# An Appraisal of Secure, Wireless Grid-Enabled Data Warehousing

**Gaolathe Seelo**

# An Appraisal of Secure, Wireless Grid-Enabled Data Warehousing

By

**Gaolathe Seelo**

Submitted in fulfilment
of the requirements
for the degree

**Magister Technologiae**

in

**Information Technology**

in the

**School of Information and Communication Technology**

in the

**Faculty of Engineering, the Build Environment and Information Technology**

of the

**Nelson Mandela Metropolitan University**

**Promoter: Dr. Maree Pather**

**January 2007**

# Declaration

I, **Gaolathe Seelo**, hereby declare that:

- The work in this dissertation is my own work.
- All sources used or referred have been documented and recognized.
- This dissertation has not been submitted in full or partial fulfillment of the requirements for an equivalent or higher qualification at any other recognized educational institution.

Gaolathe Seelo

# Abbreviations and Acronyms used

| | |
|---|---|
| 3GPP | Third-Generation Partnership Project |
| AES | Advanced Encryption Standard |
| ANSI | American National Standards Institute |
| AP | Access Point |
| API | Application Programmer Interface |
| APN | Access Point Name |
| AuC | Authentication Center |
| BS | Base Station, also known as Node B |
| BSC | Base Station Controller |
| BSS | Base Station Subsystem |
| BTS | Base Transceiver Station |
| CA | Certification Authority (for digital certificates) |
| CAMEL | Customised Applications for Mobile network Enhanced Logic |
| CDMA | Code-Division Multiple Access |
| CN | Core Network |
| CS | Circuit Switched Domain |
| CSCF | Call Session Control Function |
| CSG | Candidate Set Generator |
| CSS | Catching and Synchronization Server |
| DES | Data Encryption Standard |
| DNS | Domain Name System |
| DOM | XML Document Object Model |
| DPHS | Data Participant HTTP Server |
| DPR | Data Participant Router |
| DTD | XML Document Type Definition |
| EAP | Extensible Authentication Protocol |
| EDGE | GSM and Enhanced Data rates in GSM Environment |
| EIR | Equipment Identity Register |
| EMS | OGSA  Execution Management Services |
| EPR | WS-Addressing Endpoint Reference |
| EPS | Execution Planning Services |
| ERD | Entity Relationship Diagram |
| FDMA | Frequency-Division Multiple Access |
| FTP | File Transfer Protocol |
| GGF | Global Grid Forum |
| GGSN | Gateway GPRS Support Node |
| GMSC | Gateway MSC |
| GPRS | General Packet Radio Services |
| GRAM | GT4's Grid Resource Allocation and Management |

| | |
|---|---|
| GridFTP | Grid File Transfer Protocol |
| GSI | Grid Security Infrastructure |
| GSM | Global System for Mobile communication |
| GT4 | Globus Toolkit 4.x |
| HSCSD | High Speed Circuit Switched Data |
| HLR | Home Location Register |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transport Protocol |
| I-CSCF | Interrogating CSCF |
| IEEE | Institute of Electrical and Electronics Engineers |
| IETF | Internet Engineering Task Force |
| IMEI | International Mobile-station Equipment Identity |
| IMS | IP Multimedia Subsystem |
| IMSI | Mobile Subscriber Identity |
| IN | Intelligent Network |
| IP | Internet Protocol, a connectionless network layer protocol. |
| IPSec | Secure Internet Protocol, provides security services between pairs of nodes |
| IS-41 | **I**nterim **S**tandard-**41**: The network standard that allows all switches to exchange cellular subscriber information from one carrier to another based on both analog and digital United States standards. |
| ISO | International Standards Organisation |
| ISP | Internet Service Provider |
| JMS | Job Manager Service |
| JSDL | Job Submission Description Language |
| MAP | Mobile Application Part |
| Mcps | Megachips per second |
| MDS4 | Monitoring and Discovery Service in GT4 |
| ME | Mobile Equipment |
| MGCW | Media Gateway Control Function |
| MGW | Media Gateway |
| MSC | Mobile-services Switching Center |
| MT | Mobile Termination |
| MTOM | Message Transmission Optimisation Mechanism |
| NMS | Network Management System |
| NMSI | National Mobile Subscriber Identity |
| NSAPI | Network-layer Service Access Point Identifier |
| NSS | Network Switching Subsystem |
| OASIS | Organization for the Advancement of Structured Information Standards |
| OGSA | Open Grid Services Architecture |
| P-CSCF | Proxy CSCF |
| PDP | Packet Data Protocol |

| | |
|---|---|
| PDU | Packet Data Unit |
| PKI | Public Key Infrastructure (for digital certificates) |
| PS | Packet Switched Domain |
| PSTN | Public Switched Telephone Networks |
| P-TMSI | Packet TMSI |
| QoS | Quality of Service |
| RAB | Radio Access Bearer |
| RADIUS | Remote Authentication Dial In User Service |
| RAN | Radio Access Network |
| RFT | Reliable File Transfer |
| RLS | Replica Location Service |
| RNC | Radio Network Controller |
| RNS | Radio Network Subsystem |
| RPC | Remote Procedure Call |
| SAML | Security Assertion Markup Language |
| S-CSCF | Serving CSCF |
| SGSN | Serving GPRS Support Node |
| SIM | Service Identity Module |
| SIP | Session Initiation Protocol |
| SMS | Short Message Service |
| SMSC | Short Message Service Centre |
| SOA | Service-Oriented Architecture |
| SOAP | Simple Object Access Protocol, an XML-based messaging protocol |
| SPR | Service Participant Router |
| SQL | Structured Query Language |
| SSL | Secure Sockets Layer, a protocol for incorporating encryption into e-commerce transactions, developed by Netscape |
| SS7 | Signalling System Number 7 |
| TCP | Transmission Control Protocol |
| TCP/IP | Transmission Control Protocol/ Internet Protocol |
| TDMA | Time-Division Multiple Access |
| TE | Terminal Equipment |
| TEID | Tunnel Endpoint Identifier |
| TKIP | Temporal Key Integrity Protocol |
| TLS | The IETF’s Transport Layer Security Protocol |
| TMSI | Temporary Mobile Subscriber Identity |
| TRAU | Transcoding and Rate Adaptation Unit |
| UDDI | Universal Discovery Description and Integration standard for Web services |
| UE | User Equipment |
| UMTS | Universal Mobile Telecommunications Service |
| URL | Uniform Resource Locator |
| U-RNTI | UTRAN - Radio Network Temporary Identity |
| USIM | UMTS Subscriber Identity Module |

| | |
|---|---|
| UTRAN | UMTS Terrestrial RAN |
| VLR | Visitor Location Register |
| WAP | Wireless Application Protocol |
| WCDMA | Wideband Code-Division Multiple Access |
| WGDW | Wireless Grid-enabled Data Warehousing |
| WMSC | WCDMA Mobile Switching Center |
| WSDL | Web Services Description Language, an XML grammar for describing Web services |
| WS-* | Refers to all Web services standards |
| WS-S | Web services security standard |
| XML | Extensible Mark-up Language |
| XML DSIG | XML Digital Signature |
| XSD | XML Schema Document |
| XSLT | XML Stylesheet Language Transformation |

# Abstract

In most research, appropriate collections of data play a significant role in aiding decision-making processes. This is more critical if the data is being accessed across organisational barriers. Further, for the data to be mined and analysed efficiently, to aid decision-making processes, it must be harnessed in a suitably-structured fashion. There is, for example, a need to perform diverse data analyses and interpretation of structured (non-personal) HIV/AIDS patient-data from various quarters in South Africa. Although this data does exist, to some extent, it is autonomously owned and stored in disparate data storages, and not readily available to all interested parties. In order to put this data to meaningful use, it is imperative to integrate and store this data in a manner in which it can be better utilized by all those involved in the ontological field. This implies integration of (and hence, interoperability), and appropriate accessibility to, the information systems of the autonomous organizations providing data and data-processing. This is a typical problem-scenario for a Virtual Inter-Organisational Information System (VIOIS), proposed in this study.

The VIOIS envisaged is a hypothetical, secure, Wireless Grid-enabled Data Warehouse (WGDW) that enables IOIS interaction, such as the storage and processing of HIV/AIDS patient-data to be utilized for HIV/AIDS-specific research. The proposed WDGW offers a methodical approach for arriving at such a collaborative (HIV/AIDS research) integrated system. The proposed WDGW is virtual community that consists mainly of data-providers, service-providers and information-consumers. The WGDW-basis resulted from systematic literature-survey that covered a variety of technologies and standards that support data-storage, data-management, computation and connectivity between virtual community members in Grid computing contexts.

A Grid computing paradigm is proposed for data-storage, data management and computation in the WGDW. Informational or analytical processing will be enabled through data warehousing while connectivity will be attained wirelessly (for addressing the paucity of connectivity infrastructure in rural parts of developing countries, like South Africa).

## Acknowledgements

My sincere thanks are due to:

- My promoter, Dr. Maree Pather, for his outstanding guidance, invaluable support, wisdom and great supervision;

- To my mother and my family, for the wonderful upbringing, love and support;

- To my three best friends, J. Twani, T. Gaadingwe and J. Dikgang, for playing a significant role in my life;

- To Prof. Botha, Prof. Von Solms and Dr. Pottas, for all the help and guidance;

- To Mrs Bron Kaplan, “The Language Specialist”, for all the help correcting my work;

- To Nelson Mandela Metropolitan University, for the financial assistance;

- God, for blessing me in every way.

# Table of Contents

# List of Tables

## List of Figures

# Chapter 1

# Introduction

## INTRODUCTION

As with most research, in medical research, appropriate collections of data can play a fundamental role in aiding decision-making processes. However to maximise its usefulness, the data should stored in a structured fashion. Structured data enables end-users/decision makers to analyse the data efficiently and to make informed decisions from these analyses.

For the sake of easier illustrating concepts explored in this dissertation, in relation to a generic problem-context, a specific hypothetical problem-context will be used. The generic characteristics (delimitations) which are intended to be encompassed by the hypothetical problem included the following:

- Integration of the information systems amongst a group of autonomous organizations, with some common business objectives to constitute a Virtual Inter-organisational Information System (VIOIS);
- Using disparate computing platforms;
- Wishing to share resources (e.g. data- and computing resources) across trust domains; and
- Operating in a developing country, with limited last-mile Internet connectivity.

The VIOIS is a separate “entity” although it derives from the autonomous systems below it.

The specific hypothetical problem-context can be described as follows: There is a need to perform analyses and interpretation on HIV/AIDS patient-data in South Africa. However, typically, HIV/AIDS patient-data is autonomously owned by hospitals, clinics, hospices, private medical practitioners, etc. This data can potentially be held in disparate databases, structured documents in structured assemblies of binary files, data warehouses and data marts. It would be most convenient if the collected data on HIV/AIDS were to encompass data from across the country, if it were shared with other research institutions, and if the various data-collecting institutions were to allow for collaborative research. For this to occur, infrastructure and facilities for data-storage, data management, computation and connectivity between the research institutions, etc., would be required. Moreover, various communities of researchers that need to collaborate and analyse data would need to transcend their geographic distribution and disparate storage and computing platforms.

It will be contended in this dissertation that: data-storage, data management and computation facilities can provide VIOIS integration at this level through employing a proposed Grid computing paradigm, using evolving wireless technologies (for addressing the paucity of connectivity infrastructure in rural parts of developing countries, like South Africa).

With the advent of Grid computing, computing environments no longer had to be homogeneous and centrally-managed. Grid computing evolved to address the need for collaboration, resource-sharing, and other new modes of interaction that involve heterogeneous distributed systems belonging to different administrative domains over a network using open standards and protocols (Foster, Kesselman & Tuecke, 2001; Foster, Kesselman, Nick & Tuecke, 2002). The use of open standards and protocols in Grid computing solutions has gained them one thing that is essential for their success, which is ubiquity of peer resources (Pullen *et al*., 2004).

Grid computing offers a model for solving massive computational problems by making use of the unused resources of large numbers of disparate computers. The envisaged goal is to enable users and applications to seamlessly access these resources to solve complex, large-scale problems, whether in science, engineering, or commerce (Foster *et al*., 2001; Foster *et al*., 2002; Czajkowski, Fitzgerald, Foster & Kesselman, 2001).

Grid computing can be utilized to provide an HIV/AIDS VIOIS (collaboration-platform) by enabling access to distributed computation and data-storage facilities. These resources, for example, computers, networks, servers, storage systems, etc., could be contributed by, e.g., medical institutions, universities and government research centres. (Please note, in the context of Grid computing, the terms “services” and “resources” will be used interchangeably, in this text; service providers can also provide data/information resources and data-processing resources. Data-providers contribute data to service-providers).

HIV/AIDS patient-data (which will be aggregated before being stored) is contributed to the service-providers by data-providers, which would be mainly health institutions, including hospitals (private and public), clinics, hospices, private medical practitioners, etc. The submission of this data can be accomplished through the Internet. However, some of the health institutions are located in rural areas and lack viable Internet-connectivity infrastructure which would impede their data contribution. A wireless connectivity solution is thus a necessary option; the data-providers can utilize wireless connections to contribute data to the service-providers.

Once the aggregated HIV/AIDS patient-data reaches the service-providers, it can be stored in data warehouse and data marts. End-users, mostly HIV/AIDS researchers, academics and medical practitioners, can subscribe to the use of the aggregated HIV/AIDS patient-data. This data can be used to perform HIV/AIDS-specific research and data analyses.

A joint-venture among the data-providers, service-providers and subscribers to the services, would, ideally, motivate HIV/AIDS VIOIS. All the participants could work together to carry out HIV/AIDS research, and in turn, exchange knowledge and aid in decision-making processes. The different research-institutions could share research-findings and collaborate to, for example, detect trends in the data, which would benefit all involved.

Such a collaborative HIV/AIDS research-project should utilize existing standards and technologies, such as is implicit in Grid computing and data warehousing, to ensure that interoperability can be attained. Connectivity between the data- and service-providers could be attained wirelessly.

The following sections summarize some of the requirements for a proposed solution – provided merely in basic outline in this chapter – for the hypothetical problem-domain.

**Data Collection**

Data-collection for this purpose ensues as a system-within-a-system; a common application is placed on the data-providers' operational systems. The data-collection components are concerned only with collecting HIV/AIDS patients' data from distributed data-providers' operational systems. The extracted data has to be re-formatted before being transferred to the destination data-storage resources. Only data which is relevant to (HIV/AIDS) VIOISs is extracted, for example, diagnoses and treatments (which are frequently updated in the operational environment), sexual-orientation, ethnicity, etc. However, personal information, for example, first name, surname, contact information, etc., is not collected.

The collection of data is performed periodically (periodicity is agreed upon by data-providers and service-providers). The data-collection components extract only required data (e.g., new patient-data, updated diagnosis and treatment, etc.) from the data-providers. This data, mainly demographic, is transformed (aggregated), integrated, and then stored in data warehouses and data marts. Granularity of the data warehouse determines the data relevant for collection from the data-providers. Granularity is the most important aspect in a data warehouse environment, because it very much affects the volume of data stored in a data warehouse, and the type of queries that can be answered. Thus, it enables data analysis capabilities, such as identifying trends, correlations and anomalies, testing theories and deriving patterns.

**Data Analysis**

The large collections of aggregated HIV/AIDS patient-data in the data warehouses and data marts enable end-users (subscribers) of (HIV/AIDS) VIOISs to perform HIV/AIDS-specific research and data analysis. Subscribers can perform statistical analyses, in order to derive patterns, test theories and identify trends, correlations and anomalies in the HIV/AIDS patients' data. Thus, a subscriber is able to pose analysis queries such as, "Find out the opportunist diseases, which are prevalent in the Eastern Cape province of South Africa" or "Find out the number of patients who died while on anti-retroviral drugs compared to those who were not". These, and many more analyses, can be performed in the HIV/AIDS research collaboration environment.

The format and the structure of the data in the data warehouses and data marts are standardized, in order to achieve interoperability and seamless sharing of data among the different service-providers. To this end, a "global" format to be followed will be described in a generic global XML Schema, distributed among the participating service-providers. This will enable the processing of user queries to be standardized, which saves users and providers considerable time.

The data analysis components depend on the researchers' requirements to determine whether a query should be executed remotely or locally. If the query requires data or computation components that are not available locally, the query can be executed remotely.

### Data and Knowledge Sharing

The data collections on the different service-providers in (HIV/AIDS) VIOISs will be shared among the service-providers and the research community at large. The service-providers offer speciality data warehouses and data marts dependent on the research direction and the research interest. If an end-user submits a query, it is re-directed to the appropriate service-provider. However, some of the data can be redundant across service-providers, in order to avoid single-point-of-failure.

Service-providers will publish the type and volume of aggregated HIV/AIDS patient-data they possess, in order to enable other service-providers to discover it. This will enable end-users to submit any query to a service-provider, which will, in turn, locate the appropriate service-provider (including itself) to process the query. Once a query has been processed, the results are stored in a specific data mart for future use, for instance, if the user needs to use the same results again or another user poses the same query.

The publishing of data and query results enables data-providers to share data and knowledge among each other. This also minimizes costs of individually storing large amounts of data and reduces the time it takes to perform (HIV/AIDS-specific) data analysis. For instance, suppose an HIV/AIDS researcher at the Nelson Mandela Metropolitan University (NMMU) needs to analyze data that is held in data warehouses at the University of Cape Town (UCT) and the University of Johannesburg (UJ). The researcher launches his/her query through their institution (NMMU), which in turn locates the required data for processing the query (available in UCT and UJ data warehouses), by querying data registries of participating service-providers. Once the data is located (in UCT and UJ data warehouses) the data analysis can commence. (Resource- and execution-management of distributed processing-components are, of course, also implicated and will be considered in later chapters).

**Security**

Data-sharing is a significant aspect of (HIV/AIDS) VIOISs. However, allowing different institutions access to one's resources and data poses many potential security risks. There must be security mechanisms in place, such as, access control, authentication and authorisation. These mechanisms handle the data-sharing relationships among participants, allowing new participants to be added dynamically.

## 1.1 Motivation for this Study

In computing-intensive and data-intensive research environments, such as in HIV/AIDS research, in developing countries there is lack of suitable infrastructure and facilities to support data-storage, data management and computation. Typically, the research institutions operate individually. Furthermore, these research institutions cannot collaborate with one another, because they might use different systems, and could structure their data differently.

## 1.2 Problem Statement

The principal problem that needs to be addressed can be stated as follows: There exists a need for collaboration between various related institutions in South Africa, regardless of their type, size or location. HIV/AIDS research typifies this need. Can a collaboration platform be designed that will facilitate secure data-sharing, data analysis, connectivity and interoperability among different (HIV/AIDS research) institutions, which overcomes the problems inherent in computing-intensive and data-intensive research that includes rural institutions?

In order to address this problem the following questions must be investigated.

- How will data from remote participants be submitted to service-providers?
- What technologies and standards should be utilized to ensure connectivity, data-sharing, data analysis and interoperability?
- How will (HIV/AIDS research) institutions access the disparate distributed resources?

- How will data be securely transferred from data-providers to service-providers?
- How will data be secured once at the service-providers?

## 1.3 Objectives

The objective of this study is to design a platform that will facilitate secure data-sharing, data analysis and interoperability between the related (HIV/AIDS research) organisations. The focus of this study is to arrive at a proposed collaboration "platform", which will support groups of (HIV/AIDS research) organisations in South Africa, regardless of their type, size or location, and enable them to perform meaningful (HIV/AIDS-specific) data analysis.

As part of the process of developing the "platform", a number of points will be addressed, including:

- A discussion on how HIV/AIDS research institutions will access the disparate distributed resources
- A description of what technologies and standards should be utilized to ensure data-sharing, data analysis and interoperability
- A discussion on how service-providers will submit data in order to include remote participants
- A description of how data will be securely transferred from data-providers to service-providers
- A discussion on how data will be secured once at the service-providers

## 1.4 Methodology

The methodology primarily comprises literature-survey and scientific argument. This implies an in-depth literature study of models, technologies and standards pertaining to data-sharing, data analysis and interoperability in the chosen generic problem-context (described earlier). Specific focus is given to wireless networks, Grid Computing, as well as data warehousing, for reasons delineated above. By examining the requirements for addressing the research problem, and by utilizing the available literature available on various pertinent aspects of the problem-domain, a solution framework will be proposed through scientific argument.

## 1.5 Brief Chapter-Analysis

Chapter 2 explores wireless networks and how they can be implemented in (HIV/AIDS) VIOISs. The chapter then discusses the evolution of wireless networks, up to 3G networks, and also the architecture of 3G networks and the radio access technologies utilized. The 3GPP network elements will then be examined, with a focus on the Core Network Packet-Switched (CN PS) Domain. The CN PS will be utilized to transport data packets from the various health institutions. The way in which these packets are transported will also be examined. Thereafter, the future of wireless networks, the IP Multimedia System (IMS), will be explored.

In chapter 3, Grid computing is discussed. Grid computing offers a collaboration platform by enabling access to distributed computation and data-storage facilities in (HIV/AIDS) VIOISs. Grid computing also enables data-sharing between the different service-providers. This chapter discusses how Grid computing has evolved from several non-interoperable distributed systems, to a seamless service-oriented, collaborative, and dynamic virtual environment, utilizing Web service standards. An elaboration is made on standard services needed to build service-oriented Grid applications, specified in the Open Grid Services Architecture (OGSA) standard (Foster *et al*., 2006), and also the implementation of these services, specified in the Web Service Resource Framework (WSRF) standard (Czajkowski *et al*., 2004$^2$).

.

Chapter 4 extends the concept of WSRF by describing an implementation of a Grid service. A Grid service is essentially made up of at least one Web service that provides a set of well-defined open standards-based interfaces that follow specific conventions. This chapter discusses the implementation of the WRSF specification through the Globus Toolkit Version 4 (GT4). GT4 offers execution management, information services, data management and security services in a Grid environment.

In chapter 5, Data Warehousing, which provides data-storage for large collections of aggregated HIV/AIDS patient-data, while enabling informational or analytical processing, will be elaborated. The implementation of Data Warehousing in the Grid environment will also be discussed.

Chapter 6 proposes a Wireless, Grid-enabled Data Warehouse, that integrates wireless networks, Grid computing and Data Warehousing for collaborative HIV/AIDS research.

In chapter 7, the dissertation is concluded with a summary of the research. Possible further research is proposed for the future.

# Chapter 2
# Wireless Networks

## INTRODUCTION

In Chapter 1, the concept of an HIV/AIDS VIOIS was introduced. (HIV/AIDS) VIOISs enable subscribers (HIV/AIDS researchers) to perform complex data analysis on aggregated (HIV/AIDS) data in a Grid environment. This data is contributed by various (health) institutions (including private and public hospitals, clinics, hospices, private medical practitioners, etc). Both the submission of this data and access to information (processed data) is via the Internet. However, the lack of viable Internet connectivity infrastructure in rural areas (where some of the health institutions are located) makes this difficult. A wireless connectivity solution is thus proposed.

This chapter discusses wireless networks, and their evolution to the current 3G networks. The architecture of 3G networks will then be discussed, followed by the radio access technologies that are utilized. The 3GPP network elements will be explored, with a focus on the Core Network Packet-Switched (CN PS) Domain. The CN PS will be utilized to transport data packets from the various health institutions, in order to enable institutions in rural areas to contribute data. The way in which the packets are transported will be examined. Thereafter, the future of wireless networks, the IP Multimedia System (IMS), will be explored.

## 2.1 Wireless Networks

There has been a proliferation of mobile and wireless networks during the past few years (Lee, 2006). These networks are contrasted from one another by the word “generation”, for example, “second generation”, “third generation”, etc., as illustrated in Figure 2.1 below. This is fitting since there is a huge “generation gap” that exists between the network technologies. The network technologies are further explained on Table 2.1.

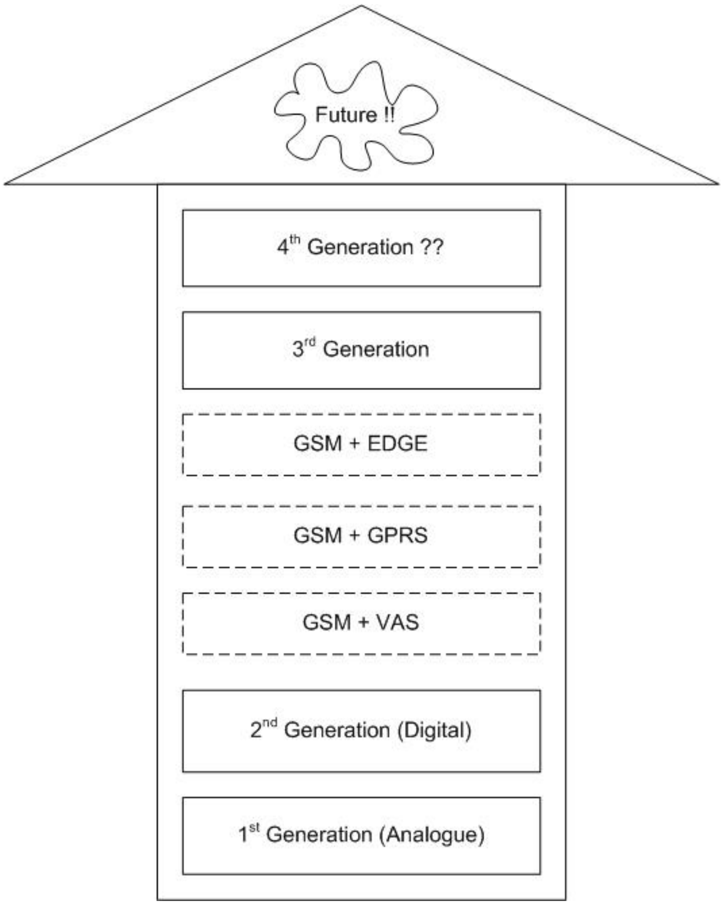

**Figure 2.1: Mobile Generations**

The first-generation mobile systems used analogue radio technologies and circuit-switched transmission techniques. The three main first-generation mobile systems were Advanced Mobile Phone System (AMPS) in North America, Total Access Communications Services (TACS) in the UK, and Nordic Mobile Telephone (NMT) in Nordic countries. Their use started in the early 1980s, and they provided primarily speech and related services. However, these services were limited, thus there was lack of security, quality and they were highly incompatible (Bannister, Mather & Coope, 2004).

The need for more services, such as mobile communication, and compatibility led to the creation of the second-generation mobile systems. They came into play by the late 1980s and were oriented towards offering a digital solution. Globalization of networks in order for them to be compatible, while offering better services, like data transfers and non-voice services, was what was expected of these networks (Bannister *et al*., 2004; Mishira, 2004). Although international standards bodies played a significant role on the evolution of these networks, the major players, like Japan, USA and Europe were using different standards. However, it must be noted that the Global System for Mobile communication (GSM) of the second-generation surpassed all the wireless network standards in accomplishing the commercial and technical expectations, for example, global roaming (Bannister *et al*., 2004).

Even though the standardization bodies anticipated one set of standards for global networks, this dream was not realized in the second-generation mobile systems. The third-generation mobile systems have thus been developed to fulfil this dream. Unlike second-generation networks, which were tailored towards voice traffic, the third-generation networks are designed to carry mainly data traffic (Mishira, 2004). Table 2.1 below, compares and contrasts the network elements and transmission speed and services provided, beginning from the second-generation to the present third-generation networks.

**Table 2.1: Network Elements and Transmission Speed and Services for Wireless Networks**

| Standard | Gene-ration | Elements | Transmission Speed & Services |
|---|---|---|---|
| Global System for Mobile Communication | 2G | **Base Station Subsystem (BSS)** with a Base Transceiver Station (BTS) and Base Station Controller (BSC)<br>**Network Switching Subsystem (NSS)** with a Mobile Switching Centre (MSC), Visitor Location Register (VLR), Home | 9.6 Kbps<br>Speech and data services |

| | | | |
|---|---|---|---|
| | | Location Register (HLR), Authentication Center (AC), and Equipment Identity Register (EIR). | |
| GSM and Value Added Services (VAS) | 2G | No additional elements but instead 2 platforms were added, namely the Voice Mail System (VMS) and Short Message Service Centre (SMSC) | 9.6 Kbps<br>Speech and data services |
| GSM and General Packet Radio Services (GPRS) | 2.5G | Additional elements to existing GSM system are Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). Plus Internet Protocol (IP) Routers, firewall servers and Domain Name System (DNS) servers. | 150 Kbps<br>Wireless Internet access |
| GSM and Enhanced Data rates in GSM Environment (EDGE) | 2.5G | No additional elements but dates rates are increased through sophisticated coding methods over the Internet. | 384 Kbps<br>Higher data rates |
| Universal Mobile Telecommunications Service (UMTS) | 3G | **Radio Access Network (RAN)** with a Base Station (BS) and a Radio Network Controller (RNC)<br>**Core Network (CN)** with WCDMA Mobile Switching | Up to 2 Mbps<br>Internet based services, video phoning, imaging |

| | | | |
|---|---|---|---|
| | | Center (WMSC) and VLR, HLR, Media Gateway (MGW), SGSN, and GGSN. | |

As illustrated in Figure 2.2 below, the UMTS standard is an evolution from the 2G GSM standard via GPRS and EDGE (Halonen, Romero & Melero, 2003). This evolution ensures that a bulk of the GSM network components can still be utilized on the UMTS network. This is a sound justification since as of August 2006, there were over 2 Billion GSM subscribers worldwide (GSM World, 2006), accounting for 82 % of the global mobile market. The UMTS standard is currently the most widely implemented 3G standard (Bannister *et al*., 2004).

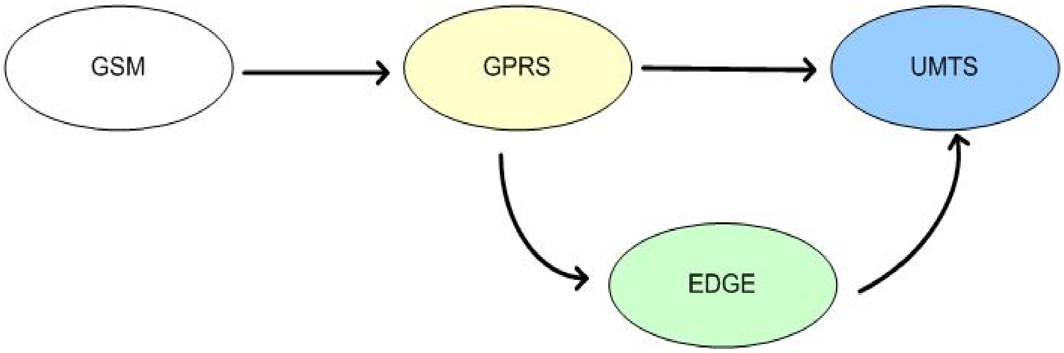

**Figure 2.2: Evolution to UMTS (Bannister *et al*., 2004)**

### 2.1.1 Multiple-Access Techniques

In any mobile network that is utilized by many subscribers, it is imperative to control allocation and usage of radio resources. In order to control the multiple simultaneous radio access allocation to requesters, multi-access techniques have been devised (Mishira, 2004). Their responsibility is to provide a solution to counter radio systems deficiencies, such as bandwidth limitation, interference from other users and multipath fading. Thus, they should present effective use of a frequency; whereas many simultaneous users can be accommodated in a fixed bandwidth within regulated parameters (Bannister *et al*., 2004; Kaaranen *et al*., 2005). The following sections discuss the various multi-access techniques that have developed.

### *2.1.1.1 Frequency-Division Multiple Access (FDMA)*

As the multiple access method widely used for the first-generation mobile systems, FDMA relies on splitting the frequencies amongst carriers (Mishira, 2004). This affords users an individual frequency or channel, or even a set of them (Kaaranen *et al.*, 2005). Figure 2.3 below, illustrates an example of FDMA.

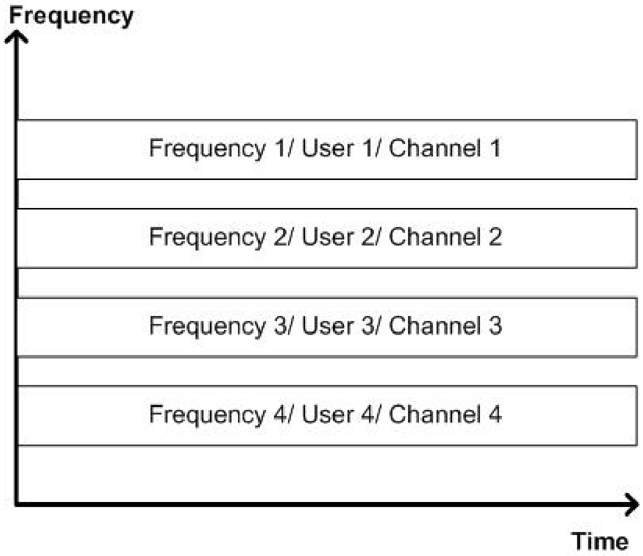
Frequency
Frequency 1/ User 1/ Channel 1
Frequency 2/ User 2/ Channel 2
Frequency 3/ User 3/ Channel 3
Frequency 4/ User 4/ Channel 4
Time

**Figure 2.3: Frequency-Division Multiple Access (FDMA) (Mishira, 2004)**

### *2.1.1.2 Time-Division Multiple Access (TDMA)*

The efficacy of FDMA on frequency utilization was found lacking as the second-generation mobile systems came into operation. Time-division multiple access proved to be the better solution as scores of users could utilize the same frequency. TDMA continuously partitions the frequency into small slices of time, known as time slots (Bannister *et al.*, 2004; Mishira, 2004). A time slot is repeated frequently in order to create an illusion of a continuous connection (see Figure 2.4). The number of users on a frequency is dependent on the time slots a frequency offers (GSM allows a maximum of eight time slots) (Kaaranen *et al.*, 2005, p. 40).

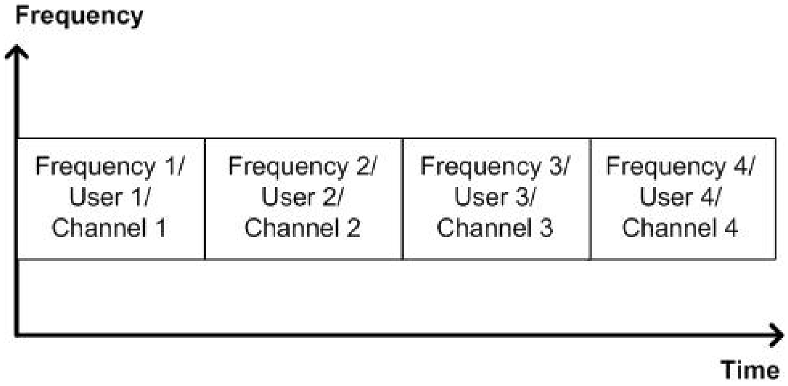
Frequency
Frequency 1/ User 1/ Channel 1
Frequency 2/ User 2/ Channel 2
Frequency 3/ User 3/ Channel 3
Frequency 4/ User 4/ Channel 4
Time

**Figure 2.4: Time-Division Multiple Access (TDMA) (Mishira, 2004)**

### *2.1.1.3 Code-Division Multiple Access (CDMA)*

CDMA employs the spread spectrum technique to coalesce multiple access and modulation in order to attain information confidentiality and efficiency. CDMA allocates users different code(s) as opposed to frequency or time slot, as in FDMA and TDMA (Mishira, 2004). All the users utilize the same frequency band simultaneously; hence (Kaaranen *et al*., 2005, p. 41):

- The power required to spread low-bit rate transaction is relatively minimal as depicted by the thin layer on Figure 2.5
- On the other hand, the power required to spread high bit rate transaction is higher, as shown by the thick layer on Figure 2.5.

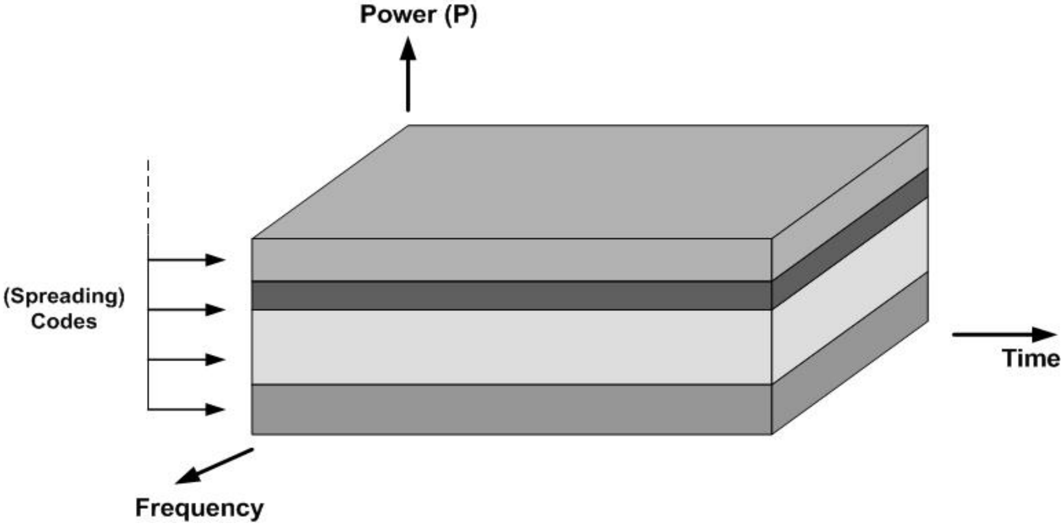

**Figure 2.5: Code-Division Multiple Access (CDMA) (Kaaranen *et al*., 2005)**

3G networks are built on existing 2G network infrastructures. The following sections discuss the evolution from the 2G networks to 3G networks.

## 2.2 2G Networks

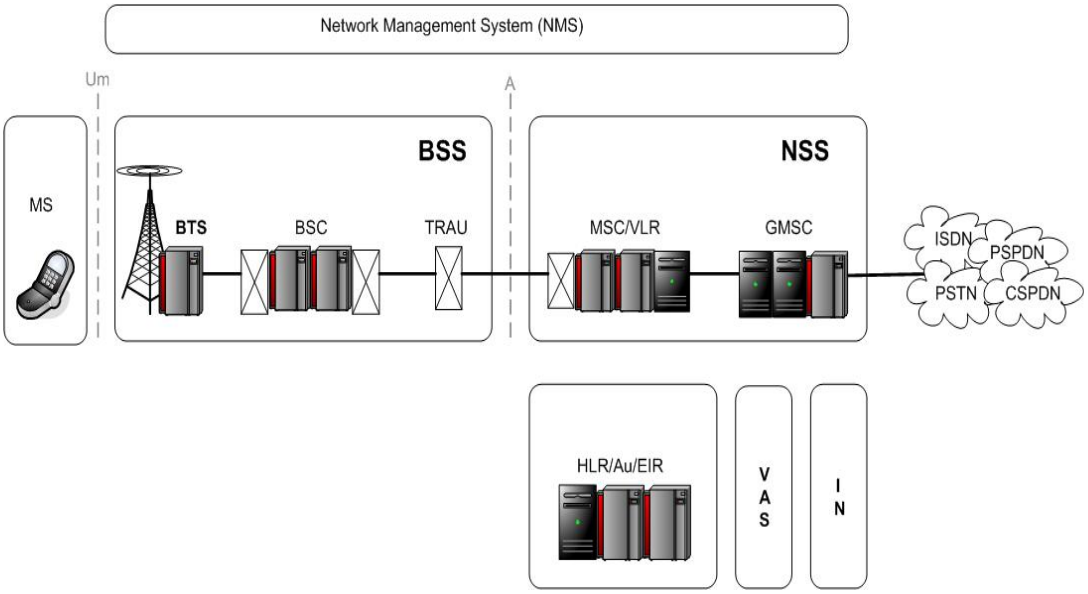

**Figure 2.6: GSM Network with Value Added Services (Kaaranen *et al*., 2005)**

On the GSM network (depicted in Figure 2.6 above), the Base Station Subsystem (BSS) takes care of the radio path control, while the Network Switching Subsystem (NSS) handles the call control functions (Halonen *et al*, 2003). The NMS is responsible for the overall operation and maintenance of the network. The *Um* interface, located between the MS and BSS, is responsible for providing circuit and packet data services to the MS. The *A* interface, which rests between the BSS and NSS, handles traffic and signalling (Mishira, 2004).

The Mobile Equipment (ME) and Service Identity Module (SIM) make up the Mobile Station (MS). Although SIM officially stands for Subscriber Identity Module, the use of Service Identity Module better defines its functionality (Chen & Zhang, 2004).

The Base Station Controller (BSC) maintains the radio connections made towards the MS and the terrestrial connections en route to the NSS. The Base Transceiver Station (BTS) maintains the air interface signalling, ciphering and speech processing. The Transcoding and Rate Adaptation Unit (TRAU) handles speech trancoding; thus, it enables conversion of speech format from one digital coding to another (Kaaranen *et al*., 2005).

The Mobile Services Switching Center (MSC) takes care of call control, BSS control functions, interworking functions, charging, statistics and interface signalling towards BSS. The MSC also interfaces with external networks, such as Packet Switched Public Data Network (PSPDN), Circuit Switched Public Data Network (CSPDN), Public Switched Telephone Networks (PSTN) and Integrated Services Digital Network (ISDN) (Chen & Zhang, 2004). The MSC consists of the MSC/Visitor Location Register (MSC/VLR) (which maintains the BSS connections, mobility management and interworking) and the Gateway MSC (GMSC) (which takes part in communication management, mobility management and connecting with other networks) (Mishira, 2004; Kaaranen *et al*., 2005).

The Home Location Register’s (HLR) primary functions are subscriber data and service handling, statistics and mobility management (Lee, 2006). The HLR permanently saves all the subscriber information. The Authentication Centre (AuC) deals with subscriber-identity-related security information. The Equipment Identity Register (EIR) handles mobile-equipment-identity-(hardware)-related information. The AuC and EIR perform their functions in collaboration with the VLR (Kaaranen *et al*., 2005).

## 2.3 Network Evolution

### 2.3.1 From Digital to Value Added Services

In order to develop the basic GSM, it was imperative to add service nodes and service centres on top of the existing network infrastructure (Wisely, Eardley & Burness, 2002). These service nodes and centres are commonly referred to as Value

Added Service (VAS) platforms and are mainly equipment added to the network, as illustrated in Figure 2.6. The minimum equipment requirement for a VAS platform consists of: the Short Message Service Centre (SMSC) and the Voice Mail System (VMS) (Bannister *et al.*, 2004). From a technical point of view, a VAS platform should be relatively simple while providing a certain type of service. It utilizes standard interfaces towards the GSM network and may or may not have external interfaces towards other networks. From a service evolution perspective, VAS is the first step towards using services to generate revenue and partially tailoring them. SMS has been the great success story as it has become extremely popular among GSM subscribers (Chen & Zhang, 2004).

Although VAS platform added services to the basic GSM, they were not tailored towards individuals. In order to achieve a more individual type of service, the Intelligent Network (IN) concept was introduced into the GSM network (Kaaranen *et al.*, 2005). Technically speaking, this implies major changes to the switching network elements in order to add the IN functionality; furthermore the IN platform is quite a complex entity in itself. From a service evolution point of view, this means major steps towards individuality; moreover, it enables the operator to carry out business in a more secure way (for instance, prepaid subscriptions are mostly implemented using the IN platform) (Chen & Zhang, 2004). The IN technology does not address all the mobile requirements, as it is deeply rooted in Public Switched Telephone Networks (PSTNs). To meet the deficiencies of IN technology, it has been enhanced with Customised Applications for Mobile network Enhanced Logic (CAMEL) (3GPP, 2006[8]).

### 2.3.2 Need for Higher Speeds and Introduction of the Packet World

The Internet and electronic messaging have triggered anticipation for their incorporation into the mobile world, and thus, increased data transfers on mobiles. Although this might have been overlooked when the GSM system was being specified, there have been several enhancements. Firstly, channel coding is optimised, and thus, the effective bit rate has been increased from 9.6 kbps up to 14 kbps (Lee, 2006). Secondly, in order to transmit more data across the air interface, several traffic channels can be utilised instead of one. This is referred to as ‘High Speed Circuit Switched Data’ (HSCSD) (Kaaranen *et al*., 2005). Optimally, an HSCSD user can reach data transfer rates of 40-50 kbps. Although this might seem to be a straightforward solution technically, it wastes resources, and thus, will not go down well with end-users, due to the pricing policy (i.e., the operators set the price for HSCSD use). Moreover, there is the issue of data traffic being asymmetric in nature; thus, a very low data rate is utilized from the terminal towards the network direction (uplink) and higher data rates are utilized in the opposite direction (downlink) (Bannister *et al*., 2004).

Due to the CS symmetric Um interface not offering the best possible access media for data connections, and also a great majority of data traffic being packet-switched (PS) in nature, there was need for an ‘upgrade’ to the GSM to address the situation. To enhance the network, General Packet Radio Service (GPRS) was introduced (3GPP, 2006[4]). GPRS introduced two additional mobile-network-specific service nodes: Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). These nodes enable the MS to form a PS connection with external packet data networks (e.g. the Internet) through the GSM network. Although GPRS is not a complete Internet Protocol (IP) mobility solution, it brings the Internet and IP mobility closer to the mobile subscriber (Halonen *et al*., 2003). From the service perspective, GPRS converts traditional CS services, which were suited for PS connections, to be used over GPRS (e.g., Wireless Application Protocol (WAP)) (McGuiggan, 2004).

### 2.3.3 Evolution of Mobile Data and Multimedia Services

*Text-Based Instant Messaging* – Short Message Service (SMS) is the first mobile data service to be successful globally. With SMS, a mobile user can instantly send and receive text-based short messages (up to 160 characters). Ironically, SMS does not utilize a packet core network, but rather employs the Mobile Application Part, which is a signaling protocol originally designed for mobility support in GSM networks (Wisely *et al*., 2002). This implies that SMS services were operational on circuit-switched networks, which went before the advent of packet core networks (Lee, 2006).

*Low-Speed Mobile Internet Services* - Following the success of SMS services, there was a need for more interactive and information-based data services. These advanced data services were offered when the mobile Internet came to being (Smith & Collins, 2002). Mobile operator NTT DoCoMo, from Japan, had a successful mobile Internet application called i-Mode, which was deployed over its Personal Digital Cellular (PDC) network. The i-Mode affords users email services (e.g., send, receive, instant messaging), directory services (e.g., phone directory, dictionary), commercial transaction services (e.g., banking, stock trading, and ticket reservation), entertainment services (e.g., network games, karaoke) and daily information services (e.g., weather reports, traffic reports, news) (Chen & Zhang, 2004).

*High-Speed and Multimedia Mobile Internet Services* - The integration of IP technologies into wireless networks affords users advanced new ways of communicating (Wisely *et al*., 2002). Users are no longer limited to talking on their phones and sending instant text-based messages; they can record videos, exchange pictures, receive location-dependent services and play sophisticated games with remote users in real time. Some of the multimedia applications and advanced communication modes include (Lee, 2006):

- Camera Phones: Users can capture still pictures, record short videos with sound, and exchange these videos and pictures through email or multimedia messaging using mobile phones that have cameras (Holma & Toskala, 2004).
- Multimedia Messaging Services (MMS): Allows users to send and receive multimedia messages (e.g., voice, music, videos, photos, data, etc.) (Nicopolitidis *et al*., 2003).

- Networked gaming: Users can download games that they can play individually or with other remote users in real time (Halonen *et al*., 2003).
- Location-Based Services: Real-time navigation services, maps, interesting spots, tourist destinations and hospitality locations can be accessed through mobile devices (e.g., shopping malls, cinemas, restaurants, hotel, hospitals, etc.) (Holma & Toskala, 2004)
- Streaming videos to mobile devices: Users can view real-time and non-real-time videos on their mobile devices (e.g., downloaded videos, television, movie trailers) (Kaaranen *et al*., 2005; Holma & Toskala, 2004).
- Vehicle information systems: Users can surf the Internet or access enterprise networks while on moving vehicles (e.g., trains, cars, airplanes) akin to when they are in their offices or homes. This means they will be able to access their corporate networks, obtain real-time weather information, traffic information, download and play games, etc. (Halonen *et al*., 2003).

## 2.4 3G Networks

As stated by Chen & Zhang (2004) and Lee (2006), the third-generation mobile systems have been designed to:

- Offer more radio system capacities over 2G networks and per-user data rates: A user moving at a vehicle speed will have data rates of up to 144 Kbps, a pedestrian user will have data rates of up to 384 kbps, and a stationary user will get up to 2Mbps.
- Provide IP-based data, voice and multimedia services support: integration between 3G networks and the Internet will afford mobile users the ability to seamlessly access the plethora of resources and applications available on the Internet.
- Significantly improve support for Quality of Service (QoS): Enhanced QoS support and performance is required to handle multiple sets of services, such as streaming video, real-time voice, best-effort data and non-real-time video (Holma & Toskala, 2004).

- Achieve more interoperability: Facilitate better roaming among diverse radio access technologies, different network providers, and different countries.

There are two international partnerships (depicted in Figure 2.7) that define 3G wireless network standards applying different approaches (Nicopolitidis *et al*., 2003; Bannister *et al*., 2004; Chen & Zhang, 2004; Holma & Toskala, 2004).

*Third-Generation Partnership Project* (3GPP): 3GPP aims to produce globally applicable standards for third-generation mobile systems based on evolved GSM core networks and the radio access technologies that are supported by these evolved GSM core networks. Thus,

- The GSM core network platform will be evolved to support circuit-switched mobile services, while the GPRS core network platform will be evolved to support packet-switched services.
- 3G radio access technologies will be based on Universal Terrestrial Radio Access Networks (UTRANs) that utilize Wideband Code-Division Multiple Access (WCDMA) radio technologies.

*Third-Generation Partnership Project 2* (3GPP2): 3GPP2 aims to produce globally applicable standards for third-generation mobile system based on evolved IS-41 core networks. Thus,

- The IS-41 core network will be evolved to support circuit-switched mobile services and a new packet core network architecture will be defined to support IP services.
- 3G radio access technologies will be based on CDMA2000 radio technologies.

3G Networks in South Africa evolved from GSM core networks and thus discussions in this dissertation will be from a 3GPP perspective.

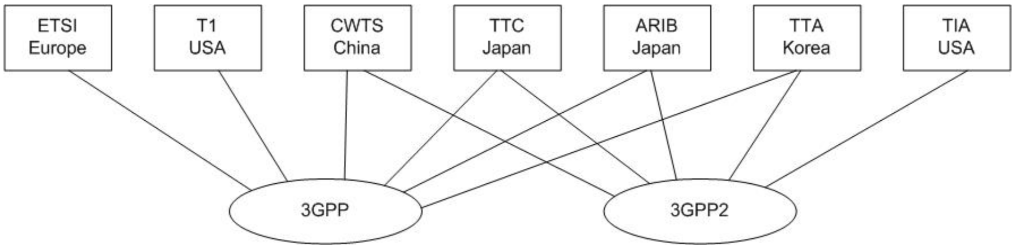

**Figure 2.7: 3GPP, 3GPP2 and the Standardization Bodies**

ETSI – European Telecommunication Standard Institute –Europe
ARIB –Association of Radio Industries and Business – Japan
CWTS –China Wireless Telecommunication Standard group – China
T1 –Standardization Committee T1- Telecommunications – United States
TTA – Telecommunication Technology Association –Korea
TTC – Telecommunication Technology Committee –Japan
TIA – Telecommunication Industry Association - United States

### 2.4.1 3G Network Architecture

Although a 3G network structure can be modelled in different ways, there are architectural approaches that delineate the structure of a basic 3G network. These approaches define a universal infrastructure that allows for evolution to new technologies and service changes on the current network structure, whilst still offering existing services (Lee, 2006). To realize this infrastructure, it is imperative that the transport technology, access technology, service technology and user applications are separated from each other (Kaaranen *et al.*, 2005). This can be achieved using:

- The Conceptual Network Model
- Structural Network Architecture
- Resource Management Architecture
- UMTS Bearer Architecture.

#### *2.4.1.1 Conceptual Network Model*

The conceptual network model divides the architecture according to the nature of traffic, protocol structures and physical elements. To cater for the traffic, the 3G network makes use of the packet-switched (PS) and circuit-switched (CS) domains. A domain refers to the highest-level group of physical entities and the reference points (defined interfaces) between such domains (3GPP, 2006[7]).

As for the protocol structure, the 3G network is divided into the access stratum and the non-access stratum. A stratum refers to the way of grouping of protocols, related to one aspect of the services provided by one or several domains (3GPP, 2006[7]).

#### *2.4.1.2 Structural Network Architecture*

This dissertation essentially presents issues from the structural network architecture point of view. The 3G network utilizes the GSM network as its basis for building, and thus, reuses everything that is realistic.

The structural network architecture is made up of a variety of network elements that are grouped according to their functionality or sub-network. The network elements that execute radio-related functions are grouped into the Radio Access Network (RAN), and in UMTS, with WDCMA access referred to as UMTS Terrestrial RAN (UTRAN) (Kaaranen *et al*., 2005, p. 9). These elements consist of Radio Network Subsystem (RNS), Base Station (BS, known as Node B), and the Radio Network Controller (RNC). The main function of UTRAN is to create and maintain Radio Access Bearers (RABs) that support communication between the Core Network (CN) and the User Equipment (UE) (Lee, 2006). The switching of calls and routing of data connections to external networks is handled by the Core Network (CN) (Chen & Zhang, 2004). The CN is made up of the circuit-switched domain and packet-switched domain. The User Equipment (UE) completes the architecture by defining an interface between the user and the radio interface.

The UE is made up of (Smith & Collins, 2002):

- The Mobile Equipment (ME), which is a radio terminal utilized for radio communications over the Uu interface.
- The UMTS Subscriber Identity Module (USIM), which is a smartcard that possesses the identity of the subscriber, executes authentication algorithms, and stores authentication keys, encryption keys and essential subscription information on the terminal.

#### *2.4.1.3 Resource Management Architecture*

The Resource Management Architecture separates the network responsibilities and functional elements into (Kaaranen *et al.*, 2005, p. 12):

- *Communication Management (CM)* - deals with function and procedures that are concerned with managing user connections, for example, call handling during CS connections, management of sessions in PS connections, in addition to managing supplementary services and short-message services (Smith & Collins, 2002). The CM is supported by Communication Control (COMC), which executes control duties, such as call control and packet session control.
- *Mobility Management (MM)* - deals with function and procedures that are concerned with enabling mobility and ensuring security, for instance, location update procedures and procedures for securing connections. The Mobility Control (MOBC) supports the MM through control mechanisms, such as execution control for security and location updates.
- *Radio Resource Management (RRM)* - is an assortment of algorithms utilized by UTRAN to manage radio resources, such as controlling the power of radio connections, administering various handovers, system load and admission control (Lee, 2006). The RRM utilizes the Radio Resource Control (RRC), which is radio control protocol that carries out radio-link establishment duties and maintenance between the UE and UTRAN .

### *2.4.1.4 Bearer Architecture*

The 3G network provides end-users and their applications with an infrastructure with facilities, sufficient bandwidth and connection quality. This provision of facilities, allocation of bandwidth and provision of quality connection collectively is referred to as Quality of Service (QoS) (Bannister *et al*., 2004). The different elements of a 3G network must contribute to fulfilling QoS requirements from end users.

The end-to-end service (QoS) requirements have been decomposed into: the local bearer service, the UMTS bearer service and the external bearer service, as illustrated in Figure 2.8 below. The local bearer service consists of methods that define the mapping of an end-user service between the terminal equipment and Mobile Termination (MT). The MT is responsible for terminating UE’s radio transmission to and from the network and also adapts capabilities of the terminal equipment to those of radio transmission. Successively, the UMTS bearer service consists of methods that designate QoS over the UMTS/3G network, made up of UTRAN and CN. In order to address the QoS requirements towards external networks the external bearer service is utilized (Kaaranen *et al*., 2005, p. 13).

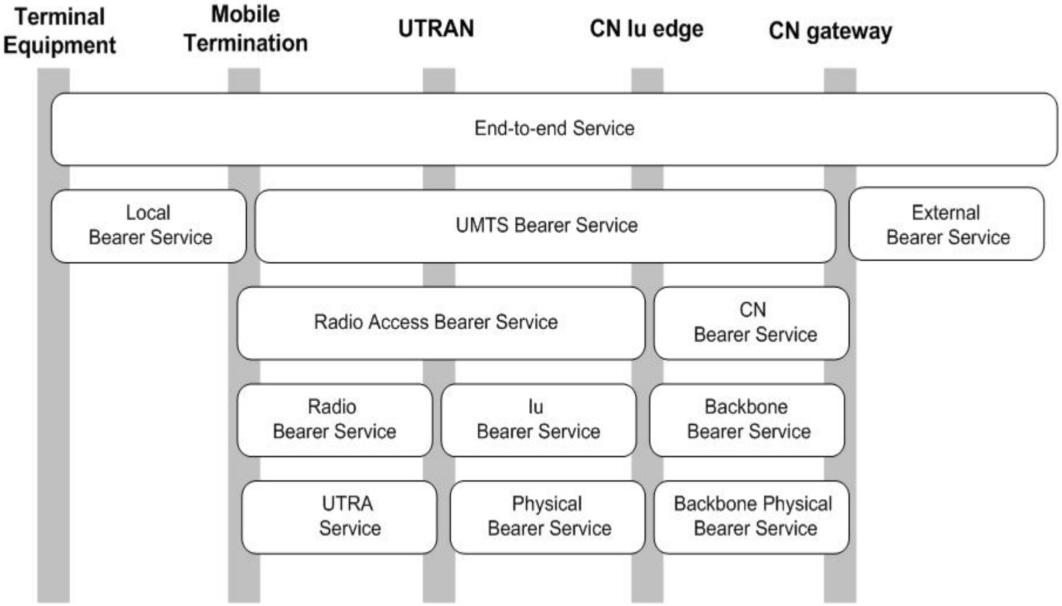

**Figure 2.8: Bearer Architecture in UMTS (Kaaranen *et al*., 2005).**

### 2.4.2 Universal Mobile Telecommunications Service (UMTS)

Universal Mobile Telecommunications Service (UMTS) is not a solitary access technology, but is rather a combination of selected access technologies. The envisaged goal is to afford users sufficient coverage at all times and ample platforms and spectrums for services using these technologies (Nicopolitidis *et al*., 2003).

The three main access technologies are (Lee, 2006; Kaaranen *et al*., 2005):

- Wideband Code Division Multiple Access (WCDMA) – UMTS utilizes the WCDMA Frequency Division Duplex (WCDMA-FDD), and data rates in the downlink direction are enhanced through High Speed Downlink Packet Access (HSDPA).
- Global System for Mobile Communication/Enhanced Data for GSM Evolution (GSM/EDGE) (Halonen *et al*., 2003).
- Complementary Accesses.

WCDMA is currently the dominant radio access technology, and will be discussed in the next section.

### 2.4.3 Wideband Code Division Multiple Access (WCDMA)

WCDMA is a wideband Direct-Sequence Code Division Multiple Access (DS-CDMA) system. User data is spread over a wide bandwidth by being multiplied with the quasi-random bits (known as chips) obtained from CDMA spreading codes. Variable spreading factor and multicode connections are utilized in order to attain very high bit rates of up to 2Mbps (Holma & Toskala, 2004, p. 47). A carrier bandwidth of approximately 5 MHz is achieved through a 3.84 Megachips per second (Mcps) chip rate. This wide carrier bandwidth affords users high data rates and performance benefits, for example multipath diversity.

The WCDMA physical layer structure is made up of radio frames, which have durations of 10 ms, and are separated into 15 slots, as depicted in Figure 2.9. A radio frame is used to transmit blocks of data across the air interface, and the radio frame granularity determines the data rate that information will be sent at (Bannister *et al*., 2004).

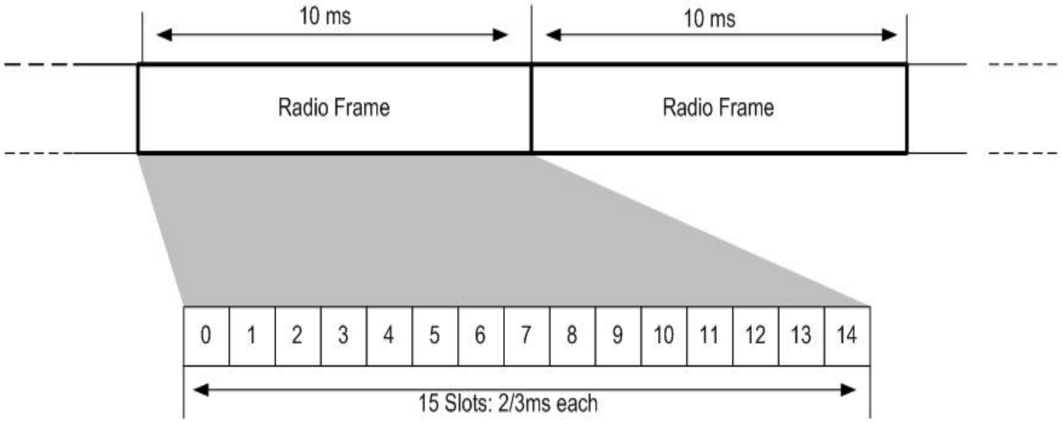

**Figure 2.9: UMTS Radio Frame Structure (Bannister *et al*., 2004)**

*2.4.3.1 WCDMA Radio Channels*

A UMTS channel refers to the bandwidth allocated to an end-user by WCDMA and its controlling functions. The types of channels needed and their organization is defined through WCDMA's functional implementation. The implementation is made up of three layers: logical channels, transport channels, and physical channels (see Figure 2.10 below) (Lee, 2006; Kaaranen *et al*., 2005; Bannister *et al*., 2004).

A logical channel consists of information streams that are concerned with transferring information of a particular type over the radio interface. The transport channel, on the other hand, is dedicated to transporting the logical channels between the UE and RNC. The physical channel, which is defined through a WCDMA code and frequency, provides the radio platform across the air interface through which the information is actually transferred (Kaaranen *et al*., 2005, p. 66).

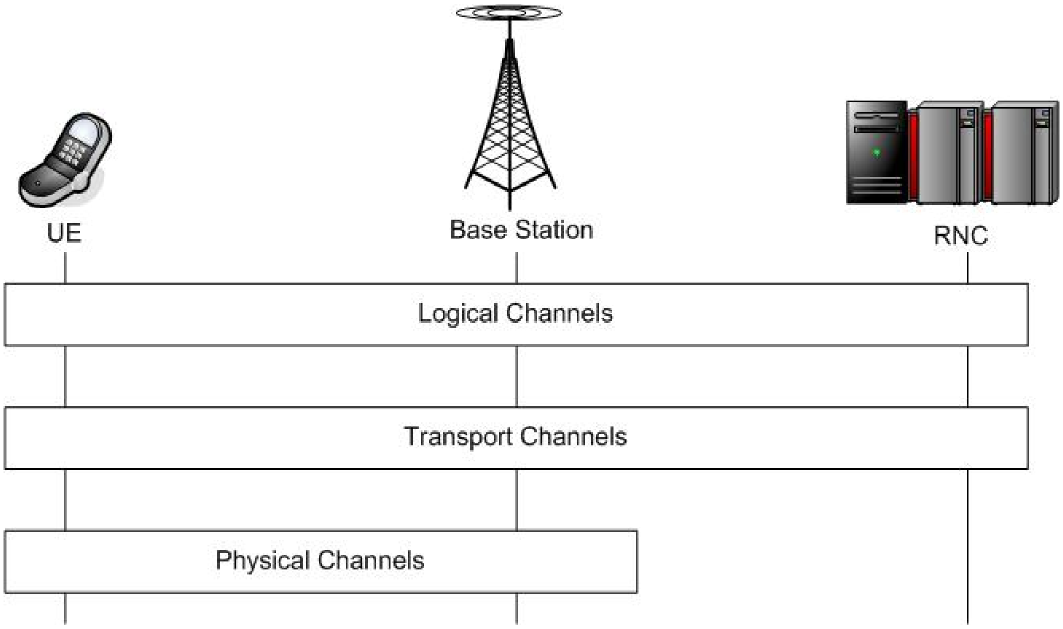

**Figure 2.10: UMTS Channel Structure (Based on Bannister *et al*., 2004)**

#### *2.4.3.2 Logical Channels*

Unless otherwise stated, the information below is sourced from Lee (2006), Kaaranen *et al*. (2005) and Bannister *et al*. (2004).

*Broadcast Control Channel (BCCH)* – downlink channel utilized for transmitting general system information between UE and the 3G network

*Paging Control Channel (PCCH)* – downlink channel utilized by the network to carry paging information to the user about a pending communication request.

*Common Traffic Channel (CTCH)* – downlink channel that transmits dedicated user information to a group of specified UEs.

*Common Control Channel (CCCH)* – both uplink and downlink channel utilized for carrying control information from the network to UEs that do not have any dedicated channels. To differentiate the UEs, they are identified using UMTS Terrestrial Access Network (UTRAN) Radio Network Temporary Identity (U-RNTI).

*Dedicated Traffic Channel (DTCH)* – bidirectional, point-to-point channel, dedicated to a single UE, and utilized for the transfer of user information.

*Dedicated Control Channel (DCCH)* – bidirectional, point-to-point channel, utilized for the carrying dedicated control information between the network and a UE.

#### *2.4.3.3 Downlink transport and physical channels*

The BCCH is transported by the Broadcast Channel (BCH) (Lee, 2006). This is then transmitted in the Primary Common Control Physical Channel (PCCPH). The physical layer coding (WCDMA channelization code) utilized is a system constant, in order for to enable all UEs to decode information on the BCH (Kaaranen *et al*., 2005). After being decoded, the BCCH will possess system information that specifies the coding of all other channels present. The PCCH is transported by the Paging Channel (PCH), which is transmitted in the Secondary Common Control Physical Channel (SCCPH) (Lee, 2006). Furthermore the SCCPH also carries the Forward Access Channel (FACH). The FACH is capable of transporting a number of logical channels transporting common (CCCH, CTCH) and dedicated (DTCH, DCCH) control and traffic. Alternatively, dedicated traffic and control can be transported on a Dedicated Channel (DCH) or on a Downlink Shared Channel (DSCH), where multiple user channels can be multiplexed into a single channel (Bannister *et al*., 2004).

#### *2.4.3.4 Uplink transport and physical channels*

Dedicated traffic and common and dedicated control information are transported by the Random Access Channel (RACH) (Lee, 2006). This is transmitted in the Physical RACH (PRACH). The Dedicated Channel (DCH) is similar to the downlink; only differing at the physical layer, where instead of being transmitted in a single multiplexed channel, it is transmitted in two separate channels. Data traffic can also be carried on the Common Packet Channel (CPCH), although it was specifically designed for transporting packet data. This is transmitted in the Physical Common Packet Channel (PCPCH) (Bannister *et al*., 2004).

## 2.5 3GPP Network Architecture

A public network that is administrated by a single network operator and provides land mobile services is known as a Public Land Mobile Network (PLMN). Figure 2.11 below illustrates the conceptual architecture of a 3GPP PLMN. It comprises of one or more Radio Access Networks (RANs) which are interlinked through a Core Network (CN) (Lee, 2006, p. 254).

A RAN provides radio resources (such as radio channels, bandwidth) for users to access the CN. 3GPP Release 5 supports GSM/EDGE RAN (GERAN) and UMTS Terrestrial RAN (UTRAN) (Chen & Zhang, 2004, p. 34).

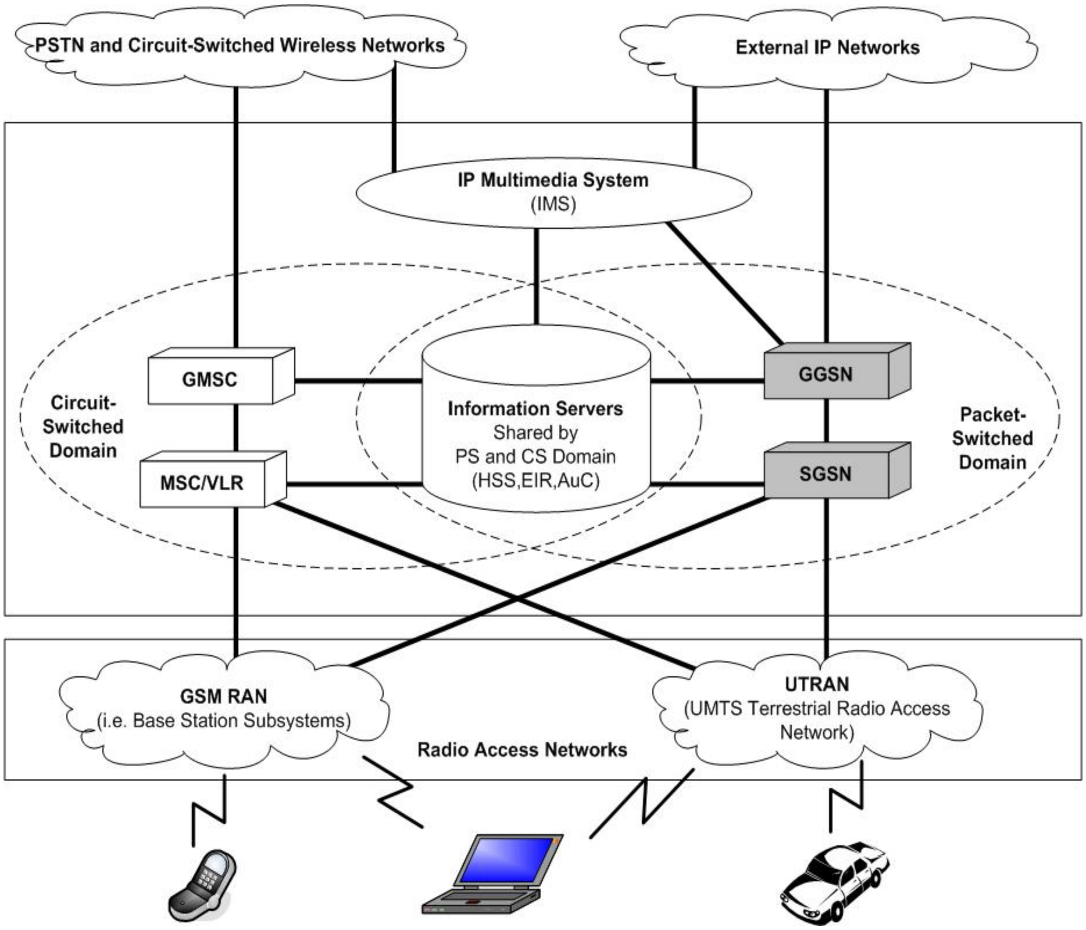

**Figure 2.11: 3GPP Conceptual Network Architecture (Release 5)**
**(3GPP, 2006$^2$)**

### 2.5.1 Core Network

The Core Network (CN) implements the mechanisms for maintaining both packet-switched and circuit-switched communication services to end-users (Lee, 2006). The communication services comprise both basic and advanced services. Basic circuit-switched communication services supported include voice and data call switching and call control functions to maintain basic point-to-point circuit-switched calls, while basic packet-switched services include the routing and transport of user IP packets. Advanced (value-added or supplementary) circuit-switched services include toll-free calls, prepaid calls, multiparty communications and call forwarding, while advanced packet-switched services include location-based services, World Wide Web, multimedia messaging services, email and networked gaming (Chen & Zhang, 2004).

The CN is made up of the following functional blocks (3GPP, 2006[1]; 3GPP, 2006[2])

- Circuit Switched (CS) Domain
- Packet Switched (PS) Domain
- IP Multimedia Subsystem (IMS)
- Information Servers.

The User Equipment (UE) is made up of the Mobile Equipment (ME) and UMTS Subscriber Identity Module (USIM) (3GPP, 2005). USIM is analogous to, and thus has been developed based on GSM's Subscriber Identity Module (SIM).The ME, which is made up of Mobile Termination (MT) and Terminal Equipment (TE), is utilized by end users to access services on the network. The TE enables access protocols operation while the MT maintains radio transmission and channel management. An MT is identified through a globally unique International Mobile Station Equipment Identity (IMEI) (3GPP, 2006[3]).

A mobile device can be set up to access the CS domain only, the PS domain only, or both the CS and PS domains (Chen & Zhang, 2004, p. 36). In order to access services from these domains, the subscribers are assigned a globally uniquely identifier, known as the International Mobile Subscriber Identity (IMSI). An IMSI is stored in the USIM, and the USIM can be moved and used on different mobile devices whilst the subscriber identity remains the same. An IMSI can contain only numerical values 0 to 9 (Eberspacher *et al.*, 2001). It consists of three parts as illustrated in Figure 2.12 below

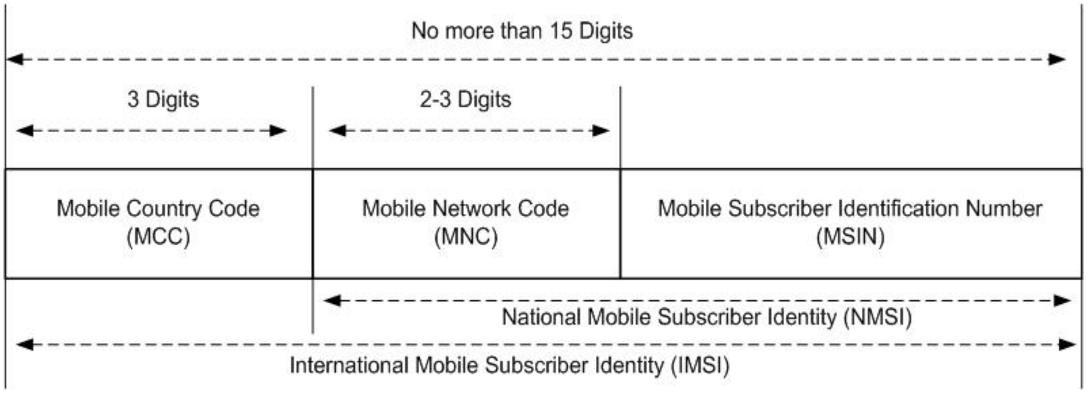
No more than 15 Digits
3 Digits
2-3 Digits
Mobile Country Code (MCC)
Mobile Network Code (MNC)
Mobile Subscriber Identification Number (MSIN)
National Mobile Subscriber Identity (NMSI)
International Mobile Subscriber Identity (IMSI)

**Figure 2.12: Structure of IMSI (Based on 3GPP, 2006[3])**

- Mobile Country Code (MCC) – uniquely identifies the mobile subscriber’s home country.
- Mobile Network Code – is made up of two or three digits depending on the value of the MCC.
- Mobile Subscriber Identification Number (MSIN) – uniquely identifies a mobile subscriber within a Public Land Mobile Network (PLMN).

The ITU-T administers the allocation of MCCs according to the ITU-T Blue Book Recommendation E.212. The combination of the MNC and MSIN is known as the National Mobile Subscriber Identity (NMSI). The allocation of NMSIs is controlled by the numbering administrations in each country. A unique MNC is assigned to each PLMN, if there is more than one PLMN in a country.

To minimize the need of transmitting the IMSI, which uniquely identifies a mobile subscriber, over the air, 3GPP utilizes a Temporary Mobile Subscriber Identity (TMSI) to identify the mobile whenever possible (3GPP, 2006[3]). A TMSI is a four-octet number assigned to a mobile temporarily by an MSC/VLR for circuit-switched services or by an SGSN for packet-switched services. The two most important bits in a TMSI specify whether the TMSI is for packet-switched services. A TMSI for packet-switched services is known as Packet TMSI or P-TMSI. An MSC or SGSN utilizes a TMSI to uniquely identify a mobile. The TMSI will only be allocated in ciphered form. Moreover, there will be measures undertaken to make sure that the mapping between a mobile's IMSI and TMSI are only known by the mobile and the network node (SGSN or MSC) that assigned the TMSI. A mobile's TMSI, as opposed to its IMSI, will then be utilized as the mobile's identity whenever possible in signalling transmitted messages over the air. The security impact of transmitting unencrypted TMSI over the air is lower than transmitting unencrypted TMSI, because only the mobile and the MSC or SGSN that assigned it the TMSI know the mapping between a mobile's IMSI and TMSI, and the TMSI is only valid if the user is served by the MSC or SGSN that assigned it (Eberspacher *et al*., 2001).

In order to send and receive IP packets over the PS CN, a mobile must also be configured with at least one IP address. The mobile might utilize multiple IP addresses simultaneously. However, it is not necessary for a mobile to have a valid IP address at all times when it is attached to the PS domain. Alternatively, the mobile can acquire an IP address only when it needs to activate packet data services over the PS CN (Lee, 2006).

### 2.5.2 Circuit-Switched Core Network Domain

The CS CN domain is built on the GSM core network technologies and consists of all the CN entities that provide circuit-switched voice and data services. The major network entities are (Lee, 2006, pp. 250 - 251; 3GPP, 2006[1]; 3GPP, 2006[2]):

- Mobile-services Switching Center (MSC)
- Gateway MSC
- Visitor Location Register (VLR)
- Home Subscriber Server (HSS), Equipment Identity Register (EIR), and Authentication Center (AuC).

The MSC carries out call control and switching functions while executing mobility management functions, such as location registration and handoff functions for mobile devices. The MSC interlinks RANs with the CS CN domain, for instance, an MSC can be interconnected to multiple GSM BSSs or UTRAN RNSs (Bates, 2002).

The 3GPP Release 5 made a major development on the MSC by separating and implementing its call control and switching functions on disparate network entities (3GPP, 2006[1]). Call control and mobility management are performed on the MSC Server while circuit switching, media conversion payload are processed on the CS Media Gateway (CS-MGW). The separation of these functions enables them to evolve independently while facilitating increase of the network scalability.

To interconnect with external circuit-switched networks, a dedicated MSC, known as the Gateway MSC (GMSC), is utilized. AGMSC ensures that a circuit-switched call is routed to its final destination in the external network. The GMSC call-control functions and switching can also be separated and implemented on a GMSC server and a CS-MGW (Chen & Zhang, 2004).

The VLR is responsible for temporarily maintaining a visiting mobile's location and service subscription whilst it is inside the network part controlled by the VLR. The VLR keeps track and exchanges a mobile's current location information with the mobile's HLR and retrieves the visiting mobile's service subscription information from the HLR in order to provide service control to the mobile. Signalling between the VLR and the HLR is performed through the Mobile Application Part (MAP) protocol (3GPP, 2006[6]).

### 2.5.3 Packet-Switched Core Network Domain

To support packet-switched services, the PS CN domain provides the following main functions (3GPP, 2006[2]):

- *Network Access Control* – Determines which mobiles are permitted to operate on the PS domain. Functions performed include registration, admission control, authentication and authorization, message filtering, and usage data collection.
- *Packet routing and transport* – responsible for routing user packets towards their destination either inside the same PLMN or to external networks.
- *Mobility management* – Provides network-layer mobility management functions such as tracking a mobile's location, initiating paging in order to determine the precise location of an idle mobile in order to send data to the mobile, and maintaining up-to-date CN routes to mobiles when they are in motion.

The PS domain is built on the GPRS network platform and consists of two main types of network nodes (3GPP, 2006[4]):

- Serving GPRS Support Node (SGSN)
- Gateway GPRS Support Node (GGSN).

An SGSN interlinks a single or multiple RANs to a PS CN. A SGSN is responsible for carrying out the following specific functions (Bates, 2002):

- *Access Control* – The SGSN provides the first line of control over the users' access to the PS CN domain. The GGSN furthermore provides an additional line of control.
- *Location Management* – The SGSN keeps track of the locations of mobiles that utilize packet-switched services. The location information can be reported to the HLR, and it can be utilized by the GGSN to carry out network-initiated procedures to set up mobile connections.
- *Route Management* – The SGSN is responsible for the maintenance of individual routes for mobiles to a GGSN and relaying traffic between the mobile and the GGSN.
- *Paging* – The SGSN is responsible for initiating paging operations after receiving user data that is destined to idle mobiles.

- *Interface with service control platforms* – The SGSN acts as the contact point with Customized Applications for Mobile Enhanced Logic (CAMEL) (3GPP, 2006[8]) functions and IP-Based services. CAMEL is a set of protocols and procedures that allow a network operator to provide operator-specific services to its subscribers, even when the subscribers are currently in foreign networks.

A GGSN interconnects the PS CN domain with any other packet-based networks such as, an intranet, the Internet and the 3GPP IP Multimedia Subsystem. A GGSN is responsible for carrying out the following specific functions (3GPP, 2006[1]):

- *Routing and Forwarding of User Packets* – A GGSN acts as a packet routing and forwarding centre for user packets. All the user packets to and from PLMN mobile device are first sent to a GGSN, known as the serving GGSN. The serving GGSN will then be tasked with forwarding the user packets towards their final destination.
- *Management of Routes and Mobility* – A serving GGSN tracks the SGSN that is presently serving each mobile, which is referred to as the serving SGSN. The GGSN maintains a route to the serving SGSN and utilizes this route to forward and retrieve user traffic with the SGSN.

To transport user traffic between SGSNs and between an SGSN and a GGSN, IP is utilized as the basic protocol (Lee, 2006).

### 2.5.4 IP Multimedia Subsystem

The IP Multimedia Subsystem (IMS) was introduced on the 3GPP Release 5 (MCC, 2003) and it provides core network entities for supporting real-time and multimedia IP services. The IMS employs the Session Initiation Protocol (SIP) (Rosenberg *et al*., 2002) for signalling and session control for all real-time multimedia services. SIP is an application-layer signalling protocol utilized for establishing, modifying and terminating multimedia sessions between one or more participants. For instance, a session can be an Internet call between two parties or a multimedia conference among multiple parties (Lee, 2006).

### 2.5.5 Information Servers

Information that is essential to provide services to users and for the network to operate efficiently is kept on the information servers (Chen & Zhang, 2004). These information servers, which are shared by both the CS and the PS domain include (3GPP, 2006[1]):

- *Home Subscriber Server (HSS)* – The HSS is the PLMN's master logical database that is responsible for maintaining user subscription information that the network utilizes to control the network services that are offered to the users. The Home Location Register (HLR) is the HSS's main component, which is responsible for maintaining, among other things, mobile locations, users' identities and service subscription information.
- *Authentication Server (AuC)* – The AuC is a logical entity that is responsible for maintaining information that the network utilizes to authenticate each user and also encrypt the communication over the radio path. The AuC is accessed through the HSS, which removes hassles of defining individual interfaces between each network entity and the AuC when they need to access the AuC.
- *Equipment Identity Register (EIR)* - The EIR is a logical entity that is responsible for maintaining IMEIs of subscribers.

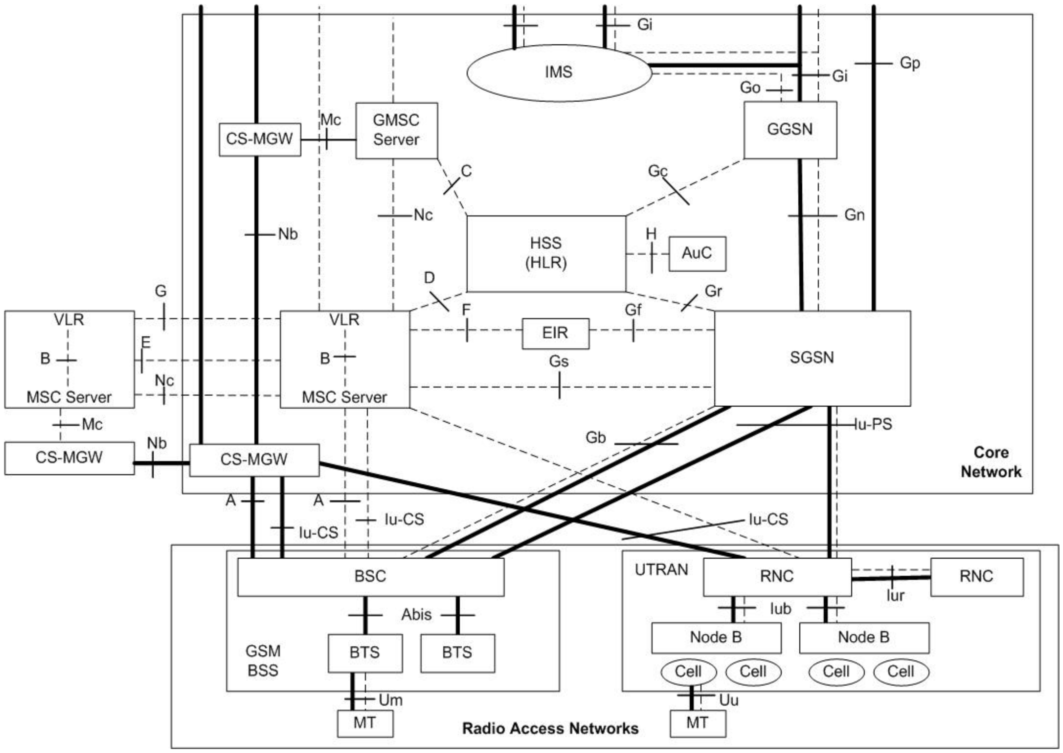

**Figure 2.13: 3GPP Network Architecture and Protocol Reference Model (Release 5) (3GPP, 2006[1])**

*CS CN Internal Interfaces:*

The CS CN interfaces are depicted in Figure 2.13. Unless otherwise stated, the information below is sourced from Chen & Zhang (2004) and 3GPP (2006[1]).

*Interface B* – Non-standardized signalling interface to exchange location information between the MSC server and VLR.

*Interface C* – A signalling interface utilized by the GMSC to retrieve routing information for the mobile device from the HLR. This interface's signalling utilizes the MAP protocol.

*Interface D* – A signalling interface utilized by the HLR and VLR to exchange location and service subscription information. This interface's signalling utilizes the MAP protocol.

*Interface E* – A signalling interface utilized to maintain handoff between MSCs and for transporting Short Message Service (SMS) messages between MSCs. This interface's signalling utilizes the MAP protocol.

*Interface G* – A signalling interface utilized to maintain location registration when a mobile device moves from one VLR area (a network part served by a single VLR) to the next. This interface's signalling utilizes the MAP protocol.

*Interface F* - A signalling interface utilized by the MSC server and EIR to exchange data.  This interface's signalling utilizes the MAP protocol.

*Interface $N_b$*: A transport interface utilized for bearer control and transport between CS CN entities. It allows different protocols to be employed for carrying different upper layer traffic. For instance, Real-time Transport Protocol (RTP), User Datagram Protocol (UDP), or Internet Protocol (IP) can be utilized to transport Voice-over-IP traffic. RTP is a protocol specified by the IETF for end-to-end network transport of real-time data, for example, as audio, video or simulation data (Schulzrinne *et al*., 1996).

*Interface $N_c$*: A signalling interface utilized for call control between MSC Servers or between an MSC Server and a GMSC Server. This interface's signalling typically employs the Signaling System 7 (SS7) ISUP.

*PS CN Internal Interfaces*: The interfaces comprise (1) all the GPRS-specific interfaces that are specified in the 23-series and 24-series of the 3GPP technical specification, (2) the interfaces between a GGSN or SGSN and the MSC, and (3) the interfaces between a GGSN or SGSN and the information servers shared by the PS and the CS domains. The main interfaces in the PS CN domain are:

*Interface $G_n$*: A signalling and transport interface utilized between the SGSN and the GGSN and also between SGSNs within the same PLMN to maintain packet data transport and mobility.

*Interface $G_p$*: A signalling and transport interface utilized between the SGSN and GGSN in different PLMNs. In addition to providing functions of the $G_n$ interface, the $G_p$ interface provides security functions for communication between PLMNs.

*Interface $G_i$*: A standard IP interface between a GGSN (and the IMS) and other IP networks. This interface utilizes IP as the network layer routing protocol. From an external network's perspective, the GGSN plays the role of a regular IP router, while the 3GPP PS domain acts as a regular IP network.

*Interface $G_c$*: A signalling interface utilized by the GGSN to retrieve location and service subscription information from the HSS, in order for the GGSN to determine how traffic to and from a user is handled.

*Interface $G_r$*: A signalling interface utilized by the SGSN to exchange location and other subscriber information with the HSS. For instance, the SGSN can utilize this interface to notify the HSS of a mobile device's current location. This interface can also be used by the SGSN to retrieve information from HSS in order to provide mobile users services. This information includes, for instance, a mobile user's subscribed services, information for controlling the mobile user's network access, etc. This interface's signalling utilizes the MAP protocol.

*Interface $G_f$*: A signalling interface utilized by the SGSN to exchange information with the EIR, in order for the SGSN to verify the IMEI supplied by a mobile user. This interface's signalling utilizes the MAP protocol.

*Interface $G_s$*: A signalling interface between the SGSN and the MSC Server/VLR that allows the SGSN to send location information to the VLR in the CS domain and allows the MSC/VLR to send a paging request to the SGSN. Moreover, it allows the MSC/VLR to inform the SGSN that a mobile device is utilizing the services handled by the MSC. This is a crucial capability that fosters close integration of the networking capabilities provided by the PS and CS domains. . This interface's signalling utilizes the SS7 Signalling Connection Control Part (SCCP) protocol.

## 2.6 Exchanging Data Packets in a 3GPP Network

A mobile device utilizes a Packet Data Protocol (PDP) to exchange user packets with other mobiles over the 3GPP PS CN domain, either within the same 3GPP network or with external IP networks (3GPP, 2006[1]). The PDP Packet Data Units (PDUs) (i.e., the user packets) are transported within the 3GPP network utilizing traffic bearers. A traffic bearer refers to a set of network resources and data transport functions that are utilized to transfer user traffic between two network entities. This includes a path, a logical connection, and physical connections between two network nodes (3GPP, 2006[2]).

In order for a mobile to be able to send and receive user packets over a 3GPP PS CN, a dedicated path needs to be maintained between the mobile and its serving GGSN. This path is known as a 3GPP Bearer or a UMTS Bearer in a UMTS network (Kaaranen *et al*., 2005). A UMTS Bearer is constructed through concatenating a Radio Access Bearer (RAB), which connects a mobile over a RAN to the edge of the CN (i.e., a SGSN), and a CN Bearer that transmits user traffic between the edge of the CN and a GGSN (Chen & Zhang, 2004).

- A RAB is a logical connection that is constructed through concatenating a Traffic Radio Bearer and an $I_u$ Traffic Bearer. A Traffic Radio Bearer is a logical connection that is provided by the protocol layer immediately below the PDP layer for transporting user packets between a mobile and an RNC. An $I_u$ Traffic Bearer is a logical connection that is provided by the protocol layer immediately below the PDP layer for transporting user packets between the RAN (i.e., an RNC) and an SGSN (3GPP, 2006[1]).

- A CN Bearer is a logical connection that is provided by the protocol layer immediately below the protocol layer for transporting user packets between the GGSN and an SGSN (Korhonen, 2001).

A dedicated logical *Signaling Connection* must be established between the mobile and the SGSN, before network resources in a RAN or the PS CN can be allocated to provide packet-switched services to a mobile (Lee, 2006). This signaling connection is utilized, for instance, when the mobile needs to register with the PS CN domain, when the mobile needs CN Bearers established by the SGSN and has to send a request, and for the establishment of Radio Access Bearers (3GPP, $2006^1$).

### 2.6.1 Packet Data Protocol (PDP)

A mobile device should, on its own, acquire and configure a PDP address, for example, an IP address, if the PDP is IP, in order to be able to send and receive user packets (Chen & Zhang, 2004). The mobile device can be able to utilize multiple PDP addresses concurrently. Prior to sending or receiving user packets on a PDP address, a PDP context for the address should be established and activated in the 3GPP PS CN domain (i.e., on an SGSN and a GGSN) as well as on the mobile device (3GPP, $2006^1$).

A PDP context contains information that is utilized by the network to determine how user packets destined to or originating from a particular PDP address can be forwarded. The PDP context, sustained by the mobile device, an SGSN, and a GGSN, join the Radio Access Bearer and a CN bearer to structure a 3GPP for the mobile (Korhonen, 2001).

Important information contained on the PDP contexts maintained by the SGSN and the GGSN include (3GPP, $2006^4$):

- *PDP Address* – utilized by the mobile device to exchange PDP packets.
- *Routing Information* – utilized to determine where a user packet should be forwarded to. This may include identifiers of the tunnels, established

between an SGSN and a GGSN for the particular PDP context, and an Access Point Name (APN). An APN refers to logical name that is utilized by the SGSN to determine the GGSN to be used for the mobile and by the GGSN to determine which services a user has requested, or an external packet network's access point that the user packets should be forwarded to (Chen & Zhang, 2004).

- *Quality of Service (QoS) Profiles* – There are three categories defined (3GPP, 2006[4]):
    - *QoS Profile Subscribed* – Specifies the QoS characteristics subscribed by a mobile user.
    - *QoS Profile Requested* – Specifies the QoS that is presently requested by the mobile user.
    - *QoS Profile Negotiated* – Specifies the QoS that is presently being provided to the mobile device.

The SGSN is responsible for maintaining all the three types of QoS profiles, while the GGSN maintains only the QoS Profile Negotiated.

The state of a PDP context can be either ACTIVE or INACTIVE (3GPP, 2006[1]).

***ACTIVE state***: In this state, the PDP context contains up-to-date information for forwarding PDP packets between the mobile and the GGSN. Although the PDP context is in an ACTIVE state, this does not imply that Radio Access Bearers (RABs), required to transport user packets over the RAN are established (Chen & Zhang, 2004). Instead, the RABs may only be established when the mobile has user packets to send to the network, or the network has packets for mobile.

***INACTIVE state***: In this state, the PDP context may contain a valid PDP address; however, it will not contain valid routing and mapping information needed to determine how to process PDP packets (3GPP, 2006[1]). This means no user data can be transferred between the mobile and the network, and also a mobile's location change will not cause PDP context's update.

If the PDP context for the destination mobile that the GGSN is trying to reach is INACTIVE, the GGSN may utilize the Network-requested PDP Context Activation procedure to change the PDP context of the destination mobile into ACTIVE state. Packets destined to a mobile with an INACTIVE PDP context may also be discarded by the GGSN.

The following actions can be performed on a PDP context in order to alter it between the ACTIVE and INACTIVE states (Chen & Zhang, 2004).

- *PDP Context Activation* - is responsible for creating and activating a PDP context. A successful PDP Context Activation changes the PDP context from an INACTIVE state to an ACTIVE state. The PDP Context Activation procedure may be initiated by either the mobile or the GGSN. However, the GGSN can only initiate the PDP Context Activation under some strict limitations.
- *PDP Context Modification* – is responsible for changing the characteristics of an ACTIVE PDP context. For instance, it can be utilized to modify the PDP address or the attributes of the QoS profile to be supported by the network. The PDP Context Modification procedure may be initiated by the mobile terminal, the RNC in a UTRAN, the SGSN, or the GGSN. However, only GGSN-initiated PDP Context Modification procedure is allowed to modify the PDP Address on 3GPP Release 5.
- *PDP Context Deactivation* – is responsible for removing an existing PDP context. A successful PDP Context Deactivation changes the PDP context from an ACTIVE state to an INACTIVE state. The PDP context state can also be moved from ACTIVE to INACTIVE when the mobile's Packet Mobility Management (PMM) state changes into PMM-IDLE or PMM-DETACHED (3GPP, 2006[1]).

### 2.6.2 Accessing the 3GPP Network and Services Through a Mobile

To activate a mobile user's access to the 3GPP packet-switched network and services, a three-phased process is invoked, as illustrated in Figure 2.14.

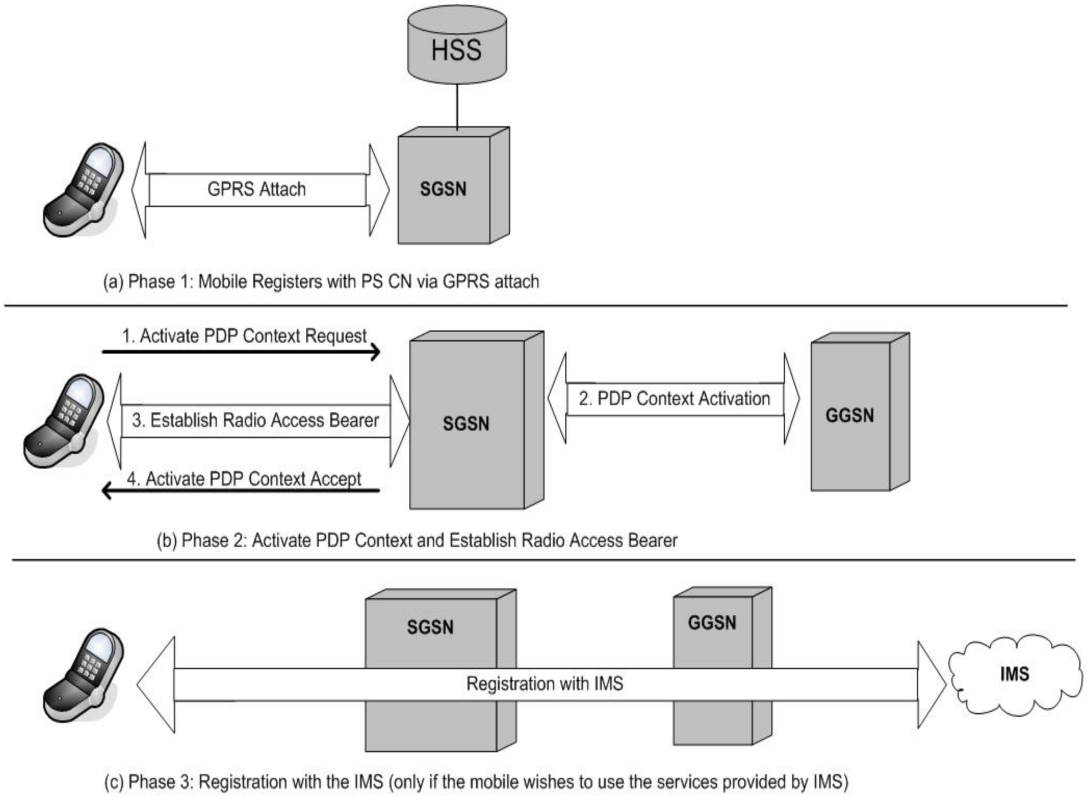
HSS
GPRS Attach
SGSN
(a) Phase 1: Mobile Registers with PS CN via GPRS attach
1. Activate PDP Context Request
3. Establish Radio Access Bearer
4. Activate PDP Context Accept
SGSN
2. PDP Context Activation
GGSN
(b) Phase 2: Activate PDP Context and Establish Radio Access Bearer
SGSN
GGSN
Registration with IMS
IMS
(c) Phase 3: Registration with the IMS (only if the mobile wishes to use the services provided by IMS)

**Figure 2.14: Three-Phased Access to 3GPP Packet-Switched Network and Services (Chen & Zhang, 2004)**

*(a) GPRS Attach* – A mobile utilizes the GPRS Attach procedure to register with an SGSN on the PS CN domain. When performing the GPRS Attach, the mobile provides its identity and service requirements to the SGSN in order to be authenticated and authorized (3GPP, 2006[4]).

As stated by Chen & Zhang (2004), a successful GPRS Attach not only registers a mobile with an SGSN, it also;

- Establishes a Mobility Management Context on the mobile, in the RAN (such as on the RNC in a UTRAN), and on the SGSN. This enables RAN and the SGSN to track the mobile's location.

- Establishes a signalling connection between the mobile and the SGSN. This signalling connection is utilized to exchange signalling and control messages that are required to perform a GPRS Attach procedure between the mobile and the SGSN. After completing the GPRS attach, the signalling connection can still be utilized by the mobile to exchange signaling messages with the SGSN, for instance, carrying out PDP context activation.
- Enables the mobile to access some services provided by the SGSN. These services may include, for instance, sending and receiving SMS messages and being paged by the SGSN. Delivery of SMS messages over signaling connections utilizes the MAP signalling protocol.

On its own, the GPRS procedure does not establish any Radio Access Bearer or CN Bearer for the mobile (3GPP, 2006[4]). As a result, by itself the GPRS attach is not sufficient to enable a mobile to send and receive user packets over the PS CN domain.

*(b) PDP Context Activation and RAB Establishment* – In order for a mobile to be able to make use of a PDP address, the PDP context for the address should be established and activated on the mobile, and the 3GPP PS CN (3GPP, 2006[1]). Following a successful GPRS Attach the mobile can request the network to establish and activate a PDP context for its PDP Address (3GPP, 2006[4]). The PS CN domain will establish the Radio and CN Bearer (i.e., the Radio Bearer and Iu Bearer) utilized to transport user data packets to and from the mobile, after a successful PDP Context Activation. This means a mobile will be able to send and receive user packets over the PS CN domain, following a successful PDP context activation (Chen & Zhang, 2004).

*(c)Register with the IMS*: In order for the mobile to be able to utilize IP-based real-time voice or multimedia services offered by the IMS, it has to complete registration with the IMS (3GPP, 2006[5]). The SIP registration procedure is utilized to register the user with the IMS (Camarillo & García-Martín, 2006).

#### *2.6.2.1 Device Mobility Management*

Even though an end-user's mobile device should be reachable while in motion, it is imperative to keep the network connection and manage the mobility of the device. Whilst this proves to be a challenging task, the network architecture handles it through mechanisms, such as paging, location updating and connection handover (Kaaranen *et al*., 2005). The handover mechanism ensures a smooth handover of a radio connection when an end-user's mobile device moves from one cell (base station) to the next (Korhonen, 2001). Conversely, the location update procedure allows the network to monitor the subscriber camping within the network coverage, whilst paging ensures that the mobile device that the call is destined to is reached. Although there is no continuous active radio link between the network and the mobile device, paging and location update give surety to the mobile device being reached (3GPP, 2006$^1$).

As a way of monitoring the mobile device, the entire geographical area is demarcated into Location Areas (LAs), which are made up of a logical group of cells. A cell contains a base station that services a specific number of simultaneous users through emitting a low level transmitted signal. If a mobile device is switched on, or it changes a LA, it sets off the location update procedure by means of a location update request to the network. After the subscriber information has been validated, the network responds with new location information and nullifies the old information. Location updates can be performed either periodically or on predefined timing (Halonen *et al*., 2003).

If an end-user's mobile device is the destination, paging determines its position through examining the Home Location Register (HLR) and Visitor Location Register (VLR). The whole LA can be paged through a method called *simultaneous paging* (Lee, 2006). However, part of the LA, called the Paging Area (PA), can also be paged to locate the mobile device. If the called device is not reached, paging is performed sequentially over the paging areas utilizing a technique known as *sequential paging* (Korhonen, 2001).

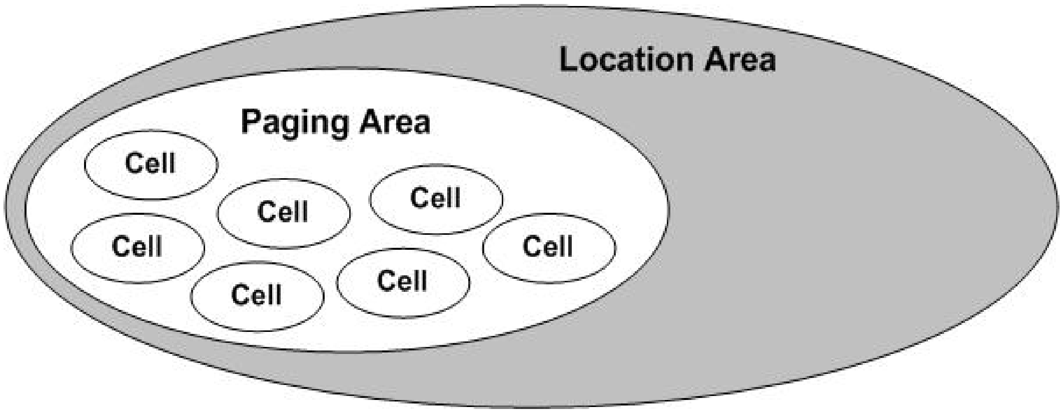

**Figure 2.15: Relationship Between Cell, Paging Area (PA) and Location Area (LA) (Kaaranen *et al*., 2005).**

### 2.6.3 Routing and Transporting User Packets

The main protocol for transporting user packets between network nodes in PS CN domain is IP (for example, between SGSN and GGSN, between SGSNs, between GGSNs, and between RNC and SGSN) (Lee, 2006). Moreover, IP is utilized for routing between GGSNs. However, routing of user packets between SGSN and GGSN does not directly utilize the IP routing protocol, but rather employs GPRS-specific protocols and procedures (3GPP, 2006$^4$). Specifically,

- *GGSN acts as a central point for routing of all user packets*: As mentioned earlier (Section 2.5.3 Packet-Switched Core Network), all user packets to and from any mobile are first forwarded to the mobile's serving GGSN, which will then forward the packets towards their final destinations (depicted on Figure 2.16 below). As illustrated in the figure, both the source and destination mobiles are connected to different GGSNs. Although both the source and destination mobiles can share the same GGSN, user packets will still have to be forwarded to the GGSN first, and then forwarded to the destination by the GGSN (Chen & Zhang, 2004).

- *User packets are tunneled between RNC and SGSN, between SGSN and GGSN, and between two GGSNs.* Tunneling means putting a packet inside another packet (known as an encapsulating packet) and then routing the encapsulating packet based on the information contained in the header of the encapsulating packet (Korhonen, 2001). In the 3GPP packet domain, tunneling is performed utilizing the GPRS Tunneling Protocol (GTP) (Eberspacher *et al*., 2001; 3GPP, 2006$^1$). The GTP tunnel that transports the mobile's packets between an SGSN and the user's serving GGSN forms a CN Bearer for the Mobile

  The two main objectives of tunneling are to (Bates, 2002; 3GPP, 2006$^4$):

  - Enable GRPS protocols between SGSN and GGSN, as opposed to IP routing protocol and IP mobility management protocols, in order to perform routing and mobility management inside the PS domain.
  - Ensure that the transport of user packets inside the PS CN domain is independent of the protocols utilized in external networks. Thus, a 3GPP network does not need to understand external packet data protocols and as a result, this will facilitate the independent evolution of the protocols utilized by mobiles and external packet networks.

- *Host-specific routes are used to forward user packets between a mobile and a GGSN.* The SGSN and the GGSN keep an individual routing entry as part of a PDP context for each mobile device that has an active PDP context – hence the phrase host-specific routes (Bates, 2002). Specifically, for every mobile that has an active PDP context, the routing entry maintained by a GGSN specifies the SGSN that is currently serving the mobile. If the mobile moves from one SGSN to another, the GPRS mobility management procedures will be utilized to update the routing entries on the GGSN and the SGSN (Chen & Zhang, 2004).

User packets that originate from the mobile device's PDP address are conveyed to a Service Access Point provided by the protocol layer, immediately below the IP layer, for transmission to the PS CN. Packets destined to the PDP address will also be delivered by the lower protocol to the IP layer through the Service Access Point. The Service Access Point is identified by a Network-layer Service Access Point Identifier (NSAPI). A unique NSAPI is assigned to every IP address configured on a mobile (3GPP, 2006[1]).

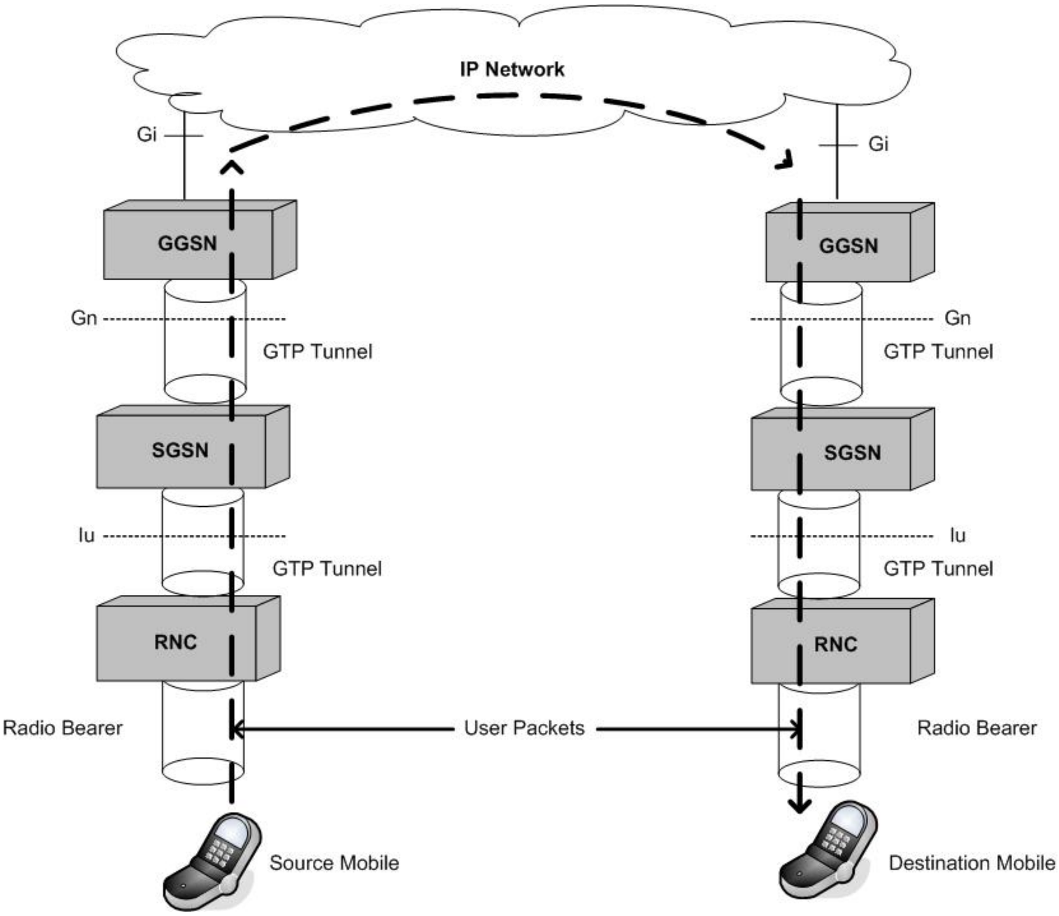

**Figure 2.16: Packet Routing in 3GPP Packet-Switched Domain**
**(Based on Chen & Zhang, 2004)**

The NSAPI for an IP address is sent to the SGSN and GGSN during the PDP context activation process. The SGSN and the GGSN will then utilize the NSAPI to identify the PDP context corresponding to the IP address (3GPP, 2006[1]).

The receiving end side of GTP tunnel locally assigns a Tunnel Endpoint Identifier (TEID) that the transmitting side of the tunnel has to utilize in sending user packets over the tunnel to the receiving side. A TEID is created based on the mobile's IMSI and the NSAPI associated with the PDP context for which the GTP tunnel is established (Eberspacher *et al*., 2001). The TEIDs for a GTP tunnel are exchanged between the transmitting and the receiving sides of the tunnel during the tunnel setup process (3GPP, 2006[4]).

The Radio Access Bearers (RABs) on the mobile, the SGSN and the RNC are identified by a RAB Identifier (RAB ID). The Radio Bearer (RB) that makes up the mobile-RNC portion of a RAB is identified by a RB ID on the mobile and the RNC.

There is a one-to-one mapping between NSAPI, PDP context, and RAB, as illustrated in Figure 2.17. As a matter of fact, the RAB ID, utilized by a mobile and an SGSN to communicate with an RNC, will be identical to the NSAPI (Chen & Zhang, 2004). Moreover, for PS services, each PDP context can employ only one RB. Therefore, there is also a one-to-one mapping between NSAPI, PDP context, RAB and RB.

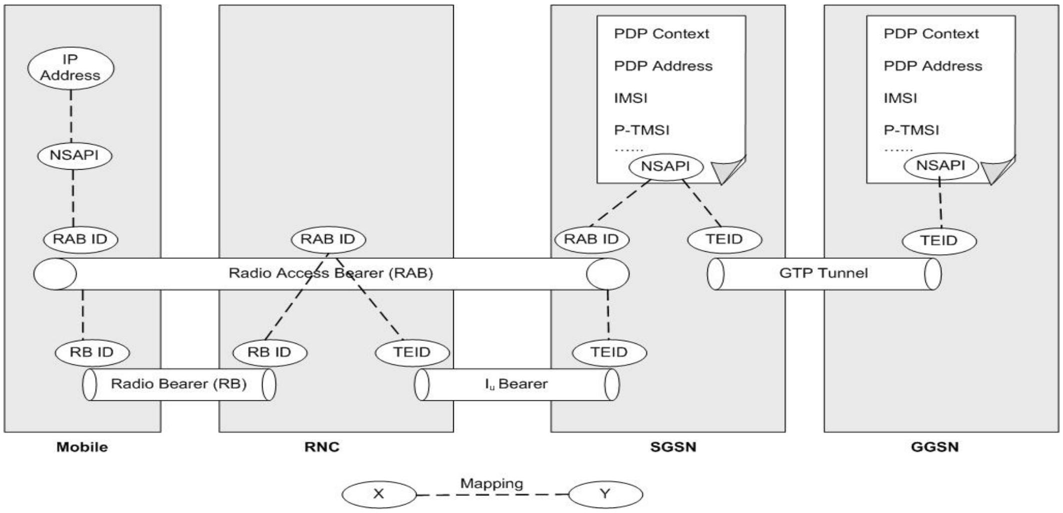

**Figure 2.17: Identifiers of Bearers and Mapping Between These Identifiers (Chen & Zhang, 2004)**

## 2.7 3GPP IP Multimedia System (IMS)

3GPP TS 22.228 specifies the service requirements of the 3GPP IMS (3GPP, 2006[9]). The description of details is in 3GPP TS 23.228 (3GPP, 2006[5]). The IMS supplies all the network entities and procedures that support real-time voice and multimedia IP applications. It utilizes Session Initiation Protocol (SIP) to support signalling and session control for real-time services.

### 2.7.1 Mobile Station Addressing for Accessing the IMS

If a mobile user needs to utilize the services offered on a visited IMS, the mobile should possess an IP address (i.e., the mobile's PDP address), which is logically part of the IP addressing domain of the visited IMS (Camarillo & García-Martín, 2006). The PDP context for this IP address will be activated so that packets destined to this address can be forwarded to the mobile by the 3GPP packet domain.

### 2.7.2 IMS Architecture

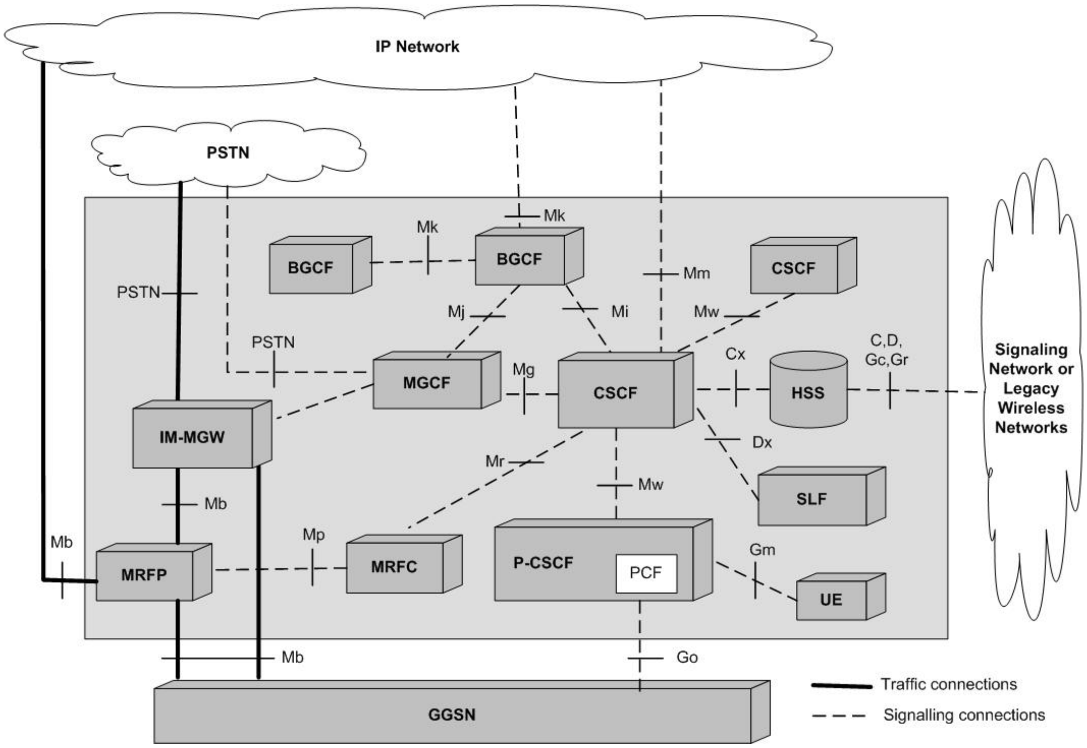

**Figure 2.18: 3GPP IP Multimedia System (Based on Chen & Zhang, 2004)**

#### *2.7.2.1 IMS Reference interfaces*

Unless otherwise stated, the information below is sourced from Camarillo & García-Martín (2006), Chen & Zhang (2004) and 3GPP (2006[5]).

1. *Interfaces for SIP-Based signalling and service control:*
   - Interface $M_g$ – enables the interaction between the CSCF and MGCF.
   - Interface $M_i$ – enables the CSCF to forward session signalling to a BGCF in order for the session to be forwarded to PSTN networks.
   - Interface $M_j$ – enables the BGCF to forward a session signalling to a selected MGCF that will transmit the session to the PSTN.
   - Interface $M_k$ - enables the BGCF to forward a session signalling to another BGCF.
   - Interface $M_r$ – enables the interaction between the S-CSCF and MRFC.
   - Interface $M_w$ – enables an I-CSCF to direct mobile-terminated sessions to an S-CSCF.
2. *Interfaces for controlling media gateways*:
   - Interface $M_c$ – enables a signalling gateway to control a media gateway. For instance it is utilized between an MGCF and an IM-MGW, between an MSC Server and a CS-MGW, or between a GMSC Server and a CS-MGW.
   - Interface $M_p$ – enables an MRFC to control media stream resources provided by an MRFP. Signalling over the Mc and Mp interfaces utilizes the H.248 Gateway Control Protocol (ITU-T, 2002).
3. *Interfaces with the Information Servers*:
   - Interface $C_x$ – enables the CSCF to retrieve mobility and routing information regarding a mobile user from the HSS in order for the CSCF to determine how to process a user's sessions.
4. *Interfaces with external networks*:
   - Interface $M_b$ – is the standard IP routing and transport interface with external IP networks. The interface $M_b$ maybe identical to the $G_i$ interface.

- Interface $M_m$ is a standard IP-based signalling interface that handles signaling interworking between the IMS and external IP networks
- Interface $G_o$ enables the PCF to apply policy control over bearer usage in the GGSN.

### 2.7.3 Call Session Control Function

The Call Session Control Function (CSCF) is the main functional entity in the IMS (Camarillo & García-Martín, 2006; Chen & Zhang, 2004). The IMS organizes the data transfers between users into sessions. The CSCF is thus responsible for controlling these sessions and carrying out functions such as (3GPP, 2006[5]):

- User authentication
- Call routing
- Establishing QoS over the IP network
- Controlling the generation of Call Detail Records (CDRs) for accounting purposes (Bannister *et al*., 2004).

All the signalling and session control in the IMS is performed utilizing the Session Initiation Protocol (SIP) and each CSCF performs a specific task. There are three different types of CSCFs: Proxy CSCF (P-CSCF), Serving CSCF (S-CSCF) and Interrogating CSCF (I-CSCF) (Chen & Zhang, 2004; Bannister *et al*., 2004; Kaaranen *et al*., 2005). To facilitate for load sharing and improved reliability, the network can provide multiple CSCF of each type (Bannister *et al*., 2004).

#### *2.7.3.1 Proxy CSCF*

The P-CSCF acts as the first point of contact for the signalling traffic from the UE (3GPP, 2006[5]). The P-CSCF acts as a SIP Proxy Server (Rosenberg *et al*., 2002), thus it accepts SIP requests from mobiles, and it either serves the requests internally or forwards them to other servers.

The P-CSCF performs four unique tasks: SIP compression, IP Security (IPSec) association, interaction with the Policy Decision Function (PDF) and emergency session detection (Kaaranen *et al*., 2005). SIP messages are larger than binary protocol encoded messages because SIP is text-based, and thus, messages contain a large number of headers and header parameters, which include extensions and security-related information. This implies that more time will be taken during session setup procedures, and performance will somehow be adversely affected by intra-call signalling. 3GPP has thus mandated the support of SIP compression in order to speed up session establishment (Camarillo & García-Martín, 2006).

During SIP registration, the P-CSCF is tasked with maintaining Security Association (SA) and applying confidentiality and integrity protection to SIP signalling. The UE and the P-CSCF will thus negotiate IPSec SAs (3GPP, 2006[5]). The P-CSCF can be able to apply confidentiality and integrity protection to SIP signalling following a successful initial registration. The P-CSCF is also responsible for relaying session- and media-related information to the PDF when an operator wants to apply IP policy control. The PDF is able to derive authorized IP QoS Information from the received information and pass it to the GGSN, if the GGSN needs to carry out IP policy control before accepting a secondary PDP context activation (Kaaranen *et al*., 2005).

Although IMS emergency sessions are not yet fully specified, it is imperative for the IMS network to detect emergency session attempts and, as such, guide the UMTS UE to utilize the CS network for emergency sessions. The P-CSCF is thus responsible for this detection (3GPP, 2006[5]).

#### *2.7.3.2 Serving CSCF*

The S-CSCF maintains a registered user's ongoing session state as well as performing these main tasks (Camarillo & García-Martín, 2006; 3GPP, 2006[5]):

- *Registration* – An S-CSCF can assume the role of a SIP Registrar (Rosenberg *et al*., 2002) and accept users' SIP registration requests and make users' registration and location information available to location servers. After receiving the registration request, the S-CSCF downloads authentication data from the Home Subscriber Server (HSS) and utilizes it to challenge the UE. If the response is verified, the S-CSCF accepts the registration and begins supervising the registration status. Furthermore, a service profile is downloaded from the HSS as part of the registration process.

- *Session Control* – An S-CSCF can carry out SIP session control functions for a registered user and transmission of SIP requests and responses between calling and called party (Camarillo & García-Martín, 2006).
- *Proxy Server* – An S-CSCF can assume the role of a SIP Proxy Server that transmits SIP messages between users and other CSCFs or SIP servers.
- *Interactions with Application Servers* - An S-CSCF can assume the role of an interface that can be accessed by application servers and other IP or legacy service platforms.
- *Other functions* – In addition to the above functions, an S-CSCF can provide service-related event notifications to users and create Call Detail Records (CDRs) needed for accounting and billing (Chen & Zhang, 2004).

#### *2.7.3.3 Interrogating CSCF*

The I-CSCF carries out four unique tasks (3GPP, 2006[5]; Camarillo & García-Martín, 2006; Chen & Zhang, 2004):

1. Obtains the name of the S-CSCF from the HSS.
2. Assigns an S-CSCF based on received capabilities from the HSS. The S-CSCF assignment is performed when a user registers with the network, or a user receives a SIP request while unregistered but still wants to receive services related to the unregistered state (e.g. voicemail).
3. Route incoming requests further to an assigned S-CSCF.
4. Provide Topology Hiding Inter-network Gateway (THIG) functionality.

The Media Gateway Control Function (MGCF) is responsible for signalling whilst the IP Multimedia Media Gateway (IM-MGW) handles media interworking between the PS domain and circuit-switched networks (e.g., PSTN) (Camarillo &

García-Martín, 2006). The Multimedia Resource Function Processor (MRFP) is responsible for controlling the bearer on the Mb interface, including processing the media streams (e.g., audio transcoding). The Multimedia Resource Function Controller (MRFP) is responsible for interpreting signalling information from an S-CSCF or a SIP-based Application Server and controls the media stream resources in the MRFP accordingly (3GPP, 2006[5]). The MRFC also handles the creation of CDRs. The Breakout Gateway Control Function (BGCF) handles the selection of a PSTN network that a session should be forwarded to (Chen & Zhang, 2004).

## 2.8 Conclusion

This chapter discussed wireless networks and how they can be implemented in (HIV/AIDS) VIOISs. Wireless networks based on mobile technologies evolved from analogue (first-generation) networks to packet-based digital (third-generation) networks. 3G networks were designed to offer more radio system capacities, improve interoperability, and enhance QoS support over 2G networks. They also support a broader range of IP-based mobile services and increase per-user data rates up to 2Mbps.

The architecture of 3G networks and the radio access technologies utilized were explored in this chapter. The 3GPP network elements were also examined, with a focus on the Core Network Packet-Switched (CN PS) Domain. The CN PS will be utilized to transport data packets from the various health institutions. The way in which these packets are routed and transported was also examined. Subsequently, the future of wireless networks, the IP Multimedia System (IMS), was discussed.

Once data has been wirelessly transported from various data-providers to the service-providers, it will be stored in data warehouses. However, in order to determine which data warehouse should store the data, middleware known as the Globus Toolkit Version 4 (GT4) is, typically, utilized. GT4 is the widely accepted *de facto* standard implementation of Web Service Resource Framework (WSRF) specification, which specifies a Web Services-based implementation of Open Grid

Services Architecture (OGSA) defined services. The Open Grid Services Architecture (OGSA) specification is the standard for building service-oriented Grid systems. Grid systems and Grid computing are further explored in the next chapter. GT4 is examined in chapter 4 while data warehousing is discussed in chapter 5.

# Chapter 3

# Grid Computing

## INTRODUCTION

The Grid is a new discipline with the concepts and technologies utilised still fresh. Having been first pointed out by Foster and Kesselman in 1998, the core components which make up the Grid infrastructure are still under investigation and have yet to be fully determined. Efforts to coordinate wide-area distributed resources before the advent of the Grid were referred to as metacomputing (Romanenko, 1991; Smarr & Catlett, 1992). The Grid has steadfastly evolved from several non-interoperable high-performance centres (De Roure, Baker, Jennings & Shadbolt, 2003), to a seamless service oriented, collaborative, open-standards based and dynamic virtual environment (Joseph, Ernest & Fellenstein, 2004).

This chapter briefly discusses early distributed systems, identifying the pros and cons which have led to the evolution to service-oriented architectures. Thereafter, Grid computing with the use of XML, SOAP, WSDL, UDDI and Web services as supporting open standards for interoperability will be discussed. The Open Grid Services Architecture (OGSA) framework and its capabilities will be explored. The OGSA specification, which is the *de facto* standard for building service-oriented Grid systems, will then be explored. Subsequently, WSRF, which provides a technical specification for implementing OGSA Grid services, will be examined.

### 3.1 Service-Oriented Architectures

The concept of Service-Oriented Architecture (SOA) is not entirely new, as it has been around for many years. However, it has gradually evolved and the technologies utilized for its implementation have vastly improved over time.

The traditional SOA paradigms include socket programming, Remote Procedure Calls (RPC), Java Remote Method Invocation (RMI), Distributed Component Object Model (DCOM) and Common Object Request Broker Architecture (CORBA). These technologies are briefly discussed in the following sections.

### 3.1.1 Socket programming

Socket programming is a low-level Application Program Interface (API) for building distributed client/server applications. Sockets utilize either Transmission Control Protocol (TCP) (Postel, 1981) or User Datagram Protocol (UDP) (Postel, 1980) as the transport protocol and a socket endpoint should be created before communication commences between the client and the server. A client should also specify the hostname and port number that the server process is listening on. Although a standard socket API is well-defined, the implementation is language dependent (Li & Baker, 2005, p. 14). Thus, socket-based programs can be written in any language, although the socket APIs might vary with each language used.

### 3.1.2 RPC

RPC is another computing paradigm for writing distributed client/server applications. . In a distributed computing setup, where there can be processes running from disparate systems, RPC creates a façade for processes to believe there are executing in the same process space (Mahmoud, 2004). RPC employs a block and wait (suspend processing) mechanism, whereby the calling process waits for the reply from the called process before it can continue processing. The transport protocol utilized for communication is either TCP or UDP. In order to depict remote procedures running on the server side, RPC relies heavily on an Interface Definition Language (IDL). An RPC compiler can create a client-side stub and a server-side skeleton automatically from an RPC IDL interface. The stub and skeleton architecture facilitate a high-level communication abstraction for the clients while hiding the low-level communication. RPC-based client/server applications can be implemented through the most common RPC implementations, Sun Microsystems' Open Network Computing (ONC) RPC (Srinivasan, 1995) and

Open Software Foundation's (OSF) Distributed Computing Environment (DCE) RPC (Poole *et al.*, 1996). Data-flow control in RPC-based client/server applications is depicted in Figure 3.1 below.

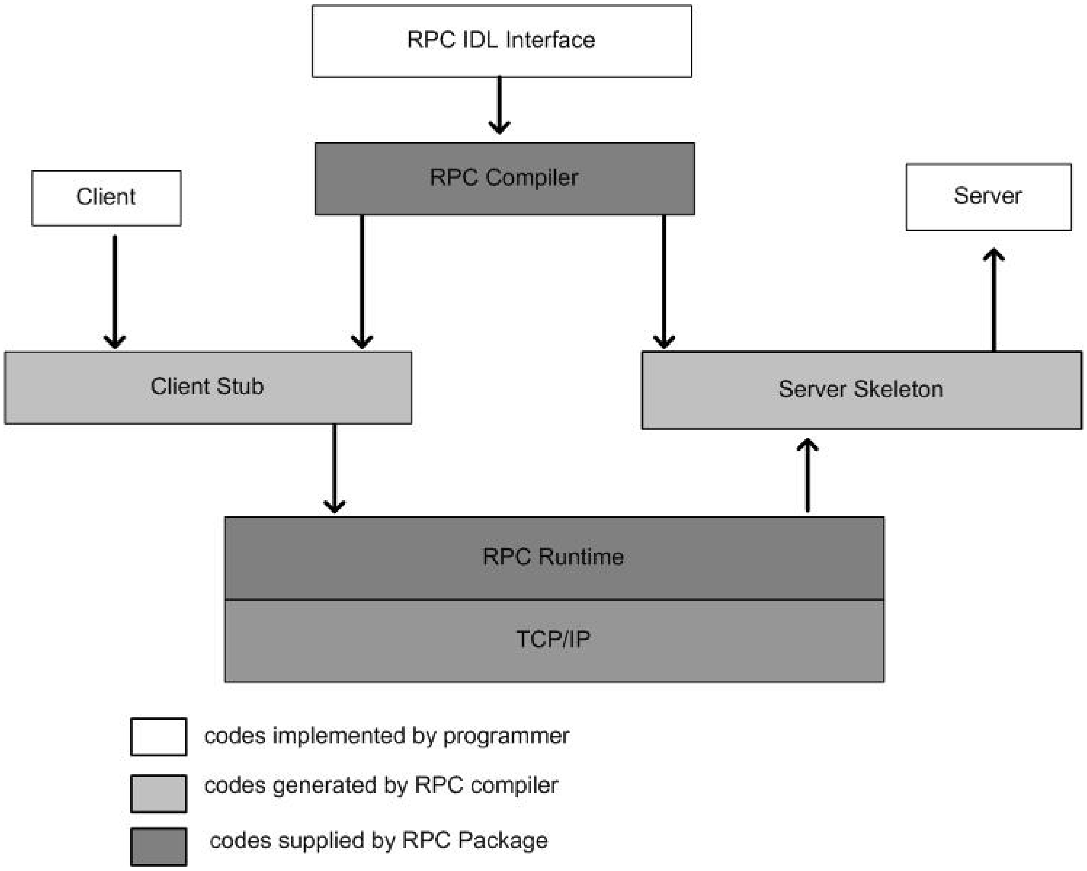
RPC IDL Interface
RPC Compiler
Client
Server
Client Stub
Server Skeleton
RPC Runtime
TCP/IP
codes implemented by programmer
codes generated by RPC compiler
codes supplied by RPC Package

**Figure 3.1: Data-Flow Control in an RPC application (Li & Baker, 2005)**

### 3.1.3 Java RMI

The Java RMI is an object-oriented mechanism from Sun Microsystems for constructing distributed client/server applications (SDN, 1997). Java RMI is an RPC implementation in Java (Java, 1997). Akin to RPC, Java RMI hides the low-level communications between the client and server by utilizing a client-side stub and a server-side skeleton (which is not needed in Java 1.2 onwards) that are generated automatically from a class that extends java.rmi.UnicastRemoteObject and implements an RMI Remote interface (Reilly, 1998).

There are three entities that interact at run time in an RMI application. These are (Reilly, 1998);

1. A client which invokes a method on a remote object
2. A server which runs the remote object, which is an ordinary object in the address space of the server space of the server process
3. The object registry (rmiregistry), which is a name server that relates objects with names. Remote objects need to be registered; the registry can be used to obtain access to a remote object using the name of that object.

Although Java RMI is restricted to the Java language because the implementation of an RMI client and server is limited to this language, it can however run on diverse operating systems in distributed locations (Li & Baker, 2005, p. 16). In order to communicate with the server, the RMI client should specify the server's hostname (IP address) and utilize the Java Remote Method Protocol (JRMP) for remote object invocation on the server.

### 3.1.4 DCOM

The Component Object Model (COM) is a binary standard for constructing Microsoft-based component applications, which is independent of the implementation language (Microsoft, 1996). The COM-interface is responsible for receiving a request or call from a client application, which it forwards to the component. DCOM simply extends COM for distributed client/server applications (Li & Baker, 2005, p. 18). Analogous to RPC, DCOM hides the low-level communication through an automatically generated client-side stub (known as proxy in DCOM) and a server-side skeleton (known as a stub in DCOM) utilizing Microsoft's Interface Definition Language (MIDL) interface (Microsoft, 1996). DCOM employs a protocol, known as Object Remote Procedure Call, when invoking remote COM components. The ORPC is layered on top of the OSF DCE RPC specification (Li & Baker, 2005, p. 19). Data-flow control in a DCOM client/server application is depicted in Figure 3.2 below.

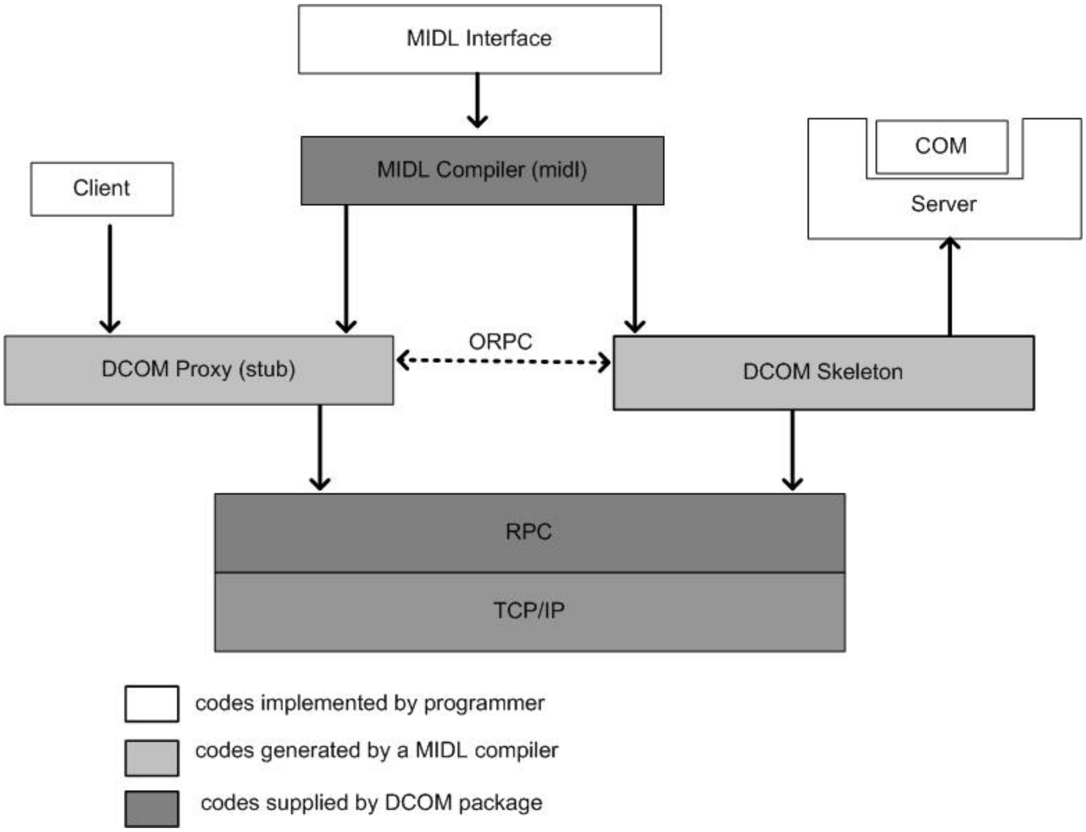
MIDL Interface
MIDL Compiler (midl)
Client
COM
Server
ORPC
DCOM Proxy (stub)
DCOM Skeleton
RPC
TCP/IP
codes implemented by programmer
codes generated by a MIDL compiler
codes supplied by DCOM package

**Figure 3.2: Data-Flow Control in a DCOM application (Li & Baker, 2005)**

### 3.1.5 CORBA

Common Object Request Broker Architecture (CORBA) is Object Management Group's (OMG) (OMG, 1997) object-oriented middleware infrastructure that is utilized to build distributed client/server applications. Akin to DCOM and Java RMI, it hides the low-level communication through an automatically generated client-side stub and a server-side skeleton by an Interface Definition Language (IDL) interface. CORBA utilizes Internet-Inter ORB Protocol (IIOP) to invoke remote CORBA objects (Chaffee & Martin, 1999). The Object Request Broker (ORB) is the central component of CORBA; it carries out data marshalling and unmarshalling between CORBA objects and clients.

### 3.1.6 Java RMI, DCOM and CORBA Summary

Java RMI, DCOM and CORBA have been is use for some time and characterize the most popular distributed object-oriented middleware that can be utilized to rapidly develop distributed client/server applications. Even though they have different implementations and features (Gopalan, 1998), they possess the following similarities:

- To invoke a remote object or a component, they need an interface.
- They hide the complex low-level communications through an automatically generated client-side stub and server-side skeleton by the interface definition.
- They utilize proprietary communication protocols - for instance, Java RMI employs JRMP, DCOM employs ORPC while CORBA utilizes IIOP – in order to invoke remote objects or components.
- Their interface definitions are in binary format. It is not easy for client applications to perform queries on an interface, for example, finding out what kinds of methods have been defined, inputs/outputs of each method in order to utilize the methods at their best.
- There is tight coupling of clients and objects with their interfaces. For instance, if client's part is changed it implies that other parts, such as the server, also need to be modified.

In summing up, computing paradigms such as Java RMI, DCOM and CORBA are proprietary technologies that are not based on open standards, thus restricting interoperability that is required in heterogeneous environments (Li & Baker, 2005, p. 21). In order to address these short comings, there is a need for an open standards-based middleware infrastructure that can be utilized to build and integrate applications in heterogeneous environments; in this case (HIV/AIDS) VIOISs and Web services have proved to be such an infrastructure (Wall & Lader, 2002; Wiehler, 2004). The next section discusses the evolution of computational grids to a service-oriented architecture that utilizes Web service standards.

## 3.2 The Grid

The motivation behind computational grids can be attributed to the notion of power grids. Power grids allow users quick and simple access to power. The users do not have to know about how the power gets to them or where it actually comes from. The same properties are imagined by computational grid architects, but for compute power as opposed to electrical power. However, the focus is not only on compute power, but also on other resources, likes data, storage, networks e.t.c. Users can then ‘*plug in*’ to the Grid and access this vast amount of resources as if they had a supercomputer.

The first generation of computational grids, known as metacomputing, was to link supercomputing sites and thus provides computational resource to a variety of high performance applications (De Roure *et al*., 2004). The systems were proprietary solutions for example FAFNER (FAFNER, 1995) and I-WAY (Foster, Geisler, Nickless, Smith & Tuecke, 1996). In order to cope with heterogeneity, scalability and adaptability issues, middleware was introduced in the second generation of computational grids. These focused on large scale computational power and large amounts of data which was implemented on grid technologies like distributed object systems, complete integrated systems, resource brokers and schedulers, and peer-to-peer systems (De Roure *et al*., 2004). Some of the Grid-related projects were Globus (Foster & Kesselman, 1997), Common Object Request Broker Architecture (OMG, 1997), Condor (Condor, 1988), Jini (Jini, 1996), Unicore (Huber, 2001), Napster (Napster, 2003) and Berkley Open Infrastructure for Network Computing (BOINC) (Anderson, 1999). Although the second generation offered interoperability there was need for existing resources and components to be reused. This brought interest into metadata and service-oriented approach. The third generation of computational grids is thus geared towards automation which in turn results in the autonomy of systems.

The term ‘Grid’ was first coined by Ian Foster and Carl Kesselman in their book, *The Grid: Blueprint for a New Computing Infrastructure,* which was published in 1998. The initial definition was “a computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities” (Foster & Kesselman, 1998). There have been subsequent definitions as the Grid evolved, such as “a grid is a software framework providing layers of services to access and manage distributed hardware and software resources” (CCA, 1999). In 2001, Foster and Kesselman, along with Tuecke, also improved their definition of a Grid to “coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organisation” (Foster *et al*., 2001). This definition is the one that is generally used to describe the Grid.

In 2002, Ian Foster produced a three-point checklist that could be utilized to help determine what can be recognized as a Grid system (Foster, 2002). The first checkpoint is that there is coordinated resource sharing without a centralized point of control although users reside in different administrative domains as is the case in (HIV/AIDS) VIOISs. The second point is the use of standard, open, general-purpose protocols and interfaces. These will enable interoperability between the different system components. The third point to check off is the delivery of non-trivial qualities of service. Thus, coordinated Grid components are utilized to deliver a variety of qualities of service such as throughput, response time, availability, security, meantime between failure, etc., that are significantly greater than the sum of individual components.

In order for Grid-related technologies, tools and utilities to be widely accepted by the community at large, they must be developed and maintained with the help of a community standards body (Mahmoud, 2004). Since there was no standard to guide the early Grid developments, there have been numerous non-interoperable Grid frameworks (Foster *et al*., 2001; Grimshaw & Wulf, 1997; Romberg, 1999; Condor, 1998). To coordinate the standardization of Grid development, the Global Grid Forum (GGF) has presented itself as the primary organisation for setting standards for the Grid, while the Organization for the Advancement of Structured Information Standards (OASIS), is a non-profit consortium responsible for the development, convergence and adoption of e-business standards, which have an increasing influence on Grid standards.

There are four document types produced by GGF that are related to standards. These are (GGF, 2006):

- *Informational*: These are utilized to inform the community about a useful idea or set of idea. These include GFD.7 ("A Grid Monitoring Architecture"), GFD.8 ("A Simple Case Study of a Grid Performance System") and GFD.11 ("Grid Scheduling Dictionary of Terms and Keywords"). Currently there are 18 informational documents from a range of working groups.
- *Experimental*: These are utilized to inform the community about a useful experiment, testbed or implementation of an idea or set of ideas. These include GFD.5 ("Advanced Reservation API"), GFD.21 ("GridFTP Protocol Improvements") and GFD.24 ("GSS-API Extensions"). There are currently three experimental documents
- *Community practice*: These are utilized to inform the community of common practice or process, with the objective of influencing the community. These include GFD.1 ("GGF Document Series"), GFD.3 ("GGF Management") and GFD.16 ("GGF Certificate Policy Model"). There are currently four common practice documents
- *Recommendations*: These are utilized to document a specification, analogous to an Internet Standards track document. These include GFD.20 ("GridFTP: Protocol Extensions to FTP for the Grid") and GFD.23 ("A Hierarchy of Network Performance Characteristics for Grid Applications and Services"). There are currently four recommendation documents.

### 3.2.1 Grid Architecture

According to von Laszewski & Wagstrom (2004), it is imperative to be acquainted with multifaceted aspects of the Grid Architecture before selecting an architectural abstraction to address a particular Grid research area. There are three architectural views, which are (a) layered, from low to high level (b) role-based, permits secure access to collective multiple resources, (c) service-oriented, allows an abstraction model that ca be simply created, deployed and contributed to.

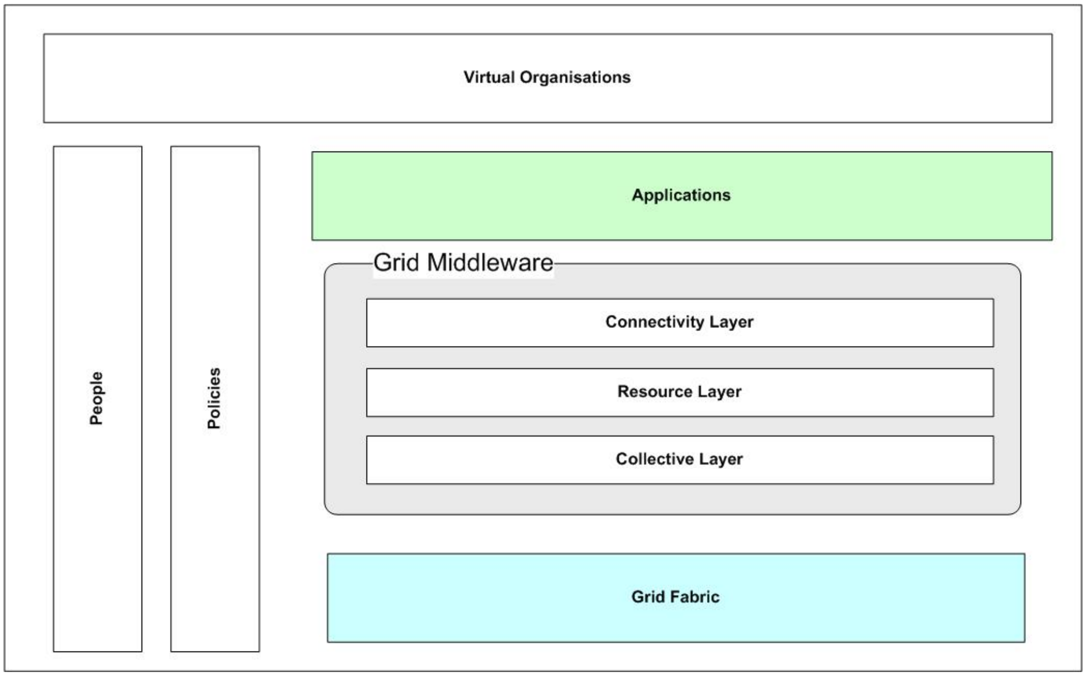

**Figure 3.3: Grid Architecture (Based on Mahmoud, 2004)**

As depicted in Figure 3.3, people access and utilize resources on a virtual organization, which are managed through policies (Mahmoud, 2004). The Grid fabric layer permits access to locally controlled resources, for example storage resources, computing resources, network resources, instruments e.t.c, through services and components developed using toolkits, application interfaces and protocols on the fabric.  Grid middleware consists of three layers; the connective layer, the resource layer, and the collective layer (Mahmoud, 2004; von Laszewski & Wagstrom, 2004).

- The connectivity layer defines Grid-specific core communication and authentication protocols that ensure secure network transactions between the multiple resources on the fabric.  These protocols enable secure message exchange, single-sign-on authentication, delegation and authorization (von Laszewski & Wagstrom, 2004; Foster *et al*., 2001).

- The resource layer deals with ensuring secure access to individual resources and monitoring thereof by collective operations.

- The collective layer defines the different resources that make up the virtual organization and co-ordinates the use of these multiple resources (Mahmoud, 2004). The services offered by this layer includes directory services for resource discovery, data replication services for data-storage and access, and scheduling, brokering and monitoring services for resource management (Foster *et al*., 2001).

Lately, there has been a shift towards service-oriented concepts within the information technology industry. A service, in regard to Grid computing, is a self-describing, platform-independent software component that is published by a service-provider on a registry or directory (Mahmoud, 2004). As depicted in Figure 3.4, services are described and published by the service-providers. A service requester can then query (find) the registry to locate the services offered through a process called resource discovery. The suitable service can be selected and invoked through binding (von Laszewski & Wagstrom, 2004).

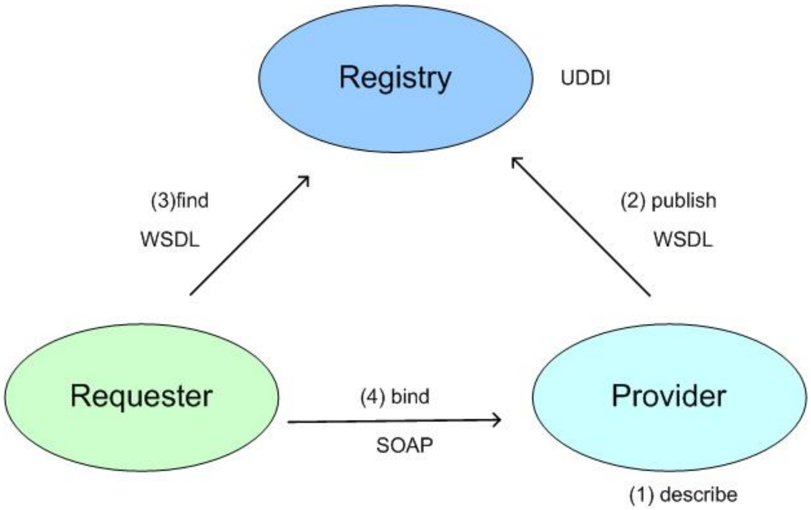

**Figure 3.4: Service-Oriented Architecture (Based on Mahmoud, 2004)**

Service-oriented architectures (SOAs), are emerging as the architectures and technologies of choice when it comes to implementing distributed systems such as grids, and integration of applications within and across organisational boundaries (Alonso & Casati, 2005). The OASIS committee described SOAs, on their latest reference model (MacKenzie *et al.*, 2006), as "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domain". The most significant benefit of employing SOA is the loose coupling among services. The consumer of a service does not have to know how a service is implemented, the implementation language of service, or the platform where the service will execute. The consumer can only be concerned about how a service can invoked as pertaining to the service interface (Chen, Cai, Turner & Wang, 2006). It is important to be able discover the services on the SOA.

As an example, a data analysis service in one institution may be generated from Java, and execute on a J2EE platform, while in another institution it can be generated from VB.Net, and execute on the .Net platform. These two services can be invoked through a composite application or could be consumed as Web services, which enables application re-use (Anantharangachar, Krishna & Shrinivas, 2006). Furthermore, the scalability of a system is vastly improved because the dependency between the loosely coupled services is low (Chen *et al*., 2006).

SOA employs WS-* standards such as UDDI, WSDL, SOAP, etc., to ensure interoperability between services. These Web services “family” of standards are considered in the next section.

### 3.2.2 XML

eXtensible Markup Language (XML) is a W3C initiative designed to describe the structure of data (Bray *et al.*, 2004). XML is utilized to describe the data elements and data structures of (HIV/AIDS) VIOISs with tags. It has been widely-adopted primarily because of its platform-independence and simplicity. XML separates style from content which enables the integration of data from different sources, as it is the case.

XML makes use of Unicode character encoding, which enables the display and exchange of the different languages all over the world. XML is the underlying technology standard for the SOAP standard and Web Service specifications, and it has also been endorsed major corporate information technology companies, such as Microsoft and IBM, as the principal technology.

An example of an XML document is illustrated below

```
<?xml version="1.0"?>
    <LabReseachReport xmlns="http://UbuntuResearch.org/LRR.xsd"
    reportDate="2006-07-20">
    </pathologyTests>

        <pathologyTest Type=“Blood Film“>
            <Date>2006-07-19</Date>
             <Category>Microscope Test</Category>
             <TestDescription>5ml blood in EDTA. Film spread, fixed, stained
             and examined by microscopy. Evaluation of changes in numbers or
             morphology of red cells, white cells and
             platelets</TestDescription>
             <Results> No abnormal cells detected </Results>
             <Notes>Another test should be carried out after 6 months</Notes>
         </pathologyTest>
         <pathologyTest Type=“Blood Group“>
             <Date>2006-07-19</Date>
            <Category>Agglutination Pattern</Category>
            <TestDescription>10ml blood in plain tube. Patient's red cells tested
             with anti-A and anti-B sera for ABO (forward) group; with anti-D
             antiserum  for Rh(D) group. Patient's serum tested with A1 and B
            cells to check ABO  (reverse) group. Tests are based on the
            detection of red cell agglutination.</TestDescription>
            <Results> Blood Group A</Results>
            <Notes> Extended phenotyping necessary</Notes>
       </pathologyTest>
      </pathologyTests>

      <HealthProvider Category=“ Private Hospital“>
            <Name>Greenacres</Name>
            <StateOrProvince>Eastern Cape</ StateOrProvince>
            <District>Nelson Mandela Metropole</District>
            <Country>South Africa</Country>
      </HealthProvider>

</LabReseachReport>
```

An XML-document can be transformed, using XML Stylesheet Language Transformation (XSLT) into various visualisation formats. XSLT enables data to be visually displayed in user friendly formats like XHTML (Web browser) or .pdf (Adobe Acrobat reader). An XML-document can be manipulated as an object tree using Document Object Model Application Programmer Interface (DOM API) (Le Hégaret, Whitmer & Wood, 1997).

XML-metadata (data structure) and data elements can be described and validated in Document Type Definition (DTD) or XML Schema Definition (XSD) documents, which are also XML documents (Cowan & Tobin, 2004). XML Schema Definitions (XSD's) are used by standard bodies, like W3C, to represent standard specifications. Thus, XML "infosets" that contain metadata for tags and data structure, and explanation of related semantics on how the specification should be used (Cowan & Tobin, 2004). Web Service Standards (WS-*) specifications, Web Services Description Language (WSDL) and the SOAP specification consist of these XML infosets.

### 3.2.3 SOAP

SOAP is a lightweight messaging framework for exchanging XML-formatted data among Web services in a decentralized, distributed environment, such as (HIV/AIDS) VIOISs (Medjahed *et al*., 2003; Box *et al*., 2000; Mahmoud, 2004). Message-structure and message-processing descriptions are also defined by the framework. The adoption of SOAP as an XML-based messaging protocol facilitates the interaction between heterogeneous systems. A major design goal for SOAP, as indicated in the SOAP Specification (Box *et al*., 2000), is simplicity and extensibility. SOAP defines a convention for making remote procedure calls (RPC) and a set of encoding rules for serializing data. RPC allows an application to make use of the functionality published by another remote application(s).

A SOAP-message contains the Envelope (top-level container), the Header (generic container that allows features to be added to the message) and the Body (information container). The SOAP-message Header and Body may contain multiple blocks as defined by different applications. This is illustrated in Figure 3.5.

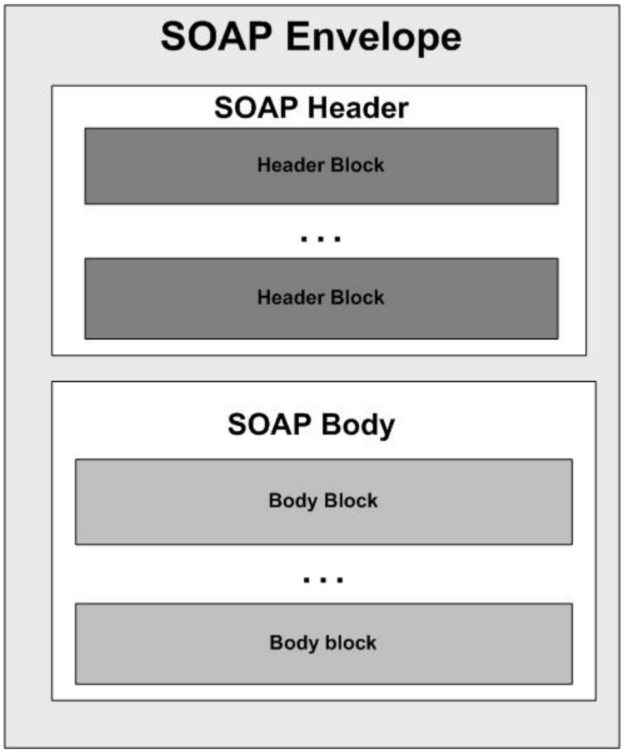

**Figure 3.5: SOAP-Message Structure**

The SOAP-envelope carries a description of the message contents and how the message should be processed. The SOAP-header contains the processing information and the SOAP-body contains the payload (data). The header section allows “features” or “modules” to be added in order to reference other XML infosets. An example of such infosets is existing Web service standards, such as Web Services Security (WS-Security) (Atkinson *et al.*, 2002) and Web Services Addressing (WS- Addressing) (Bosworth *et al.*, 2004).

A SOAP-message can contain binary attachments through the use of the Message Transmission Optimisation Mechanism (MTOM) standard (Gudgin *et al.*, 2005). A SOAP-envelope can be transported using either HTTP or SMTP.

#### *3.2.3.1 MTOM*

MTOM, published as a W3C Recommendation on 25 January 2005, defines a mechanism for optimizing the transmission and/or wire format of a SOAP-message (Gudgin *et al*., 2005). It selectively re-encodes parts of a message whilst still presenting the SOAP application with an XML infoset. It provides a description for an inclusion mechanism that is binding-independent, as well as a specific binding for HTTP. MTOM's functionality depends primarily on the use of XML-binary Optimized Packaging (XOP). The optimisation process is illustrated on Figure 3.6 below (as illustrated on the W3C document). An XOP package is created through serializing an XML infoset and placing it in an extensible packaging format, for example a MIME Multipart/Related package.

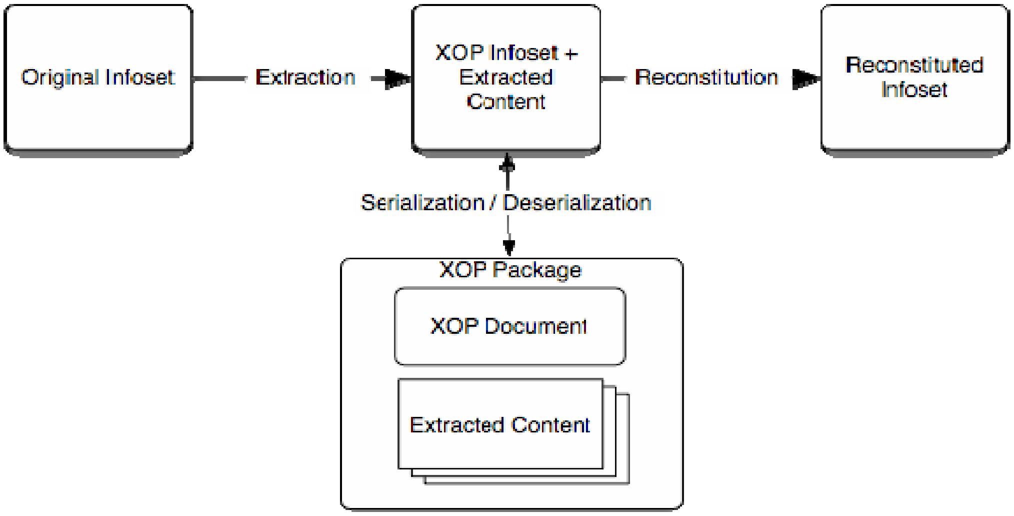

**Figure 3.6: The Mechanism of XOP as used in MTOM (Gudgin *et al*., 2005)**

The following code listing illustrates a MIME-serialized multi-part XML Document containing a .jpeg image being optimized. This is a typical example of how MTOM operates, depicting a Lab Researcher in (HIV/AIDS) VIOISs sending a lab experiment report on White Blood Cells to another researcher or medical practitioner.

```
<! Before Optimization>
<soap:Envelope
    xmlns:soap='http://www.w3.org/2003/05/soap-envelope'
    xmlns:xmlmime='http://www.w3.org/2004/11/xmlmime'>
  <soap:Body>
    <m:data xmlns:m='http://UbuntuResearch.org/LabReseacher'>
      <m:photo
            xmlmime:contentType='image/jpeg'>/aWKKapGGyQ=</m:photo>
     </m:data>
  </soap:Body>
</soap:Envelope>

<!After Optimisation>
MIME-Version: 1.0
Content-Type: Multipart/Related;boundary=MIME_boundary;
    type="application/xop+xml";
    start="< LabReseacherReport.xml@UbuntuResearch.org>";
    startinfo="application/soap+xml; action=\"ProcessData\""
Content-Description: A SOAP message with a picture of White Blood Cells
--MIME_boundary
Content-Type: application/xop+xml;
    charset=UTF-8;
    type="application/soap+xml; action=\"ProcessData\""
Content-Transfer-Encoding: 8bit
Content-ID: < LabReseacherReport.xml@UbuntuResearch.org >

<soap:Envelope
    xmlns:soap='http://www.w3.org/2003/05/soap-envelope'
    xmlns:xmlmime='http://www.w3.org/2004/11/xmlmime'>
  <soap:Body>
    <m:data xmlns:m='http://UbuntuResearch.org/LabReseacher'>
      <m:photo
  xmlmime:contentType='image/jpeg'>
<xop:Include
    xmlns:xop='http://www.w3.org/2004/08/xop/include'
    href='cid: http://UbuntuResearch.org/LabReseacher/WBC.jpeg'/></m:photo>
     </m:data>
  </soap:Body>
</soap:Envelope>

--MIME_boundary
Content-Type: image/jpeg
Content-Transfer-Encoding: binary
Content-ID: <http://UbuntuResearch.org/LabReseacher/WBC.jpeg>
// binary octets for jpeg
--MIME_boundary
```

#### *3.2.3.2 Synchronous Messaging vs Asynchronous Messaging*

Table 3.1 below compares and contrasts synchronous and asynchronous SOAP-messaging.

**Table 3.1: Synchronous Messaging vs Asynchronous Messaging**

| | Synchronous | Asynchronous |
|---|---|---|
| Coupling | Tightly coupled. Any interface change will affect both the source and target system. Increase of changes will cause a ripple effect on the cost. | Loosely coupled. Disparate source and target systems can exchange messages. Changes on one subsystem will not affect the other systems. |
| Reliability | Unreliable as any service malfunction will affect the communication between source and target system | High levels of reliability are achieved through a store and forward mechanism. The message is persisted in case of system failures or temporary system unavailability. Message delivery is guaranteed exactly once. |
| Scalability | The block and wait processing technique slows the performance of the entire system to the maximum speed of the slowest subsystem. More bandwidth is required as more subsystems interact with each other. | A load balancing technique is used to decouple performance qualities of subsystems. This permits subsystems leeway to choose rather than being dictated to receive a message. |

| | | |
|---|---|---|
| Availability | Depends on the concurrent availability of all subsystems. An imminent failure or temporary service outage will render the whole system unavailable. | Loose coupling prevents a failure affecting the whole system and thus promotes high availability capability. |

From the distinctions outlined in Table 3.1, asynchronous messaging proves to be the better option for (HIV/AIDS) VIOISs. Asynchronous messaging is further explored in the following sections.

#### *3.2.3.3 Message Queues*

The message queue is a fundamental concept within asynchronous messaging. Messages are stored on the (HIV/AIDS) VIOIS Grid Portal’s Web server queue. This provides the SMTP/HTTP servers (sender and receiver) the ability to send messages to and receive messages from this queue. The messages are sorted in a First-In First-Out order. Thus, the first message that is submitted to the queue will be the first message retrieved from the queue.

An SMTP/HTTP server (receiver) can effectively *pull* messages from the Web server queue. Alternatively the receiver can instruct the Web server to *push* relevant messages to it as soon as they get to the front of the queue. This is depicted on Figure 3.7 below.

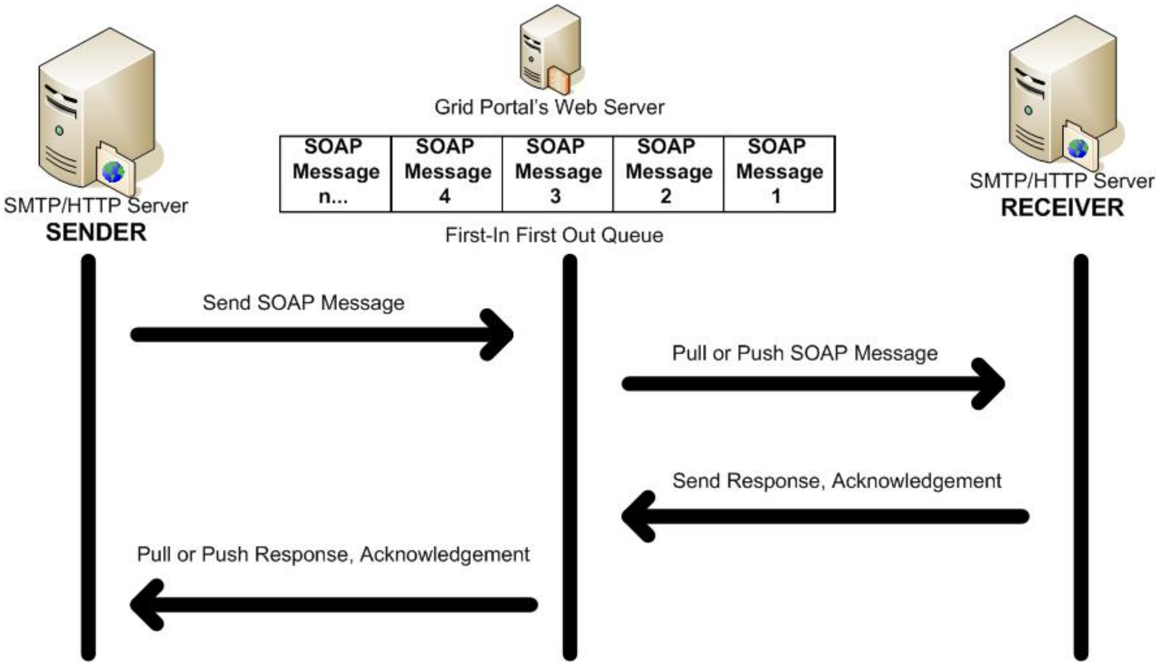

**Figure 3.7: Asynchronous SOAP Messaging**

### *3.2.3.4 Reliable Message Delivery*

Asynchronous messaging allows message delivery guarantee to be configured through Quality of Service (QoS) semantics. A typical message delivery configuration can be *once-and-once-only*, *at-least-once* or *at-most-once* (Mahmoud, 2004). The message response or acknowledgement can also be configured, as well as the number of times a retry should be attempted in case of an error or failure of delivery. The message is kept on the queue until its Time-To-Live (TTL) expires. The messages on the Web server queue should be stored on a non-volatile platform like a hard disk in order to guarantee their delivery. This affords the SMTP/HTTP server (sender) to 'fire-and-forget' the messages with assurance that the message will be delivered. The message delivery guarantee can be extended through certified message delivery report from the SMTP/HTTP server (receiver), to confirm that the message has been received and consumed.

The Web server's state is replicated across numerous servers using a technique known as *clustering*. This prevents the server from being unavailable in case of temporary runtime failures. Clustering allows an application to be reliable and scalable by seamlessly distributing it across multiple servers (Ibid: p. 47). The application appears as a single virtual entity to the users. A group of clusters lessens the workloads on the servers thereby providing high reliability and scalability.

While clustering is concerned with distributing server applications, *load balancing* on the other hand, allows a user's workload to be spread across multiple servers (NB: server refers to a server instance on software, physical hardware or both). To achieve an accurately load balanced system; the distribution of work across the servers should be dynamic, with more work allocated to the least loaded server (Medjahed *et al*., 2003).

In order to achieve a true load balanced system, two approaches known as *pull* and *push* can be employed. The pull approach places workload request messages on a point-to-point queue, and servers pull the messages from the queue once they are ready to process them. The push approach on the other hand, uses an algorithm to guesstimate the least-burdened server where the workload request messages will be pushed to. The pull approach provides a more accurate load balanced model than the push model, as the workload is distributed efficiently as compared to estimates which are not perfect all the time (Mahmoud, 2004, p. 49).

### 3.2.4 Web Services

An XML Web service can be described as a software system, designed to support interoperable machine-to-machine interaction over a network. It has an interface, described as a machine-processable format (specifically WSDL). Systems can interact with the Web service as prescribed by its interface using SOAP messages, usually conveyed using HTTP (Booth *et al*., 2004). The fundamental idea behind Web services is to develop software applications as services (Erl, 2004). This concept is based on a defined set of technologies, supported by open industry standards. These standards work together to facilitate interoperability among distributed and heterogeneous systems such as (HIV/AIDS) VIOISs. Since Web services are based on standard interfaces, they can communicate even if they are running on different operating systems (i.e., platform-independence), supplied by different vendors (i.e., vendor-independence) and are written in different languages (i.e., language-independence) (Li & Baker, 2005). Therefore, Web services provide an excellent approach for building distributed applications that must incorporate diverse systems over a network (Pullen *et al.*, 2004).

Web services are loosely-coupled; hence a service can exist independent of other services that comprise an application. Moreover the service can be individually modified or developed from scratch and deployed into an application. A Web service is identified by a uniform resource identifier (URI) (Booth *et al.*, 2004). The URI interfaces and bindings should be able to be discovered, defined and described by XML objects. Thus, publishing and discovery of Web services can be achieved through a standard infrastructure that provides a Web services description mechanism, a Web services discovery mechanism, Web service directories and open-standard wire formats.

### 3.2.5 Web Services Infrastructure

Web services utilize the Universal Description Discovery and Integration (UDDI) specification to implement Web Service Directories. UDDI provides a platform-independent, central location for registering, querying/discovering, and integrating Web services (Erl, 2004; Colgrave *et al.*, 2002). The Web Services Description Language (WSDL) standard, on the other hand, provides a discovery and description mechanism for Web services. UDDI and WSDL provide a Web service discovery mechanism that locates documents that are linked or describe a particular Web service.

### 3.2.6 WSDL

WSDL is utilized to fully describe a Web service, for example, what actions a service can perform, where the service resides and how it can be invoked. Services in WSDL are defined as a set of network endpoints or ports that utilize an RPC-based mechanism or a document-oriented message exchange for communication between the service requestor and the service-provider. In an RPC-based operation, the SOAP messages contain parameters and return values, while in a document-oriented operation such as (HIV/AIDS) VIOISs, the SOAP messages contain XML documents. RPC-based message exchanging utilizes synchronous communication, while document-oriented message exchanging usually utilizes asynchronous communication.

#### *3.2.6.1 Data types*

The data types part contains data type definitions which are relevant for message exchanging (Li & Baker, 2005, p. 24). WSDL utilizes XML XSD as the default data in order to achieve maximum interoperability and platform neutrality (Christensen *et al*., 2001; Colgrave *et al*., 2002). This part is extensible, thus it can consist of arbitrary subsidiary elements in order for general data types to be built.

*<message>*

The <message> element, which is analogous to a method call or the parameters of a function in a traditional programming language, is responsible for defining data elements of an operation. Each message can contain one or more parts.

*<portType>*

The <portType> element, which is analogous to a C++ class or a Java interface, is the core part of a WSDL document. It is responsible for defining a set of abstract operations provided by a service. Each operation utilizes messages defined in the <message> element to specify its inputs and outputs.

*<binding>*

The <binding> element identifies a concrete protocol and data format for the operations and messages defined by a particular <portType>. A particular <portType> may possess an arbitrary number of bindings (i.e., a binding can be RPC-based or documented oriented). The most commonly used mechanism for transmitting messages between a service client and the service itself is SOAP over HTTP.

*<port>*

The <port> element defines an individual service endpoint by specifying a single address for a binding.

*<service>*

The <service> element represents a set of related elements. Ports within a service possess the following characteristics (Li & Baker, 2005, p. 26):

- None of the ports are able to communicate with each other.
- If a service consists of several ports which share a <portType>, but utilize different bindings or addresses, these are alternative ports where the port provides semantically equivalent behaviour. This enables a consumer of a WSDL document to select a particular port(s) to communicate with, based on some criteria (for example protocol or distance).

### 3.2.7 UDDI

A service-provider utilizes UDDI to advertise the services which are being made available while a client utilizes UDDI to find the appropriate service(s) for its purpose (Erl, 2004; Colgrave *et al*., 2002). A UDDI registry is analogous to a CORBA trader service, or it can be seen as a Domain Name Server (DNS) service for business applications. There are two kinds of players that utilize a UDDI registry: businesses that want to publish a service, and clients who want certain services, and then use them through some binding process. Data in UDDI can be organised in the following ways (Bellwood *et al*., 2002):

- White pages: This information includes general information about a service-provider, for instance its name, contact information and other identifiers.
- Yellow pages: With this information, a Web service is described utilizing different categorizations (taxonomies) and allows the Web service to be

discovered by others based on its categorization (for example, a blood cell test or red blood cell test).

- Green pages: These contain technical information about a Web service, usually with a reference to an external WSDL document of the service, which allows the client to understand how to interact with the service.

A UDDI registry exposes a set of APIs in the form of SOAP-based Web services. The API contains Inquiry and Publishing APIs for service discovery and service publication.

### 3.2.8 Web Services Implementations

Web services are based on a set of specifications. Presently there are numerous Web service implementations that can be utilized to construct distributed applications. To utilize Web services, there are three aspects that need to be considered (Li & Baker, 2005):

- A *programming model* that specifies how a client code should be written to access Web services, how service implementations are written, how other parts of the SOAP specification are handled, for instance, headers and attachments.
- A *deployment model,* which is a framework that is utilized to deploy a service and provide a Web Service Deployment Descriptor (a WSDD file) that is used to map the implementation of the service to SOAP messages.
- A *SOAP engine* that receives SOAP messages and invokes Web service implementations.

A description of three frameworks –J2EE, .Net and Apache Axis– that implement Web services applications follows.

#### *3.2.8.1 J2EE*

J2EE refers to standard that is utilized to develop, build and deploy Java-based applications (JEE5, 2006). These include traditional Web sites, software components, or packaged applications and have been extended to include support building of XML-based Web services. J2EE provides the following APIs for Web services:

- The Java API for XML Processing (JAXP) – responsible for processing XML documents utilizing various parsers.
- The Java Architecture for XML Binding (JAXB) – responsible for processing XML documents utilizing schema-derived JavaBeans component classes.
- The Java API for XML-based RPC (JAX-RPC) – a standard for RPC. It offers APIs for XML RPC invocation and utilizes base-level protocol bindings with SOAP/HTTP, although it is not limited to HTTP.
- The Java API for XML Messaging (JAXM) and SOAP with Attachments API for Java (SAAJ) – responsible for sending SOAP messages over the Web in a standard way.
- The Java API for XML Registries (JAXR) - provides a standard way to interact with business UDDI registries.

Latest J2EE releases have, however, been upgraded and simplified, with the 2 and the dot being dropped from the release name. According to the Sun Developer site (JEE5, 2006); the latest Java platform for the enterprise is called Java Platform, Enterprise Edition 5 (Java EE 5).

#### *3.2.8.2 Apache Axis*

Apache Axis is a SOAP engine, which can be utilized to exchange SOAP messages between clients and services (Axis, 2006). Furthermore, it offers support for WSDL operations, for instance, Java2WSDL can be utilized to create a WSDL document from a Java interface, and WSDL2Java can be utilized to create a client-side stub and a server-side skeleton, based on the WSDL document. The publishing and discovery of services is not supported in Axis. However, Axis and UDDI4Java can be utilized for this purpose.

#### *3.2.8.3 Microsoft .Net*

.Net is a Microsoft platform for building, deploying, and managing Web services applications (Microsoft, 2005). Analogous to a J2EE Web service, a .Net Web service supports the WSDL specification and utilizes a WSDL document to describe itself. However, in order to uniquely identify the Web service's endpoint, the WSDL file contains an XML namespace.

.Net offers a client-side component, which enables a client to invoke a Web service defined by WSDL. Moreover, it offers a server-side component which maps Web service operations to a COM-object method call, as defined by the WSDL interface and a Web Service Meta Language (WSML), which is utilized in Microsoft's SOAP implementation. Web services can be published through a UDDI registry or utilizing DISCO files. DISCO is a Microsoft publishing/discovery technology built into .Net.

### 3.2.9 The Benefits of Web Services on the Grid

Web services are an XML-based open standard, which can be utilized to build distributed applications in a heterogeneous computing environment, such as the (HIV/AIDS) VIOIS Grid environment. Web services are independent of vendors, programming languages, platforms and locations (Pullen *et al*., 2004). Web services can be described, published and dynamically discovered and bound to WSDL, which provides a rich description of the interfaces (Booth *et al*., 2004). The technologies that support Web services offer a favourable platform for the integration of services in heterogeneous systems (Erl, 2004). In order to benefit from Web services, the collaborative HIV/AIDS research Grid can make the most out of the following factors (Li & Baker, 2005, p. 33):

- The (HIV/AIDS) VIOIS Grid needs support for dynamically discovering and composing Grid services in heterogeneous environments. This requires mechanisms that register and discover interface definitions and endpoint implementation descriptions in order to dynamically create proxies, based on (potentially multiple) bindings of specific interfaces. This requirement is attained through WSDL, which offers a standard mechanism that defines interface definitions, independent of their embodiment within a particular binding (transport protocol and data encoding format).

- Technologies that describe Web services are based on internationally recognised standards. Therefore, frameworks and implementations (such as the (HIV/AIDS) VIOIS Grid) based on Web services can exploit various tools and extended services, for instance, WSDL processors, which are able to generate bindings for diverse environments, such as Web Services Invocation Framework (WSIF) (Mukhi, 2001), workflow systems that use WSDL, and hosting environments for Web services (e.g., Apache Axis and Microsoft .NET)

### 3.2.10 OGSA

The Open Grid Services Architecture (OGSA) adopts Web service standards as the basis for a service oriented Grid framework. The framework allows for arbitrary services to be defined, discovered and invoked through their interfaces without having to understand their implementations. Therefore, the framework presents a base for composition, virtualisation and interoperability (Foster *et al*., 2006). OGSA represents everything in (HIV/AIDS) VIOISs as Grid service, which is essentially a stateful Web service which has standard interfaces and protocol binding.

#### *3.2.10.1 Grid Service*

A Grid service comprises at least one Web service that provides a set of well-defined open standards-based interfaces that follow specific conventions. The interfaces are concerned with the discovery, dynamic service creation, inspection, notification, lifetime management, and manageability of the distributed and often long-lived state that is commonly required in distributed applications in (HIV/AIDS) VIOISs (Foster *et al*., 2002; Czajkowski *et al*., 2004). Reliable invocation and authentication should also be addressed, as they are seen as service protocol bindings, although they are external to the core Grid service definition.

Grid services extend the WSDL's <portType> into a <serviceType>, and include additional information relating to versioning. A Grid service can be implemented through one or more interfaces, and each interface can specify a set of operations which can be invoked by exchanging a defined sequence of messages. The interfaces and conventions of a Grid service deal specifically with the managing transient service instances state. A transient service instance is an instantiation of a Grid service, which can be created and destroyed dynamically. Examples of transient service instances include a data mining operation, a query against a data warehouse, a running data transfer, network bandwidth allocation, and an advance reservation for processing capability (Foster *et al*., 2002). In the Open Grid Services Architecture, the collaborative HIV/AIDS research Grid is viewed as an extensible set of Grid services.

Unless otherwise stated, the information in the following sub-sections (3.2.10.2 and 3.2.10.3) is sourced from Foster *et al*. (2006).

### *3.2.10.2 OGSA Framework*

OGSA was developed to support the seamless use, management, integration and virtualization of distributed, heterogeneous resources and services within dynamic virtual organizations such as (HIV/AIDS) VIOISs. Owing to the success of OGSA is standardization, which ensures that the various resources and services in (HIV/AIDS) VIOISs can be created, published, discovered, accessed, manipulated, accounted for, allocated, monitored, etc., and on the whole managed as a single virtual system, although they are provided by different service-providers and operate on different platforms.

Standardization also plays a key role in creating portable, interoperable and reusable components and systems, while augmenting the development of scalable, robust, and secure Grid systems by facilitating the implementation of good practices. OGSA addresses the need for standardization through specifying a set of core capabilities and behaviours which address key concerns in Grid systems. The key concerns were motivated by functional and non-functional requirements for a Grid system driven by a set of use cases (Foster, Gannon, Kishimoto & Von Reich, 2004; Von Reich, 2004). The functional and non-functional requirements include job execution, interoperability, quality of service (QoS) assurance, resource sharing across organizations, data services, availability, security, scalability, etc. In order to meet the standardisation needs in (HIV/AIDS) VIOISs, OGSA specifies core capabilities such as, Execution Management, Data, Resource Management, Security, Self-Management, and Information services which are utilized to address key concerns. However, it must be noted that the entire set of OGSA capabilities does not have to be available on the system, for it to operate.

### *3.2.10.3 Core Services*

According to Foster *et al*. (2006), Execution Management Services (OGSA-EMS) are responsible for the instantiation, subsequent management, and completion of units of work. EMS services facilitate the coordinated access to underlying resources regardless of where they are located or their access mechanisms. The EMS is beneficial in the author's problem domain where there are various resources such as data-storage facilities, computation resources, etc., located in different institutions with diverse access mechanisms. In particular, EMS deals with problems arising from the execution of units of work, as well as the placement, provisioning, and lifetime management of units of work. Some of the problems addressed comprise finding execution candidate locations, selecting execution location, preparing for execution, initiating the execution and managing the execution. EMS specifies three classes of services namely: resources, job management, and resource selection services.

Resources in EMS are utilized to model processing, executables, storage, resource management, and provisioning. Thus, resources contain running entities such as jobs or running services, and hence they are known as service containers. The service containers have resources properties that depict static information comprising the kind of executables they can take - OS version, policies, libraries installed, and security environment - in addition to dynamic information.

The EMS job management service consists of a job and a job manager. A job is the nominal unit of work that is managed. It represents a unit of work's manageability facet, which is particularly different from the actual running application aspect or the execution facet. A job is identified by a WS-Addressing EPR. A job request will culminate in an instant creation although there might not be any resources committed. The job keeps track of execution state (for example, started, suspended, terminated, restarted, completed), job requirements, resource commitments and agreements, etc., by storing them in a job document. A job document describes the state of the job, for example, a description of the submission, the agreements that the job has acquired, the status of the job, metadata about the user (credentials, etc.), and the number of times the job has been started. The job document can be exposed as a resource property of the job.

A job manager, on the other hand, is a higher-level service which encapsulates all of the facets of executing a job, or a set of jobs, from creation until termination. A set of jobs can be structured or unstructured. The JM deals with orchestrating the services utilised to start a job or a set of jobs, through, for instance, negotiating agreements, interacting with containers, and configuring monitoring and logging services.

The final EMS class, resource selection services, includes Execution Planning Services (EPS), Candidate Set Generator (CSG), and Reservation services.

- An EPS refers to a service which builds schedules, wherein a schedule refers to a mapping or relation between resources and services, probably with time constraints. An EPS' distinctive challenge is to optimize some objective function, for instance, execution time, reliability, availability, etc. As an example, an EPS will initially call a CSG in order to get a set of resources, and thereafter obtain more current information about the resources from information, and finally generate a schedule by executing an optimization function. The enactment of the schedule lies with the JM.

- A CSG is responsible for determining a set of resources where a unit of work can possibly execute as opposed where it will actually execute. The CSG produces a list of EPRs of containers where the job described (with a Job Submission Description Language (JSDL) (Anjomshoaa *et al*., 2005) can probably execute. A CSG is utilized mainly by EPSs, and also other services for instance JMs that carry out functions similar to those of an EPS.

- Reservation services are tasked with managing the reservation of resources, withdrawal of reservation, etc. This is not necessarily a separate service, but can rather be an interface that acquires and manages reservations from containers and other resources. A reservation in most cases will be an agreement document that is signed. Reservation services will typically be utilised by various services, for instance, a JM may generate reservations a set of jobs that it manages, or an EPS might utilise reservations in order to guarantee the execution plan for a particular job.

The core services are built on Web-service standards, with semantics, additions, extensions and modifications that are relevant to Grids. Web Services Description Language (WSDL) is utilized to define service interfaces while XML is the *lingua franca* for description and representation. The primary message exchange format for OGSA services is SOAP. Security services such as authorisation, authentication and message protection are achieved through WS-Security standard protocols that allow OGSA security requests to transmit tokens securely. XML encryption and digital signatures are utilized to provide end-to-end message protection along with, or replacing, point-to-point transport-level security, for instance, IPSec (Microsoft, 2000; Belani & Mookhey, 2005) or Transport Layer Security (TLS) (Dierks & Allen, 1999).

On top of message-level security, security components should be individually rendered as services in order to realise an interoperable and composable infrastructure. These include, for instance, expression of security assertions through Security Assertion Markup Language (SAML) (Hughes & Maler, 2004) and access-control descriptions through eXtensible Access Control Markup Language (XACML) (Kay, 2003).

The WS-Resource Framework (WSRF) specification provides the capability to model, access and manage state; to group services; and to express faults. The WS-Notification specification defines notification techniques that enable subscriptions and notifications to subsequent changes in state components. WSRF and WS-Notification specification are discussed in the next section.

### 3.2.11 WSRF

The Web Service Resource Framework (WSRF) defines standard Web-service mechanisms that create, access, inspect, manage state, monitor changes and destroy Web Service (WS-) Resources (Foster *et al*., 2006; Czajkowski *et al*., 2004$^1$). A WS-Resource is composed of a Web service and a stateful resource, which is, (a) an articulation of the relationship between an XML document with defined type and a Web services portType, and (b) addressed and accessed in terms of the implied resource pattern, thus using WS-Addressing endpoint references (Czajkowski *et al*., 2004$^2$).

WS-Addressing provides transport-neutral mechanisms to address Web services and messages. Specifically, this specification defines XML elements to identify Web services endpoints and to secure end-to-end endpoint identification in messages (BEA *et al*., 2004; Bosworth *et al*., 2004). WS-Addressing defines two constructs, message addressing properties and endpoint references. They standardize the information typically provided by transport protocols and messaging systems in a way that is independent of any particular transport or messaging system (Gudgin & Hadley, 2005; Box *et al*., 2004). An endpoint reference can possess Web Service metadata, for example, service description information and reference properties.

WSRF is made up of a set of WS-Specifications. These specifications are compliant with the WS-Interoperability (WS-I) Basic Profile (Ballinger *et al.*, 2004), thus any WS-I-compliant client will be able to interact with services that support these specifications. From the client's point of view, WSRF specifies conventions that are utilized to query state, which make it easier to manage and use services that follow these conventions. A brief discussion of WS-Specifications ensues.

#### *3.2.11.1 WS-ResourceLifetime*

The WS-ResourceLifetime specification (Srinivasan & Banks, 2006) defines message exchanges for service requestors to request Web services to destroy associated WS-Resources, and resource properties that can be utilized for inspecting and monitoring a WS-Resource's lifetime. The destruction of the WS-Resource can take place immediately or can be scheduled to happen after a certain time. The specification also defines a way by which the WS-Resource can start a self-destruct action if a certain period of time has expired (Li & Baker, 2005).

#### *3.2.11.2 WS-ResourceProperties*

The WS-ResourceProperties specification (Graham & Treadwell, 2006) defines the means by which the definition of the properties of a WS-Resource may be declared as part of a Web-service interface. The WS-Resource's properties declaration represents a view on or a projection of the WS-Resource's state. The WS-Resource is represented in this projection as an implicit stateful resource type which specifies a basis for accessing the WS-Resource properties through Web service interfaces.

WS-ResourceProperties also specifies a standard set of message exchanges that enable a requestor to update or query the property values of the implied WS-Resource. The WS-Resource projection defines a set of properties that are associated with the Web service interface and specify the constraints on the valid contents of these message exchanges.

#### *3.2.11.3 WS-BaseFaults*

The WS-BaseFaults specification (Liu & Meder, 2006) defines an XML schema for base faults, together with the rules that specify how these fault types are utilized and extended by Web services. WS-BaseFaults aim to standardize the terminology, concepts, XML types, and WSDL usage of a base fault type for Web services interfaces. If Web service fault messages are specified in a common way, support for problem determination and fault management will be augmented. Thus, it will be easier for requestors to understand faults, because fault information from the different interfaces will be consistent.

#### *3.2.11.4 WS-ServiceGroup*

The WS-ServiceGroup specification (Maguire, Snelling & Banks, 2006) defines a way that Web services and WS-Resources can be aggregated or grouped together. A ServiceGroup refers to a WS-Resource which represents a collection of Web services. The individual services that make up the ServiceGroup are known as the ServiceGroup's members or its membership. The ServiceGroup membership constraints, membership rules and classifications are expressed through the resource property model, as defined in the WS-ResourceProperties specification.

#### *3.2.11.5 WS-Notification*

Although WS-Notification is not part of the WSRF, it has generically being referenced by various WSRF specifications. Thus, there is at least some implementation of WS-Notification functionality in a typical WSRF implementation. The WS-Notification specification (IBM *et al*., 2006) defines mechanisms for event subscription and notification utilizing a topic-based publish/subscribe pattern. It consists of:

- Standard message exchanges that will be implemented by service-providers that desire to partake in Notifications.

- Standard message exchanges for a notification broker service-provider. The notification broker interface (NotificationBroker) is responsible for defining a standard set of message exchanges that describe a message broker, thereby providing an intermediary between Publishers and Subscribers on a collection of Topics.
- Operational requirements expected of service-providers and requestors that partake in notifications, and an XML model that depicts topics.

The WS-Notification specification has been divided into three specific specifications, WS-Topics (Vambenepe, Graham & Niblett, 2006), WS-BaseNotification (Graham, Hull & Murray, 2006) and WS-BrokeredNotification (Chappell & Liu, 2006). The WS-BaseNotification is the base specification that other WS-Notification specifications build on. It defines Web services interfaces that are utilized by NotificationProducers and NotificationConsumers in exchanging messages. The WS-BrokeredNotification on the other hand, defines Web services interfaces for NotificationBrokers, which act as both NotificationProducers and NotificationConsumers as defined in the WS-BaseNotification. A NotificationBroker may subscribe to notifications which are distributed by NotificationProducers and also deliver notification messages to NotificationConsumers. Moreover, a NotificationBroker must support hierarchical topics, as well as the ConcreteTopicPath topic expression dialects which are defined in WS-Topics. The WS-Topics specification defines a mechanism to organize and categorize items of interest for subscription referred to as “topics”. It offers subscribers a convenient way of reasoning about WS-BaseNotification of interest. There are three topic expression dialects, defined in WS-Topics, which can be utilized as subscription expressions. Furthermore, it specifies an XML model that is utilized to describe metadata associated with topics.

### 3.2.12 WS-Resource

A Web service, which can be seen as a stateless message processor, can be associated with stateful resources in order to make it a WS-Resource. A WS-Resource consists of the following attributes (Li & Baker, 2005, p. 63):

- It is a stateful resource that can be utilized by a Web service as a data context to exchange messages.
- It can be created, identified and destroyed. A WS-Resource may have multiple identifiers within the same Web service or within various Web services.
- A stateful WS-Resource type can be associated with a Web service interface definition to enable well-formed queries against the WS-Resource through its service interface, and a stateful WS-Resource's status can be queried and modified through service message exchanges.

The WSRF makes no effort in defining the message exchange utilized to request the creation of new WS-Resources. As an alternative, it simply states that new WS-Resources can be created through a WS-Resource factory pattern, as defined in the WS-ResourceLifetime specification. A WS-Resource factory refers to any Web service that has the capability of making one or more WS-Resources come into existence. Typically, a response message of a factory operation consists of at least one endpoint reference that refers to the newly created WS-Resource.

### 3.2.13 Implied WS-Resource pattern

The implied WS-Resource pattern term is utilized to describe a specific kind of relationship between a Web service and one or more stateful WS-Resources (Li & Baker, 2005; Czajkowski *et al.*, 2004$^{2}$).

- The term "*implied*" indicates that if a client is accessing a Web service, the Web service will return a WS-Addressing endpoint reference that is utilized to refer to the WS-Resources associated with the Web service. Every individual WS-Resource has an identifier (ID) that is utilized to manage its state. The IDs of the WS-Resources that a client will access are automatically encapsulated in the endpoint reference and returned to the client. A WS-Resource ID is only utilized by a Web service as an implicit input to locate a specific WS-Resource, and as a result it is opaque to the client.

- The term "*pattern*" means that the relationship between Web services and stateful WS-Resources is codified through a set of conventions on existing Web services technologies, particularly XML, WSDL and WS-Addressing.

### 3.2.14 WSRF and OGSA

OGSA is based on standard Web services, and it specifies standard services needed to build service-oriented Grid applications, through the GGF OGSA working group (OGSA-WG). OGSA is only an architecture and does not specify how Grid services should be implemented. Therefore, OGSA can be enhanced with WRSF as depicted on Figure 3.8. WSRF provides a technical specification that implements Grid services as defined in the OGSA specification. OGSA platform services are Grid base services which are utilized for submissions of jobs, authentication and authorisation of users, information services, transfer and replication of data, and data access and integration. WRSF services on the other hand, are Grid core services which are utilized for the creation, destruction, and life cycle management of stateful Web services resources. The current WSRF implementations include Globus Toolkit Version 4 (GT4) (Foster, 2005), WSRF.NET (Humphrey *et al*., 2004), pyGridWare (LBNL, 2006), and WSRF::Lite (SVE, 2005). GT4 is the widely accepted WRSF implementation and will be discussed in the next chapter.

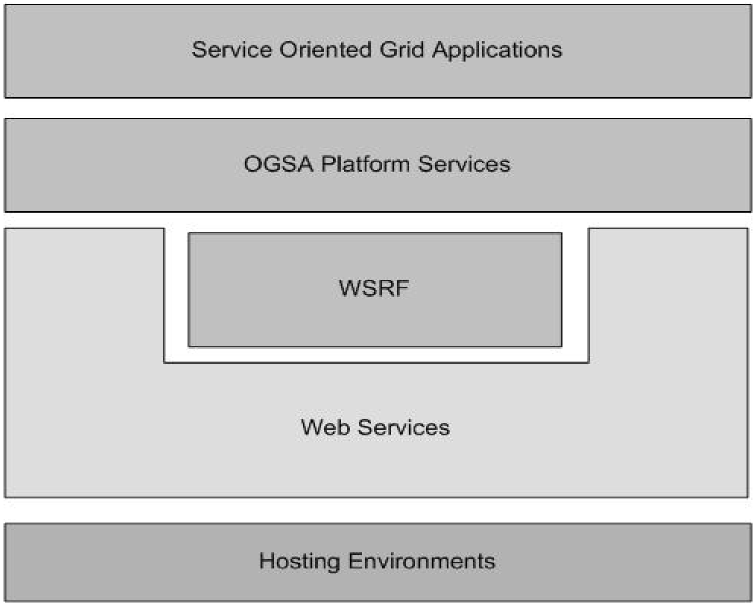

**Figure 3.8: Enriching OGSA with WSRF (Based on Li & Baker, 2005)**

## 3.3 Conclusion

This chapter introduced Grid computing which provides a collaboration platform through enabling access to distributed computation and data-storage facilities in (HIV/AIDS) VIOISs. In order to achieve this, open standards that facilitate interoperability, such as, XML, SOAP, WSDL, UDDI and Web services are utilized.

Grid computing has evolved from several non-interoperable distributed systems, to a seamless service oriented, collaborative, and dynamic virtual environment, utilizing Web service standards. The Open Grid Services Architecture (OGSA) specification, which is the de facto standard for building service-oriented Grid systems, is endorsed for building (HIV/AIDS) VIOISs. OGSA services are implemented through the Web Service Resource Framework (WSRF) specification. The Globus Toolkit Version 4 (GT4) is the widely accepted implementation of WRSF specification. The next chapter discusses this implementation of WSRF.

# Chapter 4

# Grid Service Implementation

## INTRODUCTION

The Web Service Resource Framework (WSRF) provides a technical specification for implementing Grid services that are defined by the Open Grid Services Architecture (OGSA) specification. The Globus Toolkit Version 4 (GT4) is the widely accepted implementation of WRSF specification. GT4 offers execution management, information services, data management and security services in a Grid environment.

This chapter discusses how execution management is achieved through GT4's Grid Resource Allocation and Management (GRAM). Thereafter, Grid monitoring and the implementation of information services via the Monitoring and Discovery System (MDS4) will be discussed. GT4's data management capabilities via Reliable File Transfer, Grid File Transfer Protocol and Replica Location Service will then be discussed, followed by a discussion of the Grid Security Infrastructure, which is the basis for Grid security.

## 4.1 The Globus Toolkit 4

The Globus Toolkit is an open-source software initiative, providing a set of tools specifications for grid developers. The toolkit, which is under its fourth major revision (GT4), is currently the *de facto* standard for Grid service implementations. The Globus toolkit is made up of a number of core components that are utilized to develop Grid systems and applications. In order to support the development of efficient Grid services, GT4 provides Execution Management (Grid Resource Allocation and Management (GRAM)), Information Service (Monitoring and Discovery System (MDS4)), Data Management (Reliable File Transfer (RFT), Grid File Transfer Protocol (GridFTP), Replica Location Service (RLS) and Security

(Authentication and Authorization, Grid Security Infrastructure (GSI)) tools. These components will be discussed in the following sections.

## 4.2 Grid Execution Management

### 4.2.1 Grid Resource Allocation and Management

The Grid Resource Allocation and Management (GRAM) system and its client software were developed to provide the following features (Foster, 2005):

- State: The state of the computational resource and/or its associated file systems where a submitted job is running may be affected by the input/output operations of the job. Therefore, it is imperative to employ "exactly once" execution semantics: users should not merely resubmit a request because they did not receive an acknowledgement message, because a job may be completed while its acknowledgement was lost.
- User Executables: Enabling users to submit programs they need to execute.
- Staging of input and output: It is imperative to manage the staging of input data, output data or any executables as they may be remote, large and/or being shared by different invocations.
- Streaming output: Enabling users to interact with the running jobs, thus, allowing them access to streaming output data.
- Control: Enabling users to manage running jobs, for example, terminating a job which consumes a lot of resources.
- Schedulers: Implementing allocation and prioritization policies on large computing resources whilst optimizing all the submitted jobs' execution to attain performance and efficiency.
- Monitoring: Schedulers and data stating bring about potential complexities to the state of the submitted jobs. Thus, at times a job will not be simply running, completed, or failed, but rather may be staging, suspended, pending, and so forth. It is important for users to be able to query the status of a job and/or subscribe to the job in order to receive notifications about the job's status.

The GRAM interface provides advanced management functionality for jobs submitted successfully by generating a stateful entity, called a ManagedJob, on the compute host (Li & Baker, 2005; Foster, 2005). A ManagedJob’s lifetime is somewhat longer than that of the associated job, in order to allow users to determine the state of the job, even when it has been terminated. Following a successful “submit” operation, GRAM returns a handle, known as WS-Addressing Endpoint Reference (Bosworth *et al*., 2004) or EPR, for the new ManagedJob. Users/clients can utilize this handle to query the status of job, terminate it, and/or subscribe to it in order to receive notifications about the job’s status and output. Moreover, a client can share this handle with other users, and if they are authorized can carry out the same operations. The ManagedJob construct essentially ensures that GRAM provides the advanced features mentioned previously.

## 4.3 Grid Monitoring

The Grid is essentially a complex, globally distributed system that consists of large sets of diverse, geographically dispersed components and is utilized by numerous applications. The components referred to here comprise all the software and hardware services and resources that are required by applications. The multiplicity of these components and the large number of their users make them vulnerable to failure, faults and excessive loads. Thus, there is a need to monitor these components in order to detect conditions, which may lead to faults, bottlenecks or failures. Grid monitoring plays a vital role in attaining a robust, reliable and efficient environment.

Unless otherwise stated, the information in sub-section 4.3.1 and 4.3.2 is sourced from Li & Baker (2005).

### 4.3.1 Grid Monitoring Architecture (GMA)

The Grid Monitoring Architecture (GMA) is an open architecture from the Grid Monitoring Architecture Working Group (GMA-WG) (Aydt, Gunter, & Smith, 2000). The GMA is made up of three components (Tierney *et al*., 2000):

- A Directory Service that supports the publication and discovery of consumers, producers and monitoring data (events);
- Producers, which are the sensors that generate performance data;
- Consumers that access and utilize performance data.

#### *4.3.1.1 The Directory Service*

The GMA Directory Service provides producers and consumers a place for the publication of the event types they produce or consume. Moreover, they can publish static values for some event data elements, in order to restrict the range of data they will produce or consume. This enables other producers and consumers to discover the types of events that are currently available, the characteristics of their data, and the sources or sinks that will generate or accept each type of data.

The Directory supports the following functions:

- *Authorise a search*: Establish the identity (through authentication) of a consumer that wants to undertake a search.
- *Authorise a modification*: Establish the identity of a consumer that wishes to modify entries.
- *Add*: Insert a new record in the directory.
- *Update*: Change the state of a record in the directory.
- *Remove*: Remove a record from the directory
- *Search*: Execute a search for a consumer or consumer of a particular type, probably with fixed values for some of the event elements. A consumer can specify whether only one result, or more if available, should be returned. An optional extension would enable a consumer to obtain multiple results, one element at a time, utilizing a “get next” query is subsequent searches.

There can be one central Directory Service or a Directory Service Gateway controlling multiple services in a Grid monitoring system. An extended Grid Monitoring Architecture with multiple Directory Services is illustrated in Figure 4.1.

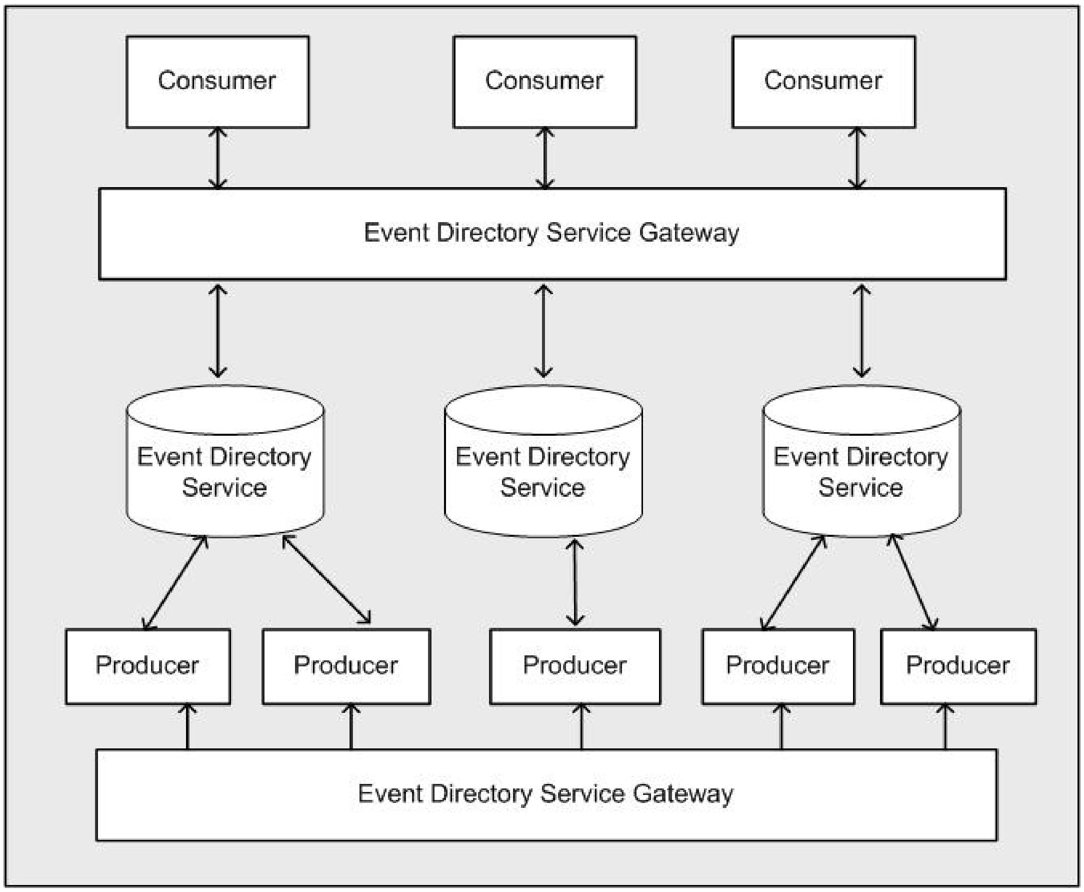

**Figure 4.1: Grid Monitoring Architecture (Li & Baker, 2005)**

*4.3.1.2 Consumer*

A consumer refers to any program that can receive monitoring data (events) from a producer. A consumer supports the following steps:

1. *Locate events*: Perform a search on a schema repository for a new event type. The schema repository can be part of the GMA Directory Service.
2. *Locate Producers*: Perform a search on the Directory Service to find a suitable producer.
3. *Initiate a query*: Consumers request event(s) from a producer, which are delivered as part of the reply.
4. *Initiate a subscription*: Consumers can subscribe for events they are interested in to the producer.
5. *Initiate an unsubscribe*: Consumers can terminate a subscription to the producer.

6. *Register*: Consumers can add/update/remove one or more entries in the Directory Service that describe events that the consumer will accept from producers.
7. *Accept query*: Consumers can also accept a query request from a producer. The query will also contain the response.
8. *Accept subscribe*: Consumers accept a subscribe request from a producer. The producer will be notified automatically once there are requests from the consumers.
9. *Accept unsubscribe*: Consumers accept an unsubscribe request from a producer. If this succeeds, no more events will be accepted for this subscription.

#### *4.3.1.3 Producers*

A producer refers to a software component that conveys monitoring data (events) to a consumer. A producer supports the following steps:

1. *Locate event*: Perform a search on the Event Directory Service for the description of an event.
2. *Locate Consumers*: Perform a search on the Event Directory Service to find a consumer.
3. *Register*: Producers can add/update/remove one or more entries in the Event Directory Service that describe events that the producer will accept from a consumer.
4. *Accept query*: Producers can accept a query request from a consumer. One or more event(s) are returned in the reply.
5. *Accept subscribe*: Producers accept a subscribe request from a consumer. Further details about the event stream are returned in the reply.
6. *Accept unsubscribe*: Producers accept an unsubscribe request from a consumer. If this succeeds, no more events will be sent for this subscription.
7. *Initiate a query*: Send a single set of event(s) to a consumer as part of a query “request”.

8. *Initiate subscribe*: Request to send events to consumers, which are delivered in a stream. Further details about the event stream are returned in the reply.
9. *Initiate unsubscribe*: Producer can terminate a subscription to the consumer. If this succeeds, no more events will be sent for this subscription.

### 4.3.2 Monitoring Data

The data utilized for monitoring purposes needs to have timing, flow and content information associated with it.

#### *4.3.2.1 Time-related data*

- Time-stamped dynamic data occurs within a flow among several regular messages and temporal information, which may be provided through a counter related to the sampling rate (frequency). This data consists of performance events and status monitoring.
- Time-stamped asynchronous data is utilized to illustrate an event happening. This data is utilized for alerts and checkpoint notifications.
- Non-time-related data comprises static information, for instance, type and version of OS, characteristics of hardware or the update time of monitoring information. The term "static", as discussed here, stems from the fact that the data remains almost constant, and is typically operator-updated. While "dynamic" refers to information, such as performance or status, which changes over time.

#### *4.3.2.2 Information flow data*

- Direct producer-consumer flow does not involve a central component when transferring data. A monitor may be active or passive depending on where the communication was initiated between the producer and the consumer. There are three interactions specified by the GMA document:
  - Publish/Subscribe
  - Query/Response
  - Notification.

- Indirect data distribution through a centralized repository. This can be utilized for static information, as the amount of data is relatively small and it is rarely updated, and also because the publication/discovery process cost is comparable to that of information gathering. Interaction in this case is through the initial notification of the producers to the directory service, and consumers can select data from this source too.
- A follow-up the workflow's path, where monitoring information is produced and stored locally. The data is tagged in order associate it with a specific part of a workflow. When the job ends, the monitoring and tag, along with the workflow output, can be returned to a consumer or discarded. A consumer can collect monitoring data and tags through following the job's path, which can be coalesced to provide a summarized view, or independently sent to the consumer.

#### *4.3.2.3 Monitoring Categories*

- Static monitoring occurs when the cost of information gathering, as pertaining to used bandwidth and time, is less or comparable to the cost of resource discovery, for instance, a query to a central Directory Service to locate the information provider. The information seldom changes, and the required data can be directly acquired through the central repository. This category consists of information, such as system configurations and descriptions.
- Dynamic monitoring occurs when the cost of information gathering is typically greater, and, more often than not, entails time series, for example, when there is provision for a continuous data flow, or there is a need for large amount of data. This category's classical examples are network and system performance monitoring.
- Workflow monitoring occurs when uneven amounts of data are produced as the processing of a job/task is being performed, and either all or part of it can be of some interest to the consumer. This category's examples include error reporting, job/task processing status information and job/task tracing.

The Globus Toolkit provides Grid monitoring through the Monitoring and Discovery Service.

### 4.3.3 Monitoring and Discovery Service (MDS4)

MDS4 was developed by the Global Alliance to provide information services for the Globus Toolkit 4.x (GT4) (GT4MDS, 2005). The aim of MDS4 information services is to offer scalable, uniform and efficient access to distributed information sources, so as to facilitate the discovery, selection and optimization of resources for users and applications within a Grid environment. The approach is aimed at masking the underlying resource heterogeneity, by standardizing reports on static and dynamic resource information.

MDS4 builds on and utilizes the standard interfaces defined by WS-Resource Framework and WS-Notification specifications to make available query and subscription interfaces to arbitrarily detailed resource data (modelled in XML) (Jennifer, 2005; Schopf *et al.*, 2006). MDS4 is a “protocol hourglass” that defines standard protocols for information access and delivery, along with standard schemas for information representation as illustrated in Figure 4.2. In order to make a resource selection, information can be obtained from a variety of sources, for instance: cluster monitoring systems, such as Ganglia, Clumon and Hawkeye; resource management and scheduling systems, including Condor, PBS, and LSF; and common services, such as GridFTP, GRAM, RLS, and RFT, which are all part of GT4 (Jennifer, 2005).

The Monitoring and Discovery System (MDS4) component of GT4 can be utilized to streamline the tasks of monitoring and discovering services and resources in a Grid environment. Monitoring refers to a process of observing resources or services (such as, computation and storage facilities), in order to track their usage or fix any problems encountered. For example, a user can utilize the monitoring system to identify resources that are running low on disk space, so as to take corrective action. Discovery, on the other hand, refers to the process of finding a suitable resource to perform a task: for instance, finding a compute host on which to run a job. Furthermore, this process can involve not only finding which resources are suitable (e.g., have the space available for storing aggregated HIV/AIDS data) but also choosing a suitable member from that set (e.g., the one with the shortest submission queue).

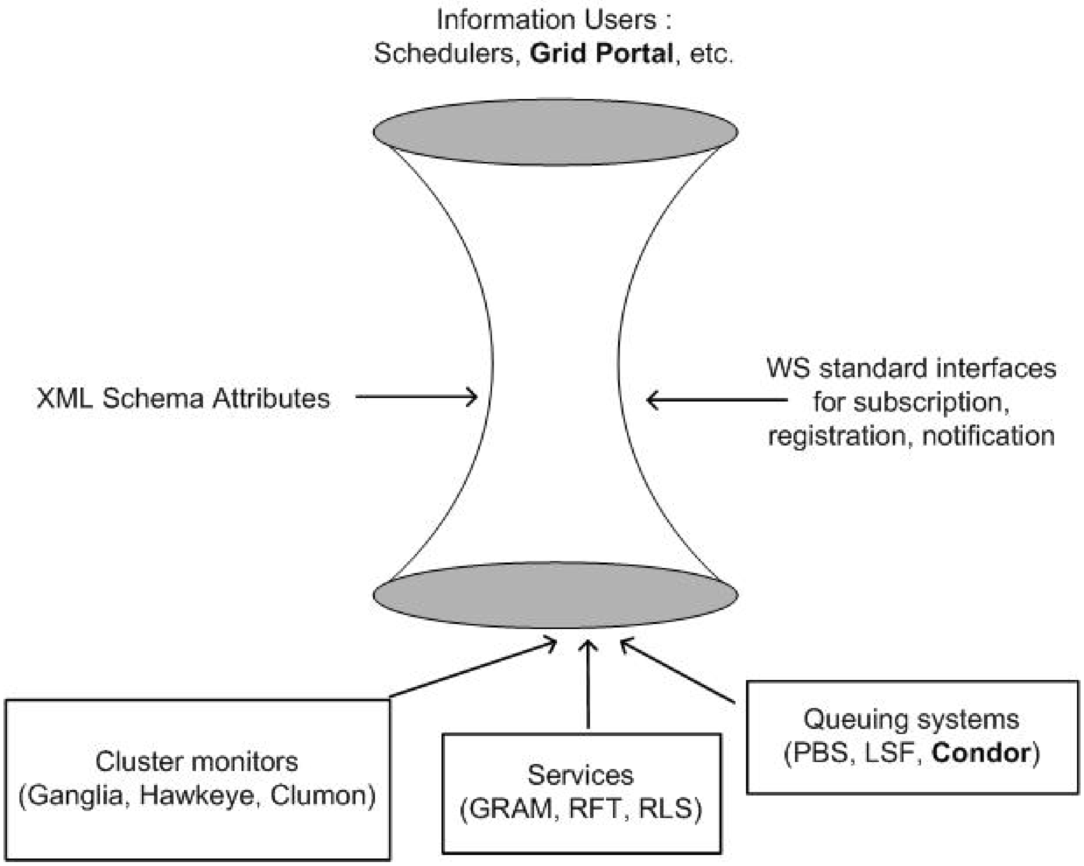
Information Users :
Schedulers, Grid Portal, etc.
XML Schema Attributes
WS standard interfaces for subscription, registration, notification
Cluster monitors (Ganglia, Hawkeye, Clumon)
Services (GRAM, RFT, RLS)
Queuing systems (PBS, LSF, Condor)

**Figure 4.2: The MDS4 Protocol Hourglass (Based on Jennifer, 2005)**

Both monitoring and discovery applications need the capability of collect information from multiple, probably distributed, information sources (Schopf *et al*., 2006). To meet this requirement, MDS4 provides so-called aggregator services that collect recent state information from registered information sources, and browser-based interfaces, command line tools, and Web service interfaces that enable users to query and access the collected information (GT4MDS, 2005).

There are three different aggregator services with different interfaces and behaviours in MDS4 (all are built on a common framework, though) (GT4MDS, 2005; Foster, 2005):

- The MDS-Index service presents data collected from information sources as XML documents. In particular, the data is maintained as WSRF resource properties. MDS-Index supports Xpath queries on the latest values obtained from the information sources.
- The MDS-Trigger service defines a Web service interface that allows a client to register an Xpath query and a program to be executed (for example, send email or create a log-file entry) whenever a new value matches the user determined criteria.
- The MDS-Archive service stores all values received from information sources in persistent storage. Clients can then submit queries and specify a time range for which data values are required.

In order to simplify the tasks of registering information sources and locating and accessing information of interest, MDS4 heavily utilizes XML and Web service interfaces (Jennifer, 2005). Specifically, all information collected by aggregator services is maintained as XML, and may be queried through Xpath queries (in addition to other Web services mechanisms) (Schopf *et al.*, 2006).

#### *4.3.3.1 Aggregators and Information Sources*

In order to understand MDS4, one has to acquaint themselves with the aggregator-information source framework. The basic ideas are as follows (Foster, 2005):

- Information sources, for which discovery or access is required, are explicitly registered with an aggregator service.
- Registrations have a lifetime; they have to be renewed periodically or else they will expire. (Thus, an aggregator will carry out self-cleaning, where outdated entries will disappear automatically if their registrations were not renewed.)

- The aggregator periodically collects up-to-date state or status information from all registered information sources, by invoking an information-source-specific access mechanism.
- The aggregator then presents all information obtained from registered information sources through an aggregator-specific Web services interface.

MDS4 aggregators are differentiated from a traditional static registry, for example, UDDI, by their soft state registration of information sources, and also the periodic renewal of the information source values that they store (Jennifer, 2005). This dynamic behaviour facilitates for scalable discovery, by enabling users to access "recent" information without direct access to the information sources.

#### *4.3.3.2 Information Sources and Registration*

An information source refers to essentially any entity from which an aggregator service can obtain information such as a file, a program, a Web service, or another network-enabled service (Schopf *et al*., 2006).

As mentioned previously, information sources must be registered periodically with any aggregator service in order to provide access to its data values. Registration is performed using a Web service (WSServiceGroup) Add operation. Two modes of registration are supported and each mode specifies the mechanism to be utilized to access the associated information source (Foster, 2005).

One mode supports a more general registration which enables information to be obtained from an arbitrary source. With this mode, an information source is registered by providing a user-supplied program that is run periodically to obtain up-to-date data (Jennifer, 2005). This user-supplied program can either generate the information locally or alternatively utilize a source-specific protocol in order to remotely access the information. The program must convert non-XML data into an appropriate XML representation (Schopf *et al*., 2006).

The other mode supports a more streamlined form of registration for WSRF-compliant Web services. These services basically need to make status and state information available as WSRF resource properties (Jennifer, 2005). At registration time, the user can specify whether the aggregator must pull the resource properties, via the WSRF “get resource property” interface, or if it should subscribe to resource property changes in order for values to be pushed via WS-Notification subscription methods (Foster, 2005).

#### *4.3.3.3 Built-In Information Sources and MDS-Index Services*

Every GT4 Web services container comprises a default MDS-Index service with which any GT4 services running in that container (such as GRAM, RFT) is automatically registered (Jennifer, 2005; GT4MDS, 2005; Foster, 2005). Therefore, each installation on a platform has an index that enables one to discover what services are available.

Furthermore, there is often a need to keep track of all available WS-Resources in a Grid environment. In order to accomplish this, GT4 presents a simple method for specifying one or more default indexes to be a Grid-wide MDS-Index, in order for each WS-Resource registered to a default MDS-Index to also be registered in the Grid MDS-Index (Schopf *et al*., 2006).

Every GT4 Web service supports a minimal set of resource properties (an informal service name and a service start-up time) and thus can be registered easily into one or more aggregators for monitoring and discovery. Further, two GT4 Web services, GRAM and RFT, also publish a larger set of service-specific information. Finally, the GT4 distribution also includes information source executables to enable registering GridFTP and RLS into aggregators (Foster, 2005).

## 4.4 Grid Data Management

### 4.4.1 Grid File Transfer Protocol (GridFTP)

The GridFTP protocol is responsible for the fast, robust, efficient, and secure transfer of (mainly bulk) data (GridFTP, 2005). The protocol builds on RFC 959 - File Transfer Protocol (FTP) (Postel & Reynolds, 1985), RFC 2228 - FTP Security Extensions (Horowitz & Lunt, 1997) , RFC 2389 - Feature negotiation mechanism for the File Transfer Protocol (Hethmon & Elz, 1998), the IETF  draft draft-ietf-ftpext-mlst-16-FTP Extensions (Elz & Hethmon, 2003), which are incorporated by reference. The most commonly used implementation of the GridFTP is on the Globus Toolkit (Allcock, 2003). GridFTP provides: a server implementation; a scriptable command line client; and a set of development libraries for custom clients. The interfaces for monitoring and controlling third-party data transfers are supplied by the Reliable File Transfer Service.

### 4.4.2 Reliable File Transfer (RFT) Service

The Reliable File Transfer (RFT) Service is a Web Services Resource Framework (WSRF) compliant Web service that is utilized to persist a data transfer state in a reliable storage (DMGT4, 2005). Users can employ service methods to query the status of the transfer, or alternatively, they can subscribe for notifications of state change events through standard WSRF tools (provided in the Globus Toolkit). The RFT Service implementation in GT4 utilizes standard SOAP messages over HTTP for submission and management of third-party GridFTP transfers, in addition to the deletion of files and directories using GridFTP (Li & Baker, 2005). Moreover, the service provides an interface that controls various transfer parameters of the GridFTP control channel, such as TCP buffer size, parallel streams, etc.

### 4.4.3 Replica Location Service (RLS)

The Replica Location Service (RLS) is a simple registry that keeps track of one or more copies, or replicas, of files in a Grid environment (DMGT4, 2005). This tool, which is part of the Globus Toolkit, is particularly helpful for users or applications that need to find where existing files are located in the Grid. Users or services register files in RLS when the files are created. Later, users can then query RLS servers to find these replicas (Schopf *et al*., 2006).

RLS is a distributed registry, thus it can consist of multiple servers at different sites. The distribution of the RLS registry allows for the increase in the overall scale of the system and the possibility to store more mappings than in a single, centralized catalog. Moreover, this avoids creation of a single point of failure in the Grid data management system. However, RLS can also be deployed as a single, centralized server if desired (Jennifer, 2005).

An RLS specifies two names (DMGT4, 2005)

- A *logical file name* is a unique identifier for the contents of a file.
- A *physical file name* is the location of a copy of the file on a storage system.

The RLS is responsible for maintaining associations, or mappings, between logical file names and one or more physical file names of replicas. A user can provide a logical file name to an RLS server and ask for all the registered physical file names of replicas. Furthermore, the user can query an RLS server to find the logical file name associated with a particular physical file location. Moreover, RLS allows users to associate attributes or descriptive information (such as size or checksum) with logical or physical file names that are registered in the catalogue. Users can also query RLS based on these attributes.

## 4.5 Grid Security

### 4.5.1 Digital Signatures

Digital signatures are utilized to ensure integrity in public-key systems through authenticating digital information, just as someone can authenticate a paper document by signing it (Ciampa, 2005; Simpson, 2005). In fact, a digital signature is merely a sequence of bits that conform to one of a number of standards.

The majority of digital signatures are based on public-key cryptography (Weise, 2001; Simpson, 2005). For instance, there can be a situation where an individual wants to send a message to another without caring about who might read it, but rather needing to prove he/she is the originator. In this scenario, they can send an encrypted copy of message, together with a copy of the message encrypted with their private key (NB: not public key). In order to determine if the message is authentically from the originator, they will unscramble (decrypt) the scrambled message utilizing the sender’s public key and then compare it with the unscrambled version. If the messages match, it implies that the message is genuinely from the originator, as the copy of message was encrypted using the private key, which is only possessed by the originator. A cryptographically strong hash function (Black, 2006) is often applied to the message, and only the resulting message digest, instead of the whole message is encrypted, which makes the signature a lot shorter than the message, while saving substantial time, because hashing is usually much faster than public-key encryption, byte for byte.

### 4.5.2 Public-Key Infrastructure

Public-Key Infrastructure (PKI) is software that provides certificate management in a large-scale and distributed setting. A PKI acts a certificate network that is utilized to locate public keys of users (Levi, Caglayan & Koc, 2004). A public-key certificate is a digitally signed binding that contains a public key, along with one or more attributes that identify the owner (Chokhani *et al*., 2003). These attributes include the owner’s name, Uniform Resource Locator (URL), email address and authorizations, which can be utilized to grant permissions and capabilities. Certificates are issued and signed by a Certificate Authority (CA) (Levi *et al*., 2004). The CA acts as a guarantor that verifies that the named entity is the rightful owner of the public key. The hierarchy of certificates in X.509 PKI systems is always a top-down tree, with the CA as the root, which does not need to be authenticated because it is well known. If a private key has been exposed, the related certificate will be revoked. The Certificate Revocation List is utilized to store the list of revoked certificates, and it is updated frequently.

Typically, a certificate contains:

- The public key being signed,
- The owner's name, for instance, an organisation, person or a computer,
- The period it is valid, and
- The URL (location) of the revocation list.

There have been numerous PKIs proposed, such as Privacy Enhanced Mail (PEM) (Kent, 1993), USPS Information-Based Indicia Program (IBIP) (USPS, 1998), Secure/Multipurpose Internet Mail Extensions (S/MIME) (Ramsdell, 1999), and PKI for X.509 certificates (PKIX) (Adams & Farrell, 1999; Housley *et al*., 2002). Most of the proposed PKIs are based on the ITU-T X.509 certificate standard. An X.509 certificate is usually a plaintext file that consists of information in a specific syntax (ITU-T, 2005):

- Subject: Represents the user's name.
- Subject's public key: This consists of the key itself along with other information, for instance, the algorithm utilized to generate the public key.
- Issuer's subject: CA's distinguished names, which can be represented through different attributes, the most common being Organisational Unit (e.g., OU = NMMU), Location (e.g., L =PE), and Common Name (typically the user's name, e.g., Gaolathe Seelo).
- Digital Signature: The digital signature to all the information in the certificate. It is generated utilizing the CA's private key. The digital signature is verified through the CA's public key, which is contained in the CA's certificate.

### 4.5.3 Certificate Authority (CA)

The CA's role is to offer entities a trustable digital identity that can be utilized to securely access resources (Ciampa, 2005). Thus, it must issue (create and sign) certificates, publish valid certificates publicly, accordingly revoke certificates and issue revocation lists regularly. Moreover, the CA must keep records of all its transactions.

A CA can issue users a personal certificate that enables users identify themselves to remote entities. A personal certificate can be utilized for digital signatures as well. A CA can furthermore issue host (server) and service certificates, which hosts and services utilize to identify themselves to the network.

A number of CAs can issue validation certificates that corroborate the identity of subordinate CAs. Every CA publishes a document, known as the Certificate Policy Statement (CPS), whether it is a subordinate CA or the one validating subordinate CAs (Li & Baker, 2005). A CPS specifies the conditions that it follows in order to issue a certificate, and the level of assurance on the certificates issued by the CA.

A CA issues public-key certificates, which state that the CA trusts the certificate owner and that they are who they say they are. It is imperative for the CA to ensure that the applicant’s identity matches certificate credentials, because a relying party trusts the CA to verify the identity in order for it to trust the user certificate in turn.

### 4.5.4 The Grid Security Infrastructure (GSI)

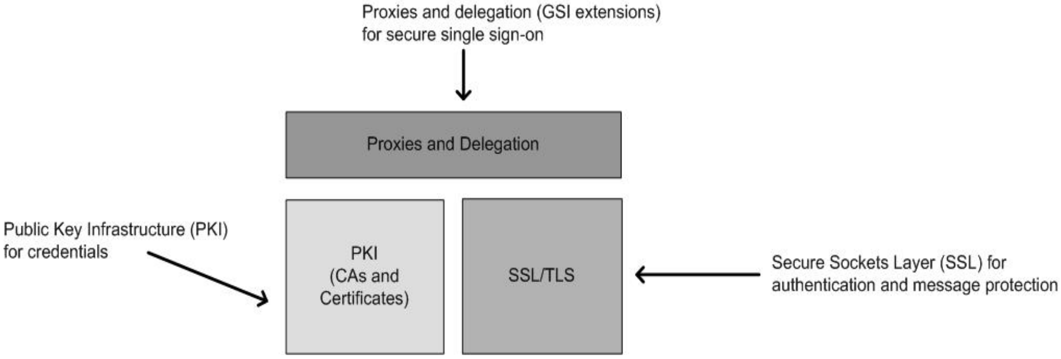
Proxies and delegation (GSI extensions)
for secure single sign-on
Proxies and Delegation
Public Key Infrastructure (PKI)
for credentials
PKI
(CAs and
Certificates)
SSL/TLS
Secure Sockets Layer (SSL) for
authentication and message protection

**Figure 4.1: The Grid Security Infrastructure**

The Global Grid Forum (GGF) specifies the Grid Security Infrastructure (GSI) as the basis for Grid security (Foster, Kesselman, Tsudik & Tuecke, 1998). GSI is made up of a set of protocols, tools and libraries which are utilized in the Globus Toolkit and other Grid middleware, to enable secure access to resources by users and applications. The GSI implementation is based on a Public Key Infrastructure that includes certificate authorities and X.509 certificates. According to Foster & Kesselman (2004) GSI offers:

- A public-key system.
- Mutual authentication through digital certificates.
- Credential delegation and single sign-on.

GSI is made up of well-known and trusted technologies. It can be configured to provide authentication, privacy and integrity; moreover, certificates can be utilized to provide strong authentication (Foster *et al.*, 1998). However, during a communication not all the features are needed. The minimum requirement for a GSI-based secure conversation is authentication. Although integrity is enviable, it can be disabled, and privacy can also be activated or disabled through encryption.

To ensure strong authentication, GSI utilizes X.509 certificates (ITU-T, 2005). When both parties involved in a secure conversation authenticate each other, it is referred to as mutual authentication. Thus, if an originator desires to communicate with a remote party, it must first establish trust with the party and vice versa. Trust, in this case, means that both parties must trust the certificate of the CA that signed each other's certificate, or else trust will not prevail (Li & Baker, 2005, p. 135).

GSI affords a user to generate and delegate proxy credentials to processes that are running on remote processes; thus, it enables remote resources and processes to act on a user's behalf (Foster & Kesselman, 2004). This feature is essential in complex applications, where there might be a need to utilize numerous resources concurrently. The proxy credentials are created by a user and are short-lived; thus, they act as a short-term binding of the user's identity to an alternate private key.

### 4.5.5 Authorization modes in GSI

There are three authorization modes supported by GSI on both the client and the server (Li & Baker, 2005, p. 136).

*Server-side authorization*

The decision to accept or decline an incoming security request to the server depends on the authorization mode chosen.

1. None: No authorization will be carried out. This is the simplest form of authorization.
2. Self: A client will only be permitted to utilize a Grid service if the identity of the client is the same as the identity of the service.
3. Gridmap: Analogous to an Access Control List (ACL), the Gridmap file is made up of a list of authorised users. Thus, it contains a list of users that are allowed to invoke the service.

*Client-side authorization*

The decision for a client to invoke a remote service or not depends the security credentials chosen.

1. None: No authorization will be carried out.
2. Self: The client will only invoke a service whose identity is similar to that of the client. If both the client- and server-side utilize self-authorisation, the service invocation will be authorized if identity of the service matches that of the client.
3. Host: A security request will only be authorized by the client if identity returned by the host contains the hostname. This is accomplished through host certificates.

### 4.5.6 Making a certificate request

A user requesting a certificate begins by generating a pair of keys (Foster & Kesselman, 2004). The private key is encrypted with a pass phrase and stored, while the public key is placed in the certificate request that is forwarded to the CA. The CA typically consists of Registration Authorities (RAs) that verify the requests by making sure that facts, such as a user's name, are unique as pertaining to the CA, and that they are real names.

### 4.5.7 Mutual Authentication

Mutual Authentication is implemented through the Secure Sockets Layer (SSL) (Netscape, 1998) protocol in GSI. However, before entities can mutually authenticate each other, they initially have to trust the CAs that signed each other's certificates. Both entities will possess a copy of the each other's CA certificate, which includes its public keys.

### 4.5.8 Confidential Communication

By default, GSI does provide encrypted communication between entities. After mutual authentication has occurred between entities, communications can ensue without the overhead of encryption. Only when there is a need can confidential communication be established. The default behaviour of GSI provides communication integrity (Li & Baker, 2005, p. 138).

### 4.5.9 Securing Private Keys

In GSI, the user's private key is expected to be stored in a file that is encrypted with a password (also referred to as a pass phrase), in a safe location on a computer's file system (Foster *et al*., 1998). If a user wants to decrypt the file containing the private key they have to enter the required pass phrase.

### 4.5.10 Delegation and Single Sign-On

GSI delegation enhances security by reducing the number of times a user must enter a pass phrase. If an activity involves several resources, or an agent or a broker is acting on behalf of the user (each requiring mutual authentication), a proxy can be generated to avoid the need to repeatedly enter the pass phrase (Li & Baker, 2005, p. 139).

A proxy is made up of a new certificate (containing a new public key), and a new private key. The new certificate has the owner's identity, slightly modified to indicate that it is a proxy. The owner, as opposed to the CA, signs the new certificate. Furthermore, the certificate indicates a time notation, after which the proxy should no longer be accepted. The lifetime of proxy is limited and cannot continue past the expiry of the original certificate (Foster *et al*., 1998).

A proxy's private key must be stored securely. Typically, the proxy's key is stored on the local storage system without encryption because it is not valid for very long, however, it will have file permissions that prevent it from being easily examined. After a proxy is generated and stored, the user can utilize the proxy certificate and private key for mutual authentication without entering a password (Foster & Kesselman, 2004).

Mutual authentication is performed differently when proxies are employed. The remote party receives the proxy certificate (signed by the owner), along with the owner's certificate. During mutual authentication, the owner's public key (acquired from their certificate) is utilized to validate the signature on the proxy certificate. The CA's public key is in turn employed to validate the signature on the owner's certificate. This sets up a chain of trust from the CA to the proxy via the owner, where the proxy process impersonates the owner for the lifetime of the proxy certificate (Li & Baker, 2005, p. 140).

## 4.6 Conclusion

This chapter discussed the Globus Toolkit Version 4 (GT4), which is currently the *de facto* standard for Grid service implementations. A Grid service is essentially made up of at least one Web service that provides a set of well-defined open standards-based interfaces that follow specific conventions. GT4 offers execution management, information services, data management and security services in a collaborative HIV/AIDS research Grid environment.

Once the data reaches service-providers in (HIV/AIDS) VIOISs it is stored in data warehouses. The following chapter discusses data warehousing.

# Chapter 5
# Data Warehousing

## INTRODUCTION

In medical research and diagnosis, large collections of data play a vital role in aiding decision-making processes. The data collection's size is expected to grow from gigabyte size to multiple terabyte-size, and even petabyte-size, as more and more data-providers (Data Participants) join or contribute data for research purposes. The sheer size of these data collections renders the analysis and interpretation of the data exigent. Also adding to the problem, are the geographically distributed researchers needing access, the geographically distributed computation and storage resources (Service Participants), and furthermore, the computationally intensive and complex analysis processes. Thus, there is a need to discover, access, analyze, share, interpret and integrate these distributed data collections. The Open Grid Services Architecture (OGSA) addresses the need for discovering, accessing, sharing, analysis and interpretation of data collections by specifying basic architectural structure and mechanisms for building a service-oriented infrastructure (Atkinson *et al*., 2004). Data warehouses and data marts address the need for integration and storage of these distributed data collections. This data is integrated and stored as structured collections (XML documents) in data warehouses and data marts.

This chapter discusses data warehousing and how it will be adopted into a Grid computing environment.

## 5.1 Databases

The concepts of databases and their implementation have been in existence for many years (Singh, 1998). The early implementations of databases involved a single database serving every purpose known to the information processing community – from handling transactions to batch processing to even analytical processing. The main focus of the early database systems was generally operational –typically transactional– processing.

However, there has been an emergence of more sophisticated concepts of databases in recent years: one that provide for operational needs and another that serves informational or analytical needs. This split of operational and informational databases transpired because of a number of reasons including (McDonald, Wilmsmeier, Dixon & Inmon, 2002):

- that there is a physical difference between data that serves operational needs and data that serves informational or analytic needs;
- the technology utilized to support operational processing is largely different from the technology utilized in supporting informational or analytical needs;
- the user community that is served by operational data is quite different from the one served by informational or analytical data;
- the processing characteristics for the operational environment are fundamentally different from those of the informational environment.

For these reasons (and many more), there is a need to build systems in a modern way, separating the operational from the informational or analytical processing and data.

### 5.2 Analytical/Informational processing

Informational or analytical processing is utilized in serving the needs of management during decision-making processes. Also referred to as Decision Support System (DSS) processing, analytical processing is employed to detect trends over broad vistas of data. However, instead of inspecting one or two records of data (as is the case in operational processing), in DSS many records of data are looked at.

Although data in operational systems is continually being updated at the individual record level, DSS data is rarely updated. Thus, even though records are constantly being accessed for analytical processing, and their contents are congregated for analysis, there is little or no alteration to the individual records. The response time requirements in analytical processing are greatly relaxed compared to those of traditional operational processing (Inmon, 2002). Analytical processing response time can be measured from 30 minutes to 24 hours. If operational processing response times were measured in this range, this would result in an absolute disaster. Table 5.1 below outlines the essential differences between operational data and informational or analytical data.

**Table 5.1 Characteristics of Operational Versus Informational Systems**

| OPERATIONAL | INFORMATIONAL/DSS |
|---|---|
| | |
| Detailed | Summarized |
| Can be updated | Snapshot records; no updates allowed |
| Accurate up to the second | Timestamp on each record |
| Used for clerical purposes | Used by management |
| Built based on requirements | Built without knowing requirements |
| Supports small uniform transactions | Supports mixed workloads |
| Yields 2- to 3-second response time | Yields 30- to 24-hour response time |
| Data designed for optimal storage | Data designed for optimal access |
| Very current data | Mainly historical data |
| Data is application oriented | Data is integrated |
| Data designed around functional usage | Data designed around subject areas |
| Referential integrity is useful | Referential integrity is not useful |
| High availability is normal | High availability is nice to have |

As depicted in Table 5.1, there is a fundamental split that exists between operational data and informational (DSS) or analytical data. The basis for DSS processing has become an architectural structure called the data warehouse. A data warehouse is physically separate from the online operational application. The data warehouse will be described in the following sections, and how it enables informational or analytical processing.

### 5.3 Data Warehouse

A data warehouse refers to a subject-oriented, integrated, non-volatile, and time-variant collection of data in support of management's decisions (Inmon, 2002, p. 31). These characteristics are further explored below:

*Subject orientation*: Data in a data warehouse is structured around a major object or process of an organization. In the envisaged collaborative HIV/AIDS research domain, the major subject areas include patient, health provider, treatment, etc.

*Integration*: Of all the characteristics of a data warehouse, integration is certainly the most important (Gorla, 2003; Inmon, 2002). Data is fed into the data warehouse from multiple disparate sources. However, before the data can be fed into the data warehouse, it must be converted, reformatted, resequenced, summarized, etc., because there is no application consistency in physical attributes, naming conventions, measurement of attributes, encoding, and so forth. Thus, integration is responsible for this restructuring and reformatting of the data.

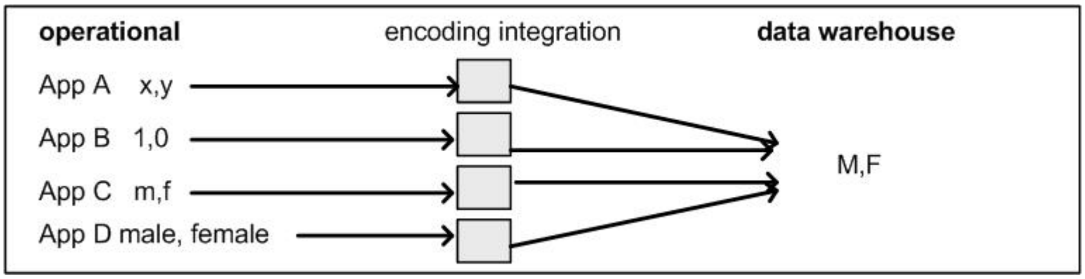

**Figure 5.1: The Issue of Integration**

When data is loaded into the data warehouse, the many inconsistencies that exist at the application level should be fixed. As depicted in the above example (Figure 5.1), when encoding of gender, it does not matter whether data in the warehouse is encoded as m/f or male/female. However, regardless of the method or source application, it only matters that the data warehouse encoding is done consistently. If application data has been encoded as X/Y or 1/0, it must be converted as it is loaded into the data warehouse (to M,F in the above example). The same consideration of consistency should be extended to all application design issues, for instance, key structure, naming conventions, measurement of attributes, and physical characteristics of data.

*Non-volatile*: Although data in the operational environment is regularly updated, data in the data warehouse exhibits a very different set of characteristics. Data is loaded (usually en masse) into the data warehouse and accessed, but it is not updated (Inmon, 2002, p. 33). In fact, data is loaded into the data warehouse in a snapshot, static format. If there are subsequent changes to the data, a new snapshot record is written. In doing so, a history of data is kept in the data warehouse.

*Time-variant*: Each unit of data in the data warehouse is accurate only as of some one moment in time. A record can be timestamped, have a date of transaction or have some form of time marking to show the moment in time during which the record is accurate. Different environments have different time horizons. A time horizon refers to the parameters of time represented in an environment. The collective time horizon for the data found inside a data warehouse is considerably longer than that of operational systems. Usually, a 60-to-90-day time horizon is normal for operational systems, while a 5-to-10-year time horizon is normal for the data warehouse. The data warehouse contains much more history than any other environment because of this difference in time horizons (Ibid: p. 32).

A data warehouse signifies the single version of truth about the organization and holds data at a granular level. Furthermore, a data warehouse contains robust amounts of historical data. A data warehouse is at the centre of the decision support system processes. The different levels of detail in a data warehouse are illustrated in Figure 5.2 below.

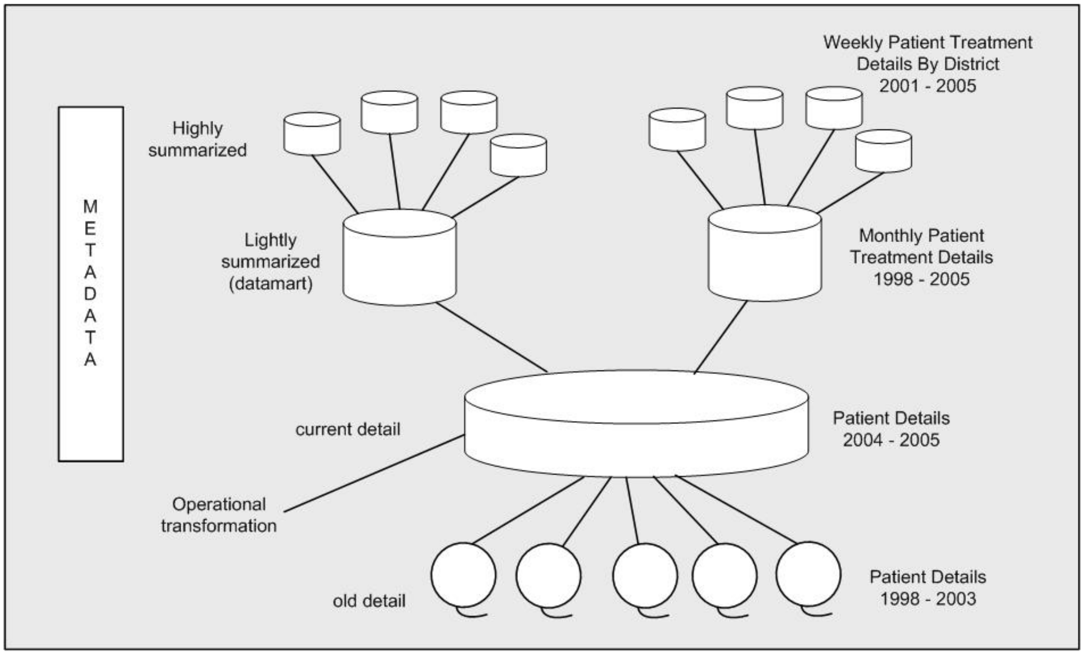
Weekly Patient Treatment Details By District 2001 - 2005
M E T A D A T A
Highly summarized
Lightly summarized (datamart)
Monthly Patient Treatment Details 1998 - 2005
current detail
Patient Details 2004 - 2005
Operational transformation
old detail
Patient Details 1998 - 2003

**Figure 5.2: The Structure of a Data Warehouse (Inmon, 2002)**

As depicted on Figure 5.2, there is an older level of detail (usually on alternate, bulk storage), a current level of detail, a level of lightly summarized data (the data mart level), and a level of highly summarized data. Data is loaded from the operational environment into the data warehouse. Typically, there is significant transformation of data as it flows from the operational level to the data warehouse level. Once the data ages, it passes to older detail from current detail. As the data is summarized, it passes from current detail to lightly summarized data, then from lightly summarized data to highly summarized data.

As stated by Gardner (1998), a data warehouse should be flexible and scalable in line with the changes of business supported. This helps decision makers to find out not only what is happening but also why certain events have taken place (Gardner, 1998).

Data warehouses are not built all at once (Inmon, 2002, p. 41). In fact, they are designed and populated one-step-at-a-time, and are thus evolutionary, as opposed to revolutionary. Therefore, the data warehouse evolves and supports the process of moving data from operational systems, transforming, and cleansing the data in order for it to be stored in an integrated data model at an atomic level of granularity. The design of a data warehouse and the structure of its data records are influenced by various factors. Some of these factors are discussed in the next sections.

### 5.3.1 Data Models

The design of the data warehouse usually begins with a data model (McDonald, Wilmsmeier, Dixon & Inmon, 2002). The highest level of definition for a data model is referred to as an entity relationship diagram (ERD). An ERD represents the abstraction of the granular data found in the data warehouse. It must be noted that in data warehouse designs, the ERD represents only granular data, and not derived data. It is imperative to delineate this distinction as it greatly limits the size and complexity of the data model (Inmon, 2002, pp. 89-94). However, there are obviously, other data models that do attempt to take into account derived data and atomic data outside of the data warehouse environment.

An ERD is made up of entities that have relationships. Each entity represents a major subject area of the organization. Typical subject areas in the envisaged collaborative HIV/AIDS research domain include patient, health provider, treatment, etc. Entities are further defined at a lower level of data modelling, known as the data item set (DIS). The DIS specifies a lower level of detail than that specified by the entity, including elements, such as keys and attributes, as well as the structure of those elements. The DIS is furthermore broken down into a low level of design, referred to as the physical design (McDonald *et al*., 2002). At the physical level of design, the physical characteristics of the data are created. Figure 5.3 depicts an example of part of a physical level of design for the collaborative HIV/AIDS domain.

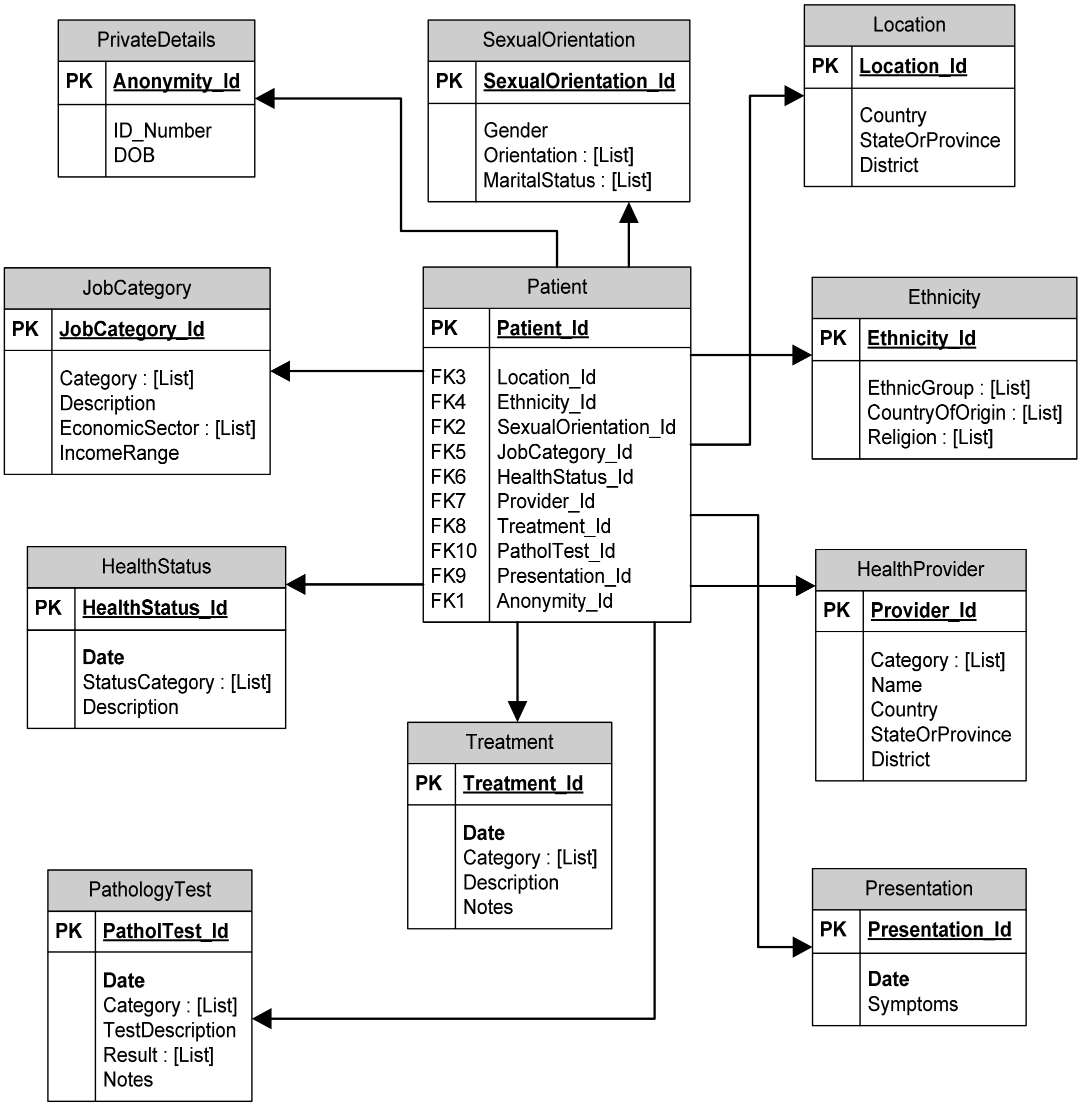

**Figure 5.3: Example Physical Design for the Envisaged Collaborative HIV/AIDS Research Domain**

While it might be tempting to say that the physical tables depicted in Figure 5.3 are ready to be cast into the concrete physical database design, there is still one last design step that remains – which is factoring in the performance characteristics (Inmon, 2002, p. 99). In data warehouse design, the first step for doing this is deciding on the granularity and partitioning of the data (Gorla, 2003). This is critical for the success of the data warehouse. (Obviously, the key structure will also be altered, by adding the element of time, to which each unit of data is relevant.)

### 5.3.2 Granularity

Granularity is the first major design issue when designing a data warehouse. In fact, the issue of granularity permeates the entire architecture that surrounds the data warehouse environment. Granularity is the level of detail, or summarization of the units of data in the data warehouse (Inmon, 2002, p. 43). If there are more details, the level of granularity will be lower. On the other hand, the less detail there is, the higher the level of granularity. For instance, a low level of granularity can be represented by a simple transaction, while a summary of all transactions in three months would represent a high level of granularity.

Granularity is the most important aspect in a data warehouse environment because it very much affects the volume of data stored in a data warehouse and the type of queries that can be answered. The volume of data in a warehouse is traded off against the level of detail of a query. Granular data also referred to as atomic data of the organization and makes up the “single version of truth” that is at the basis of reconciliation for informational or analytical processing. Having the granular data at the core of the data warehouse presents many benefits.

#### *5.3.2.1 Advantages of Granularity*

The granular data found in the data warehouse is the key to reusability, because it can be used by many people in different ways (McDonald *et al*., 2002). For example, within our problem domain, the same data might be used to satisfy the needs of health professionals, HIV/AIDS researchers and academics. All the (three types of) end-users will look at the same basic data. Health professionals might want to see how patients react to a certain treatment, HIV/AIDS researchers may want to see how effective certain treatments are per region, while academics might want analyze if there is a correlation between a treatment and an opportunistic disease. Although all of these types of information might be closely related, they are slightly different from each other. With a data warehouse, the different end users in (HIV/AIDS) VIOISs are able to look at the data as they wish to see it.

Being able to look at the data in different ways presents only one advantage of having a solid foundation. A benefit related to this is the capability to reconcile data, if needed. If there is a need to explain a discrepancy in analyses between two or more research communities, reconciliation is relatively simple because there is a single foundation ("single version of truth") on which everyone relies. An additional related benefit is flexibility. Having a foundation in place allows a research community to be able to alter how it looks at data.

The leading benefit of a data warehouse foundation probably is that future unknown requirements can be accommodated (Inmon, 2002, p. 45). Presumably there will be a new requirement from a research community to look at data differently, or the state legislature passes a new law, or system administrators need to change access rights. As change is inevitable, there will be a constant stream of new requirements for information. The data warehouse can thus be easily utilized to respond to change.

Granular data stored in the data warehouse is utilized in more than only data marts. It can be utilized to support the processes of exploration and data mining (Inmon & Terdeman, 2000). Exploration and data mining acquire masses of detailed, historical data and examine it for previously unknown or undetected patterns. The data warehouse makes a very valuable source of data for the data miner and data explorer. The data found in the data warehouse is cleansed, integrated, structured and organized. Moreover, the data is historical. This presents the data miner and the explorer with the required foundation for engaging in the exploration and data mining activities. However, it must be noted that although the data warehouse provides an excellent source of data for the data miner and the data explorer, the data warehouse is often not the only source. Data warehouse data can be utilized alongside external data and other data when performing explorations and mining.

### 5.3.3 Partitioning

Partitioning is the second major design issue of data in the warehouse (after granularity). Partitioning of data means breaking up data into separate physical units, which can be managed independently (Inmon, 2002, p. 56). In a data warehouse, the issues surrounding partitioning do not focus on whether partitioning should be done, but rather, how it should be done.

If both partitioning and granularity are handled properly, then almost all other aspects of the data warehouse design and implementation will fall into place. However, if granularity is not done properly, and if partitioning is not designed and implemented carefully, then no other aspects of design really matter.

Partitioning the data warehouse properly can offer benefits in various ways (Singh, 1998):

- Loading data
- Accessing data
- Archiving data
- Deleting data
- Monitoring data
- Storing data.

If data is partitioned properly, this will enable the data to grow gracefully and to be managed. Conversely, if data is not properly partitioned, this will prevent the data from growing and being managed.

### 5.3.4 Timestamping

Each units of data stored in the data warehouse is timestamped in order for every unit of data in the data warehouse to have some element of time associated with the record (McDonald *et al*., 2002). Timestamping of units of data in a data warehouse indicates that the data record is accurate as at the timestamp (Inmon, 2002).

Usually, a record is stored in the data warehouse through two ways: discretely or continuously, as illustrated in Figure 5.4. In a discrete record, the accuracy of the record is represented at a specific instant of time. On the other hand, for a continuous record, there is a span of time for which the record is accurate. These records make up a larger definition of information over time.

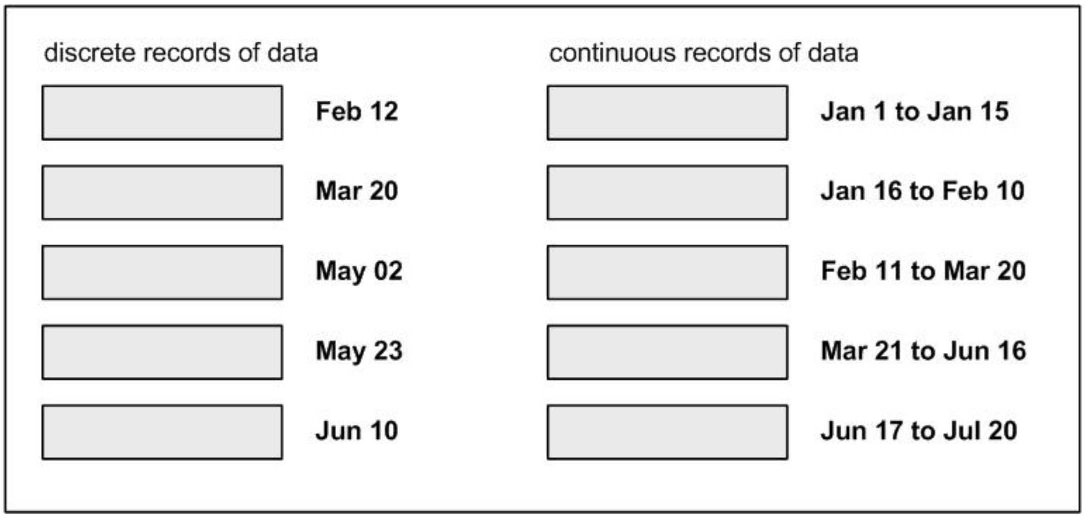

**Figure 5.4: Discrete and Continuous Timestamped Records (McDonald *et al*., 2002)**

Generally, discrete records are utilized for a large number of fast-changing variables. Continuous timestamps are utilized for a small number of variables that change gradually and for which there is value in knowing information over time.

The structure of records in a data warehouse is distinctive, including (McDonald *et al*., 2002):

- A timestamp
- A key
- Primary data
- Secondary data.

## 5.4 Scanning Operational Data

At the outset, operational transaction-oriented data is contained in existing legacy systems. One of the major hurdles is to efficiently access and scan this data from the existing systems. Because there are tons of operational data in existing systems, attempting to scan all of it every time a data warehouse is loaded, proves to be cumbersome and unrealistic. Moreover, it is difficult to delineate data that has been scanned and also data that should be scanned.

There are three types of loads that are made from the operational environment into the data warehouse (Inmon, 2002):

- Archival data
- Data that is currently in the operational environment
- Current changes to the data warehouse environment from the changes (updates) that have taken place in the operational environment from the time when the last load was performed.

In order to limit the amount of operational data scanned when loading the data warehouse, there are five common techniques which can be utilized (Inmon, 2002, pp. 85-86). The first technique that can be employed is to scan data that has been timestamped in the operational environment. If an application stamps the time of the last change or update on a record, the scanning can be efficiently run, as data with a date other than that applicable will not be touched. However, it more often than not only by happenstance that existing data in the operational environment has been timestamped.

The second technique that can be employed to manage the amount data to be scanned is to scan a “delta” file. A delta file records only the changes made in the operational environment as a result of the transactions that have been performed. The delta file ensures an efficient scanning process as data that is not a candidate for scanning is never touched. However, exactly like timestamping, not many operational applications build delta files.

The third technique that can be utilized consists of scanning a log file or an audit file generated as a by-product of transaction processing. Although a log file contains essentially the same data as a delta file, discussed previously, there are some major differences existing between the two. The log files are protected by computer operations, and they will not be utilized for anything other than their primary purpose of recovery process. Moreover, the internal format of a log tape is that it is built for systems' purposes, not applications' purposes. Therefore, an interface will be required turn to the contents of data on the log tape into an application-readable format. Furthermore, a log file typically consists of much more information than that desired by the data warehouse developer. Audit files possess many of the same limitation as log files.

The fourth technique that can be employed to limit the amount of operational data scanned is to modify application code. However, this is, in many cases, not a popular option, because a great deal of application code might be too old and fragile.

The fifth technique (the last resort, in most respects, a hideous one, mentioned mainly to convince people that there must be a better way) is rubbing a "before" and an "after" image of the operational file together. With this technique, a snapshot of a database is taken at the moment of extraction. When the next extraction is performed, another snapshot will be taken. The two snapshots are serially compared to each other in order to determine the activity that has transpired. This technique is unwieldy and complex, and it entails an inordinate amount of resources. Thus, it is purely a last resort to be done if everything else fails.

Although it is tempting to think that creating the data warehouse concerns only extracting operational data and loading it into the warehouse, nothing could be further from the truth. Simply pulling data out of the operational environment and placing it in the data warehouse achieves very little of the potential of data warehousing. Thus, data should be transformed before it passes into the data warehouse. Integration and transformation of the data presents another difficulty when loading data into the data warehouse from the operational environment.

### 5.5 Integration and Transformation Processing

Unless otherwise stated, the information in this section is sourced from Inmon (2002) and McDonald *et al*., (2002).

The movement and conversion of data from the operational/legacy source environment is but one of the essential and most difficult aspects of data warehouse development and population. During extraction, data is pulled from operational databases and transferred into the data warehouse. This data is extracted from various sources, such as appointment records, consultation and diagnosis records, treatment and therapy records, etc.

However, this data is not merely transferred from the operational databases into the data warehouse. Instead, the data has to be thoroughly transformed and integrated as it is moved. Some of the necessary functionality for integration and transformation includes:

*Reformatting data*: Non-key data should be reformatted as it passes from the operational environment into the data warehouse environment. As a basic example, input data about date is read as YYYY/MM/DD and is written to the output file as DD/MM/YYYY. (However, it must be noted that reformatting of operational data before it is ready to be moved into a data warehouse often becomes much more complex than this basic example.)

*Realigning encoded values*: Encoded data should be realigned so that it matches the format in the data warehouse. As a simple example, gender can be encoded as m/f in one application, while in another, it is encoded as 0/1, and in yet another it is encoded as x/y. The encoding should be realigned to be consistent with data warehouse, and it does not matter how gender is encoded (thus, one of the previously mentioned encoding example can be utilized) in the data warehouse as long as it is done consistently.

*Field transformation*: The same field can exists in different applications under various names. In order to transform operational data to the data warehouse properly, a mapping from the different source fields to the data warehouse fields must occur. For example, Date of Birth field can be represented as D.O.B, Patient D.O.B, Birthdate, etc.

*Converting data into a common format*: The input records that must be read can have exotic or nonstandard formats. There are a whole host of input types that must be read, and then converted on entry into the data warehouse:

- Fixed-length records
- Variable-length records
- Occurs depending on
- Occurs clause.

Conversion must be made. But the logic of conversion must be specified, and the mechanics of conversion (what the “before” and “after” look like) can be quite complex. In some instances, conversion logic becomes very twisted.

*Restructuring data*: Usually, operational input keys need to be restructured and converted before they are written out to the data warehouse. An input key rarely remains unaltered as it is extracted from the operational environment and loaded into the data warehouse environment. A simple case might consist of an addition of the time to the output key structure. On the other hand, complex cases can need the entire input key being rehashed or otherwise restructured.

*Assigning default values*: In some situations an output value in the data warehouse will have no source of data. For these cases, the default value that can be utilized must be specified.

*Summarization*: Under some circumstances, summarization of data will often be required. In this case, multiple operational input records will be combined into a single “profile” data warehouse record. In order to perform the summarization, the detailed input records to be summarized must be properly sequenced. In a situation where there will be different record types contributing to the single summarized data warehouse record, the arrival of the different input record types must be coordinated in such as way that a single record is produced.

*Converting from one database management system (DBMS) to another*: The extraction of data from the operational environment to the data warehouse environment usually involves a change in technology. Typically this can consist of reading the operational DBMS technology, for instance, IMS, and then writing the data out in newer, data warehouse DBMS technology, for example, Informix. Therefore, there is a need for a technology shift as the data is being moved. Moreover, the technology shift does not only concern changing the DBMS. It includes the hardware changes, operating system changes, and even the hardware-based structure of the data changes.

*Data cleansing*: As data passes from the operational environment to the data warehouse environment, it usually has to be cleansed. In a simple case, an algorithm can be applied to the input data so as to get it into an acceptable output form. However, complex cases might involve invocation of artificial intelligence subroutines on the input data in order to correct it. There are various forms of data cleansing, including cross-record verification, domain checking, and simple formatting verification.

*Merging different record types*: Multiple input sources of data exist and must be merged as they pass into the data warehouse. The source of a data element for the data warehouse can be from one file specified under one set of conditions, while the source of data for the data warehouse will be another file under another set of conditions. The right set of conditions must be set as to how the data must be merged before it is submitted to the data warehouse.

Moreover, when there are multiple input files, key resolution must be carried out before the files can be merged. Thus, if different key structures are utilized in the different input files, the merging program must have the logic embedded that enables resolution. Furthermore, with multiple input files, the sequence of the files may not be the same or even compatible. When this happens, the input files must be resequenced. However, this is not a problem unless there are many records that must be resequenced, but unfortunately that is almost always the case.

*Documentation of changes*: It is imperative to keep track of data elements when they are renamed as they are moved into the data warehouse from the operational environment. Names of data element usually change when they are being into the data warehouse environment. There should be documentation of these changes as they are made. Moreover, meta data which describes the activities of the conversion must also be stored.

*Adding a timestamp*: The data warehouse reflects the historical need for information, while the operational environment focuses on the immediate, current need for information. This means that an element of time may need to be added as the data moves from the operational environment to the data warehouse environment.

### 5.6 Meta Data

Meta data is a significant component of the data warehouse environment. Meta data refers to data about data, and has been a part of the information processing milieu for as long as there were data and programs. Meta data is structured information which distinguishes the data it describes through expressing the context and meaning (Foster *et al*., 2006). Thus meta data adds value to data.

Meta data assumes a new level of importance in the world of data warehouses, because it affords the most effective use of the data warehouse (Inmon, 2002, p. 113). Meta data enables the end user/DSS analyst to explore the possibilities. In other words, if a user navigates a data warehouse that does not have meta data, it is difficult for the user to know where to start the analysis. This means the user must then poke and probe the data warehouse in order to find out what data is available and what data is not there (McDonald *et al*., 2002). This will definitely waste a considerable amount of time and even after the user pokes around, there is no guarantee that he or she will find the necessary data or correctly interpret the data encountered.

However, if there is meta data, the end user can easily and quickly go to the right data or determine that it is not available. Thus, meta data acts like an index to the contents of the data warehouse. It sits above the warehouse in order to keep track of what is where in the warehouse. Usually, items the meta data store tracks include (Inmon, 2002, p. 113):

- Structure of data as known to the programmer
- Structure of data as known to the DSS analyst
- Source data feeding the data warehouse
- Transformation of data as it passes into the data warehouse
- Data model
- Relationship between the data model and the data warehouse
- History of extracts.

## 5.7 Data Warehousing and Grid Computing

Grid computing promises users the power to have access to a vast amount of heterogeneous, distributed resources (Foster *et al*., 2001; Foster *et al*., 2002). The envisaged goal is to enable users and applications to seamlessly access these resources to solve complex large-scale problems whether in science, engineering, or commerce (Foster *et al*., 2001; Foster *et al*., 2002; Czajkowski *et al*., 2001). To realize this goal, the numerous barriers that normally separate different computing systems within and across organizations must be addressed. This can be achieved through standardisation.

Standardisation plays a crucial role in the achievement of interoperability, portability and reusability of components and systems. It allows for computers, application services, data, and other resources to be discovered, accessed, allocated and accounted for as and when required, regardless of the location of the resource (Foster *et al*., 2004). In order to meet the standardisation needs in (HIV/AIDS) VIOISs, Open Grid Services Architecture (OGSA) defines a set of core capabilities and behaviours which will be utilized to address key concerns. One of the key concerns is how data in distributed data warehouses and data marts can be discovered, accessed, shared, analysed and interpreted. OGSA addresses the discovery, access, sharing needs by specifying services that deal with the management of, access to and update of data resources, in addition to data transfers and replication between resources. Collectively these services are known as data services.

Unless otherwise stated, the information in the following sub-sections (5.7.1 and 5.7.2) is sourced from Foster *et al*. (2006).

### 5.7.1 Data Services

Data services (data stores, data-access, data-processing resources, etc) provide the key capabilities that are specific to data access and data operations. They can be utilised for moving data as required; management of replicated copies; running queries and performing updates; and federation of data resources. They also provide the capabilities which are essential for managing the meta data that describes this data, specifically the provenance of the data itself.

(HIV/AIDS) VIOISs will comprise a variety of data resources. A data resource refers to any entity that can act as a source or sink of data, such as XML files, data warehouses, data marts, asynchronous query results, data services, etc. These data resources can utilise different physical media for storing data, different software systems for managing it, and different protocols and interfaces for accessing it. However, structure of the data and its schema description will be standardized. The data can be stored locally or remotely; may be unique or replicated; may be materialized or derived on demand. In order to hide these distinctions and enable the abstract viewing of data resources, virtualizations can be provided over the data resources. Data virtualizations allow for transparency of data resources and allow for ease of data access and processing.

The data virtualizations services support federated access to distributed data, dynamic discovery of data sources by content, dynamic migration of data for workload balancing, parallel data processing, and collaboration. These services virtualize various aspects of the grid, and make it appear as a single entity to the end-user applications. Data virtualisation services can include data discovery, collaboration, federation, workflow coordination and consistency management service. The services that are relevant to (HIV/AIDS) VIOISs are discussed below:

- In the envisaged collaborative HIV/AIDS research domain, just like all the OGSA Grids, resources are represented as services. The discovery service virtualizes the name and location of data on the grid, and forms the basic data virtualization service. Other virtualization services build on top of this name transparency to offer other kinds of transparencies.
- The data federation service analyzes each and every query that it receives in order to determine how to best possibly answer the query. This can include the creation of sub-queries against one or more of the data resources making up the federation, applying transformations to the results of those sub-queries, combining those query results in (arbitrarily) complex ways, and then transforming the result into the format requested by the client. Furthermore, it can determine where intermediate processing is done, so that network traffic can be minimized. It will allow data sources to be added and removed from the federation, provided they satisfy the service semantic restrictions. The application specifies its queries in terms of logical domains and predicates; the

discovery service maps these onto relevant sources. Thus, the combination of federation and discovery services provide applications with heterogeneity, distribution, and location transparency.

- Data is often replicated and distributed across multiple sites as will be the case in the envisaged collaborative HIV/AIDS research domain. Data replication will avoid a single point of failure if there is network or a server failure. A consistency management service will be utilized to ensure a consistency policy between the replicas.

### 5.7.2 Data Access, Transfer and Management

Data access is usually concerned with the retrieval, insertion or modification of data, which may be available from a variety of infrastructures and in a range of formats. However, in the envisaged collaborative HIV/AIDS domain, data access in the data warehouse and data marts will be required mainly (but not limited to) for applying queries against the data, in order to analyse data.

In simple cases a user can employ the data services to run an XML query over the data warehouse. However, complex cases can consist of distributed queries over federated data marts. Synchronous queries return the data in the response to a request, while asynchronous queries expose the derived data as new resources. Data services can also be data resources. Thus, they can also deliver the results of a query to a specified set of other services. Query services may optimize a query before sending it to the resource. The resources may further optimize the query and may also handle issues, such as concurrent access to the data.

Data services also support the transfer of data from one location to another. This can include creating a copy of the original data or completely migrating it. A client can specify the QoS, for example, reliable file transfer, the maximum bandwidth to use, the time when delivery is required, or delivery guarantees. The base functionality enables the transfers of bytes from one source to a single sink. In order to support the transfer to multiple sinks (i.e., during replication), thin layers above this are invoked, and they also enable the preservation of semantic information, for instance, file hierarchies, byte ordering or encodings.  The majority of services have a security policy decision point before transferring data. An access operation can then check whether the resulting transfer is allowed, depending on the contents of the data, and whether any restrictions apply to that transfer (for example, encryption).

Data services also provide the following data management capabilities:

- Storage management services control the provision of data warehouse and data marts storage to applications and other services. They are responsible for managing quotas, lifetime, and other properties, for example, encryption and persistency.
- Meta data catalogues refers to data services that will store descriptions of data held in data warehouses, data marts and other data services. This meta data will consist of information about the structure of the data, including references to the schemas that describe the data. However, this is not practical for some services, because data resources contain many schemas that are frequently modified, and in these cases, the services will provide the schema information themselves.
- Provenance refers to a special form of audit trail that traces each step in sourcing, moving, and processing of data. This may be at the level of the data warehouse or at the data mart level, sometimes to the level of query result. Thus, there is a need for the data services or other processes that generate the data to maintain the consistency of the provenance information. Complete provenance information can enable the data to be reconstructed by following the workflow that originally created it. Provenance information may be provided by the data service itself, or it may be maintained in a meta data catalogue or a logging service. Users of data services can then be able to query the information about the provenance and quality of the data provided by the services.

## 5.8 Conclusion

This chapter discussed data warehousing which provides data-storage for large collections of aggregated HIV/AIDS patient-data. The wide adoption of data warehouses can be attributed to their contribution to enhancing the efficiency and effectiveness of the decision process whether in science or business domains (Rizzi & Song, 2004). A data warehouse is used to integrate and store data extracted, converted and reformatted from multiple operational systems. The data warehouse subsequently feeds area-specific data marts.

Granularity is the most important aspect of a data warehouse because it very much affects the volume of data stored in a data warehouse and the type of queries that can be answered. Meta data affords users the ability to use the data warehouse effectively.

# Chapter 6

# A Proposed Framework for WGDW

## INTRODUCTION

The proposed framework for (HIV/AIDS) VIOISs is a virtual community that provides HIV/AIDS management information through a Wireless Grid-enabled Data Warehouse (WGDW). This community consists mainly of data-providers, service-providers and information-consumers.

This chapter explores how data is collected from the various data-providers and submitted to the service-providers. The framework proposes a wireless connectivity solution, and hence, a discussion of the implementation of wireless connections follows. Thereafter, the significance of the Grid on the implementation will be reviewed. Nonetheless, the chapter begins with an exploration of how XML, WSDL, SOAP and Web Services, are implicated, as they are open standards that support interoperable Grid systems. In order to provide security for the SOAP-messages, the SOAP-Header is enhanced with WS-* features such as XML Encryption, Digital Signatures and WS-Security.

### 6.1 Data Collection in the Proposed WGDW

The envisaged solution for the HIV/AIDS VIOIS is a virtual community that provides HIV/AIDS management information through a Wireless Grid-enabled Data Warehouse (WGDW). As mentioned, this community would consist mainly of Data Participants (data-providers), Service Participants (service-providers) and Subscribers (information consumers). Data-collection in the hypothetical system can be described as follows:

The Data Participants, which include hospitals (private and public), clinics, hospices, private medical practitioners, etc, will provide operational data (which will be transformed into aggregated data) on HIV-positive patients. The Service

Participants on the other hand, provide storage for this data and computation services to yield aggregated data. Research institutions (mainly universities and government research centres) will act as Service Participants. Subscribers, who will primarily be HIV/AIDS researchers, academics and medical practitioners, are the users of aggregated data. Subscribers could also be Service Participants (and vice versa).

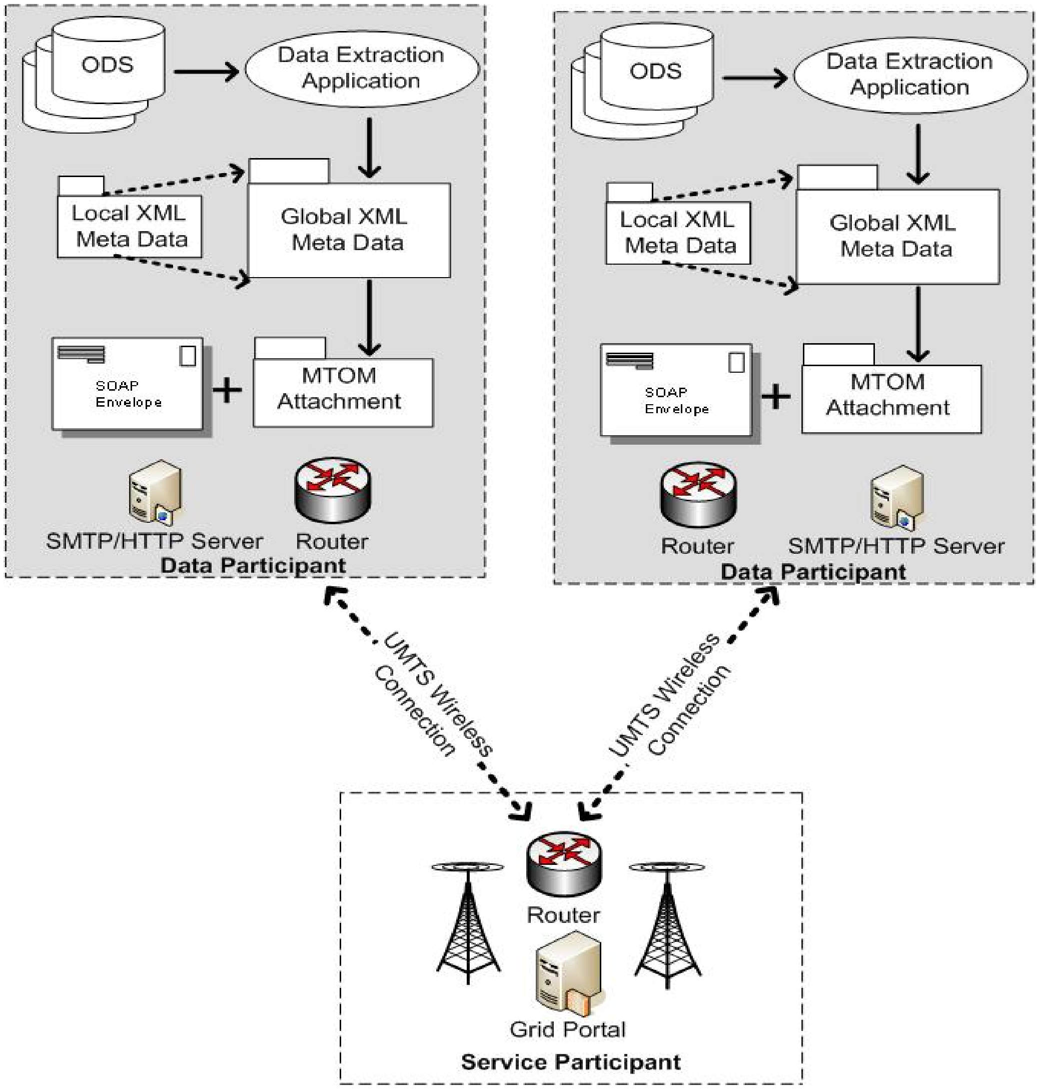

**Figure 6.1: Data Collection at the Data Participant**

Data Participants will periodically (periodicity will be agreed upon by Data Participants and Service Participants), extract/collect new or changed HIV+ patient-data that will be transferred to Service Participants, as depicted in Figure 6.1. This data will play a major role in building a data warehouse used to aggregate patients HIV/AIDS data. Local data extraction applications will be used to extract the data from the operational database systems. The data extraction application depends on the Service Participants' data requirements, in order to select data that will be extracted. The Service Participants will define global schema (Global XML Metadata) files that specify the data (such as the data participant's name, location, a patients' ID No, Date of birth, gender, sexual-orientation, ethnicity, health status, etc.) and the format of the data (for example, elements and attribute names, data types, structure, etc.). The data extraction application will first extract the required data (e.g., new patient-data, updated patient-data, etc.) with a query utilising the local schema file. The local schema (Local XML Metadata) will then be mapped into the global schema (Figure 6.1), so as to standardize the extracted XML files.

Once the data extraction/collection is complete, the extracted data will be serialised into an XOP package. The XOP package will be transmitted as a SOAP-attachment, utilising the Message Transmission Optimisation Mechanism (MTOM) standard. The next section discusses how the SOAP-message will be standardized and secured before it is transmitted over the wireless network.

## 6.2 Standardizing SOAP Messaging

Extracted files are transported as MTOM attachments, as illustrated in the next section.

### 6.2.1 Message Transmission Optimisation Mechanism (MTOM)

The MIME part of the message

```
MIME-Version: 1.0
1. Content-Type: Multipart/Related;boundary=MIME_boundary;
2.    type="application/xop+xml";
3.    start="< GPHDataCollection.xml@UbuntuResearch.org>";
4.    startinfo="application/soap+xml; action=\"ProcessData\""
5. Content-Description: A SOAP message with an XML Attachment

--MIME_boundary
6. Content-Type: application/xop+xml;
7.    charset=UTF-8;
8.    type="application/soap+xml; action=\"ProcessData\""
9. Content-Transfer-Encoding: 8bit
```

```
10. Content-ID: < GPHDataCollection.xml@UbuntuResearch.org >
11. <soap:Envelope
12.    xmlns:soap='http://www.w3.org/2003/05/soap-envelope'
13.    xmlns:xmlmime='http://www.w3.org/2004/11/xmlmime'>
14.  <soap:Body>
15.           <ur:data xmlns='http:// UbuntuResearch.org/fileTypes'
16.       <ur:file xmlmime:contentType='application/soap+xml'
17.       <xop:Include xmlns:xop='http://www.w3.org/2004/08/xop/include'
18.       href='cid:http://GPH.org/Extractions/GPH2006-06-30.xml'/></ur:file>
19.       </ur:data>
20.  </soap:Body>
21. </soap:Envelope>

--MIME_boundary
22. Content-Type: application/soap+xml
23. Content-Transfer-Encoding: binary
24. Content-ID: <http://GPH.org/Extractions/GPH2006-06-30-1430.xml>

// binary octets for XML attachment
--MIME_boundary
```

The extracted files are distinguished from each other by naming them after the data participant (GPH in this case), and a full timestamp with the date (2006-06-30) and the time (1430) the file was extracted (line 18). These MTOM attachments will be XOP'd and sent in a standardized SOAP-envelope.

### 6.2.2 Standardized SOAP-Envelope

The SOAP-Envelope header will be extended with pre-specified WS-* features, such as MTOM, XML Encryption, WS-Security and XML Digital Signatures. These SOAP-header modules/features are essential as they specify standard values for messaging parameters, addressing parameters, and security protocols. A standardized SOAP-Envelope will go a long way in ensuring and enhancing interoperability in the proposed WGDW. A hypothetical standardized SOAP-Envelope, with references to the standard SOAP- Envelope namespace, the XML Digital Signature namespace, the XML Encryption namespace, and the WS-Security namespace, is listed in the following:

```
 <?xml version=“1.0” encoding = “UTF-8”?>
  <soapenv:Envelope xmlns:soapenv=“http://www.w3.org/2001/12/soap-envelope”
                xmlns:ds=“http://www.w3.org/2000/09/xmldsig#”
                xmlns:enc=“http://www.w3.org/2001/04/xmlenc#”
                xmlns:wsse= “http://schemas.xmlsoap.org/ws/2002/04/secext”
```

The extension of the header commences on the following listing with the WS-Security header element.

```
        <soapenv:Header>
                <wsse:Security wsse:actor = “urn: receiverSP”>
                        soapenv:mustUnderstand="1"
                        <wsse:BinarySecurityToken Id=”CertToken”
                                ValueType=“wsse:X509v3”
                                EncodingType=“wsse:Base64Binary”>
                                ……..
                        <wsse:BinarySecurityToken>
```

The recipient Service Participant (wsse:actor) has to meet the required security elements. The soapenv:mustUnderstand="1" element specifies that it is compulsory for the Service Participant to process the SOAP header entry, else message processing fails. Standardization of authentication can be provided either through a UserNameToken element (with username/password combination), or through BinarySecurityToken element such as X.509 certificates and Kerberos tickets. The string-value ID can be standardized in the proposed framework. The ValueType and EncodingType values can also be standardized through available pre-defined types, such as X509v3 and Base64Binary.

The elements for XML encryption employed come next. This consists of the algorithm utilized (encryption method), the key used and the cipher value result.

```
<xenc:EncryptedKey>
        <EncryptedData Type=“http://www.w3.org/2001/04/xmlenc#Element”
                <xenc:EncryptionMethod Algorithm =
                        “http://www.w3.org/2001/04/xmlenc#aes128-cbc”/>
                <ds:KeyInfo>
                        <ds:KeyName> …</ds:KeyName>
                <ds:KeyInfo>
                <xenc:CipherData>
                        <xenc:CipherValue> … </xenc:CipherValue>
                </xenc:CipherData>
```

```
    </EncryptedData>
</xenc:EncryptedKey>
```

The encryption algorithm can be standardized through algorithms such as Advanced Encryption Standard (Rijndael).

Next, come the digital signature elements.

```
<ds:Signature>
    <ds:sSignedInfo>
    <ds:CanonicalizationMethod Algorithm=
        “http://www.w3.org/TR/2001/REC-xml-c14n-20010315’>
        <ds:SignatureMethod Algorithm=
            “http://www.w3.org/2000/09/xmldsig#dsa-sha1”/>
        <ds:Reference>
            <ds:Transforms>
            <ds:Transform Algorithm=
                “http://www.w3.org/TR/2000/CR-xml-c14n-
            20001026”/>
            </ds:Transforms><
        </ds:Reference>
    </ds:SignedInfo>
        <ds:SignatureValue>….</ds:SignatureValue>
        <ds:KeyInfo>
            <wsse:SecurityTokenReference>
                <wsse:Reference URI=’#CertToken”/>
            <wsse:SecurityTokenReference>
        </ds:KeyInfo>
</ds:Signature>
</wsse:Security>
</soapenv:Header>
```

The listings above merely indicate how a SOAP-Envelope can be standardized through the extension of the header using WS-* features. However, the actual standardized features and corresponding values can only be specified through a standards-body tasked with the standardization process.

### 6.2.3 SOAP Security Model

In order to secure HTTP connections between the Data Participant (HTTP Server) and the Service Participant (Grid Portal), WS-SecureConversation is recommended. This will present a Server-determined encrypted tunnel or communication between the two parties. To validate the identification and authentication of a Data Participant, a standardized token (such as X.509 certificate) must be agreed upon by the standards-body mentioned in the previous section. This will be included as standard Web services security element in the header of the standardized SOAP-Envelope. Security Assertions Markup Language (SAML), which could also be included as WS-Security header element, can be used to enforce security assertions. SAML can be utilized for authorisation (access-control) into the Grid Portal.

XML-Digital Signatures and XML-Encryption will be utilized to provide confidentiality and integrity. A standardized encryption algorithm –in this case, Advanced Encryption Standard (Rijndael) (NIST, 2001)– is recommended; the WS-Security EncryptedKey element can be utilized to encrypt a key with the receiver's public key – along with a standardized encryption mechanism that must be implemented.

A standardized hash-algorithm is initially employed to generate a hash for the XML-attachment before XOP packaging. This will guarantee integrity of the attachment. The hash and the XML-attachment can thereafter be encrypted with the Data Participant's private key. This will also be used to authenticate the Data Participant. The attachment can then be converted into a base64Binary and XOP'd into a MIME package using MTOM.

After all the XOP:Include elements have been added, a hash value can be generated with the standardized hash-algorithm, for the entire SOAP-body. This will guarantee integrity of data in the SOAP-body. The SOAP-body, the encrypted attachment and the hash value can then be encrypted with Service Participant's public key. This will guarantee confidentiality between the Data Participant and Service Participant, as only the Service Participant can be able decrypt the SOAP-message with its private key.

Upon receiving the SOAP-message, the Service Participant will utilize its private key to decrypt the message. Thereafter, it will employ the standardized hash algorithm to generate a hash for the entire SOAP-body. The two hash values will be compared, and if they are the same, it will imply that the integrity of the SOAP-body has been preserved. Next, the XML-Attachment will be decrypted, and a hash generated with the standardized hash algorithm. The two hash values will also be compared, and if they are the same, it will imply that the integrity of the attachment has been preserved.

### 6.3 UMTS Wireless Connection

Data Participant's HTTP Server will initiate a data file transfer through a SOAP-Request to the Data Participant Router (DPR). The SOAP-Request will contain information about the size of the file to be transferred. The SOAP-Request will trigger the DPR to perform a General Packet Radio Service (GPRS) Attach procedure and furthermore activate a Packet Data Protocol (PDP) context and establish a Radio Access Bearer (RAB) (Chen & Zhang, 2004). The DPR performs these three processes in order to establish a Universal Mobile Telecommunications Service (UMTS) wireless connection and be able to send packets over the 3GPP packet switched network. The DPR will send a SOAP-Response to the HTTP Server on whether there was an error (thus, the request is aborted, and a new one should be instituted) or the request was successful (thus, a successful GPRS Attach, PDP Context Activation and RAB Establishment) (Eberspacher *et al.*, 2001).

The DPR utilizes the GPRS Attach procedure to register with the Serving GPRS Support Node (SGSN) on the Packet Switched (PS) Core Network (CN) domain (3GPP, 2006[4]). The GPRS Attach procedure also enables the DPR to access the services that are provided by the SGSN. Once the GPRS Attach has been performed successfully, a PDP context will established and activated on the DPR and in the PS CN domain. PDP Context Activation enables the DPR to send and receive user packets through its PDP address. If the PDP Context Activation is performed successfully, it will trigger the PS CN domain to set up the Radio Access Bearer that is required transmit user packets to and from the DPR. Thus, once the PDP context has been activated successfully, it will enable the DPR to send and receive user packets over the PS CN domain (Lee, 2006). Optionally, the DPR can register with the Internet Protocol (IP) Multimedia Subsystem (IMS) if it desires to utilize IP-based real-time voice or multimedia services offered by the IMS (3GPP, 2006[5]). The mandatory processes -GPRS Attach procedure, PDP Context Activation and Radio Access Bearer Establishment- a DPR must perform to send and receive user packets over the PS CN domain are discussed in detailed in the following sections.

### 6.3.1 GPRS Attach Procedure

The GPRS Attach procedure can only be initiated through the Data Participant Router (DPR) (NB: These discussions also apply to Service Participant Router (SPR), although the DPR is used here as the example). However, the GPRS Detach procedure can be initiated either through the DPR or the network (i.e., the SGSN or the Home Location Register (HLR)) in order to disable the states instituted by the GPRS attach procedure on the network nodes and the DPR (Chen & Zhang, 2004). The SGSN that the DPR is trying to attach to is known as the new SGSN. The SGSN (if any) that was utilized by the DPR before attaching to the new SGSN is known as the old SGSN. (3GPP, 2006[4])

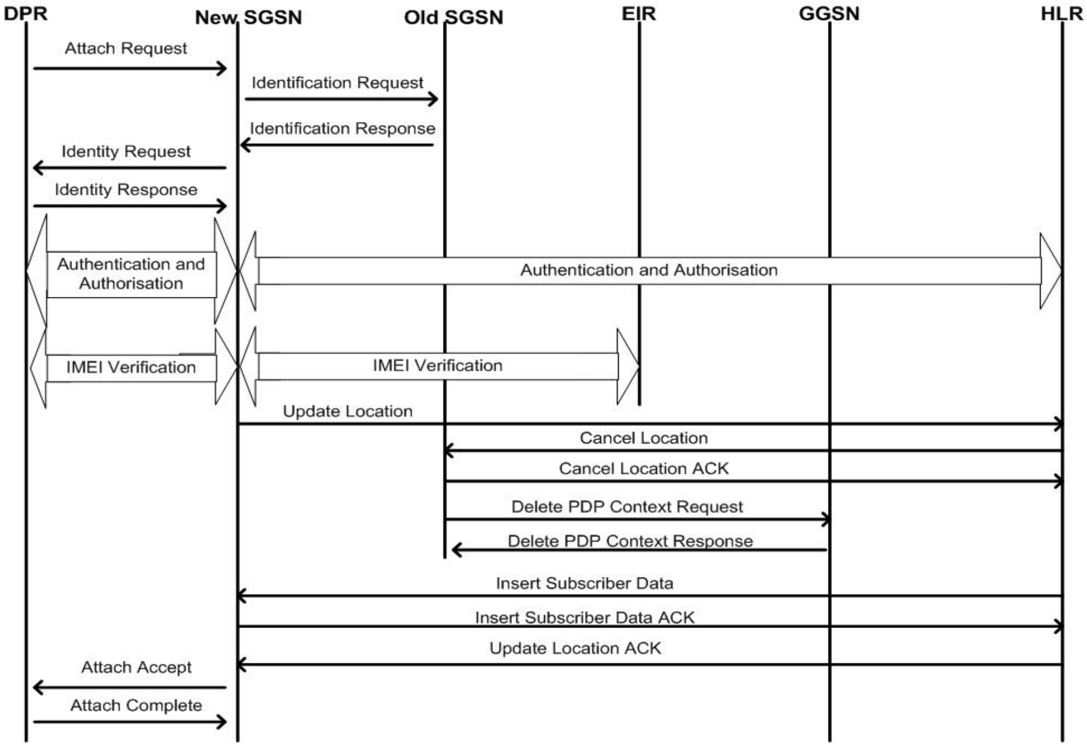

**Figure 6.2: GPRS Attach Procedure (3GPP, 2006[4])**

The DPR initiates the GPRS Attach procedure through sending an Attach Request message to the SGSN, as shown in Figure 6.2. The main information (among other information) contained in a GPRS Attach Request message includes (3GPP, 2006[4]):

- *Identifiers of the DPR*: The DPR utilizes either the Packet Temporary Mobile Subscriber Identity (P-TMSI) or the International Mobile Subscriber Identity (IMSI) (only if there is no P-TMSI), to identify the GPRS Attach Request message. This prevents adversaries from learning the mapping between the DPR’s P-TMSI and IMSI in case the GRPS Attach Request message is intercepted.
- *P-TMSI Signature*: The P-TMSI Signature is a three-octet number that is assigned to the DPR by the SGSN, which assigns the P-TMSI. The P-TMSI Signature is utilized by the SGSN in authenticating a P-TMSI. It can also be utilized by the DPR in authenticating the network node, which assigns the P-TMSI.

- *Attach Type*: The Attach Type specifies whether the Attach Request is for GPRS attach only, GPRS Attach while already IMSI attached, or combined GPRS/IMSI Attach.

After receiving the GPRS Attach Request message from the DPR, the SGSN carries out two key functions (Korhonen, 2001; 3GPP, 2006[1]):

- Authenticates the DPR.
- Performs location update when necessary.

The SGSN utilizes the DPR's IMSI as the DPR's permanent identity in authenticating the DPR (Eberspacher *et al*., 2001). Then again, because of security considerations, the GPRS Attach Request message will more often than not contain the DPR's P-TMSI as long as there is a valid P-TMSI present. If the DPR provides the P-TMSI in the Attach Request message, the SGSN will have to employ other means in order to establish the DPR's IMSI. There are two scenarios that exist:

- Scenario 1: The new SGSN assigned the DPR's P-TMSI. In this scenario the SGSN will know the DPR's IMSI as it needed the IMSI in order to assign the DPR the P-TMSI.
- Scenario 2: The old SGSN assigned the DPR's P-TMSI.

In the second scenario above, the new SGSN can employ two basic ways to retrieve the DPR's IMSI: Ask the DPR directly, or ask the old SGSN. The new SGSN will at first try to get the DPR's IMSI from the old SGSN. SGSNs are usually connected through a high-speed wireline network, and thus it will be a lot faster for a communication between SGSNs on this wireline as opposed to communication with the DPR over the bandwidth-limited radio access network. Acquiring information from the old SGSN also cuts down over-the-air signalling overhead. The new SGSN requests for the DPR's IMSI by sending an Identification Request message to the old SGSN. If the old SGSN knows the DPR's IMSI, it responds with an Identification Response message that contains the DPR's IMSI. However, if the old SGSN does not know the DPR's IMSI, the new SGSN will have to retrieve it from the DPR. The new SGSN accomplishes this through sending an Identity Request message to the DPR. The DPR will respond with an Identity Response that contains its own IMSI.

After obtaining the DPR's IMSI, the SGSN will initiate the security procedures that authenticate and authorize the DPR. Even though authentication is based mainly on the IMSI, the SGSN can optionally check the DPR's International Mobile Station Equipment Identity (IMEI). The SGSN performs this via sending an Identity Request message to the DPR. The DPR will respond with an Identity Response that contains its IMEI (Chen & Zhang, 2004).

Once the DPR has been positively authenticated and authorised, the SGSN will update the DPR's location with the mobile's HLR during any of the following scenarios:

- If the DPR is attaching to the new SGSN for the first time
- If the new SGSN's SGSN Number has changed since the last GPRS Detach. The SGSN Number is utilized by non-IP protocols inside the PS CN domain to identify the SGSN.

To initiate a location update, the SGSN sends an Update Location message to the DPR's Home Location Register (HLR) (Bannister *et al.*, 2004). After receiving the Update Location message from the new SGSN, the HLR will initiate a cancellation of the DPR's location state maintained in the old SGSN. The old SGSN will, if there are active contexts for the DPR, delete the PDP contexts through sending a Delete PDP Context Request message to the Gateway GPRS Support Node (GGSN).

The HLR will moreover send the DPR's GPRS service subscription information (thus, the GPRS services the DPR can access) to the new SGSN utilizing the Insert Subscriber Data message (Figure 6.2). The Insert Subscriber Data message also carries information that enables the SGSN to verify if the DPR is authorized the particular SGSN. The new SGSN responds through an Insert Subscriber Data ACK message, which triggers the HLR to send an Update Location ACK message back to the SGSN, and thus complete the location update procedure.

In order to complete a successful GPRS Attach, the SGSN will send an Attach Accept message to the DPR (3GPP, 2006[4]). On the Attach Accept message the new SGSN may include a new P-TMSI, if it has allocated the DPR a new P-TMSI. In this instance, a P-TMSI signature for the new P-TMSI will also be contained in the Attach Accept message. Nonetheless, the DPR's IMSI will not be contained in the Attach Accept message. As a result, even if an adversary can intercept the Attach Accept message, it will not be possible for them to learn the mapping between the DPR's P-TMSI and IMSI (Bates, 2002).

If the new P-TMSI is not the same as old P-TMSI utilized by DPR, the DPR will explicitly acknowledge the amendment through an Attach Complete message to the new SGSN, and thus complete the GPRS Attach procedure (Chen & Zhang, 2004).

### 6.3.2 Activation and Modification of the PDP Context

The primary functions carried out by PDP Context Activation are (3GPP, 2006[1]):

- *PDP Address allocation*: The network assigns a PDP address (i.e., IP address in our case) to the DPR.
- *CN Bearer Establishment*: The network generates and activates the PDP context on GGSN and SGSN and sets up all the essential bearers between SGSN and GGSN that will be utilized to transport the DPR and signalling traffic for the activated PDP context.
- *RAB Assignment*: The network sets up Radio Access Bearers to transport DPR traffic.

PDP Context Activation can be initiated by either the DPR if it needs to send packets through the 3GPP network, or the network, if it has packets for the DPR. The two processes are discussed in the following sections.

#### *6.3.2.1 DPR-Initiated PDP Context Activation and Modification*

The DPR initiates PDP context activation through conveying an Activate PDP Context Request to the SGSN, as depicted in Figure 6.3. The main information (among other information) contained in an Activate PDP Context Request message includes (Chen & Zhang, 2004; 3GPP, 2006[1]):

- *PDP Address*: The PDP address field can contain either the PDP Address specified by the DPR or 0.0.0.0 in order to specify to the network that the DPR desires to obtain a PDP Address from an external IP network.
- *Network-layer Service Access Point Identifier (NSAPI)*: An unused NSAPI the DPR utilizes to send user packets that originate from the PDP address and to receive user packets, which are destined to the PDP address.
- *PDP Type*: Specifies the type of PDP employed by the DPR, for instance IP, Point-to-Point Protocol (PPP) (Haskin & Allen, 1998; Simpson, 1994) or X.25 (Cisco, 1992).
- *Access Point Name (APN)*: Contains the APN required by the DPR.
- *QoS Requested*: Contains the QoS profile required by the DPR.
- *PDP Configuration Options*: Utilized by the DPR to communicate optional parameters directly with the GGSN (thus, the SGSN will not interpret these parameters)

The DPR utilizes the PDP Address field in the Activate PDP Context Request message to notify the network whether it prefers to employ a static PDP address or to be allocated a dynamic PDP address (3GPP, 2006[2]). In order to utilize a static PDP address, the DPR will include a static address in the PDP Address field of the Activate PDP Context Request message. However, in order to have the PDP address allocated dynamically by the PS domain or an external packet network, the DPR will set the PDP Address field of the Activate PDP Context Request message to zero.

The DPR utilizes an APN to choose a service (or a GGSN) in the PS Domain or a contact point in an external packet network. An APN consists of two main parts (Chen & Zhang, 2004):

- *APN Network Identifier*: This is a mandatory part utilized to identify an external packet network that a GGSN is connected to or a PS domain service which was requested by the DPR.
- *APN Operator Identifier*: This is an optional part utilized to identify the PLMN that houses the GGSN.

The Name Syntax of an APN is similar to that of Internet Domain Names (Mockapetris, 1987). Thus, an APN is represented as Label1.Label2.Label3. Each label can consist of only alphabetic characters (A-Z and a-z), numbers (0-9), and hyphens. The beginning of a label should only be a digit or an alphabetic character. The APN Network Identifier will consist of at least one label. A Domain Name System (DNS) can be utilized to translate an APN into an IP Address.

An SGSN utilizes the DPR's APN and the configuration information stored on the SGSN to pick the GGSN that will act as the DPR's serving GGSN. Various GGSNs can possess different capabilities and offer diverse services to the DPRs. For instance, a network operator may configure several GGSNs to facilitate dynamic PDP address assignment by external packet networks, while only local dynamic PDP address assignment will be facilitated by other GGSNs (Korhonen, 2001).

The SGSN utilizes the configuration information to be acquainted with each subscriber's PDP Type and the APN the network operator configured to support each PDP type (3GPP, 2006[1]). If the APN obtained from the DPR (in the Activate PDP Context Request) matches an APN in the DPR's subscription list, the SGSN will utilize the APN to query the DNS so as to determine the IP Address of the GGSN for the DPR. When there is no APN supplied in the Activate PDP Context Request, it can utilize an APN in the DPR's subscription profile which the network operator has configured in order to support the PDP type identified by the Activate

PDP Context Request. If the APN obtained in the Activate PDP Context Request does not match any APN on the DPR's subscription list, the SGSN will reject the DPR's Activate PDP Context Request (Chen & Zhang, 2004).

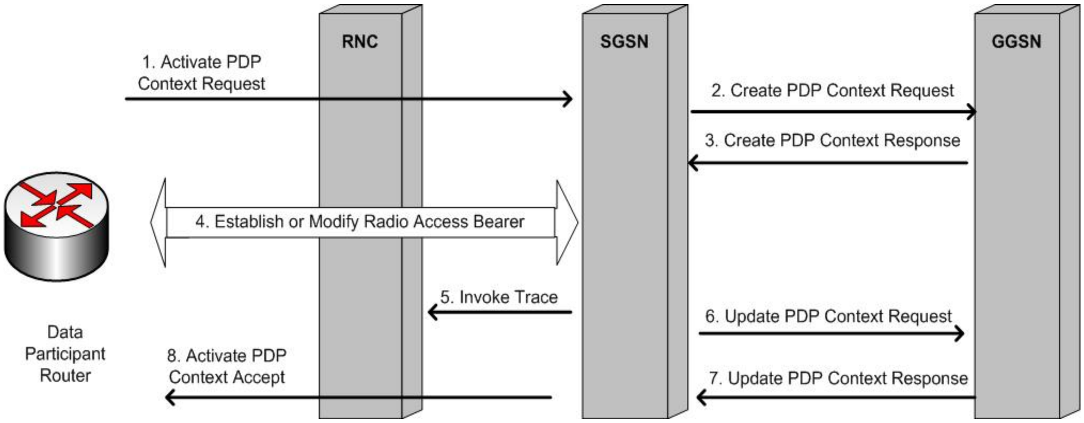
RNC
SGSN
GGSN
1. Activate PDP Context Request
2. Create PDP Context Request
3. Create PDP Context Response
4. Establish or Modify Radio Access Bearer
5. Invoke Trace
6. Update PDP Context Request
7. Update PDP Context Response
8. Activate PDP Context Accept
Data Participant Router

**Figure 6.3: 3GPP DPR-Initiated PDP Context Activation (Based on 3GPP, 2006[1])**

After the SGSN has chosen the DPR's serving GGSN, the SGSN conveys a Create PDP Context Request to the chosen GGSN to request the GGSN to set up a PDP context for the DPR and the GPRS Tunnelling Protocol (GTP) Tunnel between the SGSN and the GGSN that will be utilized in transporting user packets for this PDP context (2 in Figure 6.3). The main information contained in a Create PDP Context Request message comprises (3GPP, 2006[1]):

- *NSAPI*: Copied directly from the same field in the Activate PDP Context Request message received from the DPR.
- *PDP Type*: Copied directly from the same field in the Activate PDP Context Request message received from the DPR.
- *PDP Address*: Copied directly from the same field in the Activate PDP Context Request message received from the DPR.
- *APN*: Contains the APN Network Identifier of the APN chosen by the SGSN for the DPR.
- *QoS Negotiated*: QoS profile the SGSN agrees to support for the DPR.
- *Tunnel Endpoint Identifier (TEID)*: Generated by the SGSN based on the DPR's IMSI and on the NSAPI in the Activate PDP Context Request

message received from the PDR. This TEID identifies the SGSN side of the GTP tunnel to be set up for the DPR between the SGSN and the GGSN. This TEID will be utilized by the GGSN to tunnel PDP packets over the GTP tunnel to the SGSN.

- *Selection Mode*: Specifies whether the APN contained in the Create PDP Context request message was an APN subscribed by the DPR, or a non-subscribed APN chosen by the SGSN.
- *Charging Characteristics*: Specifies the type of charging the PDP context is liable for.
- *PDP Configuration Options*: Copied directly from the same field in the Activate PDP Context Request message received from the DPR.

Once the GGSN receives the Create PDP Context request, it utilizes the APN in the Create PDP Context request to either activate a GGSN service (e.g., dynamic PDP address allocation) or locate a contact point in an external packet network (e.g., a PDP address server. The APN can be mapped to a GGSN service through the following ways (Bates, 2002; Chen & Zhang, 2004):

- If an APN corresponds to the domain name of a GGSN, the APN Network Identifier will be deduced by the GGSN as a request for a service specified by the same APN Network Identifier.
- The first label of an APN Network Identifier may be a Reserved Service Label. Each Reserved Service Label is mapped to a GGSN service.

Thereafter, the GGSN will generate a new PDP context and insert it into its PDP context table. Moreover, the GGSN will set up a CN Bearer (i.e., a GTP Tunnel) between the GGSN and the SGSN for the PDP context (Lee, 2006). The GGSN will then send a Create PDP Context Response message back to the SGSN. The main information contained in a Create PDP Context Response message comprises (3GPP, 2006[1]):

- *TEID*: The TEID generated by the GGSN to identify the GGSN side of the GTP tunnel established for the DPR's PDP context. This TEID will be utilized by the SGSN when tunnelling user packets over the GTP tunnel to the GGSN.

- *PDP Address*: The PDP address field can contain either the PDP Address allocated by the GGSN to the DPR or 0.0.0.0 in order to specify to the network that the DPR desires to obtain a PDP Address from an external IP network.
- *QoS Negotiated*: Contains the QoS profile consented to by the GGSN.
- *PDP Configuration Options*: This information element is transmitted by intermediate nodes (i.e., the SGSN or nodes in a RAN) transparently to the DPR (thus, the intermediate network nodes will not be able to interpret this element).

If a DPR desires to acquire a dynamic PDP address from an external packet network, the SGSN and the GGSN will initially generate an active PDP context for the DPR, which does not have a valid PDP address. This active PDP context will enable a DPR to convey PDP packets through the PS domain to contact a PDP address server in the external server to obtain a PDP Address. The GGSN will examine the packets between the DPR and the external PDP address server in order to establish the PDP address allocated to the DPR by the external network (Bates, 2002). After the GGSN has established the PDP address allocated to the DPR, it will notify the SGSN of the allocated PDP address in the PDP Address field of the Create PDP Context Response message, conveyed back to the SGSN, as a response to the Create PDP Context Request. The SGSN will consecutively notify the DPR about the allocated PDP address in the PDP Address field of the Activate PDP Context Accept message, conveyed back to the DPR as a response to the Activate PDP Context Request (Chen & Zhang, 2004).

Once the SGSN has received the Create PDP Context Response, it will employ the RAB Assignment procedure to activate the process that sets up Radio Access Bearers (RABs) for the DPR. If the RABs have been set up successfully, the SGSN may convey an Invoke Trace message to the RNC to request the RAN to begin gathering statistics on the network resources utilized for the PDP context.

### *6.3.2.2 Network-Requested PDP Context Activation*

If the GGSN has user packets for the Service Participant Router (SPR) but does not have an active PDP context for its PDP address, the GGSN can invoke a Network-Requested PDP Context Activation procedure which will establish and activate the PDP context for the PDP address, so that the packets can be delivered to the SPR (3GPP, 2006[2]). The GGSN must have static information regarding the PDP address in order to be able to support the Network-Requested PDP Context Activation for the PDP address. For instance, in order to determine the SPR's serving SGSN, the GGSN must have the SPR's IMSI that will be utilized to query the HLR. The type of static information stored, the location it will be stored, and the way a GGSN will retrieve it are considered to be implementation issues, and thus, 3GPP does not standardize them (Lee, 2006).

The GGSN initiates the Network-Requested PDP Context Activation by conveying a Send-Routing-Information-for-GPRS message to the HLR to acquire the address of the SPR's serving SGSN (2 in Figure 6.4). This message contains the SPR's IMSI that the HLR utilizes to determine if the request can be granted and to search the HLR database for the information requested concerning the SPR. The HLR responds with a Send-Routing-Information-for-GPRS-ACK message. This message will contain either the address of the SPR's serving SGSN, if the HLR resolves to grant the Send-Routing-Information-for-GPRS message, or an error cause if the HLR cannot grant the Send-Routing-Information-for-GPRS from the GGSN (3GPP, 2006[1]).

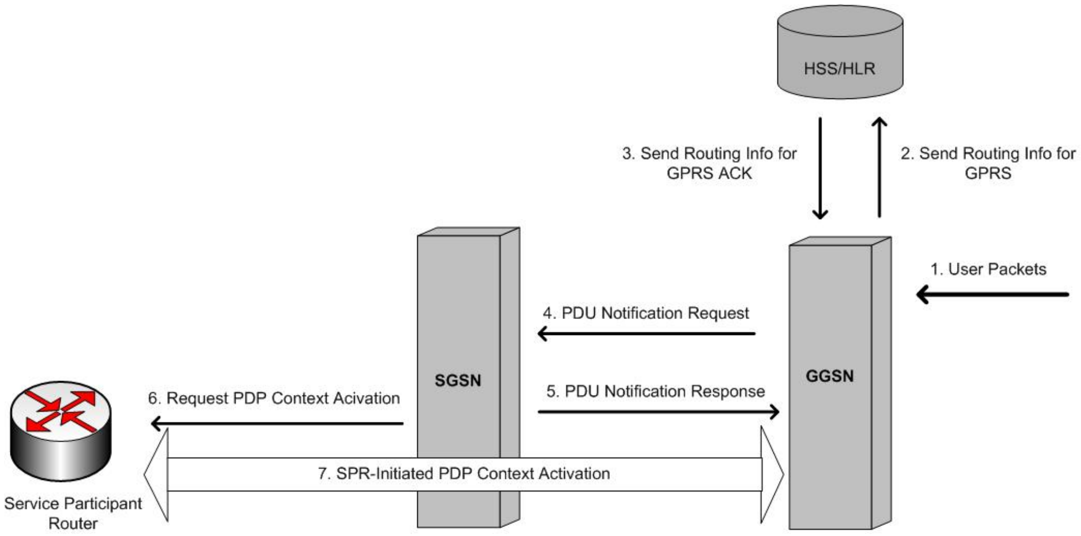
HSS/HLR
3. Send Routing Info for GPRS ACK
2. Send Routing Info for GPRS
1. User Packets
4. PDU Notification Request
SGSN
GGSN
5. PDU Notification Response
6. Request PDP Context Acivation
7. SPR-Initiated PDP Context Activation
Service Participant Router

**Figure 6.4: 3GPP Network-Requested PDP Context Activation (Based on 3GPP, 2006[1])**

The GGSN will then convey a Packet Data Unit (PDU) Notification Request message to the SPR's serving SGSN to request the SGSN to instruct the SPR to initiate PDP context activation. The PDU Notification Request message will contain the SPR's IMSI, the PDP type, the PDP Address that the PDP context should be activated for, and the APN, which the SGSN and the SPR will utilize to determine which GGSN to employ (Chen & Zhang, 2004).

Once the SGSN receives the PDU Notification Request, it will at first notify the GGSN that it will honour the GGSN's request through a PDU Notification Response message to the GGSN. Thereafter, the SGSN will convey a Request PDP Context Activation message to the SPR to instruct the SPR to initiate the SPR-Initiated PDP Context Activation procedure to activate the PDP context for the PDP address, specified in the Request PDP Context Activation message (6 in Figure 6.4). The Request PDP Context Activation message will also contain the APN that the SGSN obtained from the GGSN. This APN will be utilized by the SPR in the SPR-Initiated PDP Context Activation procedure (3GPP, 2006[1]).

### 6.3.3 Radio Access Bearer Assignment

The Radio Access Bearer (RAB) Assignment procedure carries out the assignment, modification and release of Radio Access Bearers. The RAB Assignment procedure cannot be started directly through an SPR in Release 5; it can only be started through the network (Lee, 2006). RABs for packet-switched services are specifically initiated by the SGSN after being triggered by other network entities in the CN or Radio Access Network (RAN) (Korhonen, 2001). For instance, the SGSN can initiate the RAB Assignment procedure after receiving a Create PDP Context Response message from the GGSN, notifying the SGSN of a successful PDP context activation on the GGSN for the SPR, during the PDP context activation process.

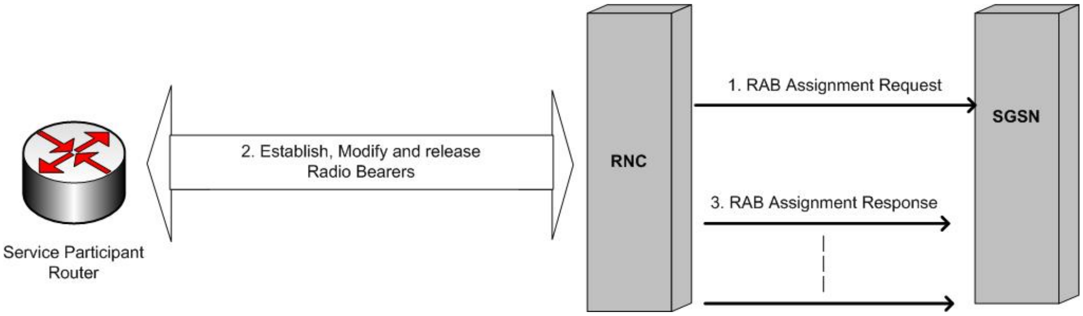
Service Participant Router
2. Establish, Modify and release Radio Bearers
RNC
1. RAB Assignment Request
3. RAB Assignment Response
SGSN

**Figure 6.5: 3GPP Radio Access Bearer Assignment (Based on 3GPP, 2006[1])**

The SGSN starts the RAB Assignment procedure by conveying a RAB Assignment Request message to a Radio Network Controller (RNC), requesting the RNC to set up, modify or release one or more RABs (1 in Figure 6.5). After receiving the RAB Assignment Request, the RNC will start a process that establishes, modifies or releases the Radio Bearers for the RABs specified in RAB Assignment Request message (Chen & Zhang, 2004). The establishment procedures of Radio Bearers are specific to each radio system. For instance, Global System for Mobile Communication and Enhanced Data rates in GSM Environment (GSM/EDGE) RAN and Universal Mobile Telecommunications Service (UMTS) Terrestrial RAN (UTRAN), utilize Radio Resource Control (RRC) to set up, maintain and release Radio Bearers (Lee, 2006).

The RNC utilizes RAB Assignment Responses to notify the SGSN about the results of the RAB Assignment Request. Multiple RAB Assignment Responses can be forwarded to the SGSN for each RAB Assignment Request to report the progress and status of the actions carried out by the RNC to satisfy the RAB Assignment Request (Chen & Zhang, 2004). For instance, the RNC can temporarily queue the request to establish a RAB while it processes other RABs. In this scenario, the RNC can convey a first RAB Assignment Response to notify the SGSN about the request being queued, and thereafter, a second RAB Assignment Response following a successful Radio Bearer establishment for RAB Assignment Request. Once the Radio Access Bearers are established successfully, the wireless UMTS connection will be set up and ready to transfer files (3GPP, 2006[1]).

## 6.4 Transferring of Data

As stated before, the Data Participant's HTTP Server (DPHS) will initiate a data file transfer through a SOAP-Request to the Data Participant Router (DPR) ((1) in Figure 6.7). The SOAP-Request will trigger the DPR to establish a wireless UMTS connection with the SPR, through performing a GPRS Attach procedure and furthermore, activating a PDP context and establishing Radio Access Bearers, as explained in section 6.3 (2). If the wireless connection is not set up, the DPR will send an error response to the HTTP Server detailing the error (3A). However, if the connection was set up successfully, the SPR will send a SOAP-Request about the transfer to the Grid Portal (3B). The Grid Portal will then acknowledge the request, stating whether the request has been queued or the data transfer can commence (4A).

Once the job transfer is ready for execution, the Globus Toolkit's Grid Resource Allocation and Management (GRAM) system in the Grid Portal will generate a Job Manager Service (JMS), to be utilised in managing the transfer (5). Before a job can be executed, the Execution Management Services (EMSs) have to set up the environment where the job will be executed (as illustrated in Figure 6.6).

### 6.4.1 Setting up the EMS for a Job Execution

The following processes will be performed before any job can be executed. The Job Manager Service (JMS) initiates the process by generating a description (written in Job Submission Description Language (JSDL)), an optional initial termination time for the job resource, and an optional state notification subscription request. Thereafter, the JMS will call the Execution Planning Service (EPS) to obtain a job scheduler (A). The EPS, in turn, calls the Candidate Set Generator (CSG) (B), which also calls the Information Services (C) in order to determine where the job can be executed, based on binary availability and policy settings. These settings may include special application requirements, security and performance issues concerning the available service containers. Information services are essentially databases that contain attribute metadata concerning resources. After finding a candidate container, the EPS will first confirm that the information about it is accurate, and then select it (D).

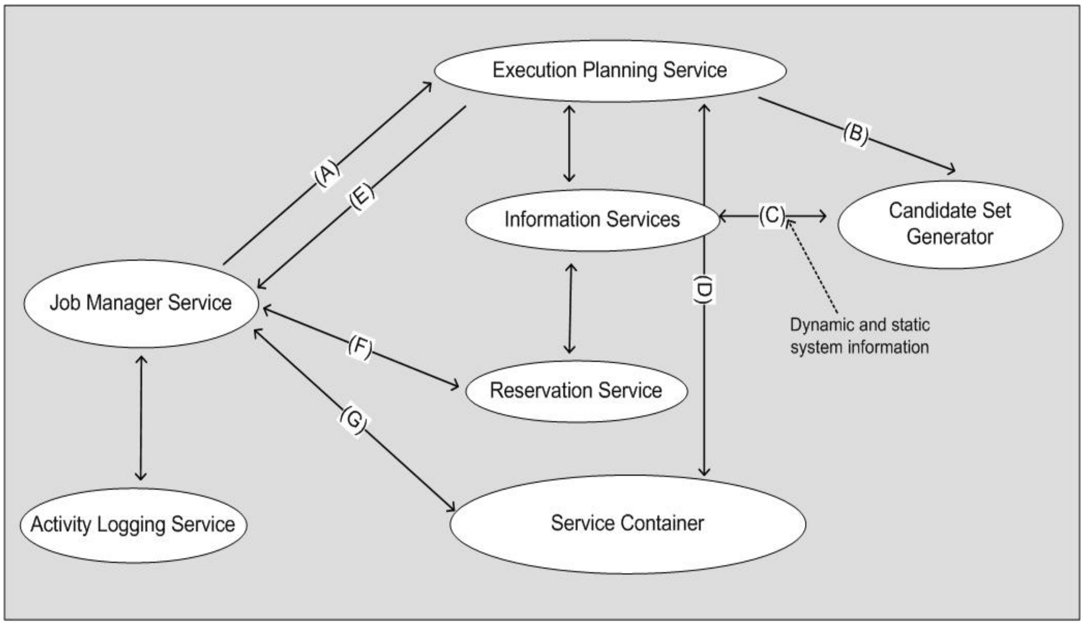

**Figure 6.6: Setting up EMS Services for Job Execution**

Next, the EPS will return the job scheduler to the JMS (E). Thereafter, the JMS will interact with the reservation (if necessary, and also deployment and configuration services) to set up the environment where the job will execute (F). As an example, in a data transfer execution, reservable resources can consist of (but are not limited to) computing resources, such as CPUs and memory, storage space, network bandwidth, etc. The JMS will then invoke the service container to start the job (G). The container has a WS-Addressing endpoint reference or EPR, which is a handle that can be queried to obtain the job's status, destroy the job, or subscribe to receive asynchronous notifications on the job status or resource property changes (Foster, 2005). The Activity Logging Service will log all the activities, tasks and processes that are taking place, for the purposes of auditing, metering, and provenance. The container will notify the JMS once the job terminates. If, for a peculiar reason, the job terminates, the whole cycle might have to be repeated again.

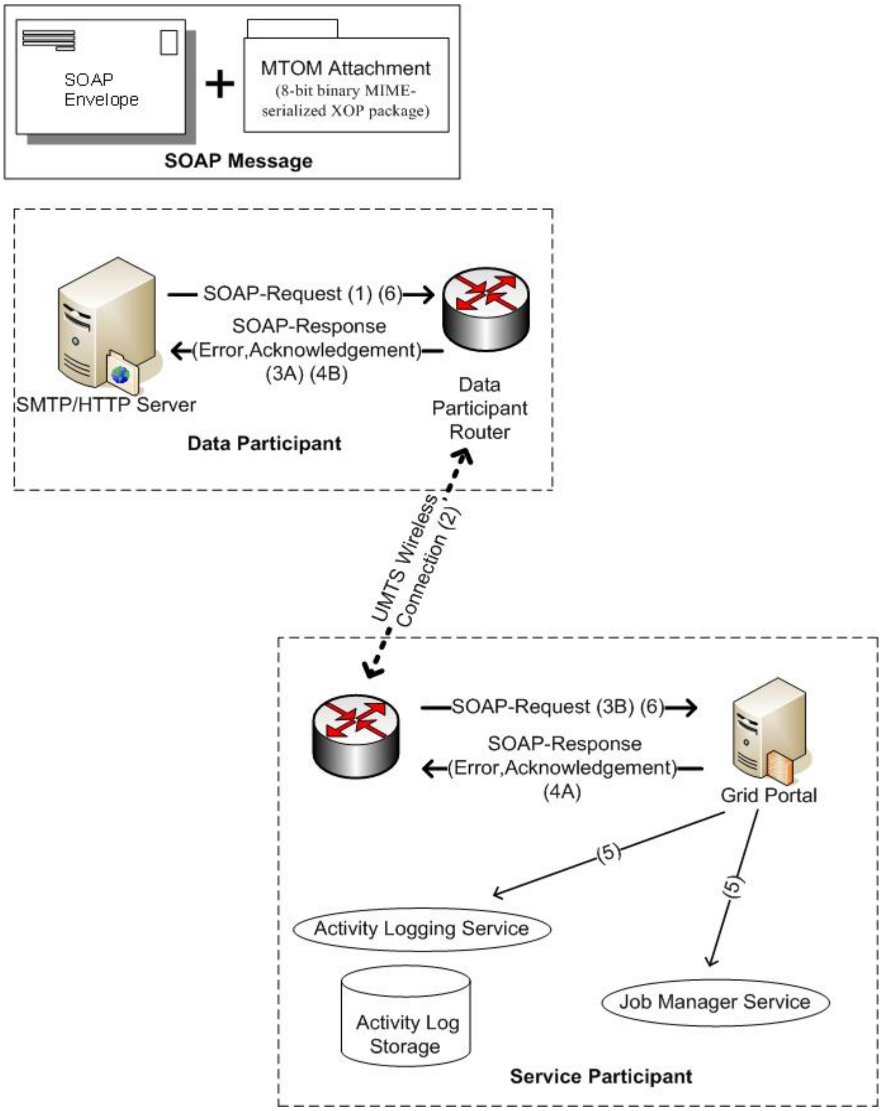

**Figure 6.7: Data Transfer from the DPHS into the Grid Portal**

Once ready for execution, the Grid Portal, through the service container, will acknowledge the request, stating that the data transfer can commence (4A). Upon receiving the response that the transfer can begin (4B); the DPHS will start sending the extracted files as MTOM attachments on the SOAP-Messages ((6) in Figure 6.7).

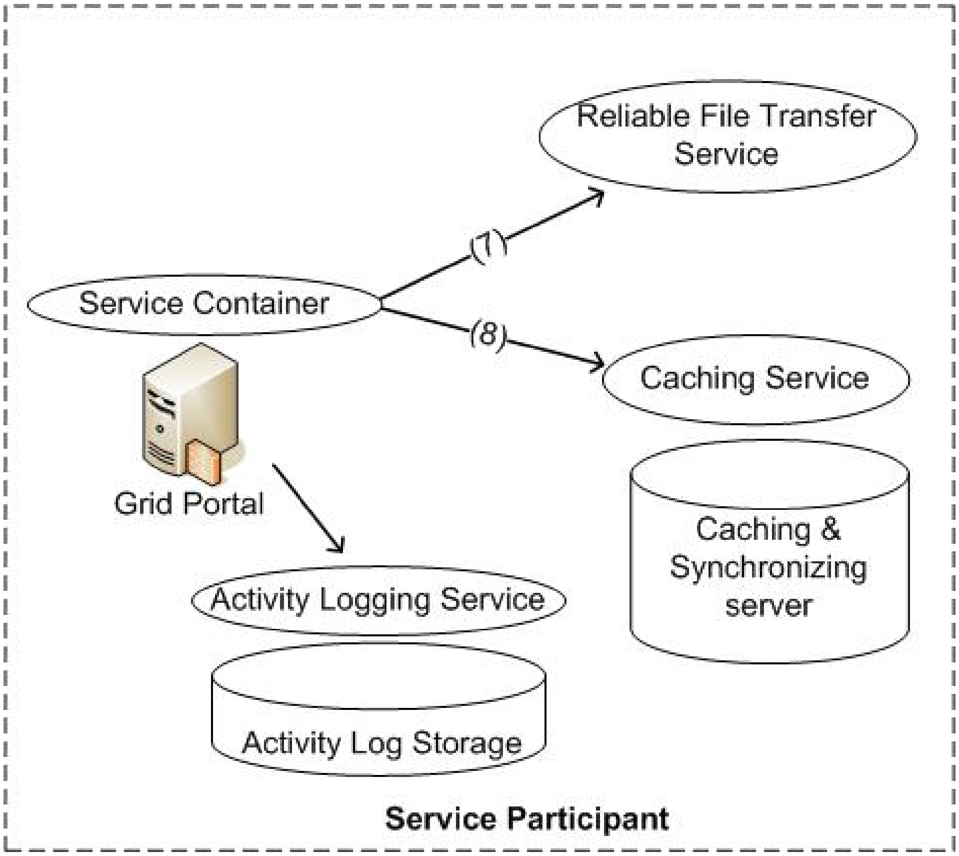

**Figure 6.8: Data Transfer from the Grid Portal into the CSS**

The service container will generate a Reliable File Transfer Service that will be utilized to transfer (also known as staging) the files into the Catching and Synchronization Server (CSS) ((7) in Figure 6.8).

The following listing shows an example of a file stage in.

```
% cat gram.epr
  <factoryEndpoint
        xmlns:gram="http://www.globus.org/namespaces/2004/10/gram/job"
        xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <wsa:Address>
```

```
6.
        https://ubuntuhealth.org:2666/wsrf/services/ManagedJobFactoryService
7.      </wsa:Address>
8.              <wsa:ReferenceProperties>
9.                      <gram:ResourceID>Condor</gram:ResourceID>
10.             </wsa:ReferenceProperties>
11.     </factoryEndpoint>

12.  <job>
13.    <fileStageIn>
14.         <transfer>
15.             <sourceUrl> gsiftp://Portal.Server:9000/tmp/newData </sourceUrl>
16.             <destinationUrl>file://${GLOBUS_SCRATCH_DIR}/newData
                </destinationUrl>
17.        </transfer>
18.     </fileStageIn>
19. </job>
```

In the above example, the WS-Addressing Endpoint Reference, or EPR, and the local scheduler are displayed (or “cat”) (line 2 to 11). The EPR specifies the service container (GRAM server) where the job will be executed (line 6). The local scheduler employed is Condor (line 9) (Litzkow, Livny & Mutka, 1988). Condor should be installed and configured on the Service Participant’s workstations before GRAM is deployed and operated. Condor is an open-source GRAM-enabled scheduler that specialises in the management of pools of desktop workstations (Litzkow & Livny, 1990). It is ideal in the author’s proposed solution as most of the service providers will be providing desktop workstations for computation and data-storage needs.

The job that is being staged in (line 12 to 19) specifies the sourceUrl (line 15) and the destinationUrl (line 16). The GRAM service utilizes the Reliable File Transfer service to fetch the files using GridFTP from the Grid Portal (line 15). The files are then cached into the CSS, utilizing the caching service instantiated by the service container ((8) in Figure 6.8).

When the transfer completes, the service container will check if the files have been transferred successfully, and if the integrity of the messages has been preserved by decrypting the messages and comparing hash values (as explained in section 6.2.3 SOAP Security Model). The container will then send a SOAP-Response to the DPHS on whether there are files that should be resent, or if the transfer has been successful. If the transfer was successful, the container will terminate the job and notify the JMS.

After a successful staging of files into the CSS, the JMS will set up the EMS services for the deserialization of the MTOM Attachments into XML Document Object Model (DOM) documents (as explained in Section 6.4.1). As stated by Le Hégaret, Whitmer & Wood (1997), a DOM is "a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page". DOM presents XML documents as a tree-structure. Thus, all the elements, their attributes, and their text are nodes in a DOM tree. The nodes' contents can updated, changed or deleted, and new nodes can be added to the tree (W3Schools, 1999).

DOM defines a standard set of objects utilized to represent XML and HTML documents, a standard model detailing how to combine these objects, and a standard interface specifying how they can be accessed and manipulated (Wood, 1998). There are various levels of DOM designs (Cover, 1999). Level 1 addresses the Core, XML and HTML document models, and the manipulation and navigation of the documents. Level 2 contains a style sheet object model, and specifies how style information attached to a document can be manipulated. It also allows for document traversals, specifies an event model and offers support for XML namespaces (Stenback, Le Hégaret & Le Hors, 2002). Of importance to the proposed framework is Level 3, which deals with loading and saving of documents, and also content models (for example, Document Type Definitions (DTDs) and schemas) with document validation support (Whitmer, 2004). Moreover, it deals with views and formatting of documents, key events and event groups (Stenback & Heninger, 2004; Chang, Kesselman & Rahman, 2004; Le Hors *et al.*, 2004).

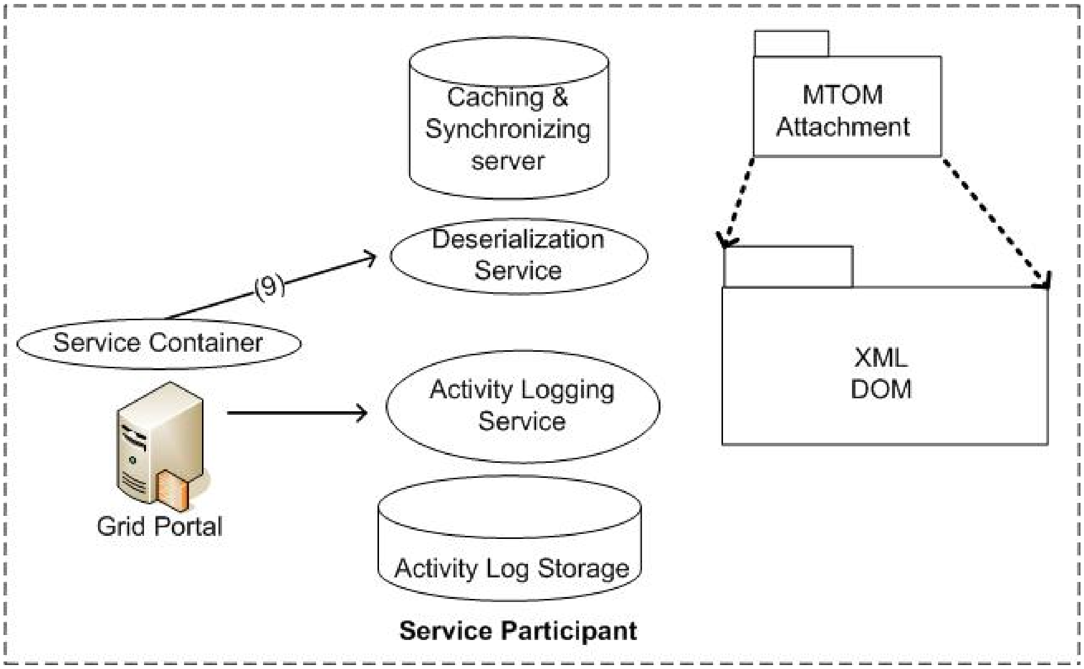

**Figure 6.9 Transforming MTOM Attachments into XML DOM**

The service container will then instantiate the deserialization service that will be utilized to deserialize the files ((9) in Figure 6.9). Following a successful deserialization of MTOM Attachments into XML DOM documents, the container will terminate the job and notify the JMS. The JMS will then set up the EMS services for the transformation of the XML DOM documents (as explained in Section 6.4.1).

The container will instantiate the Data Transformation Service that will be employed to transform the XML DOM documents ((10) in Figure 6.10). The transformations will be accomplished through performing operations, such as summarizing multiple rows of data, aggregating specific columns of data, selecting particular rows of data, deriving values from certain columns (e.g., calculating Age using Date of Birth), etc. Some of the transformed data will be used to feed area specific data marts, after which all the transformed dataset will be transferred into the data warehouse. In order to perform the data transfers, the service container will generate a Reliable File Transfer Service that will be utilized to stage the transformed datasets into the data marts and data warehouse (11).

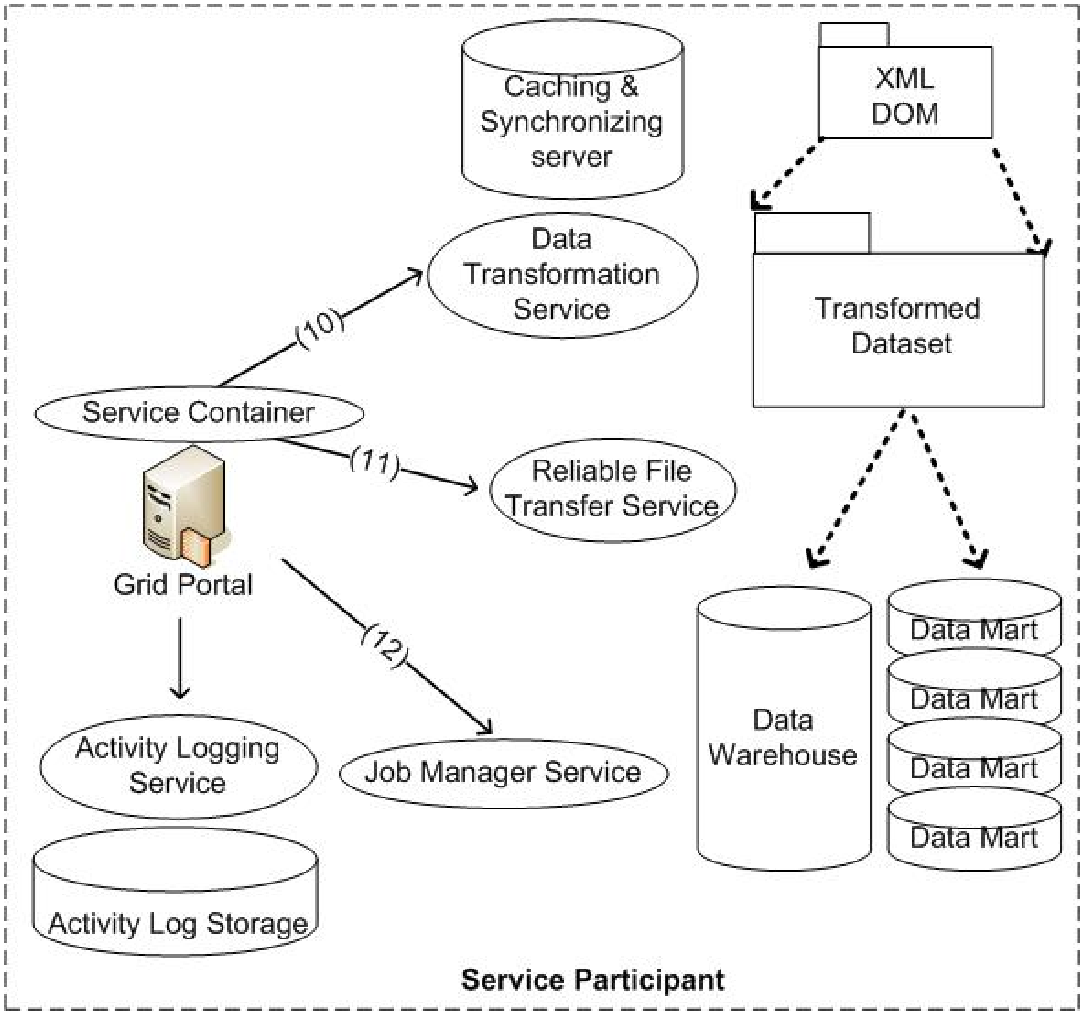

**Figure 6.10: Data Transfer from the CSS into the Data Warehouse**

After a successful staging of files into the data marts and data warehouse, the container will terminate the job and notify the JMS. The JMS will then terminate itself and its actions logged by the activity logging service.