, as to ensure that the messages will reach the right presence server. As the full example is too long and complex, the simplified one has been depicted, which works for a single-domain only.

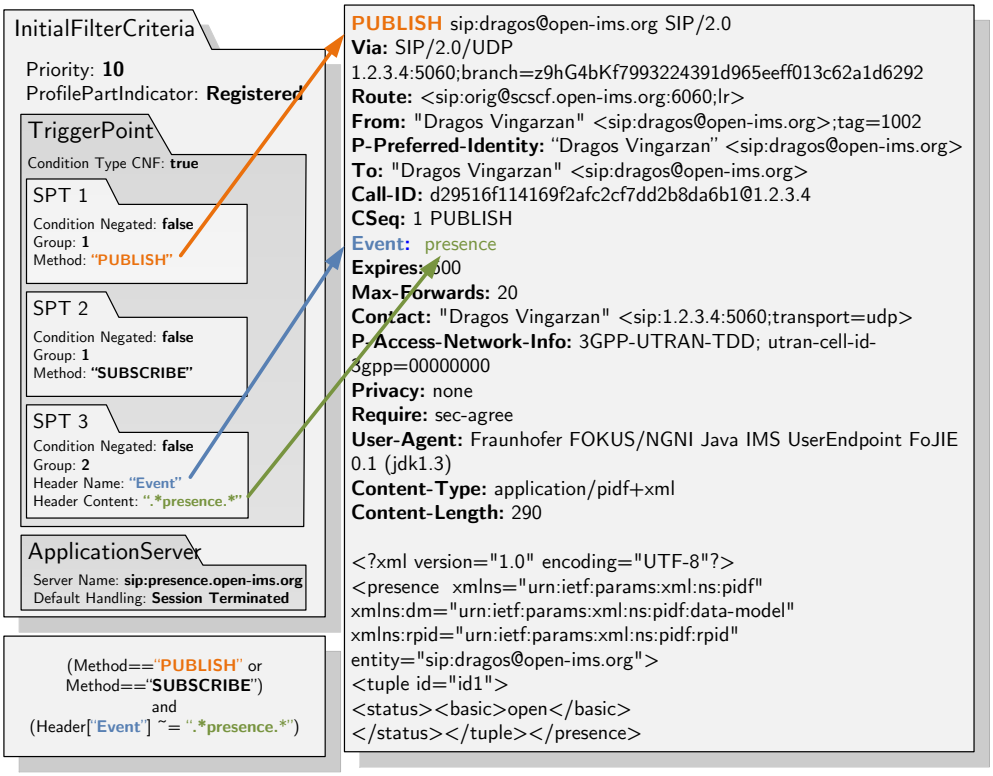


Figure 5.23: iFC Matching Example on a SIP Request

A particular case, as shown in Figure 5.25, is obtained when the triggered AS responds with a final response. In this case filtering for subsequent matches can not continue and the response is directly returned back.

5.4.3 AS Modes of Operation

Once the AS receives the SIP requests as forwarded by the S-CSCF on the ISC interface, it can decide to act in one of the following modes:

- *Terminating UA Mode* – the AS processes the request and sends back a response. Filtering then stops and the response is sent back to the originating node.

- *Redirect Mode* – the AS processes the request and sends back a response indicating a redirection. Filtering stops and the response is sent back to the originating node. The originating node will resend the message to the new target, potentially restarting filtering and triggering.

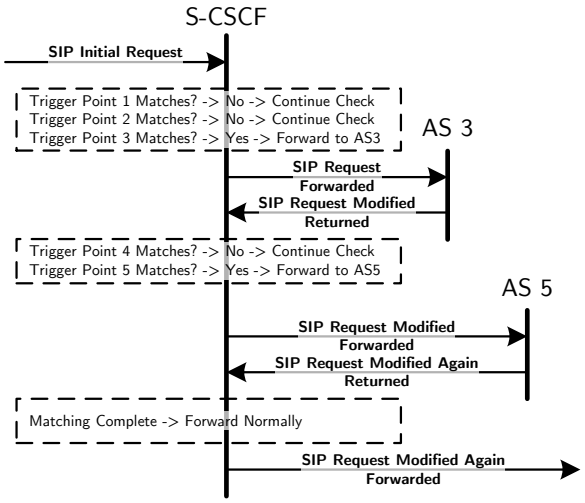


Figure 5.24: ISC Message Filtering Cycle Example with Multiple Request Forwarding

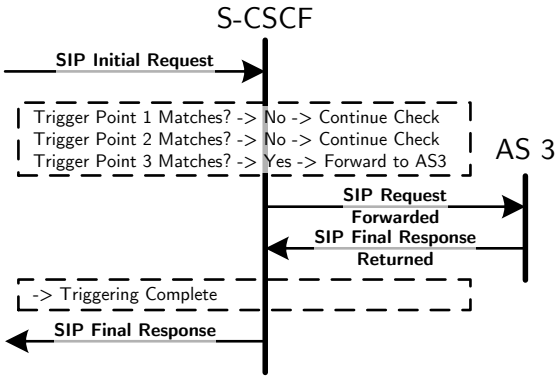


Figure 5.25: ISC Message Filtering Cycle Example with AS Sending Final Response

- *Originating UA mode* – the AS can also originate requests itself and send them out. It is though recommended that these would not be sent directly to the S-CSCF, but normally routed through the I-CSCF, which will in turn find the proper S-CSCF to forward to. This mode of operation enables in effect a bidirectional ISC interface.

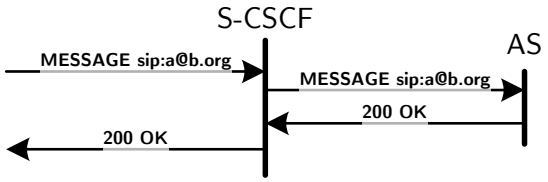


Figure 5.26: ISC with AS in Terminating UA Mode

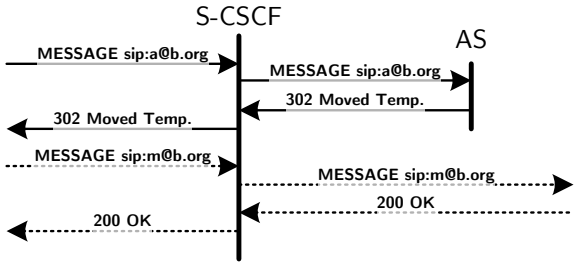


Figure 5.27: ISC with AS in Redirect Mode

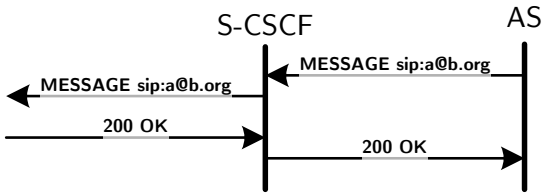


Figure 5.28: ISC with AS in Originating UA Mode

- *Proxy Mode* – the message is processed on the AS, modified and then forwarded back, following normal forwarding procedures, to the S-CSCF. Filtering continues and further ASs are potentially involved.
- *B2BUA Mode* – the message is processed and the AS creates additional leg(s)

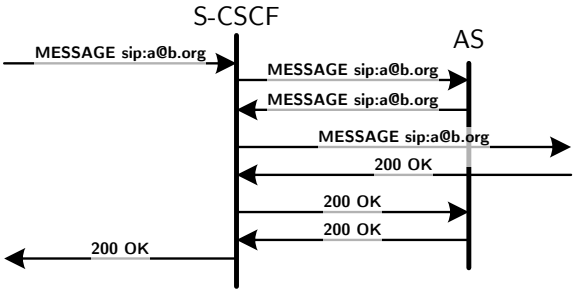


Figure 5.29: ISC with AS in Proxy Mode

by originating similar requests, yet on different transactions. The original request is answered in UAS mode and the information is passed back and forth between this side and the other leg(s). Unlike the Proxy Mode, in most situation filtering ceases and a much higher degree of separation will exist between the upstream and downstream signaling viewed from the AS perspective.

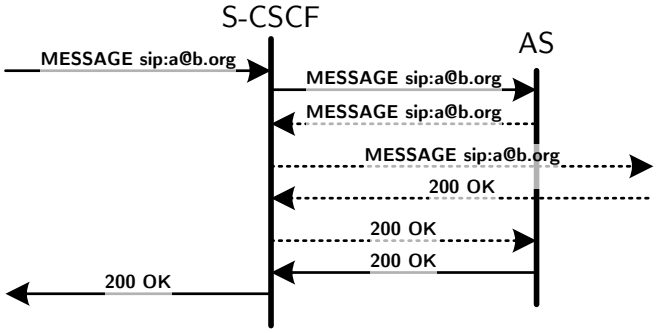


Figure 5.30: ISC with AS in B2BUA Mode

5.4.4 Signaling Routing

To enable the S-CSCF filtering as well the 5 modes of operations defined above for the AS, regular SIP routing can be employed. While the standards indicate many alternatives which use for example distinct IP addresses or ports to relay context information, most such solutions would not scale well. Instead, routing header

parameters can be used to store and indicate back context information.

When forwarding requests to the AS, the S-CSCF will fill 2 `Route` headers:

1. One containing the AS SIP URI as indicated in the `Trigger Point`. This is used to route the message from the S-CSCF to the AS, can include indicators useful for the AS to identify the service to be applied and will be eventually removed if the message is to be forwarded.

2. A second one containing the S-CSCF SIP URI, which will allow the AS to easily forward back the processed message to the S-CSCF for further processing. In this header the filtering context information can be included. To keep the system lean, a compliant stateless system was been designed and implemented by the author as part of his work preceding the OpenIMSCore formal project [133].

Whenever forwarding SIP requests, the AS will add itself to the `Via` headers' stack, ensuring as such that the respective transactional responses will visit it.

As previously mentioned, when an AS needs to remain involved in subsequent requests, it will trigger and filter on the initial dialog creating request, even if there are no internal operations to be applied. However, a `Record-Route` will be added, potentially indicating the context. All subsequent requests will be forwarded then without triggering through the same AS, enabling the service's functionality to complete.

Standalone or initial dialog request messages originating in the AS should not target potentially learned S-CSCF addresses. Instead, they should be sent simply to the destination domains, from where the I-CSCFs will correctly identify the needed S-CSCF for message processing[38]. This ensures a proper CN elasticity, enabling dynamic S-CSCF allocations as required by network management operations, while also keeping the AS platform stateless and independent of the CN instantiated topology.

To conclude, messages are routed on standard SIP routing mechanisms. Special care needs to be taken on originating messages from the AS just as to keep the AS and CN operations lean without complex inter-dependencies.

5.4.5 Summary

The S-CSCF plays a central role in serving signaling. First it downloads the Subscriber Profiles from the HSS. Then it applies the respective filters on signaling to identify matching `Trigger Points` in the standalone or initial dialog creating requests. On matches the requests are forwarded to the respective AS for further processing. Filtering can potentially resume if the requests return from the AS, hence multiple services can be provided on each signaling transaction. The AS platform has a complex model of interaction with the CN, enabling complex services to be executed and potentially defined in the future. Routing of messages is provided by standard SIP `Route` header mechanisms.

[38] For a detailed overview, please see Figure B.1

5.5 Service and Subscriber Provisioning

Within a CN architecture, the subject of Operations and Management (O&M) is quite complex, covering a wide area of topics as identified in the 3GPP TS 32.xxx series: Configuration Management, Subscriber Management, Performance Management, Fault Management, Security Management, Inventory Management and so on. The topic of Service Provisioning falls in this taxonomy mostly within the Configuration Management, while that of Subscriber Provisioning is of course within the Subscriber Management one.

In real-life deployments, both of these topics would be approached with suitable Operations Support System (OSS)/Business Support System (BSS) Network Management Systems (NMSs). These represent an entire topic in themselves which would interact with the network equipment in the CN through, in the case of IMS, standardized interfaces referred to as Integration Reference Points (IRPs). Such complex systems are critically required for large networks serving millions of subscribers with complex services, as Mobile Network Operators (MNOs) run today.

For test-bed purposes however, the requirements can be considerably relaxed. In most situations tens/hundreds of subscribers provisioned are sufficient, with similarly low number of services. In certain corner situations, as for example performance benchmarking or field trials, of course thousands or even millions of subscriber instances have to be provisioned, yet this could be approached individually by considering the uniformity of the subscriber bases and using for example script-based approaches of automation.

The minimalistic approach would be to provide direct access to the DBMS which stores the service and subscriber provisioning information. As the IMS architecture is in itself new, most of the inner concepts would be difficult to understand and properly be provisioned by the novice user. A GUI would then be highly valuable in order to guide the system administrator through the data complexity for provisioning, offer valid options at every step and enforce correct input.

Both the service as well as the subscriber provisioning information is conveniently stored and access by the executive elements of the network in and from the HSS function. Hence the HSS requires an accordingly designed provisioning interface to interact with the back-end DBMS as well as with the running HSS front-end instance.

5.5.1 Service Provisioning

The Service Provisioning is mostly executed by defining and managing the `Trigger Points` in the HSS database.

The `Trigger Points` contain two main parts:

1. the Initial Filter Criteria (iFC), which present the SPTs with the respective parameters, to be checked against the signaling passing through the S-CSCF;

2. the Application Server (AS) information, indicating where to forward matching signaling and how to handle the routing further.

The AS functionality itself is both outside the CN domain and also abstract in itself, such that new features would be easy to introduce in the future. The Service Provisioning then can not be resolved in the CN due to the abstract nature of the services. Each newly defined service will have then to cater for its provisioning, when defined and implemented.

Returning to the iFC and AS information provisioning, this is straight forward: a user interface is required to guide even the novice administrator through the provisioning of new applications, by creating first the filter expressions and parameters and then by allowing configuration of the target AS information. The process is straight forward and does not require further specification at this point.

5.5.2 Subscriber Provisioning

The Subscriber Provisioning is similarly executed, on the same functional element as the Service Provisioning, the HSS.

The information to be managed here has the following facets:

1. IMS Subscriber Identity (IMSU), which represents the commercial subscriber concept. In simple words, it represents a higher level grouping of the lower level technical identities, which provides eventually the business logic of joining together multiple communication endpoints into a single billable entity. The provisioning of an IMSU represents the creation of a subscriber context which would tie multiple identities to a single commercial user or entity.

2. IMS Private User Identity (IMPI), which provides the secure authentication identity. In technical terms, this represents the AuC side of the similar ISIM stored credentials for authentication during the IMS network attachment. As associated subtypes here the following are identified: secret key `K`, authentication management field `AMF`, operator secret `OP`/`OPc`, sequence number `SQN`, allowed authentication methods and so on.

3. IMS Public User Identity (IMPU), which provides the IMS service level identities, used by communication end-points to address each-other. These are SIP or tel URIs. Associated information elements here are: the allowed IMPIs to be used for authenticating it, the allowed roaming networks, charging parameters and so on. Each IMPU can also be provisioned together with other IMPUs into implicitly registered sets, creating subsets of identities with subscriptions.

4. Public Service Identity (PSI), similar to the IMPU, yet as it represents a service instead of customer communication device, it can be considered a subset of information in regard to the IMPU and treated similarly.

5. Various mappings between the aforementioned identities, such that they can be associated and used properly in regard to each-other.

6. Service Profile, as a mapping between the IMPI/IMPU identities and the `Trigger Points`. This represents the association and eventual customization of services (as provisioned in the previous sub-section) per individual subscribers.

Besides a human-oriented user interface to provision the information above, automation is beneficial here when considering the need to provision large subscriber bases, with usually uniform data, within short amounts of time. A template based approach is to be considered, which would include the logical structure of the data storage and directly populate the respective database, with a minimal effort from the human operator. Considering the use of the common MySQL DBMS, the requirements translate to a script utility to generate and then execute SQL queries populating the database with information.

5.5.3 AS Access to Generic Service Data, Subscriber Information and Profiles

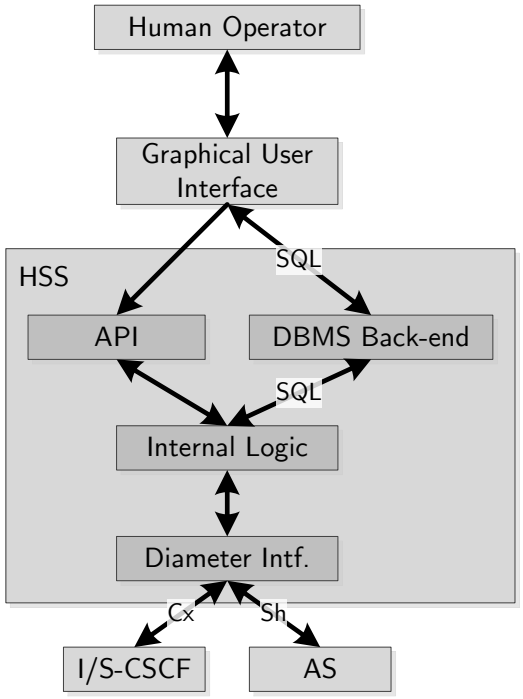


Figure 5.31: Provisioning GUI and Operational Interactions in the HSS

Additional service provisioning has to be considered for the access of services to the generic database that the HSS represents, on the AS $\longleftrightarrow$ HSS (Diameter) (Sh)

reference point. Each AS must be defined with its identity and security parameters on this Diameter interface. As multiple operations are possible and also multiple information sources can be accessed, an Access Control List (ACL)-like system must be provisioned, with various parameters for each type of operation or information that is to be accessed.

Besides provisioning tasks, a GUI represents an opportunity for a few more functional and test-bed management necessities. For example the GUI can also display information which normally does not need to or must not be changed by the human operator, like the current registration status and additional data saved for a subscriber. This will enable a monitoring system, greatly improving the debug capabilities of the platform, allowing an easy live view and a debugging interface into the inner state of the system.

Yet another point of interest for the user interface is to enable through it interactivity with the system. A series of operations as for example registration termination or live updates of service profiles require the presence of human-based triggering mechanisms, which are greatly enhanced by providing selection and validation of input information.

5.5.4 Summary

In the scope of test-beds, the service provisioning as well as the subscriber provisioning, debugging/monitoring and system interactivity can be provided through a GUI. This needs to provide a facile process, accessible even to the novice network administrator, guiding him/her through the required provisioning steps and validating the information.

The provisioning task can be almost entirely localized on the HSS function, as most of the information is either stored in its back-end DBMS or accessible through API calls.

5.6 Specification Conclusions

The present chapter has listed and specified the functional details of an IMS prototype. As following on the Chapter 4 – Design of the Open Source IMS Core, the specification followed on the set blue-print, providing low-level details and hence filling the space towards the implementation. The information here then serves also as an extension of Chapter 2 – State of the Art, providing the missing details to not fully, but more closely describe the technical background an expectations of the OpenIMSCore project.

Chapter 6

Implementation of the OpenIMSCore

The present chapter follows on the Specification of an IMS CN Prototype Implementation and exposes details of the software implementation, following the evolution of the project, from a small S-CSCF-like service triggering engine processing SIP signaling, to a comprehensive implementation of the specified main functionality.

The last part of the chapter describes the Open Source aspects of the project. Here the community processes are explained, providing insight into how this process has enhanced the project, allowing it to grow beyond a single lab experiment, by accepting contributions from a large number of co-interested researchers.

The implementation process of the OpenIMSCore is presented here, as a blueprint example of the process. The execution started small and evolved into a successful Open Source project, breaking new grounds in the telecommunication domain as one of the first such test-bed comprehensive CN prototypes. Mainly followed here are the technical aspects of the implementation, providing an insight into the inner workings of the prototype as well as how and why the technical decisions have been taken. The Open Sourcing aspects are deferred to the next section as they represent in themselves a distinct item of interest for the overall dissertation targets.

6.1 Platform details and Initial Implementation Plan

The OpenIMSCore project's ancestry can be traced to two main starting points: first the SER project [27] and second the author's diploma thesis [133] on the topic of iFC triggering.

First the SER project was developed as an ultimate flexibility SIP proxy platform, allowing for unlimited administrator control on how messages would be processed and routed, while also providing a deeply optimized platform, exhibiting best-in-class performance. When considering that the main IMS functional components, the CSCFs are in fact SIP message processors and routers, the SER platforms was and still is today a prime candidate for such a prototype implementation.

The IMS concepts were first formulated in 3GPP Release 5. The association with SIP as its signaling protocol as well as the demand for practical experimentation and feasibility studies were the catalysts for the author's work to prototype the first S-CSCF concepts. Around 2004/2005, at the Fraunhofer FOKUS Mobile Integrated Services (MOBIS) Competence Center (led by Dorgham Sisalem), the student investigation task force has been formed to start researching if and how the main concepts of service triggering would be feasible. The student diploma theses projects were centered around the practical implementation and experimentation with a prototype S-CSCF and HSS capable of filtering SIP signaling and redirecting messages on a subscriber-to-subscriber basis to various AS.

The author was back then in charge of implementing the download of the Service Profile for individual subscribers at a SER based registrar and the application of the `Trigger Points` on signaling. A colleague, Bogdan Pintea was in charge of implementing a suitable Diameter protocol library to provide the reusable and realistic interface with a prototype HSS function. After about 6 months the first prototypes were available and both students have successfully obtained their engineering degrees on the subjects after completing the respective theses [133, 185].

At the beginning of 2005, in a process of joining synergies with the Fraunhofer FOKUS Next Generation Network Infrastructures (NGNI) Competence Center (led by Prof. Dr. Thomas Magedanz), the S-CSCF prototype has been validated and integrated with the HSS developed by Peter Weik [13] and Andre Charton [187]. This marked the beginning of the OpenIMSCore project, as its base components, the S-CSCF and the HSS have been put together for the first time.

The initial S-CSCF implementation was represented by the SER base (implemented in `C`) configured with a script to handle signaling and act as a SIP registrar[1].

The first IMS implementation step was to enable authentication through the HSS, as part of the IMS AKA authentication mechanisms, with the help of the MAR/MAA operations. For the client device, a modified version of the KPhone [137] UE implementation was used as AKA authentication was not available in the SIP clients at the time.

Upon successful authentication the S-CSCF was also using the SAR/SAA operations to retrieve the XML-encoded `IMS Subscription` from the HSS. Then basic filtering was applied to the messages and the matching messages were exchanged over the ISC interface. The implementation is exhaustively described in [133].

The HSS was implemented in `Java` and while initially it used a Java Native

[1] Accept `REGISTER` requests, perform authentication and add the `Contact` information to the registrar information; replace *Request-URI* with the `Contact` information when terminating other signaling.

Interface (JNI) layer to interface to the C++ based OpenDiameter [188] implementation, this was shortly after replaced with an own implementation, purely developed in Java, the JavaDiameterPeer [189] and as such made much more flexible in development and usability.

At this point, without the rest of the IMS functional elements, the project was nothing more than a proof-of-concept demonstrator. Fortunately this was sufficient to estimate the potential and feasibility of a full blown IMS prototype implementation.

The work continued within the Fraunhofer FOKUS NGNI Competence Center, with a first phase driven by a need to evaluate NGN performance with practical benchmarking, followed by a second phase of Open Sourcing, community participation and growth to the full scopes described here, as a test-bed bridge in the telecommunication world, between the academia and the industry.

6.2 Benchmarking as Initial Driving Force

The standardization of the IMS architecture has created an industry momentum towards evaluation of the architecture. The foreseen potential around 2004-2006 regarded the IMS CN as a central part of the global networking architecture of the future. The CSCFs were potentially representing the signaling routers supporting the backbones of the future universal communication platforms.

Such a future application potential had a ripple effect on the industry as it set a target for the future equipment to perform. The target represented a major and clear-cut evolution of the telecommunication platforms from specialized and legacy-bound equipment towards a new generation, using for most part universal hardware and software platforms and fully embracing the latest Internet paradigms, concepts and protocols.

Understanding and planning such a major change can not be performed on just theoretical models, but especially the feasibility and performance characteristics must be first fully understood, such that the industry can properly craft their implementation and deployment plans. Looking from the benchmarking perspective, the manufacturing side of the telecommunications industry needed early evaluations of the performance profiles, such that investments could be started in the technological keystones.

The realization of the OpenIMSCore project is in fact related directly to such an event. A group of engineers from Intel® Corporation, have approached the Fraunhofer FOKUS NGNI [190, 191] team on the topic of performance in IMS. The starting point was the reference performance level of the SER prototypes as a SIP-proxy [154], while the main signaling routers in IMS are, simply put, specialized SIP-proxies. An IMS prototype based on SER was foreseen then also as representative in performance as well as for characterizing the load and resource consumption from the underlying computing platform.

The target of this collaboration was first of all to understand these new re-

quirements and work-loads, such that the performance of implementations could be measured in an industry-wide representative manner [192]. A SIG has been formed, which gathered a large audience from the industry and a performance benchmarking standard was set as a target. To ensure that the peculiarities of the new IMS CN functional elements would not be missed, the SIG has encouraged a prototype implementation for both the System under Test (SuT) as well as the Test System (TS), with which the newly introduced evaluation methodologies would be validated. The OpenIMSCore project was set then to satisfy this need, by implementing a close-to-real-life level of functionality, which allowed for representative results to be obtained and the performance benchmarking standards to be validated and published [12, 152, 153].

Another benefit of these efforts was that the vendors were able to start identifying the potentially problematic points in the architecture and accordingly start making early investments into critical hardware and software components[2]. While the entire implications of these performance benchmarking definition and evaluation efforts would be out of scope to be described here, the overall effort has both educated [193] and still provides today an industry representative reference.

Returning to the OpenIMSCore project, the benchmarking suitability requirements have dictated that at least a critical mass of IMS functionality had to be included, as to ensure that the performance evaluation results would be representative for the entire architecture and not just for a limited subset of functional procedures.

In practical terms this meant that once the project has been released under an Open Source license in 2006 [194, 195], it was almost immediately embraced by the community as a functional reference too, without requiring exhaustive software changes.

In effect the implementation, even though initially motivated by the performance evaluations, has not been undertaken as to provide a simulation or emulation, but as to target a realistic implementation of the standards, with certain limitations for low-incidence procedures as well as availability, security and stability.

6.3 Software Architecture

The present section analyzes the implementation and software architectures of the OpenIMSCore functional components. As this process took place over a significant number of years, the presentation follows on the implementation time-line which highlights better the individual decisions and gradual evolution.

[2] E.g. the P-CSCF IPsec security associations were posing important scalability challenges. Historically the accent was on obtaining very high traffic throughput on a very low number of security tunnels. In IMS, as each UE builds several such associations and the traffic on each is relatively low, yet requires low latency, a different type of load was created. Redesigns of the software were required, while on the hardware side newer enterprise-class systems started to feature hardware accelerated encryption and hashing capabilities.

6.3.1 Starting Up - Diameter, ISC, S-CSCF

Back in 2004 when the first intentions to prototype and practically evaluate the newly specified IMS architecture emerged, a VoIP significant revolution was in full swing. Not only enthusiasts were starting to make use of the new voice capabilities through the Instant Messengers of the day, but also telephony carriers and operators were starting to deploy H.323 capable gateways and client devices or even replace parts of their infrastructure by reusing IP networks as a cost-cutting measure.

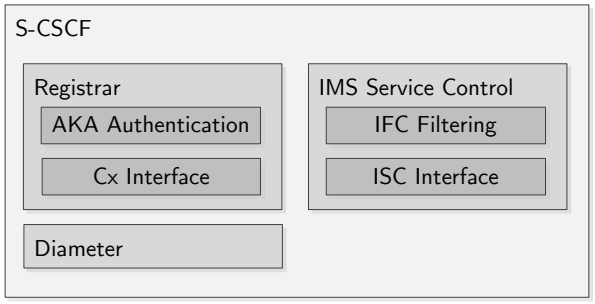


Figure 6.1: Functionality of the Initial S-CSCF Prototype

SIP-based alternatives to the telecommunication soft-switches started to gain high-capacity and carrier-grade capabilities. As an implicit example, the SER Open Source software developed at Fraunhofer FOKUS was obtaining world-wide recognition as a reference for both performance as well as flexibility in application. IMS uses similar concepts, partially also protocols and procedures to those of early carrier-grade VoIP architectures. A few differentiation points were though essential in taking the leap from a SIP-only architecture to one that would more likely resemble the standards set by IMS.

- *Standardization of the subscriber information database* – the HSS concept was introduced as a successor to the HLR. The HSS is separated through standard interfaces from the signaling processors, providing the information on demand, through a front-end standardized mechanism used to police the access, plugged into database back-end used as information storage.

- *Use of the more capable Diameter protocol* – the aging AAA protocol of choice, RADIUS was starting to show its age and limitations on satisfying the new requirements and applications. Diameter, although quite similar in some of the base concepts, comes with significant new functionality, improved security and nevertheless a significantly better extensibility towards satisfying future applications.

- *The use of improved mechanisms for authentication* – as IMS was started in the mobile domain by 3GPP, a similarly secure and capable authentication mechanism as for the then-flagship UMTS was required. Additionally, as in the mobile domain the subscriber identity is decoupled from the equipment, the mechanism was to be of course SIM based.

- *Standardization of signaling information elements and procedures* – most of the VoIP implementations were plagued by ambiguity issues. SIP as a signaling protocol was designed to be especially flexible and extensible, yet an international reference was hardly standardized, requiring equipment to often have to implement multiple alternatives for the same procedures in order to provide good interoperability. Even more, information elements were also not strictly standardized in semantics. IMS has both standardized the procedures through clear definition at each functional element and on interfaces, as well as introducing additionally new information elements (i.e. SIP headers) with strict rules on meanings and usage.

- *Clean-split architecture with separation of base functionality and services* – the IMS architecture, although clearly targeting a replacement of the legacy telephony services, did not target itself the standardization of the services and applications provided on top. The intent was rather to provide just a powerful interfacing and a set of base functionality to be re-used and re-combined in order to create new services as required in the future. The base signaling processing provided by the CSCFs was well standardized as mentioned above as CN capabilities, while the applications domain remained full of alternatives.

- *Standardization of the service control interface (ISC) between the CN and the AS platforms* – while the base functionality provided by the CN was standardized, also the interfacing point had to be equally fully specified in procedures. The signaling exchanges with the applications domain were built such that clear procedures could be used with future platforms. Similarly SIP based, these standardized unambiguous signaling flows and information elements to be used for service control.

- *Separation and standardization of SIP signaling processors based on functional requirements* – even before IMS, the need for specialized SIP signaling processing nodes was identified: security gateways, load balancers, service processors, gateway interfaces and so on. The P-CSCF, I-CSCF, S-CSCF, MGCF and so on, introduced clean functional separations, with concrete interfaces between them, in the spirit of providing interoperability between components from different manufacturers.

- *Mandatory use of secure interfaces and procedures* – with Internet-based VoIP networks security was lax initially, with early adopters having to ensure themselves their privacy and protection against attacks. Within the telecommunication domain the operators must be in firm control of their networks' security,

protecting their subscribers from security vulnerabilities and threats. The use of secure protocols and procedures was mandated, while the UNI and NNI have been clearly specified in their communication with other domains or potentially insecure networks.

The important differences from a typical SIP-based VoIP network service and a SDP based on IMS summarized above cover most of the new architecture. Yet perhaps the most innovative and the pylon of the new architecture was the IMS Service Control (ISC) interface and concept.

Accordingly, the implementation of the service triggering capabilities based on subscriber individual profiles, as well as the interfacing with the service platforms were first realized as parts of the first S-CSCF-like component. The starting point was the SER implementation, which could be tweaked and configured to act in a somehow similar manner. The first parts developed as part of [133] were an enhanced SIP registrar in SER capable of storing the iFC triggering points and a message filtering SER module capable of identifying matching signaling and routing it to and from the AS.

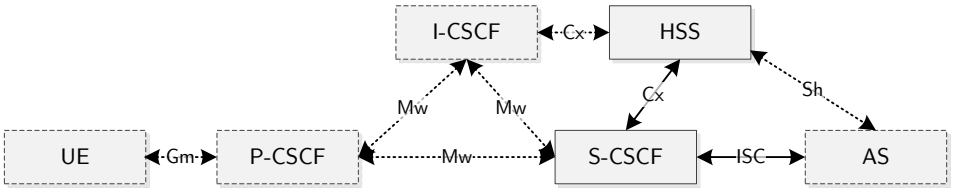


Figure 6.2: Functional Overview of the Initial IMS CN Prototypes

The Service Profiles were specified to be stored in the HSS, which prompted the implementation of a Diameter protocol stack [185] in a parallel thesis. Upon completion, the protocol has been integrated within the S-CSCF-stub and the procedures were changed to free the S-CSCF from the profile storage requirements and enable the dynamic download of the subscriber information as indicated by procedures. The correspondent HSS stub was also realized, yet at this point, with only a subset of the S-CSCF Cx functionality.

6.3.2 HSS

However limited in functionality, the initial students work has validated the feasibility of a full-blown implementation, by allowing for first-time experimentation with IMS as a base platform for services. Attracting the attention of various industry R&D laboratories, soon enough sparked the demand for replacing the HSS stub with a proper prototype implementation.

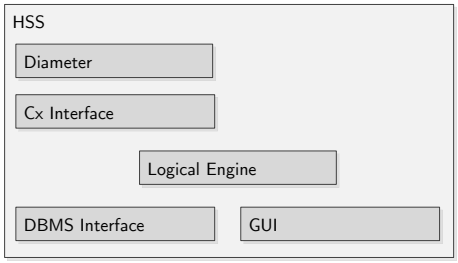


Figure 6.3: Functionality of the Initial HSS Prototype

For fast-prototyping, ease of development and future flexibility, a `Java` application has been designed and implemented. This interfaced on one side with a MySQL database back-end and implemented first a Cx interface towards the S-CSCF, later on also an Sh one towards the AS[3]. While initially a JNI interface to an existing Open Source `C++` Diameter stack was used, it was shortly replaced with a native implementation [189]. The resulting prototypes were branded as the Fraunhofer Home Subscriber Server (FHoSS) [196] component of the OpenIMSCore.

This second phase in the implementation was especially beneficial for fully opening opportunities towards the application domain. Not only the ISC interface was available, but the Sh provided also the integrations with the AS platforms for exchanging subscriber information, enabling full-blown IMS services to be prototyped and trialed.

During its lifetime, a major overhaul has been performed, as the 3GPP Release 7 [197] has represented a major upgrade in functionality, especially in regard to interfacing capabilities with AS platforms. Incremental smaller upgrades have been also performed for the successive standards versions, as for example to support additionally fixed or broadband access networks, beside the mobile ones.

6.3.3 I-CSCF, P-CSCF and Security

The combination and success of the S-CSCF to provide a stub platform for developing new services, has increased even more the interest for working with and further enhancing the OpenIMSCore platform. As part of the IMS Performance Benchmarking SIG activities, the additional P-CSCF and I-CSCF functionality were needed.

Similar to the S-CSCF implementation based on SER and in fact even sharing common utilitarian and protocol stack modules, the I-CSCF was built by extending the Cx interface implementation and by adding NDS checking and enforcing capabilities.

The P-CSCF required more efforts and a different approach. Although featuring

[3]Further BSF $\longleftrightarrow$ HSS (Diameter) (Zh) interface, application server enhancements and a web-based GUI have been incorporated.

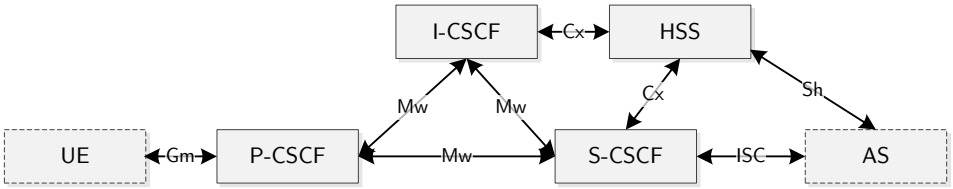


Figure 6.4: Functional Overview of the Principal IMS CN Prototypes

an internal registrar functionality, the one here was completely re-implemented as being a reversed one: information was indexed by `Contact` information instead of IMPU AoR used at the S-CSCF.

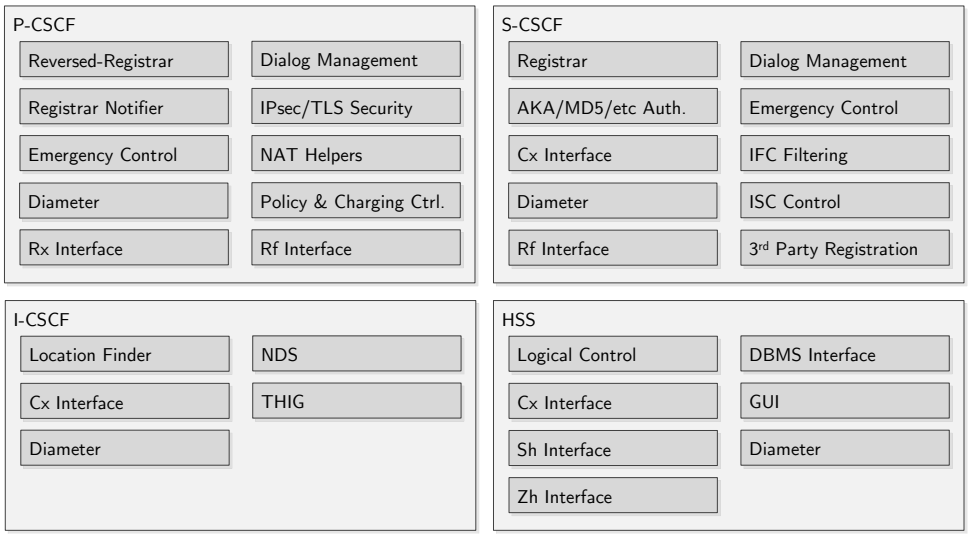


Figure 6.5: Logical Overview of the Principal IMS CN Prototypes

With these two new CSCFs the main IMS architecture has been completed. Complex end-to-end scenarios could be tested, enabling a comprehensive IMS testbed. Later in the project's lifetime additional components were contributed, to enhance the functionality and add integrations with other systems. Some of these additionally developed functions were: the MGCF to integrate with media gateways, the BGCF to provide a smooth path for subscriber migration and selection of NGN/pre-NGN servicing network, the E-CSCF to support attachment of UE clients in emergency modes with the associated emergency services capabilities and so on.

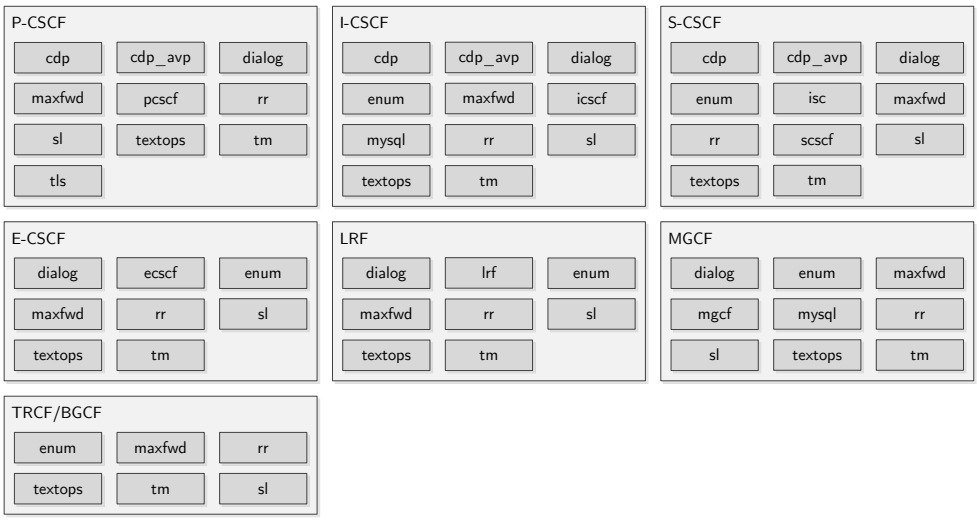


Figure 6.6: SER Modules Used in the OpenIMSCore Prototypes

6.4 Open Sourcing

Once the basic architectural prototypes have been realized, the OpenIMSCore project started to be used in more and more R&D project studying IMS. To keep the moment alive and to boost even further the covered functionality, an Open Sourcing strategy has been analyzed first and then implemented.

In 2006 the initial targets of understanding IMS performance were already well underway. Benchmarking tools were used to evaluate the performance levels for signaling in the IMS CN, with sufficient levels of functionality to ensure a good representation [198, 199]. As a next step, the project needed validation, enhancements and acceptance as a reference implementation for test-beds. Without it, the performance results could always be challenged as too far and hence non-representative with regard to future real-life products.

The present section seeks to highlight the open-sourcing process which represents in itself a second and more significant part of the OpenIMSCore project's life-time. Through it not only the reference status has been achieved, but also the project has been significantly improved and grown. Even as R&D interest in it eventually slowed, as IMS moved into deployment and exploitation phases, the seeds have spawned into related projects used even in commercial exploitations.

6.4.1 Starting a Community

In the research community of IMS many theoretical evaluations on functional, security and performance aspects of the new architecture were performed. To have also practical validations of the studies, researchers often relied on make-do simulation

by using existing SIP proxy and test-tool implementations, adapting them mostly by re-configuration to IMS-like topologies. While these initial test-beds allowed for some basic testing, their main disadvantage was that key new concepts were difficult to add within reasonable individual laboratory R&D budgets, hence the results themselves were not truly representative.

Multiple such efforts started from the common platform, the Open Source SER project. With this it is relatively easy to manipulate the behavior of the signaling routing, by simply re-working the standard configuration files.

Observing this fact, our group sent out invitations to joint discussions on such test-bed tools. Following, the OpenIMSCore project has been showcased, demonstrating that a systematized approach has been taken and most of the missing critical IMS concepts have been implemented already, with the clear intention of looking for joint synergies.

Taking a step back and looking at how Open Source communities evolve two important traits must be considered:

1. *Project forking*[4] is a natural and often beneficial process, however due to community splits the resulting tracks slow-down in development and continue so for a significant duration. Considering that the targeted IMS CN is not as universally needed and used as for example a DBMS project (e.g. the MySQL project has numerous forks, each very strong in itself: Drizzle, MariaDB, Percona Server), also the number of developers and users is orders of magnitude lower. Accordingly, a strategy avoiding forking at least in the first years until a critical mass would be achieved was crucial.

2. *The single-survivor typical behavior of Open Source communities*[5] is also a beneficial trait as it provides a single consolidated community. This can cause eventually forking as described above, yet understanding the principles is crucial for ensuring that a critical community mass would be achieved.

Both points above were guiding lights for the community organization strategy.

[4]Forking an Open Source project happens when a non-resolvable conflict on the road-map and/or implementation strategy occurs in the community. As the source code is shared, a group can easily split from the main community and continue development on a different path, gathering its own agreeing community of developers and users. This happens for various reasons, some of the most common being for example a conflict in the targeted functionality, or a different view on stability vs. functionality, or simply a major evolutionary change to be taken. Un-forks are also possible when the difference points have been resolved or when simply one of the branches failed.

[5]Multiple Open Source projects, which provide the same or similar functionality tend to eventually converge. As there is no up-front cost which would motivate users to choose a worse performing project over another, the project which fares even slightly better, on the basic directions like functionality, stability and community support, would create a cascade loop effect around it: it attracts more and more of the community, which leaves the less performing alternatives with less force to successfully compete, eventually starving of resources. A classic example is provided by the Hurd and Linux kernels[200], where even if arguably the Hurd kernel had initially a bigger potential, the Linux one became the de-facto standard for GNU systems as it managed to be the first to gather and subsequently maintain higher numbers of developers and users.

This is described in more details in the following sections, after the release of the project as Open Source is detailed.

6.4.2 Launching as Open Source

In order to maximize the potential for the project to successfully gather and build the much needed community, a launch event has been carefully planned. The moment has been chosen based on estimations on the potential that an Open Source tool-set would bring to IMS test-beds and experiments around the world.

The right moment has been found at the 2nd International FOKUS IMS Workshop 2006 [194, 201]. In a presentation followed by over 120 international participants, the project has been exposed and even lively demonstrated. Additional practical demonstrations of the platform's readiness for test-bed use have been made through 3 different applications, all using the OpenIMSCore implementation as an enabling IMS CN. The event has been a big success, establishing and kick-starting in force the OpenIMSCore community.

Figure 6.7: Launching Presentations for the OpenIMSCore Project

Following the launching event the `http://www.openimscore.org` website has been published together with the projects source code repository (hosted on the `http://www.berlios.de` platform) and various guides and manuals on how to use it.

The central points of the community in the next years have been formed around the email lists of the project. A general users one and two more technical topics lists

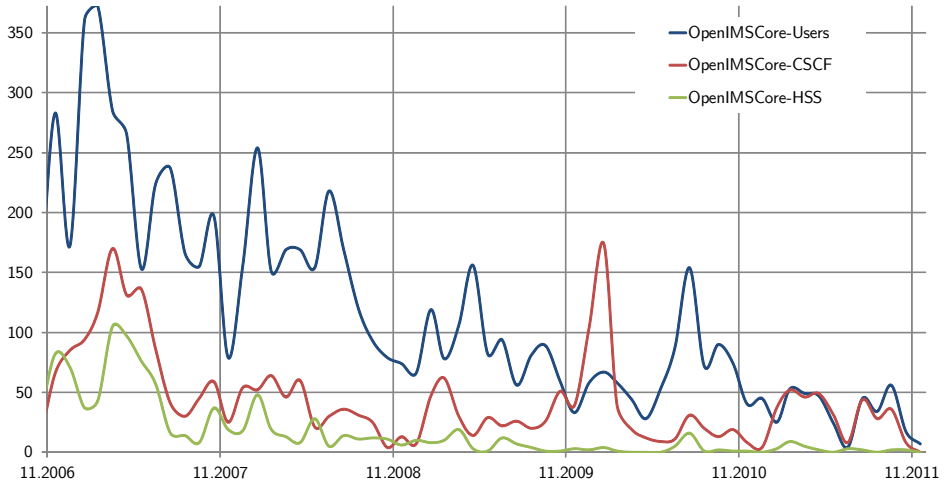


Figure 6.8: OpenIMSCore Messages Posted on Mailing Lists per Month

(CSCFs and HSS) have been opened and quickly gathered hundreds of subscribers. This represented a vivid forum for discussions (see Figure 6.8 and Figure 7.4) not only on the CN implementation itself, but also for starting new adjacent Open Source project, exchanging know-how and experiences from running such test-beds, as well as a general IMS architectural learning and improvement platform.

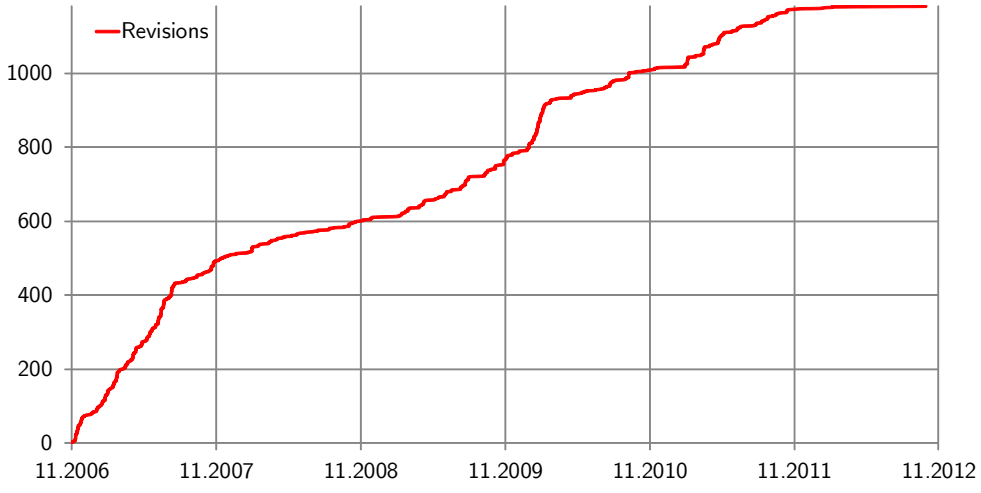


Figure 6.9: OpenIMSCore Development Progress in Number of Source Code Revisions

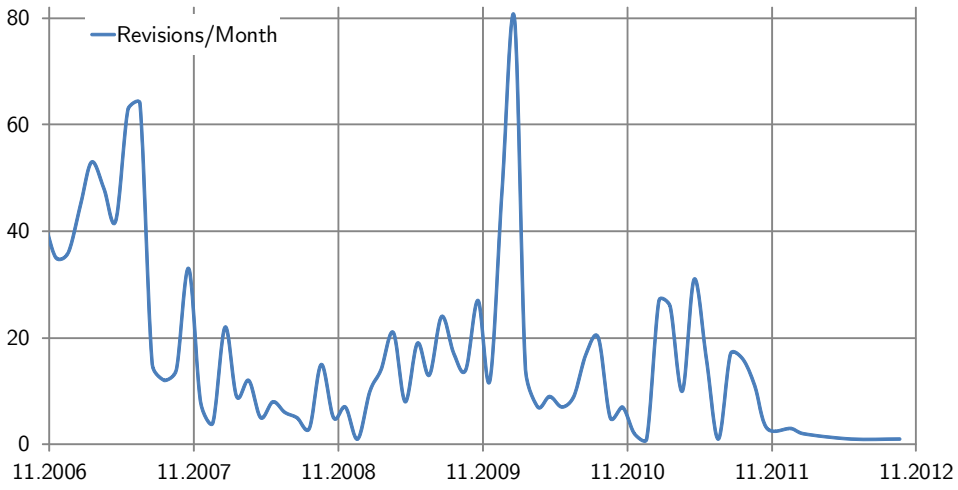


Figure 6.10: OpenIMSCore Development Progress in number of Individual Repository Changes per Month

Development wise, the first 8 months have been very productive with a steady pace of development of new features and improvements – see Figure 6.9 and Figure 6.10. Following this period the development continued at a more maintenance-like pace, with noticeable development spikes towards the end of 2009 and the beginning of 2010, when a significant push has been made to prototype emergency services. Further on, yet another noticeable event happened in 2011-2012, when the CSCF implementations have been complemented with Policy and Charging Control (PCC) support, especially as part of the integrations with the new EPC architecture[6].

6.4.3 Community Organization and Processes

To analyze the community organization, a look back at the Open Source principles, previously specified as guiding points, is required.

Towards avoiding forking in the early stages, the strategy was to create an open-access policy, where anyone interested in participating was allowed to do so. The rules communicated to the community were that anyone was free to change anything, yet they will have to directly answer to the community in case a feature would be broken or eliminated. The easiest and best way to communicate this was to present the source code repository and the attached project web-site and mailing lists as wiki-style shared resources [203]. This policy proved to be quite helpful, with just a few notable, albeit surmountable, disadvantages:

- A large number of users requested accounts. While all accounts were equally

[6]The Fraunhofer FOKUS team took on the task of starting the next major R&D project, the OpenEPC project[202].

created and had mostly equal write/change permissions, it was hard to separate the occasional spamming user, or even worse spamming robot, from the real potential contributors. Automatic account approval was suspended in favor of a manual confirmation upon request to one of the existing accounts holders. This did create unfortunately an artificial barrier, yet a necessary one to avoid unrelated or even offensive postings.

- Several implementation contributions, while bringing quantifiable advantages in new features, were provided by their authors with only a rough integration, resulting in broken stability or even adverse effects on other features. The original authors would often refuse to improve the situation as they were not interested or the costs would be too high for them. As a result, the principal maintainers had to take on a significant work load just for fixing code, rather than concentrating on driving themselves new features.

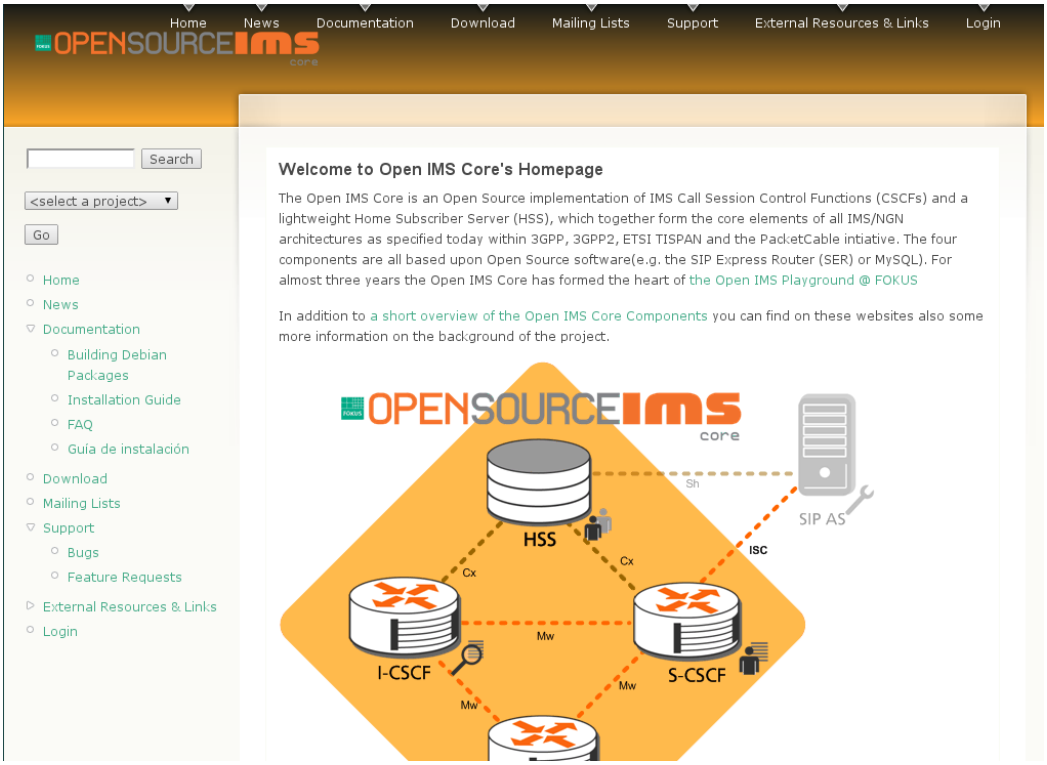


Figure 6.11: The OpenIMSCore Community Website

Considering the second trait of the single-survivor model in Open Source projects, while there were no other IMS CN projects available at that time, several similar in-house project had the potential to be launched in this space. In order to prevent such situations the wiki-style policies where envisioned as helpful, still a secondary important effort was undertaken. The strategy was to reach-out and attempt to

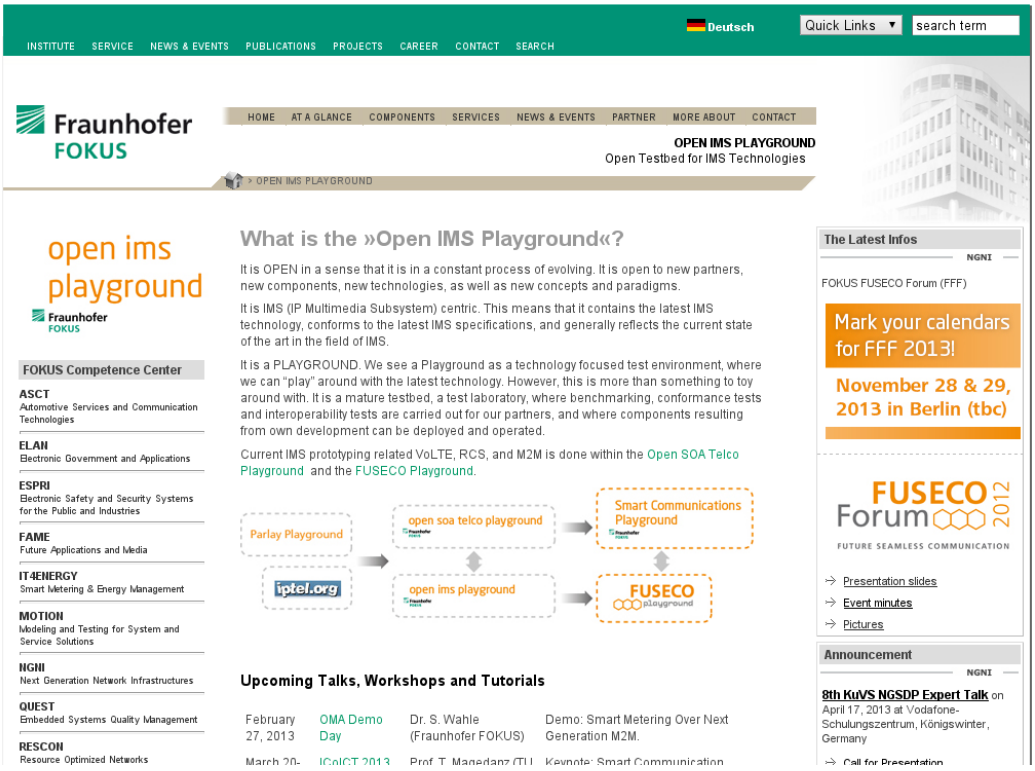


Figure 6.12: The Open IMS Playground @ Fraunhofer FOKUS Website

join-together with otherwise competing communities and projects. In this direction, although a major commitment and investment was provided by the original team of Fraunhofer FOKUS employees, the `http://www.openimscore.org` community website was not even branded or maintained as being owned by the institute itself[7].

As a notable example, a Portugal Telecom Inovação team worked on a similar project using also the SER as a base, yet with a different approach of not modifying the core or modules implementation but orchestrating everything through the configuration's routing script (very similar to Option B in the Design of the Open Source IMS Core analysis). Upon showcasing our approach, their team has agreed to join us and to help with their expertise in improving our implementation into a common platform.

From a community hierarchy perspective, one as flat as possible was targeted. For each of the major components a developer was assigned as principal maintainer. This maintainer was in charge of verifying that each commit was sound and did

[7]For presenting the institute's use of the project and the additional non-free tool-sets, the `http://www.open-ims.org` website was used in parallel, which clearly linked to the OpenIMSCore project as being hosted through a community maintained website, while also following Fraunhofer FOKUS' own interests.

not potentially break functionality, assisting as required the original authors. The changes did not had to be always piped through the principal maintainers, but their roles was more of a supervision one. The decision on who should act in this role was not systematized, but more of an implicit assignment to the developer which was most active in the respective field and as such had the best knowledge about it.

Additional community leaders emerged in the support domains, as for example to guide and help the community by providing technical support on the mailing lists, or to maintain and improve the project's website. Again this was not a formal process, but a natural selection one.

The technical support provided to the community was significant, taking a major slice of the overall efforts involved. This was mainly because not only the project required to train and specialize developers, but also quite often the supporters had to help and educate new users on the IMS concepts and guide them on the standards aligned path of the new NGN architectures.

Besides, a relatively large audience has been targeted, hence many users were not actually interested in the inner-working or configuration details of the project itself, but rather would use the project as a black-box platform for their own developments. For these situations, fully pre-installed, pre-configured and pre-provisioned virtualized environments were created, which could be downloaded and executed with minimal overhead in actually understanding the complex internal details.

6.5 Conclusions on Implementation

The prototyping started initially slow, by building some of the main concepts and show-casing their feasibility for test-bed experimentation. A major phase of adding functionality and completing the architecture was undertaken as part of an NGN performance evaluation effort. Following, these initial components have been launched as open Source and a community has been formed. Spanning over many years, the major implementation tasks have been successfully completed through the Open Source community processes, such that the OpenIMSCore project gained a world-wide reference status.

Chapter 7

Validation

The present chapter demonstrates the impact that the OpenIMSCore project had, over the years since its launch as Open Source. First an evaluation is made by going back to the design matrix defined in Chapter 4 – Design of the Open Source IMS Core. This will directly indicate how well the initial design targets have been achieved. Next the impact on the IMS research community is analyzed, through adoption of the OpenIMSCore as a prototype implementation in numerous R&D projects, spawning of adjacent Open Source project as well as it use in various test-beds around the world. Special attention is of course allocated to analyze the results as expected from the goals set in the Introduction chapter.

7.1 Evaluation of the Design Matrix

Based on the previously defined design matrix, on which the OpenIMSCore initial architectural decision has been motivated, an analysis is needed to evaluate the resulting implementation.

With regard to functionality, the OpenIMSCore has far exceeded the initial expectations. Started as a performance evaluator, only the minimal functionality was first implemented. This initial set was further supplemented with incremental new features, until the resulting prototype has been deemed as sufficiently emulating the processing characteristics of a real-life implementation, such that performance benchmarking results were considered by the R&D customers as relevant for the architecture as a whole.

Of course, 100% compliance was never targeted or attempted. Still, through the community effort in adding features and validating correct functionality, slowly the project has achieved a reference status. Essential for this achievement is of course also the Open Source nature of the project, which provides both a zero-cost entry barrier for obtaining the software, as well as the full control transferred to the experimenter, free to fix issues or develop new features.

On performance, the OpenIMSCore has provided not only first time performance indicators, but also a high standard on the respective indicators. For validating this, the Open Source SIPp IMS-Bench [199] tool can be used to measure the high performance[1].

While it is always hard to claim performance levels, mainly due to the high number of influencing parameters as well as the continuous evolution nature of the project, tests have shown that even in the most simple, non-distributed topology, signaling was processed with Round-Trip-Time (RTT) in the range of 5 to 100ms while the system was capable of a throughput in the order of 1-2 hundred(s) of registrations per second, 4-500 session setups per second or even thousands of simpler IM transactions per second.

Capacity wise, perhaps the best validation was provided by the `SIPNuke` [204] One-Million demonstration [198]. There a population of one million subscribers has been emulated starting from a realistically-looking traffic profile. The OpenIMSCore CSCFs were able to process without issues the entire signaling load, while running on a 2008-level workstation, again in the simplest topology.

It has to be stated, of course, that all of the rough performance indicators above are in the end not directly representative for real-life implementations. As the prototype nature of the project did not approach important performance influencing procedures (e.g. security, high-availability, 100% processing aligned to the specifications), it is naturally expected that future deployments will exhibit a lower performance and the obtained numbers in a test-bed must not be confused with ultimately expected real-life exploitation figures.

The remaining performance topic of scalability is implicitly tackled by providing

[1] The default OpenIMSCore configuration is designed for verbosity and ease of debugging. While conducting performance testing, a series of parameters shall be tweaked in the configuration files, resulting in much higher results. The OpenIMSCore HSS is an inherent bottleneck because of the employed RAD development pattern with `Java`. A `C`-based optimized HSS was commonly used in benchmarking. A decision has been taken to not Open Source the respective prototype as a sign of good-faith for the IMS IPR holders, in the spirit of the research-only status of the OpenIMSCore project.

standards aligned reference points between the functional elements. The project can in practice be instantiated in any topology, allowing further performance improvement by architectural implicit distribution of load over multiple instances of each functional element. However, due to the more-than-sufficient capacity and speed provided even in the simplest topology, test-beds only very rarely required it and most of them provide all of the functional component in one system, as sort of an IMS-in-a-bottle setup, sometimes even virtualized.

Regarding stability, resulting from an analysis of the support mailing-lists traffic and statistics, this has always been influenced, as expected, by the introduction of new features, yet exhibiting a generally improving trend. Certainly the community has provided an enormous amount of help, by constantly using, testing and reporting issues. The fact that in the last years, as also influenced by the reduced number of new features, not many critical problems have been discovered, indicates that the project has achieved a sufficient level of stability for the targeted test-bed usage patterns.

On the security topic the project pioneered of course the use of AKA in IMS and provided one of the first Open Source implementations. Further NDS features and THIG have been included, which together with the P-CSCF procedures cover well enough the internal security procedures. In fact, for most of the security functionality it was necessary to provide configuration options as to be able to disable it item by item, such that in various non-security concerned environments, experiments could be successfully performed even against incomplete clients or applications.

Regarding costs of implementation, the initial big push has been given by the benchmarking activities, which generated an R&D budget. For the second phase, as the benchmarking activities were winding-down, the community provided a significant investment of testing as well as implementing. The principal maintainers as well as the associated overhead costs for hosting and running the community website and repository have then further been funded from R&D work and licensing on client and application domains, which used the OpenIMSCore as a base platform, as well as test-bed deployment income generated by distributing and hot-starting numerous test-beds with operators and research organizations around the world.

On the topics of openness and relevance, the number of subscribers and participants in the community is perhaps the best indicator that the project has been accepted as a reference for test-beds, that the provided functionality was sufficiently aligned with the standards and that the configuration, debugging and management were not insurmountable (see Figure 7.4).

To round up the analysis of the design requirements versus the resulting implementation, in Table 7.1, the rows from Table 4.2 are reiterated, with appreciations on the achieved points in OpenIMSCore.

A second project is introduced, Kamailio (was initially named Open SIP Express Router (OpenSER)) [205], which recently has taken upon the task of reusing as much of the code and experienced gained through the OpenIMSCore project [141], yet is targeting a hardened implementation. This should be usable in real-life carrier-grade VoIP networks, by reusing IMS concepts, yet not limited or strictly fol-

lowing the IMS standards. Modules have been in some cases directly taken from the OpenIMSCore project, or in other cases they have been re-designed, merged and/or entirely rewritten for best integration into the current SIP-Router architecture[2].

	Implementation	
Requirements	OpenIMSCore	Kamailio with OpenIMSCore modules
Functionality		
- minimal	good	good
- sufficient	better	good
- specific	facile	average
Performance		
- capacity	good	carrier grade
- latency	good	good
- hw./platform optimizations	some, old	good
- scalability	good	better, proved
Benchmarking		
- functional completeness	better	good
- realistic performance	better	some
- relevant	better	as
Stability	test-bed	carrier-grade
Security		
- functional	functional	carrier-grade
Costs		
- implementation life-cycle	high	reusing code
- for performance	moderate	low
- re-usability	low	very high, reference
Openness		
- standards aligned	better	in concepts
- configuration	low	high/reference
- debugging	good	requires experience
Relevance		
- status/acceptance	best	ramping-up
- community	ramping-down	large and varied

Table 7.1: The Design Matrix with Implementation Results, for the OpenIMSCore and Kamailio with OpenIMSCore Imports

To conclude, besides achieving the design principles, the project, while currently

[2]The OpenIMSCore uses a common ancestor platform from which OpenSER, later renamed to Kamailio, has also spawned. Recently the Kamailio community has merged back with a part of the original SER community and the project is currently known as both Kamailio [205] and SIP-Router [206].

in a completed phase from a test-bed perspective, lives on through the direct reuse, import of the know-how and gathered experience, in the Kamailio-IMS efforts, as a carrier-grade targeting implementation.

7.2 Validation through the Initially Proposed Targets

While the designed targets have been validated in the previous section, a look at the overall motivation of the dissertation has to be taken. In the Introduction chapter a set of key questions have been listed and they will be answered here. These are in effect the main technical traction points and as such from them result the main scientific contributions provided by the work.

7.2.1 Answering the Principal Questions

Q1. What are the key design principles and implementation challenges for an NGN test-bed toolkit?

The critical design principles have been exposed in the Design of the Open Source IMS Core chapter. A series of criteria have been analyzed: functionality, performance, benchmarking, stability, security, costs, openness and relevance. Chapter 5 derived a specification from this design criteria and Chapter 6 exposed the implementation process. The project adoption numbers and its acceptance as a reference for the proposed targets, validates the methodology for the purpose of creating successful test-bed prototypes to fuel validation, improvement and adoption of new technologies.

Q2. Is the Open Source model feasible for such NGN prototype implementations?

Considering that the OpenIMSCore projects successfully provided, within a relatively short time, usable CSCFs and HSS prototypes, the implementation of a test-bed toolkit for the practical trialing of NGN architectures and concepts was demonstrated to be feasible. In fact, by reusing already established Open Source tools and platforms, the cost was further reduced for the initial phase, while the second phase of providing back to the community was again beneficial by significantly improving the momentum of the project with community contributed efforts.

Even further, the OpenIMSCore community transformed itself into an incubator for further projects, spawning additional ones, which aimed to provide similar prototypes for more NGN components and systems.

One of the most important conclusions is that partial prototypes are feasible to and critical for validating new technologies. While targeting only a subset of the functional aspects and with a fraction of the budget required for a commercial productization, by re-using Open Source tools and platforms and by fostering the community effects, still the resulting prototypes can achieve a reference status and as such have an important impact on the technology and its adoption.

Q3. Does an Open Source model bring benefits for the R&D of NGN?

Taking into account for example the successful use of the OpenIMSCore to develop, standardize as well as validate the NGN/IMS Performance Benchmark [12], the answer is positive. As IMS starts to be more and more deployed for real-life exploitation, these standards are again and again used as references for evaluating and testing implementations on a generalized comparison level, as well as for their suitability for certain customized scenarios.

Then as the project is used on a large scale to test and improve new products and services, the indirect impact on R&D of NGN products is present.

7.2.2 Answering the Secondary or Indirect Questions

Q4. Testbed scalability and Performance: can small scale testbed results be used for real networks?

Also this question has been positively answered. The OpenIMSCore has been used first as performance validation tool for IMS and later on even as to perform interoperability testing. This was of course greatly aided by the Open Source nature of the project, which did not require or mandate the limitations in disclosing performance indicators.

The best validation proof here comes from the wide use of the project in a majority of test-beds worldwide, even later on when and after large scale commercial products started to be available. The role of the OpenIMSCore switched then from an early technology validation toolkit, to a Swiss-army knife of test-beds. Quite similar and also owing to the background platform, the vast flexibility of the prototypes made it especially suitable for filling functionality gaps or compensating for limitations in certain products.

Q5. Can innovation be fostered by bridging the gap between industry and academia with independent and affordable (eventually reference) prototypes?

By analyzing the follow-up effect and the impact on the academia, it can be seen that at least a couple of the published articles and journal publications of the OpenIMSCore authors have been currently cited by 30+ and more subsequent articles. By looking at the citation numbers[3], it can be derived that a large number of research institutions and universities have directly started to influence the industry by providing their input, ideas and new concepts. Further, many of the institutions behind these academic research works have also established in-house technological test-beds. For example [207] has acted as effective meeting place for realizing both research as well as validation of the architecture and commercial equipment, with the direct implication of the academia.

[3]For example as indicated by Google Scholar searchers: http://scholar.google.com/scholar?oi=bibs&hl=en&cites=3783543709622809650, http://scholar.google.com/scholar?oi=bibs&hl=en&cites=11771248945828275455

From the academia, technical school and university professors started to use the project for teaching purposes. Simplified IMS-in-a-bottle systems could be distributed with zero-costs to students during classes and seminars, followed by project assignments on using, improving or even adding new functionality to the Open Source project. Accordingly graduates were more readily knowledgeable and empowered on the topic of NGN, without requiring anymore vendor-specific trainings on potentially closed technologies and black-box systems.

Q6. How to evaluate and quantify the usefulness of the resulting toolkit for test-beds?

Directly answering the question is the present validation chapter and the answer is provided by the fact that the author has had significant contribution to standards, as detailed in Section 7.3.1. These are centered around how to test the IMS architecture, for various criteria like performance or inter-operability, by creating the evaluation models as well as the metrics to quantify results.

7.3 Impacts on the Standardization, Academia and the Industry

After the evaluation based on the design principles as well as on the principal questions of the dissertation, this section follows on the influences that the project produced on the 3 main usage domains and targets.

7.3.1 Influences on the Standardization

As previously indicated, the sparkling incentive to realize the OpenIMSCore, at the scale that it was later implemented, was actually a need for trialing in practice the performance of IMS architecture. The SIG for IMS Benchmarking followed this goal [193] and supported practical implementations of both the SuT within the OpenIMSCore project, as well as the TS in multiple instances [199, 204, 208].

The methodology as well as the standardization of the parameters and the entire process has been brought up for standardization at ETSI/TISPAN, within the IMS Network Testing (INT) group. The resulting standard has been published as ETSI TISPAN6 TS 186.008 "IMS/NGN Performance Benchmark", part 1 Core Concepts, part 2 Subsystem Configurations and Benchmarks and part 3 Traffic Sets and Traffic Profiles [12, 152, 153].

To fully demonstrate that the OpenIMSCore was instrumental in these benchmarking activities, a result of such a benchmarking test is available for consultation in Appendix C. While the results should not be taken as definitive results for the performance of OpenIMSCore, this demonstrates that the methodology is correct and that the OpenIMSCore includes sufficient functionality and is stable enough to allow for such performance evaluations, as for example indicated in industry white-papers like [192].

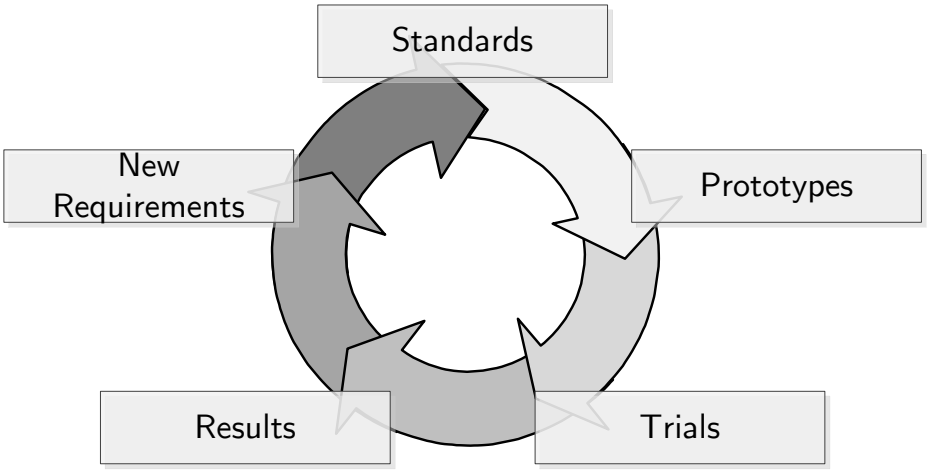


Figure 7.1: The Standardization Cycle, with Respect to Prototypes and Trials

Another example of such a standard which was directly created and influenced by the OpenIMSCore project is the ETSI TISPAN6 TS 103.029 "IMS Network Testing (INT); IMS & EPC Interoperability test descriptions". This has drawn upon the experience gained within the Open Source project, as well as into its iterative next step prototype, the OpenEPC project [202]. There the combined experiences have been used, to supplement the standard procedure with limitations found from hands-on experiments. The resulting interoperability standard goes beyond the theoretical procedures as resulting from the combined existing standards, uncovering mechanisms which are only visible through practical tests, made possible by prototype implementations.

Further standards have been for sure influenced, indirectly. The project was available as Open Source and has been used to trial IMS in what could probably be referred to as the golden years of improvements and stabilization of the IMS architecture, between Release 6 and Release 8, when the architecture, initially introduced with Release 5, has matured. Even after Release 8 up to today, while the core concepts have no longer changed, still the influence on interfacing with the new architectures is visible in the current implementation efforts.

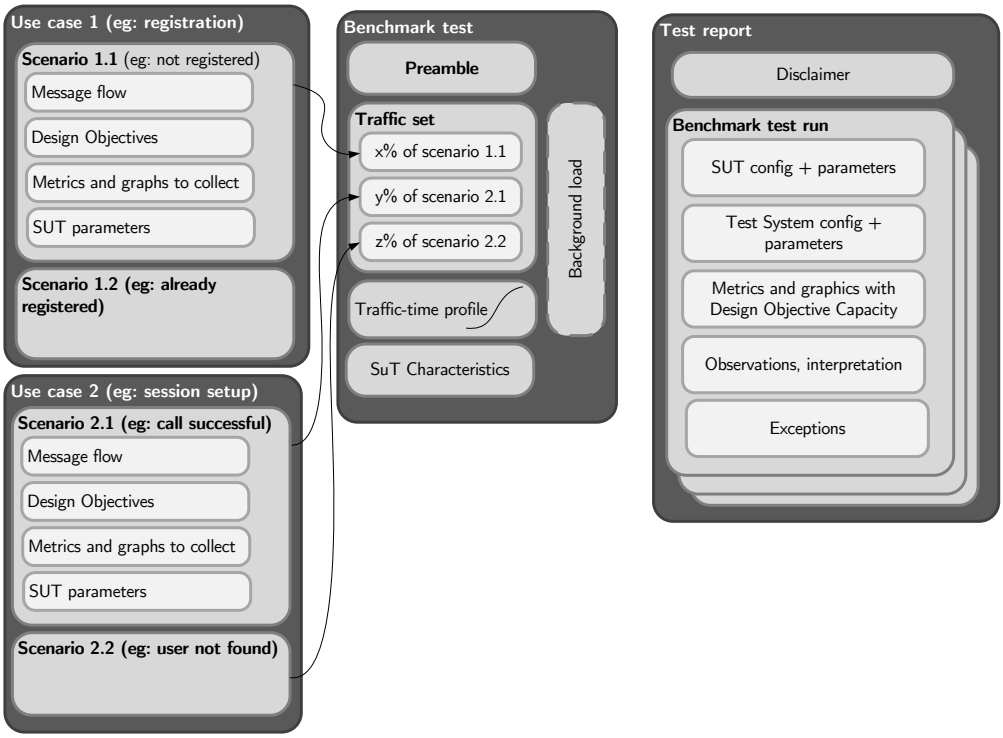


Figure 7.2: ETSI TISPAN6 TS 186.008 IMS Benchmark Information Model (from [12])

7.3.2 Implication on the Industry and on the Adoption of IMS

The implications on the industry, while in some cases easier to quantify, because of the direct R&D customers which used the OpenIMSCore and requested additional work, it is in fact harder to present because of Non-Disclosure Agreements (NDAs) as well as the secretive nature of competing industry research divisions.

The present section aims then to present at least a few proofs on the use of the project by the industry. In fact, because of the Open Source nature as well as the reference status that it achieved, the OpenIMSCore is present in most laboratory conducting research with or around IMS, even if not as the primary CN, but in many case as a flexible toolkit.

7.3.2.1 World-wide Use of OpenIMSCore

To quantify the adoption and use of the project, the simplest method would be to count downloads. Unfortunately, as to keep in line with the Open Source best practices, the early-on decision was taken not to require a registration process in order to obtain the source code. Nor would the download numbers be easily verifiable

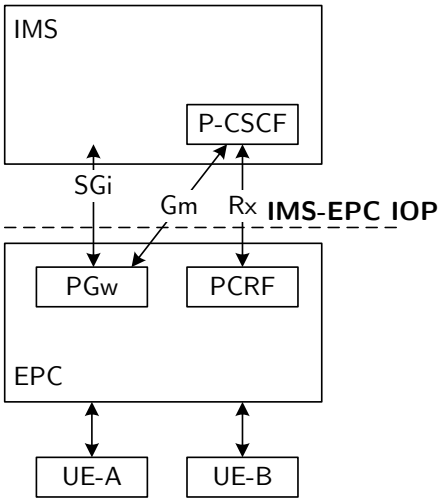


Figure 7.3: ETSI TISPAN6 TS 103.029 IMS-EPC Interoperability Overview (from [13])

as many of the source code repository accesses would be just for obtaining updates on existing installation, while also it can not be safely assumed that every check-out would translate into a running test-bed.

An indirect indicator, which perhaps would be better suited is the number of active subscribers to the project's mailing lists. As showed in Figure 7.4, the project was quick to attract 500 subscribers in 2009, after which the subscriber base stabilized and showed a slower uptake, still following though an upward trend. Such volume, considering that the project is in fact quite specialized on a specific future technology, represents a significant number of researchers, engineers and students, from all over the world, which use and participate in the project.

7.3.2.2 Use of OpenIMSCore in Major Publicly Funded Research Projects

The IMS prototypes have been used in numerous research projects around the world. Some of the most important ones, some which have actually involved the participation of Fraunhofer FOKUS are:

- MAMS: The Multi-Access Modular-Services Framework Project [209, 210, 211]

  > *"MAMS (Multi-Access, Modular-Services Framework) was a joint project funded by the Federal Ministry of Education and Research (BMBF) and managed by Deutsche Telekom Laboratories.*
  >
  > *The goal of MAMS was the specification and roll-out of a novel, unified, open Service Delivery Platform (SDP) for Next Generation*

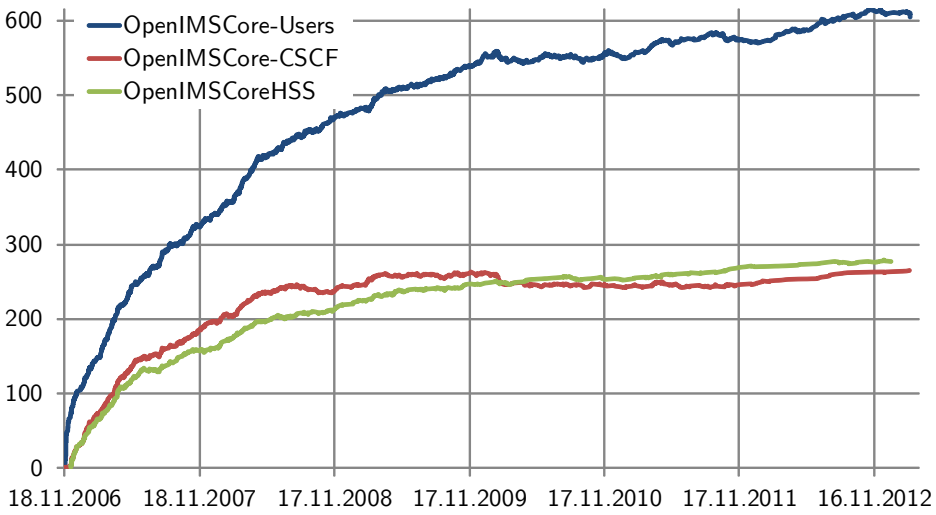


Figure 7.4: Evolution of the OpenIMSCore Mailing Lists Subscribers

> *Networks and Services (NGN). The developed SDP enables the rapid, streamlined design of new combinable services for a wide range of multimedia applications based on the uninterruptible use of various network technologies and integrated voice and data. The project results enable the deployment of distributed dynamically configurable network nodes and a modular IMS-based Overlay Service Architecture with open interfaces for the integration of third party providers to develop methodically structured processes and service-generation tools and to address general problems of open service environments. " [211]*

- MAMSplus: The Multi-Access Modular-Services Framework Project / Flexible NGN Service Creation for Small and Medium Enterprises (SMEs) [212, 210, 213]

  > *"NGN Service Exposition, together with an open, flexible and intuitively understandable Service Creation Environment not only enables innovative business models for the telecommunication industry, but also for SMEs to enrich and facilitate their communication capabilities.*
  >
  > *In June 2008 the German Federal Ministry of Education and Research (BMBF) started the MAMSplus (Multi-Access Modular Services Framework) project. Managed by the Deutsche Telekom Laboratories flexible NGN service exposure, creation and composition*

solutions for application in e-Health scenarios and for utilization by SMEs will be prototyped. In the context of MAMSplus FOKUS is developing innovative mechanisms for network abstraction, service exposition and NGN Operations, Administration, Maintenance and Provisioning (OAMP)." [213]

- CoSIMS: The Community-enabling Services on IMS Project [214, 215]

  "CoSIMS – (Community-enabling Services on IMS) is an R&D project, financed and conducted by Deutsche Telekom Laboratories in cooperation with Hewlett Packard Corp., T-Systems, trommsdorff + drüner GmbH and the Fraunhofer Institute for Open Communication Systems FOKUS, investigating the provision of community based services by emerging 3GPP IP Multimedia Subsystem (IMS) / ETSI TISPAN NGN Release 2 based service platforms. The project started in November 2005 as one of the biggest R&D projects of the T-Laboratories and is still ongoing." [215]

- Panlab: The Pan-European Laboratory for Next Generation Networks and Services Project [216, 217]

  "The Pan-European laboratory is based on the concept of federation of distributed test laboratories and testbeds that are interconnected and provide access to required platforms, networks and services for broad interoperability testing. The coordination of resources and access to the laboratory services will be controlled by a centralised entity.

  The Pan-European laboratory is a concept that is being introduced to enable the trial and evaluation of service concepts, technologies, system solutions and business models to the point where the risks associated with launching them as commercial products will be minimised. The accomplishment of this objective, which will assist many different European collaborative projects, is an important step towards the establishment of a truly pan-European collaboration network.

  Panlab is a Specific Support Action (SSA) of the European Union's 6th Framework programme, Thematic Priority 2 (IST – Information Society Technologies).It was submitted to the fifth call of the programme and addresses the strategic objective "Research networking testbeds". The project is partly funded by the European Commission and is running from June 2006 to May 2008." [217]

- IMS-ARCS: The IP Multimedia Subsystem Advanced Cluster of Services Project [218, 219]

  "Within the context of the Enterprise Ireland funded ILRP (Industry Lead Research Programme) Project IMS ARCS, FOKUS sup-

ports the establishment of an Open Source IMS based testbed infrastructure to enable the local Irish SME industry to explore business opportunities in the IMS/NGN context. [219]

- PEACE: The IP-Based Emergency Applications and ServiCes for NExt Generation Networks Project [220, 221]

  "The project IP-Based Emergency Applications and ServiCes for NExt Generation Networks (PEACE), funded through the FP7 programme "Research for SMEs", aims at providing a general emergency management framework (based on the IP Multimedia Subsystem (IMS)) addressing extreme emergency situations such as terrorist attacks and natural catastrophes as well as day-to-day emergency cases such as calls to the police or fire brigade." [221]

- SL155: The Service Line 115 Project [222, 223]

  "The institutional citizen service hotline in New York serves as example. Under the number 311, the citizens of New York can reach public administration 24 hours a day. New York's citizen service receives approximately 45.000 calls a day.

  During the IT summit-meeting (on December 18th 2006 in Potsdam), a similar project was discussed for Germany. People here, with questions about administrational services shall not have to cumbersomely search for the responsible department, but receive help from a central office. Next to the well known emergency numbers 110 for the police and 112 for the fire brigades, such a public service hotline would be another step towards an easier, more citizen friendly, and transparent public administration." [223]

- MCN: Mobile Cloud Networking [224]

  "MobileCloud is Mobile Network + Decentralised Computing + Smart Storage offered as One Service - On-Demand, Elastic and Pay-As-You-Go. The top-level objectives of the MobileCloud project are a) to develop a novel mobile network" architecture and technologies, using proof-of-concept prototypes, to lead the way from current mobile networks to a fully cloud-based mobile communication system, and b) to extend cloud computing so as to support on-demand and elastic provisioning of novel mobile services. MobileCloud will investigate, implement, and evaluate the technological foundations for that system. It will meet real-time performance needs, support efficient and elastic use and sharing of both radio access and mobile core network resources between operators.

  [...]

The chair AV of TU Berlin is mainly involved in the WP3: 'Mobile Cloud Infrastructural Foundation' and WP5: 'Mobile Platform' development around IMS-as-a-Service based on Cloud Computing principles (on-demand, elastic, pay-as-you-go)." [224]

7.3.2.3 Deploying test-beds for the operators and the industry

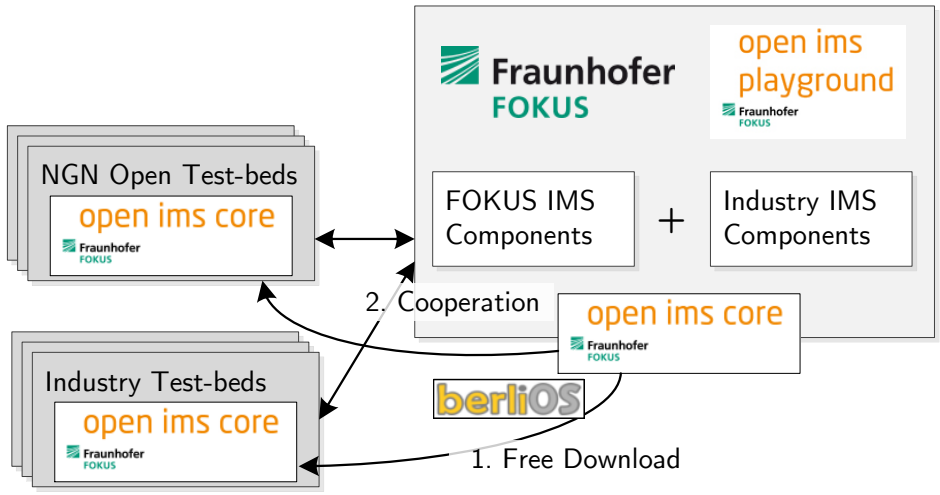


Figure 7.5: The Fraunhofer FOKUS Open IMS Playground [14]

The OpenIMSCore project has been, within Fraunhofer FOKUS, the central base foundation for one of its most important laboratories, the Open IMS Playground [225, 14, 226]. Launched in 2004, this test-bed has fostered both the use and experiences of the open source CN, as well as the design, implementation and trial of numerous IMS functional components, elements, procedures and concepts.

The research concept behind the Open IMS Playground was based on the publicity that the OpenIMSCore was about to provide, through its zero-entry-cost proposition and world-wide distribution. Next, Fraunhofer FOKUS was able to enhance the free experience and establish industry as well as academia open test-beds, where together with the OpenIMSCore, own developments for clients, application servers, service delivery platforms and other tools combined seamlessly with IMS components provided by the industry (see Figure 7.5).

As a result, the FOKUS team has been employed numerous times since 2006 and even today to deploy (customized) replicas of the Open IMS Playground around the world. Figure 7.6 provides a good overview of the instances which have been provided directly. Also indirectly, a series of other research institutes and universities

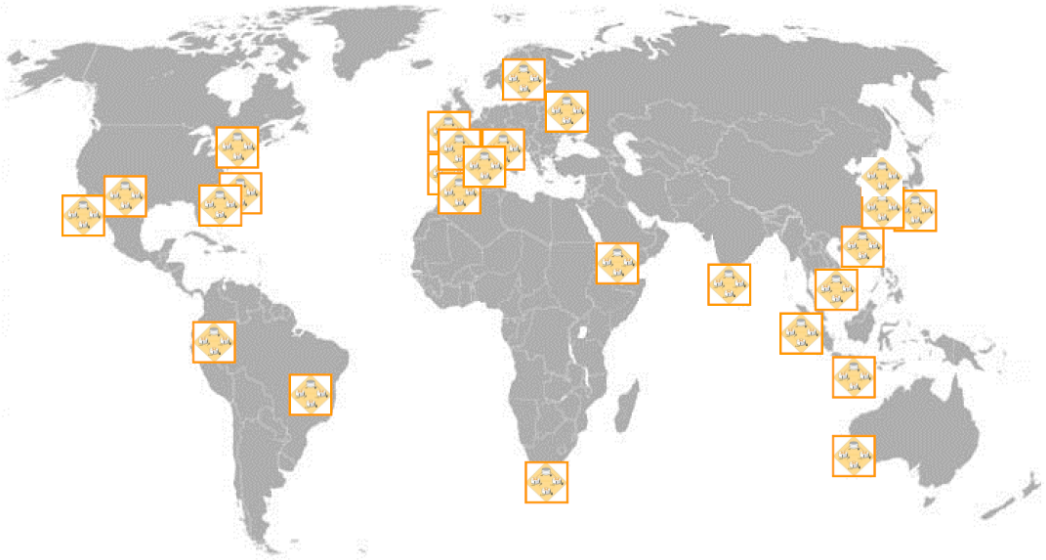


Figure 7.6: OpenIMSCore and OpenEPC Test-beds around the World

started to build and provide similar experimentation infrastructures (e.g. [227]), providing alternative IMS application prototypes similar to FOKUS ones, yet also reusing the same OpenIMSCore CN as a platform.

Even as the technologies evolved and IMS was approaching the market deployment, the project continued to be used in the successor test-beds and laboratories. Following again on the above described environment, the FOKUS research team has started the Open SOA Telco Playground, which extended and complemented the offering with a solid Service Delivery Platform (SDP) [15]. There the OpenIMSCore successfully integrated with complex and demanding service platforms, demonstrating its flexibility in serving new service concepts and paradigms. Today, the core platform is used to validate the latest multimedia communication concepts as for example Rich Communication Suite (RCS) [228].

Fast forwarding to the introduction of LTE radio and the EPC architectures, the OpenIMSCore lives on to enable features like VoLTE, in the FUSECO Playground [16] (see Figure 7.8). Undertaking a functionality overhaul, the project successfully integrated with the new IP connectivity architecture, validating new concepts and taking advantage of differentiated QoS and charging over a comprehensive set of wireless communication technologies.

7.3.3 Implication on the Academia

Just as with the industry, the project has been downloaded and used extensively also in the academia. In fact, the effects are quite hard to quantify at this moment, as a complete listing of the academic papers which are the result of evaluating, using and extending the OpenIMSCore is in the order of hundreds.

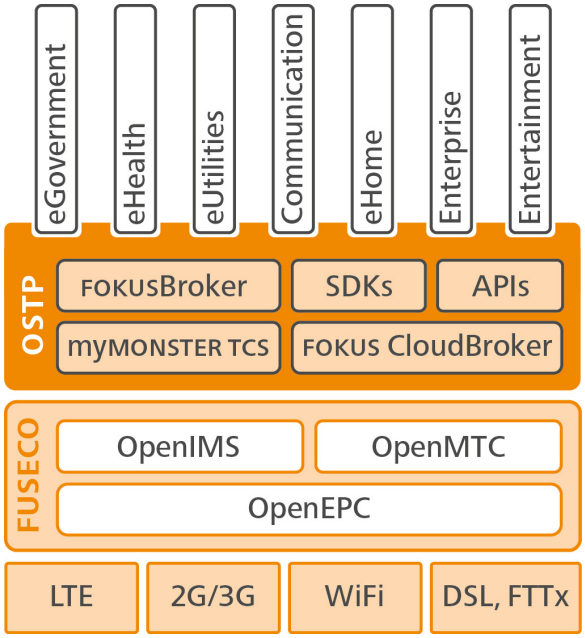


Figure 7.7: The Fraunhofer FOKUS Open SOA Telco Playground [15]

Indexing Engine	Search Keys		
	"Open IMS Core"	FOKUS, IMS	Fraunhofer, IMS
Google Scholar	408	4,220	4,190
IEEE Explore	71	55	103
ACM Digital Library	443	46	83
CiteSeer^x	9	292	492

Observation: composed on 25.11.2013

Table 7.2: Results Found on OpenIMSCore Related Keywords on Academic Articles Indexing and Search Engines

As an interesting result, a Diploma-Thesis was dedicated entirely to the analysis of the project [229]. This provides an overview of IMS as an architecture followed by an implementation study and code analysis.

7.3.3.1 Deploying test-beds for the academia and empowering education

First of all, one of the main sectors targeted by the project was all along the academia. As stated before, the target is to empower the academia with the test-bed functionality, such that it can itself experiment and influence directly the telecommunication sector. Traditionally telecommunication equipment was not available, neither on affordable costs, nor enough volume, nor as source code allowing for

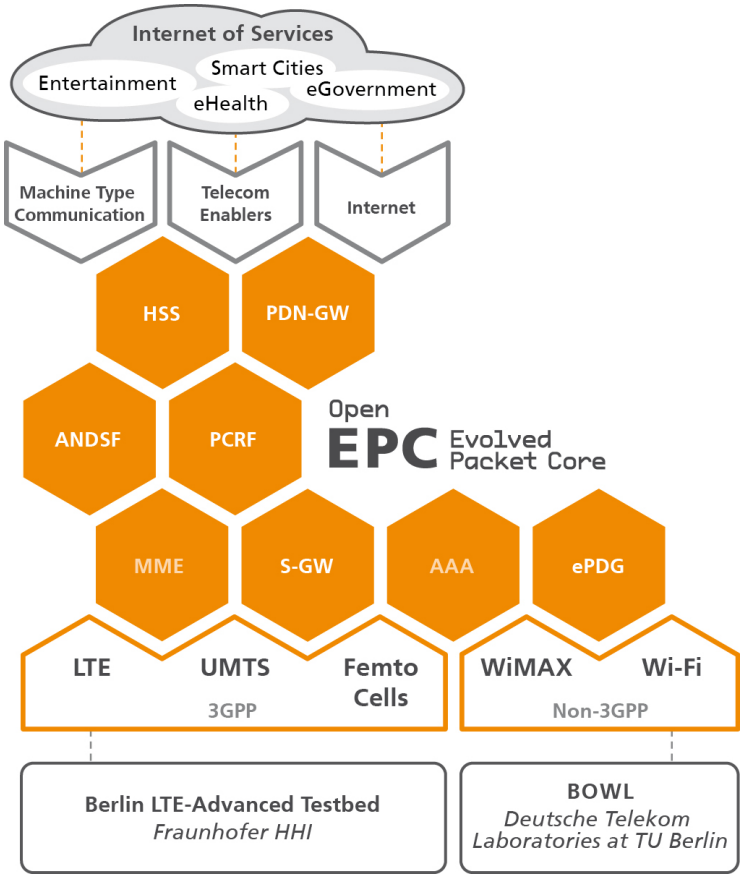


Figure 7.8: The Fraunhofer FOKUS FUSECO Playground [16]

modifications.

With the introduction of the OpenIMSCore, the academia approach to the telecom world was changed. Professors started to adopt the project and to distribute it to students directly, for example on Live-Compact Disc (CD) Linux distributions. With these, the students could practically test the learned concepts and much better comprehend the inner workings of the NGN architecture. Even further, seminar work and student projects could be finally feasibly executed, as for example implementing new services, extending the existing functionality or practically analyzing various procedures and functional elements.

Besides improving the education process, the introduction of such open source projects in the academia has enabled a direct influence on the architecture. Ideas and concepts coming from the learning institutions could be much more easily trialed for feasibility and then pushed to the standardization and implementation. Also graduates no longer required the traditional vendor-run training, but the readiness level was elevated directly in the academic education process.

7.3.3.2 Projects spawned as a result of OpenIMSCore

The academic impact can be evaluated directly from the visible output, in the form of new Open Source projects, or developments on existing ones, as influenced by the OpenIMSCore. Here are a few of such open source projects, which have a common life as associated with the OpenIMSCore project.

The University of Cape Town (UCT) IMS Client [230, 231]

The UCT IMS Client was one of the first IMS clients capable of taking advantage of the new AKA authentication method with an implementation pioneered for SIP by the OpenIMSCore. The client supports also voice and video calls, both pager-mode and session based messaging and presence. Of special interest is the integration of IPTV and XML Configuration Access Protocol (XCAP) capabilities.

The PT Inovação IMS Communicator [138]

The Portugal Telecom Inovação IMS Communicator is a similar IMS client implementation, focusing especially on service integrations and related research. It is built on top of established `Java` technologies like JAIN-SIP Reference Implementation (RI) and Java Media Framework (JMF) API.

The UCT Advanced IPtv, IPtv Charging Framework, Policy Control Framework and Back-to-back User Agent [232, 233]

The UCT IPTV implementation targets media server implementations and extensions for television delivery scenarios over an IMS NGN infrastructure. Notable is the accelerated development enabled by reuse of existing Open Source libraries for media manipulation and delivery.

The UCT PCC and charging frameworks target reference implementation for the signaling and enforcement of QoS parameters for media streams, as well as for linking towards online as well as offline charging capabilities.

The Kamailio (former OpenSER) Open Source SIP Server, as a host for the OpenIMSCore experience [141]

The Kamailio/OpenSER project is representative here as its maintainers have decided to port back the IMS functionality implemented in the OpenIMSCore project, to the mainstream development tree (after all, the OpenIMSCore is based on an older version of the SER common ancestor SIP Proxy/Router). The imported functionality has been filtered through a critical view which followed on the carrier-grade planned deployments. Hence the results from OpenIMSCore will live on well beyond the test-bed targets, with real roll-outs of NGN VoIP architectures.

7.3.3.3 ONIT Workshop as a Catalyst for Open Source Test-beds

The Open NGN and IMS Test-beds (ONIT) Workshop [234] is a yearly event which has taken place since 2009. As a community driven academic event, it aims to gather

best-of-breed articles and papers on open test-beds on the special topic of NGN and IMS. Besides the peer-reviewed publication aspects, as part of high-profile ICST and IEEE conferences, the workshop is usually executed as a full-day activity, which also attracts reference key-notes as well as demonstrations and practical hands-on sessions.

Figure 7.9: ONIT Workshop Panel Discussions [17]

The event was started as part of Testbeds and Research Infrastructures for the Development of Networks and Communities Conference (TRIDENTCOM) 2009 [235, 236] and has enjoyed a warm welcome from the many attending participants. While in 2010 it continued as part of TRIDENTCOM [12, 238], in 2011 it was co-hosted with the IEEE Computer Software and Applications Conference (COMPSAC) 2011 [239, 240] and in 2012 it was co-hosted with the IEEE Global Communications Conference (GLOBECOM) [241, 17].

The event was sparkled by the OpenIMSCore project, fact demonstrated by the constant use and appearance of the project in every year's proceedings. The author here has been serving constantly as a Technical Program Committee (TPC) member and was co-chairing it in 2010 and 2012.

7.4 Impacts on Future Technologies - Trialing VoLTE

In present, the OpenIMSCore has passed its maturity and demonstrated its usefulness in test-beds. Still, almost 10 years after the first foundation stones have been laid, its demand in the R&D community remains high. The introduction of IMS, although initially envisioned during the 3G RAN life-cycle, has lingered due to various commercial and technical reasons. Today though, with the migration towards 4G technologies, the telephony service remains a topic for trials and evaluations. Although many alternative solutions have been proposed, still the best solution, commonly referred to as VoLTE remains a combination between an IMS as support for advanced communication services (e.g. RCS, Multimedia Telephony (MMtel))

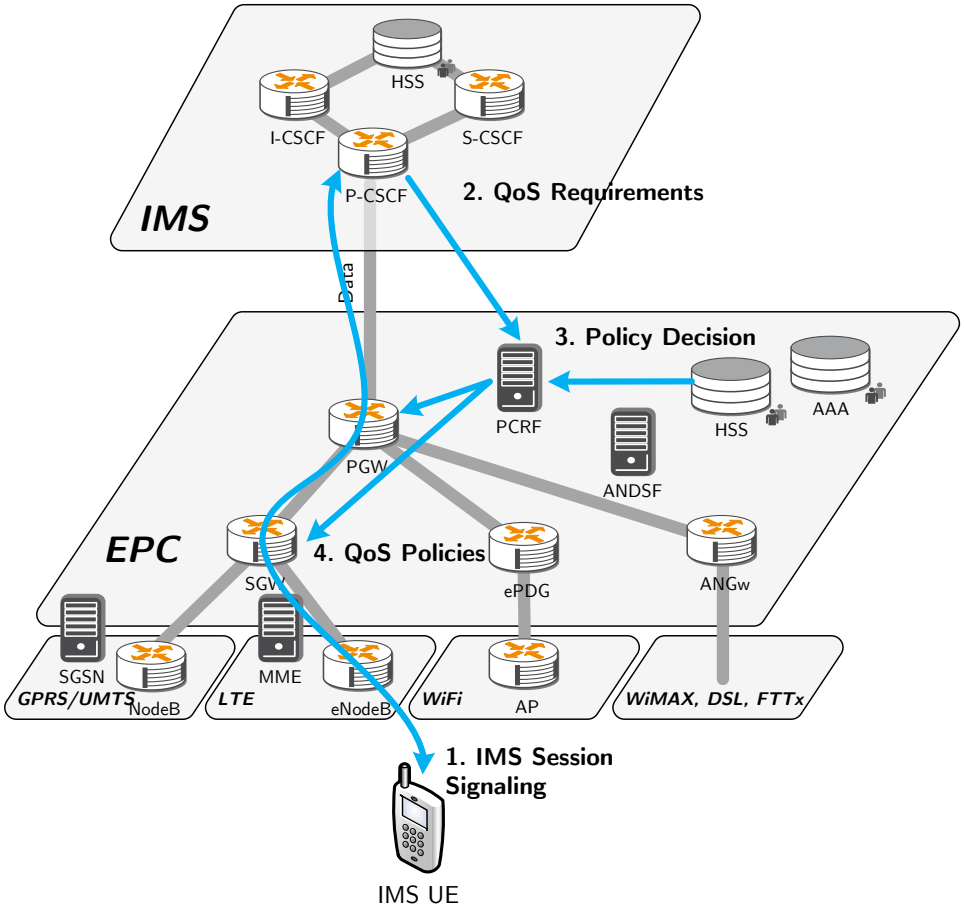


Figure 7.10: Demonstrating VoLTE with OpenIMSCore and OpenEPC

and EPC as seamless IP connectivity platform.

In this scope, the OpenIMSCore has been deeply integrated with its successor, the OpenEPC project. By exchanging commands, signaling and media bearers are managed and QoS is enforced in an ubiquitous manner, for any IMS service and over any wireless RATs. In parallel, the IMS functionality is enhanced with the transparent mobility, security and event reporting features of the underlying EPC layer, which ensure a carrier-grade delivery of services.

Both projects represent foundations of the comprehensive FUSECO Playground [16]. There the VoLTE envisioned network deployments are tested and trialed before future roll-outs. Further, advanced current R&D topics (e.g. NFV, SDN, Self-Organized Network (SON)) are being implemented and experimented with for the first time, in the continuous migration of the telecommunication infrastructures

towards the Future Internet (FI) concepts.

7.5 Conclusions on Validation

The present chapter took first a look at the design matrix introduced in the first chapters of the dissertation and analyzed how well the resulting implementation has followed on the requirements. Next a short view has been presented on the initially proposed targets, demonstrating the alignment of the results to the intended scopes.

The validation continued by providing direct results, first on the standardization, then on the industry and the adoption of IMS as a new technology suite and finishing with the impact on the academia.

While the results clearly indicate a deep use and involvement of the OpenIMSCore project for all those standards and various domains, it can be safely concluded that the methodology presented, of using Open Source to advance research by building bridges between the industry and the academia is successful for the applied use case and hence valid.

Chapter 8

Summary & Outlook

8.1 Summary

8.1.1 Results Summary

At the beginning of the dissertation, there were 3 major traction points, which also overlapped with the principal key questions:

1. **Identify the key design principles and analyze the main implementation challenges**

   As proof of the successful execution stand the requirements analysis, design, specification and implementation chapters, which result into the creation of the successful OpenIMSCore project.

2. **Reuse and improve upon Internet Open Source projects for the purpose of providing reference NGN toolkits for the test-bed environment.**

   This second point has been successfully followed by building on top of existing Open Source projects and reusing their established capabilities and features to bring up, over a short time interval, NGN relevant prototypes. The OpenIMSCore, launched as a project in 2006, has established itself as worldwide reference for test-beds.

3. **Improve R&D of NGN by using Open Source models**

The third point was also successfully achieved by attracting within the Open Source community a significant amount of academic participants and institutions, representing the base of many large scale research projects. The project acted as an incubator for academic research, as proved by the high number of papers published on the subject of practical experimentation with the OpenIMSCore. Even further, the project has inspired and helped adjacent projects to start and grow.

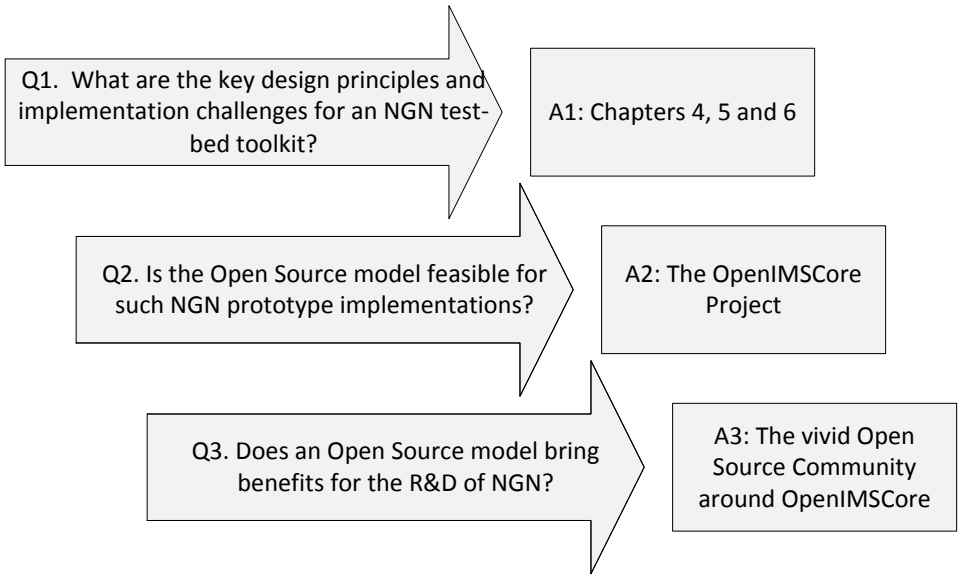


Figure 8.1: Summary Answers to the Principal Key Questions of the Dissertation

Starting from the State-of-the-Art and then composing together with the Motivation, the Requirements have been defined. From there the Design has been derived and then the Implementation followed and executed the Design. The result is the OpenIMSCore project, which became in a short time since its launching as Open Source a reference for NGN test-beds. This toolkit satisfied well the initial targets and requirements, being used in numerous projects world-wide and even spawning new related projects.

Regarding then the Key Questions (Section 1.4), which provide the main matter of the scientific contributions as resulting from the engineering process, they have all been successfully answered. Detailed answers have been provided in the main chapters of the dissertation, with summary and conclusions in the validation chapter (Section 7.2). A short overview of the answers is provided by Figure 8.1 for the principal targets, and in Figure 8.2 for the additional ones.

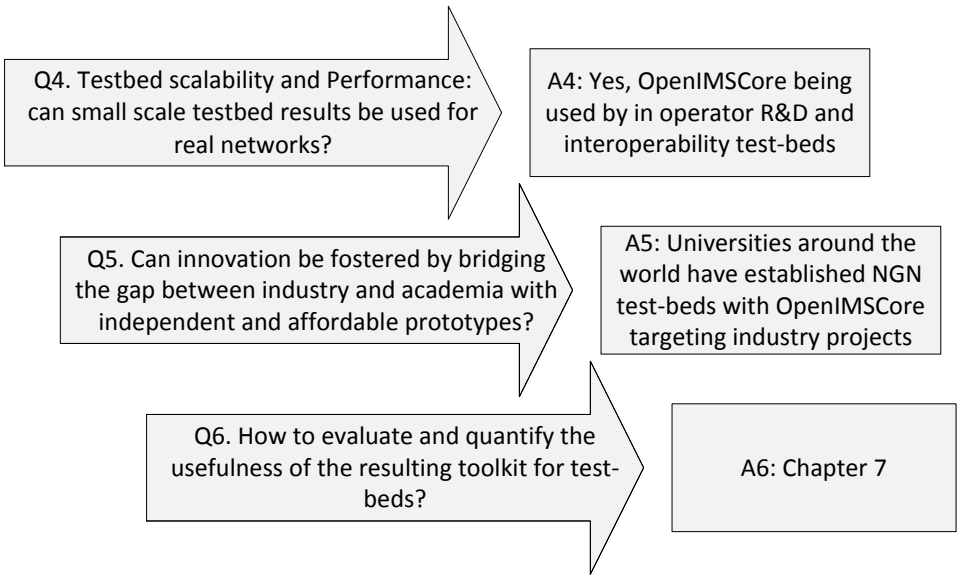


Figure 8.2: Summary Answers to the Secondary Key Questions of the Dissertation

8.1.2 Resulting Impacts

While the project itself has been the subject of tens of articles, papers, diploma theses and publications by the author and his team, the indirect impact was even greater, by empowering both academic students and industry researchers to conduct experiments and by providing them with an early access to the NGN technology set. As one of the reference activities, the ONIT series of workshops have been started and centered around the OpenIMSCore academic community, as a vehicle for exchange experiences and disseminating test-bed activities. This has seen a constant interest over the past 4 years and have been part of globally recognized IEEE and ICST conferences.

Another important impact is that the project enabled the direct use of the telecommunication core technological set directly in the university and technical school teaching process. Professors could distribute to students ready-to-run virtual images of the NGN CN. With a practical zero-cost, this allowed the learning process to happen in a vendor-independent manner and, perhaps even more important, it allowed students to directly modify the prototypes and try-out new ideas and concepts. Such a direct access was previously not possible with the traditionally closed and high-cost entry barriers of the telecommunication domain.

The OpenIMSCore was also at the core or many European Union (EU) and other national or international projects, as noted in Section 7.3.2.2. This validated

its usefulness not only in purposes of the IMS CN architecture, but also for serving as a base for more research around it, within novel service models, ever-evolving mobile broadband access and new concepts. Even today, many years after achieving the initial basic targets, it can be found at core of complex experiments for technologies like VoLTE or RCS-e.

A noticeable impact is also given by the creation and growth of the vivid open source community around the OpenIMSCore project. This resulted in a high number of researchers following the project, interesting discussion on the mailing lists as well as additional projects which have been flourished around, spinning the usefulness and relevance of the results even further.

8.2 Publications, Contributions to Standards and Presentations

A significant number of papers and articles have been published by the author as part of this work in various international conference proceedings and scientific journals, following the evolution of the toolkit since 2005. While the complete list of publications and presentations is too long to maintain the focus of this summary section, these have been listed in an additional section of the dissertation - Appendix A.

The author has authored or co-authored to date a number of 36 peer-reviewed scientific papers, journal or magazine articles[1]. These follow on the author's work on the presented as well as other projects, with the common denominator being research and establishment of test-beds for NGN Core Network (CN) architectures.

While all publications above are on NGN architectures, about 14 of these publications follow directly on the IMS topics with a special focus on the design and implementation of the OpenIMSCore project, as well as experimentation on and with the project. Worth mentioning from this subset as the most important and cited would be:

- [133] the author's engineering diploma thesis, which sparkled the implementation of the first building blocks of the OpenIMSCore;
- [242] in the proceedings of the 2nd International Workshop on Mobility Aware Technologies and Applications (MATA) 2005 - one of the first publications on the original designs and intentions for realizing an Open Source IMS Core Network (CN) project; was mostly an introductory work, yet often referenced by the OpenIMSCore users;
- [243] published in the Journal of Mobile Multimedia Volume 3, 2nd issue - it has reached a much larger audience and was cited numerous times by IMS experimenters; it provides an insight into the OpenIMSCore project goals, as well as into its technical realization;

[1] See the complete bibliographical information in Section A.1

- [244] in the proceedings of the IEEE Wireless Communications and Networking Conference (WCNC) 2006 and the following improved and extended [245] published in the IEEE Vehicular Technology Magazine Volume 2, 1st issue - presenting some of the first practical feasibility studies for using IMS as a CN architecture within wireless environments; these papers presented the first practical experiments which could be performed with the new OpenIMSCore project, while also giving a glimpse into how suitable IMS and SIP technologies are use within the wireless domain, as resulting from the first practical trials;

- [14], [13] and [246] - all following on experiences as well as potential for accelerating innovation in IMS through an Open Source model.

As the realization of the OpenIMSCore project and the associated NGN research implied significant work efforts, many colleagues and students have helped during the implementation phases of the toolkit. Numerous Master of Science (MSc) and Diploma Theses have been written at Fraunhofer FOKUS, but also around the world, 9 being so far directly supervised by the author[2].

The work also involved breaking new grounds and defining new standards[3] were none were yet available, or patenting new concepts[4].

To date, the author has participated with significant contributions in defining 4 ETSI/TISPAN technical standards related to IMS. Described in Section 7.3.1 – Influences on the Standardization, these cover mostly IMS testing aspects as required for practical evaluation and experimentation.

In the technology innovation domain, patents [247] and [248] would be worth mentioning. While these new concepts might not seem at the first sight to be related to IMS, they have been started from within the NGN test-beds, as improvements to the wireless IP connectivity domain to support IMS and to provide mechanisms for a significant reduction of IMS signalling.

As the toolkits have an international audience and they achieved a reference status recognition, the author has been invited to hold many presentation related to IMS, NGN evolution and testing, or to perform tutorials and training at internationally recognized workshop[5].

8.3 Outlook

The R&D work of the author is not limited only to IMS and the project at hand. In fact, since 2009, as IMS matured as an architecture for Next Generation Voice Services, the OpenIMSCore project entered into maturity stage. The research topics, as driven by the major TEMs have started then to relate more to the new CN architecture EPC, which was started to be defined as the IP connectivity solution

[2] See the complete bibliographical information in Section A.2
[3] See the complete bibliographical information in Section A.4
[4] See the complete bibliographical information in Section A.3
[5] See the complete bibliographical information in Section A.5

especially tailored for the LTE 4th generation of mobile radio technologies. Still, the research work of the author continued smoothly, building on the present dissertation results into the following main directions:

1. Providing and supporting IMS maturing and productization phases.

2. Starting a new CN prototype implementation for the EPC architecture, with lessons-learned from IMS.

3. Conducting research at the intersection point between IMS and EPC, especially for the purposes of providing telephony services over the 4G and future mobile radio technologies.

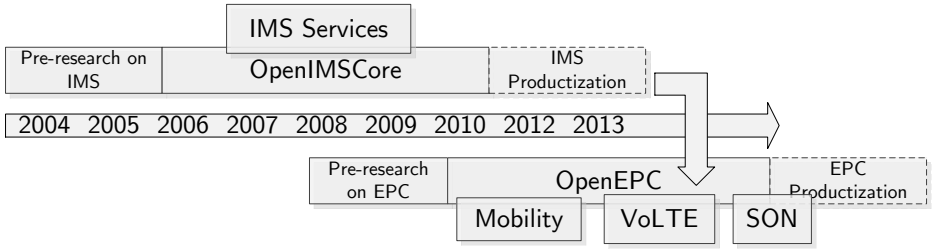


Figure 8.3: Author's Research Timeline on NGN CN Architectures

8.3.1 OpenIMSCore as catalyst for Open Source IMS in Commercial Deployments

The OpenIMSCore project, as highlighted in Chapter 4, has been designed as an exclusive toolkit for testbeds. Due to its resulting limitations, it can not be obviously used besides field-trials. This was mostly because of the reduced development budgets which limited the alignment of the prototypes to the state-of-the-art SIP proxy technologies, as well as directly prioritizing the efforts from functionality stabilization and hardening towards implementing new features and concepts.

However, a significant part of the community, which was following the project from the commercial exploitation sides in the industry, has taken over and imported almost the entire functional elements of the OpenIMSCore project back into the Kamailio/SIP-Router development trees, ultimately targeting carrier-grade implementations and deployments. While these implementations would probably not claim too soon, if ever, 100% compliance with the IMS standards, still they represent very important Open Source implementations, which will most likely power in

the near future carrier-grade telephony networks with the advantages provided by the IMS architecture.

These results, while in some cases directly sharing and reusing functional modules with the OpenIMSCore and other R&D projects, more importantly carry with them the experience gained in the test-beds. It is feasible for such Open Source evolutions to empower at least small-to-medium size telephony and other IMS-powered services. The direct advantages would then be that the academic world would have direct access to similar CN technologies used by the major network operators to provide telecommunication services on large scales around the world.

Even further, it can be argued that the existence of an independent and comprehensive technological prototype for IMS CN has enabled TEM to build their commercial products much quicker. Then regarding the operators and TSPs, they have been also enabled earlier with access to an independent implementation, allowing them to better understand IMS and properly evaluate its implications on future services and of course on the business models. Overall then the impact can be considered as an adoption accelerating one.

8.3.2 IMS as Technology Base for Voice over LTE (VoLTE)

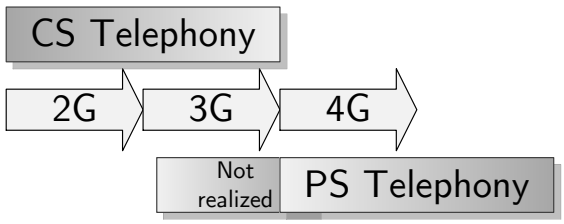


Figure 8.4: Mobile Telephony Service Between Radio Technology Generations

Even after almost 10 years since the first research steps have been started, IMS stands in 2013 as one of the principal topics to be evaluated. While the initial deployment target envisioned by 3GPP was to happen during the 3G technological evaluation, when the CS and PS architectural options were overlapping, this unfortunately failed to materialize.

However, with the introduction of 4G/LTE RAN, the CS-based services have no longer been considered in the new architecture, making the evolution and migration a difficult topic to realize. Multiple in-between solutions have been considered, as for example to fallback to legacy radio technologies (CS Fallback (CSFB)), or to tunnel legacy signaling over IP (VoLTE via Generic Access (VoLGA)), still the best

solution, as agreed by most of the industry, remains IMS. This fact implies that the OpenIMSCore actually experiences a boost in interest, as many research labs now turn to field-trials and optimizations for their VoLTE near-future deployments.

Of special interest here is of course the interfacing with the EPC CN for the purpose of obtaining seamless mobility, QoS and other features which can be better optimized by the mobile infrastructure operators. Also another hot research topic is that of Single Radio Voice Call Continuity (SRVCC), as telephony services must be provided seamlessly, even during hand-overs between CS, PS 3GPP wireless radio networks, as well as over non-3GPP cost-effective solutions like WiFi.

And of course, the IMS services topic is not limited to telephony, but extends well beyond. With the introduction of Multimedia Telephony Service (MMTel), RCS or the latest RCS-e [228], known to the consumers under the `joyn` brand [249], the productization of IMS comes today driven by this set of services. This breathes new life into the IMS research, which now changes from one targeting the core architecture, to one focusing on the services.

8.3.3 Reuse of the Methodology and Variations of the Test-bed Model within OpenEPC

The lessons learned exposed in this dissertation, on the subject of the OpenIMSCore project, are today reused in the latest R&D project led by the author, the OpenEPC. In the fast-paced evolution of mobile networks today, IMS remained quite specific to providing telephony applications, while new architectures overtook its capabilities towards generalizing support for IP application. Such is the case with EPC, where also a more generalized approach had to be taken. Additionally, a much larger architectural scope has been considered, with many more functional components, protocol stacks, interface points and overall a full set of technologies ranging from 2G up to the latest LTE advances as well as non-3GPP access options. Still, the main principles from this work have been applied there too, again repeating the success, albeit on a different plane.

OpenEPC integrates well with OpenIMSCore, especially as the evolutionary change of the architectures should not be regarded though as a replacement of IMS with EPC, but the two are set to co-exist - one providing operator optimized services and the other providing the high-performance connectivity. In effect, today every deployment and test-bed of the OpenEPC project includes the OpenIMSCore platform, primarily as a demonstration enabler, but also as a solid complement with its services.

The same principles have been applied, although due to the sheer difference in required efforts for design and implementation, an Open Source model could not be followed. Still, even though the licensing model is in this case based on a paid model, the same design methodology has been applied. The results have been as of 2013 hugely successful as 35+ operators, vendors and universities have licensed and adopted the project for their test-bed purposes. This results into a further validation of the methodology, which proved to be feasible even in a much larger

scope, with the associated adaptations and modifications.

Getting back to the OpenIMSCore, even at the leading cutting edges of research today, within SDN and NFV concepts, OpenIMSCore continues to be used as an independent proof of concept, exploiting its openness to enhance it with cloud-ready functionality. Expected for the future is that network elements will be heavily virtualized and as such benefit from the newly seeked uniformity of the telecommunication data centers in their path towards "cloud-ification".

8.4 Final Words

One folk-saying states, roughly translated to English: *I wish I had the after-thought before I started.* On this line, whenever at the end of a big project one thinks that repeating it a second time would be the same, then nothing was learned. No matter how many efforts have been done to eliminate as many mistakes from this work, without doubt there are still many within. More important though is to learn from those mistakes, as to not waste time repeating them in the future, but also to keep moving forward towards the bigger goals even if perfection can not always be obtained.

The dissertation followed on the design, implementation and exploitation of a particular project aiming to provide test-bed infrastructure for upcoming new telecommunication network architectures. Open Source has been used as a momentum enhancer, as to ensure reaching critical mass and as to increase the relevance of the prototypes to the research community. Whether in the end IMS is successful as an architecture is probably not relevant. However, the experiences gained are very much so, within work to design the next generation, improve and evolve the networking concepts.

As we are today experiencing a break-neck evolution of communication, probably the concepts of IMS exposed here will be deprecated in less than a decade. Hopefully though the idea of using Open Source to bring academia and the industry together will survive and then this will be relevant for a longer time from methodology point of view.

The lessons learned from OpenIMSCore are applied by the author and the research continues today at a bigger scale within the OpenEPC project. There the results of this work have tremendously helped in realizing even more complex CN concepts and cover in shorter time even more technologies and concepts. Hence the never-ending battle of seeking a better telecommunication architecture continues, peeking at things to come beyond NGN and IMS.

To be continued within EPC, SDN and beyond!

Bibliography

[1] Bennett, R.L. and G.E. Policello II: *Switching systems in the 21st century*. Communications Magazine, IEEE, 31(3):24 –28, march 1993, ISSN 0163-6804. (Cited on pages xv, 25 and 26.)

[2] Magedanz, Thomas: *Intelligent Network Evolution - Impact of Internet, CORBA, TINA and Mobile Agent Technologies*. In *Telecommunications Information Networking Architecture Conference 1999*. Hawaii, USA, April 1999. http://www.tinac.com/conference/tina99/tutorial_p.pdf. (Cited on pages xv and 30.)

[3] Magedanz, Thomas and Radu Popescu-Zeletin: *Intelligent Networks*. International Thomson Computer Press, 1996, ISBN 9781850322931. http://books.google.de/books?id=_hMfAQAAIAAJ. (Cited on pages xv and 30.)

[4] Heart, F., A. McKenzie, J. McQuillian, and D. Walden: *ARPANET Completion Report, Bolt, Beranek and Newman*, 1978. (Cited on pages xv and 38.)

[5] IETF: *RFC791 - Internet Protocol*, September 1981. http://tools.ietf.org/html/rfc791. (Cited on pages xv, 39, 40 and 42.)

[6] IETF: *RFC768 - User Datagram Protocol*, August 1980. http://tools.ietf.org/html/rfc768. (Cited on pages xv, 39 and 43.)

[7] IETF: *RFC793 - Transmission Control Protocol*, September 1981. http://tools.ietf.org/html/rfc793. (Cited on pages xv, 39 and 44.)

[8] Beijnum, Iljitsch van: *2010 IPv4 Address Use Report*, June 2011. http://www.bgpexpert.com/addrspace2010.php. (Cited on pages xv, 45 and 46.)

[9] IETF: *RFC2460 - Internet Protocol, Version 6 (IPv6) Specification*, December 1998. http://tools.ietf.org/html/rfc2460. (Cited on pages xv, 45 and 46.)

[10] TISPAN: *Defining the Next Generation Network*, October 2008. http://www.etsi.org/tispan/. (Cited on pages xv, 53 and 54.)

[11] ETSI: *Common IMS to be centred in the 3GPP Services Specification Group*, June 2007. http://www.etsi.org/website/NewsandEvents/2007_06_common_ims.aspx. (Cited on pages xv, 2, 55 and 145.)

[12] Vingarzan, Dragos *et al.*: *IMS/NGN Performance Benchmark Part 1: Core Concepts*, 2007. http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=24161, ETSI/TISPAN 6 Workitem 06024-1. (Cited on pages xvii, 104, 178, 198, 199, 201 and 271.)

[13] Vingarzan, Dragos: *IMS Network Testing (INT); IMS & EPC Interoperability test descriptions*, 2011. http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=34724, ETSI/INT Workitem INT/DTS00050. (Cited on pages xvii and 202.)

[14] Magedanz, Thomas, Peter Weik, Dragos Vingarzan, Fabricio Carvalho de Gouveia, and Sebastian Wahle: *Experiences on the Establishment and Provisioning of NGN/IMS Testbeds - The FOKUS Open IMS Playground and the Related Open Source IMS Core*. In *11th International Conference on Intelligence in Service Delivery Network 2007 (ICIN 2007)*. Bordeaux, France, October 2007. http://www.icin.biz/files/programmes/Session2B-2.pdf. (Cited on pages xvii, 206 and 219.)

[15] Fraunhofer FOKUS: *Open SOA Telco Playground*, 2008. http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_soa_telco_playground/home/at_a_glance/index.html. (Cited on pages xvii, 207 and 208.)

[16] Fraunhofer FOKUS: *The FUture SEamless COmmunication (FUSECO) Playground*, 2010. http://www.fokus.fraunhofer.de/en/fokus_testbeds/fuseco_playground/index.html. (Cited on pages xvii, 207, 209 and 212.)

[17] IEEE GLOBECOM Workshops: *4th International IEEE Workshop on Open NGN and IMS Testbeds (ONIT 2012) @ GLOBECOM 2012 - Open Source Tools and Testbeds for Fixed and Mobile Next Generation Network Evolution toward the Future Workshop*, December 2012. http://www.onit-ws.org/2012/. (Cited on pages xvii and 211.)

[18] ITU-T: *The World in 2013 - ICT Facts and Figures*. Technical report, February 2013. http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013.pdf. (Cited on pages 1 and 102.)

[19] ITU-T: *ITU-T Recommendation Y.2001 (12/2004) - General overview of NGN*, December 2004. http://www.itu.int/rec/T-REC-Y.2001. (Cited on pages 1, 6 and 52.)

[20] 3GPP: *TS 23.228 - IP Multimedia Subsystem (IMS); Stage 2*. http://www.3gpp.org/ftp/Specs/html-info/23228.htm. (Cited on pages 1 and 59.)

[21] 3GPP: *Service Architecture Evolution*. http://www.3gpp.org/LTE.html. (Cited on pages 1 and 65.)

[22] European Future Internet Portal: *Future Internet*. http://www.future-internet.eu/. (Cited on page 1.)

[23] *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Std 610.12-1990, pages 1–84, 1990. http://standards.ieee.org/findstds/standard/610.12-1990.html. (Cited on page 7.)

[24] OECD: *Frascati manual 2002: Proposed standard practice for surveys on research and experimental development*, 2002, ISBN 978-92-64-19903-9. http://www.uis.unesco.org/Library/Documents/OECDFrascatiManual02_en.pdf. (Cited on page 7.)

[25] Raymond, Eric S.: *Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2nd edition, 2001, ISBN 0596001312. (Cited on page 8.)

[26] Fraunhofer FOKUS: *The Open Source IMS Core Project*. http://www.openimscore.org. (Cited on page 11.)

[27] iptel.org: *The SIP Express Router*. http://www.iptel.org/ser. (Cited on pages 18, 90 and 175.)

[28] Bell, Alexander Graham: *Patent no. 174,465 - Improvement in Telegraphy*. United States Patent Office, March 1876. (Cited on page 24.)

[29] Ambrosch, W. D., A. Maher, and B. Sasscer (editors): *The intelligent network: a joint study by Bell Atlantic, IBM and Siemens*. Springer-Verlag New York, Inc., New York, NY, USA, 1989, ISBN 3-540-50897-X. (Cited on page 26.)

[30] Modarressi, A.R. and R.A. Skoog: *Signaling System No.7: a tutorial*. Communications Magazine, IEEE, 28(7):19 –20, july 1990, ISSN 0163-6804. (Cited on page 27.)

[31] ITU-T: *SERIES Q: SWITCHING AND SIGNALLING - Specifications of Signaling System No. 6*, November 1998. http://www.itu.int/rec/T-REC-Q.251-Q.300/en. (Cited on page 27.)

[32] ITU-T: *Q.700 - Specifications of Signaling System No.7; Introduction to CCITT Signaling System No.7*, March 1993. http://www.itu.int/rec/T-REC-Q.700/en. (Cited on pages 27 and 67.)

[33] ITU-T: *SERIES X: Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*, July 1994. http://www.itu.int/rec/T-REC-X.200/en. (Cited on pages 27 and 37.)

[34] ITU-T: *Q.702 - Specifications of Signaling System No.7; Signaling Data Link*, November 1988. http://www.itu.int/rec/T-REC-Q.702/en. (Cited on page 27.)

[35] ITU-T: *Q.703 - Specifications of Signaling System No.7; Message transfer part; Signaling Link*, July 1996. http://www.itu.int/rec/T-REC-Q.703/en. (Cited on page 27.)

[36] ITU-T: *Q.704 - Specifications of Signaling System No.7; Message transfer part; Signaling Network Functions and Messages*, July 1996. http://www.itu.int/rec/T-REC-Q.704/en. (Cited on page 27.)

[37] IETF: *RFC3331 - Signaling System 7 (SS7) Message Transfer Part 2 (MTP2) - User Adaptation Layer*, September 2002. http://tools.ietf.org/html/rfc3331. (Cited on page 28.)

[38] IETF: *RFC4666 - Signaling System 7 (SS7) Message Transfer Part 3 (MTP3) - User Adaptation Layer (M3UA)*, September 2006. http://tools.ietf.org/html/rfc4666. (Cited on page 28.)

[39] IETF: *RFC4165 - Signaling System 7 (SS7) Message Transfer Part 2 (MTP2) - User Peer-to-Peer Adaptation Layer (M2PA)*, September 2005. http://tools.ietf.org/html/rfc4165. (Cited on page 28.)

[40] IETF: *RFC4960 - Stream Control Transmission Protocol*, September 2007. http://tools.ietf.org/html/rfc4960. (Cited on pages 28 and 44.)

[41] ITU-T: *Q.711 - Specifications of Signaling System No.7; Signaling Connection Control Part (SCCP); Functional Description of the Signaling Connection Control Part*, March 2001. http://www.itu.int/rec/T-REC-Q.711/en. (Cited on page 28.)

[42] ITU-T: *Q.761 - Specifications of Signaling System No.7; ISDN User Part Functional Description*, December 1999. http://www.itu.int/rec/T-REC-Q.761/en. (Cited on page 28.)

[43] ITU-T: *Q.771 - Specifications of Signaling System No.7; Transaction Capabilities Application Part - Functional Description of Transaction Capabilities*, June 1997. http://www.itu.int/rec/T-REC-Q.771/en. (Cited on page 29.)

[44] 3GPP: *TS 29.078 - Customised Applications for Mobile network Enhanced Logic (CAMEL) Phase X; CAMEL Application Part (CAP) specification.* http://www.3gpp.org/ftp/Specs/html-info/29078.htm. (Cited on page 29.)

[45] ITU-T: *General Recommendations on Telephone Switching and Signaling; Intelligent Network; Introduction to Intelligent Network Capability Set 1*, March 1993. http://www.itu.int/rec/T-REC-Q.1211/en. (Cited on pages 31 and 32.)

[46] ITU-T: *SERIES Q: SWITCHING AND SIGNALLING; Intelligent Network; Introduction to Intelligent Network Capability Set 2*, September 1997. http://www.itu.int/rec/T-REC-Q.1221/en. (Cited on pages 32 and 33.)

[47] ITU-T: *SERIES Q: SWITCHING AND SIGNALLING; Intelligent Network; Introduction to Intelligent Network Capability Set 3*, December 1999. http://www.itu.int/rec/T-REC-Q.1231/en. (Cited on page 33.)

[48] ITU-T: *H.323 - SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS; Infrastructure of audiovisual services – Systems and terminal equipment for audiovisual services; Packet-based multimedia communications systems*, December 2009. http://www.itu.int/rec/T-REC-H.323/en/. (Cited on pages 33, 53 and 68.)

[49] IETF: *RFC3261 - SIP: Session Initiation Protocol*, June 2002. http://tools.ietf.org/html/rfc3261. (Cited on pages 33, 53, 68, 69, 71, 73, 90, 99, 140, 143, 159, 160, 163 and 164.)

[50] ITU-T: *SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS; Infrastructure of audiovisual services - Communication procedures; Gateway control protocol: Version 3*, September 2005. http://www.itu.int/rec/T-REC-H.248.1/en. (Cited on page 33.)

[51] IETF: *RFC3015 - Megaco Protocol Version 1.0*, November 2000. http://tools.ietf.org/html/rfc3015. (Cited on page 33.)

[52] IETF: *RFC3525 - Gateway Control Protocol Version 1*, June 2003. http://tools.ietf.org/html/rfc3525. (Cited on page 33.)

[53] ITU-T: *SERIES Q: SWITCHING AND SIGNALLING; Intelligent Network; Introduction to Intelligent Network Capability Set 4*, July 2001. http://www.itu.int/rec/T-REC-Q.1241/en. (Cited on page 33.)

[54] Noldus, Rogier: *CAMEL: intelligent networks for the GSM, GPRS and UMTS network*. Wiley, 2006, ISBN 9780470016947. http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470016949.html. (Cited on page 34.)

[55] 3GPP: *TS 02.78 - Customized Applications for Mobile network Enhanced Logic (CAMEL); Service definition (Stage 1)*. http://www.3gpp.org/ftp/Specs/html-info/0278.htm. (Cited on page 34.)

[56] 3GPP: *TS 22.078 - Customized Applications for Mobile network Enhanced Logic (CAMEL); Service description; Stage 1*. http://www.3gpp.org/ftp/Specs/html-info/22078.htm. (Cited on page 34.)

[57] Java Community Process: *JSR 22: JAIN SLEE API Specification*. http://jcp.org/en/jsr/detail?id=22. (Cited on page 35.)

[58] Java Community Process: *JSR 240: JAIN SLEE (JSLEE) v1.1*. http://jcp.org/en/jsr/detail?id=240. (Cited on page 35.)

[59] ETSI: *Open Service Access (OSA)*. http://etsi.org/WebSite/Technologies/OSA.aspx. (Cited on page 35.)

[60] 3GPP: *TS 29.199-01 - Open Service Access (OSA); Parlay X web services; Part 1: Common*. http://www.3gpp.org/ftp/Specs/html-info/29199-01.htm. (Cited on page 35.)

[61] GSMA: *GSMA OneAPI*. http://oneapi.gsma.com/. (Cited on page 35.)

[62] Fielding, Roy Thomas: *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvina, 2000. http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm. (Cited on page 36.)

[63] O'Reilly, Tim: *REST vs. SOAP at Amazon*, April 2003. http://www.oreillynet.com/pub/wlg/3005. (Cited on page 36.)

[64] Dryburgh, L. and J. Hewett: *Signaling System No. 7 (SS7/C7): Protocol, Architecture, and Services*. Networking Technology Series. Cisco, 2005, ISBN 9781587050404. http://www.ciscopress.com/bookstore/product.asp?isbn=1587050404. (Cited on page 36.)

[65] Licklider, J. C. R.: *Man-Computer Symbiosis*. Human Factors in Electronics, IRE Transactions on, HFE-1(1):4 –11, March 1960. (Cited on page 37.)

[66] Licklider, J. C. R. and Welden E. Clark: *On-line man-computer communication*. In *Proceedings of the May 1-3, 1962, spring joint computer conference*, AIEE-IRE '62 (Spring), pages 113–128, New York, NY, USA, 1962. ACM. http://doi.acm.org/10.1145/1460833.1460847. (Cited on page 37.)

[67] Bolt Beranek and Newman Inc.: *Report No. 1822 - Interface Message Processor - Specifications for the Interconnection of a Host and an IMP*, January 1976. http://www.bitsavers.org/pdf/bbn/imp/BBN1822_Jan1976.pdf. (Cited on page 37.)

[68] Cerf, Vinton G. and Robert E. Kahn: *A Protocol for Packet Network Intercommunication*. Communications, IEEE Transactions on, 22(5):637 – 648, may 1974, ISSN 0090-6778. (Cited on pages 37, 43 and 45.)

[69] Hauben, Ronda: *From the ARPANET to the Internet - A Study of the ARPANET TCP/IP Digest and of the Role of Online Communication in the Transition from the ARPANET to the Internet*, June 1998. http://www.columbia.edu/~rh120/other/tcpdigest_paper.txt. (Cited on page 39.)

[70] IETF: *RFC792 - Internet Control Message Protocol*, September 1981. http://tools.ietf.org/html/rfc792. (Cited on page 40.)

[71] IETF: *RFC3286 - An Introduction to the Stream Control Transmission Protocol (SCTP)*, May 2002. http://tools.ietf.org/html/rfc3286. (Cited on page 44.)

[72] IETF: *RFC1981 - Path MTU Discovery for IP version 6*, August 1996. http://tools.ietf.org/html/rfc1981. (Cited on page 47.)

[73] IETF: *RFC2675 - IPv6 Jumbograms*, August 1999. http://tools.ietf.org/html/rfc2675. (Cited on page 47.)

[74] IETF: *RFC1958 - Architecture Principles of the Internet*, June 1996. http://tools.ietf.org/html/rfc1958. (Cited on pages 47, 49 and 50.)

[75] Willinger, Water and John Doyle: *Robustness and the Internet: Design and evolution*, March 2002. http://www.maoz.com/~dmm/complexity_and_the_internet/robustness_and_the_internet_design_and_evolution.pdf. (Cited on page 48.)

[76] IETF: *RFC3439 - Some Internet Architectural Guidelines and Philosophy*, December 2002. http://tools.ietf.org/html/rfc3439. (Cited on pages 49 and 50.)

[77] Scott, D.: *Making Smart Investments to Reduce Unplanned Downtime*. Tactical Guidelines, TG-07-4033, March 1999. http://m106.maoz.com/~dmm/complexity_and_the_internet/downtime.pdf. (Cited on page 49.)

[78] ISO/IEC: *ISO/IEC 7498-1 - Information Technology - Open Systems Interconnection - Basic Reference Mode: The Basic Model*, November 1994. http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?ics1=35&ics2=100&ics3=1&csnumber=20269. (Cited on page 50.)

[79] Kawadia, Vikas and P.R. Kumar: *A cautionary perspective on cross-layer design*. Wireless Communications, IEEE, 12(1):3–11, 2005, ISSN 1536-1284. (Cited on page 50.)

[80] Smirnov, Mikhail: *Advanced Internet Services, Winter Semester 2010/2011, Technische Universitaet Berlin*, October 2010. http://autonomic.fokus.fraunhofer.de/teaching/ais/slides/1112/01-overview-arch.pdf. (Cited on page 50.)

[81] Tim Hills: *What's Up With IMS?* http://www.lightreading.com/ip-convergence/whats-up-with-ims/240077682. (Cited on page 57.)

[82] Gustafsson, E. and A. Jonsson: *Always best connected*. Wireless Communications, IEEE, 10(1):49–55, feb. 2003, ISSN 1536-1284. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1182111. (Cited on page 66.)

[83] 3GPP: *TS 24.312 - Access Network Discovery and Selection Function (ANDSF) Management Object (MO)*. http://www.3gpp.org/ftp/Specs/html-info/24312.htm. (Cited on page 66.)

[84] IETF: *RFC3588 - Diameter Base Protocol*, September 2003. http://tools.ietf.org/html/rfc3588. (Cited on pages 67 and 75.)

[85] IETF: *Multiparty Multimedia Session Control (mmusic) Working Group*. http://datatracker.ietf.org/wg/mmusic/charter/. (Cited on page 68.)

[86] IETF: *RFC2705 - Media Gateway Control Protocol (MGCP) Version 1.0*, October 1999. http://tools.ietf.org/html/rfc2705. (Cited on page 69.)

[87] IETF: *RFC3435 - Media Gateway Control Protocol (MGCP) Version 1.0*, January 2003. http://tools.ietf.org/html/rfc3435. (Cited on page 69.)

[88] IETF: *RFC2543 - SIP: Session Initiation Protocol*, March 1999. http://tools.ietf.org/html/rfc2543. (Cited on pages 69 and 90.)

[89] IETF: *RFC2616 - Hypertext Transfer Protocol – HTTP/1.1*, June 1999. http://tools.ietf.org/html/rfc2616. (Cited on page 69.)

[90] IETF: *RFC2327 - SDP: Session Description Protocol*, April 1998. http://tools.ietf.org/html/rfc2327. (Cited on page 70.)

[91] IETF: *RFC3266 - Support for IPv6 in Session Description Protocol (SDP)*, June 2002. http://tools.ietf.org/html/rfc3266. (Cited on page 70.)

[92] IETF: *RFC4566 - SDP: Session Description Protocol*, July 2006. http://tools.ietf.org/html/rfc4566. (Cited on page 70.)

[93] IETF: *RFC3265 - Session Initiation Protocol (SIP)-Specific Event Notification*, June 2002. http://tools.ietf.org/html/rfc3265. (Cited on pages 71, 73 and 159.)

[94] IETF: *RFC3903 - Session Initiation Protocol (SIP) Extension for Event State Publication*, October 2004. http://tools.ietf.org/html/rfc3903. (Cited on page 71.)

[95] IETF: *RFC3428 - Session Initiation Protocol (SIP) Extension for Instant Messaging*, December 2002. http://tools.ietf.org/html/rfc3428. (Cited on page 71.)

[96] IETF: *RFC3515 - The Session Initiation Protocol (SIP) Refer Method*, April 2003. http://tools.ietf.org/html/rfc3515. (Cited on page 71.)

[97] IETF: *RFC3262 - Reliability of Provisional Responses in the Session Initiation Protocol (SIP)*, June 2002. http://tools.ietf.org/html/rfc3262. (Cited on page 71.)

[98] IETF: *RFC3311 - The Session Initiation Protocol (SIP) UPDATE Method*, September 2002. http://tools.ietf.org/html/rfc3311. (Cited on page 71.)

[99] IETF: *RFC2976 - The SIP INFO Method*, October 2000. http://tools.ietf.org/html/rfc2976. (Cited on page 71.)

[100] IETF: *RFC3969 - Uniform Resource Identifier (URI) Parameter Registry for the Session Initiation Protocol (SIP)*, December 2004. http://tools.ietf.org/html/rfc3969. (Cited on pages 71 and 127.)

[101] IETF: *RFC3966 - The tel URI for Telephone Numbers*, December 2004. http://tools.ietf.org/html/rfc3969. (Cited on pages 71 and 128.)

[102] IETF: *RFC5031 - A Uniform Resource Name (URN) for Emergency and Other Well-Known Services*, January 2008. http://tools.ietf.org/html/rfc5031. (Cited on pages 71 and 128.)

[103] IETF: *RFC3608 - Session Initiation Protocol (SIP) Extension Header Field for Service Route Discovery During Registration*, October 2003. http://tools.ietf.org/html/rfc3608. (Cited on page 73.)

[104] IETF: *RFC4168 - The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP)*, October 2005. http://tools.ietf.org/html/rfc4168. (Cited on page 74.)

[105] IETF: *RFC1492 - An Access Control Protocol, Sometimes Called TACACS*, July 1993. http://tools.ietf.org/html/rfc1492. (Cited on page 74.)

[106] IETF: *RFC2058 - Remote Authentication Dial In User Service (RADIUS)*, January 1997. http://tools.ietf.org/html/rfc2058. (Cited on page 74.)

[107] IETF: *RFC2138 - Remote Authentication Dial In User Service (RADIUS)*, April 1997. http://tools.ietf.org/html/rfc2138. (Cited on page 74.)

[108] IETF: *RFC2865 - Remote Authentication Dial In User Service (RADIUS)*, June 2000. http://tools.ietf.org/html/rfc2865. (Cited on page 74.)

[109] IETF: *RFC2866 - RADIUS Accounting*, June 2000. http://tools.ietf.org/html/rfc2866. (Cited on page 74.)

[110] IETF: *RFC4004 - Diameter Mobile IPv4 Application*, August 2005. http://tools.ietf.org/html/rfc4004. (Cited on page 79.)

[111] IETF: *RFC4005 - Diameter Network Access Server Application*, August 2005. http://tools.ietf.org/html/rfc4005. (Cited on page 79.)

[112] IETF: *RFC4006 - Diameter Credit-Control Application*, August 2005. http://tools.ietf.org/html/rfc4006. (Cited on page 79.)

[113] IETF: *RFC4072 - Diameter Extensible Authentication Protocol (EAP) Application*, August 2005. http://tools.ietf.org/html/rfc4072. (Cited on page 79.)

[114] IETF: *RFC4740 - Diameter Session Initiation Protocol (SIP) Application*, November 2006. http://tools.ietf.org/html/rfc4740. (Cited on page 79.)

[115] Free Software Foundation, Inc.: *About the GNU Operating System*. http://www.gnu.org/gnu/gnu.html. (Cited on page 78.)

[116] Free Software Foundation, Inc.: *GNU's Bulletin, Volume 1 No. 1*, February 1986. http://www.gnu.org/bulletins/bull1.txt. (Cited on page 78.)

[117] Free Software Foundation, Inc.: *The Free Software Definition*, February 1986. http://www.gnu.org/philosophy/free-sw.html. (Cited on page 79.)

[118] Perens, Bruce: *The Debian Free Software Guidelines - Debian's "Social Contract" with the Free Software Community*, July 1997. http://lists.debian.org/debian-announce/1997/msg00017.html. (Cited on page 80.)

[119] Open Source Initiative: *The Open Source Definition*. http://www.opensource.org/docs/osd. (Cited on page 80.)

[120] Free Software Foundation, Inc.: *Various Licenses and Comments about Them*. http://www.gnu.org/licenses/license-list.html. (Cited on page 82.)

[121] Open Source Initiative: *Open Source Licenses*. http://www.opensource.org/licenses/alphabetical. (Cited on page 82.)

[122] Free Software Foundation: *GNU General Public Licenses*. http://www.gnu.org/licenses/gpl.html. (Cited on page 82.)

[123] W3Techs: *Usage of operating systems for websites*, September 2010. http://w3techs.com/technologies/overview/operating_system/all. (Cited on page 83.)

[124] Torvalds, Linus: *The Linux Kernel*. http://www.kernel.org/. (Cited on page 83.)

[125] TOP500.org: *Operating system Family share for 11/2010*, November 2010. http://top500.org/stats/list/36/osfam. (Cited on page 83.)

[126] Google, Inc.: *The Android Open Source Project*. http://www.android.com/. (Cited on page 83.)

[127] IDC Worldwide Mobile Phone Tracker: *Android Pushes Past 80% Market Share While Windows Phone Shipments Leap 156.0% Year Over Year in the Third Quarter, According to IDC*, November 2013. http://www.idc.com/getdoc.jsp?containerId=prUS24442013. (Cited on page 83.)

[128] The Apache Software Foundation: *The Apache HTTP Server Project: open-source HTTP server for modern operating systems including UNIX and Windows NT*. http://httpd.apache.org/. (Cited on pages 83 and 93.)

[129] Oracle Corporation: *MySQL: The World's most popular open source database*. http://www.mysql.com/. (Cited on pages 83 and 92.)

[130] The Wireshark Foundation: *Wireshark, the world's foremost network protocol analyzer*. http://www.wireshark.org/. (Cited on page 84.)

[131] iptel.org: *SER History*. http://www.iptel.org/ser_history/. (Cited on page 90.)

[132] Westerholt, Henning: *Linux at 1&1* , May 2011. http://www.kamailio.org/events/2011-linuxtag/linux-at-1and1.pdf. (Cited on page 90.)

[133] Vingarzan, Dragos: *A 3rdGeneration IP Multimedia Session Handling implementation into SIP Express Router*. Diploma-thesis, "Politehnica" University of Bucharest, February 2005. (Cited on pages 92, 169, 175, 176, 181 and 218.)

[134] Netcraft: *March 2012 Web Server Survey*, March 2012. http://news.netcraft.com/archives/2012/03/05/march-2012-web-server-survey.html. (Cited on page 93.)

[135] The PHP Group: *PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML*. http://www.php.net/. (Cited on page 93.)

[136] Foundation, The Apache Software: *Apache Tomcat, the open source software implementation of the Java Servlet and JavaServer Pages technologies*. http://tomcat.apache.org/. (Cited on page 94.)

[137] (Wirlab), Billy Biggs *et al.*: *KPhone: the SIP UA for Linux*. http://sourceforge.net/projects/kphone/. (Cited on pages 95 and 176.)

[138] Portugal Telecom Inovacao: *The IMS Communicator*. http://imscommunicator.berlios.de/. (Cited on pages 95 and 210.)

[139] Jacques, Olivier and HP: *SIPp: the free Open Source test tool / traffic generator for the SIP protocol*. http://sipp.sourceforge.net/. (Cited on page 95.)

[140] Yukio, Okada and Kei Mochida: *HOTARU(open source IMS)*. http://ec.europa.eu/information_society/activities/foi/research/eu-japan/prog/docs/day2ndam/testbeds_and_experimentation1-2/yokada.pdf,http://www.slashdocs.com/wyxvz/yokada.html. (Cited on page 95.)

[141] Kamailio: *IMS Extensions Available for Testing*, January 2011. http://www.kamailio.org/w/2011/01/ims-extensions-available-for-testing/. (Cited on pages 96, 195 and 210.)

[142] Bock, Carsten *et al.*: *OpenHSS - Open-Source-HSS for IMS* . http://sourceforge.net/projects/openhss/. (Cited on page 96.)

[143] Tekelec: *Tekelec X-CSCF*. http://www.tekelec.com/resource-center/briefs/call-session-control-function-cscf-.asp. (Cited on page 96.)

[144] Digium, Inc.: *Asterisk: the open source framework for building communication applications*. http://www.asterisk.org/. (Cited on page 96.)

[145] Minessale, Anthony: *FreeSWITCH: a scalable open source cross-platform telephony platform designed to route and interconnect popular communication protocols using audio, video, text or any other form of media.* http://www.freeswitch.org/. (Cited on page 96.)

[146] 3GPP: *TS 24.229 - IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3.* http://www.3gpp.org/ftp/Specs/html-info/24229.htm. (Cited on pages 99, 101 and 145.)

[147] 3GPP: *TS 33.102 - 3G security; Security architecture.* http://www.3gpp.org/ftp/Specs/html-info/33102.htm. (Cited on pages 100, 145 and 146.)

[148] 3GPP: *TS 33.203 - 3G security; Access security for IP-based services.* http://www.3gpp.org/ftp/Specs/html-info/33203.htm. (Cited on pages 100, 145, 146, 150, 151, 152, 153 and 154.)

[149] IETF: *RFC3310 - Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)*, September 2002. http://tools.ietf.org/html/rfc3310. (Cited on pages 100, 145 and 146.)

[150] IETF: *RFC4169 - Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA) Version-2*, November 2005. http://tools.ietf.org/html/rfc4169. (Cited on page 100.)

[151] China Mobile: *China Mobile Operation Data - Customer Numbers*, October 2013. http://www.chinamobileltd.com/en/ir/operation.php?section=number. (Cited on page 102.)

[152] Vingarzan, Dragos *et al.*: *IMS/NGN Performance Benchmark Part 2: Subsystem Configurations and Benchmarks*, 2007. http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=25501, ETSI/TISPAN 6 Workitem 06024-2. (Cited on pages 104, 178, 199 and 271.)

[153] Vingarzan, Dragos *et al.*: *IMS/NGN Performance Benchmark Part 3: Traffic Sets and Traffic Profiles*, 2007. http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=25502, ETSI/TISPAN 6 Workitem 06024-3. (Cited on pages 104, 178, 199 and 271.)

[154] Cortes, Mauricio, J. Robert Ensor, and Jairo O. Esteban: *On SIP Performance.* Bell Labs Technical Journal, 9(3):155–172, 2004, ISSN 1538-7305. http://dx.doi.org/10.1002/bltj.20048. (Cited on pages 104 and 177.)

[155] Juran, Joseph M.: *Managerial breakthrough: The classic book on improving management performance / The Pareto Principle.* McGraw-Hill, 1995, ISBN 9780070340374. http://books.google.de/books?id=v2JEAQAAIAAJ. (Cited on page 109.)

[156] National Institute of Standards and Technology: *JAIN-SIP Reference Implementation*. http://jsip.java.net/. (Cited on page 119.)

[157] IETF: *RFC6116 - The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)*, March 2011. http://tools.ietf.org/html/rfc6116. (Cited on page 128.)

[158] IETF: *RFC3455 - Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)*, January 2003. http://tools.ietf.org/html/rfc3455. (Cited on pages 135 and 139.)

[159] IETF: *RFC3680 - A Session Initiation Protocol (SIP) Event Package for Registrations*, March 2004. http://tools.ietf.org/html/rfc3680. (Cited on page 135.)

[160] IETF: *RFC3327 - Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts*, December 2002. http://tools.ietf.org/html/rfc3327. (Cited on page 139.)

[161] IETF: *RFC4301 - Security Architecture for the Internet Protocol*, December 2005. http://tools.ietf.org/html/rfc4301. (Cited on pages 145 and 152.)

[162] IETF: *RFC1321 - The MD5 Message-Digest Algorithm*, April 1992. http://tools.ietf.org/html/rfc1321. (Cited on pages 145 and 148.)

[163] IETF: *RFC2617 - HTTP Authentication: Basic and Digest Access Authentication*, June 1999. http://tools.ietf.org/html/rfc2617. (Cited on pages 145 and 148.)

[164] IETF: *RFC2246 - The TLS Protocol Version 1.0*, January 1999. http://tools.ietf.org/html/rfc2246. (Cited on pages 145, 154 and 158.)

[165] IETF: *RFC5246 - The Transport Layer Security (TLS) Protocol Version 1.2*, August 2008. http://tools.ietf.org/html/rfc5246. (Cited on pages 145, 154 and 158.)

[166] ETSI/TISPAN: *ES 282.004 - NGN Functional Architecture; Network Attachment Subsystem (NASS)*. http://www.etsi.org/deliver/etsi_es/282000_282099/282001/03.04.01_60/es_282001v030401p.pdf. (Cited on page 151.)

[167] IETF: *RFC2403 - The Use of HMAC-MD5-96 within ESP and AH*, November 1998. http://tools.ietf.org/html/rfc2403. (Cited on pages 153 and 156.)

[168] IETF: *RFC2404 - The Use of HMAC-SHA-96 within ESP and AH*, November 1998. http://tools.ietf.org/html/rfc2404. (Cited on pages 153 and 156.)

[169] IETF: *RFC2451 - The ESP CBC-Mode Cipher Algorithms*, November 1998. http://tools.ietf.org/html/rfc2451. (Cited on pages 153 and 156.)

[170] IETF: *RFC3329 - Security Mechanism Agreement for the Session Initiation Protocol (SIP)*, January 2003. http://tools.ietf.org/html/rfc3329. (Cited on pages 153 and 154.)

[171] 3GPP: *TS 33.210 - 3G security; Network Domain Security; IP network layer security*. http://www.3gpp.org/ftp/Specs/html-info/33210.htm. (Cited on page 155.)

[172] IETF: *RFC2401 - Security Architecture for the Internet Protocol*, November 1998. http://tools.ietf.org/html/rfc2401. (Cited on page 156.)

[173] IETF: *RFC2407 - The Internet IP Security Domain of Interpretation for ISAKMP*, November 1998. http://tools.ietf.org/html/rfc2407. (Cited on page 156.)

[174] IETF: *RFC2408 - Internet Security Association and Key Management Protocol (ISAKMP)*, November 1998. http://tools.ietf.org/html/rfc2408. (Cited on page 156.)

[175] IETF: *RFC2409 - The Internet Key Exchange (IKE)*, November 1998. http://tools.ietf.org/html/rfc2409. (Cited on page 156.)

[176] IETF: *RFC3325 - Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks*, January 2003. http://tools.ietf.org/html/rfc3325. (Cited on page 156.)

[177] 3GPP: *TS 33.238 - IP Multimedia Subsystem (IMS) media plane security*. http://www.3gpp.org/ftp/Specs/html-info/33328.htm. (Cited on page 158.)

[178] IETF: *RFC3550 - RTP: A Transport Protocol for Real-Time Applications*, July 2003. http://tools.ietf.org/html/rfc3550. (Cited on page 158.)

[179] IETF: *RFC4568 - Session Description Protocol (SDP) Security Descriptions for Media Streams*, July 2006. http://tools.ietf.org/html/rfc4568. (Cited on page 158.)

[180] IETF: *RFC6043 - MIKEY-TICKET: Ticket-Based Modes of Key Distribution in Multimedia Internet KEYing (MIKEY)*, March 2011. http://tools.ietf.org/html/rfc6043. (Cited on page 158.)

[181] IETF: *RFC3711 - The Secure Real-time Transport Protocol (SRTP)*, March 2004. http://tools.ietf.org/html/rfc3711. (Cited on page 158.)

[182] IETF: *RFC4975 - The Message Session Relay Protocol (MSRP)*, September 2007. http://tools.ietf.org/html/rfc4975. (Cited on page 158.)

[183] IETF: *RFC4028 - Session Timers in the Session Initiation Protocol (SIP)*, April 2005. http://tools.ietf.org/html/rfc4028. (Cited on page 159.)

[184] 3GPP: *TS 29.228 - IP Multimedia (IM) Subsystem Cx and Dx interfaces; Signaling flows and message contents*. http://www.3gpp.org/ftp/Specs/html-info/29228.htm. (Cited on page 161.)

[185] Pintea, Bogdan: *Diameter AAA Protocol Library. SIP Express Router Integration*. Diploma-thesis, "Politehnica" University of Bucharest, February 2005. (Cited on pages 176 and 181.)

[186] Weik, Peter, Dragos Vingarzan, and Thomas Magedanz: *IP Multimedia Subsystem (IMS) Handbook: The FOKUS Open IMS Core - A Global Reference Implementation*. CRC Press, November 2009, ISBN 9781420064599. http://www.crcpress.com/product/isbn/9781420064599. (Cited on pages 176 and 219.)

[187] Charton, Andre: *Design and Implementation of a Home Subscriber Server for the 3GPP IP Multimedia Subsystem*. Diploma-thesis, Technische Universitaet Berlin, 2005. (Cited on page 176.)

[188] Fajardo, Victor I. and the OpenDiameter project community: *The OpenDiameter C++ Library*. http://diameter.sourceforge.net/. (Cited on page 177.)

[189] Vingarzan, Dragos and the Open IMS Core project community: *The Java Diameter Peer*. http://www.openimscore.org/project/jdp. (Cited on pages 177 and 182.)

[190] Fraunhofer FOKUS: *Fraunhofer Institute for Open Communication Systems - FOKUS*. http://www.fokus.fraunhofer.de. (Cited on page 177.)

[191] Fraunhofer FOKUS: *Fraunhofer FOKUS - Next Generation Network Infrastructures Competence Center*. http://www.fokus.fraunhofer.de/en/ngni/. (Cited on page 177.)

[192] Intel Corporation: *IMS Performance Benchmark - Building Out the Promise of Next-Generation Service Solutions*, February 2008. http://download.intel.com/embedded/applications/ipservices-wireless/IMSPerformanceBenchmark.pdf. (Cited on pages 178 and 199.)

[193] Intel Corporation: *White Paper - Understanding the New Performance Benchmark for IP Multimedia Subsystem (IMS) Architecture to Enable Next-Generation Networking (NGN)s*, December 2008. http://download.intel.com/embedded/applications/ipservices-wireless/IMSPerformanceBenchmark.pdf. (Cited on pages 178 and 199.)

[194] Fraunhofer FOKUS: *2nd International FOKUS IMS Workshop 2006*, November 2006. http://www.fokus.fraunhofer.de/en/fokus_events/ngni/ims_ws_06/index.html. (Cited on pages 178 and 186.)

[195] Fraunhofer FOKUS: *Open Source IMS @ FOKUS*, November 2006. http://www.fokus.fraunhofer.de/en/fokus_events/ngni/ims_ws_06/more_about/osims/index.html. (Cited on page 178.)

[196] Fraunhofer FOKUS: *The Open Source IMS Core Project - Fraunhofer Home Subscriber Server*. http://www.openimscore.org/project/FHoSS. (Cited on page 182.)

[197] 3GPP: *3GPP Release 7*. http://www.3gpp.org/specifications/Releases/article/release-7. (Cited on page 182.)

[198] Vingarzan, Dragos, Bogdan Harjoc, and Thomas Magedanz: *White Paper - Generating Realistic NGN Load with SIPNuke on Inexpensive Hardware - The "One Million" Demonstartion* , November 2008. http://sipnuke.berlios.de/sites/default/files/SIPNuke_One_Million_Whitepaper.pdf. (Cited on pages 184 and 194.)

[199] SIPp: *IMS Bench SIPp*, 2007. http://sipp.sourceforge.net/ims_bench/. (Cited on pages 184, 194, 199 and 271.)

[200] Stallman, Richard: *http://www.gnu.org/software/hurd/hurd-and-linux.html*. http://www.gnu.org/software/hurd/hurd-and-linux.html. (Cited on page 185.)

[201] Fraunhofer FOKUS: *Open Source IMS @ FOKUS*, November 2006. http://www.fokus.fraunhofer.de/en/fokus_events/ngni/ims_ws_06/more_about/osims/index.html. (Cited on page 186.)

[202] Fraunhofer FOKUS: *The Open EPC Project*. http://www.openepc.net. (Cited on pages 188 and 200.)

[203] Leuf, B. and W. Cunningham: *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley, 2001, ISBN 9780201714999. http://books.google.de/books?id=JmvbAAAAMAAJ. (Cited on page 188.)

[204] Fraunhofer FOKUS: *SIPNuke - the flexible NGN load generator*, 2008. http://www.sipnuke.orghttp://sipnuke.berlios.de/. (Cited on pages 194 and 199.)

[205] Kamailio SIP Server Project: *Kamailio - the Open Source SIP Server (also known as OpenSER in the past)*, 2013. http://www.kamailio.org/. (Cited on pages 195 and 196.)

[206] SIP Router Project: *SIP-Router Project - the common development framework for projects related to SIP Express Router (aka SER)*, 2013. http://www.sip-router.org/. (Cited on page 196.)

[207] Interoperability Lab (IOL) at University of New Hampshire: *First IMS Forum Plugfest Successfully Complete – IMS Interoperability Event for Services and Applications Gains Widespread Industry Support and Momentum.* https://www.iol.unh.edu/pressroom/IMSForumPlugfest2007.pdf. (Cited on page 198.)

[208] Din, George: *An IMS Performance Benchmark Implementation based on the TTCN-3 Language.* International Journal on Software Tools for Technology Transfer, 10(4):359–370, 2008, ISSN 1433-2779. http://dx.doi.org/10.1007/s10009-008-0078-x. (Cited on page 199.)

[209] Magedanz, Thomas, Simon Dutkowski, B. Freese, and H. Stein: *Multi-Access Modular-Services Framework – Supporting SMEs with an innovative Service Creation toolkit based on integrated SDP/IMS infrastructure.* In *11th International Conference on Intelligence in Service Delivery Networks (ICIN 2007)*, 2007. (Cited on page 202.)

[210] BMBF: *The MAMS Platform.* http://www.mams-platform.net/. (Cited on pages 202 and 203.)

[211] Fraunhofer FOKUS: *The MAMS Project (Multi-Access Modular-Services Framework)*, 2007. http://www.fokus.fraunhofer.de/en/ngni/projects/archive/archive_2007/mams/index.html. (Cited on pages 202 and 203.)

[212] Blum, Niklas, Thomas Magedanz, Horst Stein, and Ingo Wolf: *An open carrier controlled service environment for user generated mobile multimedia services.* In *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '09, pages 137–145, New York, NY, USA, 2009. ACM, ISBN 978-1-60558-659-5. http://doi.acm.org/10.1145/1821748.1821779. (Cited on page 203.)

[213] Fraunhofer FOKUS: *The MAMSplus Project (Multi-Access Modular-Services Framework / Flexible NGN Service Creation for SMEs)*, 2009. http://www.fokus.fraunhofer.de/en/ngni/projects/archive/archive_2009/mams_plus/index.html. (Cited on pages 203 and 204.)

[214] Dobes, Zuzana Krifka, Daniel von Heynitz, Andreas Rederer, Amparo Sanmateu, and Roland Schwaiger: *IMS Web Service Controls for Extending Service Functionality in Traditional Online Services.* ICIN Proceedings, 2007. (Cited on page 204.)

[215] Fraunhofer FOKUS: *The CoSIMS Project (Community-enabling Services on IMS)*, 2008. http://www.fokus.fraunhofer.de/en/ngni/projects/archive/archive_2008/cosims/index.html. (Cited on page 204.)

[216] Witaszek, Dorotha, Fabricio Caralho de Gouveia, Sebastian Wahle, and Thomas Magedanz: *IMS Playground in Pan-European Network of Testbeds;*

Benefits and Challenges. In *Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007. TridentCom 2007. 3rd International Conference on*, pages 1–6, 2007. (Cited on page 204.)

[217] Fraunhofer FOKUS: *The Panlab Project (Pan-European Laboratory)*, 2008. http://www.fokus.fraunhofer.de/en/ngni/projects/archive/archive_2008/panlab/index.html. (Cited on page 204.)

[218] Mullins, Robert, Shane Dempsey, and Tom Pfeifer: *An Architecture for IMS Services Using Open Source Infrastructure*. In Pfeifer, Tom and Paolo Bellavista (editors): *Wired-Wireless Multimedia Networks and Services Management*, volume 5842 of *Lecture Notes in Computer Science*, pages 176–182. Springer Berlin Heidelberg, 2009, ISBN 978-3-642-04993-4. http://dx.doi.org/10.1007/978-3-642-04994-1_15. (Cited on page 204.)

[219] Fraunhofer FOKUS: *The IMS ARCS Project (IMS Advanced Cluster of Services)*, 2009. http://www.fokus.fraunhofer.de/en/ngni/projects/archive/archive_2009/ims_arcs/index.html. (Cited on pages 204 and 205.)

[220] Rebahi, Yacine, Andreea Ancuta Onofrei, Fernando Lopez Agilar, Jose Manuel Lopez, Anders Fredriksson, Nelson Blanco, Luis Teixeira, Miguel Silva, and Miguel Campos: *A framework for daily emergency services support and disaster management in next generation networks (ngns)*. ICT Mobile Summit, 2009. (Cited on page 205.)

[221] Fraunhofer FOKUS: *The PEACE Project (IP-Based Emergency Applications and ServiCes for NExt Generation Networks)*, 2010. http://www.fokus.fraunhofer.de/en/ngni/projects/archive/archive_2010/peace/index.html. (Cited on page 205.)

[222] Bundesministerium des Innern: *Your telephone number for government agencies and offices: 115*. http://www.115.de/cln_330/nn_1614438/EN/Home/home__node.html?__nnn=true. (Cited on page 205.)

[223] Fraunhofer FOKUS: *The Service Line 115 Project*, 2010. http://www.fokus.fraunhofer.de/en/ngni/projects/archive/archive_2010/service_line_115/index.html. (Cited on page 205.)

[224] Fraunhofer FOKUS: *The Mobile Cloud Networking Project*, 2013. http://www.fokus.fraunhofer.de/en/ngni/projects/current_projects/mobilecloud_networking/index.html. (Cited on pages 205 and 206.)

[225] Magedanz, Thomas, Dorotha Witaszek, and Karsten Knuettel: *The IMS playground @ FOKUS-an open testbed for generation network multimedia services*. In *Testbeds and Research Infrastructures for the Development of Networks*

and Communities, 2005. Tridentcom 2005. First International Conference on, pages 2–11, 2005. (Cited on page 206.)

[226] Fraunhofer FOKUS: *Open IMS Playground*, 2004. http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_ims_playground/index.html. (Cited on page 206.)

[227] University of Patras - ECE - NAMG: *The Patras Open Source IMS testbed*. http://nam.ece.upatras.gr/ppe/?q=node/2. (Cited on page 207.)

[228] GSMA: *Rich Communication Services*. http://www.gsma.com/rcs/. (Cited on pages 207 and 222.)

[229] Hallwachs, Rainer: *Evaluation of the Fraunhofer Open Source IMS Core platform with special focus on the Call Session Control Function (CSCF)*. GRIN Verlag, 2009, ISBN 9783640420469. http://books.google.de/books?id=yVMIuYFIH1oC. (Cited on page 208.)

[230] University of Cape Town, South Africa: *The UCT IMS Client*. http://uctimsclient.berlios.de/. (Cited on page 210.)

[231] Waiting, David, Richard Good, Richard Spiers, and Neco Ventura: *The UCT IMS client*. In *Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, 2009. TridentCom 2009. 5th International Conference on*, pages 1–6, 2009. (Cited on page 210.)

[232] University of Cape Town, South Africa: *The UCT IPtv Client*. https://developer.berlios.de/project/showfiles.php?group_id=7844. (Cited on page 210.)

[233] Waiting, David, Richard Good, Richard Spiers, and Neco Ventura: *Open source development tools for IMS research*. In *Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities*, TridentCom '08, pages 41:1–41:10, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ISBN 978-963-9799-24-0. http://dl.acm.org/citation.cfm?id=1390576.1390627. (Cited on page 210.)

[234] Various ICST and IEEE Conferences: *The Open NGN and IMS Testbeds Workshop*. http://www.onit-ws.org. (Cited on page 210.)

[235] ICST: *TridentCom 2009 - The 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, April 2009. http://tridentcom.org/tridentcom09/. (Cited on page 211.)

[236] ICST TridentCom 2009 Workshops: *Open NGN and IMS Testbeds Workshop @ TridentCom 2009 - Infrastructure as a Service – A Paradigm for Open NGN and IMS Testbeds?* http://www.onit-ws.org/2009/. (Cited on page 211.)

[237] ICST: *TridentCom 2010 - The 6th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, May 2010. http://www.tridentcom.org/2010/. (Cited on page 211.)

[238] ICST TridentCom 2010 Workshops: *2nd Open NGN and IMS Testbeds Workshop @ TridentCom 2010 - Driving efficient R&D through Open NGN/NGMN and IMS Testbeds*, May 2010. http://www.onit-ws.org/2010/. (Cited on page 211.)

[239] IEEE: *Computer Software and Applications Conference COMPSAC 2011 - The Computed World: Software Beyond the Digital Society*, July 2011. http://compsac-2012.cs.iastate.edu/?y=3. (Cited on page 211.)

[240] IEEE COMPSAC 2011 Workshops: *3rd International IEEE Workshop on Open NGN and IMS Testbeds (ONIT 2011) @ COMPSAC 2011 - Next Generation Network Evolution Towards the Future Internet*, July 2011. http://www.onit-ws.org/2011/. (Cited on page 211.)

[241] IEEE: *Global Communications Conference, Exhibition and Industry Forum GLOBECOM 2012 - The Magic of Global Connectivity*, December 2012. http://www.ieee-globecom.org/2012/. (Cited on page 211.)

[242] Vingarzan, Dragos, Peter Weik, and Thomas Magedanz: *Design and Implementation of an Open IMS Core*. In Magedanz, Thomas, Ahmed Karmouch, Samuel Pierre, and Iakovos Venieris (editors): *Mobility Aware Technologies and Applications*, volume 3744 of *Lecture Notes in Computer Science*, pages 284–293. Springer, Berlin / Heidelberg, October 2005, ISBN 978-3-540-29410-8. http://dx.doi.org/10.1007/11569510_27. (Cited on page 218.)

[243] Vingarzan, Dragos, Peter Weik, and Thomas Magedanz: *Development of an open source IMS core for emerging IMS testbeds, the academia and beyond*. Journal of Mobile Multimedia, 3(2):131–149, June 2007, ISSN 1550-4646. http://dl.acm.org/citation.cfm?id=2010540.2010544. (Cited on page 218.)

[244] Vingarzan, Dragos and Peter Weik: *End-to-end performance of the IP multimedia subsystem over various wireless networks*. In *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, volume 1, pages 183–188, April 2006. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=1683461. (Cited on page 219.)

[245] Vingarzan, Dragos and Peter Weik: *IMS Signaling over Current Wireless Networks: Experiments Using the Open IMS Core*. Vehicular Technology Magazine, IEEE, 2(1):28–34, March 2007, ISSN 1556-6072. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4286917. (Cited on page 219.)

[246] Albaladejo, Alberto Diez, Dragos Vingarzan, Peter Weik, and Thomas Magedanz: *Digital inclusion opportunities in the telecommunications sector through NGN and open source tools: The Open IMS Core experience*. In *Innovations for Digital Inclusions, 2009. K-IDI 2009. ITU-T Kaleidoscope:*, pages 1–6, September 2009, ISBN 978-92-61-12891-3. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5338929. (Cited on page 219.)

[247] Zhou, Qing, Konstantinos Pentikousis, Cornel Pampu, Marius Corici, Dragos Vingarzan, and Thomas Magedanz: *WO/2012/113155 - METHOD FOR MANAGING A DATA PATH, MOBILITY CONTROLLER AND COMMUNICATION SYSTEM*, August 2012. http://www.freepatentsonline.com/WO2012113155.html, http://www.freepatentsonline.com/WO2012113155.html. (Cited on page 219.)

[248] Zhou, Qing, Konstantinos Pentikousis, Cornel Pampu, Marius Corici, Dragos Vingarzan, and Thomas Magedanz: *WO/2012/113156 - METHOD FOR ESTABLISHING A DATA PATH, DEVICE AND COMMUNICATION SYSTEM*, August 2012. http://www.freepatentsonline.com/WO2012113156.html, http://www.freepatentsonline.com/WO2012113156.html. (Cited on page 219.)

[249] GSMA: *joyn - Branding name for the GSMA Rich Communication Services*. http://www.joynus.com/. (Cited on page 222.)

[250] ITU-T: *The World in 2011, ICT Facts and Figures*. Technical report, ITU Telecom World '11, 2011. http://www.itu.int/ITU-D/ict/facts/2011/material/ICTFactsFigures2011.pdf.

[251] Gartner, Inc.: *Gartner Says Worldwide Mobile Phone Sales Grew 35 Percent in Third Quarter 2010; Smartphone Sales Increased 96 Percent*. http://www.gartner.com/it/page.jsp?id=1466313.

List of Acronyms

2G	2nd Generation Wireless Telephone Technology (GSM)
3DES	ESP Triple DES
3G	3rd Generation Mobile Telecommunications (UMTS, CDMA2000)
3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
4G	4th Generation Mobile Telecommunications (LTE, WiMAX)
AAA	Authentication, Authorization and Accounting
ABC	Always Best Connected
ACL	Access Control List
ACM	Association for Computing Machinery
AES	Advanced Encryption Standard (a.k.a. Rijndael)
AIN	Advanced Intelligent Network
AKA	Authentication and Key Agreement
ALG	Application Level Gateway
AMF	Authentication Management Field
AN	Access Network
ANDSF	Access Network Discovery and Selection Function
ANGw	Access Network Gateway
ANSI	American National Standards Institute
AoR	Address of Record
API	Application Programming Interface
ARPANET	Advanced Research Projects Agency Network
ARPU	Average Revenue Per User
AS	Application Server
ATM	Asynchronous Transfer Mode
AuC	Authentication Center
AVP	Attribute-Value Pair
B2BUA	Back-to-Back User Agent
BGCF	Breakout Gateway Control Function
BSD	Berkley Software Distribution
BSF	Bootstrapping Server Function
BSS	Business Support System
CableLabs	Cable Television Laboratories, Inc.
CAMEL	Customized Applications for Mobile Networks Enhanced Logic
CAP	CAMEL Application Part
CAPEX	Capital Expenditures
CATV	Community Access Television / Cable TV
CBC	Cypher-block Chaining
CCITT	Comité Consultatif International Téléphonique et Télégraphique
CCS	Common-Channel Signaling

CCS6	Common-Channel Signaling System # 6
CD	Compact Disc
CDMA	Code Division Multiple Access
CDMA2000	CDMA 2000 - first 3G technology deployed
CGI	Common Gateway Interface
CK	Cypher Key
CLF	Connectivity Session and Repository Location Function
CN	Core Network
CNF	Conjunctive Normal Form
COMPSAC	Computer Software and Applications Conference
CORBA	Common Object Request Broker Architecture
CPE	Customer Premises Equipment
CPU	Central Processing Unit
CS	Circuit Switched
CS-1	IN Capabilities Set 1
CS-2	IN Capabilities Set 2
CS-3	IN Capabilities Set 3
CS-4	IN Capabilities Set 4
CSCF	Call Session Control Function
CSFB	CS Fallback
DAAD	Deutscher Akademischer Austausch Dienst / German Academic Exchange Service
DARPA	Defense Advanced Research Projects Agency
DBMS	Database Management System
DES	Data Encryption Standard
DNF	Disjunctive Normal Form
DNS	Domain Name System
DOC	Design Objective Capacity
DoS	Denial of Service
DSAI	Dynamic Service Activation
DSL	Digital Subscriber Line
E-CSCF	Emergency CSCF
EAP	Extensible Authentication Protocol
EDE3	ESP Triple DES
enum	Telephone Number Mapping Working Group
ENUM	E.164 (Telephone) Number Mapping
EPC	Evolved Packet Core
ESP	Encapsulating Security Payload
ETSI	European Telecommunications Standards Institute
EU	European Union
FHoSS	Fraunhofer Home Subscriber Server
FI	Future Internet
FMC	Fixed Mobile Convergence
FOKUS	Fraunhofer Institute for Open Communication Systems

FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
FUSECO	FUture SEamless COmmunication
GIBA	GPRS-IMS Bundled Authentication
GLOBECOM	Global Communications Conference
GNU	GNU's Not Unix!
GPL	GNU General Public License
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GSM-R	GSM for Railways
GSMA	GSM Association
GUI	Graphical User Interface
HDLC	High-Level Data Link Control
HMAC	keyed-Hashing MAC
HLR	Home Location Register
HSS	Home Subscriber Server
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
I-CSCF	Interrogating CSCF
iptel	IP Telephony Working Group
IANA	Internet Assigned Numbers Authority
IBCF	Interconnection Border Control Function
ICMP	Internet Control Message Protocol
ICST	Institute for Computer Sciences, Social Informatics and Telecommunications
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
iFC	Initial Filter Criteria
IHS	Inadequately Handled Scenarios
IK	Integrity Key
IKE	Internet Key Exchange
IM	Instant Messaging
IMP	Interface Message Processor
IMPI	IMS Private User Identity
IMPU	IMS Public User Identity
IMS	IP-Multimedia Subsystem
IMSU	IMS Subscriber Identity
IN	Intelligent Networks
IP	Intelligent Peripheral
INAP	Intelligent Network Application Part
INT	IMS Network Testing
IOT	Interoperability Testing
IP	Internet Protocol
IPC	Inter Process Communication

IPR	Intellectual Property Rights
IPsec	IP Security
IPTV	IP Television
IPv4	IP version 4
IPv6	IP version 6
IRP	Integration Reference Point
ISC	IMS Service Control
ISDN	Integrated Services Digital Network
ISIM	IMS SIM
ISP	Internet Service Provider
ISUP	ISDN User Part
IT	Information Technology
ITU	International Telecommunication Union
ITU-T	ITU - Telecommunication Standardization Sector
IVR	Interactive Voice Response
JAIN	Java APIs for IN
JMF	Java Media Framework
JNI	Java Native Interface
JSR	Java Specification Requests
JVM	Java Virtual Machine
Kbps	Kilo Bits per Second (1,000 bits / second)
KISS	Keep it Simple Stupid
KMS	Key Management Service
KPI	Key Performance Indicator
LAN	Local Area Network
LIA	Location-Info-Answer
LIR	Location-Info-Request
LTE	Long Term Evolution
MAA	Multimedia-Authentication-Answer
MAC	Message Authentication Code
MAP	Mobile Application Part
MAR	Multimedia-Authentication-Request
MB	Mega Bytes (1,048,576 bytes)
Mbps	Mega Bits per Second (1,000,000 bits / second)
MD5	Message-Digest algorithm 5
MGCF	Media Gateway Control Function
MGCP	Media Gateway Control Protocol
MGW	Media Gateway
MIPS	Microprocessor without Interlocked Pipeline Stages
MITM	Man-in-the-Middle
MMS	Multimedia Messaging Service
MMTel	Multimedia Telephony Service
MMtel	Multimedia Telephony
mmusic	Multiparty Multimedia Session Control Working Group

MNO	Mobile Network Operator
MOBIS	Mobile Integrated Services
MPLS	Multiprotocol Label Switching
MRFC	Media Processing Function Controller
MSc	Master of Science
MSRP	Message Session Relay Protocol
MTC	Machine Type Communication
MTP	Message Transfer Part
MTU	Maximum Transmission Unit
NAS	Network Access Server
NASS	Network Access Sub-system
NAT	Network Address Translation
NCP	Network Control Program
NDA	Non-Disclosure Agreement
NDS	Network Domain Security
NE	Network Entity
NFV	Network Function Virtualization
NGN	Next Generation Network
NGNI	Next Generation Network Infrastructures
NIST	National Institute of Standards and Technology
NMS	Network Management System
NNI	Network-to-Network Interface
O&M	Operations and Management
ONIT	Open NGN and IMS Test-beds
OOP	Object Oriented Programming
OpenEPC	Open Evolved Packet Core
OpenIMSCore	Open Source IMS Core
OpenSER	Open SIP Express Router
OPEX	Operating Expenditure
OSA	Open Service Access
OSI	Open System Interconnection
OSS	Operations Support System
OTA	Over-The-Air
OTT	Over-The-Top
P-CSCF	Proxy CSCF
p2psip	Peer-to-Peer SIP Working Group
PABX	Private Automatic Branch Exchange
PBX	Private Branch Exchange
PCC	Policy and Charging Control
PCRF	Policy and Charging Rules Function
PDP	Packet Data Protocol
PowerPC	Performance Optimization With Enhanced RISC - Performance Computing
PPA	Push-Profile-Answer

PPP	Point-to-Point Protocol
PPR	Push-Profile-Request
PS	Packet Switched
PSI	Public Service Identity
PSTN	Public Switched Telephone Network
PX	Protocol Implementation eXtra Information for Testing
QoE	Quality of Experience
QoS	Quality of Service
R&D	Research and Development
RAD	Rapid Application Development
RADIUS	Remote Authentication Dial In User Service
RAN	Radio Access Network
RAT	Radio Access Technology
RCS	Rich Communication Suite
RCS-e	RCS enhanced
RCS	Rich Communication Suite
RDBMS	Relational DBMS
RI	Reference Implementation
RISC	Reduced Instruction Set Computing
RTA	Registration-Termination-Answer
RTP	Real-time Transport Protocol
RTR	Registration-Termination-Request
RTT	Round-Trip-Time
S-CSCF	Serving CSCF
SAA	Server-Assignment-Answer
SAE	System Architecture Evolution
SAPS	Scenario Attempts per Second
SAR	Server-Assignment-Request
SBC	Session Border Controller
SCE	Service Creation Environment
SCTP	Stream Control Transmission Protocol
SCP	Service Control Point
SCCP	Signaling Connection Control Part
SDES	Session Description Protocol Security Descriptions
SDF	Service Data Function
SDLC	Synchronous Data Link Control
SDN	Software-Defined Networking
SDP	Service Delivery Platform
SDP	Session Description Protocol
SDPng	Session Description and Capability Negotiation
SDR	Software Defined Radio
SEG	Security Gateway
SEP	Signaling End Point
SER	SIP Express Router

SHA	Secure Hash Algorithm
SIB	Service Independent Building Block
SIG	Special Interest Group
sigtran	Signaling Transport Working Group
simple	SIP for Instant Messaging and Presence Leveraging Extensions Working Group
sip	SIP Working Group
sipping	Session Initiation Proposal Investigation Working Group
SIGTRAN	Signaling Transport
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SLAAC	Stateless Address Auto Configuration
SLEE	Service Logic Execution Environment
SME	Small and Medium Enterprise
SMTP	Simple Mail Transfer Protocol
SMS	Service Management System
SMS	Short Messaging Service
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SON	Self-Organized Network
SPARC	Scalable Processor Architecture
SPI	Security Parameter Index
SPT	Service Point Trigger
SQL	Structured Query Language
SRTP	Secure RTP
SRVCC	Single Radio Voice Call Continuity
SS7	Signaling System # 7
SSP	Service Switching Point
STP	Signaling Transfer Point
SuT	System under Test
tel	telephone
TACACS	Terminal Access Controller Access Control System
TCAP	Transaction Capabilities Application Part
TCO	Total Cost of Ownership
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol, the Internet Protocol Suite
TEM	Telecommunication Equipment Manufacturer
TFTP	Trivial File Transfer Protocol
THIG	Topology Hiding Internetwork Gateway
TISPAN	Telecommunications and Internet converged Services and Protocols for Advanced Networks
TLS	Transport Layer Security

TPC	Technical Program Committee
TRIDENTCOM	Testbeds and Research Infrastructures for the Development of Networks and Communities Conference
TRT	Total Round-trip Time
TS	Technical Specification
TS	Test System
TSP	Telecommunication Service Provider
UA	User Agent
UAA	User-Authorization-Answer
UAC	User Agent Client
UAR	User-Authorization-Request
UAS	User Agent Server
UCT	University of Cape Town
UDP	User Datagram Protocol
UE	User Equipment/Endpoint
UMTS	Universal Mobile Telecommunications System
UNI	User-to-Network Interface
UNIX	Uniplexed Information and Computing System
UP	User-Plane
URI	Uniform Resource Identifier
URN	Uniform Resource Name
USIM	UMTS SIM
VoIP	Voice over IP
VoLGA	VoLTE via Generic Access
VoLTE	Voice over LTE
WiFi	WLAN
WiMAX	Worldwide Interoperability for Microwave Access
WG	Working Group
WLAN	Wireless LAN
WPA	WiFi Protected Access
WWW	World Wide Web
x86	Intel® 80x86 Instruction Set Architecture
XCAP	XML Configuration Access Protocol
XML	Extensible Markup Language

Appendix A

Author's Publications, Contributions to Standards and Presentations

The present appendix comes as an additional support for Section 8.2 – Publications, Contributions to Standards and Presentations. While an exhaustive listing of all the published scientific papers, journal articles, conference presentations, standards or patents was restricted there by the purposes of the respective summary chapter, this appendix lists all of the author's work on NGN core network architectures and technologies, with comprehensive bibliographical referencing.

Also included are a number of MSc and Diploma Theses which have been realized as part of the design and implementation of the OpenIMSCore project itself or as additional tooling for test-beds. The project was used extensively at Technische Univesität Berlin and many other universities around the globe (some of which supported still by the author), such that many more such items could be enumerated, yet only those directly supervised by the author have been listed here.

While the list of publications is presented here in a consistent style with the bibliography section, with numbered references, in order to avoid confusions, all citations and references in the dissertation link exclusively to the bibliography section starting on page 225. Hence none of the listed elements in the present appendix are cited directly, but when required they have been also included in the bibliography section.

A.1 Publications

[1] Vingarzan, Dragos: *A 3rdGeneration IP Multimedia Session Handling implementation into SIP Express Router*. Diploma-thesis, "Politehnica" University of Bucharest, February 2005.

[2] Vingarzan, Dragos, Peter Weik, and Thomas Magedanz: *Design and Implementation of an Open IMS Core*. In Magedanz, Thomas, Ahmed Karmouch, Samuel Pierre, and Iakovos Venieris (editors): *Mobility Aware Technologies and Applications*, volume 3744 of *Lecture Notes in Computer Science*, pages 284–293. Springer, Berlin / Heidelberg, October 2005, ISBN 978-3-540-29410-8. http://dx.doi.org/10.1007/11569510_27.

[3] Vingarzan, Dragos and Peter Weik: *End-to-end performance of the IP multimedia subsystem over various wireless networks*. In *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, volume 1, pages 183–188, April 2006. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=1683461.

[4] Vingarzan, Dragos, Peter Weik, and Thomas Magedanz: *Towards an Open Source IMS Core System Enabling Rapid prototyping of NGN Services*. In *3rd International WORKSHOP on 'Next Generation Networking Middleware'(NGNM06) in the scope of Networking 2006 5th IFIP, TC6 Networking Conference*. University of Coimbra,Coimbra, Portugal, May 2006.

[5] Vingarzan, Dragos and Peter Weik: *IMS Signaling over Current Wireless Networks: Experiments Using the Open IMS Core*. Vehicular Technology Magazine, IEEE, 2(1):28–34, March 2007, ISSN 1556-6072. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4286917.

[6] Din, George, Dragos Vingarzan, and Peter Weik: *Leistungsvergleich von IMS*. funkschau, 05(5):44–46, May 2007, ISSN 0016-2841.

[7] Magedanz, Thomas, Peter Weik, Dragos Vingarzan, Fabricio Carvalho de Gouveia, and Sebastian Wahle: *Experiences on the Establishment and Provisioning of NGN/IMS Testbeds - The FOKUS Open IMS Playground and the Related Open Source IMS Core*. In *11th International Conference on Intelligence in Service Delivery Network 2007 (ICIN 2007)*. Bordeaux, France, October 2007. http://www.icin.biz/files/programmes/Session2B-2.pdf.

[8] Vingarzan, Dragos, Peter Weik, and Thomas Magedanz: *Development of an open source IMS core for emerging IMS testbeds, the academia and beyond*. Journal of Mobile Multimedia, 3(2):131–149, June 2007, ISSN 1550-4646. http://dl.acm.org/citation.cfm?id=2010540.2010544.

[9] Blum, Niklas, Piotr Jacak, Florian Schreiner, Dragos Vingarzan, and Peter Weik: *Towards Standardized and Automated Fault Management and*

Service Provisioning for NGNs. Journal of Network and Systems Management, 16:63–91, 2008, ISSN 1064-7570. http://dx.doi.org/10.1007/s10922-007-9094-5.

[10] Harjoc, Bogdan, Florian Schreiner, Dragos Vingarzan, and Thomas Magedanz: *Automated fault localization based on unified Web service and NGN benchmarking*. In *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, pages 77–82, July 2009, ISBN 978-1-4244-4672-8. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5202413.

[11] Albaladejo, Alberto Diez, Dragos Vingarzan, Peter Weik, and Thomas Magedanz: *Digital inclusion opportunities in the telecommunications sector through NGN and open source tools: The Open IMS Core experience*. In *Innovations for Digital Inclusions, 2009. K-IDI 2009. ITU-T Kaleidoscope:*, pages 1–6, September 2009, ISBN 978-92-61-12891-3. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5338929.

[12] Corici, Marius, Alberto Diez, Dragos Vingarzan, Thomas Magedanz, Cornel Pampu, and Qing Zhou: *Enhanced access network discovery and selection in 3GPP Evolved Packet Core*. In *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, pages 677–682, October 2009, ISBN 978-1-4244-4488-5. http://ieeexplore.ieee.org/xpls/abs_all.jsp?tp=&arnumber=5355058.

[13] Weik, Peter, Dragos Vingarzan, and Thomas Magedanz: *IP Multimedia Subsystem (IMS) Handbook: The FOKUS Open IMS Core - A Global Reference Implementation*. CRC Press, November 2009, ISBN 9781420064599. http://www.crcpress.com/product/isbn/9781420064599. (Cited on pages 176 and 219.)

[14] Thanh, Tran Quang, Dragos Vingarzan, Yacine Rebahi, and Thomas Magedanz: *A Diameter based testing system in Next Generation Mobile Network*. In *Telecommunications Network Strategy and Planning Symposium (NETWORKS), 2010 14th International*, pages 1–5, September 2010, ISBN 978-1-4244-6704-4. http://ieeexplore.ieee.org/xpls/abs_all.jsp?tp=&arnumber=5624958.

[15] Corici, Marius, Thomas Magedanz, Dragos Vingarzan, and Peter Weik: *Prototyping mobile broadband applications with the open Evolved Packet Core*. In *Intelligence in Next Generation Networks (ICIN), 2010 14th International Conference on*, pages 1–5, December 2010, ISBN 978-1-4244-7443-1 (PRINT) 978-1-4244-7444-8 (ELECTRONIC). http://ieeexplore.ieee.org/xpls/abs_all.jsp?tp=&arnumber=5640899.

[16] Corici, Marius, Thomas Magedanz, Dragos Vingarzan, and Peter Weik: *Enabling Ambient Aware Service Delivery in Heteroge-*

neous Wireless Environments. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6, December 2010, ISBN 978-1-4244-5636-9 (PRINT) 978-1-4244-5637-6 (ELECTRONIC). http://ieeexplore.ieee.org/xpls/abs_all.jsp?tp=&arnumber=5683952.

[17] Corici, Marius, Dragos Vingarzan, Thomas Magedanz, Cornel Pampu, and Qing Zhou: *Access Network Reselection Based on Momentary Resources in a Converged Wireless Environment.* In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6, December 2010, ISBN 978-1-4244-5636-9 (PRINT) 978-1-4244-5637-6 (ELECTRONIC). http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5684363.

[18] Corici, Marius, Jens Fiedler, Ancuta Onofrei, and Dragos Vingarzan: *Enabling dynamic service delivery in the 3GPP Evolved Packet Core.* In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pages 2012–2016, December 2010. http://ieeexplore.ieee.org/xpls/abs_all.jsp?tp=&arnumber=5700297.

[19] Corici, Marius, Dragos Vingarzan, and Thomas Magedanz: *3GPP Evolved Packet Core - the Mass Wireless Broadband all-IP architecture.* Telecommunications: The Infrastructure for the 21st Century (WTC), 2010, pages 1 –6, September 2010. http://ieeexplore.ieee.org/xpls/abs_all.jsp?tp=&arnumber=5755093.

[20] Corici, Marius, Jens Fiedler, Thomas Magedanz, and Dragos Vingarzan: *Access Network Discovery and Selection in the Future Broadband Wireless Environment.* In Cai, Ying, Thomas Magedanz, Minglu Li, Jinchun Xia, Carlo Giannelli, Ozgur Akan, Paolo Bellavista, Jiannong Cao, Falko Dressler, Domenico Ferrari, Mario Gerla, Hisashi Kobayashi, Sergio Palazzo, Sartaj Sahni, Xuemin (Sherman) Shen, Mircea Stan, Jia Xiaohua, Albert Zomaya, and Geoffrey Coulson (editors): *Mobile Wireless Middleware, Operating Systems, and Applications*, volume 48 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 70–83. Springer Berlin Heidelberg, June-July 2010, ISBN 978-3-642-17757-6 (PRINT) 978-3-642-17758-3 (ONLINE). http://dx.doi.org/10.1007/978-3-642-17758-3_6.

[21] Lange, Lajos, Thomas Magedanz, Julius Müller, Daniel Nehls, and Dragos Vingarzan: *Evolutionary future internet service platforms enabling seamless cross layer interoperability.* In *Internet Communications (BCFIC Riga), 2011 Baltic Congress on Future*, pages 1–6, February 2011, ISBN 978-1-4244-8511-6. http://ieeexplore.ieee.org/xpls/abs_all.jsp?tp=&arnumber=5733228.

[22] Corici, Marius, Ancuta Onofrei, Dragos Vingarzan, Julius Müller, and Thomas Magedanz: *Massive deployment of small coverage area cells in the all-IP wireless system - Opportunities, issues and solutions.* In *Telecom World (ITU WT), 2011 Technical Symposium at ITU*, pages 147–152, October 2011,

ISBN 978-1-4577-1148-0 (PRINT) 978-92-61-13681-9 (ELECTRONIC). http://ieeexplore.ieee.org/xpls/abs_all.jsp?tp=&arnumber=6100946.

[23] Müller, Julius, Thomas Magedanz, Marius Corici, and Dragos Vingarzan: *UE & network initiated QoS reservation in NGN and beyond.* In *Network of the Future (NOF), 2011 International Conference on the*, pages 62–67, November 2011, ISBN 978-1-4577-1605-8. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6126684.

[24] Corici, Marius, Jens Fiedler, Dragos Vingarzan, and Thomas Magedanz: *Device connectivity management for mobile All-IP broadband network architecture.* In *Signal Processing and Communication Systems (ICSPCS), 2011 5th International Conference on*, pages 1–7, December 2011, ISBN 978-1-4577-1179-4 (PRINT) 978-1-4577-1178-7 (ELECTRONIC). http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6140857.

[25] Corici, Marius, Jens Fiedler, Thomas Magedanz, and Dragos Vingarzan: *Evolution of the resource reservation mechanisms for machine type communication over mobile broadband Evolved Packet Core architecture.* In *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, pages 718–722, December 2011, ISBN 978-1-4673-0039-1 (PRINT) 978-1-4673-0038-4 (ELECTRONIC). http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6162547.

[26] Corici, Marius, Jens Fiedler, Dragos Vingarzan, and Thomas Magedanz: *Optimized low mobility support in massive mobile broadband Evolved Packet Core architecture.* In *Networks (ICON), 2011 17th IEEE International Conference on*, pages 322–327, December 2011, ISBN 978-1-4577-1824-3. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6168496.

[27] Corici, Marius, Fabricio Gouveia, Thomas Magedanz, and Dragos Vingarzan: *OpenEPC: A Technical Infrastructure for Early Prototyping of NGMN Testbeds.* In Magedanz, Thomas, Anastasius Gavras, Nguyen Huu Thanh, Jeffry S. Chase, Ozgur Akan, Paolo Bellavista, Jiannong Cao, Falko Dressler, Domenico Ferrari, Mario Gerla, Hisashi Kobayashi, Sergio Palazzo, Sartaj Sahni, Xuemin (Sherman) Shen, Mircea Stan, Jia Xiaohua, Albert Zomaya, and Geoffrey Coulson (editors): *Testbeds and Research Infrastructures. Development of Networks and Communities*, volume 46 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 166–175. Springer Berlin Heidelberg, May 2011, ISBN 978-3-642-17851-1. http://dx.doi.org/10.1007/978-3-642-17851-1_13.

[28] Corici, Marius, Dragos Vingarzan, Thomas Magedanz, Cornel Pampu, and Qing Zhou: *Proactive Vertical Handover Optimizations in the 3GPP Evolved*

Packet Core. In Pentikousis, Kostas, Ramón Agüero, Marta García-Arranz, Symeon Papavassiliou, Ozgur Akan, Paolo Bellavista, Jiannong Cao, Falko Dressler, Domenico Ferrari, Mario Gerla, Hisashi Kobayashi, Sergio Palazzo, Sartaj Sahni, Xuemin (Sherman) Shen, Mircea Stan, Jia Xiaohua, Albert Zomaya, and Geoffrey Coulson (editors): *Mobile Networks and Management*, volume 68 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 109–122. Springer Berlin Heidelberg, September 2011, ISBN 978-3-642-21443-1 (PRINT) 978-3-642-21444-8 (ONLINE). http://dx.doi.org/10.1007/978-3-642-21444-8_11.

[29] Corici, Marius, Jens Fiedler, Thomas Magedanz, and Dragos Vingarzan: *Access Network Discovery and Selection in the Future Wireless Communication*. Mobile Networks and Applications, 16:337–349, June 2011, ISSN 1383-469X (PRINT) 1572-8153 (ONLINE). http://dx.doi.org/10.1007/s11036-011-0309-3.

[30] Corici, Marius, Dragos Vingarzan, Thomas Magedanz, Peter Weik, Nico Beyer, and Hans Einsiedler: *Connectivity Support Harmonization in Future Internet Architecture*. In *Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on*, pages 93–99. Berlin, Germany, October 2012. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6376041.

[31] Müller, Julius, Yahya Al-Hazmi, Mohammad Fal Sadikin, Dragos Vingarzan, and Thomas Magedanz: *Secure and Efficient Validation of Data Traffic Flows in Fixed and Mobile Networks*. In *Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, PM2HW2N '12, pages 159–166, New York, NY, USA, October 2012. ACM, ISBN 978-1-4503-1626-2. http://dl.acm.org/citation.cfm?id=2387213.

[32] Corici, Marius, Mihai Constantin, Dana Satriya, Dragos Vingarzan, Valentin Vlad, and Lukas Wöllner: *Integrating off-the-shelf 3GPP access networks in the OpenEPC software toolkit: Realizing cost-efficient and complete small-scale operator testbeds*. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, pages 1724–1729, December 2012. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6477845.

[33] Müller, Julius, Andreas Wierz, Dragos Vingarzan, and Thomas Magedanz: *Elastic Network Design and Adaptive Flow Placement in Software Defined Networks*. In Computer Communications, International Conference on and ContextQoS Workshop Networks ICCCN 2013 (editors): *To-be-published*. Nassau, Bahamas, IEEE, July-August 2013.

[34] Corici, Marius, Dragos Vingarzan, Valentin Vlad, and Thomas Magedanz: *Self-adaptable IP Connectivity Control in Carrier Grade Mobile Operator Networks*.

In Borcea, Cristian, Paolo Bellavista, Carlo Gianelli, Thomas Magedanz, and Florian Schreiner (editors): *Mobile Wireless Middleware, Operating Systems, and Applications*, volume 65 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 150–163. Springer Berlin Heidelberg, January 2013, ISBN 978-3-642-36659-8. http://dx.doi.org/10.1007/978-3-642-36660-4_11.

[35] Katanekwa, Nicholas, Neco Ventura, Marius Corici, Dragos Vingarzan, and Thomas Magedanz: *Enhanced gateway selection for optimal routing in a distributed evolved packet core (epc) network.* In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*, pages 1–6. Krabi, Thailand, ECTI/IEEE, May 2013. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6559549.

[36] Corici, Marius, Julius Mueller, Dragos Vingarzan, and Thomas Magedanz: *Part I: History and perspectives on the telecom standardized assets - Network and Control Platforms.* In *Evolution of Telecommunication Services: The Convergence of Telecom and Internet: Technologies and Ecosystems*, page 339. Bertin, E. and Crespi, N. and Magedanz, T., 2013, ISBN 978-3642415685. http://www.springer.com/computer/communication+networks/book/978-3-642-41568-5.

A.2 Education & Diploma Theses

[1] Dinu, Florin: *Security Enhancements for IMS: Topology Hiding and HSS Based Diameter Messaging.* Diploma-thesis, "Politehnica" University of Bucharest, 2006.

[2] Popescu, Adrian Daniel: *Presence Service in the IMS.* Diploma-thesis, "Politehnica" University of Bucharest, 2006.

[3] Harjoc, Bogdan: *Benchmarking Tool for the IP Multimedia Subsystem.* Diploma-thesis, "Politehnica" University of Bucharest, 2007.

[4] Weiss, Paul: *Design and Implementation of a Performance Benchmarking System for IMS Core Components.* Diploma-thesis, Technische Universität Berlin, 2007.

[5] Emin, Umut: *Design and Implementation of MRF-Controller and MRF-Processor within an IMS Architecture.* Msc-thesis, Technische Universität Berlin, 2008.

[6] Ilie, Alexandru: *Deployment and Configuration Enablers for Next Generation Networks.* Diploma-thesis, "Politehnica" University of Bucharest, 2008.

[7] Satriya, Dwianto Dana: *Design and Implementation of a Flat SGSN Functionality for 3GPP Release 11 Evolved Packet System.* Msc-thesis, Rheinisch-Westfälische Technische Hochschule Aachen, 2012.

[8] Kocur, Jakub: *Design and Implementation of an eNodeB emulator for the 3GPP Release 11 Evolved Packet System.* Diploma-thesis, Politechnika Warszawska, 2013.

[9] Vlad, Valentin: *SDN Impacts on Packet Core Evolution.* Msc-thesis, Technische Universität Berlin, 2013.

A.3 Patents

[1] Cornel Pampu, Qing Zhou, Marius Julian Corici, Alberto Diez, Thomas Magedanz, and Dragos Vingarzan. WO/2011/120218 - METHOD FOR RE-SELECTING A COMMUNICATION NETWORK, October 2011. http://www.freepatentsonline.com/WO2011120218.html.

[2] Cornel Pampu, Qing Zhou, Corici Marius, Alberto Diez, Thomas Magedanz, and Dragos Vingarzan. USPA20130064221 - SYSTEM AND METHOD FOR MANAGING AN ACCESS NETWORK RE-SELECTION, March 2013. http://www.freepatentsonline.com/y2013/0064221.html.

[3] Qing Zhou, Konstantinos Pentikousis, Cornel Pampu, Marius Corici, Dragos Vingarzan, and Thomas Magedanz. EP2507955 - METHOD FOR ESTABLISHING A DATA PATH, DEVICE AND COMMUNICATION SYSTEM, October 2012. http://www.freepatentsonline.com/EP2507955A1.html.

[4] Qing Zhou, Konstantinos Pentikousis, Cornel Pampu, Marius Corici, Dragos Vingarzan, and Thomas Magedanz. WO/2012/113154 - METHOD AND DEVICE FOR ADAPTING A DATA PATH FOR A MOBILE ENTITY IN A COMMUNICATION NETWORK, August 2012. http://www.freepatentsonline.com/WO2012113154.html.

[5] Qing Zhou, Konstantinos Pentikousis, Cornel Pampu, Marius Corici, Dragos Vingarzan, and Thomas Magedanz. WO/2012/113155 - METHOD FOR MANAGING A DATA PATH, MOBILITY CONTROLLER AND COMMUNICATION SYSTEM, August 2012. http://www.freepatentsonline.com/WO2012113155.html.

[6] Qing Zhou, Konstantinos Pentikousis, Cornel Pampu, Marius Corici, Dragos Vingarzan, and Thomas Magedanz. WO/2012/113156 - METHOD FOR ESTABLISHING A DATA PATH, DEVICE AND COMMUNICATION SYSTEM, August 2012. http://www.freepatentsonline.com/WO2012113156.html.

A.4 Contributions to Standards

[1] Vingarzan, Dragos *et al.*: *IMS/NGN Performance Benchmark Part 1: Core Concepts*, 2007. http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=24161, ETSI/TISPAN 6 Workitem 06024-1, TS 186.008-1.

[2] Vingarzan, Dragos *et al.*: *IMS/NGN Performance Benchmark Part 2: Subsystem Configurations and Benchmarks*, 2007. http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=25501, ETSI/TISPAN 6 Workitem 06024-2, TS 186.008-2.

[3] Vingarzan, Dragos *et al.*: *IMS/NGN Performance Benchmark Part 3: Traffic Sets and Traffic Profiles*, 2007. http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=25502, ETSI/TISPAN 6 Workitem 06024-3, TS 186.008-3.

[4] Vingarzan, Dragos: *IMS Network Testing (INT); IMS & EPC Interoperability test descriptions*, 2011. http://www.etsi.org/deliver/etsi_ts/103000_103099/103029/05.01.01_60/ts_103029v050101p.pdf, ETSI/INT Workitem INT/DTS00050, TS 103.029.

A.5 Conference Presentations

[1] Vingarzan, Dragos and Peter Weik: *Introducing the Open IMS Core - Technical Tutorial*. In *2nd FOKUS International IMS Workshop*. 16th-17th of November, Berlin, Germany, 2006. http://www.fokus.fraunhofer.de/en/fokus_events/ngni/ims_ws_06/index.html.

[2] Vingarzan, Dragos and Thomas Magedanz: *Towards Open Softswitching - Open Source IMS Core for Multi Access Multi Service Environments*. In *IQPC Softswitching*. 24th-26th of October, Berlin, Germany, 2006. https://kb.iqpc.co.uk/events/3116/topic/24.

[3] Benchmark, ETSI/TISPAN 6 IMS/NGN Performance: *An Intel Description and Prototype Results*. In *The 3GSM World Congress*. 12th-15th of February, Barcelona, Spain, 2007.

[4] Weik, Peter and Dragos Vingarzan: *The Open IMS Core - an Open Source Reference Implementation for the Telecommunications Industry*. In *Design& Developers Forum at the 50th IEEE Global Communications Conference , Exhibition & Industry Forum (GLOBECOM)*. 26th-30th of November, Washington D.C., USA, 2007.

[5] Vingarzan, Dragos, Sebastian Wahle, Peter Weik, and Thomas Magedanz: *Interoperability: How Does This Actually Work For IMS? - Experiences From The Open IMS Playground*. In *IIR Telecoms IMS Global Congress 2007*. 27th of November - 1st of December, Amsterdam, Netherlands, 2007.

[6] Vingarzan, Dragos and Thomas Magedanz: *Testing for IMS - Turning IMS into Reality*. In *Marcus Evans 2nd Annual IMS Implementation Strategies 2007*. 29th-31st of January, Amsterdam, Netherlands, 2007. https://kb.iqpc.co.uk/events/3116/topic/24.

[7] Vingarzan, Dragos and Peter Weik: *The Open IMS Core - Technical Tutorial*. In *3rd FOKUS International IMS Workshop*. 15th-16th of November, Berlin, Germany, 2007. http://www.fokus.fraunhofer.de/en/fokus_events/ngni/ims_ws_07/index.html.

[8] Weik, Peter and Dragos Vingarzan: *The Open IMS Core - A Reference Implementation for NGN Prototyping and Testing Tutorial*. In *4th FOKUS International IMS Workshop*. 6th-7th of November, Berlin, Germany, 2008. http://www.fokus.fraunhofer.de/en/fokus_events/ngni/ims_ws_08/index.html.

[9] Magedanz, Thomas, Alberto Diez, Marius Corici, and Dragos Vingarzan: *Understanding Next Generation Mobile Network and Related Technologies - LTE, EPC and IMS*. In *5th FOKUS International IMS Workshop*. 11th of November, Berlin, Germany, 2009. http://www.fokus.fraunhofer.de/en/fokus_events/ngni/ims_ws_09/index.html.

[10] Vingarzan, Dragos, Hassnaa Moustafa, and Eugen Mickoczy: *Chairing the 2nd Open NGN and IMS Testbeds Workshop (ONIT).* In *TRIDENTCOM*. 18th of May, Berlin, Germany, 2010. http://www.onit-ws.org/2010/index.html.

[11] Taggesspiegel: *Fraunhofer Institut - Ein Spielplatz für Forscher*, August 2010. http://www.tagesspiegel.de/wirtschaft/fraunhofer-institut-ein-spielplatz-fuer-forscher/1908292.html.

[12] Corici, Marius, Thomas Magedanz, and Dragos Vingarzan: *Tutorial:Towards the Wireless Future Internet - Understanding the role of future mobile broadband networks and the Evolved Packet Core.* In *International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, TRIDENTCOM*. 17th of May, Berlin, Germany, 2010. http://www.tridentcom.org/2010/. (Cited on page 211.)

[13] Corici, Marius, Dragos Vingarzan, and Thomas Magedanz: *3GPP Evolved Packet Core - the Mass Wireless Broadband all-IP architecture.* In *World Telecommunications Congress*. 13th-14th of September, Vienna, Austria, 2010. http://conference.vde.com/wtc2010/Pages/Start.aspx.

[14] Vingarzan, Dragos and Marius Corici: *Getting started with the OpenEPC and OpenIMSCore to set-up your own FUSECO Testbed.* In *2nd FOKUS FUSECO Forum*. 17th of November, Berlin, Germany, 2011. http://www.fokus.fraunhofer.de/en/fokus_events/ngni/fuseco_forum_2011/index.html.

[15] Vingarzan, Dragos and Marius Corici: *Getting started with OpenEPC to set-up your own LTE/FUSECO Testbed.* In *3rd FOKUS FUSECO Forum*. 15th of November, Berlin, Germany, 2012. http://www.fokus.fraunhofer.de/en/fokus_events/ngni/fuseco_forum_2012/index.html.

[16] Thomas Magedanz an, Dragos Vingarzan, Vijay K. Varma, Nguyen Huu Thanh, and Neco Ventura: *Chairing the 4nd Open NGN and IMS Testbeds Workshop (ONIT).* In *GLOBECOM*. 3-7th of December, Anaheim, California, USA, 2012. http://www.onit-ws.org/.

[17] Tafazolli, Rahim and Laurent Herault: *Strategic Research and Innovation Agenda "Internet on the Move"*. Technical report, Europe 5G Public Private Partnership Programme - Net!Works Expert Advisory Group, May 2013.

Appendix B

Relevant IMS Information

B.1 Reference Point and Interface Naming Conventions

Cx	I-CSCF/S-CSCF ⟷ HSS (Diameter)
e2	P-CSCF ⟷ CLF (Diameter)
Gm	UE ⟷ P-CSCF (SIP)
Mw	CSCF ⟷ CSCF (SIP)
Sh	AS ⟷ HSS (Diameter)
Za	SEG ⟷ SEG (IPsec)
Zb	NE ⟷ SEG/NE (IPsec)
Zh	BSF ⟷ HSS (Diameter)

B.2 IMS Signaling Routing

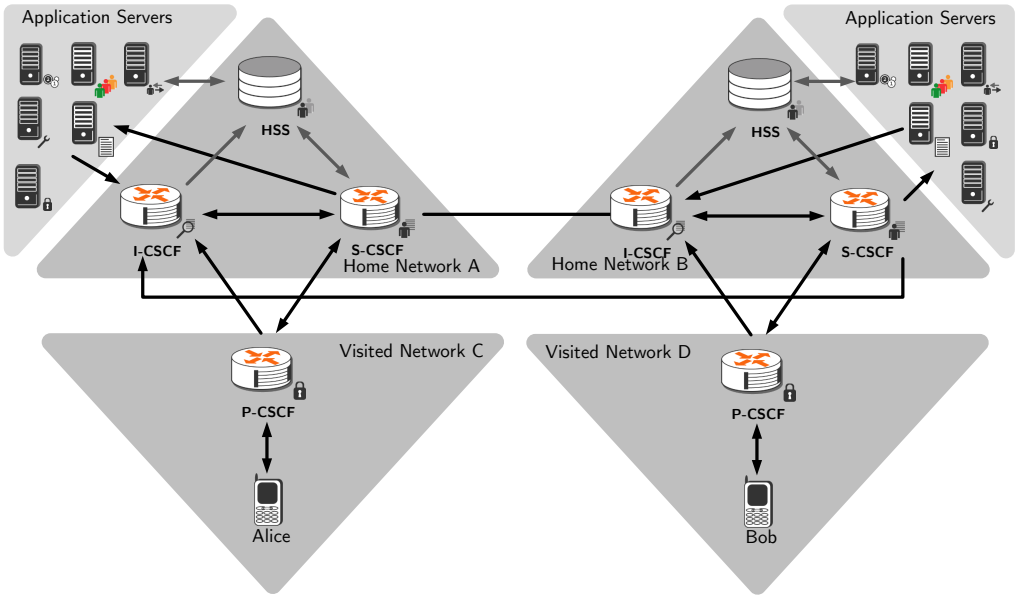


Figure B.1: IMS Signaling Routing Overview

B.3 Comprehensive IMS Functional Overview

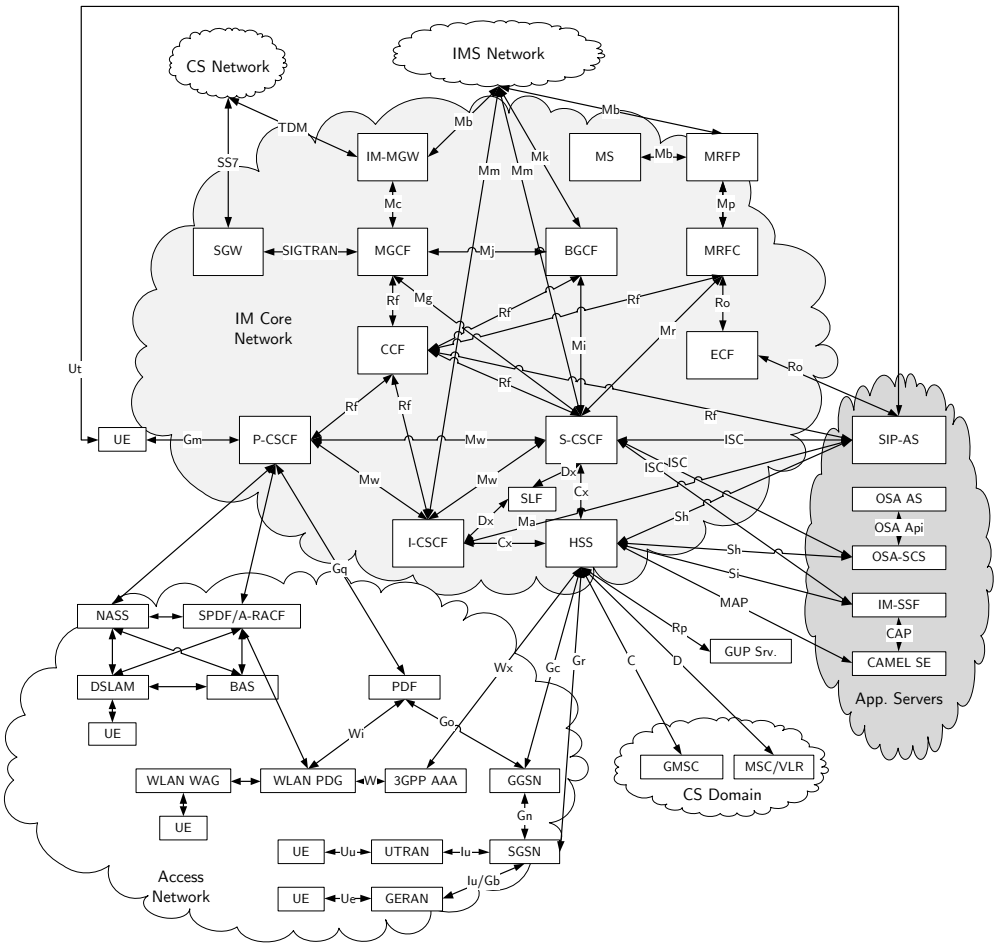


Figure B.2: IMS Functions

Appendix C

IMS Benchmark Sample on OpenIMSCore

The present appendix is a direct copy of one benchmarking run performed by the SIPp IMS-Bench tool [199] on the OpenIMSCore. The results should not be seen as representative for the performance as the testing environment was quite limited. The purpose of this appendix is to rather highlight and validate the OpenIMSCore successful use to establish and verify the IMS/NGN Performance Benchmark [12, 152, 153].

The following text is pasted as much as possible verbatim from the generated

report, with only formatting changes required due to Hypertext Markup Language (HTML) to paginated conversion.

C.1 Summary

This report shows the result of a benchmark run performed by "IMS Bench SIPp", an implementation of the **IMS/NGN Performance Benchmark** suite, **ETSI TS 186.008**.

The test was started on 13-May-2013 10:55, and the total time for the test execution was **0h 40m 17s**. The **Design Objective Capacity (DOC)** is **40** scenarios per second.

The following systems and parameters were used for the test.

Role	Server	IP	Nb Users
SuT 1	HSS	10.0.103.6	
SuT 2	S-CSCF	10.0.103.17	
SuT 3	P-CSCF	10.0.103.8	
SuT 4	I-CSCF	10.0.103.19	
Manager	ims-bench-small	10.0.103.3	
TS1	ims-bench-small	10.0.103.3	24000

Parameter Name	Parameter Value	Parameter Info
Ring Time	5000	Ringing Time (ms)
Hold Time	120000	Conversation Time (ms)
Registration Expire	1000000	Registration Timeout (ms)
Transient Time	30	Time after the start of a step for which data is ignored (in seconds)

The following table shows the average of the key measurements for each step of the test. Each steps is characterized by the requested load, the effective load, the global Inadequately Handled Scenarios (IHS) (total of all Inadequately handled scenarios for this step divided by number of Session Attempts for this step) the scenario IHS (number of inadequately handled scenarios for this step divided by the number of scenario attempts for this step), the Central Processing Unit (CPU) utilization and the available Memory on the SuT. The available Memory is expressed in MegaBytes, and the requested and effective loads in Scenario Attempts per Second (SAPS).

Note that the IHS percentages represented in this table are the number of failures for a step divided by the number of scenario attempts for this step, and so is not the average of (IHS per seconds)

	Pre-registration	Step 1
Requested load	40	50
Effective Load	39.02	49.46
Ratio ims_rereg %	0.00	15.09
Ratio ims_reg %	100.00	2.55
Ratio ims_uac %	0.00	49.73
Ratio ims_dereg %	0.00	2.56
Ratio ims_msgc %	0.00	30.07
CPU HSS	46.75	13.08
CPU S-CSCF	12.21	6.68
CPU P-CSCF	5.61	6.83
CPU I-CSCF	11.92	3.74
Memory HSS	1072.87	1063.60
Memory S-CSCF	1371.39	1165.53
Memory P-CSCF	894.77	819.46
Memory I-CSCF	1272.41	1267.74
SIPp CPU ims-bench-small	6.14	9.16
SIPp MEM ims-bench-small	1110.06	1076.83
IHS ims_rereg %	0.00	1.55
IHS ims_reg %	3.71	0.89
IHS ims_uac %	0.00	73.50
IHS ims_dereg %	0.00	1.18
IHS ims_msgc %	0.00	1.15
global IHS %	3.71	37.18

The following chapters show details on different measurement, like delay between two messages, response time or number of messages per seconds.

Each measurement can be represented in one of the four following forms.

1. Evolution in function of the time. On such graphs, the raw information is plotted, like number of messages per seconds, or response time of each scenario. This graph is useful in giving for instance a good idea on the distribution of response times, and it's evolution over the time.

2. Evolution (mean) in function of the time. While previous graph gives a good indication, it may sometimes be easier to see the evolution of the mean of the measurement over a second in function of the time.

3. Histogram. This graph shows the histogram of the measurement, so how many times each value of the measurement occured.

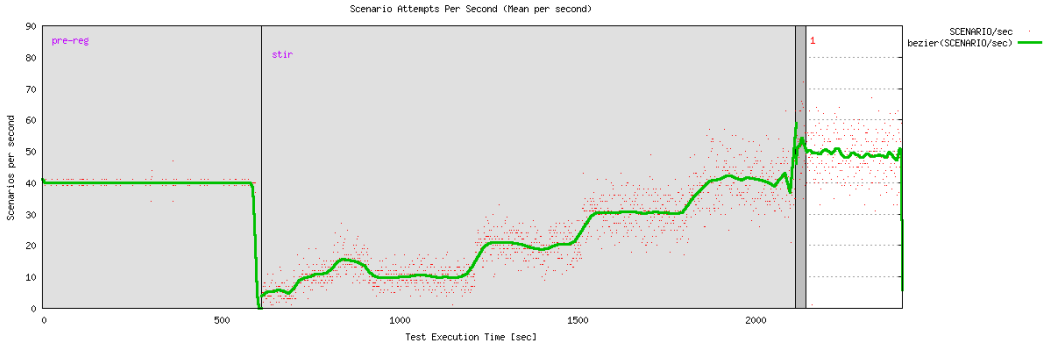
4. Probability. This graph gives the probability of the measurement to be higher than a certain value. This graph can be used to determine percentile for instances.

For some graphs, a cubic Bezier curve is plotted as well.

C.2 Scenario Attempts Per Second

This graph represents the number of scenario per seconds generated by the test system. For each step, the generation was based on a Poisson.

		Effective Load								
Step	Requested Load	Mean	Variance	Std Dev	Min	Max	% 50	% 90	% 95	% 99
Pre-reg	40	39.02	38.50	6.20	0.00	47.00	40.0	40.0	41.0	41.0
1	50	49.46	64.19	8.01	1.00	90.00	49.0	58.0	62.0	67.0



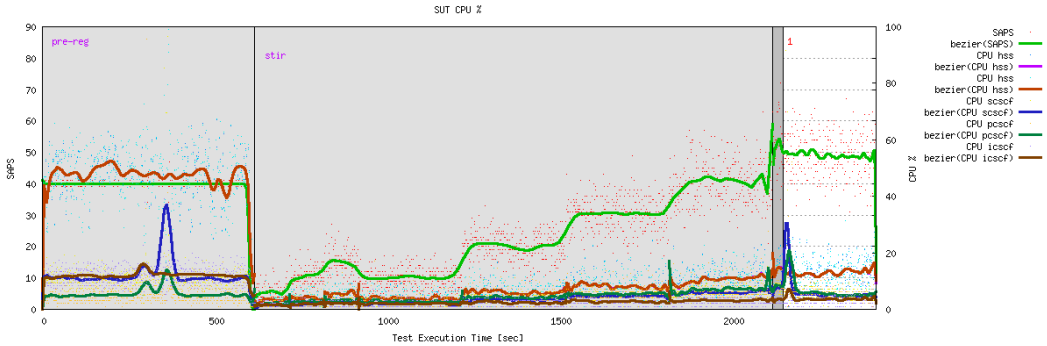
Scenario Attempts Per Second (Mean per second)

C.3 SuT CPU %

This graph represents the CPU of the system under test (SuT).

		CPU HSS				CPU S-CSCF			
Step	Requested Load	Mean	Std Dev	Min	Max	Mean	Std Dev	Min	Max
Pre-reg	40	46.75	11.61	1.04	99.01	12.21	10.98	0.00	100.00
1	50	13.08	4.76	1.04	31.31	6.68	10.33	1.06	100.00

		CPU P-CSCF				CPU I-CSCF			
Step	Requested Load	Mean	Std Dev	Min	Max	Mean	Std Dev	Min	Max
Pre-reg	40	5.61	7.24	0.00	100.00	11.92	4.97	1.03	100.00
1	50	6.83	7.76	1.05	100.00	3.74	3.24	0.00	51.35



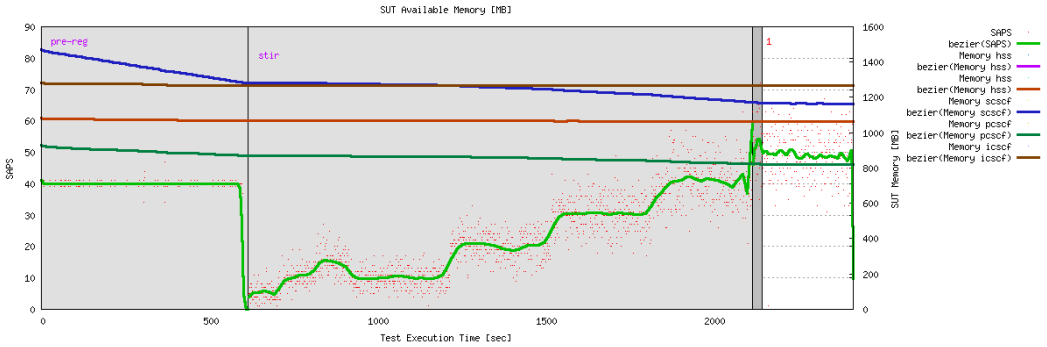
SuT CPU % over time

C.4 SuT Available Memory [MB]

This graph represents the Available memory on the system under test, in MBytes (SuT).

		Memory HSS				Memory S-CSCF			
Step	Requested Load	Mean	Std Dev	Min	Max	Mean	Std Dev	Min	Max
Pre-reg	40	1072.87	3.54	1067.32	1081.06	1371.39	53.60	1282.66	1471.01
1	50	1063.60	0.35	1062.96	1064.27	1165.53	1.67	1163.01	1173.40

		Memory P-CSCF				Memory I-CSCF			
Step	Requested Load	Mean	Std Dev	Min	Max	Mean	Std Dev	Min	Max
Pre-reg	40	894.77	14.99	870.75	930.19	1272.41	3.89	1267.63	1283.25
1	50	819.46	1.48	818.24	824.81	1267.74	0.06	1267.51	1267.88

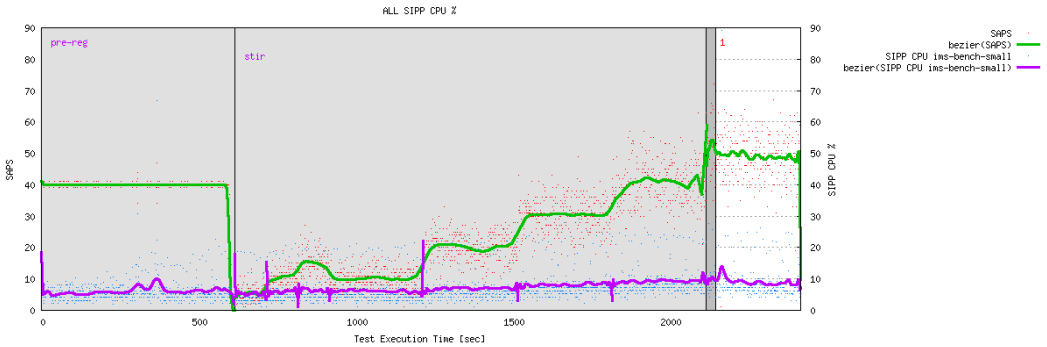


SuT Available Memory [MB] over time

C.5 All SIPp CPU %

This graph represents the CPU of SIPp on All Test Machines

		SIPp CPU ims-bench-small			
Step	Requested Load	Mean	Std Dev	Min	Max
Pre-reg	40	6.14	4.07	2.08	66.67
1	50	9.16	6.60	3.12	89.29

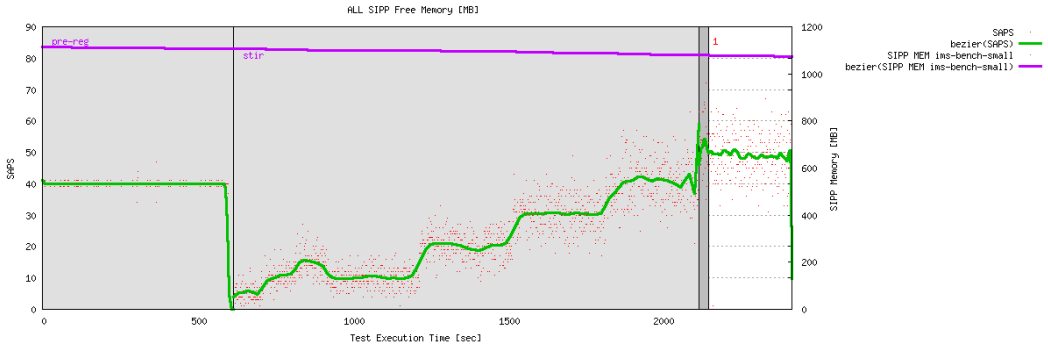


All SIPp CPU % over time

C.6 All SIPp Free Memory [MB]

This graph represents the free memory of SIPp on ALL Test Machines, in MBytes

		SIPp MEM ims-bench-small			
Step	Requested Load	Mean	Std Dev	Min	Max
Pre-reg	40	1110.06	2.02	1106.45	1113.96
1	50	1076.83	1.61	1073.95	1079.83

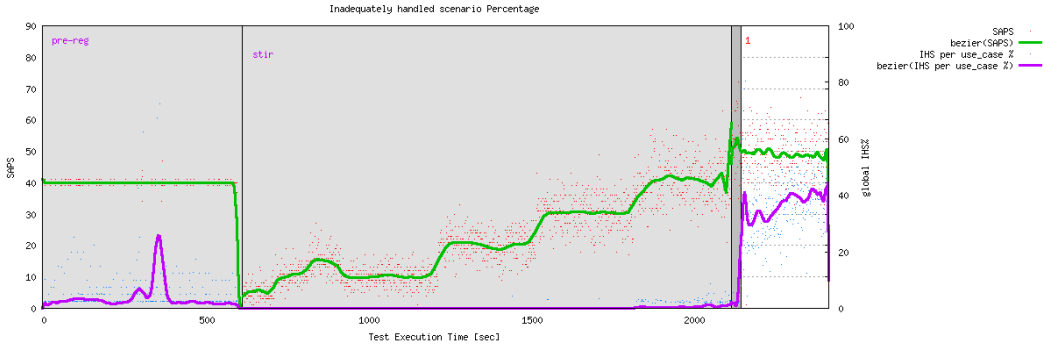


All SIPp Free Memory [MB] over time

C.7 Inadequately handled scenario Percentage

This graph represents the percentage of inadequately handled scenarios.

		%IHS per use_case			
Step	Requested Load	Mean	Std Dev	Min	Max
Pre-reg	40	3.63	11.47	0.00	100.00
1	50	33.63	14.81	0.00	100.00

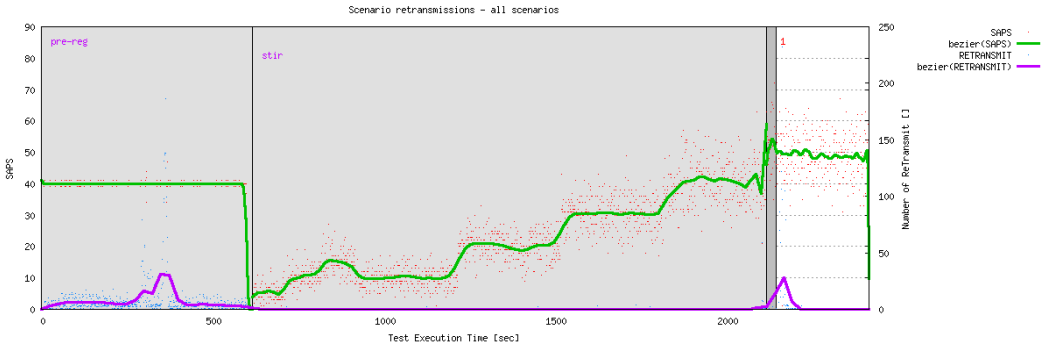


Inadequately handled scenario Percentage over time

C.8 Scenario retransmissions - all scenarios

This graph represents the number of retransmissions per seconds for all scenarios.

		RETRANSMIT							
Step	Requested Load	Mean	Std Dev	Min	Max	% 50	% 90	% 95	% 99
Pre-reg	40	7.75	17.23	0.00	186.00	4.0	12.0	22.0	99.0
1	50	4.15	21.52	0.00	232.00	0.0	1.0	11.0	122.0



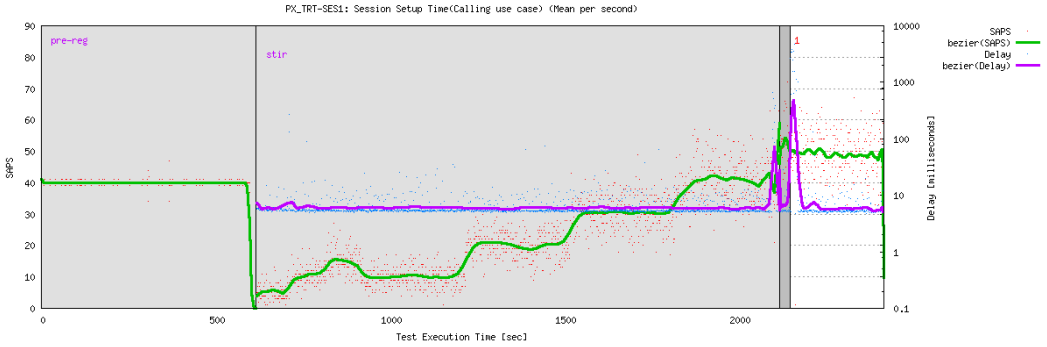
Scenario retransmissions - all scenarios over time

C.9 Calling

C.9.1 PX_TRT-SES1: Session Setup Time

This graph represents the delay between the Caller sending INVITE and callee receiving ACK.

		PX_TRT-SES1 (msec)							
Step	Requested Load	Mean	Std Dev	Min	Max	% 50	% 90	% 95	% 99
1	50	122.78	704.99	4.10	11558.10	5.2	13.4	513.8	2080.0

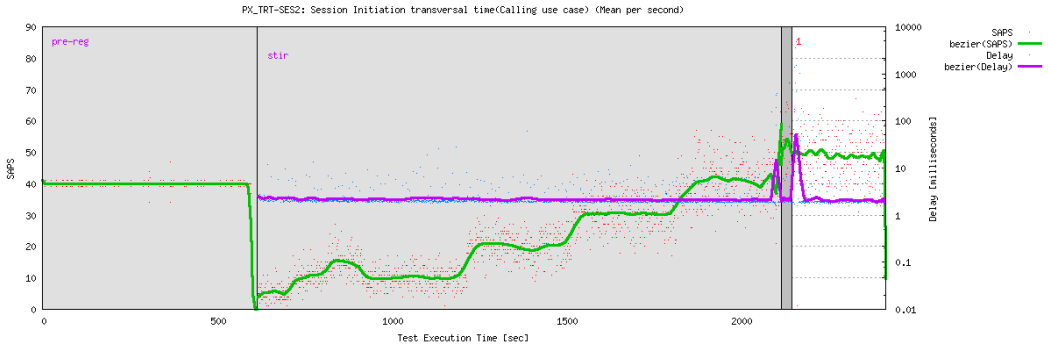


PX_TRT-SES1: Session Setup Time(Calling use case) (Mean per second)

C.9.2 PX_TRT-SES2: Session Initiation transversal time

This graph represents the delay between the caller sending INVITE and the callee receiving INVITE.

		PX_TRT-SES2 (msec)							
Step	Requested Load	Mean	Std Dev	Min	Max	% 50	% 90	% 95	% 99
1	50	38.22	326.59	1.25	8010.91	1.9	2.2	9.9	1512.0

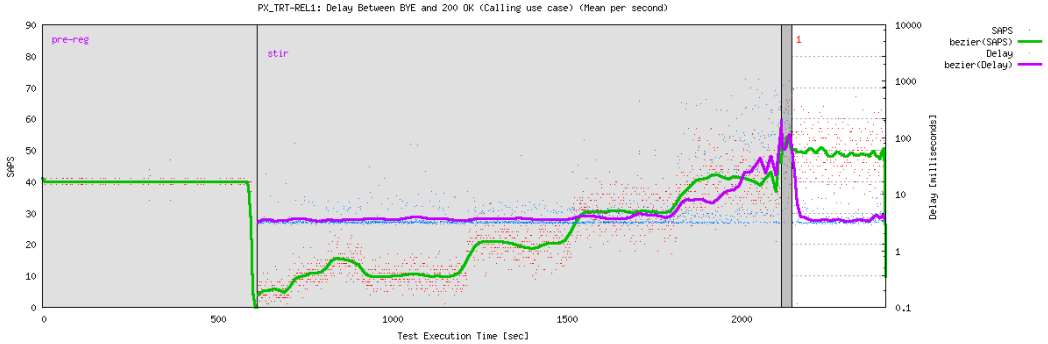


PX_TRT-SES2: Session Initiation transversal time(Calling use case) (Mean per second)

C.9.3 acsPX_TRT-REL1: Delay Between BYE and 200 OK

This graph represents the delay between the first BYE and the corresponding 200 OK.

		PX_TRT-REL1 (msec)							
Step	Requested Load	Mean	Std Dev	Min	Max	% 50	% 90	% 95	% 99
1	50	63.38	513.06	2.34	8824.14	3.1	4.0	8.7	1657.0



PX_TRT-REL1: Delay Between BYE and 200 OK (Calling use case) (Mean per second)

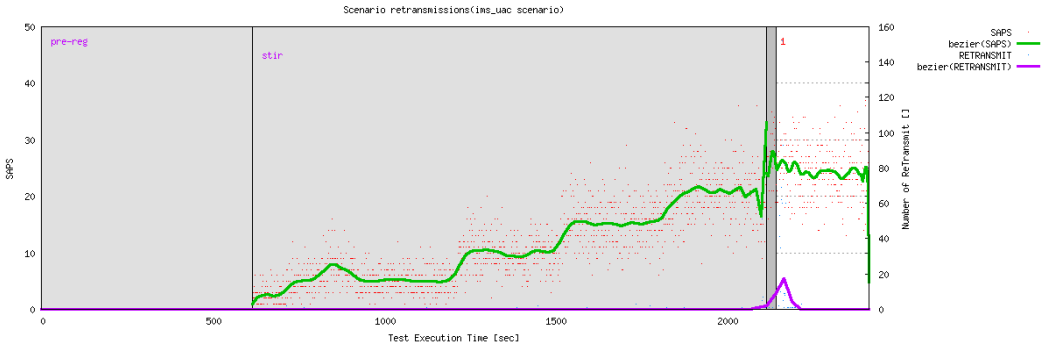
C.9.4 PX_TRT-SES3: INVITE and re-INVITE cost

This graph represents the caller sending first INVITE and callee receiving second ACK.

C.9.5 ims_uac : Scenario retransmissions

This graph represents the number of retransmissions per seconds for this scenario.

		RETRANSMIT							
Step	Requested Load	Mean	Std Dev	Min	Max	% 50	% 90	% 95	% 99
Pre-reg	40	0.00	0.00	0.00	0.00	0.0	0.0	0.0	0.0
1	50	2.59	13.98	0.00	152.00	0.0	1.0	8.0	82.0

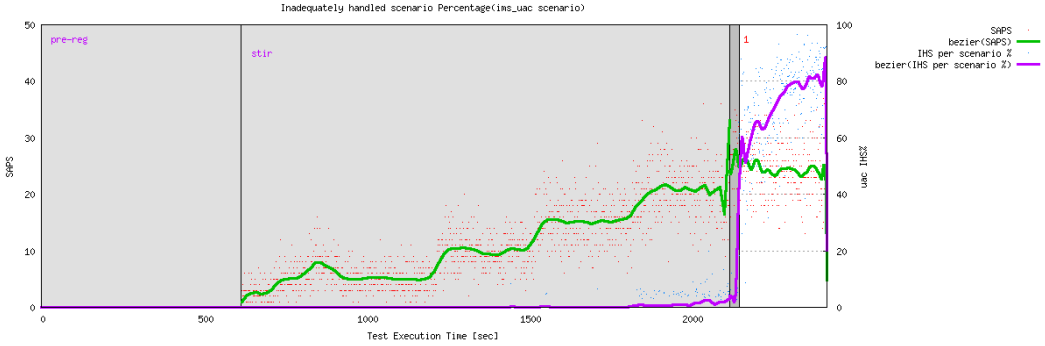


Scenario retransmissions(ims_uac scenario) over time

C.9.6 ims_uac : Inadequately handled scenario Percentage

This graph represents the percentage of Inadequately handled scenarios for the uac.

		IHS per scenario %			
Step	Requested Load	Mean	Std Dev	Min	Max
Pre-reg	40	0.00	0.00	0.00	0.00
1	50	65.99	24.38	0.00	100.00



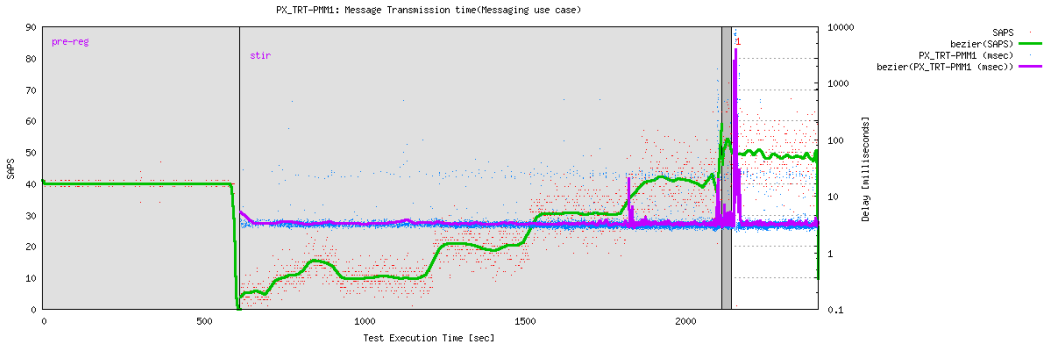
Inadequately handled scenario Percentage(ims_uac scenario) over time

C.10 Messaging

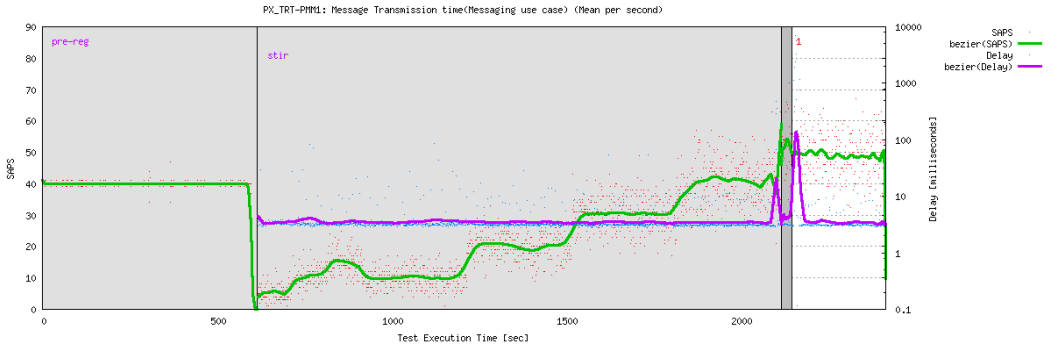
C.10.1 PX_TRT-PMM1: Message Transmission time

This graph represents the delay between the message and the 200 OK.

		PX_TRT-PMM1 (msec)							
Step	Requested Load	Mean	Std Dev	Min	Max	% 50	% 90	% 95	% 99
1	50	82.73	661.44	2.32	9130.66	3.0	3.5	22.2	1889.0



PX_TRT-PMM1: Message Transmission time(Messaging use case) over time



PX_TRT-PMM1: Message Transmission time(Messaging use case) (Mean per second)

C.10.2 PX_TRT-PMM2: Message Transmission time (error case)

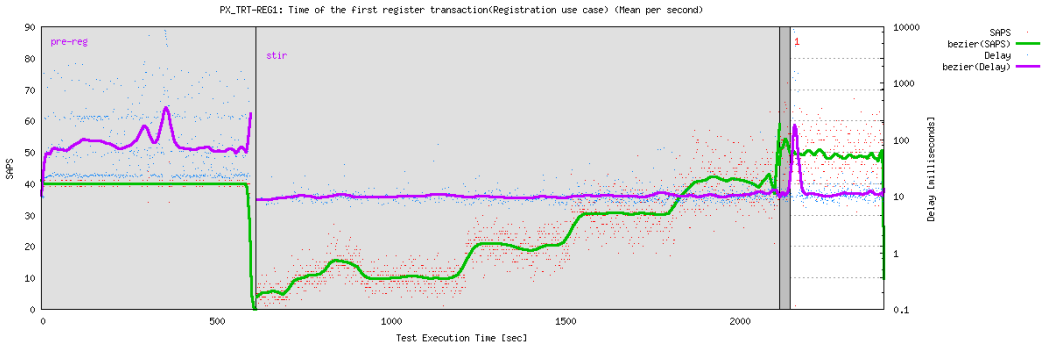
This graph represents the delay between the message and the 404 Not Found.

C.11 Registration

C.11.1 PX_TRT-REG1: Time of the first register transaction

This graph represents the time of the first register transaction in the registration use_cases i.e. the time between the REGISTER and the 401 Unautorized for all scenarios in the Registration use_case.

		PX_TRT-REG1 (msec)							
Step	Requested Load	Mean	Std Dev	Min	Max	% 50	% 90	% 95	% 99
Pre-reg	40	295.30	1641.96	6.17	27635.17	30.5	134.0	520.2	7619.0
1	50	110.04	832.47	6.23	11563.91	10.3	19.0	24.4	1639.0

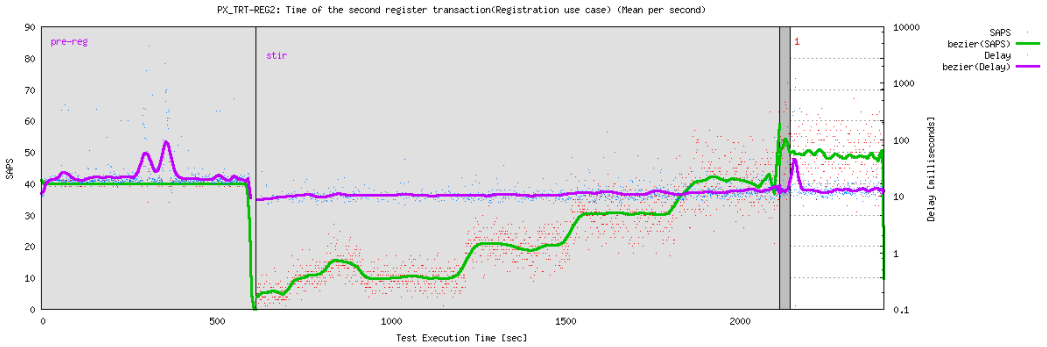


PX_TRT-REG1: Time of the first register transaction(Registration use case) (Mean per second)

C.11.2 PX_TRT-REG2: Time of the second register transaction

This graph represents the time of the second register transaction in the registration use_cases, i.e. the delay between the second REGISTER and the 200 OK.

		PX_TRT-REG2 (msec)							
Step	Requested Load	Mean	Std Dev	Min	Max	% 50	% 90	% 95	% 99
Pre-reg	40	52.65	552.47	6.92	23570.84	18.8	27.0	52.9	393.1
1	50	18.99	66.06	6.95	1206.93	11.9	18.7	25.2	86.7

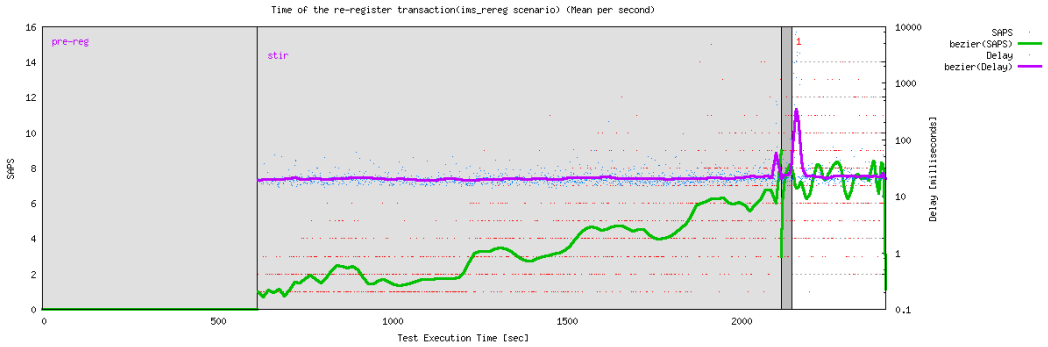


PX_TRT-REG2: Time of the second register transaction(Registration use case) (Mean per second)

C.11.3 ims_rereg : Time of the re-register transaction

This graph represents the time of re-register transaction i.e. the time between the REGISTER and the 200 OK.

		PX_TRT-REG4 (msec)							
Step	Requested Load	Mean	Std Dev	Min	Max	% 50	% 90	% 95	% 99
1	50	127.13	818.40	12.39	9202.82	20.6	35.2	70.7	4715.0

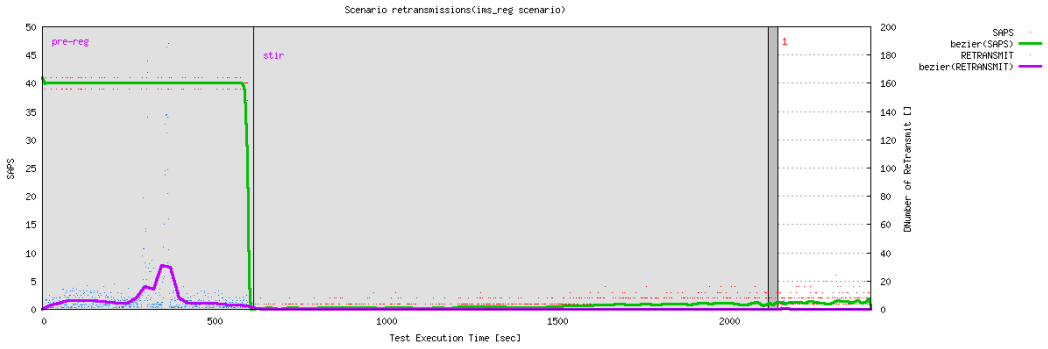


Time of the re-register transaction(ims_rereg scenario) (Mean per second)

C.11.4 ims_reg : Scenario retransmissions

This graph represents the number of retransmissions per seconds for this scenario.

		RETRANSMIT							
Step	Requested Load	Mean	Std Dev	Min	Max	% 50	% 90	% 95	% 99
Pre-reg	40	7.75	17.23	0.00	186.00	4.0	12.0	22.0	99.0
1	50	0.09	0.52	0.00	6.00	0.0	0.0	0.0	3.0

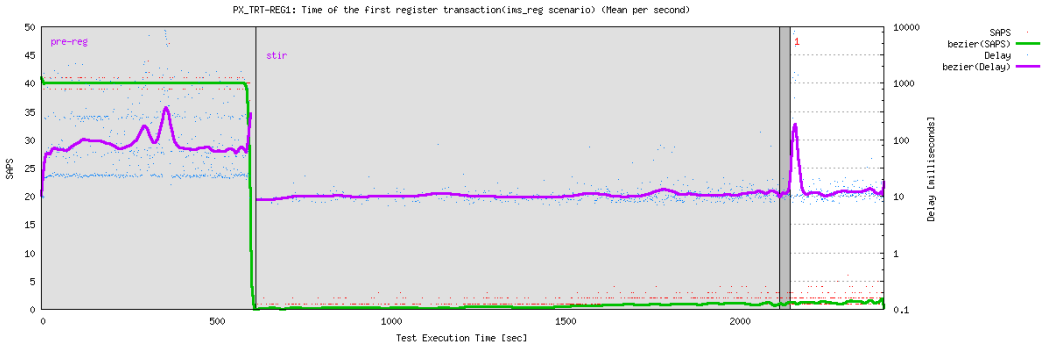


Scenario retransmissions(ims_reg scenario) over time

C.11.5 ims_reg : PX_TRT-REG1: Time of the first register transaction

This graph represents the time of the first register transaction i.e. the time between the REGISTER and the 401 Unautorized.

		PX_TRT-REG1 (msec)							
Step	Requested Load	Mean	Std Dev	Min	Max	% 50	% 90	% 95	% 99
Pre-reg	40	295.30	1641.96	6.17	27635.17	30.5	134.0	520.2	7619.0
1	50	105.07	730.38	7.29	8275.72	10.8	19.2	25.4	1576.0

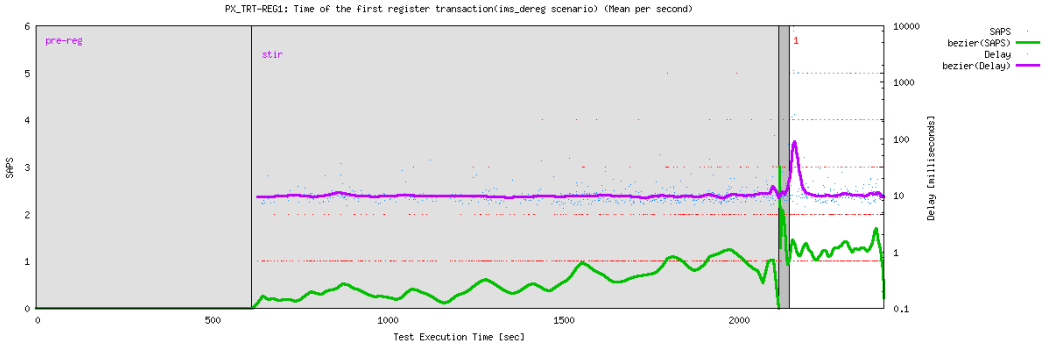


PX_TRT-REG1: Time of the first register transaction(ims_reg scenario) (Mean per second)

C.11.6 ims_dereg : PX_TRT-REG1: Time of the first register transaction

This graph represents the time of the first register transaction i.e. the time between the REGISTER and the 401 Unautorized.

		PX_TRT-REG1 (msec)							
Step	Requested Load	Mean	Std Dev	Min	Max	% 50	% 90	% 95	% 99
1	50	115.12	921.56	6.23	11563.91	9.6	18.1	21.2	7867.0



PX_TRT-REG1: Time of the first register transaction(ims_dereg scenario) (Mean per second)

C.12 Appendix

The following information is also available for the test

Parameter Name	Parameter Value	Parameter Info
rand_seed	1368442501	Value used to initialize the random number generators
prep_offset	2000	Time (ms) for scenario preparation (user reservation, etc.) prior to actual execution
highest_measured_time_offset	117	Highest time offset observed at startup between any test system and the manager (microseconds)

System	Command Line
TS1	/home/ubuntu/ims_bench/sipp -id 1 -i 10.0.103.3 -user_inf ./ims_users_1.inf -rmctrl localhost:5000 10.0.103.8:4060 -trace_err -trace_cpumem -trace_scen -trace_retrans
Manager	/home/ubuntu/ims_bench/manager -f manager.xml
SuT 1	./cpum 10.0.103.3
SuT 2	/opt/ims_bench/cpum 10.0.103.3
SuT 3	./cpum 10.0.103.3
SuT 4	./cpum 10.0.103.3
SuT 5	./cpum 10.0.103.3

ISBN 978-3-8442-7788-3