

# Thèse

présentée pour obtenir le grade de docteur

de l'Université Paris XIII

Spécialité : Réseaux et Technologies de l'Information

**Netzahualcoyotl ORNELAS GARCIA**

Modèle Informationnel pour une session User-Centric :  
Inférence et Intégration Intelligente

Soutenue le 15 décembre 2011 devant le jury composé de

Véronique VEQUE	Président
Michelle SIBILLA	Rapporteurs
Jean-Pierre CLAUDE	Rapporteurs
Guy PUJOLLE	Examineurs
Haim DAYAN	Examineurs
Gladys DIAZ	Examineurs
Philippe COUDE	Examineurs
Ken Chen	Directeur de thèse
Noémie Simoni	Co-Directrice de thèse



A me parents,

Inés María García Beltrán et Victor Manuel Ornelas López

*« Knowledge is experience. Everything else is just information » - Albert Einstein*

## Résumé

L'utilisateur d'aujourd'hui est un utilisateur nomade, et ne désire pas avoir des barrières techniques lors de l'accès à un service. Il souhaite avoir toujours le meilleur choix pour accéder au maximum de ses services de façon personnalisée, et aux services offerts par les fournisseurs de services d'après sa localisation ou activité. Cette vision que nous dénommons « User-Centric » cherche des solutions qui soient transparentes (pas d'interruption de la session utilisateur) pour l'utilisateur dans un contexte où l'hétérogénéité (au niveau terminaux, réseaux, et services) et la mobilité (aux niveaux terminaux, réseaux, services, utilisateur et session) sont omniprésentes.

Pour répondre à ces besoins, l'axe couvert par cette thèse est celui de la dimension informationnelle. Nous nous intéressons au système d'information du fournisseur de service (Infoware) mais aussi à celui de l'utilisateur (Infosphère) puisque nous sommes dans un contexte « User-Centric » et transorganisationnel. Le Système d'Information est le référentiel de toutes les activités, il est l'image du monde réel. Devant la complexité, l'hétérogénéité et la dynamique de ce monde réel, une modélisation s'impose. Nous basons nos contributions sur des résultats de travaux précédents qui ont abouti à un modèle récursif « Noeud, Lien, Réseau » s'abstrayant sur 4 niveaux de visibilité qui sont « équipement », « réseau », « service » et « utilisateur ».

Les problématiques auxquelles cette thèse répond sont les suivantes : « Est-ce que les architectures informationnelles proposées aujourd'hui par IMS<sup>1</sup> (HSS<sup>2</sup>) sont suffisantes pour répondre à cette approche « User-Centric ? (Verrou 1) ». « Quelles sont les informations pertinentes pour représenter la session mobile de notre utilisateur nomade ? (Verrou 2) », « Comment prendre en compte la dynamique du monde réel ? (Verrou 3) »,

Le premier point concerne une étude de l'existant et des solutions préconisées par IMS. Nous avons constaté que le HSS (base de données de l'opérateur) permet de réaliser la gestion des informations de l'utilisateur (SLA<sup>3</sup>), ainsi que les services auxquels l'utilisateur est abonné. Dans ce HSS, il n'y a pas les préférences de l'utilisateur. Pas d'information sur la nature ou moyen de communication. Pas d'informations liées à la mobilité (présence de terminaux et

---

1 IP Multimedia Subsystem

2 Home Subscriber Server

3 Service Level Agreement

de services en fonction de la localisation de l'utilisateur). C'est pour cela que, nous avons considéré que les informations stockées dans le HSS d'aujourd'hui, ne sont pas suffisantes pour prendre en compte le contexte dans lequel on se situe.

Pour le second verrou nous avons proposé une base de connaissance appelée « VPIN<sup>4</sup> », qui permet de représenter cet environnement mobile et hétérogène de l'utilisateur. Cette base de connaissance permet d'assurer une session dynamique de l'utilisateur. Ceci est possible grâce au fait que nous avons identifié les informations permettant la prise de décision au bon moment et au bon endroit. Ces informations sont décisionnelles grâce à une sémantique bien définie et à des informations de QoS qui représentent l'aspect comportemental des composants.

Pour le troisième verrou, il s'agit de gérer l'impact sur la base de connaissance des différents événements qui sont induits soit par la mobilité (utilisateur, équipement, réseau, service), soit par suite de changement, de comportement (QoS<sup>5</sup>) d'un composant ou de préférences de l'utilisateur. L'image de cet événement est prise en compte par notre base de connaissance qui par inférence va propager les modifications qui s'imposent grâce à la structuration du modèle

Finalement, nous montrons la faisabilité de nos propositions avec le démonstrateur « UBIS<sup>6</sup> » qui représente une solution des scénarii NGN/NGS. L'architecture NGN représentée est celle d'IMS sur laquelle nous avons fait évoluer le HSS. Nous avons utilisé une base de données Oracle pour implémenter notre base de connaissance, et des routeurs virtuels pour faire la partie réseau d'accès.

---

4 Virtual Private Information Network

<sup>5</sup> Quality of Service

<sup>6</sup> Le projet UBIS a pour objectif de faciliter l'intégration des usages dans le contexte de mobilité et d'ubiquité tel qu'il existe aujourd'hui et tel qu'il se développera demain dans les réseaux du futur. Des limitations importantes inhérentes aux architectures des réseaux actuels imposent de *penser autrement* la mise à disposition des contenus. En effet, l'approche « User-Centric » impacte la continuité de service, la personnalisation, la QoS de bout en bout, et demande l'omniprésence des services sans oublier le transorganisationnel.

---

# Table des matières

Résumé .....	3
Table des matières.....	5
Table des figures.....	7
<b>I. Introduction générale .....</b>	<b>11</b>
<b>I.1 Contexte .....</b>	<b>12</b>
<b>I.2 Motivation.....</b>	<b>14</b>
<b>I.3 Contributions de la thèse.....</b>	<b>15</b>
<b>I.4 Organisation de la thèse .....</b>	<b>16</b>
<b>II. Le contexte .....</b>	<b>18</b>
<b>II.1 Système d'Information.....</b>	<b>19</b>
<b>II.2 Session «user-centric» .....</b>	<b>21</b>
<b>II.2.1 Gestion de l'information de la session .....</b>	<b>22</b>
<b>II.2.2 Mobilité de la session .....</b>	<b>23</b>
<b>III. Etat de l'art .....</b>	<b>24</b>
<b>III.1 Propositions existantes au niveau informationnelles du plan de gestion....</b>	<b>26</b>
<b>III.1.1 Le CIM (Common Information Model).....</b>	<b>26</b>
<b>III.1.2 Le DEN (Directory Enable Network).....</b>	<b>29</b>
<b>III.1.3 Le GNIM (Generic Network Information Model).....</b>	<b>32</b>
<b>III.1.4 Le SID (Shared Information Model).....</b>	<b>34</b>
<b>III.1.5 Travaux sur la gestion de l'information à partir des modèles.....</b>	<b>37</b>
• <b>Towards a Context-Aware Information Model for Provisioning and Managing Virtual Resources and Services .....</b>	<b>37</b>
• <b>Ontology-based Capability Self-Configuration .....</b>	<b>38</b>
• <b>The Applicability of Self-Awareness for Network Management Operations .</b>	<b>39</b>
<b>III.2 Gestion de l'information des architectures NGN : IMS (IP Multimedia Subsystem).....</b>	<b>40</b>
<b>III.2.1 Architecture NGN : «IMS».....</b>	<b>40</b>
<b>III.2.2 Protocoles associés à l'architecture IMS .....</b>	<b>43</b>
<b>III.2.3 Architecture informationnelle d'IMS (HSS).....</b>	<b>47</b>
<i>Manques du HSS de Fokus .....</i>	<i>49</i>
<b>III.3 De la gestion de l'information à la gestion de la connaissance (Knowledge Management).....</b>	<b>53</b>
<b>III.3.1 Passage des données à la connaissance .....</b>	<b>54</b>
<b>III.3.2 Le passage vers un système de gestion de connaissance (SGC).....</b>	<b>57</b>
<b>III.4 Gestion des événements du système .....</b>	<b>58</b>
<b>III.4.1 Architecture EDA (Event Driven Architecture).....</b>	<b>59</b>
<b>III.5 Sélection de piste de travail.....</b>	<b>61</b>
<b>III.5.1 Meta-modèle NLR : Nœud, Lien, Réseau .....</b>	<b>61</b>
<b>III.5.2 Niveaux de visibilité .....</b>	<b>62</b>
<b>III.5.3 Modèle de QoS .....</b>	<b>64</b>
• <b>Critères du modèle de QoS .....</b>	<b>65</b>
• <b>Valeurs du modèle de QoS.....</b>	<b>65</b>
<b>III.6 Conclusion .....</b>	<b>66</b>
<b>IV. Des besoins NGN aux propositions.....</b>	<b>69</b>
<b>IV.1 Besoins NGN sous forme de scénario (mobilité et composants de services)71</b>	
<b>IV.2 Le diagramme du scénario NGN.....</b>	<b>74</b>
<b>IV.2.1 Les événements du réel.....</b>	<b>75</b>

---

IV.3	Gestion des evenements et inférence du réel .....	76
IV.3.1	Changement de QoS d'une ressource.....	78
IV.3.2	Changement d'état d'une ressource.....	79
IV.4	Modèle Informationnel : Real Time Profile .....	80
IV.4.1	Real Time Profile d'une ressource .....	81
IV.4.2	Real Time Profile d'un réseau de ressource .....	84
IV.4.3	Real Time Profile du système.....	86
IV.5	Le VPIN (Virtual Private Information Network) .....	89
IV.6	Conclusion.....	94
V.	Plate-forme de validation de concepts .....	96
V.1	Plate-forme du scénario final.....	97
V.1.1	Architecture du scénario .....	97
V.2	Plate-forme IMS Fokus .....	98
V.3	Besoins au niveau de la base de données.....	100
V.4	Implémentation de la base de connaissance .....	102
V.5	Fonctionnalités dans la base informationnelle .....	105
V.6	Conclusions.....	113
VI.	Conclusion générale & perspective .....	115
VI.1	Conclusion générale .....	116
VI.2	Perspectives.....	118
	Références Bibliographiques .....	119
	Liste des publications .....	121
	Mise en place de la plateforme OS-IMS .....	123
	Mise en place d'Oracle 11g .....	134

## Table des figures

Figure I.1 Contexte NGN de l'utilisateur .....	12
Figure II.1 Approche User-Centric .....	22
Fig. III.1 CIM Schema .....	28
Figure III.2 Classe « Managed élément » décrite en UML .....	28
Figure III.3 Structure de DEN .....	30
Figure III.4 Hierarchie GNIM .....	34
Figure III.5 Méthodologie NGOSS .....	35
Figure III.6 Domaines SID .....	36
Figure III.7 Prov. d'un SLA personnalisé avec des ressources virtuelles.....	38
Figure III.8 Structure FOCALE Lexicon .....	39
Figure III.9 Release 8 Architecture IMS .....	41
Figure III.10 Structure de Diameter .....	44
Figure III.11 Flags de Diameter .....	44
Figure III.12 - Les codes des AVPs .....	46
Figure III.13 Entête des AVPs .....	46
Figure III.14 HSS d'IMS de Fokus .....	47
Figure III.15 HSS de Fokus actuel .....	51
Figure III.16 Exemple du passage de données à la connaissance .....	56
Figure III.17 Architecture d'un Systèmes de Gestion de Connaissance (SGC)..	58
Figure III.18 Approche par Event Driven Architecture .....	60
Figure III.19 Approche par Event Driven Architecture, CEP .....	61
Figure III.20 Méta-modèle NLR .....	62
Figure III.21 Niveaux de visibilité .....	63
Figure III.22 Modèle architecturale .....	64
Figure IV.1 Scénario qui représente 2 types de mobilités .....	71
Figure IV.2 Session de Bob .....	75
Figure IV.3 Première session de Bob .....	76
Figure IV.4 Changement de QoS détecté par l'agent EMA .....	78
Figure IV.5 Changement d'état détecté par l'agent EMA .....	79
Figure IV.6 Real Time Profile ressource .....	81
Figure IV.7 Real Time Profile réseau .....	86
Figure IV.8 Real Time Profile du système complet .....	87
Figure IV.9 Ressource PC .....	88
Figure IV.10 Ressource PDA .....	88
Figure IV.11 Ressource 3G .....	89
Figure IV.12 Ressource Wi-Fi .....	89
Figure IV.13 Structure du VPIN .....	92
Figure V.1 NGN/NGS Middleware .....	98
Figure V.2 IMS vs IMS de Fokus .....	99
Figure V.3 FHoSS .....	101
Figure V.4 Structure de la base de données FHoSS (version FOKUS) .....	102
Figure V.5 Nouvelle Structure de la base de données .....	103
Figure V.6 HSS + .....	104

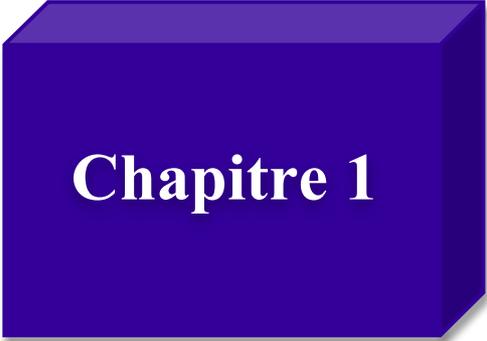
## Liste des Abréviations

3GPP	The 3rd Generation Partnership Project
AAA	Authentication, Authorization, and Accounting
ABS	Activity Based Service
ACK	Acknowledgement
ACS	Ambiant Control Space
AE	Autonome Element
ANG	Access Network Gateway
ANR	Access Network Router
AP	Access Point
API	Application programming interface
AR	Access Router
AS	Access Service
ASG	Access Service Gateway
ASP	Application Service Provider
ASR	Access Service Router
AU	Access User
BBM	Break-before-make
BCPP	Bounded Composition Probing Protocol
BDC	Base de Connaissance
BDD	Base de Données
BTS	Base Transceiver Station
CH	Communication Host
CN	Core Network
CoI	Community of Interest
CSCF	Call Session Control Function
CV	Communauté Virtuelle
CVE	Communauté Virtuelle des Equipements
CVR	Communauté Virtuelle de Réseaux
CVS	Communautés Virtuelles de Services
EC	Equipement Communication
EDA	Event Driven Architecture
ETSI	European Telecommunications Standards Institute
FA	Foreign Agent
FCAPS	Fault, Configuration, Accounting, Performance, Security
FES	Fiorano Enterprise Server
GPRS	General Packet Radio Service
GS	Global Service
GSM	Global System for Mobile Telecommunication
HA	Home Agent
HHO	Horizontal HandOver
HLR	Home Location Register
HSS	Home Subscriber Server)
I-CSCF	Interrogating CSCF
IETF	Internet Engineering Task Force
IFC	Initial Filter Criteria
IMS	IP Multimedia Subsystem

---

IM-SSF IP Multimedia Service Switching Function  
INAP Intelligent Network Application Part  
ISUP ISDN User Part  
ITU-T International Telecommunication Union - Telecommunication  
standardization sector  
JMS Java Message Service  
KM Knowledge Management  
KMS Knowledge Management System  
LLC Logical Link Control  
MBB Make-before-break  
MGW Media gateways  
MI Modèle Informationnel  
MIH Media Independent Handover  
MIPv4 Mobile IP v4  
MIPv6 Mobile IP v6  
MN Mobile Node  
MRF Multimedia Resource Function  
MSC Mobile Switching Center  
NE Network Element  
NGN Next Generation Network  
NGS Next Generation Service  
NGSS Next Generation Service Session  
OS Operating System  
OSI Open Systems Interconnection  
P2P Peer-to-peer  
PAN Personal Access Network  
PCSCF Proxy CSCF  
PnP Plug and Play  
PS Packet Switch  
PUI Public User Identity  
QoS Quality of Service  
RNC Radio Network Controller  
ROI Return On Investigation  
SCA Service Component for Application  
SCFs Service Capability Features  
SCN Service Component for Network  
SCS Service Capability Server  
S-CSCF Serving-CSCF  
SCU Service Component for User  
SE Service Element  
SGSN Serving GPRS Support Node  
SGW Security Gateway  
SI System Information  
SIP Session Initiation Protocol  
SLA Service Level Agreement  
SLBS Service Location Based Service  
SOA Service Oriented Architecture  
SPBS Service Presence Based Service

SPN Service Provider Network  
SSON Specific Service Overlay Network  
TINA Telecommunications Information Networking Architecture  
TISPAN Telecommunications and Internet converged Services and  
Protocols for Advanced Networking  
TMN Telecommunications Management Network  
TSAS Telecom Service Access and Subscription  
TTL Time to live  
TTM Time To Market  
UABS Activity Based Service  
UC User-Centric  
UDBS User Discovery Based Service  
UGA User Generated Application  
UGC User Generated Content  
ULBS User Location Based Service  
UMA Unlicensed Access mobile  
UMAS User Manager Agent System  
UMTS Universal Mobile Telecommunications System  
UPBS User Presence Based Service  
UICC UMTS Integrated Circuit Card  
VDC Virtuelle Device Community  
VHO Vertical HandOver  
VLR Visiter Location Register  
VM Virtual Machine  
VPSN Virtual Private Service Network  
VPUN Virtual Private User Network  
VPxN Virtual Private User/Service/Network/Equipment Network  
VSC Virtuelle Service Community  
VUC Virtuelle User Community  
W3C World Wide Web Consortium  
WDM Wavelength-Division Multiplexing



## Chapitre 1

# Introduction générale

De nos jours, les services offerts et les réseaux de communication prennent une place importante dans la société d'information. Les acteurs de télécommunication doivent relever des défis majeurs qui conditionnent leur maintien sur des marchés très concurrentiels et volatils. En effet, avec l'évolution de la taille et la complexité de l'industrie des médias et des télécommunications, les offres de services sont de plus en plus variées et complexes. Les exigences en matière d'introduction rapide des nouveaux services, de garantie de leur QoS conforme aux contrats, de la maîtrise des nouvelles technologies de communications et de réduction des coûts associés, sont parmi les facteurs les plus déterminants du succès des opérateurs de télécommunication et fournisseurs de services sur les marchés d'aujourd'hui.

Cette introduction générale présente le contexte dans lequel nous focalisons nos travaux de recherche (§I.1). Nous montrons la motivation qui nous a mené à travailler sur ce champ d'investigation (§I.2). L'aboutissement de ces travaux, à savoir les contributions de cette thèse sont mentionnées (§1.3) ainsi que l'organisation de ce manuscrit (§I.4).

### I.1 Contexte

L'avancée des technologies des télécommunications a considérablement augmenté ainsi que les moyens de communication et la diversité des contenus. De nos jours, l'utilisateur est de plus en plus nomade, il souhaite ne pas avoir de barrières techniques lors de l'accès à un service. Il souhaite accéder au maximum de services de façon personnalisée, et aux services offerts par les fournisseurs de services selon sa localisation ou son activité (Figure I.1).

Instant T	À 10:00 hrs. → À 19:00 hrs.			
<b>Temps</b>				
<b>Localisation (Spatiale)</b>	 Bureau		 Maison	
<b>Terminaux</b>	 Vidéoconférence	 IP		 Vidéoconférence
<b>Réseau</b>	 - Wi-fi - Wimax - Ethernet - PSTN		 - Wi-fi - ADSL	
<b>Services</b>	Propres à l'utilisateur	Offertes par le réseau	Propres à l'utilisateur	Offertes par le réseau
	- Mail - Téléphone	- Vidéoconférence	- Mail	- VoD
<b>User</b>	 Jean Pierre		 Jean Pierre	

Figure I.1 : Contexte NGN de l'utilisateur

---

Nous devons considérer, l'évolution de ce contexte des technologies de l'information et communication (TIC). Tout a commencé par la fourniture des systèmes installés dans des équipements (PCs) pour que l'utilisateur accède à son service. Cette approche a été définie comme « System-Centric ». Mais avec l'évolution des nouveaux besoins de l'utilisateur, il nous fallait partager des ressources distribuées et distantes accessibles à travers des réseaux différents, c'est là que l'approche « Network-Centric » a fait son apparition. Cela veut dire que l'utilisateur doit avoir la connaissance de son environnement pour accéder à son service. Alors comment faire quand le paysage se complexifie avec la nomadicité des utilisateurs ? En effet, il lui faut, non seulement accéder à son service mais avoir la continuité de sa session lors de ses déplacements. Nous passons alors, à l'approche « User-Centric ». Cette vision sous entend que l'utilisateur crée son propre environnement pour accéder à son service mais sans se soucier de son environnement, ceci en ayant des solutions transparentes (pas d'interruption de sa session) lors de sa mobilité. Cette mobilité, que ce soit celle de l'utilisateur, ou celle du terminal ou celle du réseau ou encore celle du service, peut se produire dans un contexte technologique hétérogène à tous les niveaux. Pour l'utilisateur, ce contexte hétérogène ne doit pas l'empêcher d'accéder à son service, tout au contraire il faut que ses préférences soient prises en compte.

En effet, l'utilisateur peut avoir plusieurs équipements de communication (EC) : ordinateur, PDA, téléphone mobile etc. Ces équipements lui permettent d'atteindre le même service mais à des coûts et des qualités différentes. Les équipements de communication étant multi-technologies en termes de télécom (WiFi, WiMax, Bluetooth, GSM, 3G, etc.), ils sont susceptibles de se connecter simultanément à plusieurs réseaux d'accès parmi les réseaux disponibles ou de commuter de l'un à l'autre. Le choix se fera selon son environnement (QoS « Quality Of Service », disponibilité, etc.), mais aussi selon ses préférences (coût, facilités, etc.). Si nous poursuivons le raisonnement au niveau service, on peut supposer que demain, l'utilisateur aura le choix de son fournisseur de service et qu'il puisse combiner dans une même session des composants de services offerts par des organisations différentes (que nous qualifions comme vision transorganisationnelle).

Ces nouveaux besoins introduisent une nouvelle perception du paysage des télécommunications. L'industrie de télécommunications doit proposer des nouvelles solutions, afin de répondre à ces besoins de : « anywhere, anytime, anyhow, anyservice », en assurant la continuité de service sans coupure et sans couture. Ceci est un réel défi inhérent au contexte de Next Generation Network/Next Generation Service (NGN/NGS).

Dans ce défi, le système d'information est une dimension maîtresse. En effet, c'est à travers le SI que nous pouvons avoir un référentiel de toutes les activités qui représente l'image du monde réel hétérogène et dynamique. Si l'on se positionne

du côté opérateur, c'est à travers le HSS (Home Subscriber Server) que l'on aura le système d'information de l'opérateur avec une architecture IMS (IP Multimedia Subsystem). Pourtant, nous avons constaté que le HSS d'aujourd'hui ne comporte pas toutes les informations nécessaires. On n'a pas les informations concernant les préférences de l'utilisateur. En ce qui concerne les informations au niveau terminal, aucune information n'est stockée, ce qui ne permet pas de faire la gestion des terminaux qui pourraient être disponibles pour l'utilisateur. Ensuite, au niveau réseau, lorsque l'utilisateur se déplace, on trouve seulement le nom du réseau par lequel on accède (roaming), cela peut empêcher que l'on soit transorganisationnel, car on doit pouvoir traverser tout type de réseau même si ce réseau n'appartient pas au même opérateur. Au niveau service, on a le profil de service de l'utilisateur, ce qui permet de savoir à quel service il est abonné, par contre, on ne connaît pas la QoS du composant auquel on accède. Cela veut dire que, si un composant de service tombe en panne, il y aura une coupure de la session utilisateur. Les informations stockées sont donc insuffisantes pour répondre aux nouveaux besoins.

## ***1.2 Motivation***

Comme nous l'avons décrit dans le contexte (§1.1), nous savons que l'utilisateur doit maîtriser la technologie afin d'accéder à un service. Cette évolution technologique peut-être faite au niveau terminal, réseau ou service. Sachant que l'utilisateur est de plus en plus nomade, nous devons donc casser ces barrières techniques qui empêchent l'utilisateur d'accéder librement à ses services. En prenant comme point de départ l'approche « user-centric », nous avons concentré nos efforts sur la dimension informationnelle.

C'est à travers l'information que l'on peut avoir la connaissance de l'environnement de l'utilisateur et les prises de décisions sont possibles, si nous avons la bonne information au bon moment et au bon endroit. La connaissance de l'environnement de l'utilisateur suppose la connaissance non seulement des réseaux d'accès et services de proximité, mais aussi des terminaux et des utilisateurs des différentes communautés auxquelles il appartient ou désire appartenir.

Nous nous sommes efforcés de proposer une architecture informationnelle qui permette à l'utilisateur d'avoir la connaissance de ces moyens de communications comme les terminaux, réseaux à travers lesquels il peut accéder à son service. Nous devons faire en sorte que les équipements puissent s'autogérer pour avoir une dynamique lors du changement de comportement d'un composant.

Par conséquent, une des solutions viables est de rendre les bases de données qui existent chez les opérateurs et fournisseurs de services en base de connaissances.

---

Les bases de données comme on le sait, fournissent les informations d'après des applications, Elles sont dépendantes du groupe d'applications qu'elles servent. Elles stockent les données qui sont utilisées par l'utilisateur. Par contre, la majorité, des bases de données n'ont pas l'intelligence pour tenir compte des informations en temps réel du système d'information qui peut évoluer dans le temps. A la différence d'une base de connaissance, qui prend en compte l'environnement hétérogène de l'utilisateur, ainsi que les possibles événements qui peuvent se produire dûs à la mobilité (utilisateur, terminal, réseaux, service) ou à la hétérogénéité (terminal, réseau, service), c'est-à-dire le monde réel. En effet, comme on se retrouve dans un environnement dynamique, il faut une adaptation dynamique à tous les niveaux afin d'avoir une continuité de la session de l'utilisateur.

Pour ce faire, on doit avoir une information, qui nous donne le comportement des composants qui sont à la disposition de l'utilisateur, qui nous donne une image du réel. Afin de représenter ce monde réel, une modélisation homogène s'impose. Le modèle informationnel sera donc un élément important dans le système d'information que ce soit du côté opérateur ou du côté utilisateur. La modélisation doit être uniforme car le monde réel est très hétérogène. Si chaque composant (que ce soit un terminal, un réseau ou un service) suit sa propre modélisation, les problèmes d'incohérence et de compatibilité vont apparaître. Nous devons donc faire en sorte qu'au niveau gestion, on puisse avoir la même représentation des entités du monde réel. Ceci permettra de faciliter l'interopérabilité entre composants, ce qui aidera ainsi une communication de bout-en-bout à tous les niveaux.

Nous devons considérer lors de la gestion de ce monde réel, d'avoir toujours une cohérence informationnelle, c'est-à-dire, ne pas avoir une duplication d'information ou une information différente, ce qui peut induire à une mauvaise représentation de ce monde réel.

L'autogestion des composants se veut une solution viable afin de savoir lorsqu'on demande un service, si le composant peut ou pas rendre le service demandé. Ceci est possible en tenant compte de la QoS du composant (que ce soit terminal, réseau, service). Ceci permettra d'avoir une dynamique de la session utilisateur sans présenter des coupures ni coutures tout au long de la session. Ce principe, constitue le travail de recherche essentiel de cette thèse qui sera développé dans les chapitres suivants.

### ***1.3 Contributions de la thèse***

Cette thèse vise à proposer des solutions pour la gestion de la session de l'utilisateur. Afin de mieux représenter la problématique traitée, nous avons

représenté les besoins NGN à l'aide d'un scénario. Les principales contributions de cette thèse sont :

- **Une Base de connaissances inférente**

Nous proposons d'avoir une base de connaissance indépendante des applications. Pour ce faire, elle doit représenter le monde réel et suivre les événements pertinents (au sens du contexte NGN) de ce monde réel. Ces événements peuvent être produits par la mobilité (utilisateur, réseau, service, utilisateur, session) ou par le changement de comportement d'un composant (QoS) ou par le changement d'activité d'un utilisateur. Des agents de QoS, sont intégrés aux ressources. Ils jouent un rôle important et permettent de gérer les changements de comportement de chaque ressource.. Si un composant n'accomplit pas la QoS demandée, il pourra être remplacé par un autre composant qui satisfait la QoS demandée. Le fait de pouvoir remplacer un composant permet d'avoir une continuité de session (MBB: Make Before Break).

- **Une Structure Informationnelle « Real Time Profile »**

Nous avons proposé une structure informationnelle qui se base sur une architecture de service (SOA) à quatre niveaux de visibilité. Le modèle de service et de QoS s'applique à l'ensemble des ressources que nous devons gérer, c'est-à-dire : au niveau terminal, réseau ou service. Cette structure permet lors de l'exploitation de service, d'avoir une image qui correspond au profil du système réel. Ce "Real Time Profile" est construit à partir des profils de ressources. et du profil des utilisateurs.

- **Un modèle organisationnel basé sur une partition informationnelle pour gérer la session de l'utilisateur (VPIN)**

Cette proposition permet d'avoir la vue globale du système d'information de chaque utilisateur. La session de l'utilisateur aura l'information adéquate à tous les niveaux de visibilité (que ce soit utilisateur, terminal, réseau, service) pour assurer une continuité sans coupure ni couture. Les règles et automatismes introduits décrivent le modèle organisationnel (le qui fait quoi).

## ***1.4 Organisation de la thèse***

Le travail de cette thèse a été divisé sur sept chapitres afin de répondre à la problématique que nous avons traité.

---

Le premier chapitre (§1), concerne une introduction générale qui permet de délimiter le périmètre de travail de recherche de nos travaux. Nous avons présenté nos motivations qui nous ont amené à travailler sur cette problématique. Ensuite, nous avons présenté la contribution de cette thèse ainsi que son organisation.

Le deuxième chapitre (§2), traite le contexte dans lequel se base le travail de cette thèse. Nous nous focalisons plus particulièrement sur le système d'information, que ce soit au niveau opérateur ou utilisateur. Nous regardons également le contexte hétérogène auquel l'utilisateur d'aujourd'hui est confronté lorsqu'il accède à ses services. Dans ce contexte hétérogène, nous regardons de près les conséquences du nouveau paradigme « User-Centric » de ce nouvel environnement.

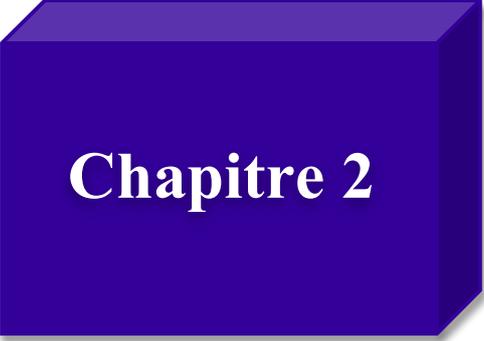
Le troisième chapitre (§3), montre les travaux existants dans notre périmètre de recherche qui est la dimension informationnelle. Nous avons fait une étude autour des organismes de standardisation (DMTF, TMF, 3GPP) pour analyser leurs propositions notamment au niveau informationnel. Ceci nous a permis d'identifier les manques de ces propositions et ainsi, partir avec de bonnes bases pour faire de nouvelles propositions pour répondre aux nouveaux besoins.

Nous présentons également les travaux qui ont été réalisés par Télécom ParisTech sur lesquels nous avons basé nos travaux de recherche. Ces travaux concernent un modèle récursif « N, L, R », un modèle de QoS afin de gérer le comportement des composants. Un tableau de synthèse permet de mettre en évidence les défis à relever.

Le quatrième chapitre (§4) est consacré aux contributions de la thèse. Nous avons proposé "une base de connaissances inférente" avec une "structure informationnelle" pour la gestion de ressources (Real Time Profile) et enfin un modèle organisationnel (VPIN) qui permet de gérer la session de l'utilisateur dans son environnement hétérogène et mobile.

Le cinquième chapitre (§5) décrit le démonstrateur réalisé afin de valider les concepts que nous avons proposés. Ce démonstrateur est basé sur une plate-forme IMS sur laquelle, nous avons travaillé la dimension informationnelle. Un scénario est déroulé pour montrer la mise en place et la faisabilité de nos propositions.

Le sixième chapitre (§6) consigne globalement les conclusions concernant ce travail, ainsi que les perspectives à réaliser par la suite.



## Chapitre 2

# Le contexte

---

Nous sommes aujourd'hui dans une époque où l'utilisateur ne peut pas se passer des nouvelles technologies. L'évolution de ces technologies implique que les systèmes d'informations occupent une place de plus en plus importante dans l'organisation d'une entreprise ou des utilisateurs finaux dans la vie de tous les jours. C'est pour cela que nous devons comprendre ce nouveau contexte où les systèmes d'information (SI) jouent un rôle important (§II.1). Nous présentons comment l'information est indispensable pour gérer les nouveaux paradigmes comme celui d'user-centric et de la session sans coupure et sans couture dans un contexte NGN (§II.2).

## **II.1 Système d'Information**

Dès nos jours, les entreprises cherchent toujours à mieux gérer leurs ressources avec une meilleure efficacité, elles sont donc confrontées au savoir de *comment* gérer leurs ressources avec efficacité.

L'arrivée des nouvelles technologies numérisant les échanges et les contenus a apporté plus de flexibilité mais également un florilège de technologies facilement accessibles et personnelles. L'homme dispose ainsi d'une «bulle de communication», individuelle, personnelle, avec différentes possibilités pour communiquer tels que téléphone fixe, téléphone mobile, messagerie électronique, messagerie instantanée, etc, et s'affranchit de problèmes spatiaux et temporels. Le nomadisme, l'ubiquité sont désormais intégrés dans les usages et les comportements. Ce nouvel environnement doit être pris en compte dans l'architecture des systèmes d'information afin de créer de la valeur ajoutée par rapport aux solutions existantes.

Le système d'Information permet de représenter l'ensemble de tous les éléments qui participent au sein de l'entreprise. Le système d'information contribue donc à la gestion, dissémination, et traitement de l'information au sein de l'entreprise. C'est donc cette information gérée qui permet à l'entreprise d'avoir un référentiel pour atteindre ses objectifs.

Les systèmes d'information peuvent être différents ou similaires à d'autres entreprises, mais la plupart considère avoir les composants suivants : Applications métiers, Base de Données, Logiciels de gestion de la relation client (Customer Relationship Management – CRM), Gestion de la chaîne logistique (Supply Chain Management – SCM), Serveurs applicatifs, ainsi que des dispositifs de sécurité et la MIB (Management Information Base) de l'infrastructure réseau.

L'infrastructure réseau est un élément stratégique pour l'entreprise car, c'est à travers le réseau que les informations seront distribuées à chaque entité qui en aura besoin. Le réseau sert donc comme support des flux générés par l'ensemble des applications et donc il intervient dans toutes les activités.

Nous devons aussi considérer l'évolution de croissance de l'entreprise, là où elle cherche à trouver la rentabilité maximale. En effet, une entreprise ne pourra pas rester sur les marchés d'aujourd'hui si elle n'est pas capable d'avoir les informations pertinentes, fiables, précises, qui permettent de gérer avec efficacité son activité. Par conséquent, une entreprise qui arrive à avoir ces informations, pourra prendre les décisions pertinentes au bon moment et au bon endroit. C'est pour cela que la mise en place d'un système d'information qui tient compte de ces nouveaux besoins est donc essentielle. Cela permettra de fournir à l'entreprise la connaissance de toutes ses ressources.

Cependant, la grande problématique pour les entreprises est de savoir comment rechercher ces informations indispensables pour son bon fonctionnement, car l'abondance informationnelle de l'entreprise peut entraîner des erreurs de cohérence et donc une mauvaise prise de décision de la part des utilisateurs.

Les sources d'information gérées par l'utilisateur, peuvent être de différents types comme : Informations techniques, comptables, commerciales, sociales.

- **Les informations techniques** permettent à l'entreprise de définir son mode de production en tenant compte entre autre des contraintes qui y sont associées (planification des tâches). La maîtrise de ces informations techniques est indispensable pour définir un processus productif le plus efficace possible. De plus, le suivi des innovations techniques permet de modifier le plus rapidement possible l'organisation du système productif de manière à pouvoir conserver un avantage compétitif dans ce domaine.

- **Les informations comptables** permettent tout à la fois d'avoir une idée sur la situation patrimoniale d'une entreprise (ensemble de ses avoirs et de ses dettes), sur sa capacité à dégager un profit (compte de résultat) mais permettent aussi de calculer ses coûts de production (comptabilité analytique) et de fournir une idée de ses marges de manœuvre futures (analyse financière).

- **Les informations commerciales** (provenant du réseau de vente de l'entreprise) permettent à cette dernière de mesurer le niveau d'adéquation entre la production d'une entreprise et la demande qui lui est adressée, mais permettent aussi d'évaluer l'impact d'une politique de prix ou de fournir des informations quant aux évolutions des modes de consommation (évolution de la part de marché).

- **L'information social** est indispensable dans le sens où elle détermine le champ d'action des différents acteurs de l'entreprise soit au travers de la définition claire des fiches de poste (qui fait quoi, sous le contrôle de qui), soit à travers de la connaissance des droits et devoirs relatifs à ces différents acteurs (code du travail).

---

La mise en place formelle ou informelle d'un système d'information dans l'entreprise ou chez l'utilisateur répond à la nécessité de gérer l'information d'une manière efficace.

Le système d'information doit être organisé, finalisé, construit, animé et contrôlé, ce qui constitue un moyen d'optimisation de la performance de l'entreprise et de l'utilisateur.

Les travaux de cette thèse consistent à savoir comment fournir à l'utilisateur les informations dont il a besoin lors de l'accès à son service. Il sera toujours utile d'avoir la connaissance des moyens de communication à travers lesquels il peut accéder à son service.

Donc, nous pouvons en conclure que la donnée contextualisée donne l'information et l'information intégrée par une personne dans son système de valeurs devient une connaissance.

Pour une Base de Connaissance, il faut savoir transférer dans une machine un système de valeurs et de jugements, ainsi que de l'information fortement structurée, fortement contextualisée, et présentée de manière sensible à l'utilisateur du système. De cette façon l'utilisateur peut s'approprier facilement cette information de telle sorte qu'il en fasse *sa propre connaissance*.

Pour avoir sa propre connaissance, il faut établir des règles, c.-à-d. : opter pour des règles d'inférence : Communautés, valeurs de QoS, etc. Nous verrons par la suite les travaux à propos de la gestion de la connaissance appliquée aux systèmes (chapitre III.3).

## **II.2 Session «user-centric»**

Comme nous l'avons décrit dans notre contexte (§I.1), l'approche « user-centric » se veut une solution apte à considérer les nouveaux besoins de l'utilisateur. L'avantage de cette approche est que le système complet est à la disposition des nouveaux besoins de l'utilisateur, contrairement aux approches que nous avons identifiées comme « network-centric » (contrainte par le réseau) ou « system-centric » (contrainte par le système) (Figure II.1).

Pour avoir des réponses, face à la complexité des choix et des configurations possibles, il s'avère important de mettre en place cette approche. Elle donne un fil conducteur et une temporalité pour agréger tous les niveaux, que ce soit le niveau terminal, réseau, ou service, car il faut pouvoir fournir le service à l'utilisateur sans avoir de coupures ni coutures. Les contraintes peuvent surgir suite soit à la mobilité (utilisateur, terminaux, réseaux, services) soit à l'hétérogénéité (terminaux, réseaux, services).

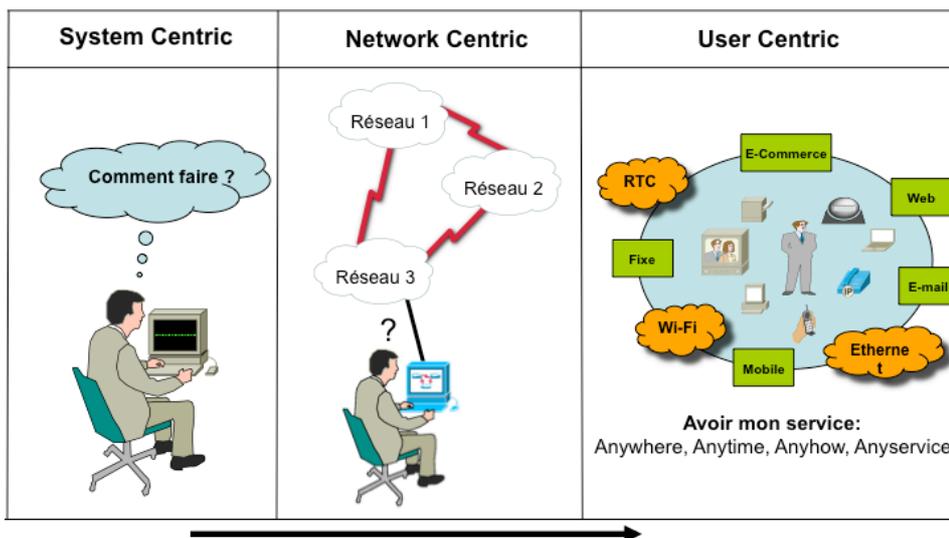


Figure II.1: Approche User-Centric

### II.2.1 Gestion de l'information de la session

Nous devons prendre en considération ce nouvel environnement afin de satisfaire les nouveaux besoins de l'utilisateur. On doit casser toute barrière technique qui empêche l'utilisateur d'accéder à son service. Par exemple, lorsqu'un utilisateur accède à son service de vidéo à la demande auquel il est abonné, il lui faut un terminal (PC, PDA) pour afficher son film, il lui faut une connectivité à travers un réseau donné (Wi-Fi, Ethernet, 3G), ainsi que le serveur de streaming qui contient le film. Imaginons que l'utilisateur se déplace de chez lui au parc, si l'utilisateur utilisait son PC fixe, il lui faut basculer sa session de son PC fixe vers son PDA. Pour cela, il lui faut savoir si le PDA qui remplacera son PC fixe peut supporter le service qu'il est en train d'utiliser. Pour pouvoir basculer sa session, l'utilisateur doit connaître toutes les informations de QoS du PDA en question, notamment s'il a la batterie nécessaire, une bonne résolution au niveau de l'écran, carte réseau (Wi-Fi, 3G). Toutes ces informations permettront à l'utilisateur de prendre la bonne décision par rapport à son environnement.

Pour pouvoir regrouper toutes ces informations, nous avons recours à la modélisation pour répondre à cette problématique d'ordre multidimensionnelle. La modélisation permettra de suivre tout changement de situation relatif à la session de l'utilisateur. Cela veut dire, qu'à travers le modèle informationnel, on aura toutes les informations relatives aux terminaux, réseaux et services de l'utilisateur. La gestion de chaque composant ne pourra se faire que si on possède toutes les informations nécessaires et décisionnelles.

Le modèle informationnel représente donc l'image du système d'information du contexte que nous voulons gérer.

---

Dans notre contexte, nous modélisons la session de l'utilisateur en tenant compte de son environnement, de sa localisation et de son activité. Ainsi, on pourra assurer une continuité de son service durant toute sa session.

### ***II.2.2 Mobilité de la session***

Dans le domaine des télécommunications, la notion de « session » est un espace temps, dans lequel se déroule une succession d'interactions entre les deux extrémités du réseau de transport. Une session est définie dans le modèle référentiel OSI (Open Systems Interconnection) comme un ensemble d'échanges point à point avec un interlocuteur engagé dans la mise en relation. C'est lorsque les éléments qui constituent la session sont en mouvement qu'il existe un risque d'interruption de la session. Ceci est dû à la mobilité des éléments comme le terminal, le réseau d'accès ou le service.

La mobilité du terminal référence un terminal mobile qui se déplace à travers des zones couvertes par des points d'accès tout en ayant accès à toute la liste des services auxquels il a souscrit. Cette mobilité repose principalement sur le handover vertical et horizontal.

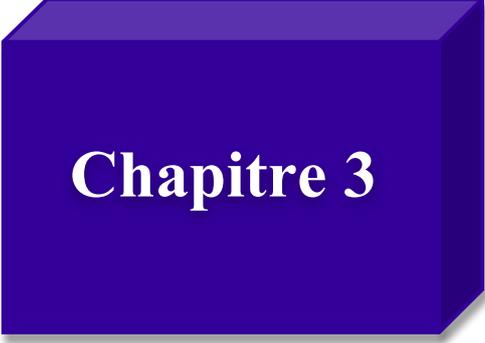
La mobilité de réseau d'accès référence un réseau d'accès qui se déplace tout en mettant à jour sa table de routage. Les points d'accès peuvent se déplacer et s'organiser arbitrairement et ceci malgré le changement rapide et imprévisible de la topologie du réseau. Un tel réseau peut opérer d'une manière autonome ou bien être connecté au grand réseau de l'internet.

La mobilité de services référence des services souscrits et personnalisés qui sont délivrés même si l'utilisateur est connecté à un autre réseau de fournisseur de services. La mobilité de service exige que l'accès au service soit garanti n'importe où et n'importe quand avec la qualité de service prévue.

La mobilité de service est définie comme étant la méthode pour réaliser une organisation personnalisée de service qui mette à jour dynamiquement les services dépendamment des mobilités de l'utilisateur ou du terminal à travers les réseaux hétérogènes. Cette mobilité est indispensable pour assurer la continuité de services.

Mais sa mise en œuvre passe en fait par la mobilité de la session. Cette mobilité est la seule qui intervient dans tous les types de mobilités (mobilité de l'utilisateur, terminal, réseau, service) car elle couvre l'aspect temporel. Elle est prise en compte à tout niveau de visibilité, car elle représente le service rendu à l'utilisateur (binding) crosslayer.

C'est la gestion de cette mobilité de session qui nous permettra de maintenir la session sans coupure ni couture concernant une demande utilisateur. C'est ainsi que nous couvrirons les besoins de NGN.



## Chapitre 3

# Etat de l'art

---

Dans le deuxième chapitre, nous avons présenté le contexte dans lequel nous avons réalisé nos travaux de recherche. Nous avons identifié les nouveaux besoins de l'utilisateur qui sont, l'accès à ses services à n'importe quel moment, quelque soit l'endroit, en tenant compte de ses préférences, de sa localisation et de son activité. Comme il a été expliqué dans l'introduction (§I), l'utilisateur est de plus en plus nomade, ce qui peut empêcher d'avoir une continuité de service dû à l'hétérogénéité des technologies (terminaux, réseaux, services) auxquelles il peut être confronté. Afin de répondre à ces nouveaux besoins, nous nous sommes intéressés à la dimension informationnelle, car c'est à travers l'information que nous pouvons avoir un référentiel pour tous les événements, permettant de représenter l'image du monde réel.

Gérer les différents éléments qui utilisent les technologies d'information est un problème assez complexe. Il est aggravé par chaque technologie (et parfois par chaque gamme de produits) qui possède sa propre gestion des données, sa sémantique, sa terminologie, sa structure de données et ses protocoles. Au lieu de se concentrer sur la résolution de problèmes ou d'améliorer la performance globale, la plupart des efforts sont consacrés à la gestion des données en essayant de parvenir à une représentation unique et cohérente de toutes les données de gestion.

Pour pouvoir aborder ces nouveaux besoins, nous avons réalisé un état de l'art des propositions existantes qui traitent cette dimension informationnelle (§III.1). Il est essentiel aussi de comprendre comment la gestion informationnelle est faite dans les nouvelles architectures NGN (Next Generation Network). C'est pourquoi nous avons étudié plus finement l'architecture d'IMS (IP Multimedia Subsystem), dans laquelle, nous avons analysé ses blocs fonctionnels, ainsi que les protocoles utilisés. Nous portons une attention particulière à la structure informationnelle, car c'est de cette façon que nous pouvons faire une analyse des informations qui y sont stockées notamment dans le HSS (Home Subscriber Server) (§III.2). Bien que IMS permette la convergence des différentes architectures d'accès, nous devons avoir la connaissance de chaque ressource qui fait partie du système d'information de l'utilisateur. C'est pour cette raison que nous devons étudier la façon d'avoir une connaissance de notre environnement. C'est pour cela que nous regardons le comment passer "des données brutes" et des informations vers "des connaissances" dans les systèmes d'information (§III.3). Bien qu'il soit important d'avoir une connaissance des ressources, il est nécessaire de gérer aussi le changement de comportement de ressources d'après les événements possibles qui peuvent se produire (§III.4). Enfin, nous présentons les travaux existants (essentiellement les travaux et résultats du groupe AIRS de Télécom-ParisTech) concernant la gestion de la connaissance afin de structurer les informations pour qu'elles deviennent une connaissance (§III.5). Nous finissons par une analyse de cet existant en guise de conclusion (§III.6).

### **III.1 Propositions existantes au niveau informationnelles du plan de gestion**

Afin de réaliser une gestion de ressource efficace, nous avons besoin de connaître les ressources et la façon d'accéder aux informations représentatives du comportement de toutes ces ressources. C'est la raison pour laquelle un modèle informationnel doit être défini, fournissant une structuration des données de gestion, permettant l'accès à l'information élémentaire. C'est donc, le modèle informationnel qui fournit une représentation des ressources et des systèmes par les données dans le but de la gestion. Ces informations se rapportent à la description et au comportement des protocoles (états, compteurs, seuils, statistiques, etc.), des éléments de service et à l'environnement (droits d'accès, profils, etc.). Les acteurs principaux sont les organismes de normalisation qui ont fortement contribué à la définition d'un modèle informationnel. Nous présentons ci-dessous les approches proposées par des organismes comme la DMTF<sup>7</sup> qui développe et maintient des standards pour l'administration de systèmes informatiques d'entreprise comme CIM<sup>8</sup> (§III.1.1), DEN<sup>9</sup> (§III.1.2). Nous regardons aussi du côté ITU-T TMN<sup>10</sup> qui propose un modèle informationnel générique pour la gestion de réseau dénommé GNIM<sup>11</sup> (§III.1.3). Nous prêtons attention aussi à l'organisme TMF<sup>12</sup> qui se focalise sur l'automatisation de la gestion opérationnelle et des processus « business ». Cet organisme à travers le programme NGOSS<sup>13</sup> propose une vue système sur laquelle ils ont proposé le SID<sup>14</sup> dénommé aujourd'hui IF<sup>15</sup> (§III.1.4). Des travaux à propos de ces approches seront aussi présentés (§III.1.5).

#### **III.1.1 Le CIM (Common Information Model)**

Le Common Information Model (CIM) [1] est un modèle conceptuel proposé par la DMTF qui est utilisé pour décrire une large gamme d'informations de gestion dont le but est de fournir une définition commune pour les : Systèmes, réseaux, applications et services. Il fournit une définition cohérente et une structure des données, en utilisant des techniques orientées objet. Nous analysons le CIM afin de savoir si les informations qui sont spécifiées dans les ressources nous permet

---

7 Distributed Management Task Force, Inc.

8 Common Information Model

9 Directory Enable Network

10 Telecommunications Management Network

11 Generic Network Information Model

12 Telemanagement Forum

13 New Generation Operations Systems and Software

14 Shared Information and Data Model

15 Information Framework

---

d'avoir l'image réelle de la ressource que nous voulons gérer. Le standard CIM est divisé en deux parties : *CIM Infrastructure Specification* et le *CIM Management Schema*. Il reste à savoir si le CIM permet de résoudre le nouveau contexte NGN (§II) où il nous faut gérer les ressources suivant les différents types de mobilité à différents niveaux de visibilité.

Le *CIM Infrastructure Specification* définit l'architecture et les concepts de CIM. C'est une approche pour la gestion des systèmes et des réseaux qui utilise une structuration et une technique de conceptualisation basées sur le paradigme orienté objet. Ce modèle comprend les expressions pour les éléments communs, des classes d'objets, propriétés, méthodes et associations qui doivent être clairement présentés aux applications de gestion. La spécification définit les syntaxes et les règles. Cette syntaxe peut être basée sur IDL (Interface Definition Language) et sur le MOF (Managed Object Format).

Le *CIM Management Schema* (Figure III.1) utilise un méta-schéma qui est une définition formelle du modèle. Il définit les limites employées pour exprimer le modèle, son utilisation et sa sémantique. Il couvre la plupart des éléments d'un environnement IT. Il capture des notions qui sont applicables à des champs communs. Il offre des classes d'objets, propriétés, méthodes, et des associations communes afin d'utiliser la gestion des applications sous forme de schéma.

Les langages utilisés pour définir les éléments du Schéma de CIM sont, le MOF<sup>16</sup>, UML<sup>17</sup> et XML<sup>18</sup>. Le CIM Management Schema est divisé en trois parties qui sont : Core Model, Common Model, et Extension Schema.

Le «*Core Model*» est un modèle d'information qui capture les notions qui sont applicables à tous les domaines de la gestion et qui n'est pas spécifique à une plate-forme. Il s'agit d'un ensemble de classes et des associations qui établit un cadre conceptuel pour le schéma du reste de l'environnement géré. Des Systèmes, des applications, des réseaux et des informations connexes sont modélisés comme des extensions du modèle de base.

Le «*Common Model*» est un modèle d'information qui capture les notions qui sont communes à un domaine de gestion, mais indépendant d'une technologie particulière ou d'une implémentation. Ces domaines communs sont des systèmes, des applications, des bases de données, des réseaux et des équipements. Le modèle d'information est suffisamment précis pour servir de base pour le développement d'applications de gestion. Ce modèle fournit un ensemble de classes de base pour l'extension dans le domaine de schémas spécifiques à la technologie.

---

16 Managed Object Format

17 Unified Model Language

18 Extensible Markup Language

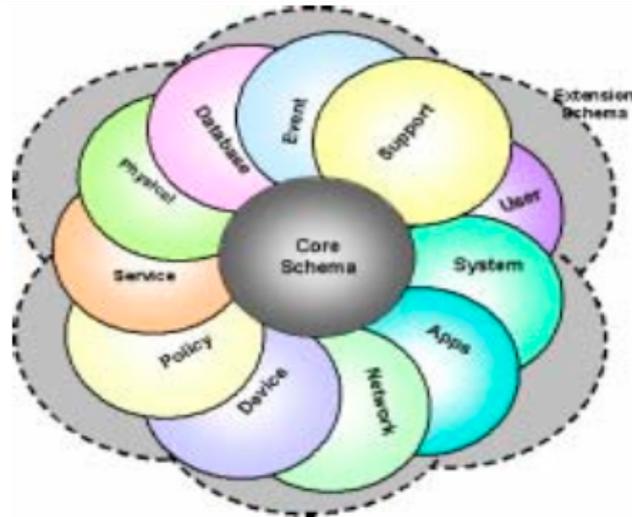


Fig. III.1 CIM Schema

Le «*Extension Schéma*» représente des extensions à des technologies spécifiques du «*Common Model*», ces schémas sont spécifiques à des environnements comme: Systèmes d'exploitation, (Unix, Microsoft Windows).

La classe du modèle cœur du CIM est la classe «*ManagedElement*» (Figure III.2). C'est donc toutes les classes du cœur, et des modèles communs qui héritent directement ou indirectement des propriétés de cette classe abstraite.

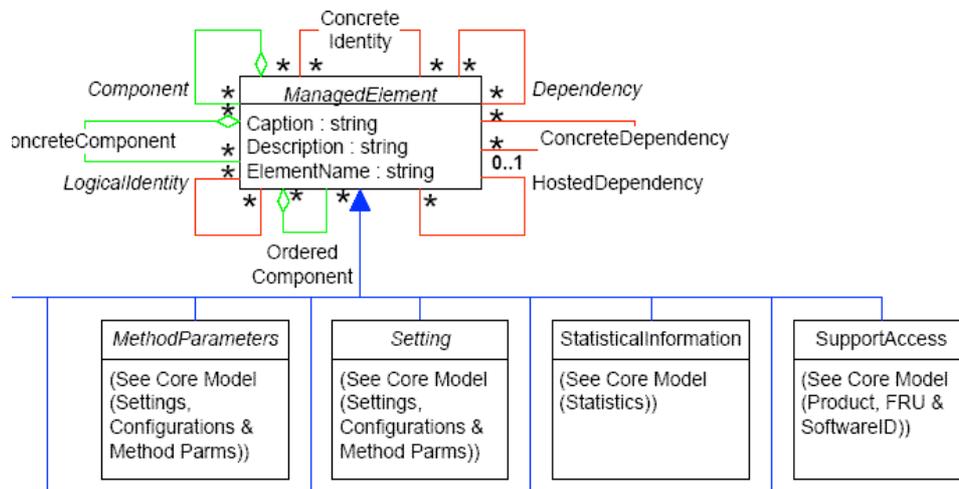


Figure III.2 Classe «*Managed élément*» décrite en UML

La classe «*ManagedElement*» est donc la classe principale de toutes les autres classes. La classe *ManagedSystemElement* décrit tous les éléments gérés du système qui peuvent être soit des éléments logiques (logiciels, réseau, services, etc.) ou physiques (équipements, stockage, etc.). Cette distinction entre les éléments logiques et physiques est intéressante dans la mesure où elle permet d'une part de structurer le modèle cœur et, d'une autre part, de concevoir des éléments logiques indépendamment des contraintes (technologiques, etc.)

---

imposées par des éléments physiques particuliers. Ceci peut permettre d'avoir la possibilité de choisir à tout moment quel est l'élément physique le plus approprié pour supporter les éléments logiques.

Pour conclure, CIM permet de consolider l'information dans l'entreprise et d'unifier l'image des entités à gérer. Ces éléments contribuent à faciliter l'interopérabilité des applications et le partage de l'information dans l'entreprise. Cette solution permet de répondre aux premiers besoins de la gestion, mais présente des limites notamment au niveau du schéma CIM.

La limite réside dans la représentation de l'information comportementale (QoS) au niveau des entités. Il faudrait une maîtrise de cette information qui constitue une condition importante pour une gestion efficace dans les nouveaux contextes dynamiques. Le schéma permet l'obtention des informations statistiques pour chaque élément, mais il ne prend pas en compte la relation et l'impact du comportement d'une entité sur l'autre. De plus, l'absence de machine à états pour diriger le changement du comportement, constitue l'un des problèmes majeurs du CIM, ce qui empêche de garantir automatiquement la cohérence des données.

### ***III.1.2 Le DEN (Directory Enable Network)***

Le Directory Enable Network (DEN) [2] a été conçu par la DMTF pour fournir les blocs de base pour une gestion plus intelligente en tenant compte des concepts de CIM (tels que les systèmes, services et politiques) dans un répertoire, et en intégrant cette information avec d'autres éléments WBEM<sup>19</sup> dans l'infrastructure de gestion.

DEN est une initiative des organismes de standardisation pour répondre au comment construire et stocker des informations des utilisateurs d'un réseau, des applications et des données dans un répertoire central. Une méthode standard de description des éléments du réseau dans un référentiel central peut permettre le développement d'applications qui va automatiquement apprendre des privilèges d'accès de l'utilisateur, de sa bande passante, des politiques des ressources de l'entreprise, ainsi que de fournir des services en rapport à ses politiques. Le résultat devrait permettre de réduire les coûts de fonctionnement du réseau et permettre de déployer de nouveaux services.

Certains préconisent que le service d'annuaire est désormais au cœur du réseau, l'endroit où toutes les opérations de réseau sont définies et gérées.

DEN vise à garantir l'interopérabilité entre les fournisseurs d'équipements réseau, des services d'annuaires et des applications. Les principaux composants de DEN comprennent une base de données distribuée, un modèle de données cohérent et

LDAP<sup>20</sup>. La base de données distribuée peut être reproduite à d'autres endroits. Le modèle de données définit la structure des données, c'est-à-dire qu'il spécifie comment les objets du monde réel sont définis dans le répertoire. LDAP est un protocole que les applications utilisent pour accéder au répertoire.

Donc, en résumé, on peut en conclure que DEN :

- Est une extension de CIM (§III.1.1)
- Est une corrélation de l'information dans un format qui peut être stocké dans un répertoire qui utilise LDAP comme protocole d'accès

Le schéma d'un répertoire définit l'ensemble des objets qui peuvent être créés dans ce répertoire et l'ensemble des attributs qui peuvent être utilisés pour décrire ces objets.

Le schéma DEN se compose de classes de base abstraites dont toutes les autres classes spécifiques au réseau sont dérivées. Les classes de base sont affinées par la spécialisation du modèle de base pour représenter les éléments de réseau, les services, les consommateurs, etc. La Figure III.3 montre la structure fonctionnelle des classes de base DEN, avec des classes clés définies par la CIM, X.500, et DEN énumérées séparément.

Le DEN est l'agrégation de concepts à partir de la version actuellement publiée des spécifications CIM (2,0), la version actuellement publiée de la spécification X.500 et un assemblage de nouveaux éléments. Ces derniers ont été proposés à partir des spécifications de CIM et X.500 afin de modéliser les éléments du réseau et des services.

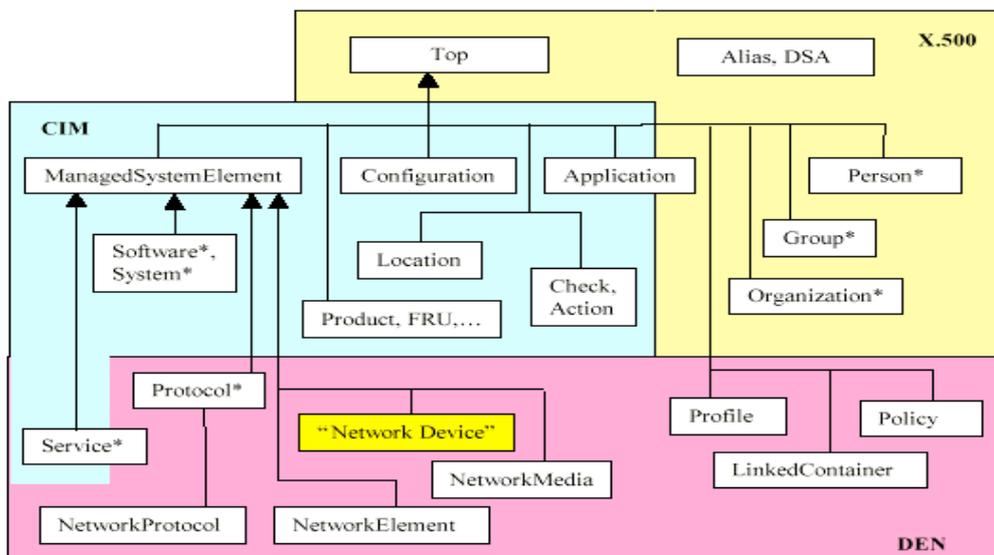


Figure III.3 Structure de DEN

20 Lightweight Directory Access Protocol

---

Voici une brève explication des classes dérivées du **X.500** (Figure III.3, partie jaune).

**Top** : Racine de l'arborescence de répertoires.

**Person** : concept générique d'une personne, un employé, une personne ayant une résidence. DEN l'utilise comme un client de services de réseau à lier, ou en tant que propriétaire / l'administrateur d'un dispositif ou un service.

**Group** : regroupement fournissant des constructions pour les utilisateurs ainsi que des dispositifs.

**Organization** : entité commerciale à laquelle les dispositifs et les services appartiennent.

**Application** : Cette classe définit les processus d'application et les entités de l'application. Cette classe permet d'ajouter des informations afin que ces classes puissent être associées à des éléments de réseaux et de services.

**Alias, DSA**: entités nécessaires à l'exploitation correcte du répertoire.

Voici une brève explication des classes dérivées du **CIM** (Figure III.3, partie bleu).

**Product, FRU, etc** : une collection de classes qui représentent un produit et pièces de rechange d'un produit.

**ManagedSystemElement** : Classe de base pour tout système ou composant du système qui doit être géré.

**Configuration** : Elle permet de modéliser et provisionner la configuration des éléments de réseaux et services.

**Service** : Définition, gestion et prestation de services de réseau.

**Software** : la notation générale du logiciel.

**System** : Permet de se rendre compte du concept d'un élément du réseau logique.

**Location** : Spécification de l'adresse et de l'emplacement d'un élément physique.

**Check and Action** : Utilisé par DEN pour augmenter les notions de configuration et redémarrer des périphériques réseau.

**Application** : CIM ajoute le concept selon lequel une application est utilisée pour maintenir une fonction en particulier.

Nous donnons ensuite les classes spécifiques du DEN (partie rose) :

Nouvelles classes DEN :

**NetworkService** : racine de la hiérarchie des services réseau.

**NetworkProtocol** : Racine de toutes les classes de protocole réseau.

**Enhancements to PhysicalPackage and Card**: des extensions et améliorations afin d'inclure les fonctionnalités requises par les périphériques réseau.

**NetworkElement** : aspects logiques d'un élément du réseau.

De nouveaux concepts définis par DEN:

**Policy** : Règle chargeant un nœud de réseau sur la façon de gérer les demandes de ressources du réseau.

**Profile** : Modèle d'attributs qui décrivent un objet ou un ensemble d'objets.

**NetworkMedia** : Associer les médias notamment d'une interface donnée avec les services qui s'exécutent sur elle.

**LinkedContainer** : classe Container mise en œuvre d'un lien.

Voici quelques avantages de l'utilisation de DEN :

- La première est que DEN est un modèle d'information orienté objet qui décrit les différentes composantes d'un environnement géré de manière commune. Cela permet qu'une relation étroite soit établie entre les classes qui définissent des éléments de réseau et des services et des classes qui définissent d'autres objets. C'est le principal mécanisme utilisé pour définir les services réseau dont le client a besoin.
- Deuxièmement, DEN est orienté objet, il est donc intrinsèquement extensible. Cela signifie que des concepts non encore définis dans DEN peuvent être facilement modélisés et ajoutés à la norme.
- Troisièmement, DEN permet au développeur d'une application ainsi qu'au concepteur du réseau de penser que le réseau est vu comme un fournisseur de services intelligents. Cela permet aux développeurs d'applications de décrire les fonctions et le traitement que le trafic de leurs applications exige dans le sens où le réseau peut le représenter directement. Ainsi, DEN peut être utilisé pour définir l'ensemble des services qui, ensemble, répondent à ces exigences.
- Quatrièmement, DEN permet aux entreprises de donner la priorité au traitement des différentes applications qui sont en concurrence pour les ressources réseau. Cela permet à un administrateur d'entreprise, d'écrire une politique qui dit que l'application « PeopleSoft » devrait recevoir un traitement préférentiel sur le trafic FTP. Cela permet au réseau d'être conçu pour traiter les applications qui sont essentielles pour que l'entreprise fonctionne selon ses règles et son organisation.

Le désavantage de l'approche DEN, est similaire à celle de CIM, c'est-à-dire, que l'on peut obtenir des informations au niveau des différentes ressources, mais on n'a pas toutes les informations pour gérer le comportement de ces éléments (QoS).

### **III.1.3 Le GNIM (Generic Network Information Model)**

Le GNIM [3] a été proposé par l'ITU-T dans la recommandation M.3100. GNIM est un modèle informationnel générique de réseau de télécommunication. C'est un schéma de gestion générique des informations qui peuvent être utilisés pour

---

modéliser les réseaux de télécommunications qui offrent des services orientés connexion. Il fournit le point de départ pour définir des informations de gestion plus spécialisées pour des technologies spécifiques et de services de télécommunications.

Il offre grâce à sa généralité l'avantage d'être indépendant d'une technologie de réseau particulière; il peut donc être utilisé dans différentes technologies comme l'ATM, le FrameRelay, la SDH, le PDH, le RNIS, etc. Il se focalise principalement sur la description des ressources physiques du réseau géré. Le modèle peut être vu comme un ensemble de classes réutilisables pour la gestion d'un réseau de télécommunication. Le modèle a pour but de couvrir toutes les couches logiques du TMN, mais dans son état actuel il est applicable principalement à la couche EML.

GNIM prend l'information mise au point pour appuyer les modèles SDH (en G.803), et ajoute plusieurs autres classes d'objets gérés qui fournissent une base pour l'élaboration de modèles d'application spécifiques. En utilisant les classes M.3100, l'intégration des modèles d'information dans la vision d'entreprise à plus grande échelle est possible. Comme un «modèle générique de l'information», M.3100 fournit généralement les classes d'objets utiles qui peuvent être appliquées à l'appui de l'architecture TMN.

La spécification M.3100 est organisée en 6 «segments» qui se combinent pour former le modèle d'information générique et qui sont :

***Network fragment*** : définit la relation entre un réseau géré et ses voies connexes, les connexions, et des éléments gérés. Dans ce cas, un fragment de réseau est indiqué pour contenir tous les éléments.

***Managed Element*** : définit les composantes et les relations contenues dans un seul élément réussi. Dans ce cas, un élément géré est montré pour contenir l'équipement (y compris les logiciels).

***Termination Fragment*** : Le fragment de point de terminaison contient les types de sorties d'une seule pièce que l'équipement géré peut contenir. Les deux voies et des points de terminaison de connexion sont inclus dans ce fragment.

***Transmission Fragment*** : Fournit une autre vue non-orientée équipement de communication à travers un réseau. Dans ce cas, deux formes d'entités de transmission sont définies, les voies et les connexions. Les relations entre ces entités et les références à ses points de terminaison du rapport sont incluses dans ce fragment.

***Cross Connection Fragment*** : aide à gérer les Croix-topologies connectées. Dans ce cas, le fragment de la Croix-connexion multipoint contient les interconnexions, les points de terminaison générique, et un pool de points de terminaison.

***Functional Area Fragment*** : définit les classes d'objets contenus dans un élément qui fournit des services de gestion supplémentaires. Les classes d'objets contenus

dans le fragment de domaine fonctionnel sont : gestion des opérations de l'annexe, les journaux (par exemple, les alarmes, attribuer les changements de valeur, la création d'objets et de documents, suppression, les dossiers de changement d'état), les profils de cession d'alarme, discriminateurs transfert d'événements et le contrôle d'alarme en cours de synthèse. Parmi ces classes d'objets, à l'exception de la sévérité de l'alarme du profil assigné, toutes sont définies dans X.721 ou Q.821. On peut constater (Figure III.4), à travers ces fragments que, le GNIM (M.3100) fournit un ensemble d'objets pour identifier les topologies de réseau sous plusieurs angles, et de fournir les fonctions de gestion standard.

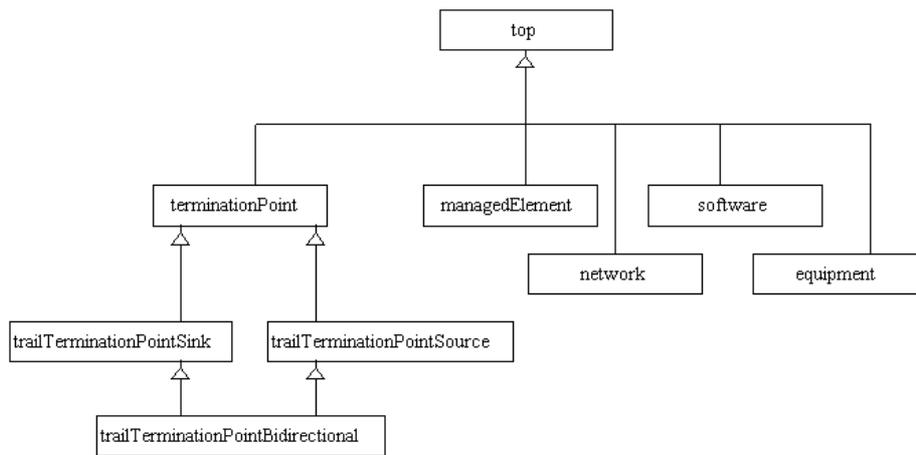


Figure III.4 Hierarchie GNIM

Elle montre la relation d'héritage d'un grand nombre des principaux objets nécessaires pour soutenir le M.3100 « Functional Fragment ». L'objectif est de fournir un cadre utile pour l'organisation des services de gestion de réseau. Le reste des objets dans le diagramme sont définis dans la spécification M.3100.

Pour conclure, GNIM n'a pas des informations au niveau comportemental des entités que l'on veut gérer. Cela veut dire que les aspects de QoS (c'est-à-dire, le comportement) n'est pas tout à fait pris en compte. Ce qui pourrait faire son grand défaut dans un contexte où l'environnement est en constant mouvement. Surtout si, nous considérons que la QoS est un facteur à considérer.

### **III.1.4 Le SID (Shared Information Model)**

Le SID (IF) est un modèle générique, indépendant des plates-formes, des protocoles et des langages spécifiques de mises en œuvre qui a été proposé par le TMF<sup>21</sup> [4], [5], [6] dans le cadre de son programme NGOSS<sup>22</sup>.

Le NGOSS (Figure III.5) propose une approche intégrant les axes suivants :

- Une analyse des processus et flux opératoires de l'opérateur, cette étape utilise le eTOM<sup>23</sup>.
- L'analyse et la conception du système, qui sont basées sur le SIM (System Integration Map).
- L'analyse et la conception du référentiel informationnel (données utilisées dans les architectures NGOSS), basées sur SID.
- L'analyse et la conception de la solution, qui s'intéressent aux problèmes d'interopérabilité et proposent une approche orientée « contrat » pour les interfaces entre systèmes, et l'Architecture TNA (Technology Neutral Architecture).
- La validation de la solution vis-à-vis des critères de conformité émise par le TMF Forum.

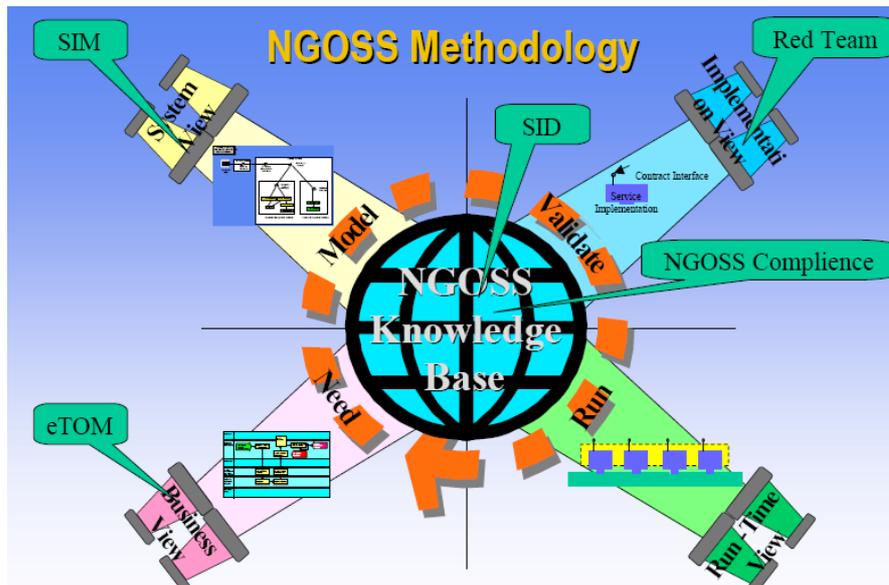


Figure III.5 Méthodologie NGOSS

Le TMF a prévu 4 vues du programme NGOSS qui sont :

<sup>21</sup> Telemanagement Forum : Le TeleManagement Forum est un consortium privé à but non lucratif qui fournit une stratégie ainsi que des solutions pratiques pour améliorer la gestion et le fonctionnement des services de communication et d'information.

<sup>22</sup> New Generation Operations Systems and Software : Le NGOSS est un programme intégré innovant, pour le développement et le déploiement de systèmes de gestion.

<sup>23</sup> enhanced Telecom Operations Map

- **Business view** : elle se focalise sur la modélisation des entités métier. Elle décrit les entités nécessaires aux processus de l'eTOM.
- **System view** : se focalise sur les entités du système. Ces entités sont construites soit par l'extension des entités métier pour supporter les besoins du système NGOSS ou par la définition de nouvelles entités non présentes dans la vue Business. Les extensions se présentent par exemple sous la forme d'ajouts d'attributs ou de méthodes relatifs au niveau système. Quant aux nouvelles entités, elles correspondent à des concepts du niveau systèmes (message bus, observateur, interfaces, etc.).
- **Implementation view** : Dans cette vue, on regarde la mise en place du SID.
- **Deployment view** : se focalise sur les aspects du déploiement.

Le SID est une abstraction des entités business impliquées dans la gestion des opérations de télécommunication, il fournit un modèle de référence d'informations/données partagées et un vocabulaire commun des entités business. Le SID apporte la définition des informations/données qui doivent être affectées par les processus de l'eTOM ainsi que la définition des attributs de ces entités. En utilisant conjointement SID et eTOM, ils offrent un outil efficace pour définir comment les entités s'organisent pour répondre à un besoin business. Les domaines de SID sont structurés en couches (appelés Domaines), chaque couche contient les informations/données liées au domaine concerné (Figure III.6).

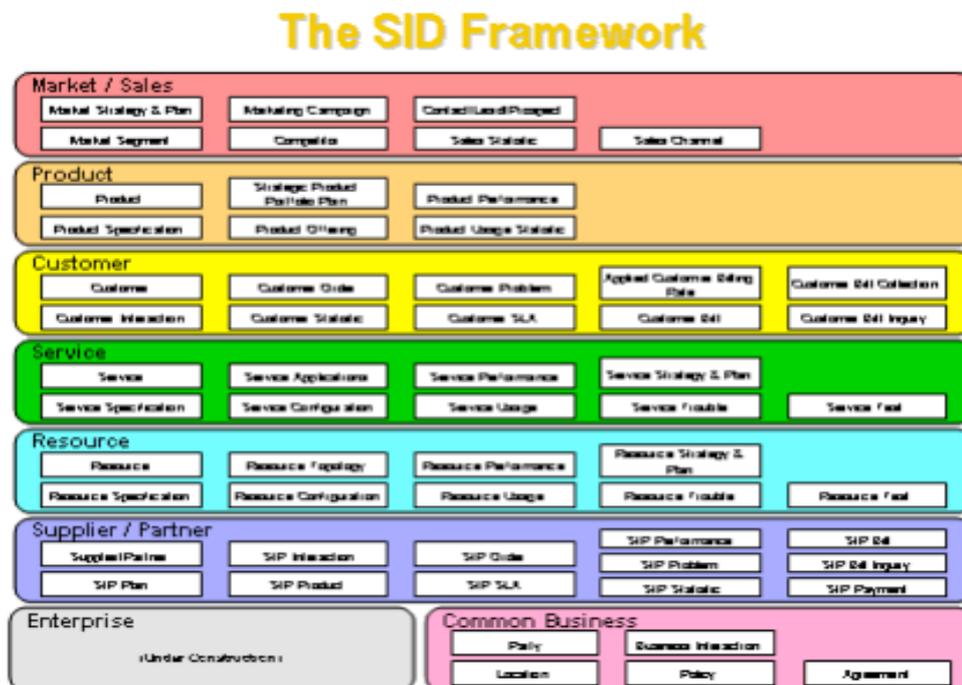


Figure III.6 Domaines SID

---

Ce cadre présente les domaines ainsi que les entités métier de chaque domaine. Il y a une ressemblance frappante entre ces domaines et la carte des processus eTOM et plus exactement avec le niveau 0 de l'eTOM (eTOM process level 0).

Cet alignement entre les domaines du SID et ceux de l'eTOM nous donne une première idée sur la nature des domaines du SID. Ce sont des domaines de gestion dans lesquels des informations de gestion vont être structurées et modélisés. Le SID couvre toutes les informations nécessaires aux processus du eTOM.

On peut en conclure que le principal point fort de SID réside dans la modélisation des services et leurs relations avec les ressources et les produits. SID propose dans le document GB922-QoS la classe « ServiceQoS ». Cette classe représente les mécanismes de QoS mis en œuvre dans le réseau (diffserv, meter, dropper, etc.) pour estimer la qualité du trafic du réseau. La classe est utile pour avoir le comportement des flux et des services offerts par le réseau (ADSL, VPN, Internet, etc.) mais cette information concerne les services réseaux (délivrance du média) et est insuffisante pour le comportement des services applicatifs. C'est également insuffisant pour la délivrance du service lors des différentes mobilités. Ceci constitue un manque majeur dans le cadre de notre problématique NGN/ NGS que nous voulons résoudre.

### ***III.1.5 Travaux sur la gestion de l'information à partir des modèles***

Nous avons fait une analyse des travaux concernant la gestion de l'information à travers les modèles. Cette analyse a permis de mettre en exergue les manques que nous voulons combler par nos travaux et propositions.

- ***Towards a Context-Aware Information Model for Provisioning and Managing Virtual Resources and Services***

Tout d'abord nous avons le travail proposé par Yeongrak Choi et al [7] qui propose l'extension du modèle d'information DEN-ng pour représenter et gérer l'approvisionnement virtuel et non virtuel de ressources hétérogènes et des services supportés.

Dans l'approvisionnement, ce modèle contribue à diminuer l'effort de configuration des ressources correspondantes virtuelles et non virtuelles, ce qui simplifie sa conception, le déploiement et la gestion des services. En outre, ce modèle montre comment réduire la complexité de gestion en fournissant une gestion unifiée des mécanismes de ressources à la fois virtuelles et non virtuelles.

Leur approche de modélisation permet d'identifier les éléments et les fonctions en reliant à des ressources virtuelles et non virtuelles, et ensuite en définissant comment les deux sont associées au même SLA (§Figure III.7).

Cet article a brièvement décrit un modèle, et, le plus important, a montré comment il peut être utilisé pour personnaliser les services avec des ressources virtuelles.

La Figure III.7 montre comment ce cas d'utilisation peut être appliqué en utilisant leur modèle proposé lorsque un Système Virtuel est provisionné à base des Templates Virtuels et des images virtuelles.

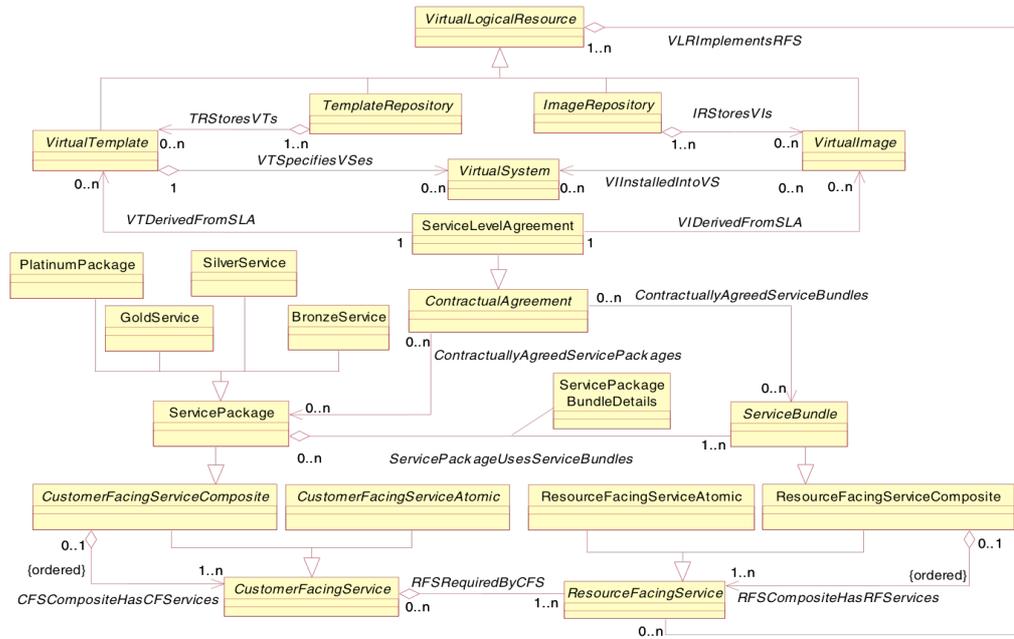


Figure III.7 Provisionnement d'un SLA personnalisé avec des ressources virtuelles

Pour fournir des services personnalisés, un fournisseur peut définir un ensemble de services, où chaque service a des caractéristiques différentes pour différentes applications spécifiques à l'utilisateur. Par exemple, le fournisseur peut définir quatre différentes classes de service concrètes, telles que le service au niveau premium, or, argent et bronze, pour décrire les aspects uniques des différents services personnalisés.

Bien que leur approche permette de gérer des ressources virtuelles personnalisées, il manque la prise en considération des préférences de l'utilisateur afin que les services soient vraiment personnalisables.

### • **Ontology-based Capability Self-Configuration**

Nous regardons ensuite le travail qui a été proposé par Patcharee Thongtra et al, dans lequel il a été proposé une autoconfiguration basée sur les ontologies [8]. En effet, les ontologies proposées jusqu'à maintenant ont été utilisées pour la gestion de réseaux à travers des modèles d'information comme DEN-ng et CIM. Ces ontologies peuvent représenter seulement les noeuds de réseau et leurs propriétés. La difficulté se base sur le manque de représentation des relations entre les éléments. La description des possibilités décrit des généralités au niveau réseau sans avoir d'informations spécifiques de capacité propre et d'état.

Le travail de fond de leur proposition est donc basé sur comment représenter et gérer les capacités des ressources et leur relation. Ils ont proposé un modèle sur trois couches, une couche serait pour l'interface, la deuxième couche serait pour le traitement et la troisième serait pour les noeuds du réseaux. D'après la spécification de la gestion des web-services (WS-Management), les noeuds du réseau implémentent des services Web permettant à d'autres ressources de les gérer. Ils utilisent pour cela les actions get, set, create, delete afin de paramétrer les capacités des ressources.

Cette approche est fortement dépendante des web-services et des l'interface Web. Ce qui ne permet pas à une ressource être totalement autonome, et de répondre à des compositions de service personnalisées.

- ***The Applicability of Self-Awareness for Network Management Operations***

Le travail de John Strassner et al, traite la complexité de la gestion de réseaux et terminaux dans le contexte NGN (Netx Generation Network) où l'hétérogénéité des terminaux présente un problème pour proposer de nouvelles approches au niveau « Business ». Ceci est dû à la structuration de chaque mobile qui est différente selon le constructeur. Il est donc difficile d'avoir la connaissance fiable des ressources à travers lesquelles l'utilisateur accède à son service. La majorité de langages, interfaces et structures de données utilisées dans les terminaux (ex: SNMP) ne permet pas de connaître les terminaux. Selon John Strassner et al, ceci peut être corrigé en proposant une nouvelle représentation de la gestion des opérations et des données.

Leur travail (Figure III.8) a exploré le concept de « self-awareness » (capacité d'une entité à à comprendre son propre comportement). Ils argumentent que le self-awareness est essentielle pour progresser dans les scénarios actuels et ceux de demain en ce qui concerne la gestion de réseau.

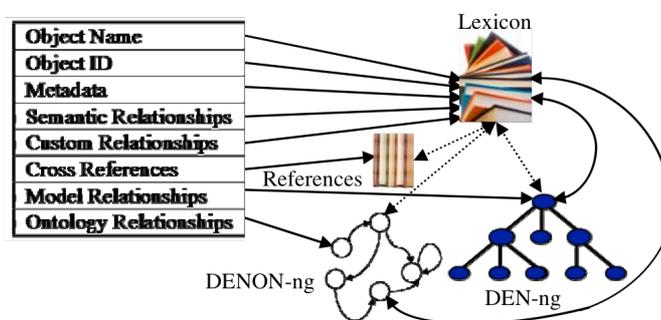


Figure III.8 Structure FOCAL Lexicon

John Strassner et al, décrivent une approche de formalisation de la sémantique du « self-awareness » en utilisant un lexique pour coordonner et faire la traduction

entre différentes sources de connaissance. C'est ainsi que leur projet nommé FOCALE utilise le modèle d'information DEN-ng et l'ontologie DENON-ng. Ils ont été conçus pour fonctionner ensemble pour fournir des connaissances consensuelles dans une forme officielle. Ils utilisent le concept de relation sémantique en conjonction avec une nouvelle méthodologie permettant de construire une représentation graphique de la théorie de la connaissance. Ceci donne une application souple et extensible qui peut intégrer des changements à leur base de connaissances (Knowledge Base) à travers laquelle ils apprennent par l'expérience, en observant leur système déployé.

Bien que ce travail a permis d'homogénéiser la structure de données de différents types de terminaux, leur approche ne prend pas en compte les aspects de l'environnement à travers lequel la ressource peut être utilisée. Il manque également les informations de QoS qui permettrait d'avoir la connaissance totale de la ressource.

### ***III.2 Gestion de l'information des architectures NGN : IMS (IP Multimedia Subsystem)***

Nous devons maintenant analyser comment les architectures NGN comme c'est le cas d'IMS (IP Multimedia Subsystem) gère l'information qui lui est confiée. Rappelons que l'architecture IMS (IP Multimedia Subsystem) [9] a été définie initialement par le 3rd Generation Partnership Project (3GPP), qui est un accord de collaboration entre un certain nombre d'organismes de normalisation des télécommunications. Ceci a été réalisé dans le cadre de leurs travaux de normalisation pour soutenir les réseaux GSM et l'évolution de la technologie radio. Il faut savoir que l'IMS a été introduit dans la release 5 de la 3GPP.

Avant de regarder les aspects informationnelles, nous présentons tout d'abord les blocs fonctionnels d'IMS (§III.2.1) et ensuite les protocoles qui sont utilisés (III.2.2). Ces protocoles sont SIP (Session Initiated Protocol) et Diameter. SIP a été défini par l'Internet Engineering Task Force (IETF) et a été choisi comme protocole de signalisation. Diameter a été aussi défini à l'IETF et est utilisé pour accéder au HSS d'IMS, c'est pour cela que nous portons une attention particulière à ce protocole d'accès aux bases informationnelles. Enfin, nous regardons la structure informationnelle du HSS (Home Subscriber Server) (§III.2.3).

#### ***III.2.1 Architecture NGN : «IMS»***

Les objectifs de l'IMS sont d'offrir des services, existants ou futurs, proposés sur le réseau et d'avoir la possibilité d'utiliser ces services aussi bien en déplacement, que depuis chez soi. D'avoir aussi une continuité de session, une indépendance des services face aux différents réseaux d'accès et terminaux. L'IMS a été conçu à

l'origine pour les réseaux mobiles, mais avec l'ajout des travaux de TISPAN dans la version 7, les réseaux fixes sont également supportés. On appelle cela la convergence Fixe/Mobile (FMC). Et donc globalement IMS supporte la convergence de réseaux. C'est-à-dire que l'on peut accéder à un même service quel que soit le réseau d'accès.

Voici une brève description des composants qui composent l'architecture IMS, tel que le CSCF et le HSS (§III.9).

Le contrôle d'appel initié par un terminal IMS doit être pris en charge dans le réseau nominal (réseau auquel l'utilisateur a souscrit à ses services IMS) car l'utilisateur correspondant peut souscrire à un grand nombre de services et certains d'entre eux peuvent ne pas être disponibles ou peuvent fonctionner différemment dans un réseau visité, notamment suite à des problèmes d'interaction de service. Cela a induit la définition de trois entités CSCF (Figure III.9, couche 2) : P-CSCF (Proxy CSCF), I-CSCF (Interrogating CSCF) et S-CSCF (Serving-CSCF).

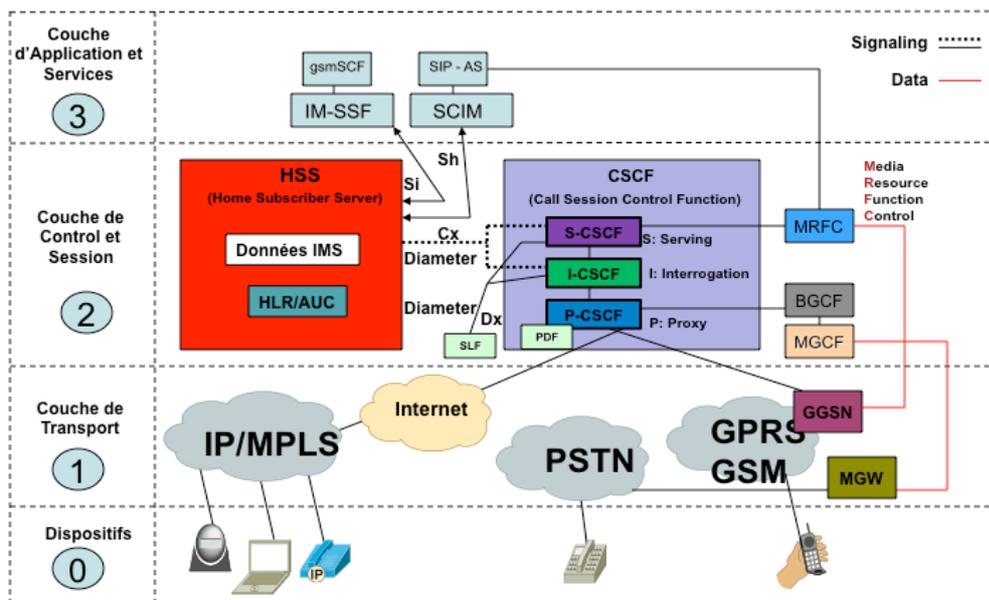


Figure III.9 Release 8 Architecture IMS

### Le « P-CSCF » (Proxy - Call Session Control Function)

Le composant P-CSCF est le premier point de contact pour les utilisateurs au sein de l'IMS. Cela signifie que tous les messages SIP de signalisation de l'équipement de l'utilisateur seront envoyés au P-CSCF. De même, toutes les fins de signalisation SIP à partir du réseau sont envoyées par le P-CSCF à l'équipement de l'utilisateur. Il y a quatre tâches réalisées par le P-CSCF qui sont : la compression SIP, l'association de sécurité IPsec, l'interaction avec le composant PDF<sup>24</sup> et la détection de session extraordinaire d'urgence.

<sup>24</sup> Policy Decision Function

Le P-CSCF est responsable du maintien de la sécurité d'associations de l'application de l'intégrité et de la protection de la confidentialité de la signalisation SIP.

#### **Le « I-CSCF » (Interrogating - Call Session Control Function)**

Le composant I-CSCF est un point de contact au sein d'un opérateur réseau pour toutes les connexions à destination d'un abonné de ce réseau opérateur. Il y a quatre tâches attribuées à l'I-CSCF qui sont :

- Obtenir le nom du prochain saut (soit S-CSCF ou serveur d'applications) du HSS. Serveur abonné à domicile (SAD).
- Attribuer une S-CSCF fondée sur les informations reçues du HSS.
- Assigner le S-CSCF, lorsque un utilisateur s'enregistre dans le réseau ou lorsqu'un utilisateur reçoit une requête SIP, alors qu'elle n'est pas enregistrée à partir du réseau, mais offre des services liés à un état non inscrit (messagerie vocale, par exemple).
- Router les requêtes entrantes qui sont en outre attribuées à un S-CSCF dont dépend le serveur d'application (AS).

#### **Le « S-CSCF » (Serving - Call Session Control Function)**

Le composant S-CSCF est le point clé pour IMS, car il est responsable de la gestion du processus d'enregistrement, il prend des décisions de routage, et maintient les états des sessions. Le profil de l'utilisateur qui est stocké dans le HSS et aussi téléchargé dans le S-CSCF afin de choisir le serveur d'applications qui correspond au service demandé.

Le S-CSCF prend en charge le contrôle de la session. Il maintient un état de Session afin de pouvoir invoquer des services. Dans un réseau d'opérateur, différents S-CSCF peuvent présenter des fonctionnalités différentes.

Les fonctions réalisées par le S-CSCF pendant une session comprennent :

- L'émulation de la fonction Registrar puisqu'il accepte les méthodes SIP d'enregistrement et met à jour le HSS.
- L'authentification des utilisateurs.
- L'émulation de la fonction Proxy server puisqu'il accepte les méthodes SIP et les achemine.
- L'émulation de la fonction User Agent, lorsqu'il exécute par exemple des services complémentaires.
- L'interaction avec des serveurs d'application après avoir analysé les critères de déclenchement des services correspondants.

Après avoir fait connaître les composants de l'architecture IMS, nous devons regarder la façon de communiquer entre ces composants, IMS utilise le protocole SIP pour la signalisation et le protocole diameter pour l'accès aux informations au niveau du HSS. Nous décrivons ces deux protocoles ci-dessous.

---

### **III.2.2 Protocoles associés à l'architecture IMS**

#### **Le Protocole SIP**

SIP [10] est un protocole de signalisation défini par l'IETF (Internet Engineering Task Force) permettant l'établissement, la libération et la modification de sessions multi-médias. SIP est utilisé dans l'IMS comme protocole de signalisation pour le contrôle de sessions et le contrôle de service. Il remplace donc à la fois les protocoles ISUP (ISDN User Part) et INAP (Intelligent Network Application Part) du monde de la téléphonie en apportant la capacité multimédia. Il hérite de certaines fonctionnalités des protocoles, HTTP (Hyper Text Transport Protocol) utilisé pour naviguer sur le WEB, et SMTP (Simple Mail Transport Protocol) utilisé pour transmettre des messages électroniques (E-mails). SIP s'appuie sur un modèle transactionnel client/serveur comme HTTP.

SIP fournit les possibilités de :

- **Établir une session entre le point origine et le point de terminaison.** Si l'appel peut être accompli, SIP établit une session entre les points d'extrémité.

Les avantages de SIP sont de prendre en compte les changements, tels que l'addition d'un autre point d'extrémité à la conférence ou le changement d'une caractéristique ou d'un codec.

- **Manipuler le transfert et l'arrêt de l'appel.** SIP permet le transfert des appels d'un point d'extrémité à l'autre. Pendant un transfert d'appel, SIP établit simplement une session entre le cessionnaire et un nouveau point de terminaison (indiqué par la partie de transfert) et termine la session entre le cessionnaire et la partie de transfert. À la fin d'un appel, SIP termine les sessions entre toutes les parties.

#### **Le protocole Diameter**

Avec le déploiement des mobiles, il est devenu nécessaire de développer des protocoles permettant à des utilisateurs de se déplacer de réseau en réseau. Les protocoles AAA (Authentication, Authorization, Accounting) permettent aux opérateurs d'authentifier des utilisateurs, de leur autoriser certains services et de collecter des informations sur l'utilisation de ces ressources. Diameter [11] est actuellement le protocole utilisé, il permet de répondre aux nouveaux besoins. En particulier, il permet aux opérateurs d'authentifier un utilisateur ayant souscrit un abonnement auprès d'un autre opérateur.

Nous décrivons dans la Figure III.10 l'entête du protocole Diameter qui est de la forme suivante :

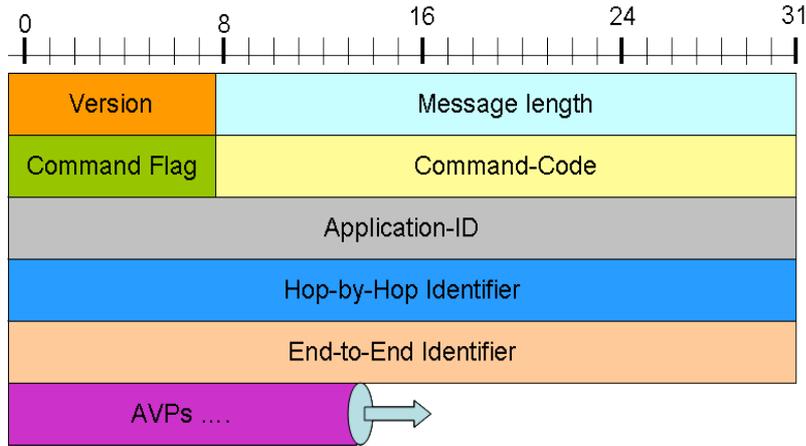


Figure III.10 Structure de Diameter

**Les Composants de Diameter sont :**

**Version** : indique la version de Diameter.

**Message length** : ce champ est de 3 octets et indique la longueur du message en incluant le header.

**Command flag** : ce champ est de 8 bits, et chaque bit correspond à une information :



Figure III.11 Flags de Diameter

**R(equest)** - Si R=1, le message est une requête, sinon c'est une réponse.

**P(roxiabile)** - Si P=1, le message doit être transmis, sinon il doit être traité localement.

**E(rror)** - Si R=1, le message contient une erreur de protocole et le message ne va pas être conforme au ABNF décrit dans cette commande.

**T (Potentially re-transmitted message) :**

- Ce flag est égal à 1 après l'échec d'une procédure de mise en relation pour aider à la suppression des requêtes dupliquées. Il ne doit pas être égal à 1 dans les messages de réponse.

**r (eserved)** - Ces flags qui sont réservés pour un usage futur, doivent être égaux à 0 et ignorés par le destinataire.

---

**Command-Code** : ce champ est de 3 octets, il est utilisé pour communiquer la commande associée au message. Cet espace de 24 bits pour l'adresse est géré par l'IANA.

**Application-ID** : ce champ est de 4 octets, il est utilisé pour identifier à quelle application s'adresse le message. Cette application peut être une application d'authentification, d'autorisation, de comptes, ou une application spécifique du vendeur.

**Hop-by-Hop Identifier** : le message Diameter passe à travers plusieurs routeurs. Le Hop-by-Hop Id est un champ de 32 bits qui aide à faire correspondre les requêtes et les réponses.

**End-to-End Identifier** : c'est un champ de 32 bits qui est utilisé pour détecter les messages dupliqués. La valeur de ce champ, qui est souvent l'identité de l'initiateur de la requête, est inchangée en cours d'acheminement de la source à la destination.

Le protocole Diameter est utilisé dans le cœur IMS par différents composants, tels que le HSS avec le S-CSCF, le I-CSCF et les AS, pour communiquer des données entre eux (AAA). Ces communications avec le protocole Diameter se font en parallèle de la signalisation supportée par le protocole SIP.

### ***Les AVPs (Attribute Value Pairs) du protocole diameter***

Les AVPs sont une méthode d'encapsulation des informations pertinentes du message Diameter. Elles sont employées par le protocole Diameter pour réaliser les services suivants :

- le transport d'information d'authentification de l'utilisateur, dans le but de permettre au serveur de Diameter d'authentifier l'utilisateur.
- le transport d'informations spécifiques d'autorisation de service, entre le client et les serveurs, permettant de décider si on doit accorder une demande d'accès de l'utilisateur.
- l'échange d'informations d'utilisation de ressources.
- la transmission et la réorientation des messages de Diameter.

Quelques exemples de commandes de Diameter avec leurs codes :

Command-Name	Abbrev.	Code
Abort-Session-Request	ASR	274
Abort-Session-Answer	ASA	274
Accounting-Request	ACR	271
Accounting-Answer	ACA	271
Capabilities-Exchange-Request	CER	257
Capabilities-Exchange-Answer	CEA	257

Device-Watchdog-Request	DWR	280
Device-Watchdog-Answer	DWA	280
Disconnect-Peer-Request	DPR	282
Disconnect-Peer-Answer	DPA	282
Re-Auth-Request	RAR	258
Re-Auth-Answer	RAA	258
Session-Termination-Request	STR	275

Figure III.12 - Les codes des AVPs

Les AVPs de Diameter supportent des informations spécifiques d'autorisation, d'authentification, de compte, de routing, et de sécurité aussi bien que des détails de configuration pour la demande et la réponse.

L'entête d'un AVP est de la forme suivante :

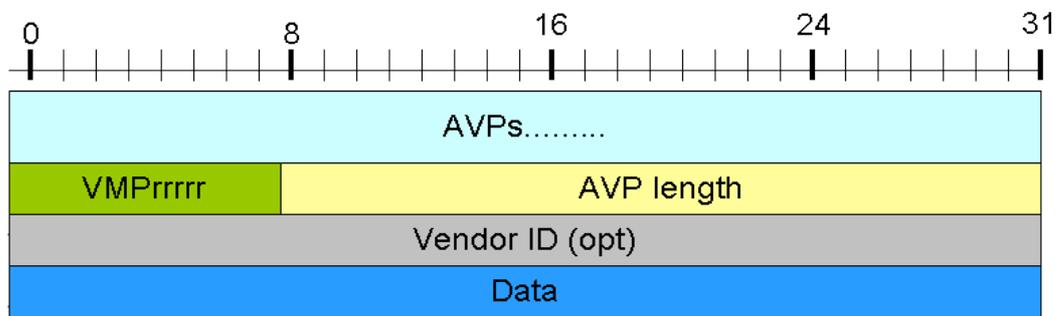


Figure III.13 Entête des AVPs

AVP code : identifie uniquement l'attribut.

AVP flag : indique au destinataire comment les attributs doivent être traités.

- « r » : sont réservés et doivent restés à 0.
- « P » : indique la nécessité de codage pour une sécurité de bout en bout.
- « M » : indique si le support d'AVP est requis.
  - M=1 : si le message AVP ou sa valeur n'est pas reconnu par un client Diameter, un serveur, un proxy, ou un translation agent, il doit être rejeté.
  - M=0 : information seulement.
- « V » : indique si le champ optionnel vendor ID est présent dans le header AVP.

AVP length : indique la taille de l'AVP au total (header + data)

Pour permettre la convergence des protocoles, et permettre ainsi d'ouvrir des sessions complexes, on se sert des AVPs, qui contiennent les données des messages du protocole Diameter.

Ces AVPs, quels qu'ils soient, sont constitués de champs tels que leur code (qui les identifie), leur longueur et des flags indiquant au destinataire comment ils

doivent être traités. Ils sont aussi constitués de données qui varient d'un AVP à l'autre : identity, value.

Le but d'avoir fait l'analyse de ce protocole est de pouvoir identifier les informations transportées dans les AVP, et qui interagissent entre les composants de contrôle d'IMS comme le I-CSCF et le S-CSCF avec le HSS. Ceci nous permettra par la suite de modifier ou d'ajouter des informations décisionnelles au niveau des AVP, dans le but de compléter des informations que nous considérons comme manquantes dans le HSS d'aujourd'hui et qui nous permettra de représenter le réel.

### III.2.3 Architecture informationnelle d'IMS (HSS)

Dans cette thèse, nous travaillons sur la dimension informationnelle qui dans l'architecture IMS se trouve au niveau du HSS (Home Subscriber Server) [12]. Nous avons fait une analyse des manques que l'on en a trouvé sur la dimension informationnelle, afin de faire des propositions essayant de répondre aux besoins identifiés dans la section précédente qui sont d'offrir le service à l'utilisateur « anywhere », « anytime », « anyhow », « anyservice » de façon transparente pour l'utilisateur.

Le HSS est la base de données de l'opérateur. Il est l'entité contenant l'information de la souscription des utilisateurs.

Un opérateur peut avoir une ou plusieurs HSSs en fonction du nombre d'abonnés, de la capacité des éléments du réseau, et de l'organisation du réseau.

Le HSS offre un soutien aux serveurs de contrôle d'appels afin de compléter le routage / procédures de « roaming » en résolvant l'authentification, l'autorisation de nommage / adressage de résolution, les dépendances de localisation, etc.

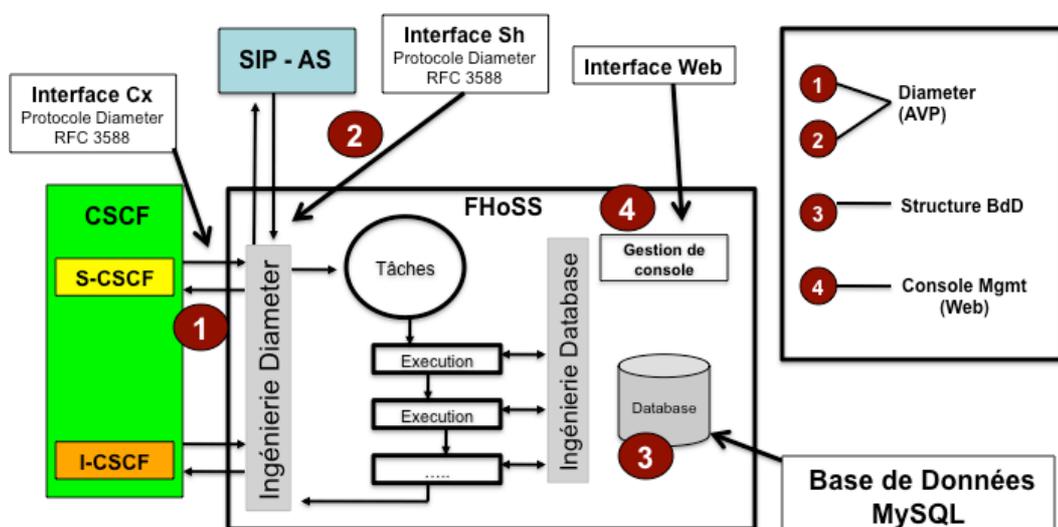


Figure III.14 HSS d'IMS de Fokus

Le HSS est chargé de contenir les informations des utilisateurs comme :

- Identification de l'utilisateur, la numérotation et des informations d'adressage.
- Information au niveau de la sécurité : Information du contrôle d'accès du réseau pour l'authentification et l'autorisation.
- Informations de la localisation de l'utilisateur au niveau inter-système: le HSS permet l'enregistrement de l'utilisateur, et il stocke les informations sur la localisation inter-systèmes, etc.
- Information au niveau profil de l'utilisateur.

Comme il a été décrit sur la section III.2.2, pour accéder à la base de données on a besoin du protocole Diameter, en utilisant les AVPs décrites. Ces AVPs, sont les moyens pour accéder à la base de données du HSS.

Voici les interactions entre le composant a) I-CSCF et HSS ainsi que le b) S-CSCF et HSS et entre le c) AS et HSS.

#### **a) Interactions entre : I-CSCF et HSS**

UAR/UAA (**U**ser **A**uthentication **R**quest / **U**ser **A**uthentication **A**nswer)

Détermine si l'utilisateur est autorisé à recevoir un service, et si oui, il indique le S-CSCF capable de fournir le service.

- LIR/LIA (**L**ocation **I**nfo **R**quest / **L**ocation **I**nfo **A**nswer)

Il indique le S-CSCF en charge de l'utilisateur

#### **b) Interactions entre : S-CSCF et HSS**

- PPR/PPA (**P**ush **P**rofile **R**quest / **P**ush **P**rofile **A**nswer)

Délivre de façon asynchrone le profil de l'utilisateur au serveur SIP à partir du HSS.

- MAR/MAA (**M**ultimedia **A**uth **R**quest / **M**ultimedia **A**uth **A**nswer)

Authentifie et autorise un utilisateur pour un service en spécifique.

- SAR/SAA (**S**erver **A**ssignment **R**quest / **S**erver **A**ssignment **A**nswer)

Il fournit un serveur spécifique pour un utilisateur et délivre le profil de l'utilisateur de façon synchrone.

- RTR/RTA (**R**egistration **T**ermination **R**quest / **R**egistration **T**ermination **A**nswer). Demande de terminaison de l'enregistrement.

#### **c) HSS - AS**

- UDR/UDA (**U**ser **D**ata **R**quest / **U**ser **D**ata **A**nswer)

Il fait la demande des informations de l'utilisateur.

- SNR/SNA (**S**ubscribe **N**otification **R**quest / **S**ubscribe **N**otification **A**nswer)

Il fait la demande les notifications de changements des informations de l'utilisateur.

- PUR/PUA (Profile Update Request / Profile Update Answer)

Il fait la mise à jour des informations de l'utilisateur.

- PNR/PNA (Push Notification Request / Push Notification Answer)

Il fait la notification de changements dans le AS.

## Manques du HSS de Fokus

Nous avons fait une implémentation du Cœur IMS en utilisant l'Open Source de Fokus. Lors de l'enregistrement d'une session IMS, on doit réaliser la procédure d'authentification de l'utilisateur et obtenir les informations concernant le service qu'il est autorisé à utiliser. C'est le protocole DIAMETER, qui est utilisé avec des AVP (Attribute Value Pairs) pour les extensions.

La plupart des informations sont stockées dans le HSS et certaines le sont dans l'équipement de l'utilisateur (UE) pour gérer sa mobilité et sa localisation.

Pour plus de détail, nous avons étudié l'implémentation IMS Open Source de l'institut Allemand FOKUS qui est installée sur la plate-forme de services à Télécom ParisTech.

Le HSS est donc une base de données qui contient les informations concernant l'utilisateur à travers un ensemble de tables de base de données (Figure III.15) pour fournir entre autre : le profil de service, ses identifiants, sa localisation, les services autorisés, l'adresse AS, les IFC, etc.

U	R	S	T																																																																				
User	Réseau	Service	Terminal																																																																				
<b>ALIASES_REPOSITORY_DATA</b> <span style="float: right;">S</span>																																																																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>P * ID</td><td>NUMBER (10)</td></tr> <tr><td>* SQN</td><td>NUMBER (10)</td></tr> <tr><td>* ID_IMPLICIT_SET</td><td>NUMBER (10)</td></tr> <tr><td>* SERVICE_INDICATION</td><td>VARCHAR2 (255 CHAR)</td></tr> <tr><td>REP_DATA</td><td>BLOB</td></tr> </table>				P * ID	NUMBER (10)	* SQN	NUMBER (10)	* ID_IMPLICIT_SET	NUMBER (10)	* SERVICE_INDICATION	VARCHAR2 (255 CHAR)	REP_DATA	BLOB																																																										
P * ID	NUMBER (10)																																																																						
* SQN	NUMBER (10)																																																																						
* ID_IMPLICIT_SET	NUMBER (10)																																																																						
* SERVICE_INDICATION	VARCHAR2 (255 CHAR)																																																																						
REP_DATA	BLOB																																																																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>PRIMARY_19</td><td></td></tr> <tr><td>◇ IDX_ID_IMPLICIT_SET</td><td></td></tr> <tr><td>◇ IDX_SQN</td><td></td></tr> </table>				PRIMARY_19		◇ IDX_ID_IMPLICIT_SET		◇ IDX_SQN																																																															
PRIMARY_19																																																																							
◇ IDX_ID_IMPLICIT_SET																																																																							
◇ IDX_SQN																																																																							
<b>APPLICATION_SERVER</b> <span style="float: right;">U S</span>																																																																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>P * ID</td><td>NUMBER (10)</td></tr> <tr><td>* NAME</td><td>VARCHAR2 (255 CHAR)</td></tr> <tr><td>* SERVER_NAME</td><td>VARCHAR2 (255 CHAR)</td></tr> <tr><td>* DEFAULT_HANDLING</td><td>NUMBER (10)</td></tr> <tr><td>* SERVICE_INFO</td><td>VARCHAR2 (255 CHAR)</td></tr> <tr><td>* DIAMETER_ADDRESS</td><td>VARCHAR2 (255 CHAR)</td></tr> <tr><td>* REP_DATA_SIZE_LIMIT</td><td>NUMBER (10)</td></tr> <tr><td>* UDR</td><td>NUMBER (3)</td></tr> <tr><td>* PUR</td><td>NUMBER (3)</td></tr> <tr><td>* SNR</td><td>NUMBER (3)</td></tr> <tr><td>* UDR_REP_DATA</td><td>NUMBER (3)</td></tr> <tr><td>* UDR_IMPU</td><td>NUMBER (3)</td></tr> <tr><td>* UDR_IMS_USER_STATE</td><td>NUMBER (3)</td></tr> <tr><td>* UDR_SCSCF_NAME</td><td>NUMBER (3)</td></tr> <tr><td>* UDR_IFC</td><td>NUMBER (3)</td></tr> <tr><td>* UDR_LOCATION</td><td>NUMBER (3)</td></tr> <tr><td>* UDR_USER_STATE</td><td>NUMBER (3)</td></tr> <tr><td>* UDR_CHARGING_INFO</td><td>NUMBER (3)</td></tr> <tr><td>* UDR_MSISDN</td><td>NUMBER (3)</td></tr> <tr><td>* UDR_PSL_ACTIVATION</td><td>NUMBER (3)</td></tr> <tr><td>* UDR_DSAI</td><td>NUMBER (3)</td></tr> <tr><td>* UDR_ALIASES_REP_DATA</td><td>NUMBER (3)</td></tr> <tr><td>* PUR_REP_DATA</td><td>NUMBER (3)</td></tr> <tr><td>* PUR_PSL_ACTIVATION</td><td>NUMBER (3)</td></tr> <tr><td>* PUR_DSAI</td><td>NUMBER (3)</td></tr> <tr><td>* PUR_ALIASES_REP_DATA</td><td>NUMBER (3)</td></tr> <tr><td>* SNR_REP_DATA</td><td>NUMBER (3)</td></tr> <tr><td>* SNR_IMPU</td><td>NUMBER (3)</td></tr> <tr><td>* SNR_IMS_USER_STATE</td><td>NUMBER (3)</td></tr> <tr><td>* SNR_SCSCF_NAME</td><td>NUMBER (3)</td></tr> <tr><td>* SNR_IFC</td><td>NUMBER (3)</td></tr> <tr><td>* SNR_PSL_ACTIVATION</td><td>NUMBER (3)</td></tr> <tr><td>* SNR_DSAI</td><td>NUMBER (3)</td></tr> <tr><td>* SNR_ALIASES_REP_DATA</td><td>NUMBER (3)</td></tr> </table>				P * ID	NUMBER (10)	* NAME	VARCHAR2 (255 CHAR)	* SERVER_NAME	VARCHAR2 (255 CHAR)	* DEFAULT_HANDLING	NUMBER (10)	* SERVICE_INFO	VARCHAR2 (255 CHAR)	* DIAMETER_ADDRESS	VARCHAR2 (255 CHAR)	* REP_DATA_SIZE_LIMIT	NUMBER (10)	* UDR	NUMBER (3)	* PUR	NUMBER (3)	* SNR	NUMBER (3)	* UDR_REP_DATA	NUMBER (3)	* UDR_IMPU	NUMBER (3)	* UDR_IMS_USER_STATE	NUMBER (3)	* UDR_SCSCF_NAME	NUMBER (3)	* UDR_IFC	NUMBER (3)	* UDR_LOCATION	NUMBER (3)	* UDR_USER_STATE	NUMBER (3)	* UDR_CHARGING_INFO	NUMBER (3)	* UDR_MSISDN	NUMBER (3)	* UDR_PSL_ACTIVATION	NUMBER (3)	* UDR_DSAI	NUMBER (3)	* UDR_ALIASES_REP_DATA	NUMBER (3)	* PUR_REP_DATA	NUMBER (3)	* PUR_PSL_ACTIVATION	NUMBER (3)	* PUR_DSAI	NUMBER (3)	* PUR_ALIASES_REP_DATA	NUMBER (3)	* SNR_REP_DATA	NUMBER (3)	* SNR_IMPU	NUMBER (3)	* SNR_IMS_USER_STATE	NUMBER (3)	* SNR_SCSCF_NAME	NUMBER (3)	* SNR_IFC	NUMBER (3)	* SNR_PSL_ACTIVATION	NUMBER (3)	* SNR_DSAI	NUMBER (3)	* SNR_ALIASES_REP_DATA	NUMBER (3)
P * ID	NUMBER (10)																																																																						
* NAME	VARCHAR2 (255 CHAR)																																																																						
* SERVER_NAME	VARCHAR2 (255 CHAR)																																																																						
* DEFAULT_HANDLING	NUMBER (10)																																																																						
* SERVICE_INFO	VARCHAR2 (255 CHAR)																																																																						
* DIAMETER_ADDRESS	VARCHAR2 (255 CHAR)																																																																						
* REP_DATA_SIZE_LIMIT	NUMBER (10)																																																																						
* UDR	NUMBER (3)																																																																						
* PUR	NUMBER (3)																																																																						
* SNR	NUMBER (3)																																																																						
* UDR_REP_DATA	NUMBER (3)																																																																						
* UDR_IMPU	NUMBER (3)																																																																						
* UDR_IMS_USER_STATE	NUMBER (3)																																																																						
* UDR_SCSCF_NAME	NUMBER (3)																																																																						
* UDR_IFC	NUMBER (3)																																																																						
* UDR_LOCATION	NUMBER (3)																																																																						
* UDR_USER_STATE	NUMBER (3)																																																																						
* UDR_CHARGING_INFO	NUMBER (3)																																																																						
* UDR_MSISDN	NUMBER (3)																																																																						
* UDR_PSL_ACTIVATION	NUMBER (3)																																																																						
* UDR_DSAI	NUMBER (3)																																																																						
* UDR_ALIASES_REP_DATA	NUMBER (3)																																																																						
* PUR_REP_DATA	NUMBER (3)																																																																						
* PUR_PSL_ACTIVATION	NUMBER (3)																																																																						
* PUR_DSAI	NUMBER (3)																																																																						
* PUR_ALIASES_REP_DATA	NUMBER (3)																																																																						
* SNR_REP_DATA	NUMBER (3)																																																																						
* SNR_IMPU	NUMBER (3)																																																																						
* SNR_IMS_USER_STATE	NUMBER (3)																																																																						
* SNR_SCSCF_NAME	NUMBER (3)																																																																						
* SNR_IFC	NUMBER (3)																																																																						
* SNR_PSL_ACTIVATION	NUMBER (3)																																																																						
* SNR_DSAI	NUMBER (3)																																																																						
* SNR_ALIASES_REP_DATA	NUMBER (3)																																																																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>◇ IDX_DIAMETER_ADDRESS</td><td></td></tr> <tr><td>◇ IDX_SERVER_NAME</td><td></td></tr> <tr><td>◇ IDX_NAME_7</td><td></td></tr> <tr><td>◇ PRIMARY_20</td><td></td></tr> </table>				◇ IDX_DIAMETER_ADDRESS		◇ IDX_SERVER_NAME		◇ IDX_NAME_7		◇ PRIMARY_20																																																													
◇ IDX_DIAMETER_ADDRESS																																																																							
◇ IDX_SERVER_NAME																																																																							
◇ IDX_NAME_7																																																																							
◇ PRIMARY_20																																																																							
<b>IFC</b> <span style="float: right;">U S</span>																																																																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>P * ID</td><td>NUMBER (10)</td></tr> <tr><td>* NAME</td><td>VARCHAR2 (255 CHAR)</td></tr> <tr><td>ID_APPLICATION_SERVER</td><td>NUMBER (10)</td></tr> <tr><td>ID_TP</td><td>NUMBER (10)</td></tr> <tr><td>PROFILE_PART_IND</td><td>NUMBER (10)</td></tr> </table>				P * ID	NUMBER (10)	* NAME	VARCHAR2 (255 CHAR)	ID_APPLICATION_SERVER	NUMBER (10)	ID_TP	NUMBER (10)	PROFILE_PART_IND	NUMBER (10)																																																										
P * ID	NUMBER (10)																																																																						
* NAME	VARCHAR2 (255 CHAR)																																																																						
ID_APPLICATION_SERVER	NUMBER (10)																																																																						
ID_TP	NUMBER (10)																																																																						
PROFILE_PART_IND	NUMBER (10)																																																																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>◇ IDX_NAME_1</td><td></td></tr> <tr><td>◇ PRIMARY_1</td><td></td></tr> </table>				◇ IDX_NAME_1		◇ PRIMARY_1																																																																	
◇ IDX_NAME_1																																																																							
◇ PRIMARY_1																																																																							
<b>IMPI</b> <span style="float: right;">U</span>																																																																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>P * ID</td><td>NUMBER (10)</td></tr> <tr><td>ID_IMSU</td><td>NUMBER (10)</td></tr> <tr><td>* IDENTITY</td><td>VARCHAR2 (255 CHAR)</td></tr> <tr><td>* K</td><td>BLOB</td></tr> <tr><td>* AUTH_SCHEME</td><td>NUMBER (10)</td></tr> <tr><td>* DEFAULT_AUTH_SCHEME</td><td>NUMBER (10)</td></tr> <tr><td>* AMF</td><td>BLOB</td></tr> <tr><td>* OP</td><td>BLOB</td></tr> <tr><td>* SQN</td><td>VARCHAR2 (64 CHAR)</td></tr> <tr><td>* IP</td><td>VARCHAR2 (64 CHAR)</td></tr> <tr><td>* LINE_IDENTIFIER</td><td>VARCHAR2 (64 CHAR)</td></tr> <tr><td>ZH_UICC_TYPE</td><td>NUMBER (10)</td></tr> <tr><td>ZH_KEY_LIFE_TIME</td><td>NUMBER (10)</td></tr> <tr><td>* ZH_DEFAULT_AUTH_SCHEME</td><td>NUMBER (10)</td></tr> </table>				P * ID	NUMBER (10)	ID_IMSU	NUMBER (10)	* IDENTITY	VARCHAR2 (255 CHAR)	* K	BLOB	* AUTH_SCHEME	NUMBER (10)	* DEFAULT_AUTH_SCHEME	NUMBER (10)	* AMF	BLOB	* OP	BLOB	* SQN	VARCHAR2 (64 CHAR)	* IP	VARCHAR2 (64 CHAR)	* LINE_IDENTIFIER	VARCHAR2 (64 CHAR)	ZH_UICC_TYPE	NUMBER (10)	ZH_KEY_LIFE_TIME	NUMBER (10)	* ZH_DEFAULT_AUTH_SCHEME	NUMBER (10)																																								
P * ID	NUMBER (10)																																																																						
ID_IMSU	NUMBER (10)																																																																						
* IDENTITY	VARCHAR2 (255 CHAR)																																																																						
* K	BLOB																																																																						
* AUTH_SCHEME	NUMBER (10)																																																																						
* DEFAULT_AUTH_SCHEME	NUMBER (10)																																																																						
* AMF	BLOB																																																																						
* OP	BLOB																																																																						
* SQN	VARCHAR2 (64 CHAR)																																																																						
* IP	VARCHAR2 (64 CHAR)																																																																						
* LINE_IDENTIFIER	VARCHAR2 (64 CHAR)																																																																						
ZH_UICC_TYPE	NUMBER (10)																																																																						
ZH_KEY_LIFE_TIME	NUMBER (10)																																																																						
* ZH_DEFAULT_AUTH_SCHEME	NUMBER (10)																																																																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>◇ IDX_IDENTITY_1</td><td></td></tr> <tr><td>◇ IDX_ID_IMSU</td><td></td></tr> <tr><td>◇ PRIMARY_2</td><td></td></tr> </table>				◇ IDX_IDENTITY_1		◇ IDX_ID_IMSU		◇ PRIMARY_2																																																															
◇ IDX_IDENTITY_1																																																																							
◇ IDX_ID_IMSU																																																																							
◇ PRIMARY_2																																																																							

50 Modèle Informationnel pour une session User-Centric : Inférence et Intégration Intelligente

CAPABILITIES_SET		S
P *	ID	NUMBER (10)
*	ID_SET	NUMBER (10)
*	NAME	VARCHAR2 (255 CHAR)
*	ID_CAPABILITY	NUMBER (10)
*	IS_MANDATORY	NUMBER (10)
	PRIMARY_21	
	IDX_CAPABILITY	
	IDX_ID_SET	
	IDX_NAME_8	

CAPABILITY		S
P *	ID	NUMBER (10)
*	NAME	VARCHAR2 (255 CHAR)
	IDX_NAME_9	
	PRIMARY_22	

PREFERRED_SCSCF_SET		S
P *	ID	NUMBER (10)
*	ID_SET	NUMBER (10)
*	NAME	VARCHAR2 (255 CHAR)
*	SCSCF_NAME	VARCHAR2 (255 CHAR)
*	PRIORITY	NUMBER (10)
	IDX_PRIORITY_1	
	IDX_SET	
	IDX_NAME_3	
	PRIMARY_7	

CHARGING_INFO		S
P *	ID	NUMBER (10)
*	NAME	VARCHAR2 (255 CHAR)
*	PRI_ECF	VARCHAR2 (255 CHAR)
*	SEC_ECF	VARCHAR2 (255 CHAR)
*	PRI_CCF	VARCHAR2 (255 CHAR)
*	SEC_CCF	VARCHAR2 (255 CHAR)
	PRIMARY_23	
	IDX_NAME	

REPOSITORY_DATA		S
P *	ID	NUMBER (10)
*	SQN	NUMBER (10)
*	ID_IMPUI	NUMBER (10)
*	SERVICE_INDICATION	VARCHAR2 (255 CHAR)
	REP_DATA	BLOB
	IDX_ID_IMPUI_4	
	IDX_SQN_1	
	PRIMARY_8	

IMPI_IMPUI		U
P *	ID	NUMBER (10)
*	ID_IMPI	NUMBER (10)
*	ID_IMPUI	NUMBER (10)
*	USER_STATE	NUMBER (3)
	PRIMARY_3	
	IDX_ID_IMPI	
	IDX_ID_IMPUI_2	

DSAI		S
P *	ID	NUMBER (10)
*	DSAI_TAG	VARCHAR2 (255 CHAR)
	PRIMARY_25	

SH_NOTIFICATION		S
P *	ID	NUMBER (10)
*	ID_IMPUI	NUMBER (10)
*	ID_APPLICATION_SERVER	NUMBER (10)
*	DATA_REF	NUMBER (10)
	REP_DATA	BLOB
	SON	NUMBER (10)
	SERVICE_INDICATION	VARCHAR2 (255 CHAR)
	ID_IFC	NUMBER (10)
	SERVER_NAME	VARCHAR2 (255 CHAR)
	SCSCF_NAME	VARCHAR2 (255 CHAR)
	REG_STATE	NUMBER (10)
	PSI_ACTIVATION	NUMBER (10)
	DSAI_TAG	VARCHAR2 (255 CHAR)
	DSAI_VALUE	NUMBER (10)
	HOPBYHOP	NUMBER (24)
	ENDTOEND	NUMBER (24)
*	GRP	NUMBER (10)
	IDX_ID_IMPUI_5	
	IDX_AS	
	PRIMARY_9	

CX_EVENTS		U
P *	ID	NUMBER (10)
	HOPBYHOP	NUMBER (24)
	ENDTOEND	NUMBER (24)
	ID_IMPUI	NUMBER (10)
	ID_IMPI	NUMBER (10)
	ID_IMPLICIT_SET	NUMBER (10)
*	TYPE	NUMBER (3)
*	SUBTYPE	NUMBER (3)
	GRP	NUMBER (10)
	REASON_INFO	VARCHAR2 (255 CHAR)
	TRIALS_CNT	NUMBER (10)
	DIAMETER_NAME	VARCHAR2 (255 CHAR)
	IDX_HOPBYHOP	
	IDX_ENDTOEND	
	IDX_TYPE	
	IDX_GRP	
	PRIMARY_24	

IMSU		U S
P *	ID	NUMBER (10)
*	NAME	VARCHAR2 (255 CHAR)
	SCSCF_NAME	VARCHAR2 (255 CHAR)
	DIAMETER_NAME	VARCHAR2 (255 CHAR)
	ID_CAPABILITIES_SET	NUMBER (10)
	ID_PREFERRED_SCSCF_SET	NUMBER (10)
	PRIMARY_6	
	IDX_CAPABILITIES_SET	
	IDX_PREFERRED_SCSCF	
	IDX_NAME_2	

VISITED_NETWORK		R
P *	ID	NUMBER (10)
*	IDENTITY	VARCHAR2 (255 CHAR)
	PRIMARY_17	
	IDX_IDENTITY	

SP_IFC		S
P *	ID	NUMBER (10)
*	ID_SP	NUMBER (10)
*	ID_IFC	NUMBER (10)
*	PRIORITY	NUMBER (10)
	PRIMARY_13	
	ID_SP	
	ID_IFC	
	IDX_PRIORITY	

S_CSCF		S
P *	ID	NUMBER (10)
*	NAME	VARCHAR2 (83 CHAR)
*	S_CSCF_URI	VARCHAR2 (83 CHAR)
	PRIMARY_27	

NDS_TRUSTED_DOMAINS		R
P *	ID	NUMBER (10)
*	TRUSTED_DOMAIN	VARCHAR2 (83 CHAR)
	PRIMARY	

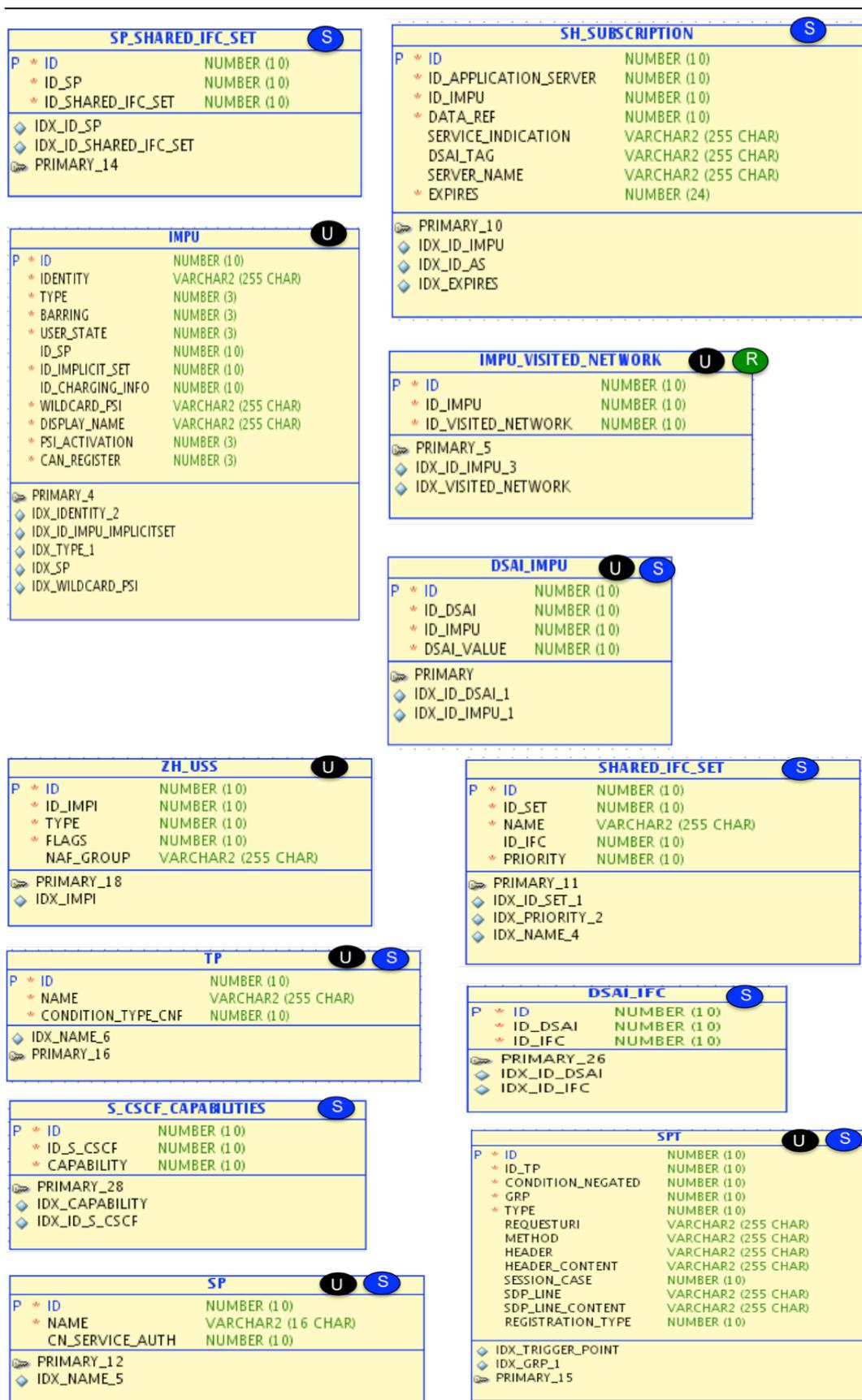


Figure III.15 HSS de Fokus actuel

Comme il a été présenté sur la Figure III.14, le HSS contient des informations à différents niveaux, il peut contenir des informations au niveau d'utilisateur « U », au niveau terminal « T », réseau « R », et service « S ». Voici la description de ces informations :

- Le profil de service qui est associé à chaque utilisateur, table SVP (SerVice Profile).
- Les identifiants qui représentent les adresses publiques et privées de l'utilisateur avec les informations d'authentification et autorisation (table IMPI<sup>25</sup>), d'enregistrement (table IMPU<sup>26</sup>), de souscription (table IMSU<sup>27</sup>) et de tarification (table Chginfo<sup>28</sup>).
- La localisation qui représente le réseau dans lequel l'utilisateur est abonné (Table Networks). Nous avons aussi les réseaux avec lesquels des accords de roaming sont passés (Table Roam).
- Les services autorisés qui identifient les services auxquels l'utilisateur est abonné, (Table as\_perm\_list).
- L'adresse de l'AS à contacter si les trigger points sont conformes. Des informations sur le comportement par défaut de la session sont données si le contact avec l'AS a échoué (table APSVR<sup>29</sup>).
- Les IFC<sup>30</sup> qui représente les informations de filtrage initial, caractérisant un certain service pour l'utilisateur (table IFC). Cette table fait correspondre un trigger point (table Trigpt) qui associe un ensemble de service point trigger (table spt) à un serveur d'application.

Après une première analyse de ces tables du HSS, on constate que, nous n'avons pas des informations sur la qualité de service, ni sur les préférences de l'utilisateur, ni sur les différents équipements de l'utilisateur. Ces tables qui stockent les informations sont donc insuffisantes pour garantir le service demandé avec la qualité de service souhaité en tenant compte de la mobilité et de la continuité de service.

Les informations dans le HSS sont très limitées et ne sont pas assez dynamiques pour offrir des informations en temps réel sur les ressources ambiantes lorsque l'utilisateur se déplace. Par exemple dans le profil de service de l'utilisateur, on ne peut pas trouver les informations concernant ses préférences et les terminaux

---

<sup>25</sup>IMPI : IP Multimédia Private Identity

<sup>26</sup> IMPU : IMS Public User identity

<sup>27</sup> IMSU : IMS SUbscription

<sup>28</sup> Chginfo : Charging information

<sup>29</sup> APSVR : Application Server

<sup>30</sup> IFC : Initial Filter Criteria

disponibles. Cette information est importante pour pouvoir faire le choix du moyen de communication à utiliser pour accéder au service.

Comme nous avons constaté sur l'image III.14, nous n'avons pas des informations concernant le terminal des utilisateurs, ni des informations de QoS qui représentent les ressources qui participent dans la session de l'utilisateur. C'est pour cela que nous devons améliorer cette structure informationnelle, afin de compter avec les informations adéquates lorsque l'on en aura besoin.

Dans le Tableau III.1 nous exposons les fonctions principales identifiées ainsi que ses carences informationnelles au niveau du HSS :

Besoins fonctionnels	HSS dans IMS	Carences
Joignabilité	Adresse SIP, identifiants de services	Pas de préférences de l'utilisateur pour la QoS, pour la tarification, etc.
Authentification et Autorisation (AA)	Table IMPI	Pas d'authentification pour une session comportant plusieurs services
Enregistrement	Table IMPU	Pas d'information sur la nature du moyen de communication
Exploitation de la Session : Adaptation dynamique		Pas d'informations liées à la mobilité (présence de terminaux et de services en fonction de la localisation de l'utilisateur)

Tableau III.1 *Les manques de HSS*

Afin de palier ces carences informationnelles contenues dans le HSS d'IMS, il faudrait structurer les informations de façon efficace pour avoir l'information décisionnelle, ceci permettrait avoir la connaissance du comportement des ressources. Le fait de connaître le comportement de la ressource nous permet de mieux choisir la ressource à utiliser lorsque l'utilisateur accède à son service. C'est pourquoi, nous regardons dans la section III.3 des travaux qui concernent la gestion de la connaissance afin que l'on puisse proposer une solution la mieux adaptée par rapport aux besoins exprimés auparavant.

### **III.3 De la gestion de l'information à la gestion de la connaissance (Knowledge Management)**

Comme il a été indiqué auparavant dans le contexte, nous devons savoir et comprendre le processus nécessaire pour passer de l'information vers la connaissance. Ceci permettra de créer de nouvelles façons pour concevoir des systèmes qui se gèrent de façon autonome. C'est pour cette raison que nous allons regarder tout d'abord le processus du passage de données vers la connaissance (§III.3.1). Ensuite, nous regarderons ces aspects mais au niveau des systèmes

d'information, ce qui est connu comme SGC pour Système de Gestion de Connaissances (§III.3.2). Par la suite, nous approfondirons cette approche vis-à-vis les bases de données (§III.3.3), enfin, nous regarderons les différentes approches qui existent autour des SGC (§III.3.4).

### **III.3.1 Passage des données à la connaissance**

La gestion des connaissances (GC) [13] est la gestion explicite des connaissances essentielles et des informations détenues par des particuliers de telle sorte qu'elles soient réellement partagées et utilisées par d'autres dans une organisation ou système.

Pour les philosophes (Aristote, Descartes, etc..), la connaissance est divisée dans 2 domaines très importantes :

**1) Métaphysique :** Une branche de la philosophie qui cherche fondamentalement à expliquer la nature de l'être ou de la réalité (ontologie) et ainsi que l'origine de la structure du monde (cosmologie).

**2) Epistémologie:** Elle se consacre à l'étude de l'origine, la nature, les méthodes et les limites de la connaissance (représentation du monde).

Donc, d'après ces philosophes, « *la connaissance* » est de nature épistémologique, mais nécessite la disponibilité d'une ontologie.

La *connaissance* peut être cataloguée dans 2 types :

**1) Connaissance Explicite :** Elle est formalisée et inclut des procédures, des modèles, des algorithmes, des documents de toutes sortes. Elle est hétérogène, incomplète ou redondante, et très souvent ne comprend pas suffisamment de contexte (par exemple la conception logique).

**2) Connaissance Tacite :** Elle est acquise par la pratique, et est souvent transmise par un maître apprenti. Elle est située dans la personne qui la détient.

La connaissance tacite peut ou pas être de type explicite.

La question que nous devons nous poser est : Comment sais-je que je suis arrivé à la connaissance de quelque chose ? Comment passons-nous des données à la connaissance ?

Nous pouvons déjà commencer par définir les termes données, information, et connaissance.

**Données :** Base de l'interprétation, la chose

- Statistiques : Résultat d'une recherche
- Informatique : Représentation d'une information élémentaire

**Information :** Action d'informer ou de s'informer, recherche préalable à l'instruction, élément de base de l'informatique

- Journalisme : Nouvelle communiquée par les médias

---

**Connaissance** : Capacité de se représenter, façon de percevoir, ensemble des choses connues, du savoir.

Mais, maintenant que nous avons le significat, nous pouvons nous poser la question vis-à-vis du dictionnaire informatique :

**Données** : Informations utilisées par un logiciel. Elles peuvent être créées par l'utilisateur ou par le programme lui-même.

**Information** : Faits et connaissances déduits des données. L'ordinateur manipule et génère des données. La signification déduite des données est l'information : c'est-à-dire que l'information est une conséquence des données. Les deux mots ne sont pas synonymes bien qu'ils soient souvent utilisés l'un pour l'autre.

**Connaissance** : (Knowledge Base), Partie d'un système expert contenant l'ensemble des informations, en particulier des règles et des faits qui constituent le domaine de compétence du système.

Afin de mieux comprendre comment se fait le passage de données à la connaissance, je prends en considération les travaux du philosophe Russell Ackoff qui a aussi classé ces termes.

**Données** : Symboles, Il existe tout simplement et n'a pas d'importance au-delà de son existence.

**Information** : Données qui sont traités pour être utiles; répond aux questions «qui», «quoi», «où», ou «quand».

**Connaissance** : Collection des informations; Connaissance est acquise lorsque quelqu'un mémorise de l'information; elle répond à la question « Comment »

**Compréhension** : C'est le processus par lequel on peut prendre connaissance et synthétiser de nouvelles connaissances à partir de la connaissance déjà acquise.

Appréciation du «Pourquoi». La différence entre «Connaissance» et «Compréhension» est la différence entre «mémoriser» et «apprendre».

**Sagesse** : Evaluate la compréhension, pourquoi il est (il n'est pas) important, il distingue entre le bon et le mauvais, et entre le bien et le mal.

Les premières **4 catégories** concernent le passé, c.-à-d. ce qui a été ou ce qui est connu. La 5<sup>ème</sup> catégorie traite l'avenir (future) car elle implique la vision et la conception. Avec « Sagesse », les personnes peuvent créer le futur au lieu de comprendre le présent et le passé.

Nous pouvons donner un exemple du passage des données à la connaissance (Fig. III.2.1).

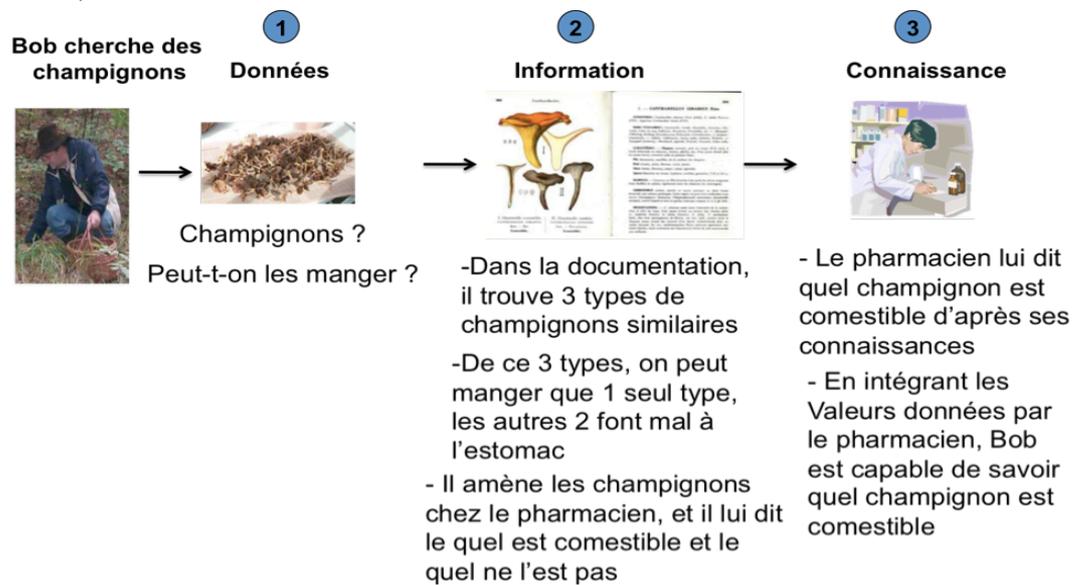


Figure III.16 : Exemple du passage de données à la connaissance

Dans cet exemple des champignons, le pharmacien transmet à Bob, les informations nécessaires pour reconnaître les champignons comestibles, donc on pourrait dire que la *donnée contextualisée* donne l'information. L'information intégrée par une personne dans son système de valeurs devient de la connaissance.

Donc, cela voudrait dire que, ce qui est échangeable entre 2 personnes (systèmes) est toujours de l'information. Et la connaissance ? Reste-t-elle toujours propres à une personne (système) ? Afin de dire que l'on compte avec une Base de Connaissance :

- Il faut savoir transférer dans une machine un système de valeurs et de jugements.
- Ainsi que de l'information fortement structurée, fortement contextualisée, et présentée de manière sensible à l'utilisateur du système de façon à ce qu'il se l'approprie facilement et qu'il en fasse *sa propre connaissance*.
- Pour avoir sa propre connaissance, il faut établir des règles, c-à-d : *Règles d'inférence, communautés, valeurs de QoS, etc...*

C'est cette réflexion qui nous permet d'analyser le passage des données (brutes) à la connaissance dans les systèmes d'Information et de la communication afin de fournir la connaissance du système d'information à l'utilisateur.

---

### **III.3.2 Le passage vers un système de gestion de connaissance (SGC)**

Maintenant que nous connaissons la manière de passer des données à la connaissance, nous devons nous poser la question de comment est il possible de faire ce passage dans les systèmes d'information ? Nous savons bien que les systèmes ne gèrent pas la connaissance, ils gèrent l'information, à moins de fournir un système de valeurs et le mécanisme d'interprétation à la machine.

La fonction principale d'un système de gestion des connaissances (SGC) devrait être celle de rendre l'information disponible aux utilisateurs autorisés. Cette information doit être pertinente afin que l'utilisateur puisse prendre les décisions adéquates en fonction de l'opération qu'il veut mener. Toutefois, la mise en œuvre d'une gestion de l'information de façon simple, l'indexation d'un document et/ou sa recherche, n'est que le début.

Par exemple, obtenir les connaissances de l'entreprise implique plus de logiciels et de technologies, il faudrait plus d'outils au niveau logiciel de gestion de documents, ainsi que des solides compétences pour que l'information qui est produite, soit générée, processée, stockée, distribuée et gérée.

Donc, le système de gestion de connaissances (SGC) se réfère à un système pour la gestion des connaissances dans les organisations, pour soutenir la création, la capture, le stockage et la diffusion de l'information. Il peut comprendre une partie d'une initiative de gestion des connaissances.

Il existe aussi l'approche de systèmes experts (SE) afin de gérer la connaissance. Ces systèmes experts sont apparus dans les années 80. L'avantage de ces systèmes experts est la suivante :

- La connaissance a été modélisée entièrement
- Les systèmes experts ont permis un raisonnement automatique complexe
- Les connaissances acquises ont donc été rendues opérationnelles

Toutefois, cette approche a eu aussi un certain nombre d'inconvénients. Le système expert a assimilé difficilement la connaissance, cela a été long et coûteux. Les SGC ont une autre philosophie, les SGC aident à la prise de décision. Ils conservent des SE que la prise de décision est fondée sur la codification de la connaissance. La codification des connaissances utilise les principes de la représentation des connaissances (partie de grandes idées théoriques de l'ingénierie des connaissances). Généralement une telle codification utilise des règles comme IF-THEN règles pour représenter les implications logiques.

Dans la Figure III.17, nous montrons l'architecture d'un système de gestion de connaissance (SGC). Cette architecture est basée sur trois blocs fonctionnels qui sont : End-User interface (1), Inference engine (2) et Knowledge Base (3).

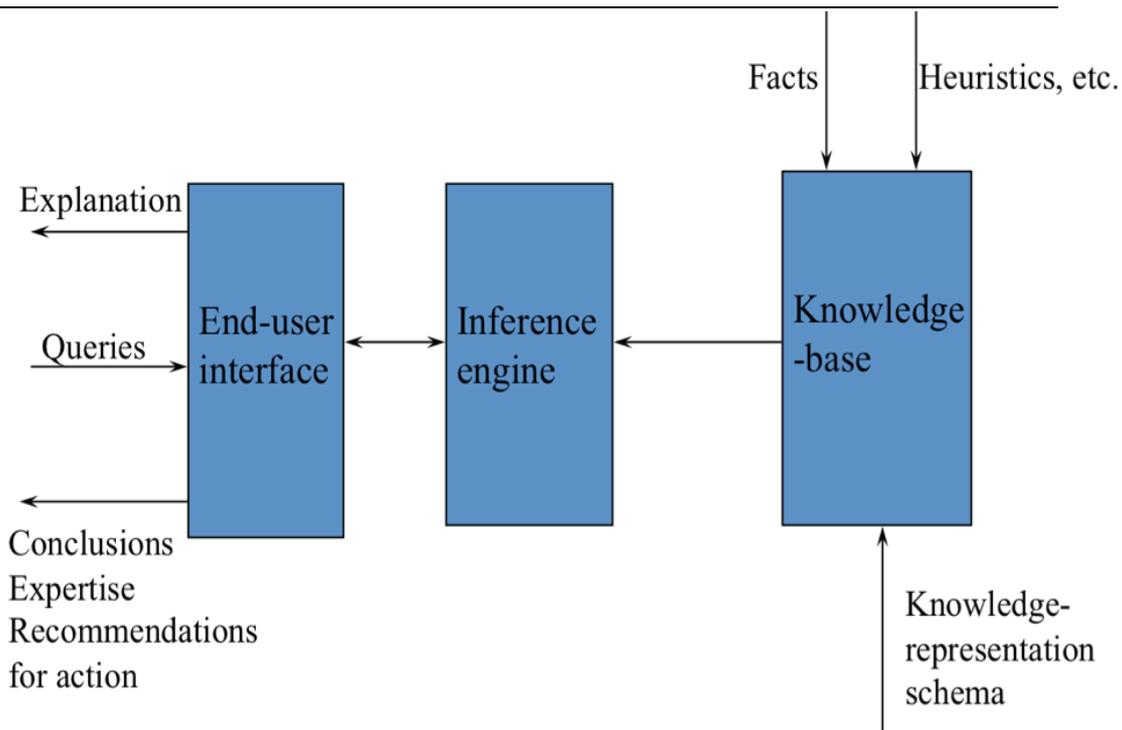


Figure III.17 : Architecture d'un Systèmes de Gestion de Connaissance (SGC)

Le système de gestion de connaissance que nous voulons mettre en place a pour objectif d'offrir à l'utilisateur une visibilité de son système d'information. Celui-ci doit être capable de fournir des informations fiables en temps réel. En effet, des contraintes dues à la mobilité à différents niveaux (utilisateur, terminal, réseau et service) et l'hétérogénéité (terminaux, réseaux et services) peuvent compliquer la tâche. L'utilité de ce système de base de connaissance est de permettre à l'utilisateur de mieux gérer l'accès à ses services et d'avoir une continuité de sa session malgré les fortes contraintes signalées.

Nous faisons l'analyse de travaux existants qui ont été faits afin de palier à cette problématique.

### **III.4 Gestion des événements du système**

Nous avons montré la façon (III.3.3) de comment gérer les connaissances dans les systèmes d'information afin de pouvoir représenter le monde réel. Ce monde réel est en constant mouvement. Nous considérons qu'une approche séquentielle ne peut pas répondre à ces événements du réel et conduirait à une dégradation des performances du système. Nous devons considérer une approche capable de réagir aux différents événements qui pourraient se produire dans le monde réel. C'est pour cette raison que nous avons pris l'approche événementielle. Ce type d'approche permet que les traitements soient plus réactifs et efficaces ce qu'ils

---

correspondent aux besoins NGN d'aujourd'hui. Nous montrons cette architecture orientée événements (III.4.1).

### **III.4.1 Architecture EDA (Event Driven Architecture)**

Nous devons maintenant prendre en compte les aspects de traitement lors d'un événement dans le système de gestion de connaissance. Par conséquent, nous devons choisir l'approche qui nous permet de satisfaire le traitement des événements qui pourraient se présenter dans le monde réel. C'est pour cela, que l'approche de gestion d'événements est nécessaire pour pouvoir assurer l'autonomie du traitement dans notre système de gestion de connaissance (SGC).

En effet, nous trouvons dans la littérature les traitements séquentiels (par procédure), ce qui signifie que les étapes de traitement ont déjà été prédéfinies à l'avance et ont un ordre bien défini. Ce type de traitement séquentiel peut être bloquant vis-à-vis de la problématique que nous traitons dans cette thèse, car les traitements face aux événements sont très importants.

Donc, afin de garantir l'efficacité dans le traitement, il faut s'assurer de l'autonomie des traitements qui vont être faits à chaque niveau de gestion, que ce soit au niveau terminal, réseau ou service. L'autonomie doit en effet avoir des traitements asynchrones, ce qui permet de réagir automatiquement à des événements.

C'est donc l'architecture EDA pour Event Driven Architecture qui est une approche événementielle que nous allons évoquer dans ce chapitre.

L'architecture EDA est basée sur un modèle de communication asynchrone « message-driven » (géré à travers des messages) pour propager l'information dans le système d'information d'une entité. Elle supporte un alignement avec le modèle opérationnel d'une organisation en décrivant les activités « business » à travers des événements contrairement à un modèle de gestion centralisée.

Par conséquent, EDA peut être une forme appropriée pour les entreprises qui ont un environnement distribué et/ou un plus haut degré d'autonomie.

On peut l'associer à une approche SOA mais, le « fournisseur » ne doit pas rendre un service à la demande d'un consommateur, c'est le service qui prévient par émission d'un événement qu'il a réalisé une opération donnée.

La notification de ces événements (Figure III.18) nous permet de notifier un changement de statut d'une ressource et il peut déclencher l'appel d'un ou plusieurs services si nécessaire.

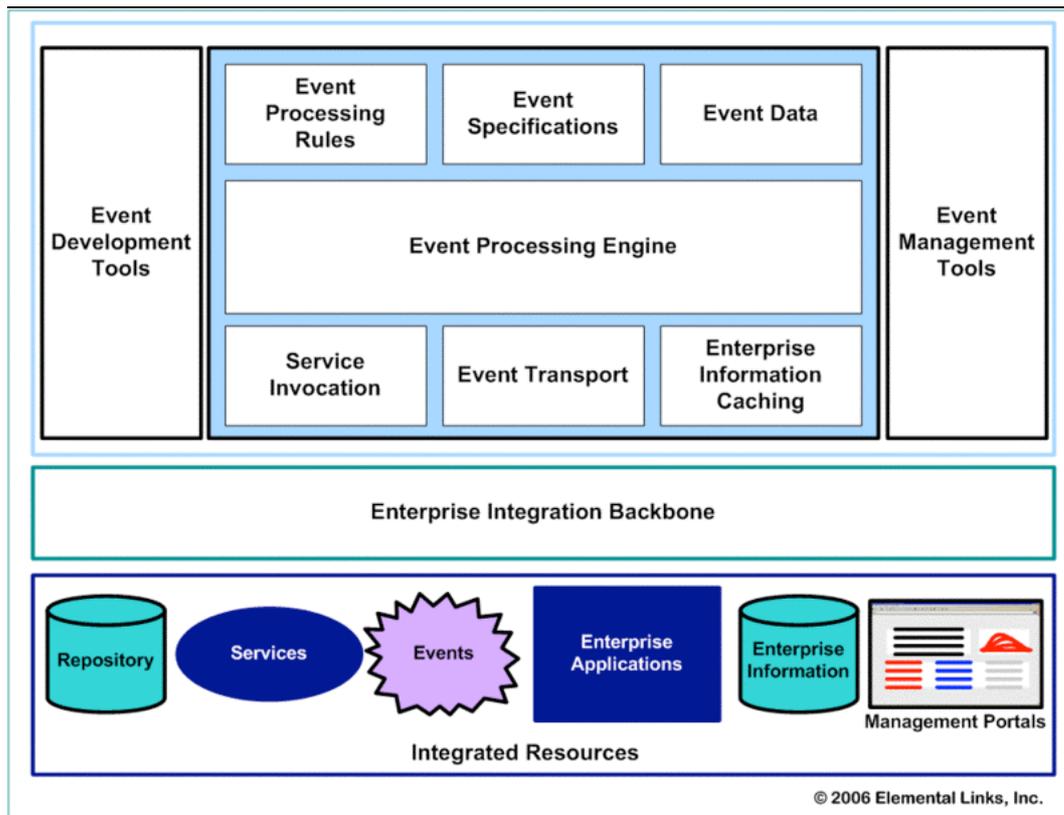
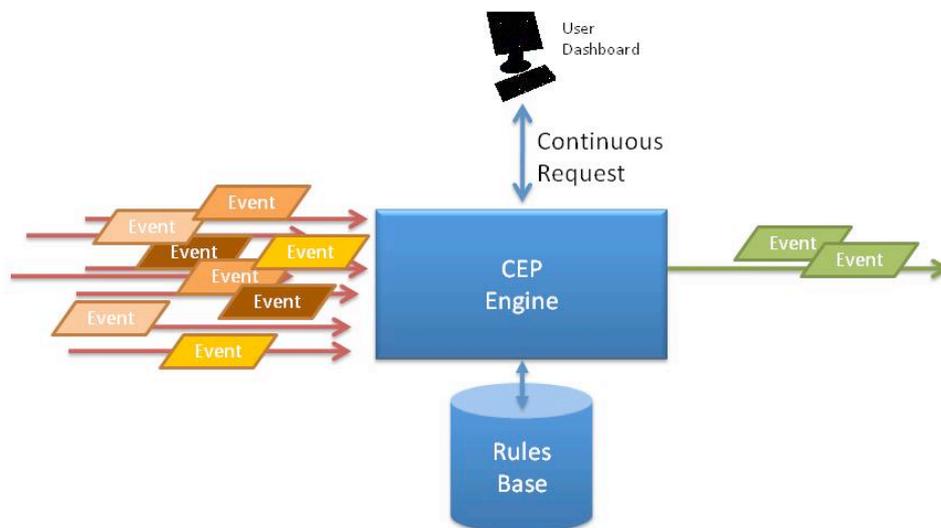


Figure III.18 : Approche par Event Driven Architecture

L'architecture EDA a 3 types de processus pour le traitement des événements. Le premier est « Simple Event Processing », le deuxième est « Stream Event Processing » et le troisième Traitement des événements complexes (Complex Event Processing, CEP).

Le Traitement des événements complexes ou CEP (Figure III.19), est principalement un concept de traitement des événements dans le but d'identifier les événements significatifs dans un nuage d'événements. Le CEP emploie des techniques telles que la détection des schémas complexes de multiples événements: corrélation, abstraction, et hiérarchies entre événements et les relations entre événements tels que les liens de causalité, l'adhésion, la chronologie et les processus pilotés par les événements. La fonction du CEP est de découvrir les informations contenues dans les événements qui passent dans toutes les couches d'une organisation et ensuite d'analyser son impact au niveau macro comme "événement complexe" et puis décider du plan d'action en temps réel.



*Complex Event Processing - Auteur : Patrick Gantet*

Figure III.19 : Approche par Event Driven Architecture, CEP

Nous pouvons en conclure que l'approche par EDA nous permet de prendre en compte les possibles événements qui se produisent dans le réel. Mais qu'en est il de son association avec SOA ?

### **III.5 Sélection de piste de travail**

Afin de réaliser une modélisation, nous avons recours à l'abstraction. Selon l'encyclopédie universalis, elle définit l'abstraction comme « un travail formel, structurant la donnée selon quatre opérations mentales bien distinctes » : simplification (en négligeant toutes circonstances environnant un acte), généralisation (en identifiant des propriétés générales), sélection (en isolant un trait spécifique) et schématisation (en analysant un phénomène et en le restituant comme un système de données). Donc, l'abstraction que nous utilisons est appliquée au monde des télécoms.

Nous présentons donc un méta-modèle NLR « Nœud, Lien, Réseau » (qui nous a permis de représenter le monde réel), ainsi que les niveaux de visibilité qui permettent d'avoir une vision globale de tout le système d'information que l'on veut prendre en compte.

#### **III.5.1 Meta-modèle NLR : Nœud, Lien, Réseau**

Le méta-modèle  $\langle N, L, R \rangle$  (Nœud, Lien, Réseau) permet de décrire tous les éléments qui existent dans le monde réel (Figure III.20). Le concept de  $\langle$ visibilités $\rangle$  permet de représenter le modèle architectural. Selon cette représentation, chaque visibilité constitue un “réseau” de “nœuds” et de “liens” de même nature sur un niveau horizontal, c’est-à-dire du même niveau. C’est ainsi que nous isolons un réseau d’un niveau donné, que ce soit utilisateur, équipement, réseau ou service. L’intégration verticale se fait par la coopération des réseaux sous-jacents.

Un **nœud** est un objet capable d’effectuer des traitements, il peut représenter un commutateur, un ordinateur, un processus, une machine protocolaire, un composant de service, etc.

Un **lien** assure le transfert d’informations entre les nœuds, il représente un câble, une liaison LLC, un circuit virtuel, une association applicative, etc.

Un **réseau** est un ensemble de nœuds et de liens de même nature, coopérant pour fournir de façon transparente un service. Il est identifié par son niveau de visibilité.

La figure III.15 permet de montrer les éléments du méta-modèle NLR.

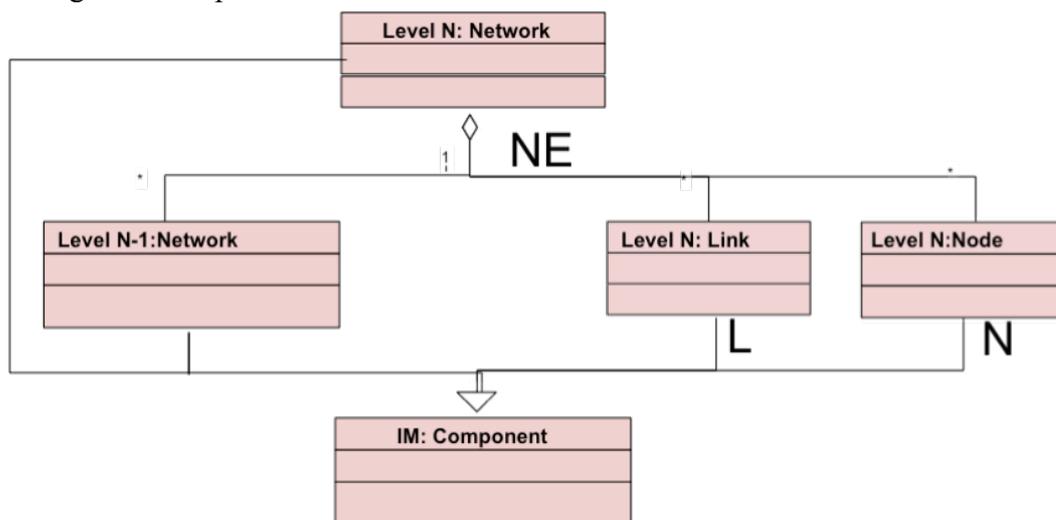


Figure III.20 Méta-modèle NLR

### III.5.2 Niveaux de visibilité

En appliquant le méta-modèle NLN, nous pouvons l’appliquer à ce que l’on appelle les niveaux de visibilité [14] [15] [16], ces éléments permettent de proposer une modélisation axée sur 4 niveaux qui sont :

**Niveau équipement (Fig. III.21, partie 1) :** Ce niveau à caractère matériel rassemble tous les équipements qui entrent en jeu dans la session de l’utilisateur. On pense aux terminaux et « gadgets »/capteurs utilisés par l’utilisateur, aux équipements réseau (routeur, passerelles, etc.) et serveurs d’application.

**Niveau réseau (Fig. III.21, partie 2) :** Il traduit la fonction de transport logique utilisée pour la mise en relation des services. On retrouve la partie réseau coeur (un transport tout IP) et la partie réseaux d'accès (avec toutes les possibilités d'accès possibles fixe/mobile).

**Niveau service (Fig. III.21, partie 3) :** Il regroupe tous les services offerts par et sur le réseau. Définis à des niveaux d'exposition différents par rapport à l'utilisateur, les services sont mis en relation d'une manière flexible et guidés par un processus qui exprime la logique globale du service ordonné par l'utilisateur.

**Niveau utilisateur (Fig. III.21, partie 4) :** Ce niveau concerne tous les acteurs du service (utilisateur final seul, fournisseurs de services, communauté d'utilisateurs partageant les mêmes intérêts). Satisfaire les besoins de l'utilisateur et ses préférences suppose de tenir compte de ses usages du service et de son contexte.

Comme la Figure III.21 peut le montrer, chaque niveau de visibilité représente un réseau (horizontal), qui pourrait être un réseau d'équipement, (Figure III.20, partie 1), un fournisseur de réseau (Figure III.21, partie 2), ou un réseau de service (Figure III.21, partie 3).

A chaque réseau d'un niveau de visibilité, on a la vision du comportement de chaque composant. Ceci aide lorsqu'un composant ne remplit plus son SLA ou la QoS demandée par l'utilisateur, un autre composant, qui lui respecte ce SLA, peut le remplacer. C'est avec l'aide de cette vision de réseaux appliquée à chaque niveau de visibilité, que les préférences de l'utilisateur et la QoS qu'il a demandé peuvent être satisfaits.

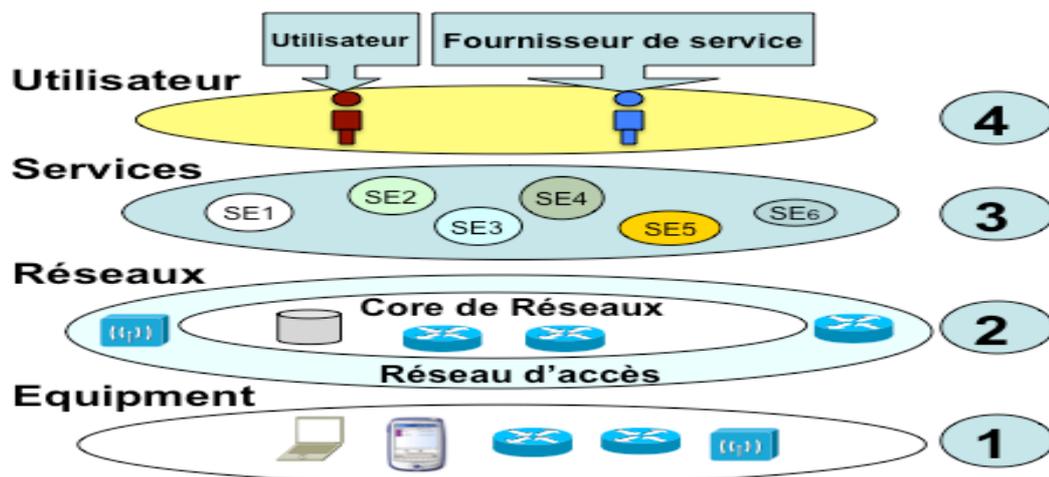


Figure III.21 Niveaux de visibilité

Outre la vue horizontale de cette modélisation, il est possible d'avoir une composition verticale, qui n'est rien d'autre que les éléments d'une session selon les préférences de l'utilisateur. Par exemple, au niveau de l'équipement et dans le réseau du niveau de l'équipement (Figure III.21, partie 1), nous pouvons appliquer

les préférences de l'utilisateur afin choisir les équipements et le réseau d'accès afin d'offrir le service nécessaire, à condition que les composants remplissent la QoS demandée par l'utilisateur.

Face à un environnement très hétérogène, l'importance de ce méta-modèle est la décomposition du monde des télécommunications en plusieurs niveaux d'abstraction selon les différents niveaux de services, ce qui permet à la gestion du système d'être complète et approfondie. Par conséquent, l'application du méta-modèle à chaque niveau de décision, on peut ainsi obtenir un modèle d'architecture (Figure III.22), qui couvre le contexte ambiant.

Les relations verticales et horizontales (Figure III.22, partie 1), sont garanties grâce à la composition et les caractéristiques fournies par récursivité du modèle.

En fait, chaque niveau de visibilité est modélisé de la même façon. Au niveau service (Figure III.22, partie 3), les composants de services sont gérés comme des services en fonction de son type comme SCU (Service Component User), SCA (Service Component Application) ou SCN (Service Component Network). C'est avec l'ensemble des liens virtuels que les composants de service sont liés par la logique de service pour former ce que l'on appelle VPSN (Virtual Private Network Service) qui est essentiel pour la manipulation du service.

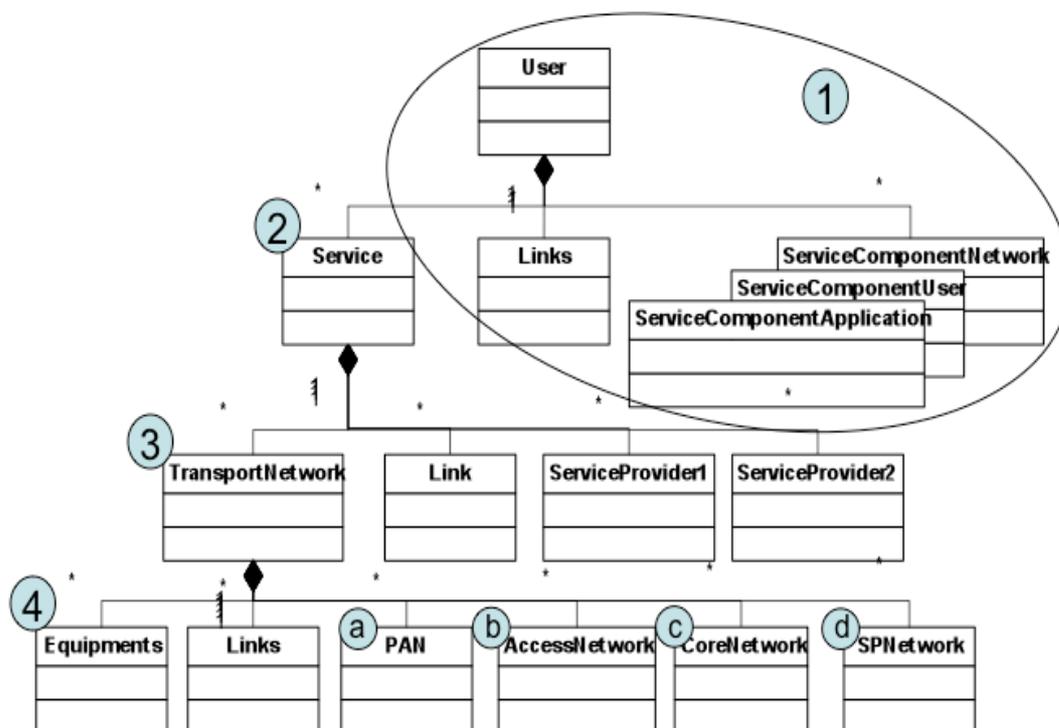


Figure III.22 Modèle architecturale

### III.5.3 Modèle de QoS

---

L'autre élément structurant que nous allons utiliser est le modèle de QoS qui repose sur des critères, des valeurs mesurables et un contrat.

- **Critères du modèle de QoS**

Quatre critères sont définis pour ce modèle de QoS qui sont : *Disponibilité*, *Fiabilité*, *Délai* et *Capacité*. Nous les avons choisis en fonction de leur transversalité aux aires fonctionnelles de gestion [FCAPS (Fault, Configuration, Accounting, Performance, Security)].

Voici donc la définition de chaque critère :

- La **Disponibilité** représente l'aptitude d'un service à être accédé à un instant donné, selon les exigences et conditions contractuelles de temps et d'espace. Elle indique le taux d'accessibilité pour les nœuds de services et les liens logiques.
- Le **Délai** représente l'aptitude d'un service à être exécuté tout en respectant le temps précisé dans les exigences et les conditions contractuelles. Il indique le temps du traitement pour les nœuds et le délai moyen d'acheminement des informations pour les liens.
- La **Capacité** représente l'aptitude du service à avoir les moyens nécessaires pour réaliser son travail selon les exigences et les conditions contractuelles. Elle indique la charge maximale des nœuds de service et le débit applicatif des liens.
- La **Fiabilité** représente l'aptitude d'un service à être exécuté sans détérioration de l'information traitée et en respectant les exigences et les conditions contractuelles. Elle indique pour les nœuds et les liens le taux de modification involontaire de l'information durant leur traitement.

En effet, la sécurité mise à part, les fautes impacteront la fiabilité, la configuration représentent la gestion des états qui seront visibles à travers la disponibilité. Quant à l'accounting qui suit l'attribution des ressources consommées, il est dépendant du délai et de la capacité. Pour terminer, les performances reposent sur les deux derniers critères (délai et capacité).

Ces critères caractérisent la qualité des traitements pour les nœuds de service, la qualité de la mise en relation des liens de service, et ainsi que la qualité totale du service fourni par le réseau d'éléments de service.

- **Valeurs du modèle de QoS**

Les 4 critères que nous avons présenté (§III.5.3) s'évaluent à travers trois types de valeur: valeur de conception (a), valeur courante (b) et valeur seuil (c) et un contrat (d).

- a) **Les valeurs de conception** sont déterminées au moment de la conception des services, elles traduisent leurs possibilités maximales de traitement pour les nœuds et d'interactions pour les liens. Elles interviennent lors de la planification des capacités et dimensionnement des services.
- b) **Les valeurs courantes** indiquent les performances courantes des traitements et des d'interactions, respectivement des nœuds et des liens de service. Ces valeurs sont à surveiller durant l'exploitation pour avoir l'image du comportement des services en temps réel.
- c) **Les valeurs seuils** indiquent la limite du fonctionnement normal des nœuds ou de la réalisation normale des interactions par les liens dans les conditions normales d'usage et d'exploitation des services. Ces valeurs jouent le rôle de seuils d'alerte pour déclencher les réactions adéquates et les processus d'autogestion.
- d) **Le contrat** de chaque composant de service est le résultat de la négociation entre la « QoS demandée » et la « QoS offerte ».

### **III.6 Conclusion**

La gestion de réseau et de service de télécommunication reste un défi majeur des nouveaux environnements qui sont de nos jours de type multi-fournisseurs, dans laquelle on intègre plusieurs types de technologies.

En effet, la gestion de réseaux est l'un des domaines les plus complexes auxquels l'on puisse se confronter; elle cumule la distribution, le temps réel, le transactionnel, la gestion d'équipements. C'est en conséquence une source de coût important pour l'opérateur, qui se voit contraint d'investir des sommes significatives et des compétences critiques dans une fonction qui semble non immédiatement rentable. Néanmoins, le souhait exprimé ici concerne la compréhension de la nécessité de gérer des terminaux, réseaux, services et son intérêt pour l'utilisateur; ce n'est qu'à travers une gestion efficace que le fournisseur sera en mesure d'offrir les services qui le différencieront de la concurrence.

Nous ne sommes pas dans un monde de certitudes dans ce domaine, mais au contraire dans une sphère d'interrogations et de remises en cause permanentes, ce qui doit contribuer à une meilleure réflexion et une plus profonde compréhension du fond de la problématique que nous traitons.

L'état de l'art réalisé a permis de regarder les propositions qui se font au niveau des organismes de standardisation en ce qui concerne la partie informationnelle.

Nous avons constaté une certaine évolution de la modélisation des réseaux télécom et leur entité, puis en adressant le niveau réseau, puis en couvrant plus ou moins bien le niveau service et en terminant par les clients. Le GNIM est particulièrement tourné vers le monde des réseaux, et le CIM vers le monde des entreprises. Pratiquement tous les modèles partagent un certain nombre de caractéristiques: la modélisation objet, la généralité, l'abstraction et l'indépendance vis-à-vis des mises en œuvre et implémentations.

La modélisation que nous appliquons se base sur l'abstraction du monde réel des télécommunications. Le modèle « Nœud, Lien, Réseau » dans lequel le Nœud est une entité collaboratrice, le Lien est l'abstraction du flux et des échanges qui s'opèrent entre les entités collaboratrices, le Réseau représente un ensemble de nœuds et de liens de même nature. Les éléments considérés sont les ressources «équipement, réseau, service, utilisateur » (niveaux de visibilité).

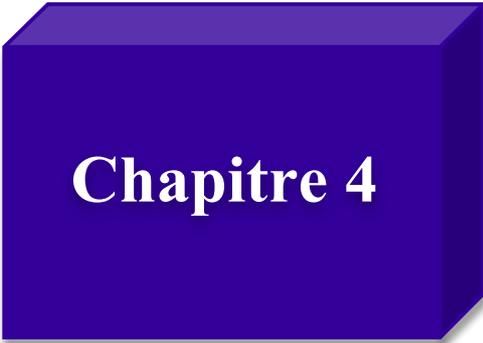
Les travaux que nous avons présentés au niveau du méta-modèle NLR, permettent d'avoir une abstraction du monde réel. Cette modélisation nous permet d'avoir une image complète du réel. A cette modélisation, nous intégrons un modèle de QoS afin d'avoir le comportement de chaque composant à n'importe quel niveau de visibilité, permettant ainsi d'avoir l'image du réel du bout en bout.

Nous résumons dans le tableau ci-dessous les défis face aux nouveaux besoins dans notre contexte NGN/NGS.

	<b>Les nouveaux besoins</b>	<b>Etat de l'art</b>	<b>Défis</b>
Structure Informationnelle	Avoir toute l'image du système d'Information de l'utilisateur dans le but de la gestion tenant compte de QoS des ressources	Solutions basées sur la gestion de réseau et de modèles client serveur. Manque des informations qui permettent de connaître le comportement des composants	Proposer une structure qui permet de représenter le monde réel et tout le système d'information de l'utilisateur en temps réel avec la QoS de la ressource
Système de Gestion de Connaissance	Avoir les informations à jour avec l'aide des règles permettant la prise de décision	Les systèmes de gestion de connaissance d'aujourd'hui	Celui-ci doit être capable de fournir des informations fiables en temps réel malgré les différentes mobilités et hétérogénéités

68 Modèle Informationnel pour une session User-Centric : Inférence et Intégration Intelligente

Evenements du réel	Prendre en considération les évènements produites du monde réel	Architectures basés sur des événements, mais il manque prendre en compte tous les événements du système d'information de l'utilisateur	Proposer un agent qui puisse prendre en compte chaque événement du réel dans l'ensemble des ressources du système
Mobilité	Prendre en compte les différents types de mobilité (user, terminal, réseau, services) sans avoir des coupures	Solutions qui traitent la mobilité du terminal et de réseau mais qui ne prennent pas en compte la mobilité de service	Prendre en compte l'ensemble des mobilités au niveau user, terminal, réseau et service



## Chapitre 4

# **Des besoins NGN aux propositions**

Après avoir fait une analyse de la problématique à traiter, et les solutions existantes (§III), nous présentons nos propositions pour la gestion de la session de l'utilisateur.

D'après les différentes définitions que nous trouvons dans la littérature, nous pouvons conclure que la session est toujours la mise en relation temporelle, entre un utilisateur (connu par son terminal) et l'application (proposée par le système). Mais si l'on veut prendre en compte les besoins NGN, notamment les impacts des différentes mobilités, il nous faut définir une session mobile et pour cela se poser la question de sa gestion durant toute l'activité et les déplacements de l'utilisateur.

Cette session se veut sans coupure ni couture, et donc il nous faut prendre en compte l'environnement très hétérogène (terminaux, réseaux et services) auquel l'utilisateur peut être confronté dans un endroit à un moment donné.

Nous avons pensé nos propositions à l'aide de la méthodologie et des concepts que nous avons résumé dans cette thèse (III.3.5). L'aspect principal de ces concepts, est comment modéliser le monde réel des télécoms pour avoir une représentation générique. La modélisation a été un moyen important pour affronter la problématique que l'on traite car elle nous a permis de représenter le monde "communiquant" réel.

Comme nous l'avons annoncé depuis le début de ce manuscrit, nos propositions concernent la dimension informationnelle de cet environnement hétérogène, pour l'ensemble des niveaux de visibilité, que se soit celui de l'utilisateur, du terminal, du réseau ou du service. Cette base informationnelle est capitale pour prendre des décisions en fonction du comportement des ressources qui interviennent dans la session de l'utilisateur.

Pour ce faire, nous commençons par présenter les besoins NGN à l'aide d'un scénario (§IV.1) pour représenter la problématique que nous désirons traiter, à savoir la gestion de la mobilité de la session « User Centric ». Ce scénario permettra de mieux positionner nos diverses propositions afin d'avoir une continuité de service.

En déroulant le scénario, nous identifierons les événements du réel (§IV.2). Ceci nous permettra de mettre en exergue les informations représentatives et significatives de notre contexte.

Ces différents événements peuvent se produire par le changement de comportement d'une ressource et/ou la mobilité de l'utilisateur, terminal, réseau ou service. La base de connaissance est directement impactée par ces événements du réel qu'il nous faut propager dans la base. Nous proposons d'utiliser l'inférence avec des règles dictées par la structure que nous avons définie pour le modèle informationnel (§IV.3) [17].

De cette façon, nous arrivons à proposer une structure informationnelle basée sur la QoS (aspect comportemental, et non fonctionnel) pour l'ensemble des ressources à gérer. Cela veut dire que la même structure (profil) est appliquée à



Dans la figure IV.1, nous avons recensé les acteurs (du 1 au 35) pour identifier les informations échangées afin de mettre en exergue les informations pertinentes lors du déroulement du scénario dont nous faisons la description ci-dessous.

En première instance, nous avons Bob qui est logé dans un hôtel où il accède aux offres qui sont affichées sur le portail de l'entreprise qui vend des maisons sur son ordinateur portable (1).

Il profite de l'accès Wi-Fi (2) de l'hôtel pour accéder aux offres qui sont affichées sur le portail du fournisseur qui vend des maisons (11). C'est à travers ce portail que Bob peut faire un calcul de prêt (10) par rapport à ses revenus. Ceci est possible grâce à l'accord qui existe entre l'entreprise qui vend les biens immobiliers et la banque de Bob.

Afin d'accéder à ce service, Bob traverse l'infrastructure réseau (2, 3, 4, 5, 6, 7, 8) qui est déployée par l'opérateur, l'utilisateur a recours à la connexion Wi-Fi (2) pour accéder à son service. Le Wi-Fi est fourni par ce même opérateur qui compte avec son infrastructure un core de réseau (4).

Ce core de réseau dont le plan de contrôle se base sur IMS (5, 6, 7, 8) permet de router les paquets. L'IMS relie l'infrastructure du réseau et les services qui sont offerts par les fournisseurs de services. L'architecture IMS est composée de différents composants qui permettent à un utilisateur final accéder au service auquel il a souscrit.

En fait, le service auquel il a souscrit est appelé service exposable, mais pour que Bob puisse accéder à son service, plusieurs composants de services (Authentification, Autorisation, Sélection d'un bien, Calcul de prêt,) (10, 11) sont sollicités de façon tout à fait transparente pour BOB. C'est l'usurware et le serviceware qui gère le service complet avec la QoS souhaitée par l'utilisateur le tout en respectant ses préférences.

Au niveau des services, ce sont les fournisseurs de services qui exposent leurs composants de services. Ces composants sont déployés sur différentes plateformes de services (9, 20) qui appartiennent à différents fournisseurs, ce qui rend cette composition multifournisseurs. Nous pouvons donc constater que la session qui est établie se compose du 1 au 11.

Une fois que Bob repère les biens immobiliers qui lui conviennent, il décide de les visiter et de se rendre sur place. Mais l'ordinateur portable (1) à travers lequel il consultait les biens n'a pas une batterie suffisante pour tenir jusqu'aux visites de ces biens immobiliers. Bob décide donc de basculer sa session de son ordinateur portable (1) vers son téléphone mobile (12). Ce changement de terminal traduit *la mobilité de l'utilisateur (Ev1)* qui doit se faire sans coupure ni couture. Le basculement de la session de son ordinateur portable (1) vers son téléphone mobile (12) doit se faire en respectant ses préférences.

---

Lors de ce changement de terminal, Bob continue à accéder à son service à travers la même infrastructure mais utilise un autre composant Wi-Fi (13). Ce changement de terminal représente donc une mobilité de session qui passe par les composants (12, 13, 14, 15, 16, 17, 18, 19, 20, 21 et 22).

Une fois que Bob accède au service à travers le téléphone portable (12), il se déplace pour visiter les biens immobiliers, et c'est donc à travers ce terminal qu'il peut accéder aux informations des biens immobiliers. Par contre, lorsqu'il se déplace, il change de stations de base (BS) et comme il n'y a plus de Wi-Fi à sa portée, le nouveau emplacement de son téléphone portable (23) accède à son service à travers le réseau de la 3G (24), ce qui représente *la mobilité du terminal*, c'est-à-dire, c'est le terminal qui a bougé et donc il y a un changement de réseau. Les éléments qui constituent maintenant la session sont (23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34 et 35). Même si l'infrastructure au niveau contrôle IMS reste la même, nous pouvons constater que Bob accède à son service au niveau réseau d'accès à travers la 3G et il a basculé sur la deuxième plate-forme de service (33, 34, 35).

C'est donc lorsque l'utilisateur se déplace avec son terminal (*mobilité du terminal*) qu'il y a un autre événement qui se produit au niveau du réseau d'accès, ce qui nous permet d'identifier l'événement 2 (*Ev2*) (Handover horizontale). C'est ainsi que le réseau d'accès (24, 25) achemine au niveau réseau core (26, 27, 28) vers une autre route car nous utilisons une autre infrastructure qui est la 3G.

Au niveau composant de services (33, 34, 35), on utilise des composants fonctionnellement équivalentes mais qui sont sur une autre plate-forme, car le fait que Bob se soit déplacé, d'après ce changement, les composants qui convenaient mieux par rapport au service demandé étaient sur la plate-forme de services numéro 2. C'est de la *mobilité de service (Ev3)* qui se traduit par la *mobilité de session*. Ceci nécessite que l'on gère les changements au niveau des composants.

La problématique traitée dans ce scénario est de savoir comment gérer efficacement au niveau informationnel les événements du réel qui se sont produites. Dans notre scénario, nous nous focalisons sur la *mobilité de l'utilisateur (Bob)* (changement de terminal) et la *mobilité du terminal* (changement de point d'accès). Les événements du monde réel peuvent être assez complexe à gérer car ces différents événements peuvent changer le comportement des ressources. Chaque événement produit une inférence inductive, c'est-à-dire, un changement d'information dans les ressources concernées.

Rappelons que la mobilité de l'utilisateur est un événement qui infère à tous les niveaux (terminal, réseau, service), car lorsque l'utilisateur se déplace, il doit utiliser un autre terminal pour continuer à accéder à son service. Ce changement de terminal va faire que l'accès au réseau soit différent ou qu'il reste le même que l'ancien terminal selon le type de connexion du terminal où l'on a passé la session.

Le réseau doit prendre en compte la nouvelle adresse du terminal et ainsi au niveau service, le composant de service doit prendre en compte la nouvelle adresse du terminal pour envoyer le flux de données au bon terminal. Afin de pouvoir basculer la session de l'ordinateur portable (1) vers le téléphone mobile (12), il a fallu compter avec la connaissance des ressources qui étaient disponibles à ce moment précis. C'est pour cela qu'il est nécessaire de compter avec la bonne information au moment adéquat pour pouvoir faire la prise de décision selon l'évènement qui se présente.

Les différents types de mobilités (considérés comme évènements majeurs) peuvent causer une interruption du service de l'utilisateur, c'est-à-dire avoir une coupure. Nous devons donc gérer efficacement les impacts sur la base de connaissance que nous proposons lors des évènements en faisant évoluer le HSS d'IMS d'aujourd'hui, afin de fournir l'information souhaitée de façon structurée pour qu'il puisse avoir une bonne interprétation par l'utilisateur.

Des travaux du groupe AIRS de Télécom ParisTech ont permis de définir des profils que nous allons utiliser et que nous avons consignés dans les Annexes. Nous avons le *profil de ressource*, qui nous donne les valeurs de conception d'une ressource afin de connaître sa QoS. Bien que il soit important de connaître la valeur de conception de la ressource, il faut aussi savoir la QoS offerte par la ressource par rapport à son utilisation. C'est grâce au *profil d'usage* de la ressource que l'on peut connaître la QoS de la ressource lors du déploiement et de son installation.. Enfin, il faut connaître la QoS courante lors de la phase d'exploitation et l'ensemble des informations donnant l'image du réel, c'est ce que nous consignons dans le *Real Time Profile* (§IV.4) (que l'on expliquera plus tard, puisque ma contribution porte sur ce profil). Mais avant d'expliquer le Real Time Profile, regardons les étapes qui sont utilisées dans les architectures d'aujourd'hui, afin de prendre en compte l'information dont nous avons besoin pour avoir une transparence lorsque l'utilisateur accède à son service.

## **IV.2 Le diagramme du scénario NGN**

Après avoir fait l'analyse du déroulement du scénario NGN (§IV.1), nous nous focalisons sur les acteurs qui interviennent lorsque l'utilisateur accède à son service, notamment lorsque Bob se prépare pour acheter son bien immobilier. Nous avons utilisé la figure du scénario (Figure IV.1), afin de représenter les requêtes nécessaires pour son bon déroulement. Nous avons donc les acteurs suivants :

**Utilisateur Bob** : Il représente les utilisateurs avec une identification unique. Cette identification permet d'accéder aux services auxquels ils sont abonnés.

**Terminaux (PC, PDA)**, ils représentent les équipements qui sont autour d'un utilisateur quelconque. Les terminaux peuvent être ceux qui sont utilisables et/ou ceux qui sont prêts à être utilisés.

**Réseau d'accès (Wi-Fi, 3G)**, qui représente les différentes technologies d'accès que Bob peut utiliser à travers un terminal.

**Services**, qui représentent les services variés supportés par les fournisseurs de services, qui pour simplifier, dans notre cas, est réduit au service de calcul de prêt. Chaque'un de ces acteurs a des interactions avec la base de connaissance, donc nous allons identifier ces échanges qui sont un reflet du réel (§IV.2.1).

### IV.2.1 Les événements du réel

Nous allons représenter à travers un tableau, les événements qui sont produits dans le scénario que nous avons décrit. Ce tableau nous permettra d'identifier les événements de notre scénario NGN. Les événements qui ont été identifiés à travers ce scénario sont la mobilité de l'utilisateur, la mobilité du terminal (pour mémoire), et la mobilité de session.

Tout d'abord nous nous focalisons dans le début de la session de Bob, nous pouvons constater dans la figure IV.2 les éléments qui réalisent la session et les protocoles qui sont impliqués.

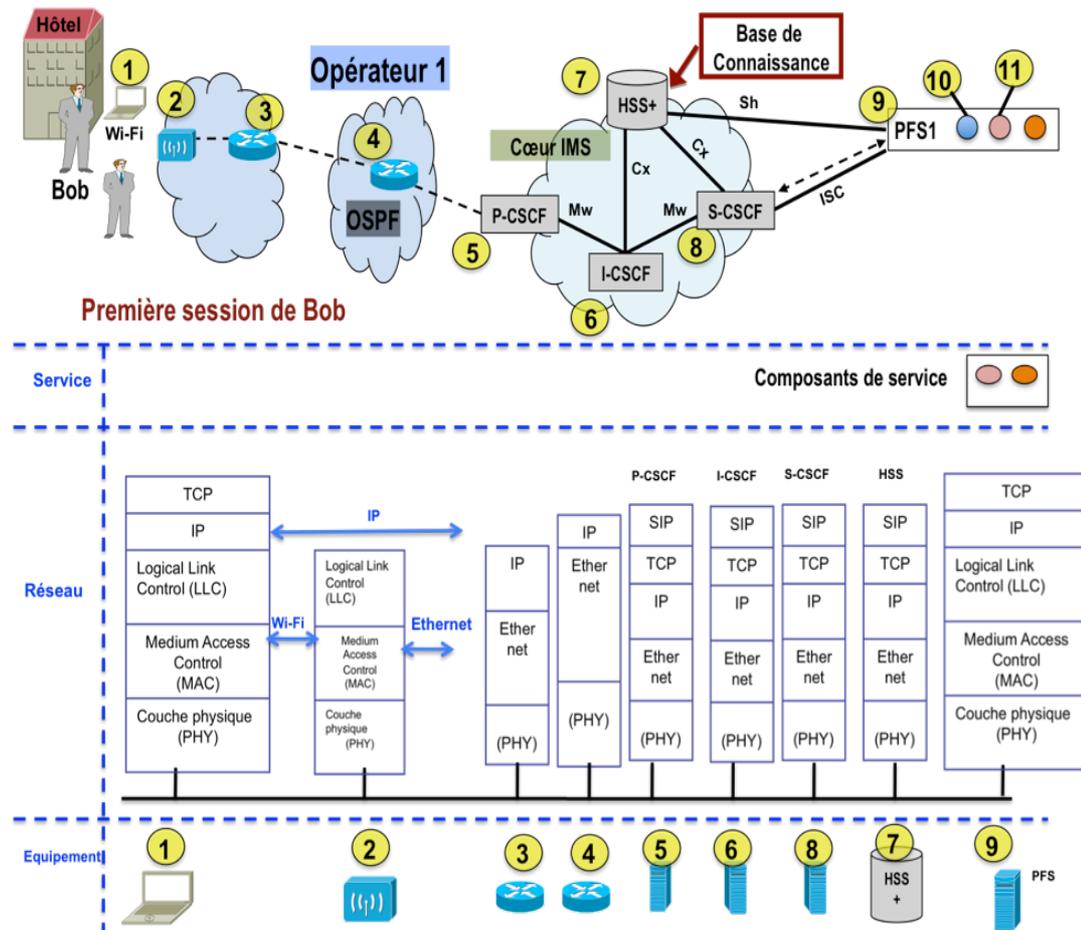


Figure IV.2 Session de Bob

Les protocoles qui interviennent dans la session de Bob nous permettent d'identifier les différentes couches protocolaires de sa session.

Ensuite, comme il a été décrit dans le scénario, Bob voudrait se déplacer pour visiter les biens immobiliers. Comme son ordinateur n'a plus de batterie, il décide alors de basculer sa session vers son téléphone portable, et donc de cette façon il bascule sa session. Ceci est possible grâce à la norme 802.21 de la IEEE (Figure IV.3).

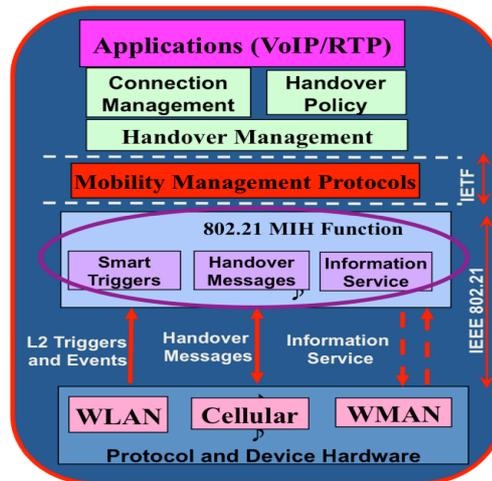


Figure IV.3 Première session de Bob

Nous allons montrer comment les événements se produisent et comment la base de connaissance est impactée par ces événements à travers un tableau.

### IV.3 Gestion des événements et inférence du réel

Un des aspects que nous avons approfondis est en effet les événements qui pourraient se produire toute au long de la session de l'utilisateur.

Afin d'avoir toujours le composant adapté aux besoins de l'utilisateur, nous avons besoin de savoir comment réagir, en interne, lorsqu'il y a un changement de QoS dans un composant. Si nous voulons avoir une efficacité dans le traitement de tous les composants qui interviennent dans la session de l'utilisateur, la prise en compte de l'autonomie est nécessaire, car les traitements par processus reste inadéquats par rapport aux contraintes de délai.

Pour ce faire, nous orientons notre choix vers une approche par événements. Ce type d'approche permet d'avoir des traitements automatiques qui soient réactifs et efficaces. Nous considérons que les approches séquentielles utilisées ne sont pas très performantes par rapport à la problématique traitée, qui est celle d'un environnement très hétérogène et mobile qui demande un temps de réponse du traitement assez court.

Afin que cette approche événementielle soit efficace au niveau traitement lors du changement de QoS d'une ressource, cette gestion a besoin d'une autonomie

---

assez forte. Nous rappelons que la caractéristique d'être autonome veut dire qu'il peut être déployé indépendamment des autres ressources.

Le fait d'être autonome permettra à la ressource d'être indépendante et d'être utilisée à n'importe quel moment, voire à être remplacée si la ressource ne répond plus à la QoS demandée. Cette composition et/ou recombinaison est déclenchée lors de l'établissement de la session de l'utilisateur.

Généralement, on a recours à un gestionnaire, qui a un certain contrôle sur la ressource, dans notre cas, c'est la ressource qui a la connaissance de la QoS qu'elle peut offrir, à travers un agent QoS intégré. Donc la gestion se fait par la ressource elle-même. C'est ainsi que nous avons mis des informations de QoS à chaque composant. De cette façon, le comportement sera surveillé à tout moment par la ressource. A travers la gestion. Cette gestion de QoS est définie par quatre critères génériques qui sont : Disponibilité, Fiabilité, Délai et Capacité dont nous avons parlé précédemment (§III.3.5).

C'est pour cette raison que le VPIN peut être décisionnel, grâce à l'agent de QoS qui est mis en chaque composant à tous les niveaux de visibilité. Les déclenchements d'événements de cet agent se font par *inférence*.

L'*inférence* est une opération mentale, ou jugement, qui consiste à tirer une conclusion. Ces conclusions sont tirées à partir de règles de base.

Nous distinguons trois types d'inférence qui sont: la déduction, l'induction et l'abduction. Ce sont les trois types de raisonnement formalisés par Charles Sanders Peirce en 1903.

L'*induction* se propose de chercher des lois générales à partir de l'observation de faits particuliers, sur une base probabiliste.

L'*abduction* est un procédé consistant à introduire une règle à titre d'hypothèse afin de considérer ce résultat comme un cas particulier tombant sous cette règle.

La *déduction* consiste à former des représentations générales à partir de faits particuliers.

Le type d'inférence qui est utilisé dans notre gestionnaire est l'inférence par induction. Nous avons identifiés des événements qui produisent l'inférence :

- Mobilité (user, terminal, réseau, service)
- Changement des préférences de l'utilisateur
- Changement de QoS d'un composant

Tenant compte de ces événements, nous avons proposé deux types de réaction d'événements, le premier gère l'événement qui est déclenché lorsque la QoS dans une ressource change (§IV.3.1), et le deuxième gère l'événement qui est produit par le changement d'état du composant (§IV.3.2).

Ces deux types d'événements sont intégrés dans un agent qui fait partie du Real Time Profile, cet agent est nommé "**EMA**", pour "Event Monitoring Agent". L'agent EMA peut être utilisé pour les deux types d'événements, soit par le

changement de QoS d'une ressource, soit par le changement d'état d'une ressource. La différence se fait, lorsque la qualité de service d'un élément change, c'est à travers l'agent "changement de qualité de service" que l'action sera prise. En revanche, si l'agent EMA identifie que les modifications se font au niveau état du composant, dans ce cas, c'est grâce à l'événement «changement d'état» que l'action sera prise.

### IV.3.1 Changement de QoS d'une ressource

Nous avons identifié un événement qui permet d'avoir des changements dans la base de connaissances. Cet événement est déclenché en raison d'un changement de QoS dans un composant. Nous avons appelé cet événement "QoS change event".

Un exemple de cet événement pourrait être, lorsqu'un terminal (PDA, Figure IV.4 partie 2) qui est utilisé dans une session arrive à la capacité maximale dans sa mémoire RAM, le terminal ne peut plus être utilisé. Ainsi, en raison du changement de qualité de service dans le composant, l'événement de changement de qualité de service lance la notification (Figure IV.4). Cette notification est reçue par toutes les ressources qui sont au même niveau de visibilité.

Donc, une action est prise, dans ce cas, il y a un changement de terminal, et la session passera du PDA à l'ordinateur fixe. De cette façon, le PDA qui n'a plus la QoS nécessaire ne sera pas pris en compte dans la session de l'utilisateur, suite à son changement de QoS.

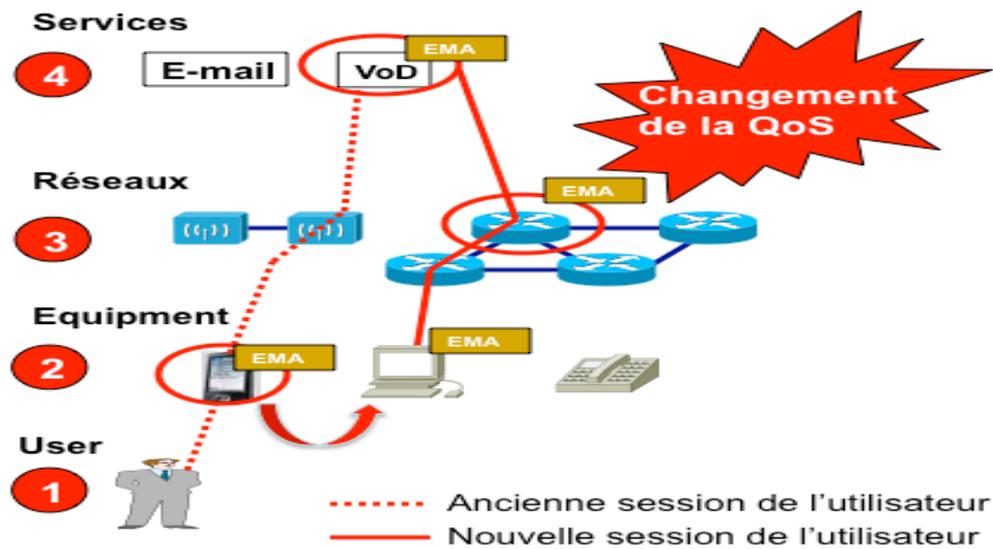


Figure IV.4: Changement de QoS détecté par l'agent EMA

C'est donc avec l'aide de l'agent EMA (Event Monitoring Agent) que le comportement d'un composant peut être surveillé en permanence.

Le deuxième type d'événement se fait à travers le changement d'état dans un composant. Nous avons appelé le deuxième type d'événement comme "state change event".

### IV.3.2 Changement d'état d'une ressource

Ce type d'évènement "changement d'état" peut être causé par un changement d'état dans le composant surveillé. Les états du composant peuvent être "Disponible, Indisponible, Activable, Activé".

Cela signifie que si un élément change l'état d'indisponible vers disponible, c'est un changement où la qualité de service du composant n'intervient pas du tout, c'est seulement l'état du composant qui change.

Comme il a été déjà expliqué dans l'introduction (§1), les différents types de mobilités sont, la mobilité de l'utilisateur, terminal, réseau et de service. L'évènement de changement d'état peut être produit, en raison de ces différents types de mobilité. Autrement dit, lorsqu'un terminal se déplace, peut-être il ne sera pas relié au même point d'accès. Néanmoins, cela ne signifie pas que le terminal avait un changement de qualité de service, le terminal a les mêmes capacités et fonctionne de la même façon, mais comme le terminal a bougé, un changement au niveau connectivité a été donc produit, c'est-à-dire, il aura un autre point d'accès pour se connecter.

Comme il est présenté sur la figure IV.5, nous présentons un exemple où la mobilité des utilisateurs est invoquée. L'agent EMA est présent sur chaque composant, lorsque l'utilisateur se déplace, l'évènement est déclenché lorsque l'utilisateur tente de se connecter à l'autre point d'accès. Par conséquent, il y a une notification que le PDA n'est plus utilisé pour accéder au service.

Donc, le terminal à utiliser désormais sera l'ordinateur fixe. Ce changement de terminal permet une déduction au niveau du réseau et le niveau de service. Le changement se fait au niveau du terminal qui est différent (du PDA vers le PC), donc une nouvelle route doit être construit afin de pouvoir toujours accéder au service.

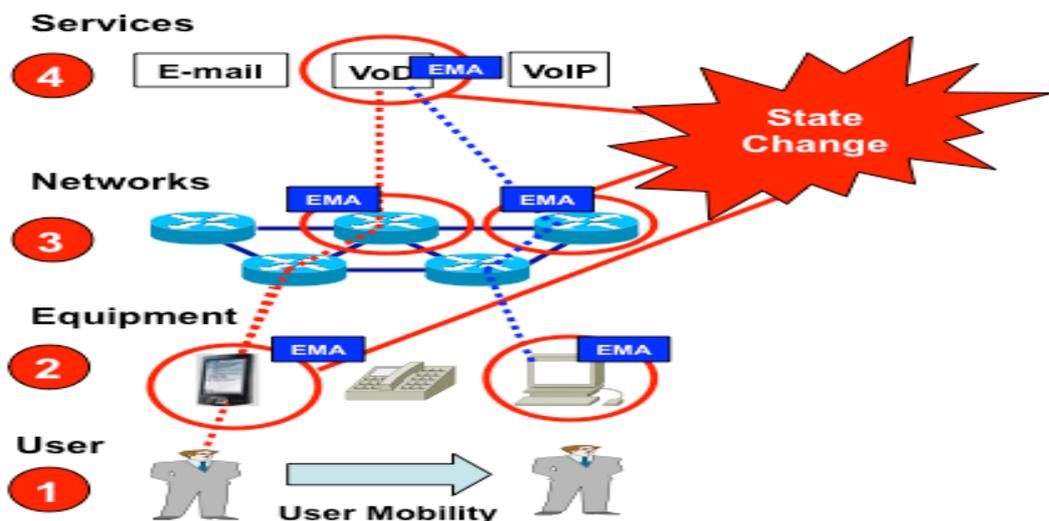


Figure IV.5: Changement d'état détecté par l'agent EMA

L'inférence au niveau du service sera, le changement de l'adresse du routeur qui reçoit le flux. Donc, un autre routeur envoie le trafic vers la vidéo sur demande (VoD).

#### **IV.4 Modèle Informationnel : Real Time Profile**

Nous avons présenté dans la section précédente (§III), l'importance d'avoir une représentation cohérente et structurée des ressources, et d'avoir cette information disponible au bon moment et au bon endroit. Le fait de compter avec les bonnes informations au moment souhaité permet aussi de prendre les bonnes décisions par rapport aux besoins de l'utilisateur.

Nous considérons donc qu'une structuration homogène d'une ressource permet d'obtenir les informations nécessaires de la ressource dans le but de sa gestion. Si nous voulons faire la gestion de bout en bout, nous devons prendre en considération l'ensemble des informations qui donnent le comportement (QoS) de chaque ressource participant au service rendu à l'utilisateur.

La gestion que nous voulons faire, doit être applicable à n'importe quel niveau de visibilité, que ce soit au niveau, terminal, réseau ou service. Nous rappelons que les niveaux de visibilité nous permettent de prendre en compte toutes les entités du monde réel, ce qui fournit l'image entière de tous les éléments à gérer et qui participent lors de l'accès à un service. Le fait d'avoir une cohérence au niveau informationnel permet la gestion dynamique de bout en bout.

Nous pouvons citer comme exemple un utilisateur qui se trouve à son bureau et qui a besoin d'accéder à ses mails. Pour ce faire, d'abord il faut qu'il sache d'abord à travers quel terminal il peut accéder à sa boîte mail. Dans son cas, il peut accéder à travers son ordinateur fixe, ou à travers son téléphone portable qui a un accès wifi et qui permet à travers une application l'accès à ses mails. Cet exemple pourrait être évident car dans ce cas là, l'utilisateur a la connaissance de ses moyens de communications qui sont disponibles à travers les quels il peut accéder à son service.

Pourtant, si jamais son téléphone portable n'a plus de batterie ou son ordinateur fixe tombe en panne, il devra accéder à son service à travers un autre dispositif. Pour cela, il doit utiliser un autre terminal qui soit autour de lui à sa disposition et qui lui permet d'accéder à son service. Le terminal choisi devra quand même supporter l'application nécessaire pour accéder à son mail, il est donc nécessaire que l'utilisateur connaisse les caractéristiques du terminal et l'état actuel de ce terminal, afin de déterminer s'il peut être utilisé pour l'accès à ses mails.

Pour ce faire, nous proposons donc le **Real Time Profile**, qui est le profil utilisé durant l'exploitation d'un service. Le Real Time Profile permet de représenter l'image réelle du composant géré. Il est décrit par son architecture et ses services

offerts via des interfaces. Nous suivons la structuration des entités à gérer par le modèle NLR en nœuds, liens et réseaux, ce qui nous amène à avoir les informations comportementales de ces trois éléments. La structuration du Real Time Profile permet de donner la vue complète d'une ressource tenant en compte son comportement (QoS). En se basant de la modélisation NLR, le Real Time Profile nous permet également d'avoir une vue au niveau réseau de ressources grâce à sa structuration qui permet représenter de la même façon une ressource qu'un réseau de ressources et ceci à tous les niveaux de visibilité.

#### IV.4.1 Real Time Profile d'une ressource

Une ressource se compose d'éléments logiciels qui exécutent et qui rendent le service demandé.

Il se compose aussi d'éléments de support qui matérialisent les contraintes du composant. Un élément logiciel se compose de : L'entité (classe entité) qui représente la fonction cœur du composant, de sa propre gestion (classe gestion) permettant de gérer le service rendu par l'entité, de sa table de connexion (classe connexion) qui matérialise les relations entre composants et des adresses (classe SAP) par lesquels les services du composant peuvent être fournis ou demandés.

La Figure IV.6 présente la structure du Real Time profile dont nous expliquons le contenu ci-dessous :

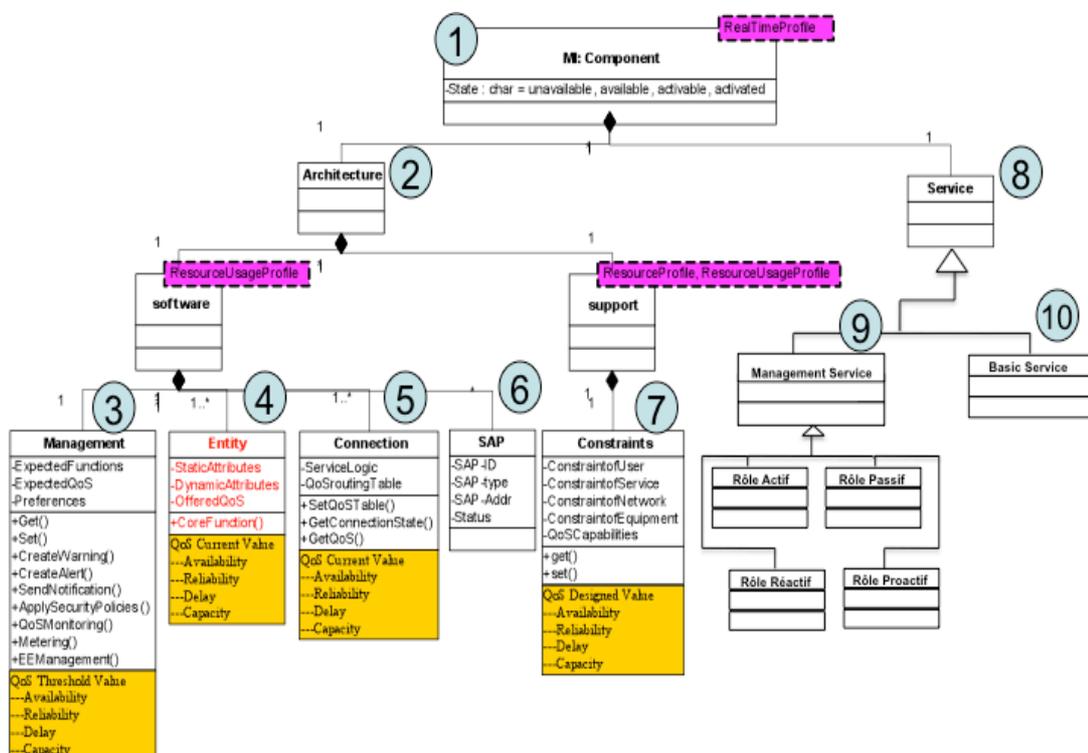


Figure IV.6 Real Time Profile ressource

Nous avons ajouté au Real Time Profile le modèle de QoS ce qui permet d'avoir le comportement en temps réel de la ressource géré.

Dans la Figure IV.6 partie 1, elle représente la première **classe** « **MI : Component** ». Cette classe contient les états qu'une ressource peut avoir dans un moment donné. Les possibles états pour cette classe sont : «disponible», «indisponible», «activable» ou «activé».

L'état «activable» apparaît lorsque la ressource est prête à être sollicitée.

L'état «activé» veut dire que la ressource est actuellement en utilisation.

L'état «disponible» indique que la ressource est ou devient accessible.

L'état «indisponible» veut dire que la ressource est inaccessible de manière provisoire ou permanente.

La deuxième est la **classe** « **Architecture** » Figure IV.6 partie 2. Sur cette classe en particulier nous devons savoir avant tout que une Architecture est une structure d'ensemble, c'est-à-dire, une définition des règles de composition du système et de la coopération des composants, à des fins de transparence pour l'utilisateur du service.

L'ensemble de matériels et logiciels s'analyse à travers une architecture physique du réseau qui se traduit par les caractéristiques de l'élément à gérer et une architecture logique du réseau, dans la quelle se trouve la partie administrable de l'élément à gérer comme les propriétés statiques et dynamiques.

La classe « *Architecture* » présente l'architecture interne d'une ressource. Elle est composée d'une partie « *support* » qui contient les contraintes de conception de la ressource et la partie « *logicielle* » (software) qui contient toute information concernant les points décrits ci-dessous :

**Classe Management** Figure IV.6 partie 3 : Cette classe décrit les valeurs seuils de QoS d'une ressource. Elle contient des méthodes d'autogestion de chaque élément, pour comparer les valeurs seuils et leurs valeurs actuelles ce qui permet de faire la gestion de la ressource par la QoS. En effet, c'est en considérant les valeurs seuils qu'il est possible de regarder les limites d'une ressource. Lorsque ces valeurs seuils sont atteintes, la gestion intervient en déclenchant une notification de cet évènement pour annoncer cet évènement.

**Classe Entity** Figure IV.6 partie 4 : Cette classe contient les informations qui décrivent la ressource, mais sur tout les informations de la QoS courant de la ressource. Elle contient des attributs (statique ou dynamique) et des méthodes qui

---

construisent les traitements de l'élément. La classe Entity représente la fonction cœur de l'entité et c'est là que l'on a les paramètres de QoS mesurables qui nous permettent d'obtenir le délai du traitement, la fiabilité du traitement du nœud et sa disponibilité.

**Classe Connection** Figure IV.6 partie 5 : Elle décrit les liens internes et/ou externes de chaque ressource qui déterminent la topologie interne et externe de la ressource. Cela permet d'avoir l'information des liens avec d'autres ressources, avec qui la ressource en question établit une relation.

**Classe SAP** Figure IV.6 partie 6 : La classe SAP (Service Access Point) décrit l'adresse de chaque ressource où le service du réseau de ressources par lesquels les services peuvent être fournis ou demandés. C'est à travers cette adresse que la ressource est sollicitée ainsi que ses fonctionnalités offertes. Chaque adresse permet d'identifier une fonctionnalité particulière de la ressource qui est vue de l'extérieur.

Sur la **classe Support**, elle dispose d'un contrat dont elle contient les valeurs de conception des quatre critères de QoS ainsi que les contraintes (7).

La **Classe Constraints** Figure IV.6 partie 7 doit être respectée au moment de l'exploitation de la ressource. Cette classe décrit la QoS que la ressource fournit. Cette QoS se traduit par les valeurs de conceptions, ces valeurs ont été données par son constructeur au moment de la conception de la ressource.

**Classe Service** Figure IV.6 partie 8 Selon notre modélisation, toutes les ressources sont accessibles par leur interface de service. Dans cette interface de service, nous comptons avec la gestion du service (**Management Service**) (9) et (**Service de Base**) (10).

La **Classe Management service** Figure IV.6 partie 9 sur son interface de gestion, il compte avec 4 rôles qui sont :

- **Rôle passif** : il peut répondre à une demande.
- **Rôle actif** : il envoie des requêtes à l'extérieur (notifications).
- **Rôle Réactif** : Après une notification, il fait une action (il réalise quelque chose, qui a été prédéfini).
- **Rôle Proactif** : Il compte avec une intelligence pour sa gestion.

La **Classe Basic service** Figure IV.6 partie 10 décrit son service rendu à son interface usager. C'est à travers cette classe qu'il peut y avoir un traitement pour une ressource, une interaction (ou mise en relation) pour un lien ou bien le service

global d'une application (réseau de ressources).

#### **IV.4.2 Real Time Profile d'un réseau de ressource**

Lorsque nous parlons de modéliser au niveau réseau, cela veut dire que nous avons le comportement de l'ensemble de ressources qui appartiennent à un niveau de visibilité donnée. Le fait de gérer un réseau n'est pas quelque chose de nouveau en soit, mais ; la nouveauté de notre approche réside sur le fait de considérer le réseau comme une entité à part entière en la modélisant comme un élément abstrait du télécom à travers le Real Time Profile. C'est la modélisation qui nous permet aussi de gérer de façon efficace le comportement global de ce composant abstrait réseau.

Cette modélisation du monde réel des télécoms nous permet d'avoir l'ensemble de nœuds et les flux, et donc par conséquence nous pouvons obtenir la traçabilité des flux échangés entre ces ressources.

Nous appliquons la modélisation de la même façon que pour une ressource et à n'importe quel niveau de visibilité. Ceci permet d'avoir une uniformité des concepts que l'on applique dans la structuration des ressources ainsi que la vue complète du comportement pendant et tout au long de la session de bout en bout.

Afin d'obtenir le comportement de bout en bout en continue de la vue réseau, nous introduisons la QoS dans le Real Time Profile. C'est à travers les 4 critères que nous regardons les aspects de QoS.

Si nous suivons le modèle NLR, un élément réseau est fait de la relation qui existe entre les éléments d'un même niveau de visibilité, et ils se reposent sur les éléments de réseau dans un niveau de visibilité inférieur. Cette relation entre niveaux reflète aussi les relations qui existent dans les éléments du monde réel, c'est-à-dire, un utilisateur a besoin toujours d'un terminal pour accéder à son service (relation « utilisateur – terminal »), de la même façon que un terminal a besoin d'un réseau d'accès pour pouvoir se communiquer (relation « terminal-réseau »).

Pour réaliser une communication de bout en bout, chaque réseau se base sur le réseau du niveau inférieur pour le transfert de proche en proche. Ces relations induisent la nature récursive du modèle informationnel et donc son application sur l'ensemble des niveaux de visibilité.

Au même titre que le Real Time Profile avec une vue ressource, la vue réseau est décrit par son architecture et ses services offerts via des interfaces. Cette vue

---

réseau peut appartenir à un niveau de visibilité donné, que ce soit, terminal, réseau ou service. Un élément réseau se conforme des éléments logiciels qui vont exécuter et rendre le service demandé. Il se conforme aussi par le support qui permet d'obtenir les contraintes à chaque niveau de visibilité.

Cela veut dire que ce profil contiendra les informations suivantes dans chaque une de ses classes :

**Classe MI** : Nom du réseau géré et le niveau de visibilité auquel il appartient, ça peut être le niveau, terminal, réseau ou service.

**Classe Management (Figure IV.7 partie 3)** : Dans cette classe, nous comptons avec l'ensemble d'informations de toutes les ressources qui conforme le réseau des ressources. Si une ressource tombe en panne, par conséquence, c'est la gestion qui prendra en compte que cette ressource n'est plus disponible ; et donc cet événement va inférer au niveau réseau qui est géré.

**Classe Entity (Figure IV.7 partie 4)** : Cette classe compte avec l'information qui décrit le réseau de ressources ainsi que le traitement entre les ressources qui conforment le réseau.

**Classe Connection (Figure IV.7 partie 5)** : Cette classe contient les liens existants avec le réseau de ressources externes. Cela veut dire que si nous avons un réseau IP d'un opérateur (donc nous avons comme ressources les routeurs), l'ensemble de routeurs forme un réseau, et donc dans cette classe nous regarderons la connexion avec un autre réseau (appartenant à un autre opérateur).

**Classe SAP (Figure IV.7 partie 6)** : La classe SAP (Service Access Point) décrit toutes les adresses des ressources où le service du réseau de ressources par lesquels les services peuvent être fournis ou demandés. C'est à travers ces adresses que les ressources sont sollicitées ainsi que ses fonctionnalités offertes.

Sur la **classe Support**, elle dispose d'un contrat dont elle contient les valeurs de conception des quatre critères de QoS ainsi que les contraintes.

La **Classe Constraints (Figure IV.7 partie 7)** : doit être respectée au moment de l'exploitation des ressources. Cette classe décrit la QoS que les ressources fournissent. Cette QoS se traduit par les valeurs de conceptions, ces valeurs ont été données par les constructeurs au moment de la conception de ces ressources.

**Classe Service** : Selon notre modélisation, toutes les ressources sont accessibles par leur interface de service. Dans cette interface de services, nous comptons avec la gestion du service (**Management Service**) et (**Service de Base**).

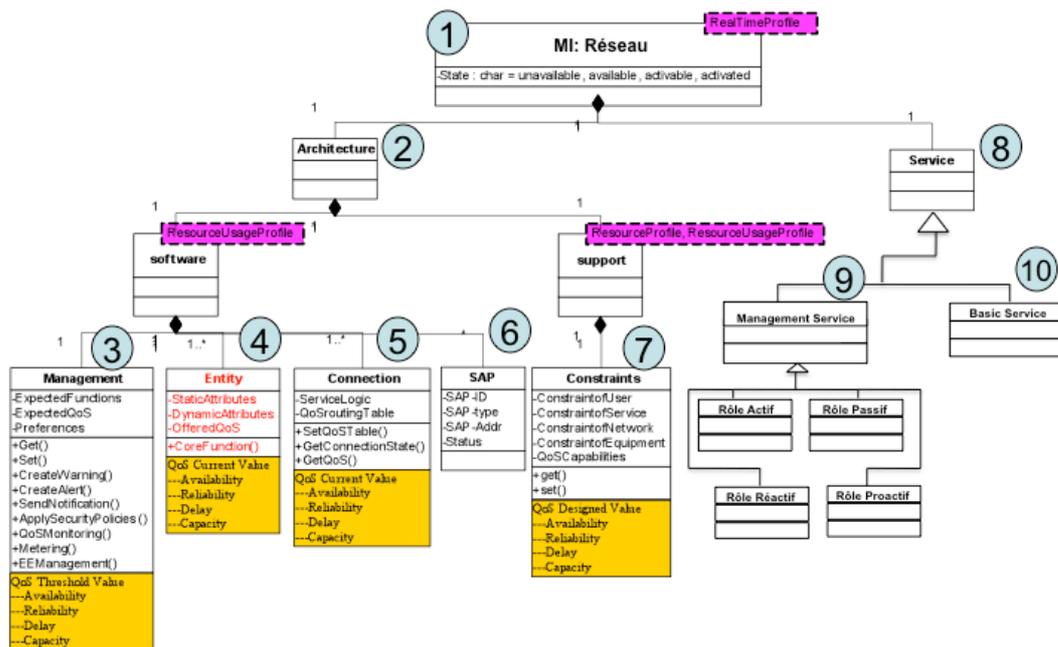


Figure IV.7 Real Time Profile réseau

### IV.4.3 Real Time Profile du système

Une des conditions principales afin de réussir une gestion de tout le système est d'être capable d'avoir une vue de l'ensemble de façon unifié et complète du système géré. Ceci est possible si l'opérateur dispose d'une base de connaissances sur la quelle les informations et processus de gestion seront visibles à tout moment.

Comme il a été montré auparavant, le Real Time Profile peut représenter une ressource que un réseau de ressources en utilisant la même et ceci à n'importe quel niveau de visibilité. En se basant de cette structuration, nous avons modélisé le système global (Figure IV.5) du scénario.

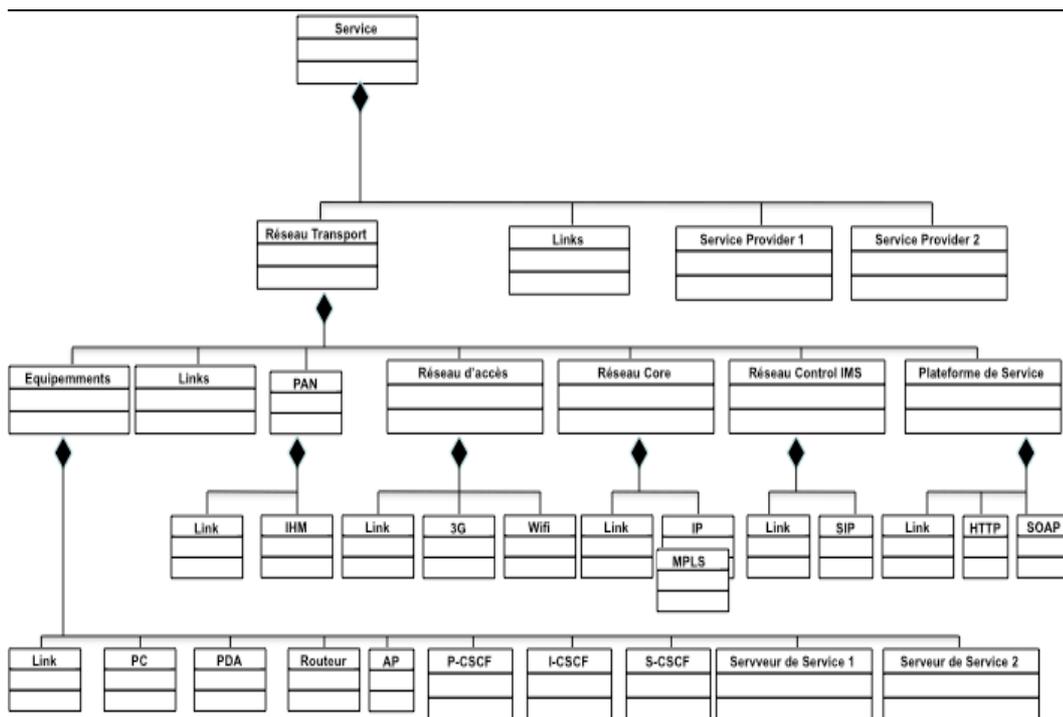


Figure IV.8 Real Time Profile du système complet

Cette modélisation permet de représenter le système global auquel l'utilisateur sera confronté lorsqu'il accédera à son service.

Nous présentons les composants PC, PDA, 3G et Wi-Fi, pour montrer les informations dont nous avons besoin lorsque l'utilisateur bascule la session de son PC vers son PDA. Ce changement fait que nous allons utiliser un autre type de réseau pour continuer l'accès au service qui est du Wi-Fi au réseau 3G.

L'introduction des informations relatives aux ressources se fait grâce au Real Time Profile de chaque ressource connaît sa QoS (comportement), ces informations sont montrés ci-dessous : id : l'identification du composant et sa description, nom\_service : description\_service, catégorie\_service, version, ...etc. Les paramètres non fonctionnelles de la QoS de la ressource comme : délai, fiabilité, disponibilité et capacité. (4) son statut, et (5) ses connexions ainsi que ses contraintes.

Toutes ces ressources PC (Figure IV.9), PDA (Figure IV.10), 3G (Figure IV.11) et Wi-Fi (Figure IV.12) suivent la même structure informationnelle. Cette homogénéisation permet que les ressources puissent s'autogérer et qu'il connaisse sa propre QoS.

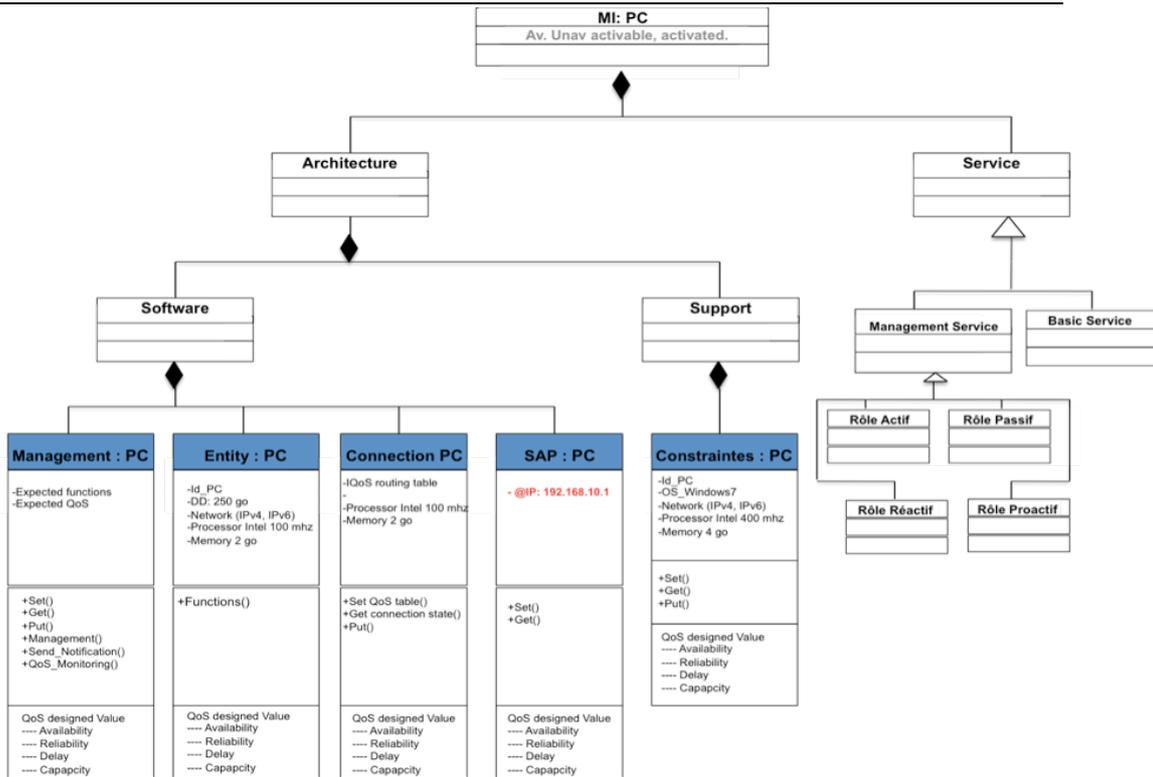


Figure IV.9 : Ressource PC

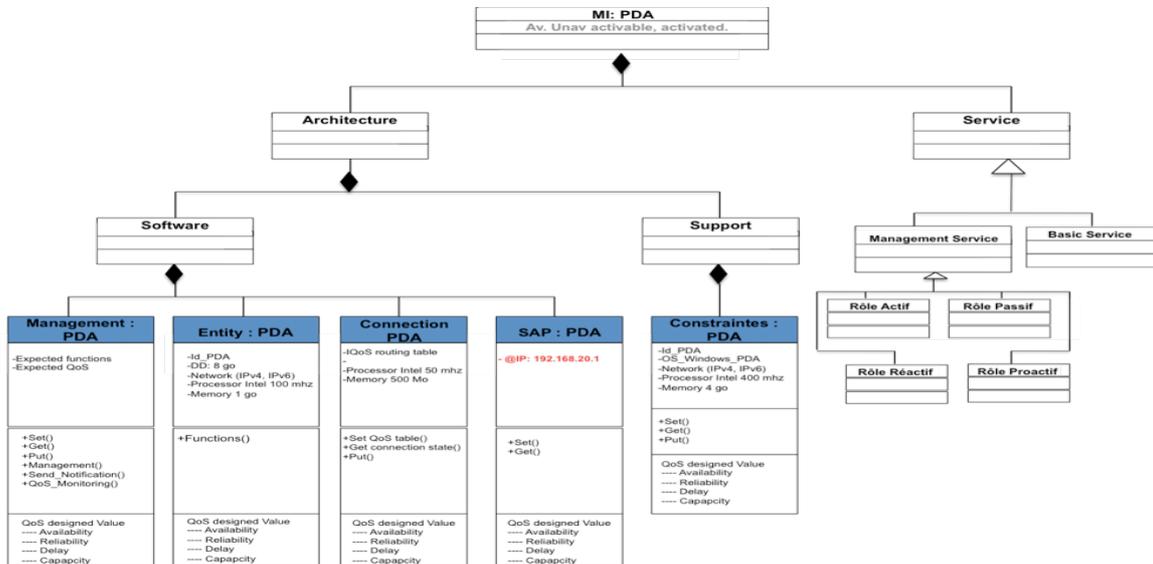


Figure IV.10 : Ressource PDA

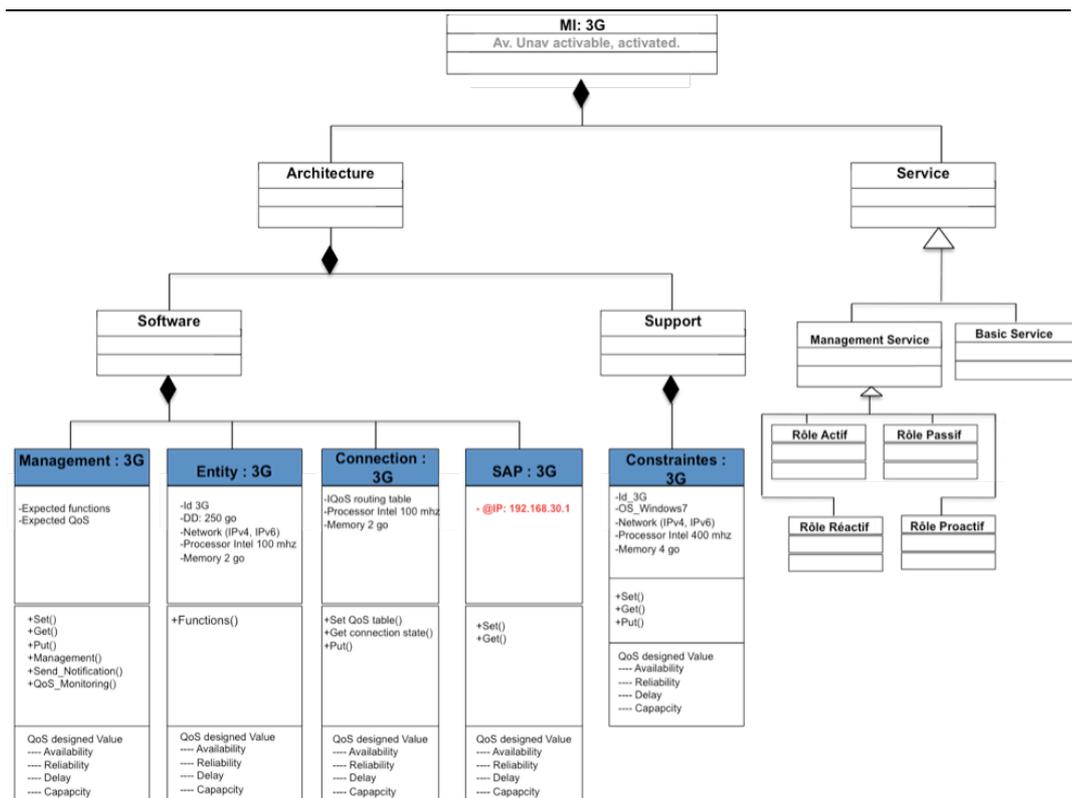


Figure IV.11 : Ressource 3G

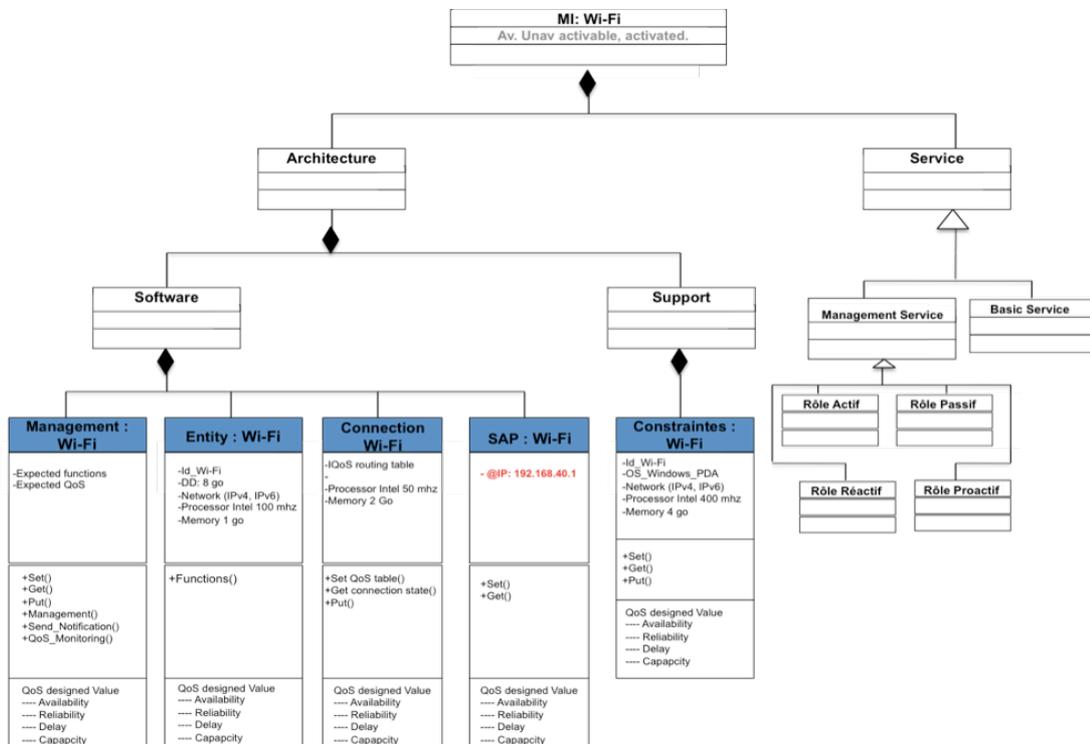


Figure IV.12 : Ressource Wi-Fi

## IV.5 Le VPIN (Virtual Private Information Network)

Nous considérons que l'information est essentielle pour décrire le comportement

(QoS) d'un élément qui fait partie du système d'information d'un utilisateur. Dans le développement des applications, l'information a toujours été très liée à son application, ce qui fait avoir une relation très étroite entre l'information que l'on montre et l'application. Ceci ne permet pas d'obtenir une information sauf si l'information en question fait partie de l'application.

Afin de pouvoir compter avec l'information d'une ressource donnée, nous devons faire la différence entre l'information qui est propre à la ressource et l'information qui est propre aux applications, ce qui permettra que l'information qui est montré dans chaque ressource soit indépendant des applications. Ceci fera que des différentes ressources et/ou applications, puissent prendre et utiliser les informations d'une ressource.

Pour ce faire, nous proposons un réseau informationnel que nous dénommons « VPIN »<sup>31</sup>. Le VPIN représente tout le système d'information de l'utilisateur en temps réel. Il tient en compte les préférences de l'utilisateur, sa localisation et/ou activité.

Le VPIN contient des informations pertinentes, y compris les informations de QoS sur le composant au niveau terminal, réseau et service. La collecte de toute cette information permet à notre base de connaissances être décisionnelle. Lorsqu'un changement se fait dans la base de connaissances, il est nécessaire de vérifier l'architecture afin de vérifier l'impact qui est affecté à la base de connaissances, par conséquent, en fonction de la structure, la déduction sera faite.

Le VPIN permet d'organiser les informations de façon structurée. Cette organisation permet de recevoir les informations pertinentes qui sont nécessaires au bon moment et au bon endroit en cas de besoin. Cette organisation suit les niveaux de visibilité afin d'avoir la vision d'ensemble du système d'information de l'utilisateur. Cette méthode est appliquée à tous les niveaux de visibilité (utilisateur, terminal, réseau et services) sur la base du méta-modèle NLN expliqué précédemment (§IV.1).

En appliquant le méta-modèle NLN au niveau du réseau, nous pouvons dire que le noeud est un routeur et le lien est la façon dont ils sont reliés, dans ce cas, ce sera par les protocoles de routage, et l'ensemble de ces noeuds et des liens forme le réseau à ce niveau de visibilité.

Le VPIN offre une vision globale de tous les composants qui constituent le système d'information au niveau de l'utilisateur (user-centric). L'avantage de cette

---

<sup>31</sup> Virtual Private Information Network

---

organisation est, que la structure du modèle permet de refléter l'image du monde réel.

Ceci permet ainsi d'avoir des informations pertinentes en temps réel. Cette organisation est toute à fait « user-centric », cela signifie qu'il a été conçu pour répondre aux besoins des utilisateurs, éliminer les obstacles techniques qui empêchent un utilisateur d'accéder à ses services.

Pour fournir le service qui répond aux besoins de l'utilisateur et s'adapte à son contexte, il est nécessaire d'avoir une vision globale de tout son environnement. La connaissance des préférences de l'utilisateur ainsi que le comportement des différentes ressources (équipements, services, réseaux) de son système permet d'avoir la bonne décision qui correspond aux exigences de l'utilisateur. En effet, pour un utilisateur mobile voulant accéder à ses services à tout moment et en tout lieu, on doit lui garantir une continuité de session avec une adaptation de son service selon les différents changements.

Ces changements regroupent le changement de son réseau d'accès, le changement du terminal utilisé, le changement du contexte utilisateur (localisation, activité, etc.) ainsi que le changement du comportement des services (disponibilité, changement de la qualité de service, etc.).

Ainsi pour établir et maintenir un système d'information, qui répond aux besoins de l'utilisateur dans un environnement caractérisé par l'hétérogénéité et la mobilité, nous considérons un modèle d'information décisionnel. Ce modèle définit les différentes actions à prendre en compte lorsqu'un événement se produit (changement de la qualité de service d'un composant de service, changement de contexte et préférences de l'utilisateurs, etc.). Ceci garantit d'avoir une représentation du monde réel de l'utilisateur et de prendre une décision correcte et transparente pour l'utilisateur.

C'est donc l'organisation du VPIN qui permet d'avoir les informations nécessaires à chaque niveau de visibilité qui sont suffisantes et persistantes dans notre base de connaissance.

Afin de prendre en compte tout le système d'information de l'utilisateur, nous avons divisé le VPIN sur 4 niveaux de visibilité, les niveaux de visibilité aux quels nous faisons références ont été expliqué auparavant (§III.5).

La Figure IV.20 montre les 4 niveaux qui composent le VPIN qui sont, le niveau utilisateur dont nous l'appelons *VPUIN* (Virtual Private User Information Network), le niveau équipement qui est *VPEIN* (Virtual Private Equipment Information Network), le niveau réseau qui est *VPCIN* (Virtual Private Connectivity Information Network) et le niveau service qui est *VPSIN* (Virtual Private Service Information Network).

Les différents niveaux dont nous faisons référence sont présentés ci-dessous :

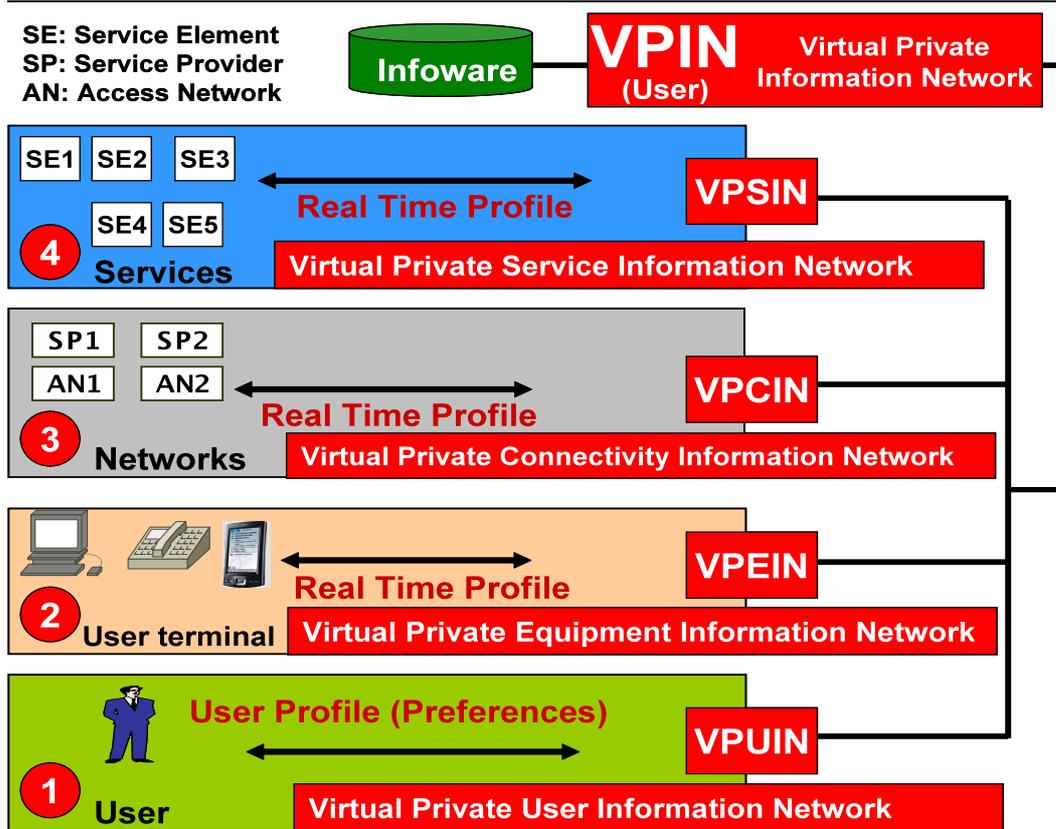


Figure IV.13: Structure du VPIN

Le premier niveau **VPUIN** "Virtual Private User Information Network" (Figure V.19, partie 1) contient le profil de l'utilisateur. Sur ce profil, nous avons les préférences de l'utilisateur. Ces préférences peuvent changer selon la localisation de l'utilisateur et de son activité. Quoi qu'il en soit, les préférences de l'utilisateur sont de plus en plus difficiles à gérer et atteindre, car elles peuvent changer dans un court période de temps selon le contexte utilisateur et ses besoins.

Une de ces préférences de l'utilisateur est le coût. Par exemple, lorsqu'un utilisateur voudrait avoir le service d'Internet le moins cher chez lui. Il va donc vérifier les différentes offres et les paquets des fournisseurs de services. Le fournisseur d'Internet choisit sera celui qui offre une majeure offre de services avec le prix le plus bas du marché.

Une autre préférence de l'utilisateur est de type technique. Par exemple, en se basant des préférences au niveau coût, on peut imaginer que après avoir choisi son fournisseur d'Internet, l'utilisateur voudrait choisir le types de connexions (par exemple, Ethernet, Wi-Fi) et par ailleurs, l'utilisateur souhaite utiliser autre type de terminal au lieu de son ordinateur fixe (par exmaple, PDA, téléphone portable). Ou encore, lorsque l'utilisateur voudrait se communiquer avec sa famille, il préfère utiliser la visioconférence au lieu d'utiliser son mail.

---

Enfin, nous pouvons considérer aussi les préférences au niveau usage. Ce type de préférence pourrait se constituer avec les différentes configurations qui sont mis dans une ressource. Par exemple, lorsque l'utilisateur décide regarder la télévision sur son ordinateur, certaines adaptations peuvent en effet être faits dans la qualité de l'image (choix noir et blanc ou à couleur), ou la qualité audio (son stéréo, etc).

Toutes ces préférences de l'utilisateur pourraient contribuer à satisfaire certains des besoins des utilisateurs, mais pas toutes, car nous devons tenir compte du fait que l'utilisateur est en constant mouvement. Il est important de rappeler que les différents types de mobilité (utilisateur, terminal, réseau, service) peuvent produire une interruption de la session de l'utilisateur lorsqu'il accède à son service, si nous n'avons pas l'information adéquat au bon moment.

Dans le deuxième niveau, nous avons le *VPEIN* "Virtual Private Equipment Information Network" (Figure IV.20, partie 2). Dans ce niveau de visibilité, nous comptons avec toutes les informations des équipements qui appartiennent à un utilisateur. Les informations de ces équipements ne sont pas seulement celles qui appartiennent à l'utilisateur, mais les équipements qui sont disponibles et peuvent être utilisés par l'utilisateur en fonction de sa localisation. L'information qui est contenue dans l'équipement permet d'obtenir son comportement. Afin d'avoir l'information complète à ce niveau, nous comptons aussi avec l'information au niveau interfaces qui relie les équipements à ce niveau de visibilité.

En troisième niveau, nous avons le *VPCIN* "Virtual Private Connectivity Information Network" (Figure IV.20, partie 3). Il concerne l'ensemble des accès aux réseaux d'information à partir de laquelle l'utilisateur peut être connecté. La particularité de ce niveau est que l'on peut trouver les réseaux d'accès des opérateurs ainsi que les fournisseurs de services. Dans ce niveau, on trouve l'information relative au comportement de chaque ressource de réseau. Le flux à ce niveau est le routage, ce qui permet l'échange des chemins et des routes à prendre selon la table de routages et le comportement des équipements au niveau réseau.

Dans le quatrième niveau, nous comptons avec le *VPSIN* "Virtual Private Service Information Network" (Figure IV.20, partie 4). Il fournit l'information relative au comportement des composants des services liés à un utilisateur. Le comportement des composants peut varier selon la QoS du composant

Un exemple de ce niveau pourrait être, lorsqu'un utilisateur se trouve à l'aéroport et il a besoin d'acheter un billet avec son PDA. Il y a un serveur qui diffuse cette information, mais si ce serveur ne fonctionne plus, dans le VPSIN, il va être un autre serveur qui prend en charge la tâche. Donc l'utilisateur aura toujours les informations nécessaires sur ce service.

A chaque niveau, nous avons les informations de toutes les ressources qui

appartiennent à ce niveau de visibilité, ceci est possible dû à ce que toutes les ressources ont la même structure qui est le Real Time Profile (§V.4) ce qui permet une homogénéisation des ressources et une cohérence lorsque l'on voudrait obtenir ou échanger une information. Il faut remarquer que les informations qui sont mises au niveau du VPIN concernent le reflet du comportement des ressources, c'est-à-dire que nous ne mettons pas des informations liées à une application donnée. Les informations qui sont stockées sont donc des informations liés à la ressource afin d'obtenir son comportement.

A chaque niveau de visibilité, un réseau de ces ressources (niveau horizontal) est formé, ce qui permet d'avoir connaissance du comportement de ces ressources (QoS). L'objectif de former un réseau à chaque niveau de visibilité est d'avoir la connaissance des ressources disponibles qui pourraient être utilisées lorsque l'utilisateur souhaite accéder à un service et souhaite établir une session.

Ce réseau informationnel (VPIN) aura toute l'information requise par un utilisateur pour avoir la continuité de son service. Le VPIN permet d'avoir toute l'information concernant le système d'information de l'utilisateur en temps réel, ceci tenant compte de sa localisation et activité. Il est l'image réelle du système d'information de l'utilisateur.

Le fait d'obtenir l'information complète d'un composant permet connaître son comportement, ce qui aidera à la prise de décisions lorsqu'un utilisateur choisit un composant à un moment donné.

Donc, s'il y a un changement de comportement (QoS) dans un composant, il pourrait être remplacé par un autre qui répond à la qualité de service attendue.

Le remplacement d'un composant peut être fait avec l'aide du Real Time Profile (§V.4), qui est présente à toutes les ressources et à toutes les niveaux de visibilité comme le terminal, réseau et service. C'est pour cette raison que notre Real Time Profile a été conçu. Les différents profils des ressources donnent les structures informationnelles pour que le Système d'Information puisse avoir toutes les informations nécessaires et décisionnelles.

## ***IV.6 Conclusion***

Sur ce chapitre, nous avons réalisé nos propositions qui concerne la partie informationnelle, ceci afin de pouvoir gérer de façon efficace la session de l'utilisateur qui se veut sans coupure ni couture. D'après les besoins NGN/NGS, notamment au niveau de l'hétérogénéité (terminal, réseau, service) et la mobilité (utilisateur, terminal, réseau, service), nous les avons identifié en forme de scénario, ce qui nous a permis d'identifier les manques au niveau informationnelle.

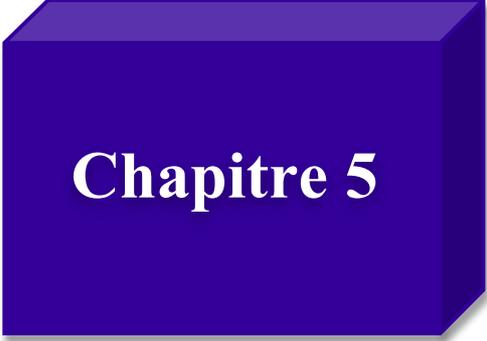
---

Nous avons donc proposé une structure informationnelle qui permet de rendre une ressource autonome à n'importe quel niveau de visibilité. Étant donné que c'est la même structure qui est appliquée aux niveaux de visibilité (terminaux, réseaux et services), ceci nous permet de pouvoir échanger des informations entre ces composants au niveau vertical ou horizontal.

En effet, pour permettre cette communication verticale et horizontale, nous avons proposé un réseau informationnel du point de vue de l'utilisateur, ce qui permet de refléter le monde réel au niveau système d'information de l'utilisateur. Ce réseau informationnel permet de prendre en compte l'environnement très hétérogène sur lequel l'utilisateur peut être confronté.

Pour prendre en compte ce monde réel, nous avons proposé un gestionnaire des événements qui permet de notifier les possibles changements de l'environnement de l'utilisateur, que ce soit par le changement de QoS d'une ressource ou par son changement d'état.

Dans le chapitre qui suit, nous allons présenter nos premiers travaux au niveau implémentation du scénario que nous avons décrit, dans lequel, nous mettons en œuvre nos propositions. Cela va permettre de regarder les aspects de faisabilité de nos propositions.



## Chapitre 5

# **Plate-forme de validation de concepts**

---

Afin de mesurer la faisabilité de nos propositions, nous présentons la plate-forme de validation de nos travaux. Tout d'abord nous décrivons la plate-forme qui a été utilisée pour travailler notre scénario final, sur laquelle nous exposons toute l'infrastructure déployée pour faire tourner le scénario (§V.1). Par ailleurs, pour les aspects réseaux, nous avons travaillé sur une plate-forme IMS de type Open Source (§V.2) de l'institut Allemand Fokus, sur laquelle nous avons appliqué nos propositions sur une amélioration au niveau informationnel, notamment au niveau du HSS de cette plate-forme (§V.3) de Fokus. Afin d'émuler une infrastructure d'opérateur, nous avons utilisé des routeurs virtuels pour faire la partie réseau d'accès et voir le comportement de nos propositions par rapport à ce nouveau environnement NGN (§V.4). Enfin, nous donnons les conclusions des expérimentations (§V.5).

## ***V.1 Plate-forme du scénario final***

Nous présentons l'ensemble des éléments que nous avons utilisé pour faire tourner le scénario que nous avons proposé.

En effet, l'originalité que nous présentons dans ce scénario est premièrement son potentiel à supporter les défis au niveau NGN (mobilité, hétérogénéité et approche user-centric) ; deuxièmement, dans sa capacité à assurer une session mobile unique pour chaque utilisateur tout en tenant compte des contraintes de QoS ; troisièmement, dans sa contribution à la création de nouveaux services personnalisés qui répondent aux besoins des utilisateurs ; et finalement, dans son potentiel à dépasser le modèle client-serveur en introduisant une nouvelle approche NGS (Next Generation Services) qui est basée sur une composition dynamique à couplage lâche de services.

### ***V.1.1 Architecture du scénario***

Afin de répondre à tous ces défis au niveau réseau et au niveau service, on introduit une nouvelle architecture de service, appelée NGN/NGS Middleware. Comme son nom l'indique, ce Middleware dépasse les défis au niveau NGN (mobilité, hétérogénéité et approche user-centric) tout en fournissant une nouvelle vision de service appelée NGS. Cette dernière est basée sur un ensemble d'éléments de services qui sont liés les uns les autres selon une composition dynamique et à couplage lâche de services. En effet, cette architecture est un ensemble de composants fonctionnels bien structurés qui supportent la personnalisation de services et qui maintiennent une session continue pour chaque utilisateur dans un environnement mobile et hétérogène, tout en respectant les exigences au niveau QoS et les préférences de l'utilisateur.

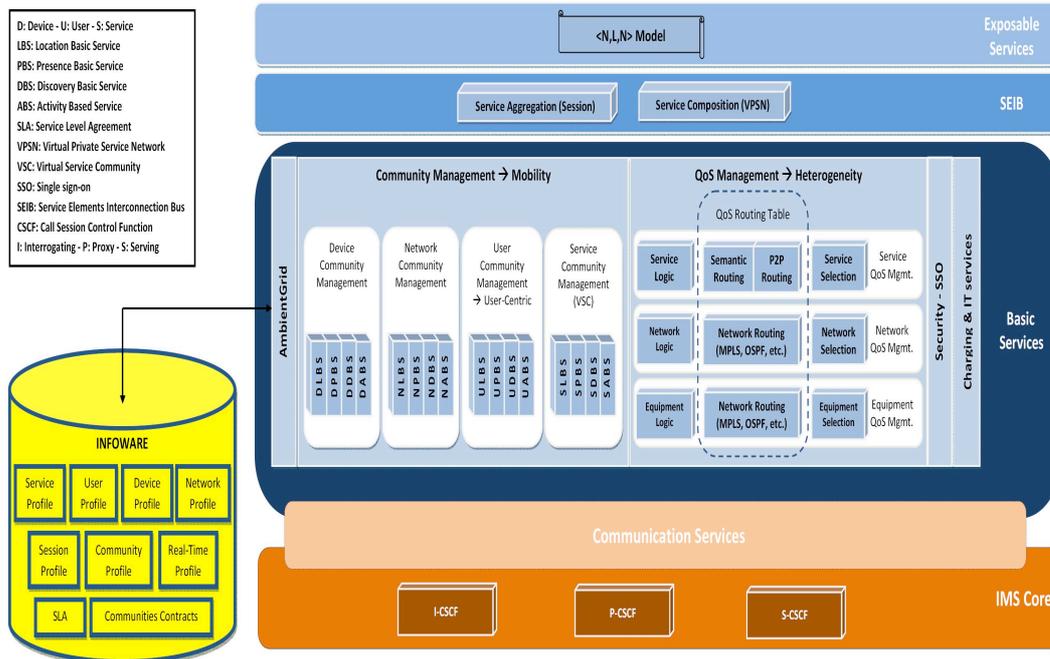


Figure V.1: NGN/NGS Middleware

Comme le montre la figure V.1, le NGN/NGS Middleware est divisé en trois parties principales: « Services de Base », SEIB et « Services Exposables ». Les « Services de Base » sont un ensemble de fonctionnalités responsables d'offrir et d'assurer les besoins fondamentaux énumérés ci-dessus (mobilité, hétérogénéité et User-Centric). Pour cette raison, cette partie est divisée en trois sous-parties: Community Management, sous-partie qui répond aux besoins de mobilité ; QoS Management, sous-partie qui couvre les exigences de l'hétérogénéité ; et enfin le User Management, sous-partie qui assure l'approche user-centric. Ainsi, tous ces services de base rendent transparent l'environnement sous-jacent et renforcent le concept NGN. Au dessus de la partie « Services de Base », nous avons le SEIB (Service Elements Interconnection Bus) qui est le support d'interconnexion (agrégation/composition) entre des éléments de services partageables, mutualisables et trans-organisationnels. Au dessus de ce SEIB, nous avons la partie « Services Exposables ». Cette partie est constituée d'un groupe de services applicatifs qui seront exposés dans les catalogues des utilisateurs finaux. Nous avons implémenté l'Infoware, qui est la base de connaissance vis à vis de l'opérateur.

## V.2 Plate-forme IMS Fokus

L'IMS, c'est avant tout une plateforme commune à tout service. En effet, un des buts principaux de l'IMS est de rendre la gestion du réseau plus facile. Pour ce faire, IMS est situé entre les couches de transport et les couches applicatives.

IMS agit donc comme une couche totalement indépendante, quels que soient les terminaux et équipements en jeu, les protocoles utilisés, séparant ainsi le réseau de transport de base et les applications. De plus, IMS est également découpée en différentes couches. Nous verrons cela plus en détail dans la partie suivante. Mais on peut déjà noter que cette plateforme commune permettra de réduire le temps de mise en oeuvre d'un nouveau service. En effet, au vue de l'architecture actuelle, ou plutôt des architectures actuelles, dès qu'un opérateur souhaite proposer un nouveau service, il doit penser au préalable à aménager son réseau. Cela passe par les équipements, les liens, les connexions. Chaque technologie actuelle (GPRS, WLAN, fixe,...) possède ainsi son propre réseau coeur. Ce qui est lourd et onéreux comme investissement et mise en exploitation pour un opérateur. Avec l'IMS, tout nouveau service viendra se greffer à l'architecture standardisée déjà en place.

Au niveau de l'existant dans la couche de contrôle, nous allons présenter l'architecture IMS par rapport aux travaux faits au niveau Core IMS de Fokus. Le core OS-IMS a été développé par l'institut Fraunhofer FOKUS en Allemagne et est destiné à des plateformes de type Linux.

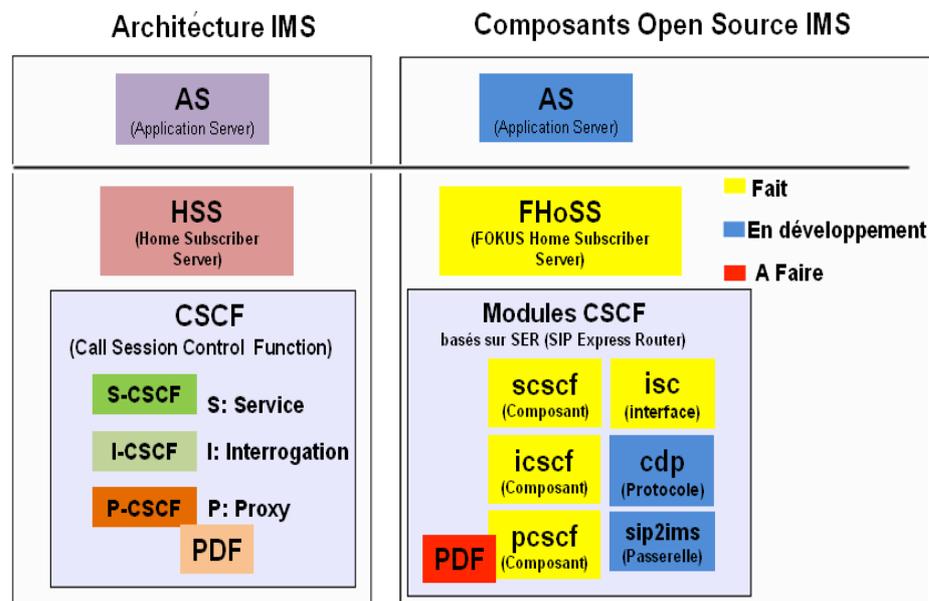


Figure V.2: IMS vs IMS de Fokus

Les modules du SER-IMS (CSCF) (Figure V.2) sont implémentés en utilisant SER (SIP Express Router) :

- Module **Proxy-CSCF** (pcscf) : Joue le rôle de Parfeu pour la signalisation et l'affirmation d'identité de l'utilisateur (SER + pcscf).
- Module **Interrogating-CSCF** (icscf) : Questionne le HSS à partir

d'identités publiques indiquées par l'appelant ou l'appelé et en fonction des réponses, route les messages vers le S-CSCF approprié, CDiameterPeer (cdp) communique avec le HSS sur l'interface Cx d'IMS (SER + icsf + cdp).

- Module **Serving-CSCF** (scscf) : Communique également avec le HSS en utilisant diameter (au-dessus de l'interface Cx) pour télécharger le profil utilisateur et décider vers quel serveur d'application (AS) le message sera expédié à travers l'interface ICS (SER + scscf + cdp + ics).
- Les **ASs** : sont des entités où on trouve hébergée la partie logique de la session. L'AS peut se trouver dans le User Home network ou dans un réseau externe. Si elle se trouve dans le User home network elle peut joindre le HSS en utilisant le protocole Diameter.

La mise en place de ce cœur «OS-IMS» est décrite dans l'annexe B.

### ***V.3 Besoins au niveau de la base de données***

L'aspect « user-centric » est un besoin primordial pour le projet UBIS. Pour cette raison, toute demande de service nécessite la personnalisation par l'utilisateur, l'accessibilité par n'importe quel terminal, une meilleure connectivité, une continuité dans le service et un provisionnement dynamique de ressources. Toutes ces exigences nécessitent une inférence informationnelle et non pas une simple mise à jour de la base de données. Pour cela, on a proposé une base de connaissances (Infosphere/Infoware) afin de tenir compte dynamiquement et en temps réel de l'information personnalisée et des ressources ambiantes.

L'Infoware est une base d'information structurée qui est située au niveau du fournisseur. Il gère efficacement et de manière dynamique les informations décisionnelles et réactives. En effet, cette structure informationnelle est composée de plusieurs profils (profil de service, profil temps réel, profil de session, profil utilisateur, etc.). D'autre part, afin de permettre à l'utilisateur de prendre la bonne décision au bon moment, l'Infosphere est situé au niveau de l'utilisateur. Dans ces deux bases de connaissances, on applique un méta-modèle bien défini qui tient compte des quatre niveaux de visibilité (utilisateur, service, réseau et équipement). Ce modèle informationnel organise toutes les informations nécessaires tout en adoptant une vision de QoS, afin d'avoir partout une adaptation dynamique de service.

Comme nous l'avons déjà noté dans le « Cœur IMS », nous utilisons le FOKUS Open IMS comme plan de contrôle. Cette solution de contrôle introduit

comme base de données le FHoSS (FOKUS Home Subscriber Server) (Figure V.3).

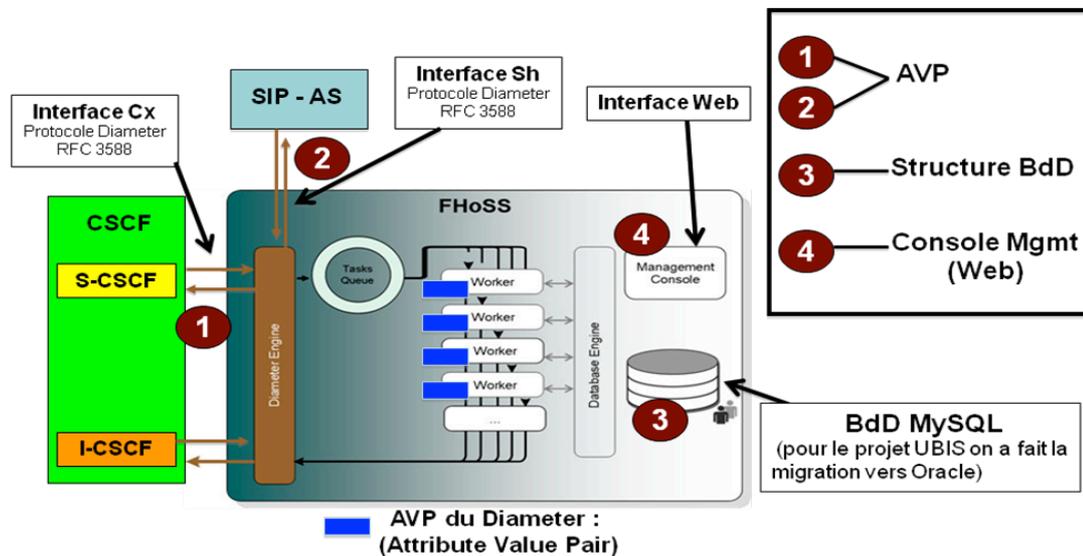


Figure V.3: FHoSS

C'est une base de donnée centralisée contenant certaines informations comme le profil des usagers (identités des usagers, services auxquels les usagers ont souscrits, le S-CSCF allouée à chaque usager, etc.) et des informations de sécurité (contrôle d'accès: authentification et autorisation). Elle est entièrement codée en Java. Les données de l'utilisateur sont stockées dans une base de données MySQL. FOKUS Open IMS définit FHoSS comme un configurateur pour le système de gestion de la base de données et pour les interfaces Diameter avec les CSCFs et la couche applicative. Cette base de données permet la génération des vecteurs d'authentification et fournit une interface de gestion HTTP afin de faciliter la gestion des profils des utilisateurs et des iFCs associés.

Les données sont structurées dans le FHoSS selon plusieurs tables (Figure V.4) :

- Identifiants et info de l'utilisateur (IMPI, IMPU)
- Relation Identifiants publiques et privés (IMPU2IMPI)
- Localisation (networks, roam)
- Adresse de l'AS assigné à l'user (apsvr, as\_perm\_list)
- Tarification (chrginfo)
- Etat de l'utilisateur register/no register (notify\_ims\_user\_state)
- Tables de SCSCF (scsf et scsf\_capabilities)
- Table repository data (rep\_data)





interne des tâches et des alarmes

- Gestion centralisée de plusieurs instances
- Concept unique de retour arrière (Flashback)
- Pérennité de l'éditeur : avec plus de 40% de part de marché.
- Réglages fins : dans la mesure où l'on connaît suffisamment le moteur, presque tout est paramétrable.
- Accès aux données système via des vues, bien plus aisément manipulable que des procédures stockées.
- Interface utilisateur remaniée et extrêmement riche, permettant des requêtes par modification des plans d'exécution.
- Architecture Multi-Générationnelle (AMG).
- Services Web, support XML.
- Ordonnanceur intégré.
- Compression des données et des sauvegardes.

Ce sont ces caractéristiques qui nous ont permis de faire un choix pour utiliser les bases de données de type Oracle, bien qu'il existe aussi des bases de données MySQL, nous avons constaté un manque des outils et de fonctionnalités pour le déploiement de nos propositions.

Au cours de la migration de la base de données du MySQL en Oracle, nous avons tenu compte des quatre niveaux de visibilité (user, service, réseau et équipement) afin de rendre cette simple base de données une base de connaissances avec laquelle le FHoSS sera nommé HSS+ (Figure VI.5).

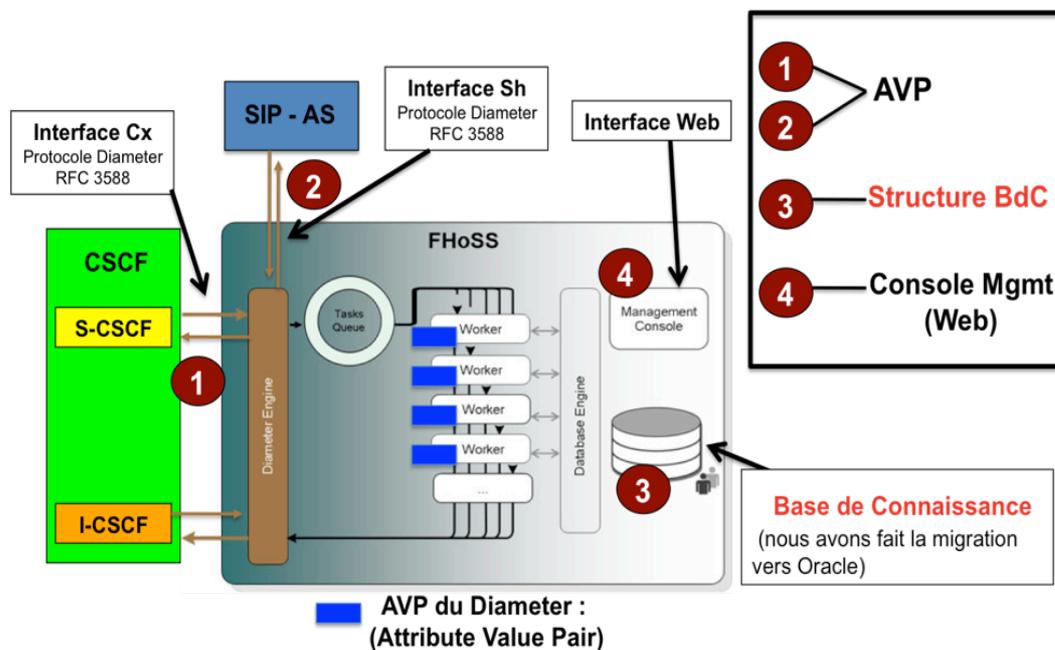


Figure V.6: HSS +

---

## V.5 Fonctionnalités dans la base informationnelle

Nous présentons les fonctionnalités qui ont été mises dans notre base informationnelle et en donnant une brève explication.

### Schéma de données

Les données sont stockées sous forme relationnelle, c'est à dire, pour simplifier sous forme de tables, constituées de colonnes typées. Toutes les données qui appartiennent à un même utilisateur (table mais aussi vues, indexs, procédures, synonymes, etc.) s'appellent un schéma.

Nous présentons ci-dessous un exemple au niveau base de données pour la création de la structure d'un ressource, dans cette structure nous prenons en compte les informations de QoS. Cette structure est valable au niveau terminal, réseau et service. Au niveau du scénario que nous avons décrits (§IV), cette table permettra d'avoir l'image réel de chaque ressource qui va intervenir dans la session de l'utilisateur.

### Création d'une table de ressource :

```
CREATE TABLE RESSOURCE (
  IdRessource      VARCHAR2(10),
  ResoName         VARCHAR2(25) NOT NULL,
  ResoDescrip      VARCHAR2(256) NOT NULL,
  -- Litteral Ressource Description
  ResoType         VARCHAR2(10) NOT NULL,
  ResoVendor       VARCHAR2(25) NOT NULL,
  ResoLogicAdr     VARCHAR2(10) NOT NULL,
  /* NEXT FOUR (4) VALUES ARE CONCEPTION QOS */
  ResoAvailabilityCncp NUMBER(5,3) NOT NULL,           -- MEMORY
  ResoReliabilityCncp  NUMBER(5,3) NOT NULL,           -- PERCENT LOSS
  ResoDelayCncp       NUMBER(5,3) NOT NULL,           -- POWER SUPPLY
  ResoCapacityCncp    NUMBER(5,3) NOT NULL,           -- MIPS
  /* NEXT EIGHTVALUES (8) ARE MIN AND/OR MAX THRESHOLD MANAGEMENT */
  MinResoThrsldAvlbl NUMBER(5,3),
  MaxResoThrsldAvlbl NUMBER(5,3),
  MinResoThrsldRlbl  NUMBER(5,3),
  MaxResoThrsldRlbl  NUMBER(5,3),
  MinResoThrsldDel   NUMBER(5,3),
  MaxResoThrsldDel   NUMBER(5,3),
  MinResoThrsldCapa  NUMBER(5,3),
  MaxResoThrsldCapa  NUMBER(5,3),
  /* NEXT VALUE IS FOR RESSOURCE STATE MANAGEMENT */
  ResoState VARCHAR2(10) NOT NULL CHECK
    (ResoState IN
 ('AVAILABLE','UNAVAILABLE','ACTIVABLE','ACTIVATED')),
  /* NEXT FOUR (4) VALUES ARE CURRENT QOS */
  ResoAvailabilityCur NUMBER(5,3) NOT NULL,           -- MEMORY
  ResoReliabilityCur  NUMBER(5,3) NOT NULL,           -- PERCENT LOSS
  ResoDelayCur        NUMBER(5,3) NOT NULL,           -- POWER SUPPLY
  ResoCapacityCur     NUMBER(5,3) NOT NULL,           -- MIPS
  CONSTRAINT pk_Ressource PRIMARY KEY(IdRessource));
```

La base de connaissance a été structuré de la façon suivante au niveau ressource :

```

CREATE TABLE RESSOURCE(
  IdRessource      VARCHAR2(10),
  ResoName         VARCHAR2(25) NOT NULL,
  ResoDescrip     VARCHAR2(256) NOT NULL,          -- Litteral Resource
Description
  ResoType        VARCHAR2(10) NOT NULL,
  ResoVendor      VARCHAR2(25) NOT NULL,
  ResoLogicAdr    VARCHAR2(10) NOT NULL,
  /* NEXT FOUR (4) VALUES ARE CONCEPTION QOS */
  ResoAvailabilityCncp NUMBER(5,3) NOT NULL,      -- MEMORY
  ResoReliabilityCncp NUMBER(5,3) NOT NULL,       -- PERCENT LOSS
  ResoDelayCncp   NUMBER(5,3) NOT NULL,          -- POWER SUPPLY
  ResoCapacityCncp NUMBER(5,3) NOT NULL,         -- MIPS
  /* NEXT EIGHTVALUES (8) ARE MIN AND/OR MAX THRESHOLD MANAGEMENT */
  MinResoThrslAvlbl NUMBER(5,3),
  MaxResoThrslAvlbl NUMBER(5,3),
  MinResoThrslRlbl NUMBER(5,3),
  MaxResoThrslRlbl NUMBER(5,3),
  MinResoThrslDel NUMBER(5,3),
  MaxResoThrslDel NUMBER(5,3),
  MinResoThrslCapa NUMBER(5,3),
  MaxResoThrslCapa NUMBER(5,3),
/* NEXT VALUE IS FOR RESSOURCE STATE MANAGEMENT */
  ResoState VARCHAR2(10) NOT NULL CHECK
    (ResoState IN
 ('AVAILABLE','UNAVAILABLE','ACTIVABLE','ACTIVATED')),
/* NEXT FOUR (4) VALUES ARE CURRENT QOS */
  ResoAvailabilityCur NUMBER(5,3) NOT NULL,      -- MEMORY
  ResoReliabilityCur NUMBER(5,3) NOT NULL,       -- PERCENT LOSS
  ResoDelayCur        NUMBER(5,3) NOT NULL,      -- POWER SUPPLY
  ResoCapacityCur     NUMBER(5,3) NOT NULL,      -- MIPS
  CONSTRAINT pk_Ressource PRIMARY KEY(IdRessource));

CREATE TABLE RESO_INTF_USG_SEBASE (
  IdRessource      VARCHAR2(10) NOT NULL,
  IdSebase         NUMBER NOT NULL,
ArgPos
  ArgType          VARCHAR2(32) NOT NULL,
  InOutArg         char(1) NOT NULL
    CHECK (InOutArg IN ('I','O')),          -- = I -> IN ; = O -> OUT
  CONSTRAINT pk_ResoIntfUsgSebase PRIMARY KEY(IdRessource, IdSebase, ArgPos),
  CONSTRAINT fk_ResoIntfUsgSebase_Reso FOREIGN KEY (IdRessource)
    REFERENCES RESSOURCE ON DELETE CASCADE,
  CONSTRAINT fk_ResoIntfUsgSebase_Mtd FOREIGN KEY (IdSebase)
    REFERENCES SEBASE ON DELETE CASCADE);

CREATE TABLE RESO_LOG_USG_SEBASE (          -- Logique de Service - plan usage
  IdRessource      VARCHAR2(10) NOT NULL,
  IdPrevSebase     NUMBER NOT NULL,
  IdNextSebase     NUMBER NOT NULL,
  CONSTRAINT pk_ResoLogUsgSebase PRIMARY KEY(IdRessource,IdPrevSebase,
  IdNextSebase),
  CONSTRAINT fk_ResoLogUsgMetod_Reso FOREIGN KEY (IdRessource)
    REFERENCES RESSOURCE ON DELETE CASCADE);

CREATE TABLE RESO_INTF_MGT_SEBASE (
  IdRessource      VARCHAR2(10) NOT NULL,
  IdSebase         NUMBER NOT NULL,
ArgPos
  ArgType          VARCHAR2(32) NOT NULL,
  InOutArg         char(1) NOT NULL
    CHECK (InOutArg IN ('I','O')),          -- =I -> IN ; =0 -> OUT
  CONSTRAINT pk_ResoIntfMgtSebase PRIMARY KEY(IdRessource,IdSebase, ArgPos),
  CONSTRAINT fk_ResoIntfMgtMetod_Reso FOREIGN KEY (IdRessource)
    REFERENCES RESSOURCE ON DELETE CASCADE,
  CONSTRAINT fk_ResoIntfMgtMetod_Mtd FOREIGN KEY (IdSebase)
    REFERENCES SEBASE ON DELETE CASCADE);

CREATE TABLE RESO_LOG_MGT_SEBASE (          -- Logique de Service - plan management
  IdRessource      VARCHAR2(10) NOT NULL,
  IdPrevSebase     NUMBER NOT NULL,
  IdNextSebase     NUMBER NOT NULL,

```

```

CONSTRAINT pk_ResoLogMgtSebase PRIMARY KEY(IdRessource,IdPrevSebase,
IdNextSebase),
CONSTRAINT fk_ResoLogMgtMetod_Reso FOREIGN KEY (IdRessource)
REFERENCES RESSOURCE ON DELETE CASCADE);

CREATE TABLE RESO_INTF_CTL_SEBASE (
    IdRessource VARCHAR2(10) NOT NULL,
    IdSebase NUMBER NOT NULL,
    ArgPos NUMBER NOT NULL,
    ArgType VARCHAR2(32) NOT NULL,
    InOutArg char(1) NOT NULL
    CHECK (InOutArg IN ('I','O')), -- =I -> IN ; =O -> OUT
    CONSTRAINT pk_Reso_Intf_Ctl_Sebase PRIMARY KEY(IdRessource, IdSebase,
    ArgPos),
    CONSTRAINT fk_ResoIntfCtlSebase_Reso FOREIGN KEY (IdRessource)
    REFERENCES RESSOURCE ON DELETE CASCADE,
    CONSTRAINT fk_ResoIntfCtlSebase_Mtd FOREIGN KEY (IdSebase)
    REFERENCES SEBASE ON DELETE CASCADE);

CREATE TABLE RESO_LOG_CTL_SEBASE ( -- Logique de Service - plan usage
    IdRessource VARCHAR2(10) NOT NULL,
    IdPrevSebase NUMBER NOT NULL,
    IdNextSebase NUMBER NOT NULL,
    CONSTRAINT pk_ResoLogCtlSebase PRIMARY KEY(IdRessource,IdPrevSebase,
    IdNextSebase),
    CONSTRAINT fk_ResoLogCtlMetod_Reso FOREIGN KEY (IdRessource)
    REFERENCES RESSOURCE ON DELETE CASCADE);

```

Cette structure permet de décrire toute ressource qui soit mis dans la base de connaissance, ceci est applicable à tout niveau de visibilité, que ce soit terminal, réseaux ou service.

Une fois que nous avons les informations pertinentes en temps reel, lors de l'exploitation, il faut tenir compte des constraints des ressources qui sont utilises. Pour ce faire, nous avons crée une structure qui permet d'avoir ces informations :

```

CREATE TABLE SERV_USAGE_PROFIL (
    IdSExp VARCHAR2(10) NOT NULL,
    /* FOUR (4) NEXT VALUES : OFFERED QOS */
    ServAvailabilityOfrd NUMBER(5,3),
    ServReliabilityOfrd NUMBER(5,3),
    ServDelayOfrd NUMBER(5,3),
    ServCapacityOfrd NUMBER(5,3),
    /* FOUR (4) NEXT VALUES : REQUESTED QOS */
    ServAvailabilityDmd NUMBER(5,3),
    ServReliabilityDmd NUMBER(5,3),
    ServDelayDmd NUMBER(5,3),
    ServCapacityDmd NUMBER(5,3),
    CONSTRAINT pk_ServUsageProfil PRIMARY KEY (IdSExp),
    CONSTRAINT fk_ServUsageProfil_Servicexp FOREIGN KEY (IdSExp)
    REFERENCES SERVICEXP (IdSExp));

/* CONSTRAINTS ABOUT THE USER */
CREATE TABLE SERV_USER_CNSTR (
    RoleType VARCHAR2(40) NOT NULL,
    IdSExp VARCHAR2(10) NOT NULL,
    ServAuthentUsr VARCHAR2(3) CHECK (ServAuthentUsr IN ('YES','NO')),
    ServAutorUsr VARCHAR2(3) CHECK (ServAutorUsr IN ('YES','NO')),
    ServForFree VARCHAR2(3) CHECK (ServForFree IN ('YES','NO','NS')),
    CONSTRAINT pk_ServuserCnstr PRIMARY KEY (RoleType , IdSExp),
    CONSTRAINT fk_ServuserCnstr_Serv FOREIGN KEY (IdSExp)
    REFERENCES SERVICEXP ON DELETE CASCADE);

/* CONSTRAINTS ABOUT SERVICEXP */
CREATE TABLE SERV_APPLICATION_CNSTR (
    IdSExp VARCHAR2(10) NOT NULL,
    IdApplication VARCHAR2(15) NOT NULL,
    CONSTRAINT pk_ServAppliCnstr PRIMARY KEY (IdSExp, IdApplication),
    CONSTRAINT fk_ServAppliCnstr_Serv FOREIGN KEY (IdSExp)

```

## 108 Modèle Informationnel pour une session User-Centric : Inférence et Intégration Intelligente

---

```
REFERENCES SERVICEEXP ON DELETE CASCADE,  
CONSTRAINT fk_ServAppliCnstr_Application FOREIGN KEY (IdApplication)  
REFERENCES APPLICATION ON DELETE CASCADE);
```

```
CREATE TABLE SERV_CODECS_CNSTR (  
  IdExp      VARCHAR2(20),  
  IdCodecs   VARCHAR2(10) NOT NULL,  
  CONSTRAINT pk_ServCodecsCnstr PRIMARY KEY (IdExp, IdCodecs),  
  CONSTRAINT fk_ServCodecsCnstr_Serviceexp FOREIGN KEY (IdExp)  
    REFERENCES SERVICEEXP ON DELETE CASCADE,  
  CONSTRAINT fk_ServCodecsCnstr_Codecs FOREIGN KEY ( IdCodecs)  
    REFERENCES CODECS ON DELETE CASCADE);
```

```
/* CONSTRAINTS ABOUT NETWORK */  
CREATE TABLE SERV_NETWORK_CNSTR (  
  IdExp      VARCHAR2(20) NOT NULL,  
  IdNetwork   VARCHAR2(20) NOT NULL,  
  CONSTRAINT pk_ServNetworkCnstr PRIMARY KEY (IdExp, IdNetwork),  
  CONSTRAINT fk_ServNetworkCnstr_Serviceexp FOREIGN KEY (IdExp)  
    REFERENCES SERVICEEXP ON DELETE CASCADE,  
  CONSTRAINT fk_ServNetworkCnstr_Network FOREIGN KEY (IdNetwork)  
    REFERENCES NETWORK ON DELETE CASCADE);
```

```
/* CONSTRAINTS ABOUT EQUIPMENT*/  
CREATE TABLE SERV_MEMORY_CNSTR(  
  IdExp      VARCHAR2(10) NOT NULL,  
  IdMemory   VARCHAR2(10) NOT NULL,  
  CONSTRAINT pk_ServMemoryCnstr PRIMARY KEY (IdExp, IdMemory),  
  CONSTRAINT fk_ServMemoryCnstr_Serviceexp FOREIGN KEY (IdExp)  
    REFERENCES SERVICEEXP ON DELETE CASCADE,  
  CONSTRAINT fk_ServMemoryCnstr_Memory FOREIGN KEY (IdMemory)  
    REFERENCES MEMORY ON DELETE CASCADE);
```

```
CREATE TABLE SERV_OS_CNSTR (  
  IdExp      VARCHAR2(20) NOT NULL,  
  IdOs       VARCHAR2(10) NOT NULL,  
  CONSTRAINT pk_ServOsCnstr PRIMARY KEY (IdExp, IdOs),  
  CONSTRAINT fk_ServOsCnstr_Serviceexp FOREIGN KEY (IdExp)  
    REFERENCES SERVICEEXP ON DELETE CASCADE,  
  CONSTRAINT fk_ServOsCnstr_Os FOREIGN KEY (IdOs)  
    REFERENCES OS ON DELETE CASCADE);
```

Afin de tenir compte des informations de l'utilisateur, nous avons créé le Profil utilisateur, qui permet prendre en compte les informations de l'utilisateur:

```
CREATE TABLE PARTY (  
  IdParty     NUMBER(38) CONSTRAINT pk_User PRIMARY KEY,  
  /* PartyType = 1 si Client, PartyType = 2 si Fournisseur et PartyType = 3  
  si les deux */  
  PartyType   NUMBER(1) NOT NULL  
    CHECK(PartyType = 1 or PartyType =2 or PartyType = 3));
```

```
CREATE SEQUENCE SEQ_PARTY_IdParty  
  INCREMENT by 1  
  START WITH 1  
  NOMAXVALUE  
  NOCACHE  
  NOCYCLE;
```

```
CREATE TABLE ETABLISSEMENT (  
  siret      NUMBER(14) CONSTRAINT pk_Etablissement PRIMARY KEY,  
  RaisonSociale VARCHAR2(30) NOT NULL,  
  Sigle      VARCHAR2(30) NOT NULL,  
  IdParty    NUMBER(38) CONSTRAINT fk_Etablissement_PARTY  
    REFERENCES PARTY ON DELETE CASCADE);
```

```
CREATE TABLE PERSONNE (  
  IdPers     NUMBER(15) CONSTRAINT pk_Personne PRIMARY KEY,  
  LastName   VARCHAR2(30) NOT NULL,  
  MiddleName VARCHAR2(30),  
  FirstName  VARCHAR2(30) NOT NULL,
```

```

MaideName      VARCHAR2(30),
DateOfBirth    DATE NOT NULL,
IdParty        NUMBER(38) CONSTRAINT fk_Personne_PARTY
REFERENCES PARTY ON DELETE CASCADE);

CREATE TABLE HOME (
  IdParty        NUMBER(38) NOT NULL,
  IdAddress      NUMBER(38) NOT NULL,
  LevelSuite     VARCHAR2(7) NOT NULL,
  StreetNr       VARCHAR2(10) NOT NULL,
  POB            VARCHAR2(9),
CONSTRAINT pk_Home PRIMARY KEY (IdParty, IdAddress, LevelSuite, StreetNr),
CONSTRAINT fk_Home_PARTY FOREIGN KEY (IdParty)
REFERENCES PARTY ON DELETE CASCADE,
CONSTRAINT fk_Home_Address FOREIGN KEY (IdAddress)
REFERENCES ADDRESS ON DELETE CASCADE);

CREATE TABLE HOMELANG (
  IdParty        NUMBER(38) NOT NULL,
  IdAddress      NUMBER(38) NOT NULL,
  LevelSuite     VARCHAR2(7) NOT NULL,
  StreetNr       VARCHAR2(10) NOT NULL,
  HomeLang       VARCHAR2(40) NOT NULL,
  HomePref       NUMBER(2) NOT NULL,
CONSTRAINT pk_HomeLang PRIMARY KEY (IdParty, IdAddress, LevelSuite,
StreetNr, HomeLang),
CONSTRAINT fk_HomeLang_Home FOREIGN KEY (IdParty, IdAddress, LevelSuite,
StreetNr)
REFERENCES HOME ON DELETE CASCADE);

CREATE TABLE OFFICE (
  IdParty        NUMBER(38) NOT NULL,
  IdAddress      NUMBER(38) NOT NULL,
  LevelSuite     VARCHAR2(7) NOT NULL,
  StreetNr       VARCHAR2(10) NOT NULL,
  POB            VARCHAR2(9),
CONSTRAINT pk_Office PRIMARY KEY (IdParty, IdAddress, LevelSuite, StreetNr),
CONSTRAINT fk_Office_PARTY FOREIGN KEY (IdParty)
REFERENCES PARTY ON DELETE CASCADE,
CONSTRAINT fk_OfficeAddr FOREIGN KEY (IdAddress)
REFERENCES ADDRESS ON DELETE CASCADE);

CREATE TABLE OFFICELANG (
  IdParty        NUMBER(38) NOT NULL,
  IdAddress      NUMBER(38) NOT NULL,
  LevelSuite     VARCHAR2(7) NOT NULL,
  StreetNr       VARCHAR2(10) NOT NULL,
  OfficeLang     VARCHAR2(40) NOT NULL,
  OfficePref     NUMBER(2) NOT NULL,
CONSTRAINT pk_OfficeLang PRIMARY KEY (IdParty, IdAddress, LevelSuite, StreetNr,
OfficeLang),
CONSTRAINT fk_OfficeLang_Office FOREIGN KEY (IdParty, IdAddress, LevelSuite,
StreetNr)
REFERENCES OFFICE ON DELETE CASCADE);

CREATE TABLE ACTIVITY (
  IdParty        NUMBER(38) NOT NULL,
  ActivityType   VARCHAR2(12) NOT NULL,
  Beginning      DATE NOT NULL,
  Ending         DATE NOT NULL,
  Description    VARCHAR2(80),
  MeetingPlace  VARCHAR2(80),
  PhoneNumber    NUMBER(15) NOT NULL
CONSTRAINT fk_Activity_Tel REFERENCES TELEPHONE,
  TelExtension  NUMBER(5),
CONSTRAINT pk_Activity PRIMARY KEY (IdParty, ActivityType, Beginning),
CONSTRAINT fk_Activity_PARTY FOREIGN KEY (IdParty)
REFERENCES PARTY ON DELETE CASCADE);

CREATE TABLE ACTIVITYLANG (
  IdParty        NUMBER(38) NOT NULL,
  ActivityType   VARCHAR2(12) NOT NULL,
  Beginning      DATE NOT NULL,

```

## 110 Modèle Informationnel pour une session User-Centric : Inférence et Intégration Intelligente

```
ActivityLang      VARCHAR2 (40) NOT NULL,
ActivityPref      NUMBER(2) NOT NULL,
CONSTRAINT pk_ActivityLang PRIMARY KEY (IdParty, ActivityType, Beginning,
ActivityLang),
CONSTRAINT fk_ActivityLang_Activity FOREIGN KEY (IdParty,
ActivityType, Beginning)
REFERENCES ACTIVITY ON DELETE CASCADE);

CREATE TABLE PARTYPAY(
  IdParty         NUMBER(38) NOT NULL,
  IdPayment       NUMBER(38) NOT NULL,
  DueDate        DATE      NOT NULL,
  CONSTRAINT pk_UserPay PRIMARY KEY (IdParty, IdPayment),
CONSTRAINT fk_PARTYPay_PARTY FOREIGN KEY (IdParty)
REFERENCES PARTY ON DELETE CASCADE,
CONSTRAINT fk_PARTYPay_Payment FOREIGN KEY (IdPayment)
REFERENCES PAYMENT ON DELETE CASCADE);

CREATE TABLE EMAIL (
  EmailAddr      varchar2(50) CONSTRAINT pk_Email PRIMARY KEY,
  IdParty        NUMBER(38) NOT NULL,
CONSTRAINT fk_Email_PARTY FOREIGN KEY (IdParty)
REFERENCES PARTY ON DELETE CASCADE);

CREATE TABLE PREF_EQPT_HOME (
  IdParty        NUMBER(38) NOT NULL,
  IdEquipment    VARCHAR2(40) NOT NULL,
  IdAddress      NUMBER(38) NOT NULL,
  LevelSuite     VARCHAR2(40) NOT NULL,
  StreetNr      VARCHAR2(10) NOT NULL,
  EqptHomePref  NUMBER(2) NOT NULL,
CONSTRAINT pk_PrefEqptHome PRIMARY KEY (IdParty, IdEquipment, IdAddress,
LevelSuite, StreetNr),
/* CONSTRAINT fk_PrefEquiHome_PARTY FOREIGN KEY (IdParty)
REFERENCES PARTY ON DELETE CASCADE), */
CONSTRAINT fk_PrefEqptHome_Home FOREIGN KEY (IdParty, IdAddress, LevelSuite ,
StreetNr)
REFERENCES HOME ON DELETE CASCADE,
CONSTRAINT fk_PrefEqptHome_Equipment FOREIGN KEY (IdEquipment)
REFERENCES EQUIPMENT ON DELETE CASCADE);

CREATE TABLE PREF_SERVICE_HOME (
  IdParty        NUMBER(38) NOT NULL,
  IdSExp        VARCHAR2(40) NOT NULL,
  IdAddress      NUMBER(38) NOT NULL,
  LevelSuite     VARCHAR2(40) NOT NULL,
  StreetNr      VARCHAR2(10) NOT NULL,
  EqptServicePref NUMBER(2) NOT NULL,
CONSTRAINT pk_PrefServiceHome PRIMARY KEY (IdParty, IdSExp, IdAddress, LevelSuite,
StreetNr),
/* CONSTRAINT fk_PrefEquiHome_PARTY FOREIGN KEY (IdParty)
REFERENCES PARTY ON DELETE CASCADE), */
CONSTRAINT fk_PrefServiceHome_Home FOREIGN KEY (IdParty, IdAddress, LevelSuite,
StreetNr)
REFERENCES HOME ON DELETE CASCADE,
CONSTRAINT fk_PrefServiceHome_Service FOREIGN KEY (IdSExp)
REFERENCES SERVICEEXP ON DELETE CASCADE);

CREATE TABLE PREF_NETWORK_HOME (
  IdParty        NUMBER(38) NOT NULL,
  IdNetwork     VARCHAR2(40) NOT NULL,
  IdAddress      NUMBER(38) NOT NULL,
  LevelSuite     VARCHAR2(40) NOT NULL,
  StreetNr      VARCHAR2(10) NOT NULL,
  NetHomePref   NUMBER(2) NOT NULL,
CONSTRAINT pk_PrefNetworkHome PRIMARY KEY (IdParty, IdNetwork, IdAddress,
LevelSuite, StreetNr),
/* CONSTRAINT fk_PrefEquiHome_PARTY FOREIGN KEY (IdParty)
REFERENCES PARTY (IdParty)
-- ON DELETE CASCADE), */
CONSTRAINT fk_PrefNetworkHome_Home FOREIGN KEY (IdParty, IdAddress, LevelSuite ,
StreetNr)
REFERENCES HOME ON DELETE CASCADE,
CONSTRAINT fk_PrefNetworkHome_Network FOREIGN KEY (IdNetwork)
```

```

REFERENCES NETWORK ON DELETE CASCADE);

CREATE TABLE PREF_EQPT_OFFICE (
  IdParty      NUMBER(38) NOT NULL,
  IdEquipment  VARCHAR2(40) NOT NULL,
  IdAddress    NUMBER(38) NOT NULL,
  LevelSuite   VARCHAR2(40) NOT NULL,
  StreetNr     VARCHAR2(10) NOT NULL,
  Pref         NUMBER(2) NOT NULL,
  CONSTRAINT pk_PrefEqptOffice PRIMARY KEY (IdParty, IdEquipment, IdAddress,
  LevelSuite, StreetNr),
  /* CONSTRAINT fk_PrefEquiHome_PARTY FOREIGN KEY (IdParty)
  REFERENCES PARTY ON DELETE CASCADE ), */
  CONSTRAINT fk_PrefEqptOffice_Office FOREIGN KEY (IdParty, IdAddress, LevelSuite,
  StreetNr)
  REFERENCES OFFICE ON DELETE CASCADE,
  CONSTRAINT fk_PrefUtiOffice_Equipment FOREIGN KEY (IdEquipment)
  REFERENCES EQUIPMENT ON DELETE CASCADE);

CREATE TABLE PREF_SERVICE_OFFICE (
  IdParty      NUMBER(38) NOT NULL,
  IdSExp       VARCHAR2(40) NOT NULL,
  IdAddress    NUMBER(38) NOT NULL,
  LevelSuite   VARCHAR2(40) NOT NULL,
  StreetNr     VARCHAR2(10) NOT NULL,
  ServiceOfficePref NUMBER(2) NOT NULL,
  CONSTRAINT pk_PrefServiceOffice PRIMARY KEY (IdParty, IdSExp, IdAddress, StreetNr),
  /* CONSTRAINT fk_PrefEquiHome_PARTY FOREIGN KEY (IdParty)
  REFERENCES PARTY ON DELETE CASCADE ), */
  CONSTRAINT fk_PrefServiceOffice_Office FOREIGN KEY (IdParty, IdAddress,
  LevelSuite, StreetNr)
  REFERENCES OFFICE ON DELETE CASCADE,
  CONSTRAINT fk_PrefServiceOffice_Service FOREIGN KEY (IdSExp)
  REFERENCES SERVICEEXP ON DELETE CASCADE);

CREATE TABLE PREF_NETWORK_OFFICE (
  IdParty      NUMBER(38) NOT NULL,
  IdNetwork    VARCHAR2(40) NOT NULL,
  IdAddress    NUMBER(38) NOT NULL,
  LevelSuite   VARCHAR2(40) NOT NULL,
  StreetNr     VARCHAR2(10) NOT NULL,
  NetworkOfficePref NUMBER(2) NOT NULL,
  CONSTRAINT pk_PrefNetworkOffice PRIMARY KEY (IdParty, IdNetwork, IdAddress,
  StreetNr),
  /* CONSTRAINT fk_PrefEquiHome_PARTY FOREIGN KEY (IdParty)
  REFERENCES PARTY ON DELETE CASCADE), */
  CONSTRAINT fk_PrefNetworkOffice_Office FOREIGN KEY (IdParty, IdAddress,
  LevelSuite, StreetNr)
  REFERENCES OFFICE ON DELETE CASCADE,
  CONSTRAINT fk_PrefNetworkOffice_Network FOREIGN KEY (IdNetwork)
  REFERENCES NETWORK ON DELETE CASCADE);

CREATE TABLE PREF_EQPT_ACTIVITY(
  IdParty      NUMBER(38) NOT NULL,
  IdEquipment  VARCHAR2(40) NOT NULL,
  ActivityType  VARCHAR2(12) NOT NULL,
  Beginning    DATE NOT NULL,
  EqptActivityPref NUMBER(2) NOT NULL,
  CONSTRAINT pk_PrefEqptActivity PRIMARY KEY
  (IdParty, IdEquipment, ActivityType, Beginning),
  /* CONSTRAINT fk_PrefEquiHome_PARTY FOREIGN KEY (IdParty)
  REFERENCES PARTY ON DELETE CASCADE), */
  CONSTRAINT fk_PrefEqptActivity_Activity FOREIGN KEY (IdParty,
  ActivityType, Beginning)
  REFERENCES ACTIVITY ON DELETE CASCADE,
  CONSTRAINT fk_PrefEqptActivity_Equipment FOREIGN KEY (IdEquipment)
  REFERENCES EQUIPMENT ON DELETE CASCADE);

CREATE TABLE PREF_SERVICE_ACTIVITY(
  IdParty      NUMBER(38) NOT NULL,
  IdSExp       VARCHAR2(40) NOT NULL,
  ActivityType  VARCHAR2(12) NOT NULL,
  Beginning    DATE NOT NULL,
  ServiceActivityPref NUMBER(2) NOT NULL,

```

```

CONSTRAINT pk_PrefServiceActivity PRIMARY KEY
(IdParty,IdSExp,ActivityType,Beginning),
CONSTRAINT fk_PrefServiceActivity_Activit FOREIGN KEY (IdParty,
ActivityType,Beginning)
REFERENCES ACTIVITY ON DELETE CASCADE,
CONSTRAINT fk_PrefServiceActivity_Service FOREIGN KEY (IdSExp)
REFERENCES SERVICEEXP ON DELETE CASCADE);

CREATE TABLE PREF_NETWORK_ACTIVITY(
  IdParty      NUMBER(38) NOT NULL,
  IdNetwork    VARCHAR2(40) NOT NULL,
  ActivityType VARCHAR2(12) NOT NULL,
  Beginning    DATE NOT NULL,
  NetworkActivityPref NUMBER(2) NOT NULL,
CONSTRAINT pk_PrefNetworkActivity PRIMARY KEY
(IdParty,IdNetwork,ActivityType,Beginning),
/* CONSTRAINT fk_PrefEquiHome_PARTY FOREIGN KEY (IdParty1)
REFERENCES PARTYON DELETE CASCADE), */
CONSTRAINT fk_PrefNetworkActivity_activit FOREIGN KEY (IdParty,
ActivityType,Beginning)
REFERENCES ACTIVITY ON DELETE CASCADE,
CONSTRAINT fk_PrefNetworkActivity_Network FOREIGN KEY (IdNetwork)
REFERENCES NETWORK ON DELETE CASCADE);

CREATE TABLE PARTYROLE (
  IdParty      NUMBER(38) NOT NULL,
  RoleType     VARCHAR2(40),
  Beginning    DATE NOT NULL,
  Ending       DATE NOT NULL,
  CONSTRAINT pk_PartyRole PRIMARY KEY (IdParty, RoleType),
CONSTRAINT fk_PartyRole_Party FOREIGN KEY (IdParty)
REFERENCES PARTY ON DELETE CASCADE);

CREATE TABLE METHODCONTACT (
  IdParty      NUMBER(38) NOT NULL,
  RoleType     VARCHAR2(40) NOT NULL,
  MethCont     VARCHAR2(60) NOT NULL,
  MethContPref NUMBER(2) NOT NULL,
CONSTRAINT pk_MethodContact PRIMARY KEY (IdParty, RoleType, MethCont),
CONSTRAINT fk_MethodContact_Party FOREIGN KEY (IdParty)
REFERENCES PARTY ON DELETE CASCADE);

```

### Triggers

Il existe deux types de triggers différents en Oracle : les triggers de table (STATEMENT) et les triggers de ligne (ROW).

*Les triggers de table* sont exécutés une seule fois lorsque des modifications surviennent sur une table (même si ces modifications concernent plusieurs lignes de la table). Ils sont utiles si des opérations de groupe doivent être réalisées (comme le calcul d'une moyenne, d'une somme totale, d'un compteur). Pour des raisons de performance, il est préférable d'employer ces triggers plutôt que les triggers lignes.

*Les triggers lignes* sont exécutés « séparément » pour chaque ligne modifiée dans la table. Ils sont très utiles s'il faut mesurer une évolution pour certaines valeurs, effectuer des opérations pour chaque ligne en question.

Le trigger se déclenche lorsqu'un événement précis survient : BEFORE UPDATE, AFTER DELETE, AFTER INSERT, ... Ces événements sont importants car ils définissent le moment d'exécution du trigger. Ainsi, lorsqu'un

---

trigger BEFORE INSERT est programmé, il sera exécuté juste avant l'insertion d'un nouvel élément dans la table.

Le trigger Oracle doit être écrit en PL/SQL. La forme d'un trigger est encore très dépendante du SGBD<sup>32</sup> utilisé.

Voici un exemple de trigger ligne Oracle :

```
CREATE OR REPLACE TRIGGER Print_salary_changes
  BEFORE UPDATE ON Emp_tab
  FOR EACH ROW
  WHEN (new.Empno > 0)
  DECLARE
    sal_diff number;
  BEGIN
    sal_diff := :new.sal - :old.sal;
    dbms_output.put(' Old : ' || :old.sal || 'New : '
  || :new.sal || 'Difference : ' || sal_diff);
  END ;
```

Ce trigger est déclenché lorsque la table Emp\_tab est mise à jour. Pour chaque modification (lignes mises à jour), le trigger va calculer puis afficher respectivement l'ancien salaire, le nouveau salaire et la différence entre ces deux salaires. La condition précisée (WHEN new.Empno > 0) restreint le déclenchement du trigger (exécuté uniquement si new.Empno > 0).

Nous avons utilisés de triggers pour implémenter l'inférence dans notre base informationnelle. C'est à travers les triggers que nous pouvons faire les notifications des événements reçus et qui ont été produits par le changement de QoS d'une ressource ou par les différents types de mobilité.

## **V.6 Conclusions**

Revenons au cas particulier d'IMS, nous considérons que une session IMS doit pouvoir s'adapter à toutes les mobilités et satisfaire le SLA de l'utilisateur. La solution que nous avons proposé est basée sur une structure informationnelle capable de répondre aux besoins, et une organisation pour un contexte transorganisationnel.

En effet, l'open source IMS a été utilisé comme couche de contrôle du démonstrateur en faisant évoluer la structure informationnelle. Il est à noter que la version actuelle d'IMS possède quelques limitations.

---

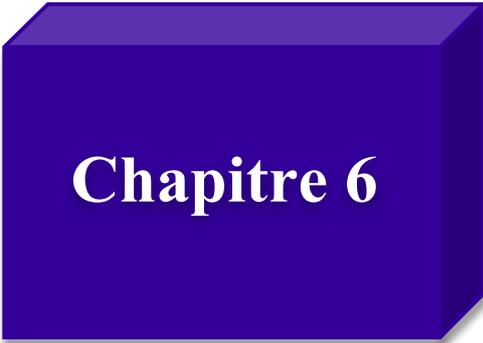
<sup>32</sup> Système de Gestion de Base de Données

L'IMS amélioré que nous proposons contient une base de données HSS (Home Subscriber Server) évoluée pour offrir des informations complètes mise à jour dynamiquement en temps réel (par exemple les préférences des utilisateurs, les terminaux des utilisateurs, les réseaux auxquels avaient souscrits les utilisateurs, les services etc...).

Le HSS amélioré ou HSS+ joue le rôle de base de connaissances, capable de prendre les décisions, non seulement lors de l'ouverture de la session mais aussi durant tout son déroulement, afin de satisfaire la QoS et les préférences de l'utilisateur. C'est donc à travers cette base de connaissance que l'on compte avec l'image réelle du système d'information de l'utilisateur en fonction de sa localisation et activité.

En effet, l'aspect « user-centric » est un besoin primordial dans lequel toute demande de service nécessite la personnalisation de l'utilisateur, l'accessibilité par n'importe quel terminal, une meilleure connectivité, une continuité dans le service et un provisionnement dynamique de ressources. Toutes ces exigences nécessitent une inférence informationnelle et non pas une simple mise à jour de la base de données. Pour cela, on a proposé une base de connaissances (Infosphere/Infoware) afin de tenir compte dynamiquement et en temps réel de l'information personnalisée et des ressources ambiantes.

Cette base de connaissance structurée a des capacités d'inférence pour le traitement de cette information, c'est-à-dire, elle est autodynamique et peut faire l'autogestion des services d'après les préférences de l'utilisateur.



## Chapitre 6

# Conclusion générale & perspective

Nous montrons ci-dessous finalement les conclusions générales (§VI.1) de cette thèse, qui était principalement focalisée sur la partie informationnelle à des fins de gestion dynamique de la session utilisateur dans notre contexte NGN. Ensuite, nous présentons les perspectives de ce travail de recherche (§VI.2).

## ***VI.1 Conclusion générale***

Le but de cette thèse, a été de proposer des nouvelles solutions, permettant à l'utilisateur d'avoir, une session sans coupure et sans couture dans le contexte NGN. Ce nouveau contexte est caractérisé par, la gestion des différentes mobilités auxquelles, l'utilisateur d'aujourd'hui peut être confrontés lorsqu'il accède à son service. Cette mobilité peut être au niveau utilisateur, terminal, réseau et service. Une autre caractéristique dans ce contexte est l'hétérogénéité à travers laquelle l'utilisateur est aussi confronté que ce soit au niveau du terminal, du réseau et du service. Cette problématique fait donc que l'on s'intéresse à comment fournir l'information nécessaire au bon moment et au bon endroit dans le but de la gestion. Pour cette raison, nous nous sommes orientés dans cette thèse vers le modèle informationnel pour proposer une structure informationnelle de façon efficace. En effet, c'est dans ce nouvel environnement qu'il a fallu proposer une architecture qui permette à l'utilisateur de compter avec l'information nécessaire pour l'accès à son service afin d'avoir une continuité de sa session.

Tout d'abord, nous avons fait une étude des travaux au niveau de la gestion de l'information. Ceci nous a permis d'identifier les manques pour arriver à faire une gestion dynamique de la session de l'utilisateur. Nous avons regardé les propositions au niveau des organismes de standardisation, en ce qui concerne le modèle informationnel. Concernant la structure informationnelle, nous avons regardé la structure du HSS d'IMS d'aujourd'hui.

La première contribution que nous avons proposé est une base de connaissance coté opérateur que nous appelons Infoware. Cette Base de Connaissance contient l'image du système d'information de l'utilisateur en temps réel et qui est stocké chez l'opérateur. Ceci permet à l'utilisateur de compter avec l'information dont il a besoin tenant compte sa localisation et activité. Ce avec ces deux derniers facteurs (localisation et activité de l'utilisateur) que nous utilisons afin d'obtenir toujours les moyens de communication disponibles à l'utilisateur et ainsi que les réseaux et services dont il peut avoir l'accès. Cette base de connaissance prend en compte les événements du réel à travers des agents de QoS, ceci permet d'avoir une image à jour du système d'information de l'utilisateur.

---

La deuxième contribution a été de proposer une structure informationnelle dans le but de la gestion de ressources que l'on a dénommé Real Time Profile. Cette structure informationnelle nous a permis de pouvoir gérer une ressource ou un réseau de ressources du même niveau de visibilité. Cette gestion nous a permis d'homogénéiser les composants afin de qu'ils puissent se communiquer et informer leur QoS à d'autres ressources. Le fait d'avoir la connaissance de la QoS de chaque ressource permet d'avoir une session dynamique, car chaque ressource informe la QoS dont il a besoin pour fonctionner. C'est en ayant la même structure que les composants peuvent se communiquer, que se soit au niveau horizontal (vision réseau) ou au niveau vertical (session), car la même structure est mise dans chaque composant à tous les niveaux de visibilité.

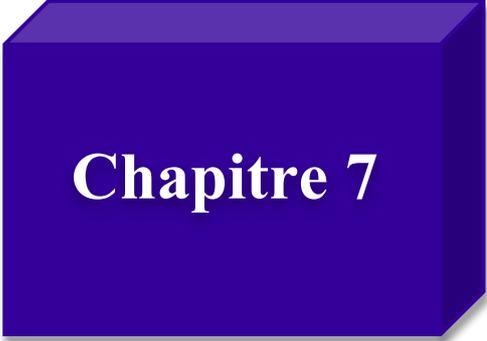
La troisième proposition est un modèle organisationnel que nous avons appelé VPIN pour Virtual Private Information Network. Le VPIN nous permet d'avoir la vision globale de tous les composants qui font partie du système d'information de l'utilisateur (vision user-centric). Cette organisation permet de répondre aux besoins de l'utilisateur en éliminant les obstacles techniques qui ne permettent pas à un utilisateur d'accéder à son service. Le VPIN prend en compte les préférences de l'utilisateur, ce qui permet à l'utilisateur d'avoir des services personnalisés. Le fait d'avoir la connaissance de tous les ressources permet de prendre la bonne décision correspondant aux exigences de l'utilisateur.

Nous avons validé nos propositions à travers une plateforme qui est composé d'une infrastructure réseau (Virtuors) ainsi que l'architecture IMS. Nous avons utilisé l'IMS dans lequel, nous avons modifié la structure informationnelle (HSS) par un HSS+. Nous avons proposé d'améliorer le HSS car si bien la session est bien gérée au plan contrôle, on a constaté que le niveau informationnel n'est pas assez riche pour personnaliser et s'adapter dynamiquement à tous les changements. Cette nouvelle structure change donc la façon de fonctionner du HSS d'aujourd'hui. Au lieu qu'il reste base de données où sa fonction principale est de stocker les informations, le HSS+ ou base de connaissance sera capable d'avoir l'image réel du système d'information de l'utilisateur.

Cette nouvelle base contient au delà de la descriptions des composants des informations de comportement à travers les informations de QoS. Ce qui nous fait dire que nous constituons une base de connaissance. Cette base de connaissance est basée sur le modèle informationnel qui a une structure bien définie et permet d'avoir l'information nécessaire au bon moment et au bon endroit. C'est donc cette base de connaissance à travers les inférences qu'il pourra assurer les adaptations dynamiques et rendre la session IMS/NGN plus efficace.

## ***VI.2 Perspectives***

Comme perspectives de cette thèse sur le plan informationnel, nous devons intégrer cette contribution au reste des travaux de notre équipe de recherche, tel que la partie fonctionnel, et protocolaire. Ceci permettra d'avoir un démonstrateur de bout en bout avec des composants autonomes. Au niveau protocolaire, il faudra regarder le SIP pour enrichir les informations qu'il transporte, car c'est SIP qui est utilisé pour la signalisation dans notre contexte IMS/NGN. Il reste un grand axe de recherche à explorer tel que les ontologies. Les ontologies nous permettent de vérifier que l'existence de quelque chose est bien c'est qu'il décrit d'y être.



**Chapitre 7**

**Références  
Bibliographiques**

- [1] Common Information Model, CIM Schema V. 2.26.1 21/07/2010
- [2] Directory Enable Network, extension of the Common Information Model (CIM) of the Desktop Management Task Force (DMTF)
- [3] Generic Network Information Model, Recommendation ITU-T M.3100
- [4] TMF GB922, Shared Information/Data (SID) Model, Business View Concepts, Principles, and Domains (Release 7.5), Mai 2008.
- [5] TMF GB922, Shared Information/Data (SID) Model, Addendum 4SO-Service Overview Business Entity Definitions (Release 8), Juillet 2008.
- [6] TMF GB922, Shared Information/Data (SID) Model, Addendum 5LR-LogicalResource and CompoundResource Business Entity Definitions (Release 8), Juillet 2008.
- [7] Towards a Context-Aware Information Model, Yeongrak Choi, Jian Li, Yoonseon Han, John Strassner, and James Won-Ki Hong, MACE conférence 2010.
- [8] Ontology-based capability self-configuration, Patcharee Thongtra, Finn Arve Aagesen and Natenapa Sriharee, IFIP conférence 2008
- [9] 3GPP TS 23.228 V8.3.0 “IP multimedia subsystem; Stage 2”; (Release 8), 2008
- [10] RFC 3261 – SIP : Session Initiation Protocol
- [11] RFC 3588 – Diameter Base Protocol
- [12] HSS d’IMS Fokus :  
[http://www.fokus.fraunhofer.de/en/fokus\\_testbeds/open\\_ims\\_playground/components/osims/fhoss/index.html](http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_ims_playground/components/osims/fhoss/index.html)
- [13] Jean-Yves Prax, Livre “Le guide du knowledge management” Concepts et pratiques du management de la connaissance. Collection : Stratégies et Management, Dunod – EAN13 : 9782100531493
- [14] Z.Benahmed Daho, N.Simoni, M.Chevanne, S.Betgé-Brezetz, “An information model for service and network management integration: from needs towards solutions”, IEEE/IFIP NOMS, Korea, 2004
- [15] “*Generic Information Model*”, ALCATEL Project – 2 – Task 3, NGNSM, 2003
- [16] F.Bennani, “IP et la QoS : vers une maîtrise dynamique de bout en bout”, Thèse de doctorat, ENST/INFRES, 2002
- [17] "VPIN: An Event Based Knowledge Inference for a User Centric Information System" International Journal on Advances in Internet Technology IARIA, 2009 vol. 2 nr. 1
- [18] Netzahualcoyotl Ornelas, Noémie Simoni, Ken Chen, Antoine Boutignon "VPIN: User-Session Knowledge Base for Self-Management of Ambient Networks" UBICOMM.08, Sep. 2008, Valencia, Spain

---



**Annexe A**

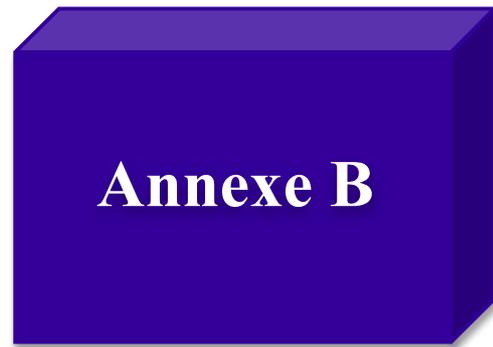
# **Liste des publications**

Netzahualcoyotl Ornelas, Noémie Simoni, Gladys Diaz, Ken Chen  
"Vers une Base de Connaissance infèrente IMS : du modèle  
conceptuel au modèle physique"  
GRES 2010, Oct. 2010, Montréal, Canada

Netzahualcoyotl Ornelas, Noémie Simoni, Chunyang Yin, Antoine  
Boutignon  
"VPIN: An Event Based Knowledge Inference for a User Centric  
Information System"  
International Journal on Advances in Internet Technology  
IARIA, 2009 vol. 2 nr. 1

Netzahualcoyotl Ornelas, Noémie Simoni, Ken Chen, Antoine  
Boutignon  
"VPIN: User-Session Knowledge Base for Self-Management of  
Ambient Networks" (*Best paper award*)  
UBICOMM.08, Sep. 2008, Valencia, Spain

Netzahualcoyotl Ornelas, Noémie Simoni, Antoine Boutignon  
"HSS et DNS pour un IMS efficace"  
GRES.07, Nov. 2007, Hammamet, Tunisia



# **Mise en place de la plateforme OS-IMS**

## Implémentation de l'Open Source IMS (Fokus)

Ce document présente la mise en place du Cœur-IMS Release 7. Pour plus d'informations, vous pouvez consulter l'adresse : <http://www.openimscore.org/>  
Le cœur IMS est installé sur une version d'OS type linux. Dans vos machines, vous pouvez le faire soit en faisant une partition dans le disque dur pour avoir deux OS (normalement Windows et linux) soit vous faite l'installation en utilisant des machines virtuelles type VMware (<http://www.vmware.com/fr/download/>).

L'implémentation est divisée en trois partis. La première (§B.1) indique l'installation du système opératif dans le quel j'ai installé le cœur IMS. La deuxième partie (§B.2) indique la procédure pour installer le cœur Open Source IMS. La troisième partie (§B.3) montre la configuration du cœur IMS pour son fonctionnement.

### B.1 Installation Ubuntu 8.04 et paquets nécessaires

L'installation peut être faite comme décrit ci-dessous :

On a installé la version 8.04 d'Ubuntu et partitionné le disque dur comme suit :

Home	64 Gigabytes	(/home)	et de type ext3 pour les données
ROOT	15 Gigabytes	(/)	et de type ext3 pour le système d'exploitation
SWAP	3 Gigabytes		

L'espace destiné aux différentes partitions dépend de l'espace de votre disque dur.

Et puis on a précisé le nom de l'utilisateur, de l'ordinateur, de la racine et du mot de passe.

On peut utiliser la deuxième option pour faire la partition automatique.

Il est conseillé de faire une mise à jour des fichiers de Système (pour la sécurité), on peut le faire de la façon suivante : (Il faut compter avec une connexion à Internet)

- 1.- Aller au Menu « Applications », cliquer sur «Ajouter/supprimer... ».
- 2.- Sur la fenêtre qui apparait, aller à l'option « Préférences » (il faut mettre le mot de passe de root).
- 3.- Aller sur l'onglet «Mises à jour ».
  - a) Cocher les deux premières options :
    - Mises à jour de sécurité importantes (gutsy-security)
    - Mises à jour recommandées (gutsy-updates)
- 4.- Cliquer sur fermer
- 5.- Sur l'écran dans la partie supérieure à droite, il y aura une petite fenêtre jeune qui demandera de faire les mises à jour, cliquer sur cette fenêtre et procéder à faire la mise à jour.
- 6.- Il faut redémarrer le système.

---

## Installation des fichiers indispensables à l'installation du cœur

Avant l'installation du cœur IMS (OpenIMSCore) il faut installer les packages suivants :

Les prés requis sont les suivants :

- Un serveur DNS
  - Typiquement BIND
- 100 Mo d'espace disque environ
- GCC3/4
- JDK 1.5 ou supérieur (sur le site est marqué à partir de 1.5, mais je vous conseille d'utiliser la version 1.6, car j'ai eu quelques problèmes de compilation avec la version 1.5)
- bison
- flex
- ant
- Un SGBD (Système de Gestion de Bases de Données)
  - Le projet "OSIMS" est prévu pour fonctionner avec MySQL
- Les librairies de développement libxml2, libmysql
  - Il est conseillé d'installer toutes les librairies libxml2-.....
- Linux, noyau 2.6
- Le package ipsec-tools pour pouvoir établir les associations de sécurité
- Un navigateur Internet destiné à l'accès aux bases de données utilisateur (HSS)
- Subversion pour le téléchargement des sources à jour

Pour regarder les paquets qui sont installés après la mise à jour, il se peut vérifier avec la commande :

```
user# sudo dpkg --get-selections > «nom_fichier»
```

«nom\_fichier» : C'es le nom du fichier où sera gardé la liste des paquets installés, ça peut être « paquets.txt ».

Avant de procéder l'installation de paquets, il est conseillé de faire la mise à jour de l'apt-get :

```
user#sudo apt-get upgrade
```

On peut ouvrir le fichier « source.list » et enlever « # » sur les urls pour pouvoir faire les mises à jours.

```
user# sudo gedit /etc/apt/source.list
```

Après on fait la mise à jour :

```
user# sudo apt-get update
```

S'il y a des erreurs, vous pouvez faire :

```
user#sudo apt-get update --fix-missing
```

**user#** sudo updatedb (Pour mettre à jour la liste des paquets installés).

Pour installer les paquets, on fait comme suit :

**- Le langage C : GCC 3.4**

**user#** sudo apt-get install gcc

**- Paquet ANT**

**root#** apt-get install ant

**- Paquet Bison**

**root#** apt-get install bison

**- Paquet Flex**

**root#** apt-get install flex

**- La base de données MySQL**

On utilise la commande **synaptic** (en mode root) et on cherche les fichiers :

mysql-client5.0, mysql-serveur5.0, libmysql, libxml2 et on les sélectionne puis on les installe en cliquant sur appliquer.

On peut installer bind9 aussi à travers synaptic

**- Le DNS : Bind9**

**user#** sudo apt-get install bind9

ATTENTION : Si le paquet ne peut pas être installé avec apt-get, il peut être téléchargé d'une page web, par exemple :

<http://telecharger.yacapa.com/telechargement/34274.html#>

Avant d'installer, il faut s'assurer d'avoir les librairies de compilation, pour ce faire taper la commande :

**user#** sudo apt-get install build-essential

**user#** sudo apt-get install libc6-dev

Pour l'installation, placer le fichier téléchargé dans le répertoire : /usr/bin/

Après on suit les pas suivants :

**root#** tar xvf <nom\_fichier>.tar.gz

**root#** cd <dossier\_du\_programme\_> (Dans ce cas le nom est : bind-9.2.3)

**root#** ./configure

**root#** make

**root#** make install

Il faut le JDK de java avec une version supérieure ou égale à 1.6

Vérifier avec la commande :

**user#** java -version

---

Si jamais, la version de JAVA est inférieure ou égal à la 1.5, on peut faire l'installation d'une nouvelle version.

On installe le java runtime environment:

```
user# sudo apt-get install sun-java6-bin sun-java6-jre  
Et on n'oublie pas d'accepter les termes de licence.
```

Puis on installe java développement kit (JDK) :

```
user# sudo apt-get install sun-java6-jdk
```

On vérifie que la nouvelle version de java est celle qu'on a installée:

```
user# java -version (et comme résultat à la commande on aura la version 1.6 de java)
```

ATTENTION : S'il y a des erreurs dans l'installation, il peut être téléchargé le JDK de la page suivante :

<http://java.sun.com/javase/downloads/index.jsp> (Sélectionner une version de Java à installer supérieur à 1.5).

Une fois téléchargé, mettre le fichier (.bin ou .sh) dans le répertoire de l'utilisateur :

```
user# mv <fichier_Java_installer>.bin /home/user
```

Ensuite, changer les permis du fichier avec la commande:

```
user# chmod 755 <fichier_Java_installer>.bin
```

Après faire l'installation en exécutant le fichier comme on décrit ci-dessous:

```
user# ./<fichier_Java_installer>.bin
```

```
user# ./jdk-6linux.bin (exemple)
```

Note : Il faut accepter les conditions d'installations.

Une fois installé, l'installation aura créé un répertoire avec le nom de version de java installé, dans ce cas là, il s'appelle : jdk1.6.0\_03

On peut bouger le répertoire au : « /opt/ » en tapant : (il faut être en mode root)

Si le répertoire « opt » n'existe pas, on peut le créer et tapant :

```
root# cd /
```

```
root# mkdir opt
```

Après on bouge le JD au répertoire opt

```
root# mv /jdk1.6.0_03 /opt/
```

Pour configurer les paths de java on ouvre le fichier .bashrc de root.

```
root# gedit /root/.bashrc
```

Une fois ouvert, on met le suivant :

```
export JAVA_HOME="/opt/jdk1.6.0_03"
export JDK_HOME="${JAVA_HOME}"
export PATH="${JAVA_HOME}/bin:${PATH}"
```

Après avoir fait ces changements, il faut fermer le terminal et re-ouvrir une autre pour qu'il prenne en charge les changements.

Après avoir installé les différents packages il faut démarrer la base de données pour tester s'il fonctionne bien :

```
root# sudo /etc/init.d/mysql start
```

**NOTE :** Pour la configuration du BIND9, on doit le faire presque à la fin, une fois que on a téléchargé tout le code dont on en a besoin.

## A.2 Installation de l'Open Source Cœur IMS

Il faut télécharger avec svn (subversion) le code source du cœur IMS (C-SCSF et HSS), mais d'abord il faut installer le svn.

```
user# sudo apt-get install subversion
```

Pour installer le cœur IMS, on le met dans le répertoire appelé « /opt/OpenIMSCore » et on accède au répertoire crée. Normalement le répertoire « opt » a été déjà créé, donc on va créer juste OpenIMSCore.

```
root# cd /opt
root# mkdir OpenIMSCore
root# cd OpenIMSCore
```

Dans le répertoire OpenIMSCore on crée deux répertoires : ser\_ims et FHoSS  
Dans le premier « ser\_ims » on mettra le CSCF :

```
root# mkdir ser_ims
```

Une fois crée le répertoire ser\_ims, on télécharge les modules avec la commande :

```
root# svn checkout http://svn.berlios.de/svnroot/repos/openimscore/ser_ims/trunk
ser_ims
```

**ATTENTION :** Il y a bien un espace dans l'URL, ca veut dire que le code sera téléchargé dans le répertoire «ser\_ims».

Et dans le deuxième répertoire «FHoSS » on mettra le HSS :

```
root# mkdir FHoSS
```

---

```
root# svn checkout http://svn.berlios.de/svnroot/repos/openimscore/FHoSS/trunk
FHoSS
```

**ATTENTION :** Il y a bien un espace dans l'URL, ca veut dire que le code sera téléchargé dans le répertoire «FHoSS».

**Une fois téléchargé, on fait l'installation et compilation du code source (en mode root).**

Voici la compilation du **ser\_ims** :

```
root# cd /opt/OpenIMSCore/ser_ims
root# make install-libs all
root# cd ..
```

Voici la compilation du **FHoSS** :

```
root# cd /opt/OpenIMSCore/FHoSS
root# ant compile deploy
root# cd ..
```

Si jamais il y a des erreurs dans la compilation du cscf ou hss, ça veut dire qu'il manque un des prés requis nécessaires (section 1).

### **B.3 Configuration du Cœur IMS de Fokus**

Maintenant, on passe à la configuration de l'environnement :

Normalement la configuration se fait d'abord pour que ça marche dans un réseau local. Après, si dans votre entité vous pouvez modifier les fichiers de configuration du DNS, vous pouvez réaliser la configuration pour que le cœur IMS marche et soit accessible à travers l'Internet.

A partir de maintenant, dépendant de votre entité, soit on est en réseau local, soit on est en réseau publique, il est conseillé tout d'abord d'utiliser le cœur IMS en réseau privé pour faire les tests.

**- Pour configurer le CSCF :**

On exécute le fichier configurator.sh

```
root# ./ser_ims/cfg/configurator.sh
```

**Après on remplace les informations suivantes**

Domain Name = open-ims.test

IP Address = 192.168.1.50 (ou l'adresse local à utiliser)

Sur la question qui suit, on écrit « all » et après entrée

**File to change ["all" for everything, "exit" to quit]: all <enter>**

Et par conséquent les fichiers icscf, pescf, scscf et mysql seront changés et mis à jour par rapport aux changements qu'on avait faits dans le fichier configurator.sh

### **Pour configurer le HSS :**

```
root# cd /opt/OpenIMSCore/FHoSS/deploy
root# gedit DiameterPeerHSS.xml
```

Dans ce fichier, on change la valeur du bind à l'adresse IP qu'on avait choisi pour notre cœur IMS :

```
<Accept port=« 3868 » bind = «192.168.1.50 » >
```

Après on garde les changements et on quitte le fichier.

### **Pour configurer le DNS :**

Comme il a été indiqué, après avoir installé le paquet bind, on va copier le fichier de configuration du dnszone dans le répertoire du bind comme décrit ci-dessous :

```
root# cd /etc/bind
root# cp /opt/OpenIMSCore/ser_ims/cfg/open-ims.dnszone .
```

**Attention :** Il y a bien un point dans la ligne d'au-dessus

Après, on reconfigure le fichier que l'on vient de copier.

```
root# gedit open-ims.dnszone
```

Et on remplace l'adresse 127.0.0.1 par 192.168.1.50 ou par l'adresse de votre machine, on garde les modifications et on ferme le fichier.

Après on edit le fichier named.conf

```
root# gedit named.conf
```

Et on ajoute la zone open-ims.test

```
zone "open-ims.test" {
type master ;
File "/etc/bind/open-ims.dnszone";
};
```

Une fois ajouté cette zone, on garde les changements et on ferme le fichier.

Il faut vérifier que le fichier /etc/resolv.conf pointe vers le nouveau DNS:

```
root# cd /etc/
root# gedit resolv.conf
```

---

Dans le fichier il doit y avoir:

```
search open-ims.test  
nameserver 192.168.1.50
```

Si le fichier a comme domaine "localdomain" ou une autre au lieu de open-ims.test, il faudra le changer.

Après on garde cette configuration dans ce fichier.

Il faut s'assurer aussi d'avoir cette adresse IP configuré dans la machine (192.168.1.50), donc il faut faire le changement manuellement.

Il faut réinitialiser le serveur DNS avec la commande

```
root# /etc/init.d/bind9 restart
```

Ou alors, réinitialiser la machine pour qu'elle prenne tous les changements. Car de fois ça ne marche pas seulement en réinitialisant le DNS, une fois réinitialisé, vous pouvez faire une test en faisant un ping aux domaine créés :

```
root# ping open-ims.test  
root# ping pscf.open-ims.test  
root# ping icscf.open-ims.test  
root# ping scscf.open-ims.test
```

**ATTENTION:** De fois ça prend un peu de temps en répondre le ping. Si vous utilisez l'installation dans une machine virtuelle (VMWARE ou autre), assurez vous que dans la configuration de la machine virtuelle il y a bien l'adresse du DNS que l'on en a configuré, et pas l'adresse DNS qui est géré par le VMWARE.

Maintenant, on passe à la création de la base de données MySQL (tables du HSS) Ces trois fichiers vont créer la structure du HSS et les utilisateurs qui pourront être utilisés, les utilisateurs qui existent par défaut sont : « bob » et « alice ».

```
user# cd OpenIMSCore
```

Et on exécute les fichiers en extension **sql**

Il ne faut pas mettre aucun mot de passe dans l'exécution des 3 fichiers. On fait la commande et lorsqu'il demande le password, on fait entrée.

```
root# mysql -u root -p -h localhost < ser_ims/cfg/icscf.sql  
root# mysql -u root -p -h localhost < FHoSS/scripts/hss_db.sql  
root# mysql -u root -p -h localhost < FHoSS/scripts/userdata.sql
```

Puis on vérifie que la base de données est bien créée.

```
root# mysql -u root -p -h localhost (ne pas mettre aucun mot de passe)
```

Utiliser des commandes de MySQL pour accéder au HSS de Fokus.

Voici un manuel des commandes de MySQL :

<http://dev.mysql.com/doc/refman/5.0/fr/sql-syntax.html>

### **Pour configurer le cœur IMS :**

D'abord pour le CSCF :

On copie les fichiers : pcsf.cfg, pcsf.sh, icscf.cfg, icscf.sh, scscf.cfg, scscf.xml et scscf.sh qui sont dans « /opt/OpenIMScore/ser\_ims/cfg/ » et on les met dans le répertoire « /opt/OpenIMScore »

```
root# cd /opt/OpenIMScore
root# cp ser_ims/cfg/*.cfg .
root# cp ser_ims/cfg/*.xml .
root# cp ser_ims/cfg/*.sh .
```

Dans les fichiers pcsf.xml, icscf.xml, scscf.xml, et pcsf.cfg, icscf.cfg, scscf.cfg, on doit remplacer l'adresse ip (127.0.0.1) avec l'adresse qui sera utilisé : 192.168.1.50. Assurez vous d'avoir bien fait gardé les modifications faites.

### **Déclenchement des entités CSCF (P, I et S CSCF) et le HSS dans OpenIMScore**

Il faut ouvrir quatre terminaux pour lancer les quatre entités pcsf, icscf, scscf et FHoSS en parallèle: Il faut attendre quelques secondes pour que les fichiers démarrent.

#### **TERMINAL VTY 1**

```
root# cd /opt/OpenIMScore
root# ./pcsf.sh
```

#### **TERMINAL VTY 2**

```
root# cd /opt/OpenIMScore
root# ./icscf.sh
```

#### **TERMINAL VTY 3**

```
root# cd /opt/OpenIMScore
root# ./scscf.sh
```

**Et puis sur le HSS, il existe le fichier fhoss.sh qu'il faut juste exécuter.**

#### **TERMINAL VTY 4**

```
root# cd /opt/OpenIMScore
root# ./fhoss.sh
```

Pour accéder à la console du HSS, on doit ouvrir un navigateur, il faut taper l'adresse : <http://localhost:8080/> dans le browser et puis s'authentifier avec le username= hssAdmin et le mot de passe = hss (ce login et mot de passe sont configurés par défaut lors de l'installation)

A ce stade, le cœur IMS est prêt pour être utilisé.

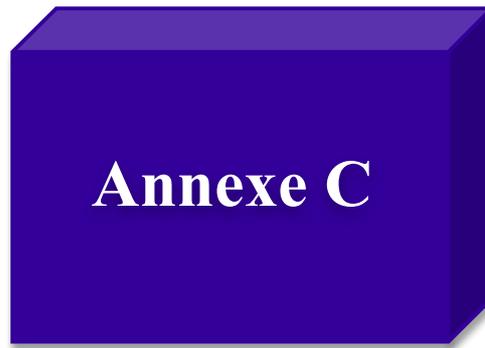
---

Pour le tester, on peut configurer un client SIP dans la même machine pour faire le test d'enregistrer un client. Pour ce faire, on peut télécharger plusieurs clients :

Le X-lite client pour Windows (<http://fr.brothersoft.com/X-Lite-131768.html>) ou une version de linux (<http://linux.softpedia.com/get/Communications/Internet-Phone/X-Lite-5595.shtml>)

Vous pouvez aussi télécharger le client qui est en constant développement et qui fait partie du projet de Fokus, UTC IMS Client (<http://uctimsclient.berlios.de/>). Vous pouvez installer la version 1.0.13

Pour la configuration du client, vous pouvez suivre cette procédure pour le faire : [http://www.openimscore.org/installation\\_guide#ste](http://www.openimscore.org/installation_guide#ste)



**Mise en place d'Oracle  
11g**

---

## Voici la procédure pour la mise en place d'Oracle.

### Vérifier la mémoire (1Go nécessaire)

```
[oracle@oracle MemTotal /proc/meminfo
MemTotal:      1024364 kB
```

### Déterminer la taille du Swap

```
grep SwapTotal /proc/meminfo
[oracle@oracle]$ grep SwapTotal /proc/meminfo
SwapTotal:     1048568 kB
```

### Déterminer la mémoire partagée

```
[oracle@oracle]$ df -k /dev/shm/
Filesystem          1K-blocks      Used Available Use%
Mounted on
tmpfs                512180          0     512180    0%
/dev/shm
```

### Déterminer l'espace disponible du TMP ( il faut environ 150Mo à 200Mo)

```
[oracle@oracle]$ df -m /tmp
Filesystem          1M-blocks      Used Available Use%
Mounted on
/dev/mapper/VGsys-tmpLV
992          36      906    4% /tmp
```

### Vérifier les packages d'installer

```
rpm -qa | grep package_name
```

### Liste des packages nécessaires

```
compat-libstdc++-33.2.3-47.3
elfutils-libelf-0.97-5
elfutils-libelf-devel-0.97-5
glibc-2.3.9.4-2.19
glibc-common-2.3.9.4-2.19
glibc-devel-2.3.9.4-2.19
gcc-3.4.5-2
gcc-c++-3.4.5-2
libaio-devel-0.3.105-2
libaio-0.3.105-2
libgcc-3.4.5
libstdc++-3.4.5-2
libstdc++-devel-3.4.5-2
make-3.80-5
sysstat-5.0.5
unixODBC-2.2.11
unixODBC-devel-2.2.11
```

### Tout d'abord nous devons décompresser le fichier .zip d'installation d'oracle

```
[oracle@oracle]$ unzip linux_11gR1_database.zip
```

```
[oracle@oracle]$cd database
[oracle@oracle/database]$ export DISPLAY=dsiege104855
[oracle@oracle/database]$ ./runInstaller
Starting Oracle Universal Installer...
```

```
Checking Temp space: must be greater than 80 MB. Actual
1392 MB Passed
Checking swap space: must be greater than 150 MB. Actual
1023 MB Passed
Checking monitor: must be configured to display at least 256
colors. Actual 16777216 Passed
Preparing to launch Oracle Universal Installer from
/tmp/OraInstall2007-08-14_02-14-05PM. Please wait ..
```

Nous allons renseigner l'ORACLE\_HOME, l'ORACLE\_BASE (qui servira pour la variable diagnostic\_dest) le type d'installation et le groupe de l'utilisateur qui va installer le moteur.

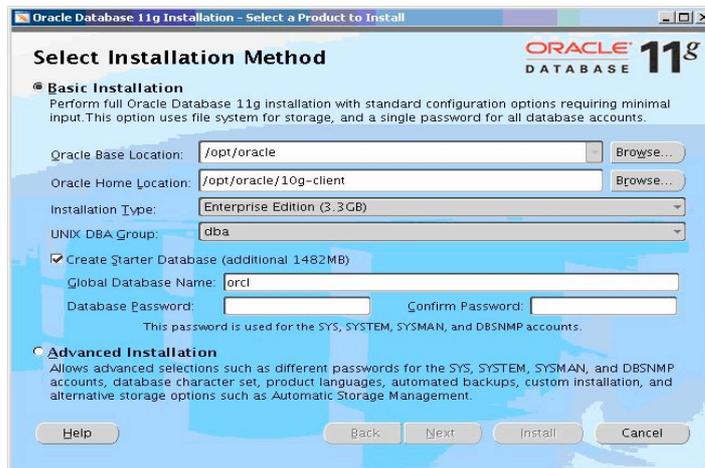


Figure C.1 Définir le nom et mot de passe pour la base de données

Nous allons renseigner l'ORACLE\_HOME, l'ORACLE\_BASE (qui servira pour la variable diagnostic\_dest) le type d'installation et le groupe de l'utilisateur qui va installer le moteur.

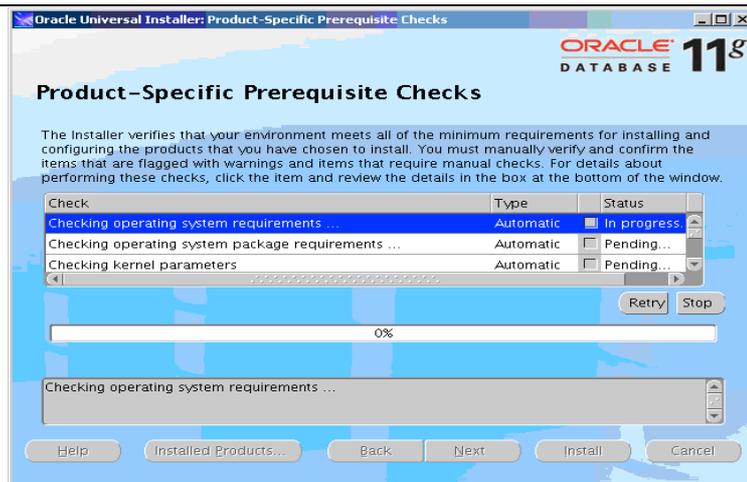


Figure C.2 Vérification de pré-requisites

Une fois que nous avons choisit le nom de la base de données, le répertoire où elle sera installé et le login et mot de passe, il faut regarder que tous les pré-requis sont bien validés (Figure C.2) afin que par la suite Oracle ne produise pas des erreurs lors de son utilisation.

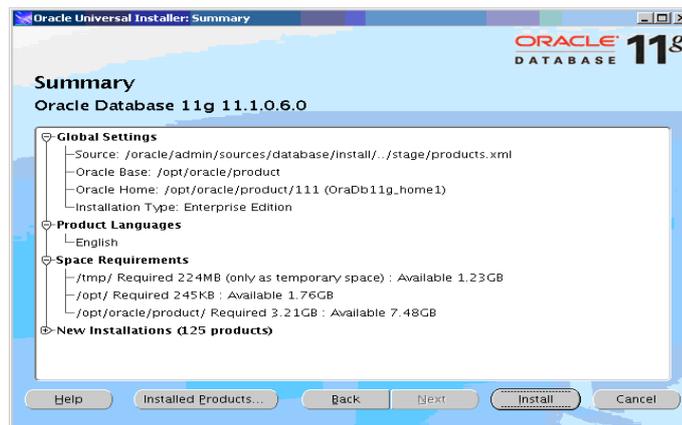


Figure C.3 Résumé avant installation

Nous regardons avant de commencer l'installation, les répertoires où Oracle sera installé (§Figure C.3). Et donc nous attendons l'écran de fin d'installation (Figure C.4).



Figure C.4 Fin d'installation