Converging towards Service Centric Networks: Requirements for a Service Delivery Platform Framework

Rolan Christian and Hu Hanrahan
Centre for Telecommunications Access and Services[1]
School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg
Telephone:(011-7177237), Fax:(011-4031929)
E-mail:{r.christian, h.hanrahan}@ee.wits.ac.za

Abstract – The telecommunications and *Information Technology (IT)* industry are converging into a single highly competitive market, where service diversity is the critical success factor. To provide diverse services, the telecommunications operator must evolve the traditional voice service centric network into a generic *Service Centric Network*. This transformation is aided by an architecture. An appropriate, but incomplete, architecture is the *Service Delivery Platform (SDP)*. The SDP aims at abstracting network, service and management complexities. Our research focuses on overcoming the complexities of attaining a SDP, by building a framework based various levels of abstraction. This paper provides a motivation and requirements for such a framework. In the paper, we first discuss convergence and its implications. We also uncover generic framework requirements based on these implications. Second, we define service centric networks and architectures. Third, we elaborate on the SDP and define additional framework requirements that overcome SDP specific complexities. Fourth, we discuss our research focus. Last, we conclude and discuss future work on the SDP.

Index Terms – Convergence, Service Centric Network, Service Delivery Platform.

I. Introduction

The *telecommunications (telecom)* industry is advancing towards a radical change in the communications business. Both incumbent and new operators realize that voice services alone do not increase revenue, as it has in the past. The source of this change is the current *Convergence* of the telecom, *Information Technology (IT)* and broadcasting sectors into a single, open and highly service competitive *Information Communication Technology (ICT)* market. Convergence has been fuelled by a variety of factors, such as technological advancements, user requirements and government deregulation.

To compete within a converged market, the telecom network operator must provide new and improved services over its competitors. In addition, services must be delivered securely, reliably and with guarantees over any technologies to any customer located anywhere using any device. Thus, *Service Diversity* is the key to dominating in a converging market.

[1]The Centre is supported by Telkom SA Limited, Siemens Telecommunications and the THRIP Programme of the Department of Trade and Industry

Pre-converged telecom networks focus on providing voice services only. To satisfy the service diversity requirements, the telecom network operator must evolve the current voice service centric network into a generic *Service Centric Network*.

To aid the evolution, an architecture abstracting various complexities is required. Examples of service centric network architectures are the traditional telecoms *Next Generation Network (NGN)*, the IT *Service Orientated Architecture (SOA)* and the IT *Service Delivery Platform (SDP)* infrastructure. These architectures are similar in nature, however in this paper and our research we focus on the SDP and the evolution of the telecom network into a SDP.

The SDP provides limited abstractions to manage telecom network complexities. Also, popular models of a SDP only provide a specific architectural view of the SDP, rather than a guiding framework to building a SDP. In addition, implementations of the SDP are based on nonstandardized technologies. Thus, the SDP concept is incomplete and its architecture alone cannot be used in evolving the telecom network. Due to these drawbacks, the telecom network operator must consider various other factors when evolving the telecom network into a SDP.

To support the evolution into a SDP a guiding framework, constituting a set of principles, concepts, rules and abstractions, must be provided. Hence, in this paper our problem focuses on *"what are the requirements for a framework that aids the evolution of the telecom network into a SDP"*.

A. Outline of Paper

We first discuss convergence, its drivers and implications. From these implications we uncover generic requirements for the framework, that are independent of the SDP. Second, we define service centric networks and architectures. Third, we elaborate on the SDP, its drawbacks and align the framework requirements with the SDP. Fourth, we discuss our research focus. Last, we conclude and discuss future work.

II. Background: Converging Networks

Worldwide telecom markets have been relatively closed to other industries. Currently, technological advancements, deregulation and opening of markets are all contributing to the opening of the telecom domain to the other domains, such as IT and broadcasting. As a result, these domains are going through a convergence.

With respect to the telecom and IT domains, a meaning of convergence is " combining the time-tested reliability of traditional voice services with highly dynamic, information-rich features of the Web and IT." [1]. This statement has the following implications for the telecom network operator:

- Business models must be based on value chains.
- Circuit and packet switching must interoperate, to reliably transmit any form of data.
- Signaling protocols must interwork across networks.
- Service platforms must operate independent of the underlying network technologies and must provide functionality to diverse application developers.
- Application development technologies must provide tools to support application creation, independent of the service platform that the applications access.
- Management technologies, such as *Operational Support Systems (OSS)* and *Business Support Systems (BSS)*, must administer mixtures of user, network, service and application technologies, across various networks.

We describe each of these convergence implications and uncover some generic framework requirements in the following sections.

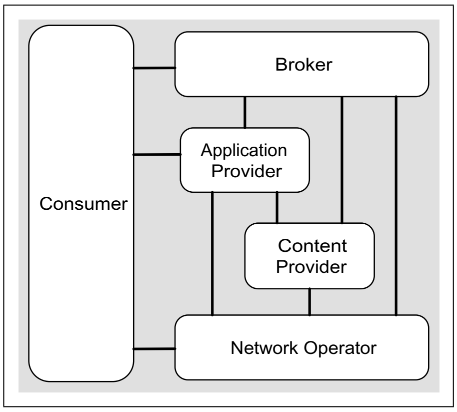

Fig. 1. Converging Networks Business Model

A. Business Models: Value Chains

Since service diversity is needed to compete within a converged market, the telecom network operator may form partnerships with application providers. These providers build applications that provide services to customers on the telecom network. Hence, the operator opens the network and its business model to external parties. An interpretation of this business model is shown in *Figure 1*.

The model depicts the following entities:

1) **Network Operator** : provides connectivity and network functionality to all entities of the business model.
2) **Application Provider** : develops applications using the network functionality. The applications provide telecom services that are used by consumers.
3) **Content Provider** : manages, sells and delivers content needed by application providers. Two categories of content are provided. First, content such as voice announcements used by a telecoms *Interactive Voice Response (IVR)* unit are provided. Second content such as video, music and images are provided.
4) **Broker** : retails services from application providers to consumers. In addition, the broker retails content from content providers to application providers.
5) **Consumer** : are individuals or enterprises who subscribe, consume and pay for services.

With this model a value chain is created, where all partners benefit. For instance, application providers use network functions to create applications. Application providers may also access content managed by content providers, via their brokers. Consequently, content providers also use network functions to delivery content to consumers. Application providers sell their services to consumers via brokers. Consumers now have more services to use, via their brokers. As a result the network operator benefits from the activity in the network.

This business model provides a starting point of the framework. That is, the model can be used in conjunction with the operator's current model to evaluate business orientated decisions. In addition, this model represents a highly abstracted view of a converged network, its partners and their functions.

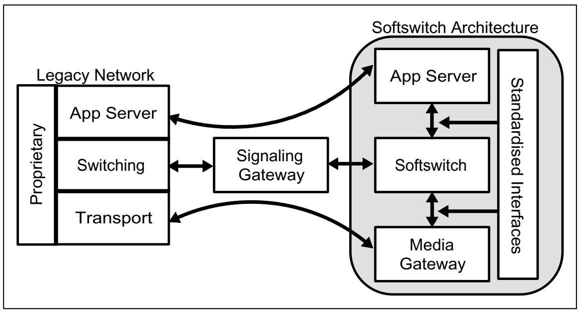

Fig. 2. Evolution of Softswitch Architecture

B. Network Technologies: Softswitch and Protocols

The *Softswitch* [2] enables different signalling and switching technologies to interoperate and reliably transmit any form of data. Also, the softswitch enables application development, independent from network technologies.

The softswitch architecture decomposes the conventional switch into various elements. The softswitch architecture and elements are modified from [3] and illustrated in *Figure 2*. This figure illustrates the evolution from proprietary closed network environments to open and standardized interfaces. The softswitch, its elements and their properties are:

- **Media Gateway** : is controlled by the softswitch and manages connections between different transport networks, such as the PSTN and Internet.
- **Signaling Gateway** : converts protocol messages originating from one network into other network specific protocol messages and transports them to the softswitch.
- **Softswitch** : contains logic for signaling, call processing and controls the media gateway. Also provides open, standardized interfaces to its functionality to the application server.
- **Application Server** : houses applications that use the softswitch interfaces to deliver services to customers on any network. The application server may be administered by an external application provider.

The softswitch architecture provides concepts and principles that are equivalent to those of the *Next Generation Network (NGN)*. These concepts and principles must also be included in the second stage of the framework, that is, the decomposition of the telecom network into various high level abstractions. These abstractions collectively represent a new open, secure and standardised network environment. These properties build upon the previous *business model stage* of the framework. In addition, the abstractions are technology independent, such that they hide transport network interworking and simplify service development and delivery.

C. Service Technologies: Parlay and Parlay X

The first form of service platform independence from network technologies, were evident in the various planes of the *Intelligent Network (IN) Conceptual Model* [4]. Currently, newer technologies are improving on network independent application and service platforms. Two of these technologies are Parlay and Parlay X.

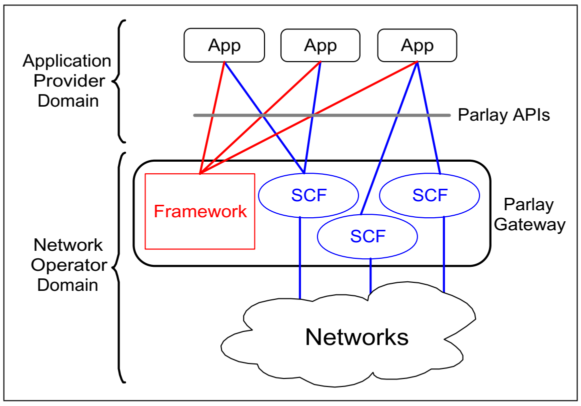

Fig. 3. Parlay Architecture

The Parlay Group [5] defines an architecture of *Application Programming Interfaces (API)*, housed within *Service Capability Features (SCF)*, that abstract various generic network functions. These APIs allow application providers to use their functionality to create services for small to large scale consumption.

Each API describes how authorized applications can use generic network functionality, such as setting up multiparty conference calls or data sessions. In addition, the APIs enable applications to query network databases to obtain user location, account or presence information. Thus, the APIs abstract from the application developer the complexity of the network equipment and protocols. The Parlay architecture is shown in *Figure 3*. Example SCFs are:

- **Framework SCF** [6] : manages all SCFs, performs load management and enforces security.
- **Generic Call Control SCF** [7] : enables applications to create and manage calls in the network.

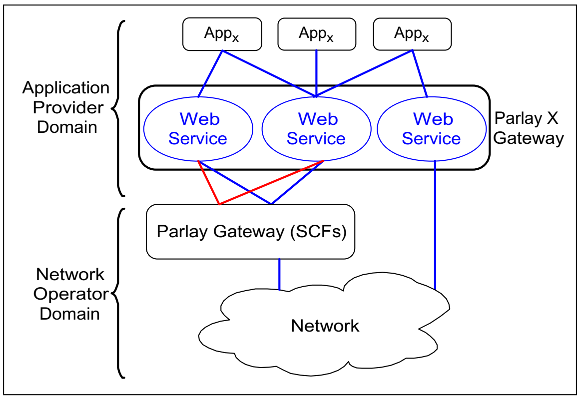

Fig. 4. Parlay X Architecture

The Parlay Group also defines an additional architecture based on web services [8]. This architecture is the Parlay X architecture [9].

In Parlay X, highly abstracted generic network functionality is represented as web services, for use by authorized external service developers. The Parlay X architecture is shown in *Figure 4*. The figure depicts the web services using either the SCFs or communicating directly over various networks, to fulfil their specific functionality. Examples of the web services are:

- **Third Party Call** : used for creating and managing a call initiated by an application.
- **SMS** : used by an application for sending and receiving a SMS over the network.

Parlay and Parlay X do not specify a standard describing how network complexities are abstracted and used by the SCFs or web services. In addition, a user domain architecture is not specified in both architectures.

Unlike Parlay, there is no overall controlling mechanism, such as a framework web service, in Parlay X . Since Parlay X is based on Internet web services it inherits other disadvantages. For example, web services do not cater for states. Also, web services lack availability, security and atomicity of transactions [10].

Though both Parlay architectures have drawbacks, they provide vital concepts that must be present in the third stage of the framework. For instance, the ability to open the network and provide standardised, secure and managed generic network services that are reused in service creation. These network services provide abstractions that simplify or hide network complexities. Also, different levels of service abstractions support different developers in service creation. These properties build upon to the previous *softswitch stage* of the framework.

D. Application Technologies

Application development technologies used to create services must provide tools to support rapid application development independent of the service platform that the applications access.

These technologies must provide execution environments as well as APIs that abstract various hardware and protocols. That is, these technologies must therefore provide a *middleware-like* environment to support application development.

Nevertheless, the onus is on the application provider to determine application technologies that provide the necessary mechanisms to support their specific business and customer needs.

With respect to choosing application technologies, the framework must remain a technology independent guide, such that any technologies may be mapped to the appropriate parts of the evolving telecom network.

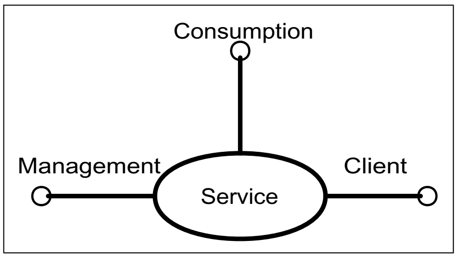

Fig. 5. Service Interface Architecture

E. Management Technologies

Management applications require access to the network's physical and logical entities, such as network gateways and service software. In addition, complex management information must be simplified, such that management applications can easily process it.

Focusing on logical entities in the telecom network, [11] suggests that every service must provide three interfaces. These are *consumption*, *client* and *management* interfaces, adapted from [11] and illustrated in *Figure 5*. The consumption interface publicly exposes a service's functionality. For instance, a messaging service exposes its functionality, to send and receive messages, via its consumption interfaces.

The client interface allows a service to use other services functionality. For instance, a video on demand service may have a client interface to use the functionality offered by a real-time content delivery service.

A service's management interface enables fault, configuration, accounting, performance and security administration to be performed on it. For instance, a conference call service must provide accounting related information to billing management services.

These interface concepts must be reused throughout the framework. The various abstracted entities that the framework will uncover must have management functionality pre-built into them. This ensures complete management of the technologies that will implement the abstractions.

III. The Telecom Service Centric Network

To manage the above convergence implications and to compete in a converged, open and deregulated market the telecom network operator must evolve the traditional voice service centric network into a generic *Service Centric Network*.

A service centric network delivers various services to customers, independent of the type of device, access, transport and signalling technologies. In addition, the network manages *authentication, authorization and accounting (AAA)*, QoS, reliability, etc. To visualize a service centric network various architectures have been proposed. We discuss service centric network architectures in the following sections.

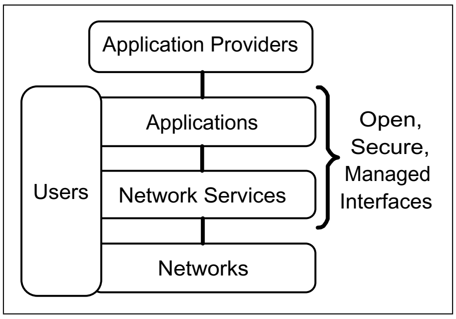

Fig. 6. A NGN Representation

A. Next Generation Network (NGN)

The NGN intended to aid in the evolution of legacy *Public Switched Telephone Network (PSTN)* into service centric packet networks. Adapting the NGN working definition from [12], the NGN:

1) provides a packet-based transport network that interworks with other network types.
2) provides telecoms and IT services to fixed and mobile users over the packet network.
3) separates network independent service related functions from underlying transport technologies.
4) offers controlled access to network service functions to both users and application providers.
5) facilitates the provision of services from application providers to users (customers).

A representation of a NGN is shown in *Figure 6*. Similar to the NGN is the IT industries *Service Orientated Architecture (SOA)*.

B. Service Orientated Architecture (SOA)

Enteprise IT companies develop or purchase applications to automate their business processes. The applications use services offered by the IT infrastructure to perform atomic transactions, setup conference calls, send emails, etc. These applications may also operate across internal and external borders, where infrastructure may vary. In an attempt to support enterprise application development, independent of the infrastructure, the SOA was developed.

The SOA manages a collection of hardware independent application functionality that is abstracted into reusable entities called services [13]. Most services abstract enterprise resources, such as data, people and communication channels. Consequently, the management of the enterprise data and communication resources may be outsourced to external entities, such as a telecoms operator.

The SOA is however based on the immature web services architecture and standards. As described in previous sections web services do not address issues such as availability or security. Another architecture that expands the contents of both NGN and SOA, is the *Service Delivery Platform (SDP)* architecture.

C. Service Delivery Platform (SDP)

The SDP is designed specifically for delivering services to users of communications technologies [11]. That is, the SDP uses telecom, IT and content-based resources to deliver managed telecom services, to users, independent of the network infrastructure. These telecom services include number translation, click-to-dial, location based services, virtual call centers and mobile gaming [14].

Unlike the web services based SOA, the SDP architecture constitutes various abstractions that can be implemented with a variety of technologies. A graphical representation of the SDP, adapted from [14], is shown in *Figure 7*. According to [14] these abstractions represent common vendor products used in SDP implementations. These abstractions are:

1) **Network Abstraction Platform** : applications providing abstracted interfaces to generic network services and network elements reside here.
2) **Service Execution Platform** : applications providing value added telecom services are deployed and executed here.
3) **Content Delivery Platform** : manages content needed by applications and delivered to users.
4) **Service Exposure Platform** : provides highly abstracted interfaces to value added telecom services to application developers service creation environments.
5) **Management Platforms** : provides interfaces to accounting, operations and other management platforms.

From the definition, the SDP concept provides a complete managed environment offering various levels of abstraction.

This SDP representation has various disadvantages. First, the matching of appropriate technologies to each abstraction of the SDP is a difficult task. These technology choices must be standardised, provide needed abstractions and enable complete management of the SDP. Second, the SDP does not provide a user domain abstraction. That is, the user access, user service provisioning and user to service signalling complexities are not managed. Third, the architecture only provides a extremely simplistic view of the abstracted components of a SDP. The contents of these abstracted components must also be defined and illustrated.

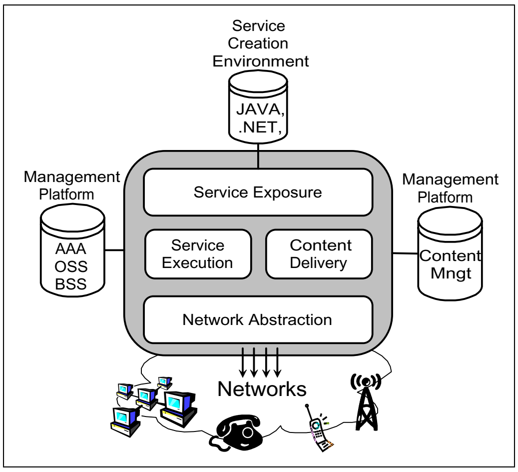

Fig. 7. SDP Architecture

Although this SDP representation has drawbacks, we may still map standardised technologies to the abstractions.

IV. Technologies for SDP

Technologies that may be used to implement the SDP architecture include the softswitch. That is, the softswitch interworks different signalling protocols and transport networks. Also, Parlay fits appropriately between the service execution and network abstraction platforms. Parlay X also fits appropriately into the service exposure abstraction. However, Parlay X must become telecom-enabled, such that the web services limitations are fixed.

SDP management may also benefit from telecom-enabled web services. These web services can be used to expose highly abstracted and generic network management functionality to management applications residing on various management platforms.

V. SDP Framework

The previously discussed convergence implications contribute generic requirements for the SDP framework. These requirements describe stages containing abstractions that simplify various complexities in evolving into a SDP.

The SDP architecture in *Figure 7* has a variety of limitations that the framework must overcome. For instance, the definition of various domain abstractions, such as a user domain, are required. These domain abstractions must expose varying details throughout the stages of the framework.

In addition, the framework requires a *network stage* that defines numerous network abstractions. These network abstractions manage the underlying transport networks and signalling protocols. This stage of the framework is built upon the *softswitch stage*. An outcome of this stage is a set of powerful and generic network services that enable the full potential of the undelying network technologies to be used. This stage may benefit from reusing concepts defined in the *Telecommunication Information Networking Architecture (TINA)* [15].

The framework also requires abstractions to simplify content delivery and management complexities of the SDP. These abstractions should define services that provide functionality enabling a variety of media types to be delivered over any network technology. These abstractions may be added to the *service stage* of the framework.

Based on these requirements, each stage of the framework contains varying levels of user, network, service, content, application or management abstractions. An abstraction may simplify its corresponding complexities by representing itself as a set of functional groupings, information distributions or logical interactions/relationships. These representations may be visualized in either a model or architecture.

VI. Research Focus

This paper identifies requirements for a SDP framework. These requirements call for further research. The focus of this research is, *"providing a framework that uses various levels of abstraction for managing complexities in attaining Service Delivery Platforms"*.

The research intends to combine generic methodologies for managing complexity with telecom-specific network methodologies, architectures and frameworks. The results of this combination will be incorporated into a prototype framework. The aim of the prototype framework is to provide a structured, managed and traceable process in building or evolving telecom networks into SDP-compliant networks.

The validity of the prototype framework will be tested against the framework requirements and network evolution scenarios. These scenarios will include mapping technologies, such as the *Internet Protocol Multimedia Subsystem (IMS)* [16] components, to SDP abstractions.

The validated framework will benefit the evolution process by providing various levels of abstractions that simplify network-wide complexities, while adhering to the fundamental concepts of the SDP. Also, once the SDP framework is applied to a telecom network, the network will have a foundation and mechanisms for building other service centric networks. For instance, a SOA for IT enterprise customers can be provided on top of the SDP.

Therefore, the framework will aid the operator in attaining an adaptable service centric network.

VII. Conclusion

In this paper we have described convergence and its effects on the telecom network. As a result, the evolution of the telecom voice service centric network into a generic service centric network is required. We have motivated a limited service centric network architecture, the SDP, to form part of the evolution. To structure and manage the evolution, a guiding SDP framework has been proposed. The generic requirements for the framework have been uncovered using various convergence-related models and architectures. We have also identified specific framework requirements based on overcoming SDP limitations. Hence, the framework provides various levels of abstraction of the SDP. These abstractions are visualized through various architectural representations. We have also identified further research that will reuse the SDP framework requirements.

References

[1] J. Soref and D. Troughton. *The Real Meaning of Convergence*. A Morina Group Thought Leader Report, Section B: Thought Leadership, Last accessed on 17/01/2005, http://www.morinagroup.com, June 2004.

[2] The International Packet Communication Consortium. Last accessed 07/03/2005, http://www.packetcomm.org.

[3] F. D. Ohrtman. JR. *Softswitch: Architecture for VoIP*. Networking Professional. McGraw-Hill, New York. USA, 2003.

[4] ITU-T. *Principles of Intelligent Networks*. Recommendation I.312/Q.1201, October 1992.

[5] The Parlay Group. Last accessed on 26/01/2005, http://www.parlay.org.

[6] ETSI ES 202 915-3. *Open Service Access (OSA); Application Programming Interface (API); Part 3: Framework (Parlay 4)*. ETSI Standard V1.2.1, August 2003.

[7] ETSI ES 202 915-4-2. *Open Service Access (OSA); Application Programming Interface (API); Part 4: Call Control; Sub-part 2: Generic Call Control SCF (Parlay 4)*. ETSI Standard V1.2.2, August 2003.

[8] D. Booth *et al*. *Web Services Architecture*. Working Group Note, World Wide Web Consortium (W3C) Web Services Architecure Working Group, February 2004.

[9] The Parlay Group. *Parlay X Web Services Specification*. Technical Sepcification 1.0.1, June 2004.

[10] K. Birman. *League of SuperNets*. *IEEE Internet Computing*, 7 (5):93–97, September/October 2003.

[11] B. Balabaskaran. *Development of New Services with a Service Delivery Platform (SDP)*. A Morina Group Thought Leader Report, Section B: Thought Leadership, Last accessed on 17/01/2005, http://www.morinagroup.com, June 2004.

[12] International Telecommunications Unioun - Telecommunication Standardization (ITU-T). *Definition of NGN*. Last accessed on 25/01/2005, http://www.itu.int, November 2004.

[13] G. Parkins. *A Service Orientated Architecture for Telecom Services*. In *KeyNote Session*. 9^{th} International Conference on Intelligence in Service Delivery Networks (ICIN) 2004, Bordeaux, France, October 2004.

[14] The Moriana Group. *Service Delivery Platforms and Telecom Web Services*. A Morina Thought Leader Report, Section A: Executive Summary, Last accessed on 17/01/2005, http://www.morinagroup.com, June 2004.

[15] The TINA Consortium (TINA-C). Last accessed on 26/01/2005, http://www.tinac.com.

[16] 3^{rd} Generation Partnership Programme (3GPP). *Technical Specification Group Services and System Aspects; Service Requirements for the Internet Protocol (IP) Multimedia Core Network Subsystem; Stage 1 (Release 7)*. Technical Specification TS 22.228 V7.0.0, September 2004.

Biographies

Rolan Christian holds a BSc and BSc Honors degree in Information Technology from the University of Natal. He also holds a MSc degree in Electrical Engineering from the University of the Witwatersrand. He is currently pursuing his Phd degree in Electrical Engineering at the University of the Witwatersrand.

Hu Hanrahan is Professor of Communications Engineering at the University of the Witwatersrand. He leads the Centre for Telecommunications Access and Services (CeTAS), a research and advanced teaching centre devoted to improving knowledge and practice in the evolving telecommunicationms access networks and telecommunicationms services.