



HAL
open science

Vers l'autogestion pour une continuité de service intégrée et “ sans couture”

Chunyang Yin

► **To cite this version:**

Chunyang Yin. Vers l'autogestion pour une continuité de service intégrée et “ sans couture”. domain_other. Télécom ParisTech, 2008. English. NNT: . pastel-00004999

HAL Id: pastel-00004999

<https://pastel.hal.science/pastel-00004999>

Submitted on 13 Aug 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Thèse

présentée pour obtenir le grade de docteur
de TELECOM ParisTech

Spécialité : Informatique et Réseaux

Chunyang YIN

Vers l'autogestion pour une continuité de
service intégrée et « sans couture»

Soutenance prévue pour le 28 novembre 2008 devant le jury composé de

Ken Chen

Rapporteurs

Jean Pierre Claudé

Ghislain Du Chéné

Examineurs

Samir Tohmé

Paul Labrogère

Noémie Simoni

Directeur de thèse

Résumé

Nous nous intéressons à la résolution de problèmes de la continuité de service sans couture en assurant une QoS de bout en bout. Dans un premier temps, nous recensons les besoins de l'utilisateur d'aujourd'hui, ce dernier nous conduit à un changement de point de vue : de « System/Application/Network Centric » à « User-Centric ». A partir de cette vision, nous commençons par préciser le contexte de nos travaux. Les verrous à lever sont donc les différentes mobilités qui impactent la continuité de service. Nos contributions sont au nombre de cinq et portent sur des solutions d'ordre organisationnel et fonctionnel.

Basée sur les concepts et résultats de notre groupe de la recherche, notre première contribution a été de proposer un modèle organisationnel supportant la vision «User-Centric » et incluant la personnalisation du service. Du côté fournisseur de service, le transorganisationnel est envisagé.

Afin de mettre en œuvre ce modèle organisationnel, nous avons proposé de nouvelles solutions de gestion de composition de service. Cette deuxième contribution repose sur les CoI (Community of Interest) dont l'intérêt commun est la continuité de service. Pour anticiper tout impact de mobilité, nous créons des communautés virtuelles (CV) regroupant les composants fonctionnellement et QoS équivalents. Ainsi le remplacement des composants n'assurant plus leurs contrats se fait automatiquement et dynamiquement. Nous avons relié chacune des mobilités à un niveau de notre architecture (Utilisateur, Service, Réseau et Equipement), en leur associant une communauté virtuelle (CV) autogérée à travers un protocole Peer-to-Peer.

Nous avons dénommé « *sessionware* » l'ensemble des « services supports » pouvant intervenir durant la session orientée utilisateur. Notre troisième contribution porte sur la fourniture du service responsable de la gestion de la continuité de service (*continuity service delivery*).

Fondé sur l'analyse des mobilités, nous avons identifié la mobilité de session pour laquelle notre quatrième contribution propose un *Binding* cross layer et dynamique des niveaux architecturaux pour rendre transparent la continuité de service.

Pour le respect de la QoS de bout en bout, nous avons recours au concept de VPxN (Virtual Private User/Service/Network/Equipment Network). Le but de cette cinquième contribution est d'auto contrôler les dysfonctionnements de QoS pour assurer le « sans couture ».

Finalement, nous montrons la faisabilité de nos contributions à travers des simulations en JXTA et des expérimentations sur une plate-forme IMS.

2 Vers l'autogestion pour une continuité de service intégrée et « sans couture»

Table des matières

Résumé.....	1
Table des matières.....	3
Table des figures	5
Liste des Tableaux.....	9
Liste des Abreviations.....	11
Chapitre I Introduction générale	15
I.1 Orientation et problématiques.....	16
I.2 Contributions de la thèse.....	19
I.3 Organisation du rapport.....	21
I.4 Références	22
Chapitre II Le contexte.....	23
II.1 Introduction.....	23
II.2 Les mobilités	24
II.2.1 <i>Mobilité de terminal</i>	24
II.2.2 <i>Mobilité de l'utilisateur</i>	31
II.2.3 <i>Mobilité de service</i>	33
II.2.4 <i>Mobilité de session</i>	34
II.3 Périmètre d'étude de la thèse	36
II.4 Références	37
Chapitre III Etat de l'art.....	39
III.1 Composition de Service	39
III.1.1 <i>Service Web</i>	40
III.1.2 <i>Web 2.0 /Web 3.0</i>	41
III.1.3 <i>SOA</i>	42
III.1.4 <i>SaaS (Software as a service)</i>	43
III.1.5 <i>Conclusion</i>	44
III.2 Architectures de service	44
III.2.1 <i>OSA/Parlay</i>	45
III.2.2 <i>TSAS</i>	46
III.2.3 <i>SOA</i>	47
III.2.4 <i>Les réseaux overlay de service</i>	49
III.2.5 <i>P2P Systèmes</i>	49
III.2.6 <i>Overlay de service dans un environnement P2P</i>	51
III.2.7 <i>Conclusion</i>	53
III.3 Les approches autonomiques	53
III.3.1 Les systèmes auto-curatifs	53
III.3.2 Les systèmes auto configurables.....	54
III.3.3 Les systèmes auto-optimisés.....	54
III.3.4 Les systèmes auto-protégés.....	55
III.3.5 Les systèmes autonomiques	55
III.3.6 Conclusion	56
III.4 Conclusion	56
III.5 Références	57
Chapitre IV Du Background aux propositions.....	61
IV.1 Background	61
IV.1.1 <i>Le modèle <N, L, R></i>	62

4 Vers l'autogestion pour une continuité de service intégrée et « sans couture»

IV.1.2	<i>Les niveaux de visibilité</i>	62
IV.1.3	<i>Le modèle informationnel</i>	63
IV.1.4	<i>Modèle QoS</i>	65
IV.1.5	<i>Les profils</i>	66
IV.1.6	<i>Conclusion du background</i>	71
IV.2	Proposition organisationnelle.....	72
IV.3	Références.....	75
Chapitre V	Gestion de la continuité de service.....	77
V.1.	Topologie Fonctionnelle.....	77
V.2.	Gestion de la composition de service.....	81
V.2.1	La composition de service.....	81
V.2.2	Le VPSN.....	86
V.2.3	« Overlay Serviceware ».....	89
V.3.	Gestion des mobilités.....	90
V.3.1	Dysfonctionnements de la QoS du composant.....	90
V.3.2	Des acteurs.....	91
V.3.3	Concept de communauté d'intérêt.....	92
V.3.4	Autogestion de communautés.....	95
V.4.	Continuité de service intégrée : Sessionware.....	104
V.4.1	Concept de session « User-Centric ».....	104
V.4.2	Le Binding de la session.....	105
V.4.3	« Sessionware ».....	106
V.4.4	« Seamless Userware ».....	109
V.5.	Synthèse.....	109
V.6.	Conclusion.....	115
V.7.	Références.....	115
Chapitre VI	L'implémentation.....	117
VI.1	Prototype.....	117
VI.1.1	Plate-forme JXTA.....	117
VI.1.2	Fonctions simulées.....	119
VI.1.3	Scénarii simulés.....	121
VI.1.4	Conclusion.....	127
VI.2	Environnement IMS+.....	128
VI.2.1	<i>Architecture de l'IMS</i>	128
VI.2.2	<i>Les entités du Réseau IMS</i>	130
VI.2.3	<i>L'architecture de service en IMS</i>	131
VI.2.4	<i>Les protocoles et les interfaces</i>	135
VI.2.5	<i>Vers IMS+</i>	136
VI.2.6	<i>Plate-forme de tests OpenCoreIMS</i>	142
VI.2.7	<i>Conclusion</i>	143
VI.3	Références.....	143
Chapitre VII	Conclusion générale.....	145
VII.1	Contributions.....	145
VII.2	Perspectives.....	146
Liste des publications	147

Table des figures

Table des figures

Figure I.1 : « User Centric » dans un contexte NGN.....	16
Figure I.2: System Centric	16
Figure I.3: Application Centric	17
Figure I.4: Network Centric	17
Figure I.5: User Centric.....	17
Figure I.6: définition de service	19
Figure II.1: Les mobilité	24
Figure II.2: HHO et VHO	25
Figure II.3 : L'architecture de l'UMA (spécification de l'UMA).....	28
Figure II.4 : l'architecture de l'UMA [7].....	28
Figure II.5 : L'architecture de la norme de 802.21	29
Figure II.6 : PS Handover :Phase de Préparation [7].....	30
Figure II.7: Mobilité de l'utilisateur.....	31
Figure II.8: mobilité de l'utilisateur supporté par SIP [10].....	32
Figure II.9: mobilité de service	33
Figure II.10 : Définition de sessions dans TINA	35
Figure II.11 : Session service	35
Figure III.1 : exemple de Service Web	40
Figure III.2 : procédure d'un Service Web	40
Figure III.3 : Web 2.0.....	41
Figure III.4 : composition de service exemple 1	42
Figure III.5 : composition de service exemple 2.....	43
Figure III.6 : exemple de la plate-forme de SaaS.....	44
Figure III.7 : La passerelle et l'interface Parlay / OSA.....	45
Figure III.8 : Service Capability Server	46
Figure III.9 : Le modèle métier de TSAS [6].....	47
Figure III.10 : Architecture de SOA	48
Figure III.11 : architecture de la Plate-forme Fiorano SOA™ 2007	48
Figure III.12 : Modèles de communication.....	49
Figure III.13 : architecture de la composition des services.....	52
Figure III.14 : Réseau overlay de service dans un système P2P.....	52
Figure IV.1: Les niveaux de visibilité.....	63
Figure IV.2: Le modèle informationnel NLR (V) du monde réel des télécoms ...	64
Figure IV.3: Le profil de ressource	67
Figure IV.4 : Profil de l'usage de ressource.....	68
Figure IV.5 : Profil de l'usage de service	69
Figure IV.6 : Informations générales du profil de l'utilisateur	69
Figure IV.7 : Profil Localisation et Profil Agenda.....	70
Figure IV.8 : Profil Actif.....	70
Figure IV.9 : Modèle abstrait au service de la mobilité.....	71
Figure IV.10 : L'organisation et Architecture génériques	73
Figure V.1: Vision de NGN	78
Figure V.2 : Topologie fonctionnelle.....	78
Figure V.3: Modélisation de l'opération : (a) Chaîne des opérations (b) Modèle de l'opération	82
Figure V.4: Diagramme d'état d'un ES	83

6 Vers l'autogestion pour une continuité de service intégrée et « sans couture»

Figure V.5: Etat <i>Activé</i> du ES.....	84
Figure V.6: Vue en couches pour les interfaces ES	85
Figure V.7: Modèle architectural.....	86
Figure V.8: Mutualisation d'un ES pour différente Composition de Service	87
Figure V.9: Scénario de la mutualisation d'un ES et Composition de Service	88
Figure V.10: Description fonctionnelle du ES "une seule authentification"	89
Figure V.11: Overlay Serviceware	89
Figure V.12: les cas d'utilisation de la gestion de mobilité.....	92
Figure V.13: Création de communauté	97
Figure V.14: Exemple de la gestion de la communauté des utilisateurs	99
Figure V.15: Gestion de la communauté des utilisateurs	99
Figure V.16: Exemple de la gestion de la communauté des équipements 1	100
Figure V.17: Exemple de la gestion de la communauté des équipements 2	100
Figure V.18: Gestion de la communauté des équipements	101
Figure V.19: Exemple de la gestion de la communauté des services.....	101
Figure V.20: Gestion de la communauté de services	102
Figure V.21 : Autogestion de communauté.....	102
Figure V.22:Sessions Multiples vs. Session Unique	104
Figure V.23:Concept de la session « User Centric ».....	105
Figure V.24:Gestion automatique selon la mobilité d'utilisateur.....	106
Figure V.25: Binding de la session de bout en bout.....	107
Figure V.26: automate de sessionware	108
Figure V.27: Diagramme séquence du Binding	108
Figure V.28 : « NGN Sessionware »	109
Figure V.29 : « Seamless Userware ».....	109
Figure V.30 : Carte fonctionnelle	110
Figure V.31: Scénario final	111
Figure VI.1 : Les protocoles JXTA	118
Figure VI.2 : Annonce de pair A	120
Figure VI.3:Annonce de la Communauté des équipements	120
Figure VI.4: Annonce de la Communauté des services.....	121
Figure VI.5: L'avertissement publié par un EE.....	122
Figure VI.6: Packages importés de JXTA	122
Figure VI.7: Code faux d'un pair	123
Figure VI.8: Création de VEC.....	123
Figure VI.9: Mobilité de l'utilisateur	124
Figure VI.10: Mobilité de terminal	125
Figure VI.11: Avertissement de SE.....	125
Figure VI.12 L'usage de service de découverte de PBS	126
Figure VI.5 : Architecture en couche du Réseau IMS.....	129
Figure VI.6 : Architecture de la plateforme de services.....	131
Figure VI.7 : Interfaces et protocoles de l'architecture de service IMS.....	133
Figure VI.8 : Profile de service des utilisateurs.....	134
Figure VI.9 : Critères de filtrage initiaux (IFCs).....	134
Figure VI.10 : Points de déclenchement des services (SPTs)	135
Figure VI.11 : Procédure générale pour accéder aux services IMS	137
Figure VI.12 : Enregistrement d'un abonné et téléchargement de son profil de service.....	137
Figure VI.13 : Méthode adoptée pour invoquer services dans l'IMS	138

Figure VI.14 : Un NGN Middleware et un HSS évolué	139
Figure VI.15: Diagramme de séquence de SIP	140
Figure VI.16: Construction d'un GS	140
Figure VI.17: Message SIP+ (a) et (b)	141
Figure VI.18: Message SIP+ (c) et (d)	141
Figure VI.19 : Comparaison entre l'architecture d'IMS et celle du cœur Open source	143

8 Vers l'autogestion pour une continuité de service intégrée et « sans couture»

Liste des Tableaux

Liste des Tableaux

Tableau II-1: L'architecture verticale versus architecture horizontale	33
Tableau III-1: Les réseaux P2P	50
Tableau V-1 : QoS du mapping entre le niveau de Service et de Réseau	103
Tableau V-2 : QoS au niveau de Réseau.....	103
Tableau V-3 : Résultat de la Requête de QoS.....	103
Tableau V-4 : La table de décision	114
Tableau VI-1: Les interfaces de l'IMS.....	136
Tableau VI-2: Multiples NGNs.....	141

10 Vers l'autogestion pour une continuité de service intégrée et « sans couture»

Liste des Abreviations

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
BTS	Base Transceiver Station
CH	Communication Host
CN	Core Network
CoA	Care of address
CoI	Community of Interest
CORBA	Common Objet Request Broker
CSCF	Call Session Control Function
CV	Communauté Virtuelle
CVE	Communauté Virtuelle des Equipements
CVR	Communauté Virtuelle de Réseaux
CVS	Communautés Virtuelles de Services
DABS	Device Activity Based Service
DBS	Discovery Based Service
DDBS	Device Discovery Based Service
DLBS	Device Location Based Service
DPBS	Device Presence Based Service
EC	Equipement Communication
EDA	Event Driven Architecture
EDA	Event Driven Architecture
ETSI	European Telecommunications Standards Institute
FA	Foreign Agent
FCAPS	Fault, Configuration, Accounting, Performance, Security
FES	Fiorano Entreprise Server
FS	Fix Server
FTP	File Transfer Protocol
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
LBS	Location Based Service
LLC	Logical Link Control
MBB	Make-before-break
MGW	Media gateways
MIH	Media Independant Handover
MIPv4	Mobile IP v4
MIPv6	Mobile IP v6
MN	Mobile Node
MRF	Multimedia Resource Function
MSC	Mobile Switching Center
mSCTP	Mobile Stream Control Transmission Protocol
NE	Network Element
NGN	Next Generation Network
NGS	Next Generation Service
NGSS	Next Generation Service Session
OASIS	Organization for the Advancement of Structured Information Standards
OB	Overlay Broker
OS	Operating System
OSI	Open Systems Interconnection
P2P	Peer-to-peer
PAN	Personal Access Network
PBS	Presence Based Service
PCSCF	Proxy CSCF
PnP	Plug and Play
PS	Packet Switch
PUI	Public User Identity
QoS	Quality of Service
RNC	Radio Network Controler
ROI	Return On Investigation
SaaS	Software as a service
SABS	Service Activity Based Service
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
SDBS	Service Discovery Based service
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
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 I Introduction générale

Dans le monde télécommunication d'aujourd'hui, la naissance des réseaux et services de nouvelle génération (NGN et NGS) impacte l'utilisateur non seulement dans sa vie professionnelle, mais aussi dans sa vie privée. Suite à l'évolution des technologies, les moyens d'accès se multiplient, les terminaux se diversifient et les services ne cessent de croître. Grâce à cette diversité, l'utilisateur moderne peut avoir plusieurs équipements, plus d'une offre de moyen d'accès et de nombreux fournisseurs de service à sa disposition selon ses déplacements et ses activités. En fait, l'utilisateur d'aujourd'hui désire que l'accessibilité à ce nouveau paysage soit aisée et abordable financièrement. C'est-à-dire, que les utilisateurs exigent de maîtriser leurs communications suivant des critères de qualité de bout en bout, des possibilités de personnalisation et des suivis de coût. Face à ce besoin, les technologies sont de plus en plus complexes, les architectures prolifèrent et les éléments interconnectés sont de plus en plus mobiles et hétérogènes.

Si nous nous plaçons du point de vue des utilisateurs, face à la mobilité et à l'hétérogénéité, ils désirent une continuité de services, en tous lieux, selon leur convenance et sur n'importe quel équipement. C'est dire que cet utilisateur réclame l'accès à n'importe quel service sans aucune barrière technique, afin d'obtenir toujours la meilleure réponse, en adéquation avec son contexte et/ou ses préférences.

Pour satisfaire ces besoins, il nous faut harmoniser mobilité, hétérogénéité de l'environnement, contexte ambiant, ainsi que les préférences de l'utilisateur. Ces besoins dénommés « user-centric » introduisent une nouvelle perception du paysage des télécommunications. Le périmètre auquel nous nous intéressons est justement celui qui se trouve à l'intersection des problématiques (§I.1) de mobilité, d'hétérogénéité et des besoins centrés sur l'utilisateur. Nos contributions (§I.2) ont pour objectif de lever les nouveaux défis de cette approche « User Centric ». L'organisation du rapport (§I.3) est présentée à la fin de ce chapitre pour donner une image globale des résultats de cette thèse.

I.1 Orientation et problématiques

L'orientation qui nous intéresse dans cette thèse est celle de « User-Centric » (Figure I.1) dans un contexte NGN. C'est-à-dire que cet utilisateur est avant tout nomade. Il souhaite d'abord avoir la possibilité d'une connexion à plusieurs réseaux hétérogènes sans coupure pour accéder à ses services. La connectivité ne s'arrête pas à l'établissement de la liaison, la connectivité ne signifie plus juste le maintien du lien, mais elle doit permettre à l'utilisateur d'être facilement relié à tout moment pendant ses déplacements, à tout réseau de sa préférence dont il possède les droits d'accès et à partir de n'importe quel terminal.

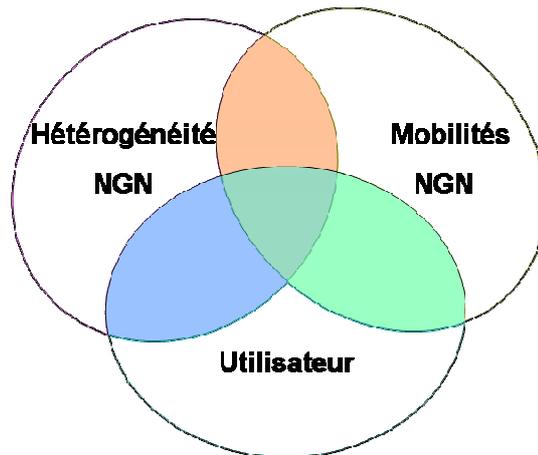


Figure I.1 : « User Centric » dans un contexte NGN

Le principal impact de cette approche est que le système complet est « au service » de l'utilisateur contrairement aux autres approches où l'utilisateur doit se plier aux différentes contraintes de traitement (System Centric) ou de connexions (Network Centric).

« System Centric » (Figure I.2) est basé sur OS (Operating System) supporté par le hardware où il est installé. Les applications tournent parallèlement dans cet OS grâce au *Compilateur*. C'est le *Compilateur* qui fait la traduction et l'optimisation statique pour avoir une meilleure exécution de service. Si l'utilisateur a besoin d'une application qui n'existe pas dans le système, les algorithmes associés sont obligés d'être traduits par le *Compilateur*.

Il est à noter que pour éviter les re-traductions pour une application à cause des changements des OS, VM (Virtual Machine) (Figure I.2- droite) est proposé comme un middleware pour cacher l'hétérogénéité des OS. Grâce à cette couche intermédiaire, les applications développées n'ont besoin que d'une seule *Compilation* pour être utilisées sur n'importe quel OS.

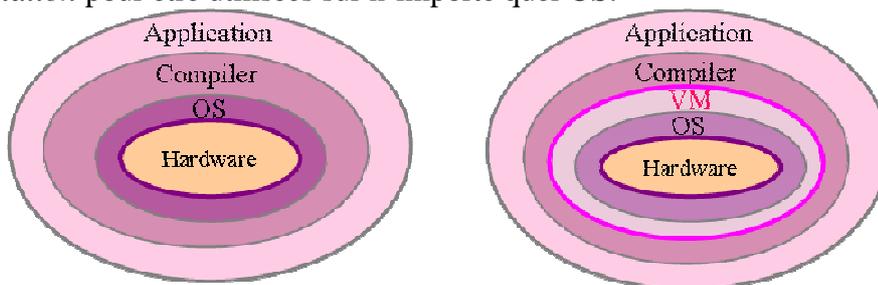


Figure I.2: System Centric

Par contre, « Application Centric » (Figure I.3) se concentre sur l'application et la considère comme le point de départ. Dans une « Application Centric » système, le programme est chargé en premier, le *Compilateur* s'exécute en temps réel pour une adaptation de l'OS. Pour l'aspect architectural, la capacité de calcul parallèle est demandée.

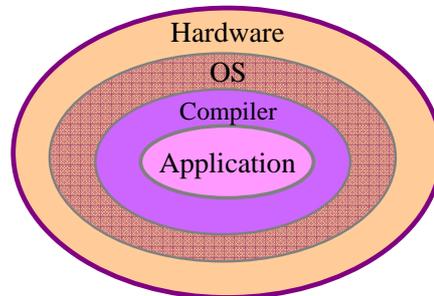


Figure I.3: Application Centric

« Network Centric » (Figure I.4) implique que les infrastructures du réseau soient au cœur de l'architecture, elles conditionnent toutes demandes de services. L'hétérogénéité du réseau impose des solutions différentes pour un service demandé à travers un support de connexion. Les communications peuvent être parallèles mais des technologies de handover sont exigées pour basculer d'un réseau à l'autre.

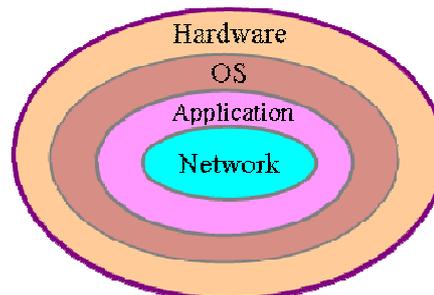


Figure I.4: Network Centric

Avec une approche « User Centric » (Figure I.5) c'est la personnalisation des services (a) qui doit être mise en oeuvre, à travers une connexion temporelle avec le système, c'est-à-dire, à travers une session unique (b) que l'utilisateur désire établir dynamiquement (c) à travers les accès offerts durant ses déplacements, selon la QoS (Quality of Service) (d) désirée. Les quatre points (a, b, c, d) que nous venons d'identifier sont les verrous que nous devons lever. Ils constituent la problématique que nous voulons traiter dans cette thèse.

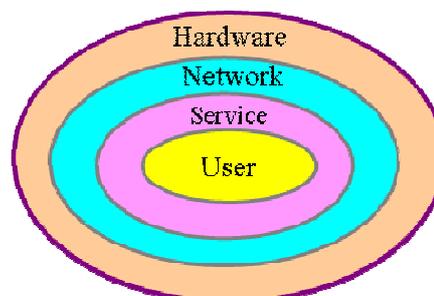


Figure I.5: User Centric

(a) Personnalisation du service

Comme nous venons de les décrire les services du futur seront constitués d'un enchaînement de services de base avec une logique de service personnalisée. Mais cette logique de service ne peut se faire que si nous avons des services de base composables. Il est donc important d'avoir une bonne approche du service. Un service n'est ni une application, ni une transaction et encore moins un système. L'ISO 20000 [1] le définit comme « une prestation composable qui doit être source de valeur pour le consommateur et le fournisseur ».

Or pour faire cette composition dynamique, pour assurer toutes sortes de personnalisation, pour suivre la mobilité de l'utilisateur, nous ne pouvons plus nous satisfaire d'une architecture de service « client / serveur » avec des services à options.

*Il nous faut penser les services autrement. Avoir une architecture permettant à l'utilisateur de faire sa propre **composition de service**, avec des changements dynamiques en fonction des propositions qui lui sont faites.*

(b) Une seule session pour des différents services

Pendant une session « User Centric », l'utilisateur est amené à composer dynamiquement le service (global) dont il a besoin durant un espace temps. Par exemple, ajouter un service de visiophonie à son appel voix ou transférer un programme de TV de son poste de télévision à son PDA lorsqu'il se déplace. Mais, on peut avoir des processus, des logiques de services plus élaborées. Par exemple, pour une session qui se déroulerait, dans le créneau horaire 8h45-10h45, l'utilisateur désire avoir ses emails en priorité quand il quitte son domicile, puis étant dans son véhicule, il souhaite recevoir ses SMS en vocal, puis une fois rendu à son bureau il voudrait avoir simultanément sur son PC email, SMS et ses fichiers.

*Cela implique une **continuité du service** global durant la session. Cette problématique est capitale pour les nouvelles générations de service et constitue un des points focaux de cette thèse.*

(c) Accessibilité au service par n'importe quel moyen

Avec le développement de la technologie des terminaux, la nouvelle génération offre plusieurs modes. Un même terminal peut accéder à différents réseaux d'accès hétérogènes. Par exemple, pour un utilisateur, le mode d'accès de WLAN et GSM est simultanément disponible, s'il est intéressé par la largeur de bande de WLAN, il préférera choisir dynamiquement ce dernier. Par ailleurs, 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. Pour le système, comment choisir son mode d'accès?

*Il nous faudra sélectionner le réseau et le terminal le plus adéquat lors de la demande de service et savoir en changer durant le déplacement de l'utilisateur en fonction de ses préférences. Si parfois, l'adaptation du contenu sera nécessaire, **la dynamique de la session** sera toujours de mise.*

(d) Une QoS de bout en bout

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. Les choix se font en fonction des préférences de l'utilisateur et du SLA (Service Level Agreement) que ce dernier demande.

Il nous faut donc non seulement satisfaire la QoS au niveau de chaque composant, mais assurer une QoS de bout en bout.

En conclusion, l'utilisateur d'aujourd'hui se déplace, change d'environnement (d'accès réseau), change de terminal, désire une continuité de service sans couture et une QoS de service de bout en bout. Cette thèse contribue à la fourniture de concepts et « d'outils supports » afin de répondre à ces besoins de : « anywhere, anytime, anyhow, every services, everyone », avec un souci de transparence et d'automatisme.

1.2 Contributions de la thèse

Le champ d'application de cette thèse se situe dans un premier temps au niveau des services. Nous considérons que ce niveau regroupe tous les services offerts *par et sur* un réseau de télécommunication. Pour en faire une gestion transparente, nous proposons une architecture de services (Figure I.6) selon les concepts et une modélisation « overlay réseau ». La couronne intermédiaire représente la couche service, avec des services applicatifs (SCA) pour supporter les applications, des services de transport réseau (SCN) pour traduire les besoins d'acheminement. Grâce aux éléments supports (SCA et SCN), un service global (SCU) « User-Centric » peut répondre à la demande de l'utilisateur.

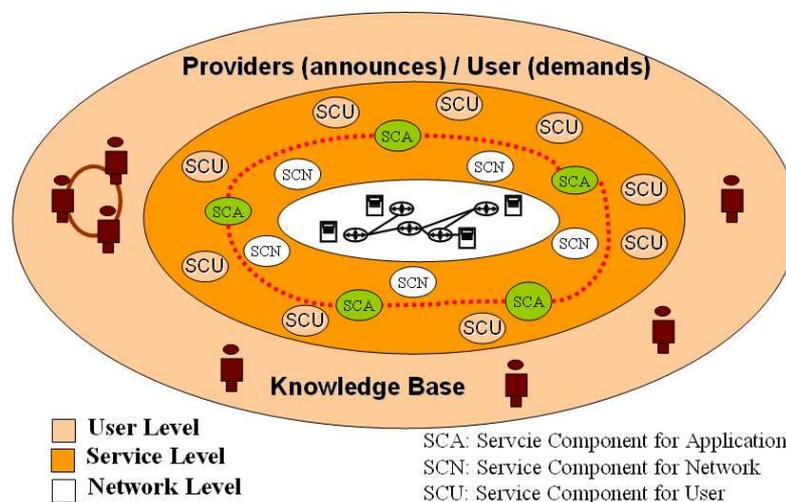


Figure I.6: définition de service

Les contributions de cette thèse constituent un ensemble de concepts et « d'outils supports » (dénommés par la suite « services supports »), pour répondre aux différentes problématiques que nous venons d'identifier à des fins d'intégration et d'automatisme.

(a) Personnalisation du service

Basé sur les résultats de notre groupe de la recherche, les applications de nouvelle génération se conçoivent à travers une composition de services et se représentent par un VPSN (*Virtual Private Service Network*) [2]. Cette solution permet de concevoir des applications (VPSN) dynamiques dans un contexte ubiquitaire, mobile et ambiant.

La *première contribution* de cette thèse est un modèle organisationnel (§4.2) supportant la personnalisation du service dans cette nouvelle architecture, qui permet d'envisager le transorganisationnel.

La *deuxième contribution* porte sur la gestion de la composition de service basée sur les approches SOA (Service Oriented Architecture) et EDA (Event Driven Architecture) (§5.2).

Nous proposons d'utiliser le concept de communautés d'intérêt pour gérer automatiquement le remplacement dynamique d'un composant en fonction du déplacement (mobilité) (§5.3). Dans cette *troisième contribution*, nous créons des communautés virtuelles de services (VSC) fonctionnellement équivalents et QoS équivalentes. Nous faisons de même pour les terminaux, pour les réseaux et pour les utilisateurs. Le « service support » de gestion proposé consiste alors à contrôler à travers un protocole Peer-to-Peer les éléments des communautés et à réorganiser les communautés en fonction des éléments défaillants (non respect de QoS, panne, etc.). Grâce à ce concept, chaque communauté virtuelle pourra se maintenir et s'autogérer pour garantir le respect du contrat de la communauté paire. Parallèlement, ce service remplacera automatiquement au niveau du VPSN le composant défaillant.

(b) Une seule session pour des différents services

La *quatrième contribution* a pour objectif d'assurer la continuité de service durant la session orientée utilisateur (§5.4). Elle est d'ordre fonctionnel et comporte un ensemble de « services supports » dénommé « *sessionware* ».

(c) Accessibilité au service par n'importe quel moyen

La *cinquième contribution* porte sur la mobilité de la session et le Binding des cinq niveaux architecturaux.

En fait, l'architecture se décline en cinq niveaux, celui de l'utilisateur, celui des services, puis ceux du réseau coeur et des réseaux d'accès et pour finir celui des équipements dans lequel nous trouvons entre autres les terminaux de l'utilisateur. L'approche « user centric » impose que nous tenions compte des préférences de l'utilisateur à tous les niveaux de cette architecture.

Lors des déplacements de l'utilisateur, chacun de ces niveaux peut être impactés. En effet, le changement de terminal nécessitera des modifications du niveau équipement, celui du point d'accès entraînera des handovers dans le niveau réseau d'accès et éventuellement le réseau coeur. La dégradation de la QoS induira des remplacements de composant de service dans le niveau service. Quant au niveau utilisateur, il y aura des mises à jour continues de son Système d'Information (SI). C'est pourquoi nous avons besoin d'un Binding dynamique entre ces cinq niveaux. C'est ce binding cross layer que nous proposons et qui recouvre la mobilité de session.

(d) Une QoS de bout en bout

L'ensemble des contributions que nous venons de consigner assurera la gestion de la continuité de service en se basant sur la QoS.

Au niveau du service, c'est le VPSN qui contrôle le respect de la QoS de bout en bout en comparant l'agrégation de la QoS de la chaîne en temps réel avec le SLA. Il se situe au niveau du réseau overlay de service basé SOA.

En fait, le VPSN est composé d'éléments de service, dont la QoS est géré dynamiquement et automatiquement par les VSCs. Une base de connaissance est mise à jour pour garder toujours une image en temps réel de chaque élément dans le système. En conséquence, toute discontinuité est signalée. Nous procédons au changement par anticipation afin de maintenir la continuité de service.

En conclusion, cette thèse propose une fourniture du service global à travers une session « User Centric » de bout en bout dont l'objectif est la continuité de service intégrée. Grâce à la gestion des communautés virtuelles, l'autogestion du VPSN se fait au fur et à mesure du déplacement de l'utilisateur. La mobilité de session est assurée par un binding dynamique et cross layer des cinq niveaux de l'architecture proposée.

1.3 Organisation du rapport

Pour répondre à notre problématique, nous avons proposé nos travaux, nos réflexions et nos résultats à travers sept chapitres.

Le premier chapitre est cette introduction qui a pour objectif de présenter la motivation et les problématiques de cette thèse. Après avoir recensé les besoins de l'utilisateur d'aujourd'hui, nous avons montré le changement de point de vue. A partir de cette vision « User-Centric », notre principale problématique est : comment assurer une continuité de service sans couture à travers une QoS de bout en bout. Suivants les besoins de « User-Centric », les cinq contributions de cette thèse sont présentées et l'organisation y est consignée.

Le chapitre deux précise le contexte de nos travaux. Les quatre types de mobilité sont étudiés en détail, afin de spécifier les fonctions manquantes pour assurer la continuité de service. Nous identifions alors la gestion de la mobilité de session comme support de cette continuité de service. Le périmètre d'étude conclut ce chapitre.

Le chapitre trois synthétise notre travail de recherche d'étude bibliographique. Avant d'envisager les nouvelles solutions, nous avons examiné les solutions existantes dans le périmètre que nous avons identifié, à savoir, les solutions de la composition de service pour satisfaire les besoins de notre « User Centric ». Puis nous avons examiné les aspects architecturaux pour lesquels nous avons étudié les architectures de service. Pour terminer, nous avons appréhendé les approches autonomiques pour couvrir les aspects de gestion de la continuité de service.

La thèse s'inscrivant dans un contexte plus global et reposant sur un certain nombre de concepts, il a été nécessaire de consigner le background de nos travaux. Le chapitre quatre présente le modèle $\langle N, L, R \rangle$, les niveaux de visibilité, le modèle Informationnel et le modèle de QoS, puis les profils, outils de base de tous

nos travaux. Après en avoir rappelé les principaux résultats, nous faisons la première contribution de cette thèse en proposant un modèle organisationnel intégrant les mobilités du notre contexte NGN/NGS.

Le chapitre cinq consigne nos propositions. Après avoir adressé à la topologie fonctionnelle, nous trouvons trois propositions : La première est relative à la composition de service qui repose sur une couche overlay de service à part entière, la seconde à la gestion des mobilités et la dernière à la gestion de la session de bout en bout. Plus globalement, un ensemble de « services supports » est spécifié. Ceux qui gèrent la session sont regroupés dans le « NGN Sessionware » pour gérer la mobilité de session à travers un Binding dynamique et cross layer pour avoir la continuité de service.

Le chapitre six est consacré à un démonstrateur basé sur la plate-forme d'IMS pour examiner la faisabilité de nos propositions. Des scénarii sont déroulés pour illustrer la mise en œuvre des solutions proposées.

Le dernier chapitre conclue notre travail de thèse et présente nos perspectives de recherche.

1.4 Références

- [1] ISO 20000 IT Service Management Standards 2005
- [2] Zakaria Benahmed Daho, Vers une Nouvelle Génération de Service pour une Autogestion de bout en bout, 2006

Chapitre II Le contexte

Dans le paysage des télécommunications d'aujourd'hui, d'après notre vision « User Centric », les utilisateurs ne souhaitent plus être limités par leur location ni par les moyens accès et ils désirent que le service dont ils ont besoin soit maintenu. L'évolution technologique nous permet d'être de plus en plus proches de cette indépendance, avec, une plus grande capacité de déplacement. Pour identifier les besoins de notre « User Centric » nous allons introduire dans ce contexte les différentes mobilités (§ II.1). Puis, une analyse profonde de toutes ces mobilités (§ II.2), nous permettra de délimiter le périmètre d'étude de cette thèse (§ II.3).

II.1 Introduction

Pouvoir se déplacer au gré de ses besoins ou de ses envies est aujourd'hui tellement simple, que la location physique n'est plus la contrainte majeure pour les utilisateurs modernes. Grâce à une grande couverture et à la diversité des technologies sur les offres d'accès, les utilisateurs peuvent avoir leurs services n'importe où. Mais dans le contexte de NGN où l'hétérogénéité est omniprésente le maintien de la communication et du service d'un utilisateur qui se déplace est de plus en plus complexe. Pour comprendre cette complexité, analysons les différents types de mobilité.

Les communications personnelles induisent des mobilités essentiellement d'ordre spatial comme la « mobilité du terminal », la « mobilité de l'utilisateur » et la « mobilité du réseau ».

La « mobilité de terminal » implique la continuité de la connexion. Les technologies du genre « tout IP » permettent de considérer une architecture de réseau unifié, laquelle peut traiter divers réseaux d'accès indépendants. Lorsque l'utilisateur passe d'un terminal à un autre, cela relève de la « mobilité de l'utilisateur », nous devons alors résoudre les adaptations pour préserver la personnalisation. La « mobilité du réseau » concerne le déplacement de l'infrastructure du support de transport. C'est le cas des réseaux locaux (trains ou

avons) en mouvement. En fait, c'est le sous réseau qui bougent avec tous ces nœuds, la problématique est la même que pour la mobilité de terminal. Par contre, lorsque c'est un nœud du réseau cœur qui se déplace, nous avons une problématique d'interconnexion et d'adaptation des débits à considérer.

Ces trois types de mobilité sont relatifs à des aspects de connectivité. Dans le monde réel, il faut également considérer les aspects de service. Avec l'évolution de fourniture de service, un même service demandé par l'utilisateur peut être offert par plusieurs fournisseurs et supporté par différentes plates-formes de service. Quand l'utilisateur se déplace, comment fait-il son choix pour avoir la meilleure adéquation à ses besoins ?

Nous avons dénommé ce quatrième type de mobilité : la « mobilité de service » car au-delà de la préférence de l'utilisateur, elle sera utilisée par le système pour respecter la QoS de bout en bout.

Cette « mobilité de service » induit en fait une autre mobilité d'ordre temporel que nous avons dénommé « mobilité de session » (Figure II.1 - A). En effet, pour assurer une continuité de service avec la chaîne des mobilités spatiales, le système doit gérer dynamiquement la session de l'utilisateur et son unicité de bout en bout (« seamless, sans couture ») en temps réel. En fait, chaque mobilité relève d'un niveau architectural, et ne permet pas à lui seul d'assurer la QoS de bout en bout. C'est la session, gestionnaire dynamique en temps réel qui maintiendra cette QoS de bout en bout.

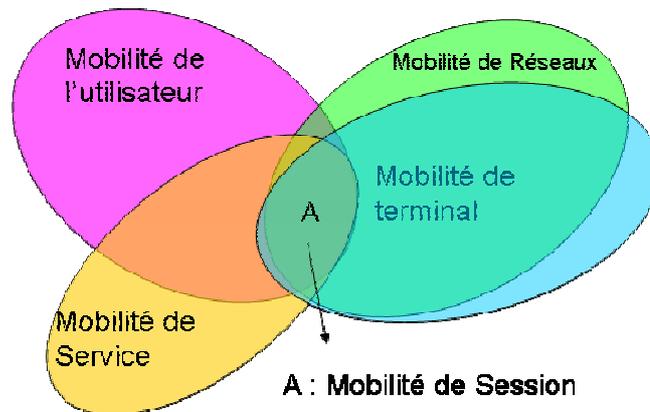


Figure II.1: Les mobilités

II.2 Les mobilités

La « mobilité de terminal » (§II.2.1) est celle la plus étudiée aujourd'hui, elle masque souvent la « mobilité de l'utilisateur » (§II.2.2) alors que cette dernière engendre des défis bien distincts. La « mobilité de réseau cœur » est surtout adressée dans un contexte militaire et ne sera pas dans le scope de notre thèse. Par contre, la « mobilité de service » (§II.2.3) et la « mobilité de session » (§II.2.4) en sont les points focaux.

II.2.1 Mobilité de terminal

Les solutions existantes dans le domaine de la mobilité reposent sur les techniques de handover qui s'appliquent plutôt à la « mobilité du terminal ».

Elle se rapporte à un terminal qui change de location, c'est-à-dire qui se déplace soit à travers différents points d'accès d'un réseau, soit à travers différents réseaux d'accès, tout en maintenant l'accès à un même ensemble de services. Ce type de mobilité doit permettre le déplacement de terminal sans coupure. Deux types de solutions existent le Roaming et le Handover.

Le *Roaming* est l'itinérance qui fait plutôt référence au changement d'opérateur ou de domaine administratif. Ce type d'itinérance offre donc une possibilité d'obtenir des services qui sont dans des domaines administratifs différents gérés par des contrats entre opérateurs. Il est d'ordre commercial et réglementaire.

Le *Handover* est d'ordre technique. Ce processus permet, à un terminal mobile, d'effectuer le passage entre deux points d'attachement de façon transparente. Le changement de point d'attachement entraîne parfois une déconnexion momentanée du terminal mobile et des perturbations des communications en cours. Néanmoins l'objectif est d'assurer des communications sans coupure.

Nous distinguons le HHO (Horizontal HandOver) et le VHO (Vertical HandOver) (Figure II.2) [1]. Le HHO permet au terminal de passer d'un domaine d'accès à un autre en utilisant la même technologie, la procédure se déroule dans un même niveau d'architecture. Quant au VHO, il permet le passage d'un terminal, d'un type de réseau à un autre (ex : de WiFi, à la 3G), ce handover se traite souvent par plusieurs niveaux et dans un environnement hétérogène.

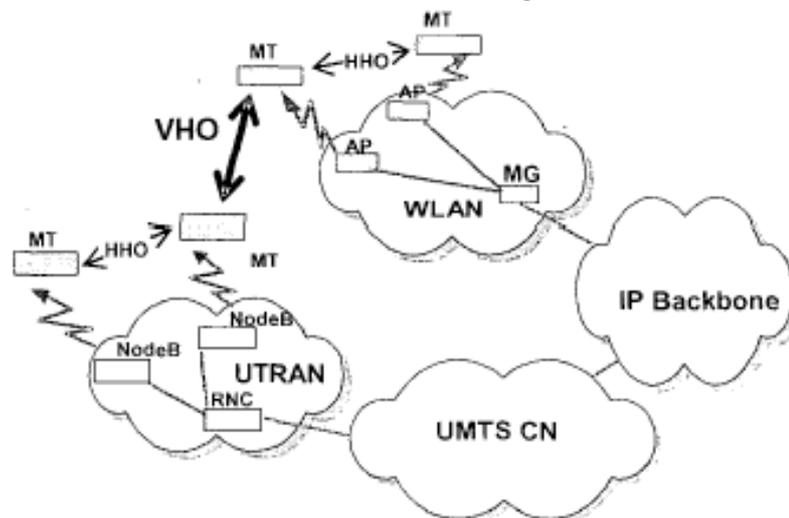


Figure II.2: HHO et VHO

Handover Horizontal (HHO)

L'utilisation de la procédure de handover horizontal la plus usitée se trouve en téléphonie mobile du niveau L2. D'une manière générale, le handover va être effectué en analysant trame par trame le signal reçu des deux cellules impliquées et la meilleure trame sera retenue. Ainsi, progressivement, le nombre de trames traitées par la cellule d'accueil devient prépondérant devant le nombre de trames traitées par la cellule cédante. Plusieurs techniques de handover ont été proposées. Par exemple, dans UMTS (Universal Mobile Telecommunications System) [2], nous trouvons du soft handover, softer handover et hard handover.

- Soft Handover

Le handover s'effectue « en douceur ». On dit qu'il s'agit d'un soft handover. Ce type de handover change de station de base. Le mobile est dans la zone de couverture qui est commune à deux stations de base. Les communications utilisent

deux canaux différents, un pour chacune des deux stations. Contrairement au mécanisme de handover traditionnel, tel que rencontré dans un réseau analogique ou GSM, il n'y a pas d'interruption de la communication, même de très courte durée.

- Softer Handover

Dans un système W-CDMA, on distingue le cas où le mobile reste dans la zone couverte par une station de base en changeant juste de secteur, on l'appelle alors softer handover. Le mobile est en communication avec une seule station de base, il utilise simultanément deux canaux radio. Dans le sens descendant, deux codes d'étalement sont activés pour que le mobile distingue les signaux issus des deux secteurs. Dans le sens montant, les signaux émis par le mobile sont reçus par les deux secteurs de la station de base et dirigés vers le même récepteur. Ils sont donc combinés au niveau de la station de base. Du côté du mobile, il n'y a pas de différence avec un soft handover. Dans le sens montant, par contre, les données sont combinées au niveau du contrôleur de réseau radio (RNC) et non plus au niveau de la station de base. Cela permet de sélectionner la meilleure trame parmi celles qui sont reçues, après chaque période d'entrelacement, toutes les 10 à 80 ms.

- Hard Handover

En dehors des handovers en douceur qui viennent d'être décrits et qui sont les plus courants, on rencontre dans un système W-CDMA deux autres types de transfert intercellulaire, qu'on appelle hard handover par opposition aux mécanismes précédents :

- Handover inter fréquence, lorsque le mobile passe dans une cellule où les fréquences sont différentes de celle qu'il quitte ;
- Handover inter système, quand le mobile change de système, par exemple pour quitter une plaque UMTS et entrer dans une plaque GSM, ou plus simplement pour passer du mode FDD au mode TDD.

Nous avons également un handover horizontal de niveau L3 en mode « paquet », c'est-à-dire, au niveau de l'IP à travers la technologie « Mobile IP ». Nous trouvons pour IPV4 le RFC3344 [3], pour IPV6 le RFC 3775 [4] qui réalise le handover en cachant le changement de l'adresse IP et la technologie mSCTP [5] (mobile Stream Control Transmission Protocol) qui met à jour l'adresse IP dynamiquement.

Mobile IP garde deux types d'adresse IP, une adresse permanente (Home address) qui peut être utilisée au dessus du niveau transport; l'autre adresse IP modifiable (Care of address) qui peut être utilisée au dessous du niveau transport. Dans la version IPV4, ce protocole propose trois composants :

- MN (Mobile Node) qui représente le terminal qui a changé de points d'attachement tout en maintenant la communication grâce au « Home Address »,
- HA (Home Agent) qui est un routeur dans le réseau mère et
- FA (Foreign Agent) qui se trouve dans le réseau visité par le MN.

Quand le MN bascule, le MN enregistre le « Care of address » auprès le HA à travers le FA. Quand un hôte de communication (CH) envoie les paquets de

données par le routage IP au MN dans le réseau mère, HA intercepte ces paquets, les encapsule en mettant la CoA comme adresse destinataire. Finalement, le HA les envoie à MN. Un tunnel virtuel entre CH et MN est alors supporté. Mais ce routage triangulaire peut provoquer de la congestion au niveau transport.

L'évolution de IPv6 est un routage optimisé. Grâce à l'enregistrement de l'association de deux adresses d'un MN (Binding) dans le nœud correspondant (CN), la communication peut être maintenue directement entre le MN et le CN.

mSCTP est défini comme SCTP avec la capacité de la reconfiguration dynamique pour l'adresse. Il est principalement prévu pour une architecture de type client/serveur, avec le client mobile qui initie l'association avec le serveur fixe (FS). Le mobile obtient une nouvelle adresse IP auprès cette association et remplace l'ancienne.

Handover Vertical (VHO)

Dans l'environnement hétérogène, nous avons étudié la technologie de l'UMA (Unlicensed Access mobile) [6] et la norme de IEEE 802.21 MIH (Media Independent Handover) [8] pour savoir les façons de rendre possible le handover Vertical dans un environnement hétérogène (handover hétérogène).

UMA (Unlicensed Access mobile) :

Dans la suggestion d'UMA [6] ce genre de handover hétérogène est basé sur l'idée qu'à l'avenir les utilisateurs préféreront avoir un seul terminal puissant (faisant tout) que plusieurs équipements indépendants. La technologie UMA rend la possibilité de basculer automatiquement du réseau WiFi aux réseaux mobiles sans intervention de l'utilisateur. Elle a pour objectif d'offrir un accès aux réseaux GSM (Global System for Mobile communications) et GPRS (General Packet Radio Service) par l'intermédiaire de réseaux Bluetooth ou WiFi. La connexion de l'UMA se fait soit de manière classique via le réseau mobile (lorsque l'abonné est à l'extérieur), soit par le biais d'un réseau local radio (lorsque ce même abonné se trouve chez lui, dans une entreprise ou dans la zone de couverture d'un hot spot Wi-Fi). Elle a été développée par un consortium d'entreprises nommé UMAC comptant entre autre Alcatel, Cingular, Ericsson, Motorola, Nokia, Nortel Networks, Siemens, T-Mobile et Kineto Wireless. L'objectif ultime de l'UMA est de faire converger les protocoles de communications des téléphones mobiles, fixes et informatiques. Le standard UMA permet le handover hétérogène entre les deux réseaux en autorisant le transport des communications et de la signalisation sur le réseau IP. Avec l'architecture proposée par l'UMA (Figure II.3), quand le téléphone mobile détecte un UMAN, il établit une connexion IP sécurisée à travers une passerelle vers un « UMA Controller » (ou GANC, GAN Controller). Ce dernier transforme le signal provenant du mobile pour le faire apparaître comme provenant d'un autre relais GSM. Par conséquent quand un utilisateur passe d'un réseau GSM vers un réseau Wifi, le cœur du réseau téléphonique considère simplement que le mobile a changé de relais GSM, comme pour un handover GSM classique. Il n'y a donc pas de coupure de communication alors que l'on passe d'un média GSM à un média Wi-Fi ou bluetooth.

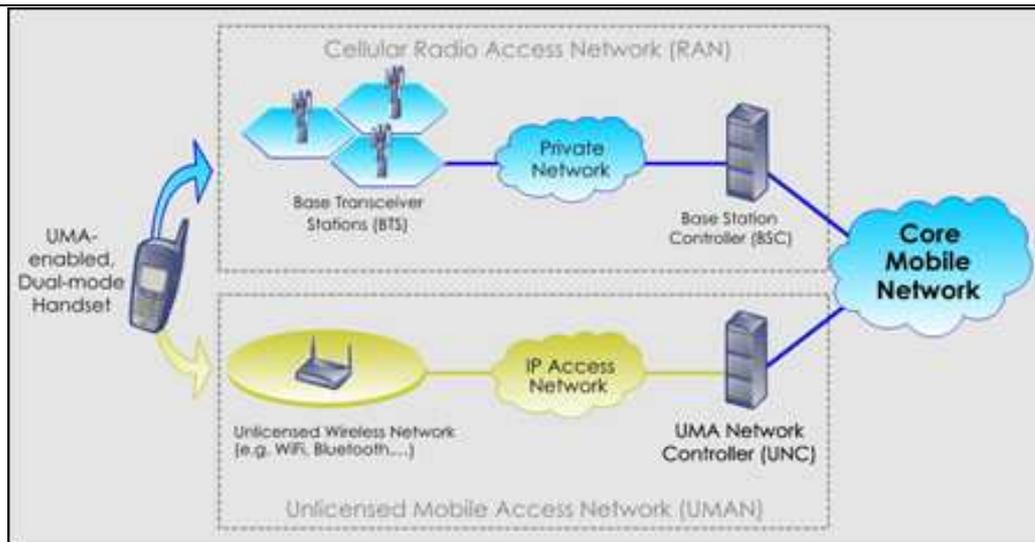


Figure II.3 : L'architecture de l'UMA (spécification de l'UMA)

Pour supporter la mobilité, dans l'architecture de l'UMA proposée par 3GPP 43.318 [7] (Figure II.4), la passerelle de sécurité (SGW : Security Gateway) authentifie l'utilisateur avant sa connexion de WLAN au réseau cœur. SGW peut consulter le HLR dans le réseau Home pour les informations de l'utilisateur. Après l'authentification faite par le Serveur de AAA, les paquets peuvent être envoyés.

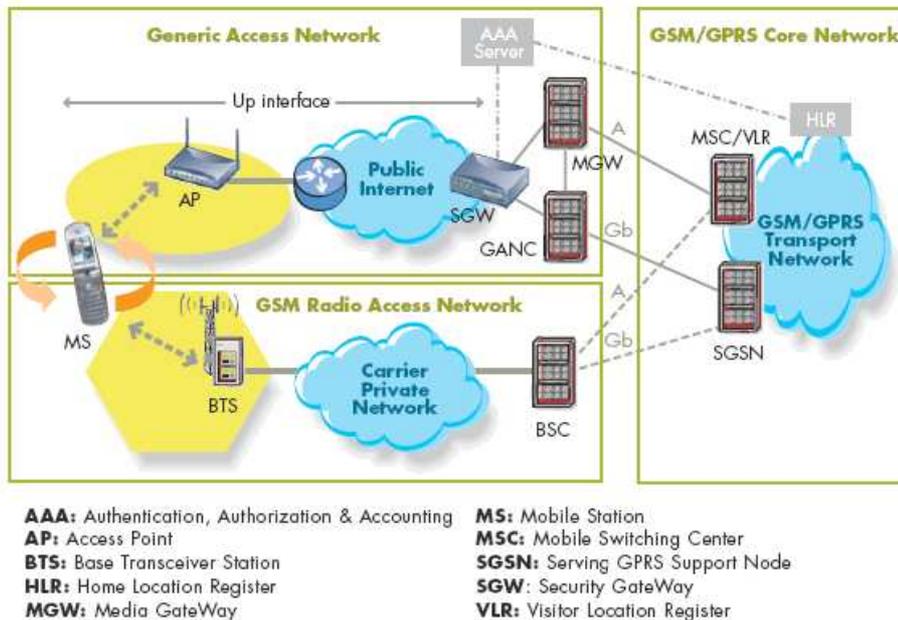


Figure II.4 : l'architecture de l'UMA [7]

Mais la non prise en compte de l'aspect QoS quand un mobile de l'UMA bascule sous les réseaux sans fil et la nécessité de contrats entre opérateurs pour le roaming limitent l'UMA à un environnement domestique ou à un petit bureau.

MIH (Media Independant Handover)

IEEE 802.21 [8] introduit le concept de MIH (pour réaliser le handover hétérogène entre technologies sans fils hétérogènes (Figure II.5). Cette norme est un standard en cours de développement. Son but principal est de gérer une connectivité sans discontinuité des différents réseaux sans fil. Cette norme gère le handover entre les réseaux cellulaires, la téléphonie mobile, le WiFi et le Bluetooth (concept de handover diagonal).

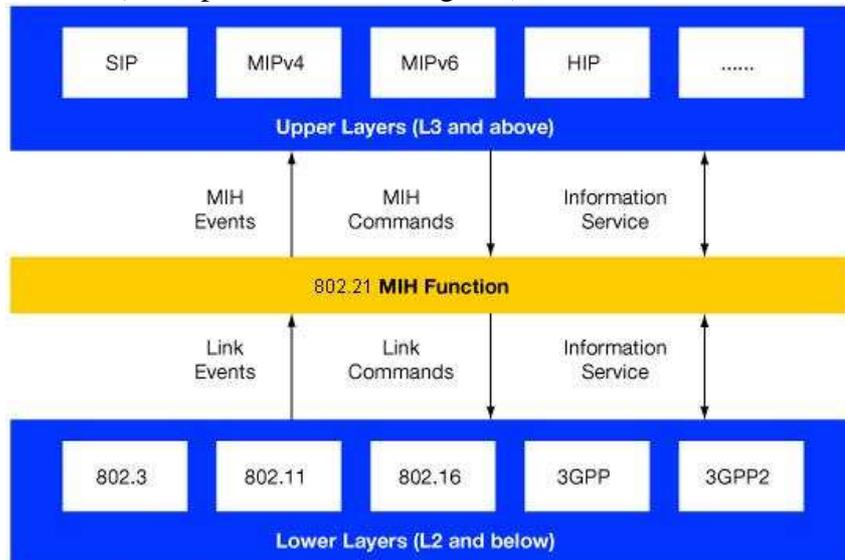


Figure II.5 : L'architecture de la norme de 802.21

Son objectif est de combler les lacunes des réseaux 802 qui permettent la détection et la connexion aux points d'accès de leurs réseaux mais pas encore aux points d'accès des autres réseaux. Il faut donc gérer la continuité de la connexion et les paramètres de sécurité liés aux différentes normes. De plus, il faut lier les différents réseaux sans fil entre eux.

Cependant, pour la continuité de service, devant l'hétérogénéité des ressources, nous avons recours au handover vertical pour assurer l'adaptation requise pour la QoS de bout en bout. Une décision du handover vertical dépend de plusieurs paramètres concernant le réseau auquel le terminal est relié et celui auquel il sera connecté. Par exemple, nous aurons les informations de largeur de bande de réseau, la charge, la couverture, le coût, la sécurité, la QoS, ou même la préférence de l'utilisateur. La préférence de l'utilisateur est importante, par exemple, si le nouveau réseau, dans lequel un dispositif mobile exécute un handover, n'offre pas la sécurité, l'utilisateur peut décider de rester dans l'ancien réseau. Selon la couverture, un utilisateur peut souhaiter employer un lien sécurisé pour son trafic officiel d'Email (par exemple en utilisant GPRS), mais peut choisir un lien moins coûteux pour accéder à l'information du Web (par exemple WLAN).

Dans notre contexte de réseaux hétérogènes, nous devons considérer le handover vertical car il permet l'adaptation entre différentes technologies du réseau d'une manière flexible. La mobilité deviendra alors réalité. Le potentiel additionnel est la possibilité pour définir des politiques pour l'accès réseau. Des règles peuvent être définies pour employer le LAN au lieu de WaveLAN dans la portée de l'optimisation de largeur de bande,

quand le LAN est disponible. Un autre scénario serait le WaveLAN qui est meilleur sur le plan coût, quand il est présent et GPRS dans les autres cas.

De plus, pour évaluer les temps de réaction, il est important d'identifier le rôle des acteurs (terminal et réseau) intervenant dans le mécanisme de handover. Les types suivants sont définis, à savoir : Initiateur, Contrôleur et Agent (collecteur des informations). Les différents cas sont :

- Dans une procédure du handover, l'acteur qui active la procédure est appelé Initiateur, celui qui contrôle le changement est appelé Contrôleur et celui qui prend les mesures pour alimenter les informations décisionnelles est appelé Agent.

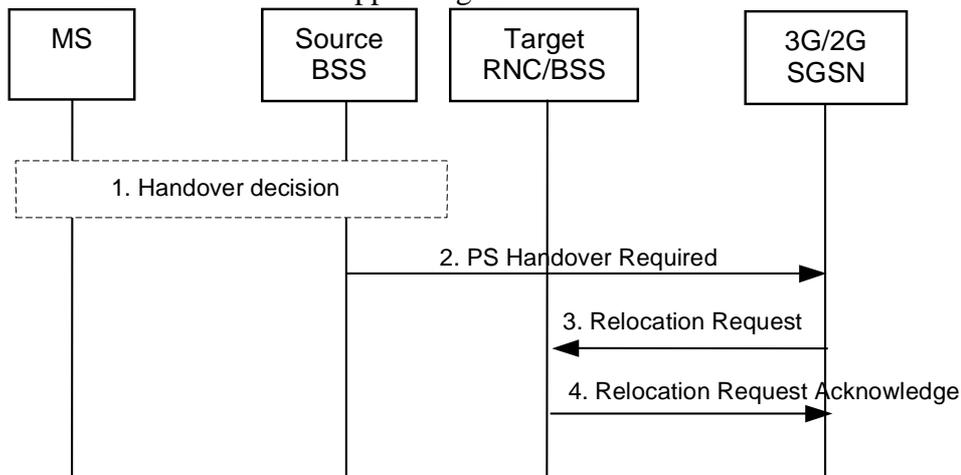


Figure II.6 : PS Handover :Phase de Préparation [7]

Par exemple (Figure II.6), dans une phase de préparation du handover dans le domaine de PS (Packet Switch), c'est le BSS ancien qui décide de faire le handover, la procédure de handover est contrôlée par le SGSN qui se trouve en bordure du réseau cœur, le rapport de mesure sur la cellule où le terminal mobile arrive est fourni par le mobile, l'information de capacité de la cellule est indiquée par le RNC/BSS du réseau d'accès.

En résumé, dans la plupart des cas que nous avons étudiés, le handover est initialisé par le point d'attache du sous réseau d'accès au réseau cœur (réseau initiateur), contrôlé par le point d'attache du réseau cœur (réseau contrôleur) et selon un rapport de mesure fourni par le terminal (Agent terminal) et l'environnement de réseau (Agent réseau).

Par ailleurs, le handover peut être classé aussi selon les différents objectifs de qualité de service :

- Seamless handover

Ce type de handover est destiné à offrir une session sans coupure pour l'application et à respecter la QoS de bout en bout, ainsi que la continuité de la session sans couture.

- Smooth handover

Ce type de handover est destiné à minimiser la perte de paquet.

- Fast handover

Ce type de handover est destiné à minimiser le délai du handover. Le délai du handover est la différence temporelle entre l'émission/réception de paquet du router précédent et l'émission/réception de paquet du router suivant.

En conclusion, les différentes techniques que nous venons d'évoquer couvrent bien la mobilité du terminal de façon transparente à l'utilisateur. Mais le terminal ne représente qu'une partie de la mise en relation de bout en bout, qu'en est-il de cette liaison lorsque l'utilisateur se déplace et change de terminal ? Et que devient la QoS de bout en bout par rapport au service auquel nous accédons. C'est ce que nous allons étudier dans les paragraphes suivants.

II.2.2 Mobilité de l'utilisateur

La « mobilité de l'utilisateur », dénommée aussi « mobilité personnelle », se rapporte à la capacité de l'utilisateur (everyone) à utiliser n'importe quel terminal (anyhow) pour accéder à des services (every services) anywhere et anytime (Figure II.7). L'utilisateur désire au cours d'une même session changer de terminal tout en gardant la personnalisation de son service.

Cette mobilité personnelle induit les fonctions suivantes sur lesquelles nous reviendrons dans nos propositions :

- Localisation
- Adaptation (les profils)
- Accès sécurisé
 - Identification
 - Authentification
 - Autorisation (contrôle d'accès)
 - Enregistrement/ dé-enregistrement.

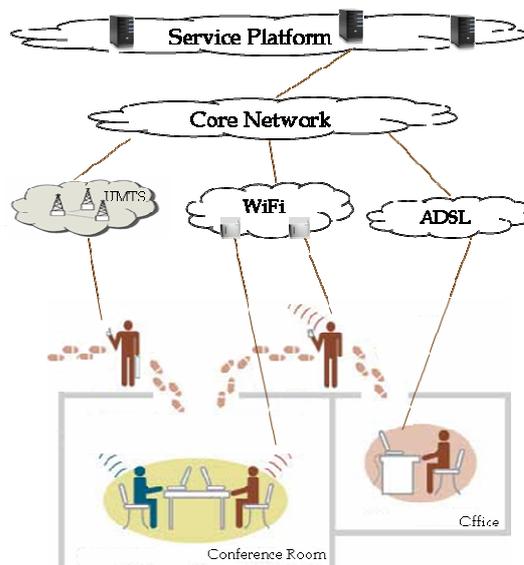


Figure II.7: Mobilité de l'utilisateur

Les solutions existantes sont peu nombreuses.

Nous retrouvons le protocole Mobile IP [4] en établissant l'association entre le CoA du terminal destinataire et l'adresse permanente de l'utilisateur.

Il y a aussi le protocole SIP (Session Initiation Protocol) [9] qui supporte cette mobilité (Figure II.8) dans la couche Application par la fourniture d'une même adresse logique. Il y a une mise en correspondance des différentes adresses. C'est au Proxy (P CSCF) et à la fonction de redirection de faire la traduction entre plusieurs identités sur différents terminaux. Cette dernière redirige la communication durant la session de service. Les adresses logiques de `alice@columbia.edu`, `7000@columbia.edu` et `Alice.McBeal@columbia.edu` pointent en fait vers le même utilisateur (Alice).

Par ailleurs, l'utilisateur d'aujourd'hui possède plusieurs terminaux et plusieurs accès, ce qui pose le problème de sa joignabilité.

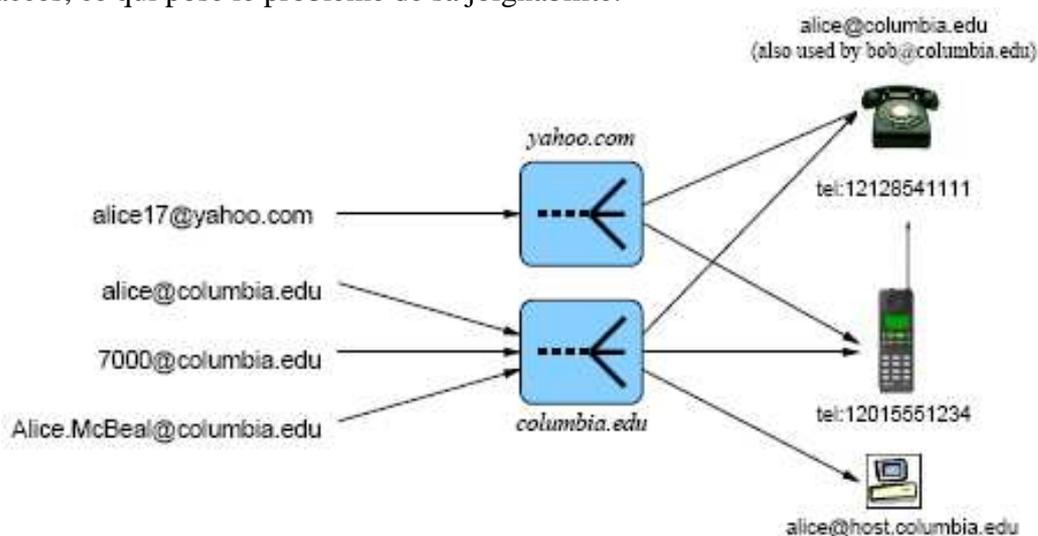


Figure II.8: mobilité de l'utilisateur supporté par SIP [10]

En effet, la communication s'est enrichie au cours des âges d'une panoplie d'outils d'abord de diffusion (telex, TSF, télévision, ...), puis d'échange (courrier, le fax, le télex...) et beaucoup plus récemment de moyens de communication plus interactifs et libérés des contraintes de temps, de lieu et d'espace (téléphone, visioconférence, internet...) rendant ainsi les interactions plus faciles et accessibles. 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. En fait, on peut dire que l'utilisateur dispose maintenant d'une «bulle de communication», individuelle, personnelle, avec différentes possibilités pour communiquer tels que téléphone fixe, téléphone mobile, fax, messagerie électronique, messagerie instantanée. En s'affranchissant de problèmes spatiaux et temporels. Mais il ne faut pas oublier que c'est la même personne qui le matin est abonné entreprise et le soir abonné résidentiel. Même parfois, il porte les casquettes alternativement soit à son domicile soit à son lieu de travail. C'est pourquoi nomadisme et ubiquité sont désormais intégrés dans les usages et les comportements.

En conclusion, pour résoudre la « mobilité de l'utilisateur », nous devons gérer cette joignabilité à travers la connaissance de son parc de terminaux et de sa localisation pour qu'il puisse, dans toutes les occasions (lieu et temps), indépendamment des terminaux utilisés être accessible. Nous devons également tenir compte des préférences et de la QoS demandée afin d'organiser de façon

transparente et sans couture les changements de ressource suite à ce type de mobilité.

II.2.3 Mobilité de service

L'augmentation massive des besoins de service a conduit ces dernières années les opérateurs de télécommunications à prendre en compte de plus en plus sérieusement la préparation des offres de service dans leurs stratégies. De plus, les opérateurs s'intéressent au développement de service prêt à être lancée sur le marché avec un TTM minimum et un ROI le plus rapide. En considérant les services existant et les concurrences entre les différents opérateurs, la transorganisation devient un verrou à lever. Ce dernier introduit la notion de la « mobilité de service ».

En fait, si l'on traite cette mobilité à travers une vraie transorganisation avec l'hétérogénéité du paysage du monde Télécom, nous devons repenser l'architecture. Cette architecture doit tenir compte des aspects de services composables provenant de plusieurs fournisseurs et du contexte ubiquitaire.

Pour l'instant nous trouvons (Figure II.9- gauche) principalement des architectures orientées « client-serveur », c'est-à-dire des architectures verticales et monolithiques. Or la prise en compte de la mobilité de terminal peut impacter la QoS de bout en bout. Donc, si on veut offrir l'accès au service de n'importe où, il faut pouvoir offrir le même service dans la zone de présence de l'utilisateur. L'architecture devient alors de type horizontal (Figure II.9-droite) avec des propriétés d'ouverture.

Architecture verticale	Architecture horizontale
Client serveur	Distribuée
Un fournisseur	Plusieurs fournisseurs
Plusieurs liens de services	Un lien de services
Couplage fort	Couplage lâche

Tableau II-1: L'architecture verticale versus architecture horizontale

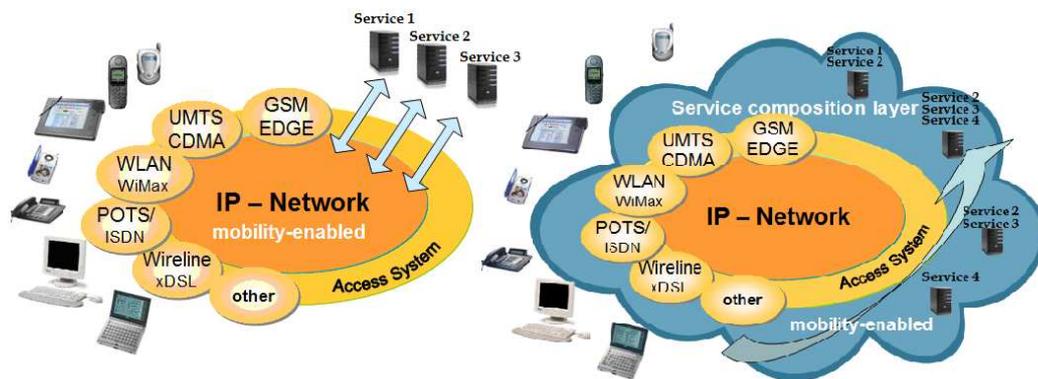


Figure II.9: mobilité de service

Ce que préconise ce schéma, c'est une couche overlay qui supporterait la composition de service comme nous le verrons plus tard dans le chapitre de nos propositions. Cette composition de service permettrait de répondre à tous les besoins de notre « User Centric ».

Grâce à cette vision de service, la virtualisation des services, la mutualisation de services et le support de la décision flexible sont possibles. La mobilité de service peut donc être supportée.

II.2.4 Mobilité de session

La prise en compte des mobilités lance un défi important pour maintenir les sessions sans coupure.

Dans le domaine de télécommunication, la notion de « session » est une succession d'interactions entre les deux extrémités du transport. Les services de transport sont des services de communication point à point, c'est-à-dire avec deux interlocuteurs. Mais le modèle référentiel OSI (Open Systems Interconnection) doit convenir aux communications multipoints. Donc, une session [11] est définie comme un ensemble d'échanges point à point avec un interlocuteur engagé dans la mise en relation.

Dans le domaine d'IP, session **Erreur ! Source du renvoi introuvable.** est définie comme la connexion logique entre les parties impliquées dans une communication basée sur PS (Packet Switched). Ce terme est utilisé pour les connexions IP. Dans le cas des « sessions Web » notamment, le terme « session » désigne une connexion de niveau application, voire un contexte partagé par plusieurs connexions de niveau application sans support protocolaire. C'est un usage dérivé des systèmes d'exploitation.

Il y a des sessions proposées entre différentes entités du réseau. Par exemple, la session de la carte UICC (UMTS Integrated Circuit Card) à partir d'un terminal [13]. La séquence de commandes reliés est délivrée à cette carte et à la suite de laquelle nous obtenons les réponses. Une application est commencée par la sélection et terminée par la désélection ou la fin de la session de la carte. Dans la session de la carte, il y a la session de channel [14], qui est pour échanger les requêtes et les réponses entre la carte et l'entité externe dans le réseau. Au niveau de l'accès, il y a la session de GPRS [15] qui peut s'offrir lorsque l'attachement GPRS ; pour le mode PS, la session IP-CAN [16] concerne l'association entre UE et réseaux IP. Avant l'entrée du monde IP, pour la session du domaine de PS, la session PDP [17] pour échanger les contextes PDP qui indiquent les profils de QoS jusqu'à GGSN. TISPAN [18] (Telecommunications and Internet converged Services and Protocols for Advanced Networking) définit une session multimédia multicast entre des expéditeurs et récepteurs, contenant les flux de données. Elle est supportée par les réseaux NGN (Next Generation Network) et les connectivités IP. IMS (IP Multimedia Subsystem) fournit une couche intermédiaire au cœur des réseaux pour passer du mode appel classique (circuit) au mode session. Un utilisateur peut ouvrir plusieurs sessions multimédia IP au cours d'une même communication. Par exemple rajouter une session de chat à de la vidéo, envoyer une photo pendant la conversation.

Par ailleurs, TINA (Telecommunications Information Networking Architecture) [19] introduit trois types de session (Figure II.10): la session d'accès, la session de service et la session de communication pour identifier les différents acteurs intervenant dans le service demandé par l'utilisateur.

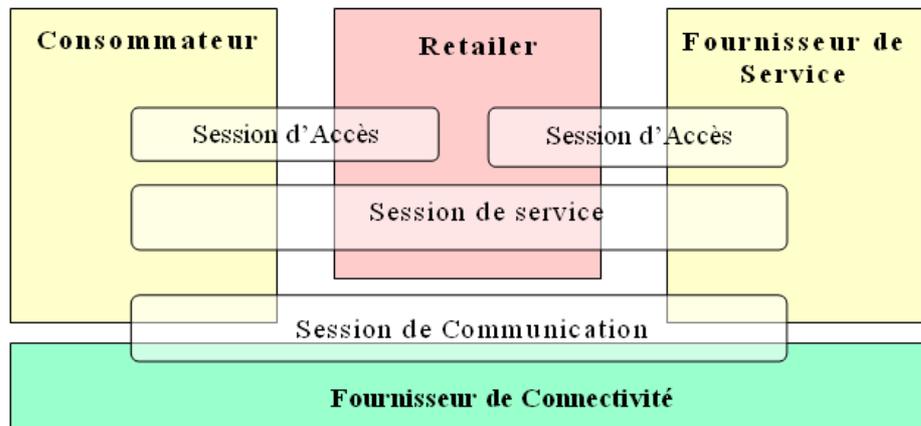


Figure II.10 : Définition de sessions dans TINA

La session d'accès est un environnement pour garantir l'association entre les objets définis. Elle couvre les objets ainsi que les interactions nécessaires pour la création et la gestion de la session service.

La session de service (Figure II.11) est un environnement pour un service spécifique, elle relie un utilisateur (résidant dans le contexte d'un terminal) et un fournisseur de services sur le réseau, qui repose sur une session d'accès.



Figure II.11 : Session service

Cette session permet de véhiculer la signalisation à destination des services. Une session de services peut déboucher, dans le cadre de l'exécution de certains services, à l'établissement d'une session d'information au dessus de la session de services, correspondant à un flux d'information entre une source d'information et un récepteur. C'est par exemple le cas pour des services à base de transfert de fichiers (FTP).

La session de communication représente la relation existante entre le terminal de l'utilisateur et le réseau. L'existence d'une telle session assure à l'utilisateur la possibilité de communiquer avec le réseau, ce qui lui permet le cas échéant d'ouvrir une session de services.

En conclusion, la session est toujours la mise en relation temporelle, concernant un service dédié. Les spécifications ont sortie récemment pour la redéfinition de session de bout en bout et multi services. Plus précisément, en rajoutant un appel basé sur circuit à la session d'IMS, la session devient un session CSI [20] ou même combinatoire [21] et entre deux utilisateurs. Dans ce cas là, les instances de services individuels sont orientés par un UE de l'utilisateur A et se sont terminés dans un autre UE de l'utilisateur B.

Mais si l'on veut prendre en compte les besoins de NGN, notamment les impacts de toutes les mobilités, il nous reste à traiter le dernier tronçon de la mise en relation c'est-à-dire, la prise en compte de la mobilité de service.

Par ailleurs, le protocole SIP comme nous l'avons brièvement vu, il fournit une transparence de redirection du service vers un autre terminal par « SIP Registrar » et « Location Services Server ». Ces composants maintiennent tous les terminaux possibles liés à un utilisateur et les adresses permanentes et provisoires de chacun de ces terminaux. SIP est un support qui nous permet, entre autre, d'avoir une vision de tous les équipements qui sont autour d'un utilisateur et d'avoir la possibilité de choisir la destination pour un utilisateur.

De même, les normes Mobile IP de l'IETF propose un home agent (HA) et fournit la transparence à cette mobilité en masquant les changements des adresses IP pour la couche application du nœud mobile et du nœud correspondant. Par conséquent, une connexion TCP (ou SCTP) peut être gardée au-dessus tandis que le nœud mobile change de sous réseau IP.

Cet ensemble de sessions parallèles ne permet pas d'avoir une vision « User Centric ».

En effet, pour l'instant la « mobilité de session » n'est proposée que du côté des terminaux et du réseau, avec un handover BBM (Break-before-make), dans le cas « avec couture ».

En effet, le handover BBM coupe le raccordement existant avant d'établir l'autre, donc avant que la transmission avec le nouveau point d'accès soit établi.

Par contre, le « sans couture » demande un handover MBB (make-before-break) fait le nouveau raccordement avant de couper celui qui existe.

Évidemment, les utilisateurs ne sont pas disposés à accepter la dégradation de la qualité de service, de la sécurité et des capacités. Si on veut faire de la qualité de service de bout en bout, il nous faut aussi avoir un « handover de service » avec une continuité de service et une certaine flexibilité pour couvrir toutes les mobilités en même temps.

II.3 Périmètre d'étude de la thèse

L'analyse du contexte que nous venons de faire a identifié les solutions existantes et a mis en exergue les besoins de nos différentes mobilités pour répondre à la vision « User-Centric ». L'intersection de quatre mobilités nous a permis d'identifier la « mobilité de la session » comme support de notre objectif de « continuité de service ».

En effet, les exigences de l'user centric nous dirigent vers une session qui gère l'ensemble de ses demandes caractérisées par le fait que les services soient accessibles par n'importe lequel de ses terminaux à travers la meilleure connectivité. Se pose alors, d'une part, le problème de la continuité de la session qui lui permet de maintenir l'accès à l'ensemble de ses applications et d'autre part, le problème de l'adaptation aux terminaux utilisés. En considérant l'application de bout en bout, le but de la continuité de service est alors d'assurer une continuité de session globale en tenant compte non seulement du déplacement du terminal et du réseau, mais aussi de celui de l'utilisateur et de celui du service. La mobilité de session de service est définie comme une

organisation de service personnalisée qui tient compte de toutes les mobilités (user, terminal et service) et des différentes hétérogénéités.

Notre périmètre d'étude est donc :

- Au niveau de service de permettre la mobilité de service. L'axe d'investigation est de palier l'architecture « Client-Serveur » en allant vers une architecture du type horizontal avec un accès au service dynamique et flexible (transorganisationnel) ;
- La gestion des ressources de l'utilisateur (l'ensemble de ses terminaux, de ses accès et de ses services) pour supporter les mobilités ;
- La gestion de la session orientée utilisateur qui se veut mobile, donc dynamique, flexible et crosslayer ;
- Au niveau du bout en bout assurer la QoS.

Avant de faire les propositions, nous allons dans le chapitre suivant étudier les solutions existantes sur les points d'investigation que nous venons d'identifier. Les architectures de service et les approches autonomiques en sont les axes importants.

II.4 Références

- [1] "Considerations of horizontal handover and vertical handover" ITU, April 2007
- [2] 3GPP TS 25.832, Manifestations of Handover and SRNS Relocation. V4.0.0, 2001
- [3] RFC3344 : IP Mobility Support for IPv4, Aug. 2002
- [4] RFC 3775 : Mobility Support in IPv6, June, 2004
- [5] S. J. Koh et al., "Use of SCTP for IP mobility," in IETF Internet Draft, June 2003, draft-sjkoh-sctp-mobility-02.txt.
- [6] Unlicensed Mobile Access (UMA) architecture (Stage 3), Release 1.0.4., 2005
- [7] 3GPP TS 43.318, "Generic access to the A/Gb interface"; Stage 2, Release 8, 2008
- [8] MIH IEEE 802.21 Draft Standard, P802-21-D01-00, 2007
- [9] 3GPP TR 29.962 "Signalling interworking between the 3GPP profile of the Session Initiation Protocol (SIP) and non-3GPP SIP usage" V6.1.1, 2005
- [10] Henning Schulzrinne et al Application-Layer Mobility Using SIP, WoWMoM'99, 1999
- [11] ISO 8327-1 OSI Session Protocol, 1996
- [12] ETSI TS 122 115 V6.5.0 Technical Specification Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Service aspects; Charging and billing (3GPP TS 22.115 version 6.5.0 Release 6), 2005
- [13] SCP TR 102.216 V3.0.0: Vocabulary for Smart Card Platform Specifications, 2003
- [14] ETSI TR 102 216 V3.0.0 Technical report Smart cards; Vocabulary for Smart Card Platform specifications 2003
- [15] ETSI TS 122 078 V8.0.0 Technical Specification Digital cellular telecommunications system (Phase 2+); Universal Mobile

- Telecommunications System (UMTS); LTE; Customized Applications for Mobile network Enhanced Logic (CAMEL); Service description; Stage 1 (3GPP TS 22.078 version 8.0.0 Release 8), 2009
- [16] ETSI TS 129 214 V7.1.0 Technical Specification Universal Mobile Telecommunications System (UMTS); Policy and charging control over Rx reference point (3GPP TS 29.214 version 7.1.0 Release 7) , 2007
- [17] ETSI TS 129 210 V6.7.0 Technical Specification Universal Mobile Telecommunications System (UMTS); Charging rule provisioning over Gx interface (3GPP TS 29.210 version 6.7.0 Release 6), 2006
- [18] ETSI TR 180.000, Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN Terminology Feb. 2006
- [19] TINA-C “Service architecture” V. 5.0, 1997
- [20] 3GPP TS 123 279 V7.7.0 Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Combining Circuit Switched (CS) and IP Multimedia Subsystem (IMS) services; Stage 2 July, 2007
- [21] ETSI TR 122 979 V8.0.0 Technical Report Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Feasibility study on combined Circuit Switched (CS) calls and IP Multimedia Subsystem (IMS) sessions (3GPP TR 22.979 version 8.0.0 Release 8) , 2009

Chapitre III Etat de l'art

Dans le chapitre précédant, nous avons délimité notre périmètre d'étude et nos axes d'investigation qui nous ont dirigés tout au long de notre recherche bibliographique. L'objectif de cette thèse est d'assurer par l'autogestion une continuité de service *intégrée et sans couture* dans un contexte mobile. La considération de la mobilité de service à des fins de QoS dans un paysage NGN nous a conduit à repenser les services indépendamment des technologies réseaux. Les défis de ce contexte que nous avons dénommé NGS (Next Generation Service) sont d'abord relatifs à la création de ces services avec les aspects de personnalisation, d'ubiquité (omniprésence des services) et donc plus généralement ceux de la composition de service (§III.1) pour satisfaire les besoins de notre « User Centric ». Puis nous avons examiné les aspects architecturaux pour lesquels nous avons étudié les architectures de service (§III.2). Pour terminer, nous avons appréhendé les approches autonomiques pour situer nos propositions sur les aspects de gestion des approches autonomiques (§III.3). Une conclusion (§III.4) est faite pour résumer ce chapitre.

III.1 Composition de Service

La motivation d'avoir un ROI maximum et un TTM minimum avec une vision « User-Centric » a dirigé notre recherche sur une nouvelle conception de service : la composition de service. Plusieurs tendances existent sur la marché : Service Web (§III.1.1) pour la simplification et la réutilisation de service, Web 2.0 et Web 3.0 (§III.1.2) pour une partageabilité de contenu et de l'application, SOA (Service Oriented Architecture) (§III.1.3) pour le couplage lâche et SaaS (Software as a Service) (§III.1.4) pour gérer les composants de service ubiquitaires. Nous comparons ces différentes approches et précisons nos exigences dans la conclusion (§III.1.5).

III.1.1 Service Web

Depuis les dernières années, l'utilisation de Services Web a connu une popularité grandissante auprès des développeurs Web. Cette tendance est due aux techniques de 'mash up' où plusieurs services Web sont combinés élégamment, en utilisant un langage de programmation coté serveur. Service Web est définie par le W3C [1] comme "un système logiciel pour supporter les interactions de machine à machine au dessus d'un réseau." Un exemple est donné (Figure III.1).

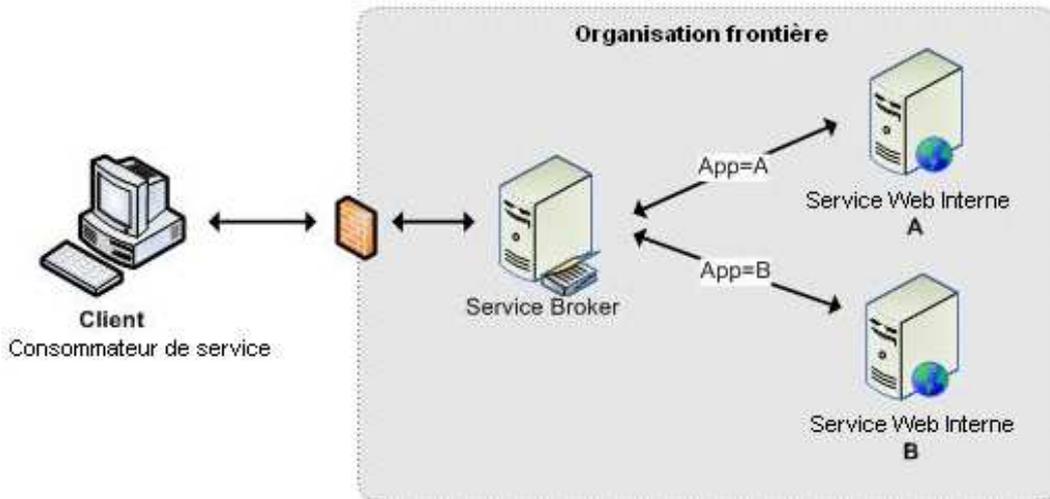


Figure III.1 : exemple de Service Web

Les Services Web sont souvent utilisés comme des APIs Web qui peuvent être accédés à travers un réseau et exécutés dans un système distant hébergeant les services demandés. Le "Service broker" (Figure III.2) est l'interface entre les demandeurs et les fournisseurs de services. Service Web est en fait un premier dénominateur commun pour la création de service.

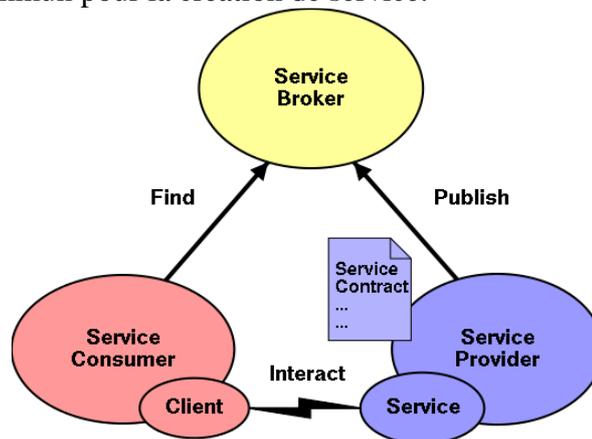


Figure III.2 : procédure d'un Service Web

Basé sur des expériences de l'Internet, Service Web est considéré comme une des solutions possibles pour fournir les services demandés par l'utilisateur à travers un réseau télécom. Principalement, les Services Web sont exécutés dans un serveur et les fonctions proposées sont du côté réseau.

Une application du concept « Service Web » est IMS [3]. Il est proposé comme une technologie motrice qui permet la convergence des services de l'Internet et de

réseau cellulaire. Il doit supporter les services traditionnels de Télécom, ainsi que des services multimédias comme la vidéo conférence.

En fait, les Services Web permettent la création de services demandés par l'utilisateur grâce à des composants proposés à travers le Web. Mais il a ses limites, par exemple : il est difficile d'ajouter ou de modifier, dynamiquement, un composant de service. Le "Service broker" reste une application client-serveur peu flexible. La question est : est ce suffisant pour les besoins de mobilité et d'application client du NGN ?

III.1.2 Web 2.0 /Web 3.0

Le Web 2.0 [3] [4] est défini comme un concept d'utilisation d'Internet qui a pour but de valoriser l'utilisateur et ses relations avec les autres. L'Internet est alors redéfinie comme non seulement un média (où les sites Web sont autant d'îlots d'informations isolées) mais aussi comme une plate-forme. Les données sont considérées comme les connaissances implicites. Web 2.0 est en fait un socle d'échanges entre les utilisateurs (l'auteur parle d'intelligence collective) de services ou d'applications en ligne (Figure III.3).

L'infrastructure du Web 2.0 est complexe et changeante, mais elle inclut les logiciels de serveur, la syndication de contenu (RSS qui signifie Really Simple Syndication), les protocoles de messagerie des standards de navigation, et des applications clientes diverses (les plugins, ou greffons, non-standard sont généralement évités). Ces approches complémentaires fournissent au Web 2.0 les capacités de stockage, de création et de diffusion qui vont au-delà de ce qui était précédemment attendu des sites web.

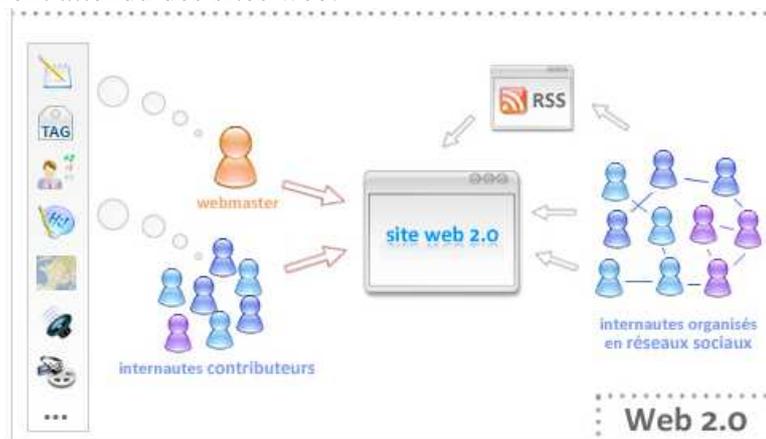


Figure III.3 : Web 2.0

Application comme "many-to-many publishing" (blogs, social book marking et wikis), logiciel social, les interfaces de la programmation application web et les services en ligne nous permettent d'avoir une possibilité de renforcer le service demandé par l'utilisateur. Des « Communauté d'intérêt » sont constituées pour partager les connaissances (Swicki), ou pour avoir un même sujet de discussion (Facebook) ou le réseau social (Expedia pour la réservation des billets basée sur une comparaison de prix en temps réel). Avec certaine plate-forme ou outil (Yahoo! Pipes Mash-up platform [5][4]), l'utilisateur peut même créer son journal par le filtrage des nouvelles en fonction de ses préférences prédéfinies.

Web 3.0 est considéré comme la troisième génération des services de web basé sur l'Internet, qui a pour but d'offrir à l'utilisateur des expériences plus ou moins productives et intuitives. La définition officielle du web 3.0 est "la création de contenu de grande qualité et de services produits par des individus utilisant la technologie de web 2.0 comme plate-forme". Par rapport à Web2.0, Web 3.0 [6] voudrait donner plus de personnalisation et rechercher plus efficacement les données Web. Malheureusement, bien que Web 2.0 et Web 3.0 supportent le partage d'UGC (User Generated Content) et UGA (User Generated Application), ils proposent également une composition de service avec les composants réutilisables et partageables, aucun ne supporte vraiment le partage d'un même service. De plus, l'utilisateur ne peut faire qu'une seule invitation pour une application, l'ajout d'une application induit obligatoirement une autre invitation..

Ces approches adressent plus le contenu et l'architecture reste de type client-serveur. Elles ne sont pas suffisantes pour faciliter la création de service orienté utilisateur.

III.1.3 SOA

L'idée principale de SOA est de mettre en œuvre les services ou les applications sous forme des composants logiciels. Par exemple, [7] a proposé des services (Figure III.4) qui sont en fait des composants autonomes qui implémentent une ou plusieurs fonctionnalités bien définies à des acteurs humains, à d'autres services, ou à d'autres parties du système. Le service fournit un accès vers une ou plusieurs fonctions en masquant l'hétérogénéité du système. Le service est utilisé sous un Contrat de service pour une demande de l'application.

L'interaction et de la communication d'une collection de services permet de composer des applications par une agrégation de services.

Cette organisation et décomposition de services rendent possible la mise en place rapide de processus métiers réellement transverses tout en préservant un couplage faible facilitant leur modification ou refonte totale et la réutilisabilité de services eux même.

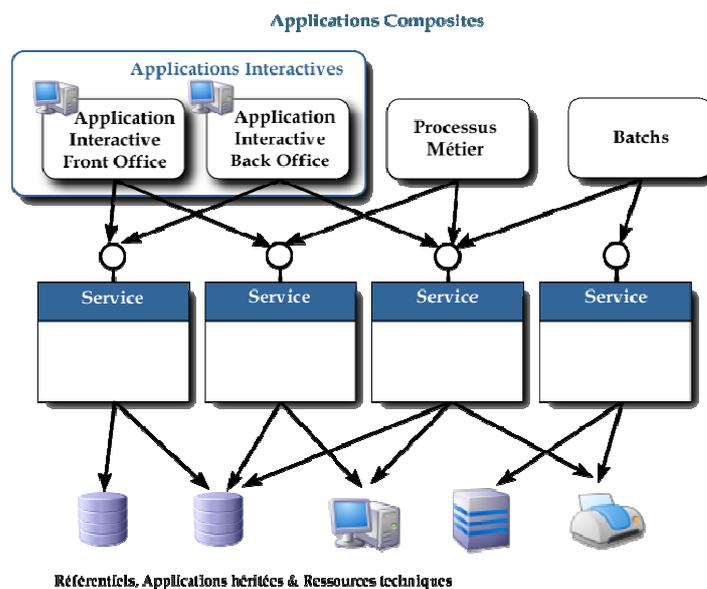


Figure III.4 : composition de service exemple 1

Dans une autre proposition (Figure III.5), SOA [8], la décomposition est faite en trois étapes :

- La première est la découverte des opérations et l'identification de phases exposées pour un processus modélisé. Il s'agit d'un regroupement d'activités rattachées aux catégories formant le périmètre fonctionnel des utilisateurs.
- La seconde consiste à décomposer les opérations et les phases. Chaque opération et phase deviennent ainsi un orchestrateur d'appel vers les services.
- La troisième décompose chaque service exposé en catégorie sous la forme de méthodes attachées aux classes qui constituent la catégorie d'appartenance.

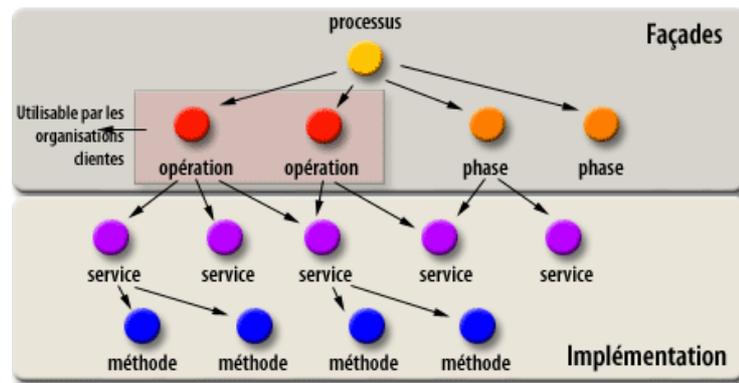


Figure III.5 : composition de service exemple 2

III.1.4 SaaS (Software as a service)

Architecturalement, l'application de SaaS [9] [10] est similaire aux autres applications qui suivent le concept de l'orienté service. Tout comme l'ASP (Application Service Provider) ou les applications à la demande, les applications SaaS s'inscrivent dans le groupe des logiciels gérés ou hébergés. A la différence de l'ASP, les applications basées sur le modèle SaaS sont construites d'emblée en mode Web et optimisées pour être délivrées par l'Internet. Le modèle SaaS permet de se décharger de la maintenance, de l'exploitation et de l'hébergement des applications. Le paiement à la consommation est un moyen d'optimiser les coûts.

Par exemple, la plupart des composants [11] (Figure III.6) sont reconnus : « Process service » qui est en fait une interface entre les demandeurs de service (smart clients) et le fournisseur de service. Les services business interagissent avec les bases de données pour les données concernant le business. “Security services” sont responsables du contrôle de l'accès de services logiciels de l'utilisateur.

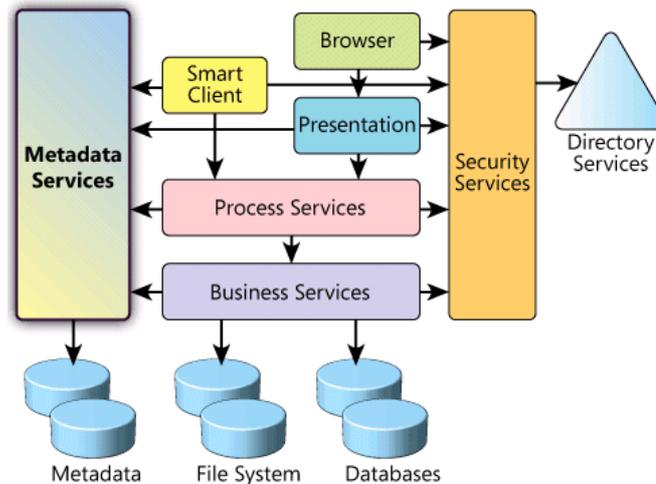


Figure III.6 : exemple de la plate-forme de SaaS

La différence la plus significative est l'ajout de services de métas données, qui sont responsables de la gestion de configuration de l'application pour certains services hébergés. Services et smart clients interagissent avec les services de métas données, afin d'extraire l'information qui décrit les configurations et les extensions qui sont spécifiques à chaque service hébergé.

III.1.5 Conclusion

En conclusion nous pouvons dire que, le Service Web contribue à la simplification et à la réutilisation de service, mais son architecture est orientée « client-serveur ». Web 2.0 et Web 3.0 ont pour objectif la partageabilité de contenu et des applications sous la forme d'option de service. SOA offre un couplage lâche mais la décomposition de service ne permet pas l'accès au composant de service par plusieurs acteurs (user, le contrôle et le management) car il n'est pas mutualisable. Finalement, SaaS propose de gérer les composants de service ubiquitaires mais l'architecture a un couplage fort car vertical. Nous devons donc poursuivre cette démarche de décomposition pour satisfaire les besoins du NGS.

III.2 Architectures de service

Afin de modéliser et mettre en œuvre les services, leur contrôle et leur gestion, différentes architectures sont proposées avec un ensemble de concepts et de règles. Dans cette partie on consignera quelques caractéristiques d'architectures de services existantes. Notre but n'est pas d'être exhaustif, mais de rapporter les concepts les plus porteurs pour la mise en œuvre des services. L'architecture OSA (Open Service Access) a pour objectif la portabilité de service (§III.2.1) par l'interface standardisée à travers laquelle les réseaux de services (ex. Parlay) peuvent accéder aux fonctionnalités des réseaux. Similaire, TSAS (Telecom Service Access and Subscription) est une architecture qui spécifie l'interface d'abonnement de l'utilisateur et l'accès aux services (§III.2.2). SOA introduit le développement de système distribué en termes de services avec un couplage lâche (§III.2.3). Grâce à la composition de service, les composants de service peuvent situer dans un réseau d'overlay (§III.2.4). De plus, le système Peer-to-Peer (§III.2.5) nous permet d'avoir les communications entre les composants de

services afin d'éviter de suivre le modèle Client-Serveur. Quelques réseaux d'overlay dans l'environnement de P2P sont présentés (§III.2.6). Nous spécifions nos exigences de l'architecture dans la conclusion (§III.2.7).

III.2.1 OSA/Parlay

Les spécifications et la standardisation de l'OSA [12] [13] est le résultat du travail de quatre organismes 3GPP CN5, ETSI SPAN12, ITU-T SG11 et le groupe Parlay. L'OSA traite les problèmes de la portabilité des services. Elle fournit une interface standardisée à travers laquelle les nouveaux services peuvent accéder aux fonctionnalités des réseaux indépendamment des technologies de réseau sous-jacentes. Ainsi, les opérateurs peuvent introduire de nouvelles fonctionnalités sans avoir besoin de modifier les applications existantes. Les objectifs principaux étaient de créer une interface API pour toute la communauté des développeurs afin de résoudre les problèmes de portabilité des services. La gestion d'abonnement des applications aux services réseaux, spécifiée par Parlay, est aujourd'hui incorporée en tant qu'interface du *framework* de l'OSA.

La Figure III.7 ci-dessous montre l'endroit où se situe l'API et la passerelle OSA. Les applications qui reposent sur OSA peuvent être aussi bien des services à valeur ajoutée que des services offerts par des fournisseurs tiers.

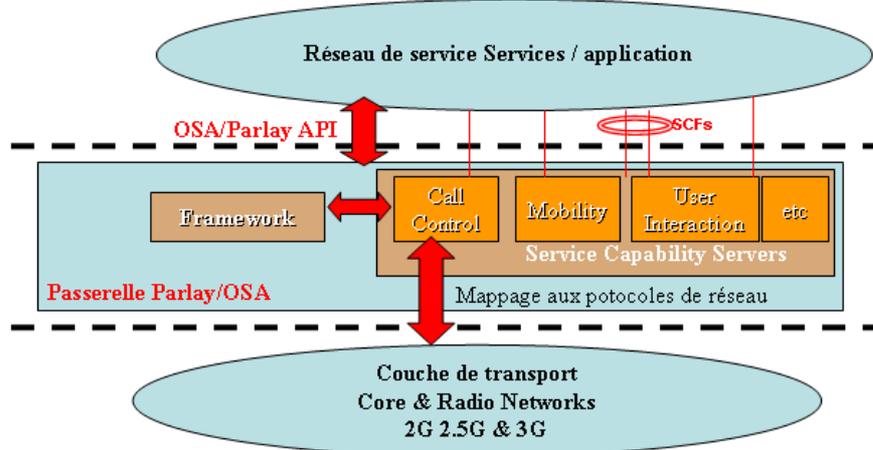


Figure III.7 : La passerelle et l'interface Parlay / OSA

La passerelle OSA/Parlay est composé de deux ensembles d'entités :

- Les SCFs (Service Capability Features) sont des interfaces CORBA standardisées qui fournissent aux applications l'ensemble des fonctionnalités du réseau. Il existe deux types de SCFs. Les premiers sont de type *framework* et offrent des fonctionnalités afin de supporter la gestion d'abonnement et l'enregistrement des fonctionnalités réseaux, et de fournir les mécanismes d'accès sécurisé. Les seconds sont de type *service réseau* et offrent l'accès aux fonctionnalités de réseau (e.g. User Interaction, Terminal Capability, User Profil Management, etc.).
- Les SCSs (Service Capability Servers) (Figure III.8) sont des serveurs de fonctionnalité de service qui jouent le rôle d'intermédiaire entre l'interface SCF et l'entité de réseau. Ils traduisent les invocations sur les SCFs en événements sur les éléments de réseau.

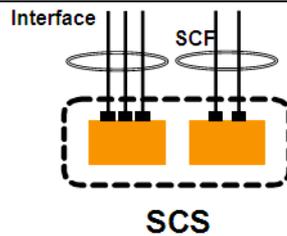


Figure III.8 : Service Capability Server

- *Les mécanismes de base entre Application et Framework*
 - **Authentification** : le modèle d'authentification d'OSA suit un modèle peer to peer, mais pourtant il n'oblige pas une authentification mutuelle. En effet, une fois qu'un accord (hors-ligne) existe, l'application peut accéder à l'interface d'authentification. Elle doit être authentifiée avant d'être autorisée à utiliser n'importe quelle autre interface d'OSA.
 - **Autorisation** : l'autorisation consiste à déterminer les droits d'une application déjà authentifiée.
 - **Découverte de *framework* et les SCFs** : après une authentification réussie, l'application reçoit les interfaces framework disponibles. Elle peut utiliser l'interface de découverte (*discovery interface*) afin de recevoir des informations concernant les SCFs auxquels elle a le droit d'accéder.
 - **Etablissement du contrat de service** : avant qu'une application puisse interagir avec un SCF, un contrat de service doit être établi. Le contrat de service peut comprendre deux parties, une partie hors-ligne et une partie en ligne. L'application doit signer la partie en ligne avant de pouvoir accéder aux SCFs.
 - **Accès aux SCFs** : le *framework* doit fournir des fonctions de contrôle d'accès afin de permettre l'accès aux SCFs ou aux services de données et ceci, pour n'importe quelle méthode d'API d'une application.
- *Les mécanismes entre Framework et SCS*
 - **Enregistrement des SCFs** : les SCFs offerts par un SCS peuvent être enregistrés auprès du *framework*. Dans ce cas, le *framework* serait capable de renseigner les applications sur les SCFs disponibles (découverte).

III.2.2 TSAS

L'OMG de son côté propose TSAS (Telecom Service Access and Subscription) afin de spécifier l'interface d'abonnement de l'utilisateur et l'accès aux services.

Le TSAS [14] définit un ensemble d'APIs qui permettent l'abonnement et l'utilisation des services par utilisateur. Elle définit un détaillant qui s'intercale entre l'utilisateur et le service tiers et qui gère l'abonnement et l'accès des utilisateurs aux services. TSAS s'inspire du principe d'abonnement de Parlay et apporte des améliorations afin de l'adapter pour supporter l'abonnement des utilisateurs à un service. L'ensemble des interfaces définies dans la spécification de TSAS fournit des fonctions à travers lesquelles les opérateurs de réseau peuvent offrir aux fournisseurs tiers un accès sécurisé aux capacités du réseau. La spécification de TSAS est techniquement alignée avec celle de Parlay. Par conséquent, les interfaces de service spécifiées par Parlay peuvent être fournies par TSAS.

Le TSAS définit trois domaines : le domaine d'utilisateur, le domaine de détaillant et le domaine de fournisseur de service (Figure III.9). Les domaines définis par TSAS sont fortement liés aux rôles.

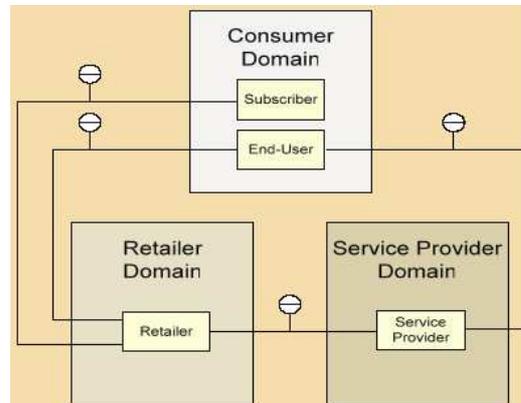


Figure III.9 : Le modèle métier de TSAS [6]

- **Le domaine d'utilisateur** : il contient deux types d'entités, celles qui ont le rôle de l'utilisateur final et celles qui ont le rôle de souscripteur. L'utilisateur final est celui qui utilise le service alors que le souscripteur signe le contrat avec le détaillant afin d'établir l'abonnement aux services pour ses utilisateurs. Dans le cas professionnel, le souscripteur peut être une entreprise qui a un contrat d'abonnement pour ses employés (utilisateurs finaux) alors que dans le cas personnel, le rôle du souscripteur et de l'utilisateur final sont les mêmes.

- **Le domaine de détaillant** : il contient les entités qui ont comme rôle celui de détaillant. Le détaillant fournit une vue intégrée de services aux souscripteurs ou aux utilisateurs finaux. Il offre une gestion unifiée de l'ensemble des services, en particulier le service d'abonnement. Le détaillant est le seul point de contact du fournisseur de service avec le souscripteur et l'utilisateur final. Il doit assurer un accès aisé et la qualité de service. Le TSAS, par le détaillant, permet aux utilisateurs de personnaliser et de customiser les services. Il fournit les possibilités pour configurer et sélectionner les services en intégrant les préférences personnelles.

- **Le domaine de fournisseur de service** : il contient les entités qui ont comme rôle celui de fournisseur de service. Ces derniers offrent leurs services aux utilisateurs finaux (ou souscripteurs) à travers un détaillant. En d'autres termes, ils fournissent une plate-forme de services pour fournir le détaillant. Les services fournis par le fournisseur de service peuvent être la logique de service, le contenu ou les deux.

III.2.3 SOA

La notion de SOA (Service Oriented Architecture) renvoie à une nouvelle manière d'intégrer et de manipuler les différentes briques et composants applicatifs d'un système informatique (comptabilité, gestion de la relation client, production, etc.) et de gérer les liens qu'ils entretiennent. Un modèle populaire de SOA, modèle

référentiel OASIS [15] définit SOA comme : "...Un paradigme de l'organisation et de l'utilisation des capacités distribuées. Il peut être sous le contrôle de différents domaines. Il fournit un moyen uniforme d'offrir, de découvrir, d'interagir et d'utiliser les capacités de produire les effets désirés avec les conditions préalables et les attentes".

SOA supporte le développement de système distribué en termes de services avec un couplage lâche. Ci-dessous une architecture (Figure III.10) proposée par BEA.

Dans SOA, les ressources ambiantes sont mises en réseau pour être autonomes et accessibles sans connaître leurs technologies sous-jacentes. Une caractéristique de SOA est que les services sont des entités indépendantes. Ils peuvent être invoqués de façon standard.

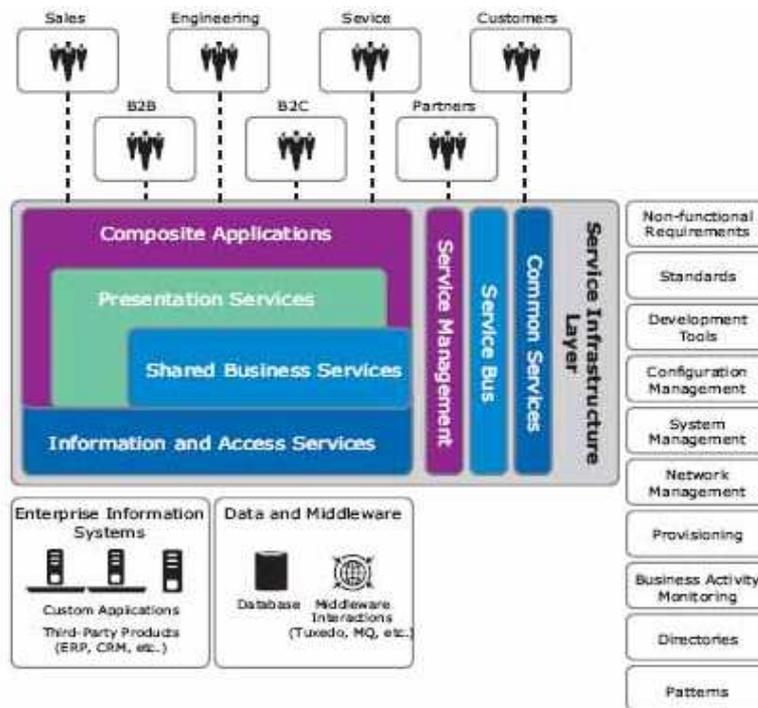


Figure III.10 : Architecture de SOA

Par exemple, la Plate-forme Fiorano SOA™ 2007 (Figure III.11) propose un bus de services entreprises pour traiter les business en temps réel. Tous les services sont présentés sous forme de PS (Peer Server). Les messages s'échangent sur le bus JMS pour faciliter FES (Fiorano Enterprise Server) d'orchestrer les services.

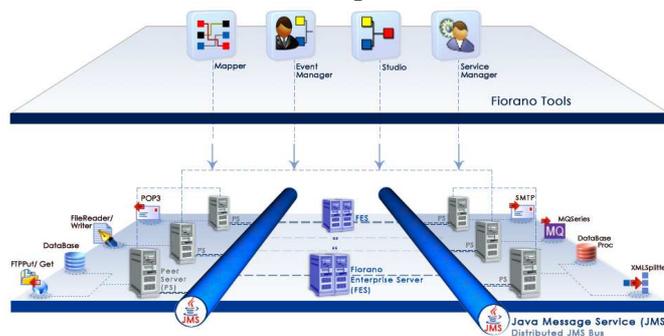


Figure III.11 : architecture de la Plate-forme Fiorano SOA™ 2007

III.2.4 Les réseaux overlay de service

Le but d'avoir une omniprésence de composants de service nous dirige vers une étude sur le réseau overlay.

Dans le contexte du multimédia, un projet sur les réseaux ambiants propose le concept de « Specific Service Overlay Network (SSON) » avec un routage intelligent multimédia et une architecture de transport [16]. Le SSON Overlay est constitué de noeuds (ONodes) et de liens. La dynamique de SSON est contrôlée par l'Overlay Control Space, qui réagit en fonction des changements du contexte de l'utilisateur, de la politique du service et du contexte du réseau. Dans un SSON, le contrôle de la QoS est local à chaque ONode. Or pour contrôler la QoS bout en bout, il faut un contrôle au niveau de chaque composant de service.

Dans [17], Li et al. introduisent des Overlay Brokers (OBs) offrant les fonctionnalités nécessaires pour former le Réseaux Service Overlay qui supporte le routage de QoS. L'algorithme de routage prend en compte la QoS des nœuds et des liens (via pondérations appliquées à la fonction distance) et une partie des chemins de sauvegarde sont maintenus pour réagir à la dégradation de la QoS. Cependant, l'objectif de cette proposition ne porte que sur l'optimisation du service transport et ne concerne pas la QoS de services applicatifs, ni leur composition.

III.2.5 P2P Systèmes

Dans chaque niveau de visibilité, nous avons besoin d'une mise en relation horizontale c'est-à-dire de protocoles de communication. Deux modèles de communication (Figure III.12) s'affrontent actuellement pour le niveau de visibilité service : client-serveur et peer-to-peer.

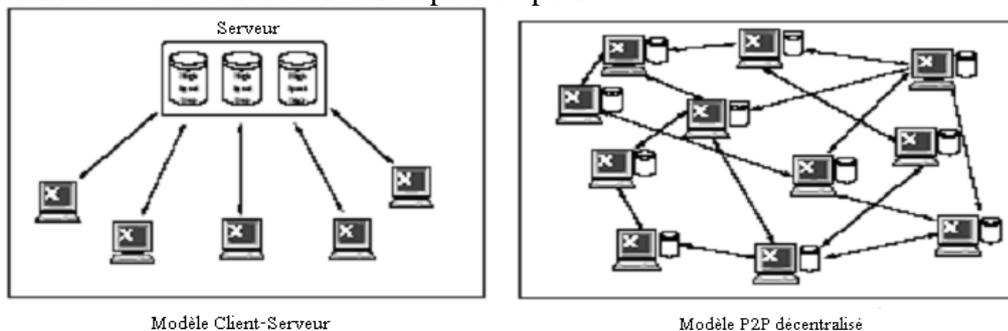


Figure III.12 : Modèles de communication

Pour comparer deux paradigmes de communication, nous devons considérer au moins les trois critères suivants :

Statut des entités communicantes : en effet, c'est le statut qui reflète le rôle de ces entités et induit l'organisation des échanges. C'est ainsi que nous déduisons les acteurs, les workflows et la répartition de la charge du système.

Adressage : car pour pouvoir communiquer, il faut connaître « l'adresse » de l'autre et de cet adressage que découle ou pas la dynamique du réseau. (adressage)
Performance : c'est à travers les performances du système qu'on va mesurer l'efficacité de la communication sans oublier que ce dernier critère va être le reflet de la scalabilité.

Dans le modèle client-serveur, c'est le client qui initie toujours l'échange en envoyant une requête au serveur. Le fournisseur de service alloue alors une ressource physique (stockage, mémoire, CPU) pour satisfaire les requêtes qui lui sont adressées. Cette répartition de rôles induit une architecture centralisée avec comme point central le fournisseur de service (en d'autres termes le serveur). Alors que ce modèle a bien fonctionné dans le passé, déployer des applications scalables en l'utilisant coûterait trop cher (augmenter les ressources pour répondre au nombre grandissant de requêtes).

Par contre dans un système peer-to-peer, les pairs sont à la fois initiateurs et répondeurs dans une communication qui est donc symétrique. Ce modèle de communication d'égal à égal possède un contrôle distribué (pas de répertoire centralisé, ni d'un point de contrôle). Par conséquent, il est dit auto-organisant puisqu'il s'adapte automatiquement aux arrivées, aux départs et aux pannes des nœuds.

Ces réseaux P2P de par leur nature distribuée sont scalables puisque les ressources et donc les coûts engendrés par un passage à l'échelle sont mutualisés.

On obtient le tableau suivant :

	Client-Serveur	P2P
Statut des entités communicantes	Maître/Esclave =>Communication Asymétrique et Architecture centralisée	Egal à Egal =>Communication Symétrique et Architecture distribuée
Adressage	Principalement statique Utilisant DNS, NAT transparent, Port connu (80)	Registres temps-réel et dynamiques.
Performance	Tend à diminuer avec une augmentation des utilisateurs	Tend à augmenter avec une augmentation des utilisateurs

Tableau III-1: Les réseaux P2P

Or, une des exigences de notre modèle est d'avoir une distribution et un partage efficace des ressources. Par conséquent, le mode de communication choisi doit garantir la symétrie et l'égalité entre des composants distribués qui se trouvent dans le même niveau, ce qui induit une communication d'égal à égal.

Pour ces raisons, le modèle client-serveur nous semble inapproprié (bien que traditionnellement les services Internet sont fournis à travers ce modèle) et que par les caractéristiques inhérentes à son fonctionnement, le modèle P2P présente une alternative qui répond à nos besoins. De surcroît, dans le contexte fortement dynamique où nous évoluons, ce P2P va permettre de maintenir un réseau de service efficace (mise à jour rapide, une recherche performante, une mise en relation garantie, etc.).

Alors que leur nature coopérative permet de surpasser l'obstacle de la limitation des ressources, ces applications P2P doivent encore affronter plusieurs défis pour aboutir à gagner en déploiement et en performance.

Malgré la multiplicité des services offerts par les applications P2P récentes, l'opération au cœur de la majorité de ces applications est une localisation efficace

des ressources qui sont identifiées par des clés (adressage dynamique). La façon dont un système P2P recherche les ressources en utilisant ces clés et comment ces dernières sont reliées à ces ressources vont déterminer à quelle catégorie appartient ce système. Parler de *localisation* revient à considérer les deux aspects qui la constituent : *l'indexation et la recherche*. Le but de l'indexation est de fournir un référencement de la ressource alors que celui de la recherche est d'arriver à trouver cette ressource à partir de son « adresse ».

La recherche dans les systèmes P2P non structurés est facile à implémenter mais ne permet pas de passage à l'échelle, sans parler de la charge représentée par le broadcast (petits TTL, recherche partielle avec risque de ne pas aboutir à un résultat ; grands TTL, surcharge). Une ressource peut ne pas être localisée même si elle existe.

Dans le cas des systèmes P2P structurés la recherche est efficace et rapide. La charge induite est faible (pas de broadcast), ce qui permet le passage à l'échelle. Cependant, on note une certaine complexité pour maintenir la cohérence des tables DHT en cas de grande dynamique. Garder la robustesse dans ce cas nécessite les répliquions des associations et/ou des ressources. De plus, la recherche est moins performante que les systèmes non structurés concernant la recherche de mots clés ou des demandes complexes.

Des mécanismes de routage hybrides pourront être proposés pour tirer profit des propriétés intrinsèques à chaque famille de systèmes P2P.

Le concept de Peer-to-peer (P2P) [18] s'est développé énormément ces dernières années.

III.2.6 Overlay de service dans un environnement P2P

Pour profiter des avantages de l'architecture P2P, qui intègre facilement un Peer dans le réseau de composition, nous allons étudier son association avec le réseau overlay.

Raman et al proposent une composition de services basée sur une superposition de Réseau Service Overlay [19]. C'est un travail intéressant, mais la qualité de la demande de services n'est pas considérée, et le système ne peut pas découvrir et localiser dynamiquement les services.

Gu et al proposent un travail important sur la composition des applications des flux dans un environnement peer-to-peer [20]. Leur principale contribution se focalise sur les algorithmes des états globaux et locaux en termes d'exigences de QoS. Ils présentent aussi des algorithmes de récupération pour maintenir la disponibilité et la qualité de service des applications de flux composées. Toutefois, elles supposent que le système de découverte des services est disponible pour trouver des candidatures des instances de service.

Dans le système de P2P (Peer to Peer), SpiderNet [21] propose un réseau overlay de services, qui est sensible à la QoS et basé sur une composition de services. Les composants de service sont en fait des unités d'application. La composition de service se fait par deux couches : une couche de « fonction » (Figure III.13) pour décomposer les fonctionnalités demandés et une couche « service » pour mapper

les fonctionnalités aux composants de service en considérant la QoS et les ressources demandées.

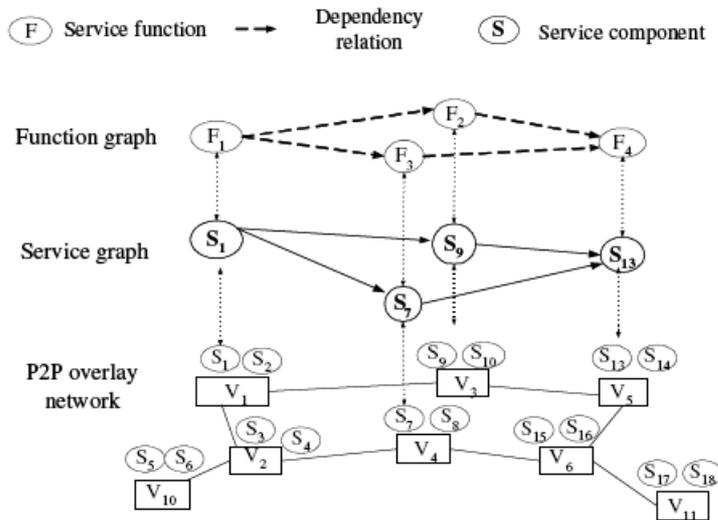


Figure III.13 : architecture de la composition des services

Le réseau overlay de services associé au P2P est montré en utilisant un graphe $G = (V;E)$, V représente les peers, E représente les liens d'overlay. Chaque peer peut supporter plusieurs composants de services. La topologie de réseau overlay peut être maintenue comme un mesh overlay qui est sensible à la topologie. Cependant, le système de composition de service est orthogonal à la topologie de l'overlay de services (Figure III.14).

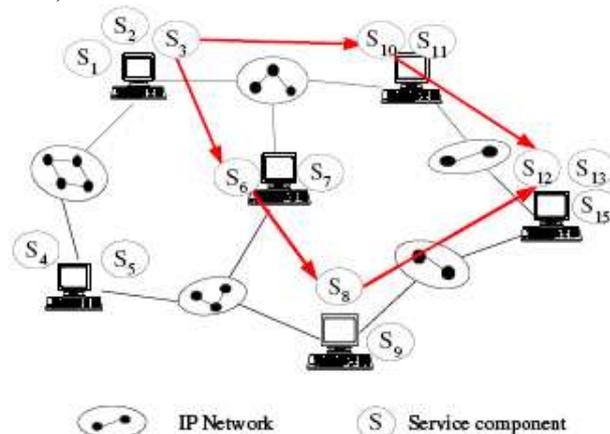


Figure III.14 : Réseau overlay de service dans un système P2P

En fait, la composition de services se fait de façon décentralisée et basée sur les sondages pendant l'initialisation de la session service. Quand une demande de composition de services arrive, le protocole BCPP (Bounded Composition Probing Protocol) est invoqué. Ce protocole s'occupe de l'initialisation des sondages sur la demande de QoS/ressources, la distribution des processus de sondages pour rassembler l'information locale sur chaque peer, la sélection des composants optimale pour avoir une meilleure QoS et finalement l'initialisation de la session service. SpiderNet supporte aussi le recouvrement proactif, grâce au travail effectué pour avoir la fréquence des erreurs dans un réseau P2P, SpiderNet garde, en moyenne, 2,74 groupes de composants de service en tant que backup pour une session service.

III.2.7 Conclusion

Le constat que nous pouvons faire sur les architectures de service présentées est leur caractère statique durant la phase d'exploitation.

En fait, ce que nous recherchons, c'est un système très flexible, avec la vision utilisateur. Ces différentes avancées sont donc à prendre à compte. On se situe dans la mouvance SaaS et réseau overlay de service intégré à une architecture de P2P, mais il faut aller plus loin avec des composants de service qui auraient non seulement des capacités de réutilisation, mais aussi et surtout des capacités de mutualisation et donc de provisioning dynamique.

Pour fournir un accès générique aux services il nous faut aussi une prise en compte dynamique de l'utilisateur et de son environnement ambiant.

III.3 Les approches autonomiques

Avant de présenter une vue d'ensemble sur approches autonomiques, il serait peut être intéressant de rappeler comment la gestion, du point de vue fonctionnel, a évolué pour arriver à l'automatisme (autogestion)[22] [23][24] [25] d'une façon dynamique.

Les premiers travaux sur l'autogestion ont été lancés par IBM qui a introduit le concept de Autonomic Computing [26][27] [28]. L'idée est de créer des systèmes informatiques capables de réaliser les fonctions sans une intervention extérieure. Le monde des télécoms a repris ce principe et a récemment introduit le concept de Autonomic Networking [22][29][30]. L'objectif est de créer des réseaux qui s'autogèrent (à l'instar des systèmes biologiques) et ce en automatisant les principales fonctions de gestion TMN (gestion de faute, de configuration, de performance et de sécurité). L'objectif étant de réduire l'intervention de l'administrateur et de lui cacher les détails de la gestion et du contrôle du réseau. C'est ainsi que plusieurs travaux ont vu le jour proposant des systèmes auto-curatifs (§III.3.1) ou qui s'auto-réparent (self-healing), qui s'auto-configurent (§III.3.2) (self-configuring), qui s'auto-optimisent (§III.3.3) (self-optimizing) et qui s'auto-protègent (§III.3.4) (self-securing). Ces systèmes adressent l'automatisation dans des niveaux différents et utilisent des mécanismes offrant des degrés différents d'automatismes (§III.3.5). Ce que nous avons besoins est le système auto-gérable (self-management) (§III.3.6). Du point de vue fonctionnel, ce système assure les quatre automatismes à la fois.

III.3.1 Les systèmes auto-curatifs

En ce qui concerne les systèmes auto-curatifs, ces derniers pourront automatiquement identifier, analyser, résoudre et réparer les problèmes. L'objectif étant de maintenir de façon continue la disponibilité des réseaux et services et de gagner du temps par rapport aux approches actuelles où parfois plusieurs jours sont nécessaires pour traiter le problème. Par exemple, dans le contexte du WDM, [31] il est proposé un design optimal par backup des composants dans une topologie arbitraire en utilisant des anneaux auto-curatifs (self-healing ring) pour assurer la survie du réseau. L'architecture Openwings [32] fait de l'auto-réparation de service en réaction aux problèmes réseau par les ajoutés des modules de la gestion, à savoir, une gestion basé sur de policy et une architecture flexible des modules de gestions (Mbean).

UMAS (User Manager Agent System) [33] propose une interface entre l'utilisateur et les réseaux de transport pour contrôler le service demandé par l'utilisateur soit toujours réalisable en considérant les ressources disponibles pour sa session d'application. L'autogestion se fait à travers plusieurs agents qui s'occupent de l'apprentissage de la connaissance (Knowledge learner), l'autorisation de l'usage des ressources pour un utilisateur (Granter), l'exécution d'une décision au niveau de routage (Enforcer) la vérification des décisions prises par le Granter (Verifier). OpenORB [34] propose par l'auto-réparation par un contrôle et un Monitoring.

III.3.2 Les systèmes auto configurables

L'auto configuration couvre le besoin de rendre la configuration et la reconfiguration du système et de ses entités plus dynamique et autonome. Elle est nécessaire quand un nouveau composant de système vient s'installer dans l'environnement du système. Ce nouveau composant doit se rendre compte des configurations existantes pour adapter son comportement selon les règles ou les politiques en vigueur tandis que le reste des composants du système doit tenir compte de la présence de ce nouvel élément. Le système pourra ainsi autodéterminer quel est le bon paramétrage de configuration et de l'exécuter. L'auto-configuration doit être réalisée en accordance avec les politiques et les SLA afin d'atteindre le comportement visé. Dans le contexte d'IPv6, [35] propose l'auto-configuration de l'adressage. Dans le contexte des réseaux actifs, la solution NESTOR [36] propose d'automatiser la configuration en utilisant des scripts qui manipulent les éléments du réseau et ce via un serveur (resource directory server). Ce dernier notifie les changements de configuration aux éléments du réseau qui eux se reconfigurent selon les informations reçues. La solution SELFCON [37] utilise un directory serveur intégré, le modèle DEN pour contenir l'information sur la configuration. Les éléments sont informés des changements de politique de configuration (suite aux changements dans le répertoire DEN). Ils pourront se reconfigurer en fonction des informations notifiées à travers des dialogues du type LDAP. [38] supporte l'auto configuration par la gestion PnP (Plug and Play) basée sur les composants de la gestion de L2 et du client/locateur d'IP pour couvrir les communications entre des éléments Réseau Ambient mais aussi le reste du domaine réseau ambient et prédéterminer les locations de l'adresse IP. L'ACS (Ambient Control Space) est aussi configurée dans une façon hiérarchique (élection des super noeuds) en fonction de l'ajout de nœuds dans le Réseau Ambient. La maintenance se fait par l'interaction de tous ces composants.

III.3.3 Les systèmes auto-optimisés

Afin d'éviter à l'administrateur de paramétrer lui même les équipements et les réseaux pour qu'ils fonctionnent correctement, les systèmes doivent continuellement chercher à améliorer leurs performances et opérations. L'auto-optimisation concerne donc l'automatisation de la gestion des performances des systèmes. Ces performances doivent de façon continue être évaluées et optimisées. Les propositions se focalisent sur l'optimisation des métriques, par exemple le temps total du transfert, disponibilité du système, etc. Ces optimisations peuvent être aussi dirigées par les politiques ou par les SLA. [39] propose un paradigme

appelé "objective driven- optimization" Dans [40], une méthode pour superviser automatiquement et adapter les paramètres d'un site de commerce électronique (e-commerce) est proposée. L'approche utilisée est basée sur la recherche automatique des espaces de valeurs des paramètres de configuration. Le système cherche à maximiser la valeur agrégée de ces paramètres. [40] propose également un monitoring dynamique et la reconfiguration des paramètres basés sur une métrique de la QoS agrégée. Alors que dans Self-Star [41], le self-optimization se fait par l'évaluation de performance de systèmes à composants et l'équilibrage de charge. Un modèle analytique est proposé [42] pour l'optimisation de l'utilité des services en fonction des préférences de leurs utilisateurs. Ce modèle décrit l'optimisation du côté réseau (la bande passante réservée pour un service, l'utilité délivrée par le réseau et les coûts des services).

III.3.4 Les systèmes auto-protégés

L'autoprotection concerne les capacités du système à se protéger et à se sécuriser. L'automatisation de la sécurité a pour but de développer un comportement autonome pour l'authentification, l'autorisation ainsi que les capacités d'audit. Les systèmes pourront se "défendre" des attaques et interdire les opérations non autorisées. Par exemple [43] propose un service distribué d'authentification dans le contexte des réseaux ad-hoc. Dans cette solution, de multiples nœuds collaborent afin de traduire le comportement d'un serveur fournissant la certification et l'authentification pour d'autres nœuds du réseau ad-hoc. Dans le contexte des Grid, [44] récapitule les principaux outils utilisés pour l'autoprotection des réseaux et systèmes. Dans [43], nous avons une proposition pour une auto protection en distribuant l'authentification à chaque nœud dans un réseau Ad-hoc.

III.3.5 Les systèmes autonomiques

En plus de ces travaux, plusieurs projets utilisant diverses approches pour l'autogestion ont été réalisés ces dernières années. Nous pouvons citer par exemple l'autonomie architecture d'IBM [45], Bio-Networking de l'université de Californie [46], Autonomia de l'université d'Arizona [47], Cognitive Networks de DARPA [48]. Sans remettre en cause toute la puissance des projets d'automatisation et leurs apports, l'analyse de ces propositions montre que chaque solution se focalise généralement sur l'automatisation d'un aspect fonctionnel de la gestion, c'est-à-dire, elle se concentre sur la réalisation automatique d'une fonction de gestion à la fois (fonction par fonction). Peu de travaux, excepté par exemple [49], couvrent plus d'une fonction. [50] introduit la capacité d'autonomie dans le routeur d'IP (élément autonome) pour atteindre les besoins de la QoS hétérogène des applications de données et multimédias. AE (élément autonome) est pour interagir avec les pairs pour la propagation des connaissances, et offrir une intelligence globale pour faire coopérer les pairs. [51] a pensé que l'évolution vers la gestion autonome des réseaux semble devoir inévitablement passer par une gestion orientée connaissances. Une architecture est proposée pour la réalisation d'un plan de connaissance pour optimiser automatiquement un routeur DiffServ.

III.3.6 Conclusion

De plus, ces solutions automatisent principalement les fonctions de niveau « Réseau » et les aspects qui lui sont associés (VPN auto- configurant, configuration automatique des routeurs, maintenance automatique de la QoS, etc.). Etant jusqu'à présent « invisible » (car dépendant du réseau), peu de propositions adressent l'autogestion dans le niveau « Service » et la plupart des mécanismes mis en oeuvre ne sont pas natifs. Par natif, nous voulons dire naturellement intégré au sein de la solution proposée (dès la conception des éléments auto-gérables) et non prises comme des fonctionnalités rajoutées (fonctions supplémentaires) pour atteindre le comportement automatique et avoir un système auto-gérable.

Par ailleurs, nous constatons que chaque domaine fonctionnel fait appel à des procédures particulières. Pour notre approche, nous avons adopté un fil directeur du comportement autonome, c'est-à-dire donné un objectif au comportement autonome de chaque composant, qu'il soit réseau ou service. L'objectif est que tous les composants coopèrent au même titre au maintien de la QoS pour satisfaire les exigences des clients. Le maintien de la QoS sera alors notre moteur pour l'autogestion dans le cadre de la NGS, c'est-à-dire que les réactions seront induites suivant la QoS (c'est l'autogestion par la QoS). Notre objectif sera alors d'adopter le comportement autonome et proactif pour la NGS afin de maintenir la QoS des applications de nouvelle génération conformément à leur contrat.

III.4 Conclusion

Dans cette partie de l'étude de l'existant, nous avons vu les propositions couvrant les axes principaux de notre problématique en terme de composition de service, d'architecture de service et d'approche autonome. La gestion des mobilités à des fins de continuité de service demande une plus grande intégration et plus de flexibilité. La composition de service, une architecture P2P, un réseau overlay et l'autogestion par la QoS seront nos directions de proposition.

III.5 Références

- [1] <http://www.w3.org/>
- [2] 3GPP TR 29.863 "Feasibility Study for the multimedia inter-working between the IP Multimedia Core Network (CN) Subsystem (IMS) and Circuit Switched (CS) networks", V8.0.0, 2007
- [3] Tim O'Reilly "Web 2.0 Compact Definition: Trying Again". Retrieved on 2007.
- [4] M. P. Papazoglou and D. Georgakopoulos, "Service-Oriented Computing," *Comm. of the ACM*, vol. 46, no. 10, pp. 24–28, Oct. 2003.
- [5] <http://pipes.yahoo.com/pipes/>
- [6] Web 3.0 Technology, *Java Jazzup Magazine*, Sep. 1, 2007
- [7] http://www.askesis.dz/index.php?option=com_content&task=view&id=23&Itemid=53, Service Oriented Architecture, ASKSIS, 2007
- [8] <http://www.orchestranetworks.com/fr/soa/concepts.cfm>, Concepts utilisés dans la démarche SOA, *orchestranetworks*, 2007
- [9] Traudt, Erin; Amy Konary "2005 Software as a Service Taxonomy and Research Guide", June 2005
- [10] Gianpaolo Carraro, Fred Chong "Software as a Service (SaaS): An Enterprise Perspective" Microsoft Corporation, 2006
- [11] Wladimiro Bedin, Moin Moinuddin "An Overview of Software as a Service in Retail", May 2007
- [12] 3GPP TS 29.198 : Open Service Access (OSA) Application Programming Interface (API)
- [13] ETSI, "OSA API, Part 3: Framework (Parlay 4)", ETSI Standard, ETSI ES 202 915-3, 2005.
- [14] OMG, "Telecommunications Service Access and Subscription Specification", version 1.0, formal/02-12-01, 2002.
- [15] M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, R. Metz: OASIS - Reference Model for Service Oriented Architecture 1.0, <http://www.oasis-open.org/>, 2006
- [16] S. Schmid, F. Hartung, M. Kampmann, S. Herbon, J. Rey, "SMART: Intelligent Multimedia Routing and Adaptation based on Service Specific Overlay Networks", in *IST Mobile and Wireless Communications Summit*, 2005
- [17] Z. Li, P. Mohapatra, "QRON: QoS-Aware Routing in Overlay Networks", *IEEE JSAC*, vol. 22, no. 1, pp. 29-40, January 2004
- [18] D. Liben-Nowell, H. Balakrishnan, and D. R. Karger, "Analysis of the evolution of peer-to-peer systems," *Proceedings of 21st ACM Symp. Principles of Distributed Computing (PODC)*, pp. 233–242, July 2002.
- [19] B. Raman, R.H. Katz, "An Architecture for Highly Available Wide-Area Service Composition", *Computer Communication Journal*, vol. 26, no. 15, pp. 1727-1740, Sept. 2003
- [20] X. Gu, K. Nahrstedt, "On Composing Stream Applications in Peer-to-Peer Environments", *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, pp. 284-837, 2006

- [21] X. Gu, K. Nahrstedt, Bin Yu “SpiderNet: An Integrated Peer-to-Peer Service Composition Framework”, EEE HPDC-13, June 2004.
- [22] M.Brunner, “Towards autonomic network management”, Tutorial, IEEE/IFIP Latin Network Operations and Management Symposium, 2005
- [23] S.R.White, J.E.Hanson, “An Architectural Approach to Autonomic Computing”, IEEE International Conference on Autonomic Computing (ICAC), 2004.
- [24] R.Boutaba, J.Xiao, “Network management: state of the art”, Proceedings of IFIP World Computer Congress (WCC'02), TC6 Stream on Communication Systems: The State of the Art, 2002
- [25] J.P.M.Flatin, S.Znaty, J.P.Hubaux, “A Survey of Distributed Enterprise Network and Systems Management Paradigms”, Journal of Network and Systems Management, 1999
- [26] M.Salehie, L.Tahvildari, “Autonomic computing, emerging trends and open problems”, DEAS, 2005
- [27] J.O.Kephart, “Research challenges of autonomic computing”, ICSE, 2005
- [28] J.O.Kephart, D.M.Chess, “The vision of autonomic computing”, IEEE Computer Society, 2003
- [29] N.Agoulmine, “le nouveau concept de réseau autonomic et self-*”, Tutorial, Gestion de Réseau et de Service, GRES, Luchon, 2005.
- [30] J.Strassner, “Autonomic Networking – theory and practice”, Tutorial, IEEE/IFIP Network Operations and Management Symposium, 2004
- [31] A.Fumagalli, I.Cerutti, “Optimal Design of Survivable Mesh Networks Based on Line Switched WDM Self-Healing Rings”, IEEE/ACM Transactions on Networking, Vol.11, pp. 501-512, 2003.
- [32] G.Bieber, J.Carpenter, Openwings: « A Service-Oriented Component Architecture for Self-Forming, Self-Healing», Network-Centric Systems, Openwings document, 2003
- [33] Abdullah Gani and G. Manson, “Towards a Self-Healing Network in Controlling Access to Network Applications”, InSITE, July, 2003
- [34] Gordon S. Blair et al. “Reflection, Self-Awareness and Self-Healing in OpenORB”, WOSS '02, Nov 18-19, 2002, Charleston, SC, USA
- [35] J.J.S.Tobella, M.Stiemerling, “Towards Self-Configuration of IPv6 Networks”, IEEE/IFIP Network Operations and Management Symposium, 2004
- [36] Y.Yemini, A.V.Konstantinou, “NESTOR: an architecture for network self-management and organization”, IEEE Journal on Selected Areas in Communications, Volume: 18, pp. 758 – 766, 2000
- [37] R.Boutabaa, S.Omari, « SELFCON: an architecture for self-configuration of networks, Journal of Communications and Networks», Vol.3, No.4, 2001.
- [38] Simon Schuetz et al. “Self-Configuration for Composable Networks”, 2005
- [39] S.Aiber, F.Gilat, “Autonomic Self-Optimization According to Business Objectives”, International Conference on Autonomic Computing, pp.206 – 213, 2004

-
- [40] D.A.Menasc, D.Barbara, «Preserving QoS of Ecommerce Sites Through Self-Tuning: A Performance Model Approach», Proceedings of the 3rd ACM conference on Electronic Commerce, pp. 224 – 234, 2001.
- [41] http://liuppa.univ-pau.fr/spip/article.php3?id_article=4
- [42] Hajer Derbel, Nazim Agoulmine, Mikaël Salaün, «Modèle d'optimisation basé sur l'utilité des services», GRES, 5-9 Novembre 2007
- [43] H.Luo, P.Zerfos, "Self-Securing Ad Hoc Wireless Networks", International Symposium on Computers and Communications, pp. 567-574, 2002.
- [44] The Grid Report, http://www.sgi.com/industries/sciences/grid_report.pdf, 2002. Bloor research document
- [45] <http://www.research.ibm.com/autonomic>
- [46] <http://netresearch.ics.uci.edu/bionet/architecture.html>
- [47] X.Dong, S.Hariri, L.Xue, H.Chen, M.Zhang, S.Pavuluri, S.Rao, "AUTONOMIA: An Autonomic Computing environment", IEEE International Performance, Computing, and Communications Conference, 2003
- [48] R.Brachman,<http://www.darpa.mil/DARPATech2004/pdf/scripts/RammingScript.pdf> "Gognitive Networks",
- [49] S.R.White, J.E.Hanson, "An Architectural Approach to Autonomic Computing", IEEE International Conference on Autonomic Computing (ICAC), 2004.
- [50] Elyes Lehtihet et al. Initial Approach Toward Self-configuration and Self-optimization in IP Networks, MMNS 2005, LNCS 3754, pp. 371–382, 2005
- [51] Maïssa MBAYE, Francine Krief « Apport de l'apprentissage dans les réseaux autonomiques», GRES, 5-9 Novembre 2007

Chapitre IV

Du Background aux propositions

Cette thèse est basée sur plusieurs concepts et méthodes qui constituent notre background (§IV.1). Après en avoir rappelé les principaux résultats, nous faisons la première contribution de cette thèse en proposant un modèle organisationnel (§IV.2) intégrant les mobilités du notre contexte NGN/NGS.

IV.1 *Background*

Le point focal de notre background est la modélisation [1][2] du monde réel de télécommunication, à partir de laquelle un certain nombre de propositions [2] ont été faites.

Pour modéliser, nous avons recours à l'abstraction. L'encyclopédie universels 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). Abstraction doit appliquée comme une succession d'opérations de l'esprit qui, à partir d'une donnée, extrait une signification utile et construit un (ou plusieurs) concept manipulable intellectuellement. Notre modélisation se base donc sur l'abstraction du monde réel des télécommunications.

Les éléments considérés sont les ressources « équipement, réseau, service, utilisateur ». La première opération est la sélection. On sélectionne la composante la plus spécifique du contexte des télécommunications : le **flux**. On délimite ainsi, le champ d'investigation aux échanges qui mettent en relation les composants interagissant. Cette simplification met en exergue les **nœuds** de même nature coopérants. En généralisant à l'ensemble des ressources, on obtient des mises en relation **par service**. La dernière opération est la représentation, la schématisation de la réalité pour laquelle on propose le modèle $\langle N, L, R \rangle$ [3][4][5] (§IV.1.1) avec les objets abstraits « nœuds, liens, réseaux ». La

juxtaposition et la superposition de « service attendu » nous conduit à identifier des niveaux de visibilité [3][4] [5] (§IV.1.2). Quatre niveaux caractérisent ce monde des télécommunications dont nous voulons réduire le champ de complexité. Le modèle informationnel [6][7] (§IV.1.3) permet de représenter le monde réel. Quant aux aspects comportementaux, c'est-à-dire les aspects non fonctionnels, ensemble de caractéristiques définissant la Qualité attendue du Service (QoS), il est nécessaire de suivre la même démarche, pour la modéliser [8] [9](§IV.1.4). Cette modélisation introduit des concepts puissants et un vocabulaire permettant d'éviter les ambiguïtés de toutes sortes. L'image de ces aspects structuraux et comportementaux constitue le modèle informationnel. Des profils génériques ont été proposés[10] [11](§IV.1.5). Pour conclure ce background, nous présentons une vision architecturale (§IV.1.6) synthétique de l'ensemble du monde réel considéré, à savoir, un monde réel dans un contexte NGN.

IV.1.1 Le modèle <N, L, R>

Nous avons identifié le flux comme le principal élément qui caractérise le monde des télécommunications. Un flux est le résultat des échanges d'unités d'information entre des entités qui s'interagissent et collaborent ensemble pour arriver à fournir ou satisfaire une fonctionnalité donnée. Nous nommons **lien** l'abstraction du flux et des échanges qui s'opèrent entre les entités collaboratrices et **nœud** l'abstraction d'une entité collaboratrice.

Plus précisément, les nœuds sont les éléments du monde réel ayant les capacités de traitement. Un nœud peut représenter un commutateur, un ordinateur, un processus, une machine protocolaire, un composant de service, etc.; alors que les liens sont les éléments du monde réel ayant les capacités de mise en relation. Un lien assure le transfert d'informations entre les nœuds, il peut représenter un câble (une ligne physique), un faisceau hertzien, une liaison LLC (Logical Link Control), un circuit virtuel, une association applicative, etc. Le troisième élément abstrait est le **réseau** qui représente 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.

IV.1.2 Les niveaux de visibilité

Pour structurer les services nous introduisons le concept de niveau de visibilité. Les niveaux d'abstraction permettent de différencier tous les services rendus. Or, on peut désirer limiter le nombre de niveaux pour ne retenir que les plus pertinents dans un contexte donné. Les besoins du contexte filtrent et identifient les niveaux de visibilité désirés.

Une première analyse du monde Télécom pour les besoins de gestion, a conduit à proposer quatre grands ensembles « Équipement, Réseau, Service, Utilisateur » (Figure IV.1). Puis, en fonction de la complexité des éléments et de la finesse de l'analyse voulue, d'autres niveaux peuvent être visibles. Les quatre niveaux sont :

- Le niveau de visibilité « Equipement » : représente le réseau de télécommunication du point de vue des équipements physiques. Le niveau « équipement » est composé d'éléments matériels (les nœuds) coopérant à travers des câbles, des faisceaux, des circuits ou des faisceaux hertziens (les liens). Les nœuds offrent des capacités de traitement et de stockage et les liens

offrent des capacités de transmission et des supports associés. Il rend donc visible le côté « hardware » des entités du réseau. Le service rendu correspond aux capacités individuelles de chaque élément physique.

- Le niveau de visibilité « Réseau » : représente le réseau de télécommunication du point de vue logique. Il regroupe les services des couches 1, 2, et 3 du modèle OSI. Les entités de ce niveau sont donc les machines protocolaires nécessaires à la réalisation de la fonction de transfert. Il rend un *service d'acheminement* à l'aide d'algorithmes distribués (nœuds) et de circuits virtuels (liens). Chaque ensemble de même nature constituera un sous réseau.
- Le niveau de visibilité « Service » : représente et rend visible les services déployés sur le réseau réel des télécommunications. Les nœuds sont les composants élémentaires, et les interactions sont les liens, dont les regroupements constituent les applications. Les entités de ce niveau rendent un service à valeur ajoutée, c'est-à-dire un service applicatif au delà du service de transport basic.
- Le niveau de visibilité « User » : représente les usagers, les clients, les fournisseurs et les organisations qui utilisent le réseau. Les entités de ce niveau rendent un service de type commercial.

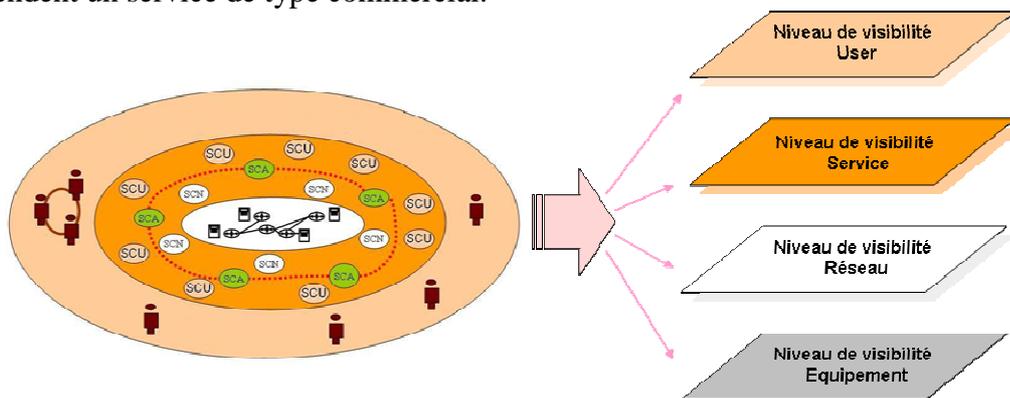


Figure IV.1: Les niveaux de visibilité

Chaque niveau se représente à travers un réseau de nœuds et de liens de même nature ($\langle N, L, R(V) \rangle$). Leur description constitue le modèle informationnel.

IV.1.3 Le modèle informationnel

Pour avoir une représentation efficace du monde réel qui est hétérogène à plus d'un titre, nous devons avoir une structure informationnelle homogène contenant les informations pertinentes et synthétiques afin de prendre les bonnes décisions au bon endroit et au bon moment. Ce qui implique d'avoir aussi bien les informations de description de toutes les ressources, mais également la connaissance des aspects comportementaux, c'est-à-dire, tout ce qui est relatif à la QoS. Le modèle informationnel (Figure IV.2) que nous définissons est générique et abstrait. Il permet la description de n'importe quelle entité du monde réel, c'est-à-dire un nœud, un lien ou un réseau, sur un niveau de visibilité V donné, par le concept abstrait $NetworkElement(V)$ ($NE(V)$). Pour chaque élément de réseau $NE(V)$, le modèle précise son architecture, ses services offerts.

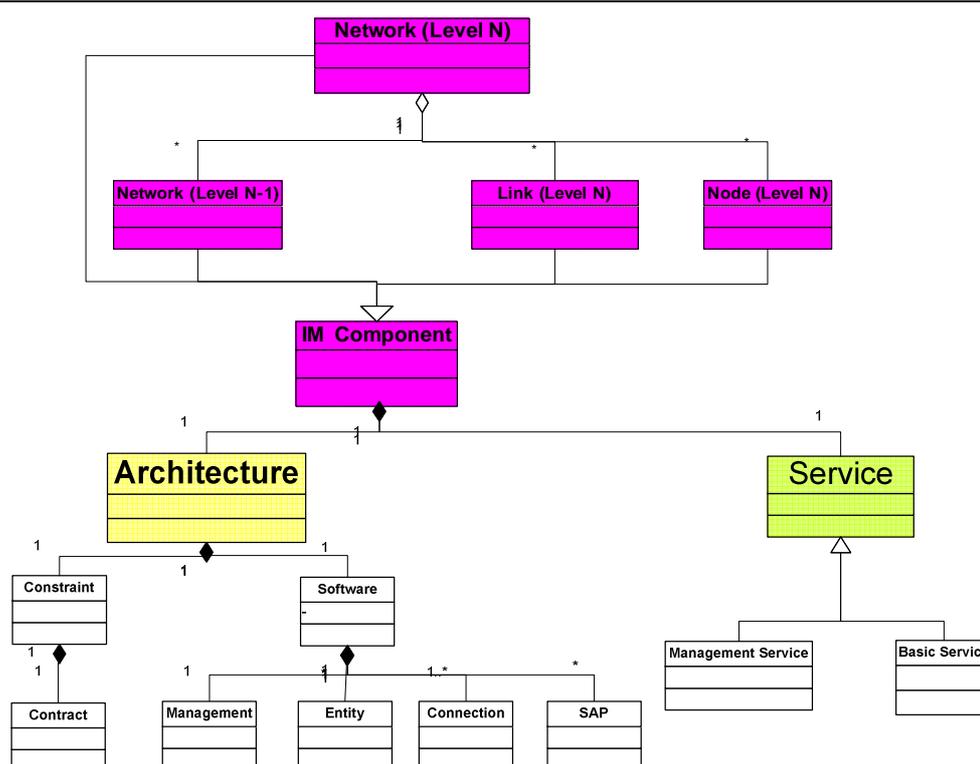


Figure IV.2: Le modèle informationnel NLR (V) du monde réel des télécoms

Selon le modèle, un élément de réseau de visibilité V se décrit comme une composition d'autres éléments de réseau, de même visibilité V et repose sur les éléments de réseau de visibilité inférieure V-1. Cette description récursive de l'élément de réseau est due à la nature même des relations qui existent entre les éléments du monde réel. Par exemple, un réseau se compose de sous-réseaux qui eux-mêmes se décomposent en d'autres éléments de réseau. Pour réaliser la fonction de transfert 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 (de façon récursive) sur l'ensemble des niveaux de visibilité.

Un élément du réseau est décrit par son architecture et ses services offerts via des interfaces. Un élément de réseau 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 ou les contrats du NE (V). Un élément logiciel se compose :

- d'une entité (classe entité) qui représente la fonction coeur du NE(V),
- 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 le NE(V) et les autres NE(V),
- et des points d'accès (classe SAP) par lesquels les services du NE(V) peuvent être fournis ou demandés.

Les interfaces permettent à l'élément de réseau de répondre à des demandes de services ou émettre des messages aux entités du monde extérieur. Selon la nature

des opérations invoquées et du service rendu, l'interface peut être une interface de gestion ou de base. Chacune permet à l'élément géré d'avoir le rôle de client ou du serveur.

Ce modèle informationnel dérivé directement de la modélisation NLR(V) permet la traçabilité des flux et le cross layer. Un même modèle, les mêmes concepts quelques soit le niveau de visibilité permettent d'unifier la description du monde réel.

Une même démarche va être utilisé pour le modèle de QoS et pour avoir une vision complète du comportement global, agrégé, de bout en bout.

IV.1.4 Modèle QoS

L'autre élément structurant que nous utilisons est le modèle de QoS qui repose sur des critères (a), des valeurs mesurables (b) et un contrat (c).

(a) Quatre critères sont définis: Disponibilité, Fiabilité, Délai et Capacité.

Lorsqu'un utilisateur demande un service, il désire un support le plus transparent possible, c'est-à-dire, avec le minimum d'impact sur la fonction de transfert entre les extrémités. on identifie quatre types de transparences : la *transparence temporelle* afin de transférer les informations à chaque fois que l'utilisateur en produit et aussi longtemps que dure leur génération (**Disponibilité**) ; la transparence à la distance afin d'opérer n'importe quel mise en relation (**Délai**) ; la transparence à l'espace afin de transférer de bout en bout le volume d'information instantanément généré (**Capacité**) ; sans oublier la transparence sémantique afin que le transfert se fasse en totalité et sans altération de son contenu (**Fiabilité**). Les définitions sont donc les suivantes :

- 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.

Ces quatre critères sont nécessaires et suffisants pour la gestion des entités réseaux du fait de leur transversalité aux aires fonctionnelles de gestion [FCAPS (Fault, Configuration, Accounting, Performance, Security)]. En effet, la sécurité mise à part, les fautes impacteront la fiabilité ; la configuration représente 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, qui sont également les critères de performance.

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

(b) Ces critères s'évaluent à travers trois types de valeur : valeur de conception, valeur courante et valeur seuil.

– Les valeurs de conception déterminées au moment de la conception des services, traduisent leurs possibilités maximales. Elles interviennent lors de la planification et dimensionnement des services.

– Les valeurs courantes indiquent les comportements courants des différentes entités (nœud, lien et réseau). Ces valeurs sont à surveiller durant l'exploitation pour avoir l'image du comportement des services en temps réel.

– 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.

(c) Le contrat de chaque composant de service est le résultat de la négociation entre la « QoS demandée » et la « QoS offerte ».

IV.1.5 Les profils

Basés sur les modélisations que nous venons de présenter (« Nœud, lien, réseau » et modèle de QoS), nous avons défini les profils suivants :

- **Profil de ressource** pour représenter n'importe quelle ressource,
- **Profil d'usage de ressource** pour tenir compte du contexte d'utilisation de la ressource,
- **Profil d'utilisateur** pour intégrer la personnalisation,
- **Profil actif** pour consigner en temps réel les états des ressources.

Profil de ressource est le profil de base pour représenter toutes les ressources. Il est indépendant de l'environnement d'exécution et est généré par le fournisseur de la ressource. Le comportement des ressources (services, réseaux et terminaux) est décrit à travers leurs possibilités de QoS dénommées dans ce profil *QoS Capabilities*. Les valeurs des quatre critères de QoS sont celles de la conception, elles représentent le maximum qu'une ressource peut fournir.

Par exemple, dans le niveau visibilité de service, nous obtenons l'instance suivante (Figure IV.3) :

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- ServiceProfile -->
- <ServiceProfile>
  <version>1.0.0.0</version>
  - <Identify>
    <ServiceName>VoD-SFR</ServiceName>
    <ServiceType>streaming</ServiceType>
  </Identify>
  - <QoSCapabilities>
    <Availability>0.998888</Availability>
    <Reliability>0.998888</Reliability>
    <Delay>600ms</Delay>
    ① - <Capacity>
      <UserMax>300</UserMax>
    </Capacity>
  </QoSCapabilities>
</ServiceProfile>

```

Figure IV.3: Le profil de ressource

Profil d'usage de ressource est le profil permettant d'intégrer la ressource dans un environnement donné. Il contient les informations permettant de décrire l'usage associé à la ressource et ses conditions pour une exécution fiable, également traduites par les critères de QoS. C'est ainsi que nous dénommons « QoS offerte » la QoS fournie par la ressource et « QoS demandée » celle que doit fournir les composants de l'environnement support à cette ressource. La « QoS demandée » représente donc les conditions demandées pour l'intégration, c'est la plupart du temps celle de la couche inférieure. Par exemple, si le composant est un composant de service, la « QoS demandée » est celle de la couche réseau, qui elle même dépendra de la « QoS offerte » par la couche équipement.

Le profil d'usage de ressource (Figure IV.4) est donc lié à l'environnement d'exécution d'une ressource. Il doit inclure toute l'information nécessaire pour son intégration, c'est pourquoi nous proposons le profil suivant :

- Identifier : ce champ donne le type de ressource et permet d'identifier chaque ressource sans restriction d'adressage en employant un identifiant unique.
- Offered QoS : ce champ indique la QoS que propose de fournir la ressource.
- Demanded QoS : ce champ détermine la QoS nécessaire afin de permettre à la ressource décrite de fonctionner correctement et d'assurer la QoS promise.
- Constraints : ce champ contient les conditions particulières intégration. Elles se déclinent par les niveaux de visibilité conformément à notre modèle « N, L, R ». C'est-à-dire, pour une ressource donnée, on indique les restrictions imposées par chacun des niveaux (User, Service, Network, Terminal).

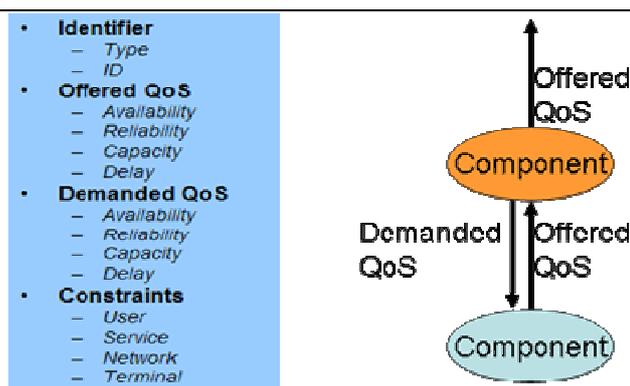


Figure IV.4 : Profil de l'usage de ressource

Exemples :

Profil d'usage de Terminal: le terminal est la ressource de plus bas niveau de notre architecture et la première ressource à laquelle accède l'utilisateur. Le champ « QoS demandée » ne sera pas renseigné. Par contre, les champs contraintes décriront toutes les configurations du terminal considéré. On trouvera la liste des services, des logiciels réseaux (protocoles), des logiciels d'entrée/sortie et des composants matériels (cartes et ports).

Profil d'usage de Réseau: il peut s'agir soit de celui du réseau cœur, soit de celui du réseau d'accès. Dans le cas d'un seul opérateur, il contient la plupart du temps l'information décrivant les réseaux d'accès (ANs). La « QoS offerte » est caractéristique de performance du réseau d'accès considéré, alors que la « QoS demandée » est caractéristique de la performance de l'équipement terminal. Les contraintes renseignées sont celles relatives aux utilisateurs autorisés, aux services exécutables et aux terminaux compatibles.

Profil d'usage de Service (Figure IV.5): il concerne chaque composant de service. La QoS offerte est caractéristique de performance d'un composant de service, alors que la QoS demandée est caractéristique de performance de l'environnement pour que ce composant fonctionne correctement. Les contraintes sont celles relatives aux autorisations pour le plan utilisateurs, celles relatives aux interfaces pour le plan réseau et celle relatives à la compatibilité des terminaux.

Ce profil contient les éléments nécessaires à sa description (caractéristique, identité, etc.) et ceux permettant son bon fonctionnement, à savoir :

- la QoS (valeurs courantes) qu'il offre aux clients
- la QoS (valeurs courantes au moment du déploiement et de la réservation des ressources) qu'il exige du réseau pour pouvoir correctement fonctionner,
- les contraintes éventuelles liées à chaque niveau de visibilité pour avoir une vue exhaustive des restrictions.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- ServiceUsageProfile -->
- <ServiceUsageProfile>
  <version>1.0.0.0</version>
  - <Identify>
    <ServiceName>VoD-SFR</ServiceName>
    <ServiceType>streaming</ServiceType>
  </Identify>
  - <OfferedQoS>
    <Availability>0.998888</Availability>
    <Reliability>0.998888</Reliability>
    <Delay>3600ms</Delay>
  </OfferedQoS>
  - <Capacity>
    <UserMax>300</UserMax>
  </Capacity>
  - <DemandedQoS>
    <Availability>0.999988</Availability>
    <Reliability>0.999988</Reliability>
    <Delay>3000ms</Delay>
    <Capacity>0.5M</Capacity>
  </DemandedQoS>
  - <Constraints>
    - <User>
      <Group>anybody</Group>
      <Language>English</Language>
      <AccessCost>0.01/m</AccessCost>
    </User>
    - <Service>
      <State />
      <Preference />
    </Service>
    - <Network>
      <ProtocoleRelated>IPv4</ProtocoleRelated>
      <ConnectionExtern>TRUE</ConnectionExtern>
    </Network>
    - <Terminal>
      <Processor>1GHz</Processor>
      <Memory>500M</Memory>
      <Device>Screen</Device>
      <Device>Speaker</Device>
      <Application>Mediaplayer</Application>
    </Terminal>
  </Constraints>
</ServiceUsageProfile>
  
```

Figure IV.5 : Profil de l’usage de service

Le profil de l’utilisateur (Figure IV.6) est le profil représentant utilisateur [12]. Il est composé plusieurs sous profils.

Information Générale : Il contient les informations générales (Figure 8) concernant un utilisateur donné, et les informations de toutes ses ressources classées conformément aux niveaux de visibilité (user, service, réseau et terminal).

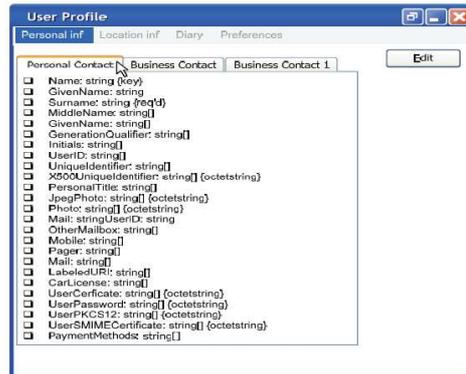


Figure IV.6 : Informations générales du profil de l’utilisateur

Profil Localisation : (Figure IV.7- gauche) il est pour décrire chacune des ressources localisées à un endroit. Les ressources ambiantes sont classées selon niveau de visibilité. Le champ « General Information » permet d’identifier la localisation.

Profil Agenda : (Figure IV.7 - droite) il est pour chaque activité répartie dans le temps nous consignons les ressources ambiantes possibles. Le champ « General Information » permet d’identifier l’activité.

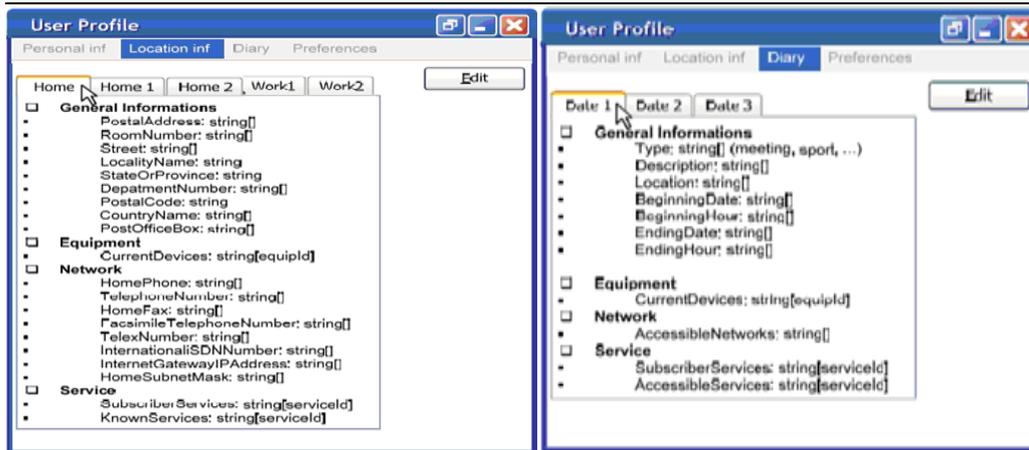


Figure IV.7 : Profil Localisation et Profil Agenda

Profil d'utilisateur doit fournir une interface commune par laquelle les utilisateurs peuvent indiquer leurs préférences concernant des terminaux, des réseaux et des services. Nous reviendrons par la suite sur « la fonction de préférence » que nous appliquerons à ce profil à des fins de gestion dynamique de la session utilisateur.

Profil Actif (Figure IV.8) : il est l'image informationnelle d'un composant architectural représentant une ressource utilisable avec son état en temps réel. Dans notre système, chaque ressource agit en tant que machine à état fini. Nous définissons quatre états distincts pour représenter le cycle de vie entier d'une ressource.

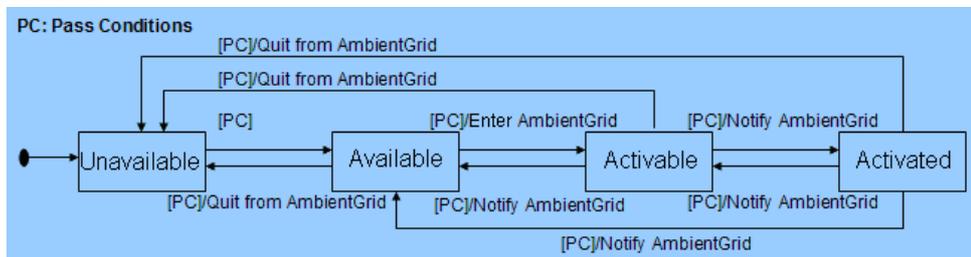


Figure IV.8 : Profil Actif

« Indisponible (Unavailable) » : cet état apparaît quand la ressource est de manière provisoire ou permanente inaccessible.

« disponible (Available) » : cet état apparaît quand la ressource est, ou devient accessible.

« Activable » : cet état apparaît quand la ressource est prête à être sollicitée. Nous avons proposé cet état intermédiaire entre disponible et activé pour tenir compte de certaines conditions d'activation. Par exemple la ressource est accessible qu'après une autorisation (financière ou d'appartenance) une fois l'autorisation (mot de passe) obtenue, la ressource devient Activé.

« activé (Activated) » : la ressource est actuellement en utilisation.

Le profil actif contient les ressources qui se trouvent dans l'état "Activable" ou "activé".

IV.1.6 Conclusion du background

D'abord, dans la dimension architecturale, avec « l'approche réseau », le modèle abstrait du « Nœud Lien Réseau » est appliqué à tous les quatre niveaux de visibilité en prenant compte le contexte de la mobilité dans NGN. Nous obtenons donc un modèle architectural au service de la mobilité (Figure IV.9). Pour couvrir tous les composants nécessaires, dans chaque niveau de visibilité, nous suivons le méta-modèle <Nœud, Lien, Réseau>. Cette architecture générique est composée de quatre couches : la couche de l'utilisateur, du service, du réseau et de l'équipement. Entre chaque deux couches, il y a un rapport entre service demandé et service offert. Par contre, ce monde est en mouvement, donc tous les niveaux sont mobiles indépendamment l'un de l'autre du point de vue A l'instar de la fonction de routage et d'adressage de technologie réseau, nous pouvons faire les cheminements désirés selon différents critères, comme la charge, mais aussi les préférences ou la logique de service.

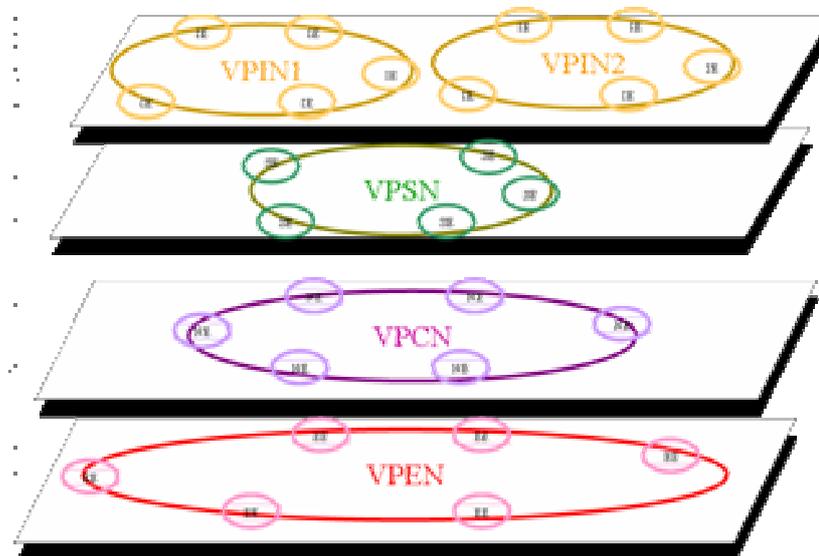


Figure IV.9 : Modèle abstrait au service de la mobilité

- VPEN : (Virtual Private Equipment Network), qui sont les réseaux d'équipements dans la visibilité d'équipement. Les nœuds (EE) sont des machines qui a chacune certain CPU et mémoire. La mise en relation de ces nœuds construit un réseau supports physiques.
- VPCN : (Virtual Private Connectivity Network), qui sont les mises en connexion de composants réseaux (NE) dans la visibilité de réseau. Un NE peut être non seulement un composant d'un réseau d'accès, mais aussi d'un réseau cœur.
- VPSN : (Virtual Private Service Network), qui sont des réseaux overlay de service dans la visibilité de service. Il représente l'application qui devient un réseau de composants de service (SE). C'est la logique de service qui relie les composants pour avoir une application globale demandée par l'utilisateur.
- VPIN : (Virtual Private Information Network) qui représente des autoroutes informationnelles, les nœuds sont des éléments informationnels (IE) dans la visibilité de l'utilisateur. Ainsi grâce à un

routage sémantique nous pouvons avoir des « canaux informationnels » représentant des communautés en mouvement.

Dans la partie informationnelle, grâce au modèle informationnel qui intègre les quatre critères du modèle QoS, nous avons une image complète et globale de chaque composant. Les différents profils des ressources donnent les structures informationnelles pour que le Système Information puisse avoir toutes les informations nécessaires et décisionnelles.

IV.2 Proposition organisationnelle

L'aspect organisationnel répond à la question « qui fait quoi ? ». Comme nous avons indiqué dans le contexte général, le paysage des télécommunications est complètement hétérogène. Depuis la déréglementation nous avons plusieurs opérateurs d'accès, chaque opérateur possédant son cœur de réseaux avec une évolution de convergence fixe mobile. Les besoins sont du domaine des interconnexions. Nous devons projeter la même évolution du côté des services, c'est-à-dire, plusieurs fournisseurs de services, voire des fournisseurs de composants de service avec des solutions d'interfonctionnement. Ce dernier identifie un contexte transorganisationnel pour le niveau service.

Toutes ces offres sont bien évidemment pour les utilisateurs qui veulent des accès faciles et complètement transparents, ils veulent pouvoir faire leur marché (multi-fournisseurs), leur logique de service, changer d'avis (de préférences) en cours de route, avoir plusieurs terminaux, etc. Puisque cette vision « User Centric » a besoin de technologies adaptées, de base de connaissances et de différents niveaux d'abstraction. Tous les niveaux soient transorganisationnels.

Grâce au modèle abstrait au service de la mobilité, nous pouvons compléter l'architecture avec une vision plus en détail en intégrant l'aspect transorganisationnel [13] (Figure IV.10). Avec le souci de préserver un même modèle sur chaque niveau de visibilité de notre architecture générique, nous allons analyser les rôles à couvrir pour assurer les interconnexions (liens de bout en bout) sur chaque niveau.

Au niveau de l'équipement, nous avons tous les équipements supportant les logiciels (blocs architecturaux) de la couche réseau et de la couche service. Nous trouvons aussi les terminaux des utilisateurs. Ce niveau possède donc une visibilité de tous les équipements depuis le terminal jusqu'au plate forme de services.

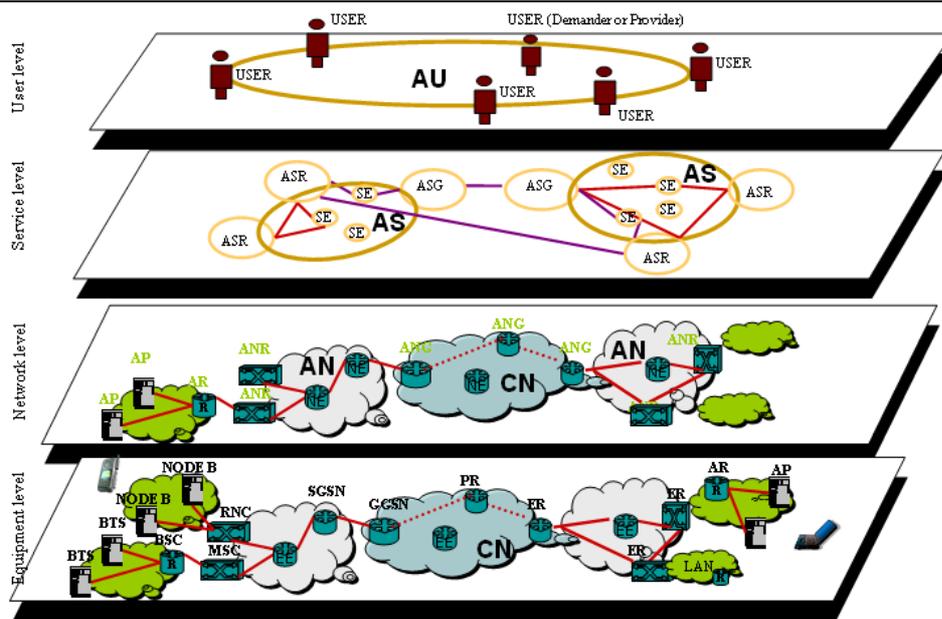


Figure IV.10 : L'organisation et Architecture génériques

Au niveau de réseau, le but est d'avoir la capacité de traverser tous les domaines représentés par des réseaux hétérogènes. Ces réseaux se répartissent en réseaux d'extrémité, réseaux d'accès, réseaux de bordure et le réseau cœur. Nous avons étudié la proposition du RFC 3753. Par conséquent, l'APs (Access Points) et l'ARs (Access Router) sont situés dans la première partie de l'accès afin de simplifier et délimiter la partie d'accès directement devant un terminal. Ensuite nous avons ANR (Access Network Router) et ANG (Access Network Gateway) comme nœuds essentiels du réseau de bordure du CN (Core Network), la connectivité est représentée par les liens dynamiques qui les relient. Un groupe de nœuds ayant une même technologie est traité comme un sous réseau de réseau d'accès. Les composants génériques de ce niveau sont les NE (Network Element). Ce niveau représente la visibilité des réseaux d'accès (AN) et réseau cœur (CN).

A l'instar de ces deux couches (couche équipement et réseau), où nous avons appliqué avec succès notre méta modèle de <Nœud, Lien, Réseau>, nous allons maintenant essayer d'appliquer la même méthodologie de définition de composants architecturaux et méta modèle aux couches supérieures du service et de l'utilisateur.

Dans notre architecture, la couche de service est séparée pour répondre à la composition de services pour notre contexte. Nous pouvons appliquer la technologie Réseau (mutualisation des ressources, routage, adressage, etc.) au niveau service. Donc, la couche de service contient tous les services que nous pouvons offrir. Les services sont organisés en composants de services autonomes SE (Service Element). Nous aurons des services réseaux propres à certain réseau, les services de type applicatif, disponibles offerts par des fournisseurs de service et les services de personnalisation et d'adaptation nécessaires à l'utilisateur.

De plus pour satisfaire la session de service considérant la mobilité de l'utilisateur, nous proposons pour la couche de service de classer les services par fournisseurs de service AS (Access Service). C'est ainsi que la composition peut être trans-organisation. Ce niveau a la visibilité des domaines de fournisseurs de composants de services. Nous supposons que nous avons deux fournisseurs. Nous

proposons deux ASs (Access Service) pour représenter les domaines de service des fournisseurs. Chaque SE (Service Element) est considéré en tant qu'un composant de service et peut offrir un service spécifié. Un ASR (Access Service Router) est le point d'entrée pour les utilisateurs au domaine de fournisseur de service. Un ASG (Access Service Gateway) se situe entre deux AS pour interconnecter des deux domaines de service. Ce mécanisme rend la possibilité d'accéder au domaine du service d'un autre fournisseur par remplacer le ASG actuel avec un ASG disponible dans un autre AS. L'interconnexion de fournisseurs est donc réalisable pour permettre de faire de la trans-organisation.

Au niveau de l'utilisateur, afin de faire la trans-organisation, tous les composants dans cette couche doivent jouer le rôle de Demandeur ou de Fournisseur. Cela nous pose la problématique de l'identification universelle, qui est à compléter dans la suite de notre travail. Supposons chaque utilisateur possède un identité unique qui identifie un seul utilisateur, un AU (Access User) peut être construit par les mettre en relation en tout moment selon la demande d'un utilisateur. Dans cet AU, l'User qui demande les services est le Demandeur, les utilisateurs qui offrent les services et Fournisseur. La construction dynamique de l'AU nous permet de proportionner l'utilisateur nomade et être maintenu toujours au courant avec les environnements hétérogènes. Le niveau de l'utilisateur propose une visibilité abstraite de l'ensemble du système des offreurs et des demandeurs, qui est en fait une visibilité informationnelle, constituée de domaine de fournisseurs d'informations.

Muni de ce modèle organisationnel, nous allons aborder notre problématique principale de la continuité de service dans le prochain chapitre.

IV.3 Références

- [1] Noémie SIMONI et al., “Des réseaux intelligents à la nouvelle génération de services” Collection IC2, Hermes - Lavoisier 2007 - 292 pages - format 15.6 X 23.4 - ISBN: 978-2-7462-1218-3
- [2] http://perso.telecom-paristech.fr/~simoni/AIRS_trial/index.htm
- [3] 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
- [4] “*Generic Information Model*”, ALCATEL Project – 2 – Task 3, NGNSM, 2003
- [5] F.Bennani, “IP et la QoS : vers une maîtrise dynamique de bout en bout”, Thèse de doctorat, ENST/INFRES, 2002
- [6] N.Simoni, S.Znaty, “*Gestion de réseau et de service – similitude des concepts, spécificité des solutions*”, InterEditions, 1997
- [7] Nathalie PERDIGUES, “Une approche pour définir des composants de service administrables : Tarification d'un service de RI” ENST, Paris, 1995
- [8] F. Bennani, Z. Benahmed Daho, N.Simoni, Chunyang.Yin: “An Informational Framework for Autonomic Networking”, GRES 2006
- [9] Noémie Simoni, Bertrand Mathieu, Chunyang Yin, Meng Song, Autogestion de service par la QoS dans un Réseau Overlay, GRES'07, Hammamet Tunisia, Nov. 2007
- [10] Noémie Simoni, Chunyang Yin, Ghislain du Chéné, “An Intelligent user centric middleware in NGN: Infosphere and AmbientGrid”, COMSWARE 08
- [11] S. Rostambeik, N. Simoni, A. Boutignon “Userware: A Framework For Next Generation Personalized Services” appears at the special issue of Computer Communications journal on Emerging Middleware for Next Generation Networks, 2006.
- [12] S. Rostambeik, “Userware : de l'intégration de l'usage à l'interfonctionnement des services”, Thèse de doctorat, ENST/INFRES, 2005
- [13] Chunyang Yin, Noémie Simoni, Ghislain du Chéné, "A personalization and mobility aware service enabler for a service continuity in heterogeneous networks", MSPE'06, 2006, Aachen Allemagne

Chapitre V

Gestion de la continuité de service

Après avoir analysé le contexte, proposé un modèle organisationnel et rapporté les principaux résultats du Background et de l'existant, nous allons consigner dans ce chapitre nos principales propositions afin d'assurer la continuité de service dans un contexte mobile. Elles se subdivisent en trois parties. La première est relative à la gestion de la composition de service qui repose sur une couche overlay de service à part entière, la seconde à la gestion des mobilités et la dernière à la gestion de la session de bout en bout.

Nos contributions sont d'ordre fonctionnel, c'est pourquoi nous commençons par la spécification de la topologie fonctionnelle (§V.1) qui adresse notre problématique de continuité de service intégrée et sans couture. Face aux besoins, un ensemble de « services supports » et leurs interfonctionnements sont proposés. Les premiers concernent la gestion de la composition de service (§V.2). Puis nous suggérons d'utiliser le concept de Communauté d'intérêt pour gérer automatiquement et dynamiquement les composants de service en fonction de différents types de mobilités (§V.3). Pour assurer la continuité de service de bout en bout, nous proposons une session mobile orientée utilisateur avec un Binding dynamique des quatre niveaux architecturaux en tenant compte les préférences de l'utilisateur. Nous avons dénommé « sessionware » (§V.4) les « services supports » qui traitent l'agrégation et la fourniture finale de service demandé. Une synthèse (§V.5) de toutes les propositions est faite à travers la carte des blocs fonctionnels pour avoir une vision globale intégrant la Qualité de Service.

V.1. Topologie Fonctionnelle

Partant des besoins de « User-Centric » et les manques des technologies existantes, nous essayons ici de consigner les « services supports » et leurs interfonctionnements pour soutenir la continuité de service intégrée et sans couture. Ces « services supports » sont des services proposés par la plate-forme de

services pour soutenir les services qui sont en train d'être délivrés et que nous désignerons comme des « services applicatifs » ou des services utilisateurs.

Nous pensons tout d'abord lever l'ambiguïté de NGN et NGS. Notre vision de NGN est comme le montre la figure (Figure I.1) pour couvrir toutes les mobilités.

Un « seamless userware » est situé dans la couche de user pour avoir des informations décisionnelles afin de faciliter la personnalisation.

Les services dans notre contexte, les NGS, sont proposés à travers un réseau overlay pour supporter la mobilité de service. Ils doivent être interopérables, accessibles par plusieurs acteurs, provenant de plusieurs fournisseurs dans un contexte ubiquitaire.

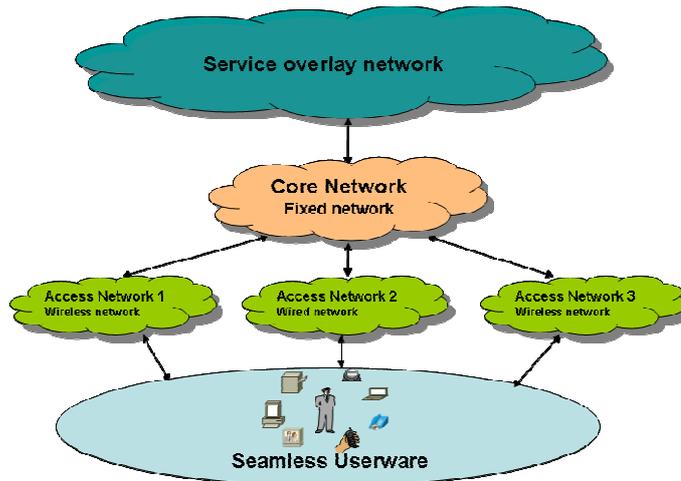


Figure V.1: Vision de NGN

Pour ce faire, nous avons besoins de « services supports » pour construire et gérer un réseau de service du type overlay. C'est pourquoi nous proposons « Overlay Serviceware » (Figure V.2 - 1) pour traiter la composition de service et sa gestion dans une architecture horizontale. Son objectif principal est de la virtualisation des services, la mutualisation de services et le support de décisions flexibles.

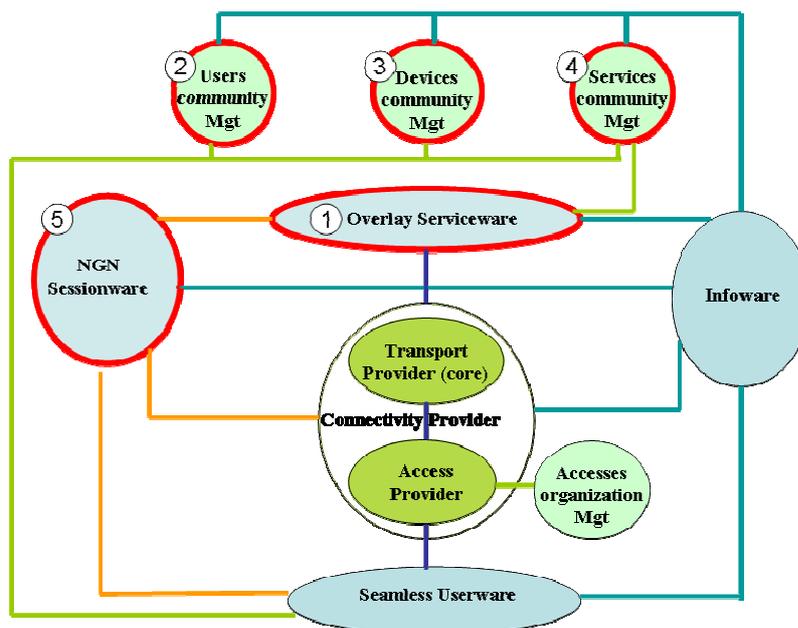


Figure V.2 : Topologie fonctionnelle

La propriété de virtualisation des services demande que les services aux clients soient fournis d'une façon transparente. Il faut donc se doter de mécanismes offrant au client la transparence de la localisation des services qu'il demande, à leur composition, à leur organisation et gestion. Le client aura donc accès aux services qu'il demande sans se soucier de leur localisation réelle ou des mécanismes qui sont mis en œuvre pour le maintien de leur QoS et de leur continuité. La virtualisation de service permet l'utilisation optimale des ressources par une allocation dynamique de service en fonction des besoins de chaque application à un instant donné. Elle offre également une isolation des différents utilisateurs simultanés d'un même service ce qui rend la sécurité de service.

La propriété de mutualisation de service impose non seulement la réutilisabilité de ce service, mais aussi sa partageabilité en tant que ressource en réseau et accessible par réseau. Cela commence par une conception du composant de service à la « superficie » aussi petite que possible pour favoriser leurs interactions et leur communication. Des propriétés de simplicité et d'indépendance entre les éléments de service induiront la viabilité du système. La mutualisation de service est gérée à travers un provisioning dynamique sur n'importe quelle plate-forme de service. La scalabilité sera assurée par le contexte ubiquitaire.

Au-delà de la mutualisation, nous pouvons appliquer tous les autres concepts de la « technologie réseaux » pour supporter des *décisions flexibles*. En particulier, la modélisation permet de représenter les composants de services sur le réseau et en réseau grâce au processus d'enchaînement d'éléments de composants de services (VPSN). Ainsi, nous pouvons optimiser ces liens en ayant recours à la simulation pour évaluer les performances de ce réseau de service. La décision flexible est alors assurée par le routage sémantique et dynamique des composants de service et la base de connaissances décisionnelle. Nous verrons plus loin les auto-gestions nous introduirons à travers la gestion des communautés ().

Fondé sur les quatre niveaux de visibilité et le modèle organisationnel proposé (), nous allons gérer chaque type de mobilité par le niveau architectural approprié. Nous rappelons que les VPxN (x=Équipement, Réseau, Service ou Utilisateur) sont préconisés par notre modélisation pour identifier les composants coopérant pour fournir le service. Donc chaque niveau gèrera dynamiquement son réseau de composants coopérant en fonction des mobilités. Ainsi tout changement intervenant lors des différentes mobilités au cours de la session « User-Centric » sera pris en compte.

Ce qui nous manque sont alors les solutions abouties pour auto-gérer l'échange de composant sans interrompre le service. Est-ce que les technologies existantes sont suffisantes pour supporter les mobilités ? Nous avons constaté dans l'étude du contexte que des solutions comme le handover étaient possibles au niveau L2 et L3, mais que rien n'existe pour la mobilité de services et des utilisateurs. Comme nous le détaillerons plus tard, nous proposons de constituer des communautés virtuelles des composants jouant le même rôle dans un contexte donné. C'est ainsi que nous allons regrouper selon des critères de coopération, les utilisateurs (Users community), les terminaux (Devices community) et les services (Services community). Ce qui permettra la gestion de mobilités à travers les autogestions de ces communautés virtuelles dans chaque niveau de visibilité.

Les utilisateurs d'aujourd'hui peuvent jouer plusieurs rôles, en les mettant selon un intérêt commun dans une organisation virtuelle, nous pourrions suivre leurs

activités et offrir des services qui les satisferont. C'est pourquoi nous proposons « Users community management » (Figure V.2 - 2).

Il en est de même pour les ressources ambiantes que nous devons gérer autour de l'utilisateur. Nous trouvons donc « Devices community management » (Figure V.2- 3) pour les terminaux auxquels l'utilisateur a potentiellement accès. En effet, pour notre vision « User-Centric », nous prenons en compte les préférences de l'utilisateur qui précèdent la mise en œuvre des Handover Horizontal et Vertical. La gestion aura pour but de préparer le changement de terminal demandé par la mobilité de l'utilisateur.

Quand un terminal bouge, l'utilisateur rencontrera différents choix des réseaux d'accès. Nous supposons que les réseaux d'accès sont gérés par une organisation d'accès (« Accesses organisation management »). Le choix de la connectivité de réseau d'accès à réseau coeur se fait par le fournisseur de connectivité. Ces deux types d'organisation sont gérés par les fournisseurs de connexion.

Grâce au « Overlay Serviceware », les composants de service sont devenus les services ubiquitaires et sont gérés par la communauté virtuelle correspondante « Service community Management » (Figure V.2 - 4). La communauté de service organise les composants de service des différents opérateurs selon les besoins de « User-Centric ». L'autogestion de la communauté des services répondra la mobilité de service.

Avec ces gestions des mobilités, nous pouvons maintenant considérer la mise en relation de bout en bout, c'est-à-dire, la « mobilité de session ». La session que nous essayons de construire et de maintenir est la session mobile orientée utilisateur. Dans cette session, le service demandé par l'utilisateur est le résultat de l'agrégation des services offerts par chacun des niveaux VPXN. L'ensemble de ces composants fonctionnels constitue les « services supports » que nous avons dénommé « NGN Sessionware » (Figure V.2 - 5).

Par ailleurs, dans notre topologie fonctionnelle (Figure V.2) nous trouvons « seamless userware » qui est situé dans la couche utilisateur. Ce dernier intègre les « services supports » de l'Userware [2] auquel nous rajoutons les fonctionnalités pour assurer la continuité de service sans couture.

L'« Infoware » rassemble les informations coopératives et décisionnelles de chaque niveau. Il contribue à la continuité de service à travers le profil real time. Ce profil donne une image globale en temps réel et supporte la QoS de bout en bout de façon transparente. Cette composante informationnelle ne fait pas partie du périmètre de cette thèse et est traitée par un autre doctorant.

La topologie fonctionnelle (Figure V.2) positionne l'ensemble de nos propositions pour résoudre la continuité de service dans le contexte NGN. Pour les aspects protocolaires, nous optons pour des communications du type Peer-to-Peer compte tenu de leurs avantages (§ existant). C'est ainsi que les composants de chaque communauté dialogueront en Peer-to-Peer. Il en sera de même pour les composants de VPSN. Quant aux interactions entre les automates fonctionnels, elles répondront aux caractéristiques de l'EDA (Event Driven Architecture), c'est-à-dire, ce sont les événements qui déclencheront dynamiquement les actions de gestion.

V.2. Gestion de la composition de service

Le besoin d'une composition flexible des « services orientés utilisateur » nous amène à proposer une approche générique pour la conception des éléments de service (ES) (§V.2.1). Nous décrivons le fonctionnement interne du composant de service ainsi que ses placements dans l'architecture et ses interactions pour supporter la gestion de VPSN (§V.2.2). Cette composition de service est autonome et dynamique. Ces services supports constituent le « Overlay Serviceware » (§V.2.3).

V.2.1 La composition de service

Dans le contexte NGN, nous avons étudié les différentes conceptions de service (§existant). Nous avons fait les constats suivants. Le Service Web contribue à la simplification et à la réutilisation de service, mais son architecture orientée « client-serveur » nous a empêché de l'adopter dans un environnement ubiquitaire et de diversité. Web 2.0 et Web 3.0 ont pour objectif la partageabilité du contenu et des applications sous la forme d'option de service qui contraint la vraie personnalisation de service. SOA (Service Oriented Architecture) offre un couplage lâche mais la décomposition de service ne permet pas l'accès au composant de service par plusieurs acteurs (user, le contrôle et le management) car il n'est pas mutualisable. Finalement, SaaS (Software as a Service) propose de gérer les composants de service ubiquitaires mais l'architecture a un couplage fort car verticale.

Motivé par le besoin de concevoir des solutions orientées services, par les besoins de mobilité ainsi que l'ubiquité de services, il devient nécessaire d'imaginer un déploiement plus flexible que les offres faites à travers les options de services. Quant aux interactions entre ses composants, elles doivent être du type lâche, afin d'assurer une composition de services répondant aux exigences du processus business (ROI & TTM).

En conclusion, techniquement, un service peut être défini en tant qu'un ES générique quand il est conformé aux quatre caractéristiques importantes : réutilisabilité, mutualisation, interopérabilité, et autonomie (l'autogestion).

Au-delà de ceux qui sont expliqués dans la topologie, *l'interopérabilité* de composants de service interagissent entre eux sur la base de schémas pour les structures et de contrats pour les comportements. Les services veulent traiter l'aspect structurel et l'aspect comportemental séparément et de la façon la plus automatisée possible. Le demandeur de services et le fournisseur de services doivent pouvoir interopérer en employant des structures et des types de données communs.

Quant à *l'autonomie* des composants de services, elle est nécessaire d'avoir des traitements asynchrones. Cet asynchronisme impose aux traitements de réagir automatiquement à des événements. C'est bien l'approche événementielle pour la gestion dont il est question ici.

L'entité réagit donc selon ses propres conditions et à ses propres règles de façon autonome. Le comportement d'une entité autonome ne nécessite pas la connaissance de l'état fonctionnel des autres entités productrices ou consommatrices des événements.

Du côté organisationnel, un ES doit être exploitable. C'est-à-dire, avoir une valeur commerciale pour les fournisseurs (vendeurs du ES) et les consommateurs (acheteurs du ES) [3].

Description fonctionnelle à base d'opération

Lorsque nous modéliserons l'ES comme un élément générique, cela lui conférera une indépendance du contexte et des autres ES. Cela facilitera la composition pour le déploiement et le changement induit par la mobilité, les préférences de l'utilisateur et les aspects de QoS.

Afin de réaliser cet objectif, nous proposons de décrire chaque ES par une chaîne d'opérations pour offrir une certaine fonctionnalité. Ces opérations sont des processus de base inspirés par la modélisation de SIB dans un INCM [3]. L'enchaînement des opérations caractérise un ES particulier et aucune d'entre elles ne peut être remplacée par une autre. Leur généricité et autonomie leur permettent d'être réutilisables dans différents contextes. Tel qui est montré dans la Figure V.3 (a), quand nous avons besoin qu'une opération participe plusieurs fois dans la chaîne fonctionnelle d'un ES, nous pouvons instancier une copie de la même opération.

Modélisée comme une « boîte noire » (Figure V.3 (b)), une opération effectue un processus particulier en prenant les opérandes en entrées et en délivrant le(s) résultat(s) correspondant(s). La modélisation d'une opération induit deux types de flux (donc aussi deux interfaces).

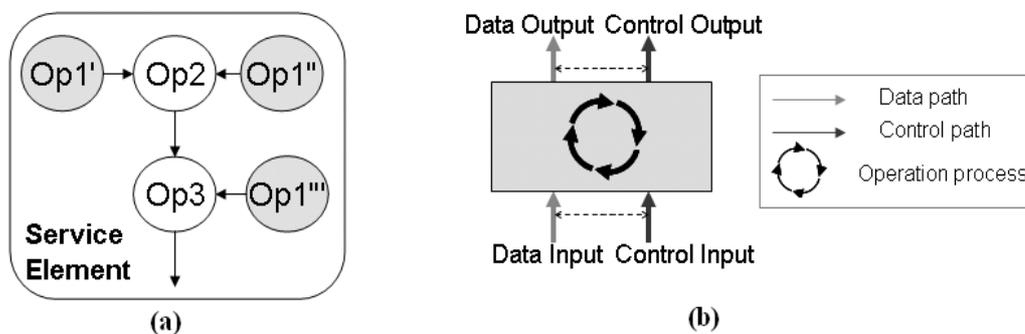


Figure V.3: Modélisation de l'opération : (a) Chaîne des opérations (b) Modèle de l'opération

- Un chemin de données : Il définit l'entrée et la sortie des données d'une opération. Son processus de traitement a besoin des entrées de données afin de fournir des données de sorties.
- Un chemin de contrôle : Pour l'exécution de l'opération appropriée, il faut fournir des informations de contrôle en même temps que le flux de données à l'entrée. Ainsi, quelques informations de contrôle peuvent être aussi montrées à la fin du traitement. Ces informations de contrôle à la sortie, peuvent dans certains cas impacter les données de sortie.

Une opération est un élément autonome, seule, elle n'a pas de contraintes ou de conditions de lien avec d'autres opérations, mais lorsqu'elles sont utilisées dans un schéma particulier d'un élément de service, un lien logique des opérations est fait, dirigé par le contrôle de flux.

Diagramme d'état d'un élément de service

Le diagramme d'états d'un ES présenté dans la Figure V.4 décrit ses états possibles et les transitions parmi ceux-ci en pour chaque VPSN. Mutualisé, l'ES peut être dans plusieurs états en même temps, chacun d'entre eux se réfère à un utilisateur particulier.

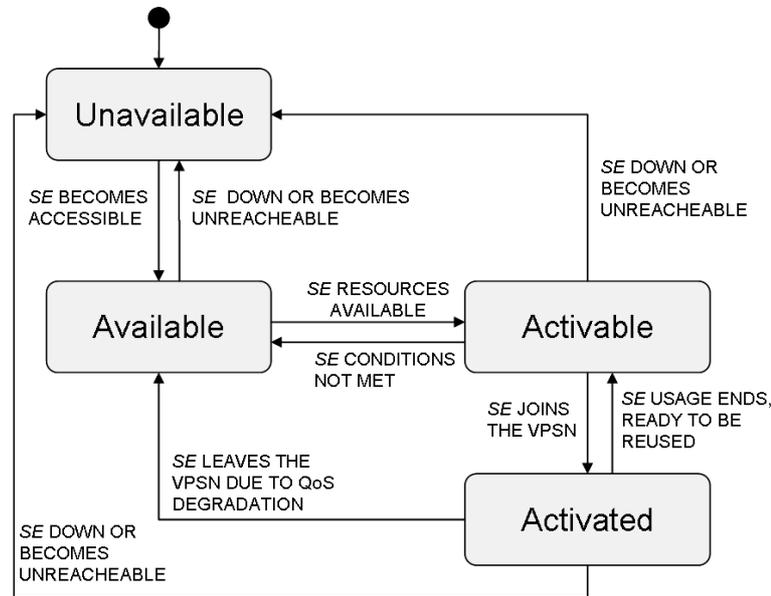


Figure V.4: Diagramme d'état d'un ES

Pour chaque VPSN, chaque élément de service peut être dans l'un de ces quatre états suivants :

- Non disponible: L'ES n'est pas accessible pour l'utilisateur (éteint, bloqué ou cessé de travailler) ou est inaccessible en raison de la mobilité des utilisateurs. Il sera également dans cet état lorsqu'il n'y aura pas assez de ressources pour un nouvel utilisateur.
- Disponible: L'ES peut être consulté mais pas activé. Les ressources de l'élément de service existent et sont disponibles à l'utilisateur ou à tout utilisateur qui le demande ainsi (l'ES ne fait pas encore partie du VPSN).
- Activable: L'ES a été choisi pour faire partie du VPSN et peut être activé à tout moment. Les conditions sont données: l'authentification et/ou l'autorisation sont faites, la QoS est vérifiée et il n'existe pas de contraintes pour une utilisation immédiate.
- Activé: L'ES est en train d'être utilisé. Un ES est considéré être en utilisation tant qu'il fournit le service demandé.

Lorsque un ES est en état *Activable* ou *Activé*, il fait partie du VPSN de l'utilisateur. La sortie du VPSN et le passage à l'état *Disponible* peuvent être déclenché par la mobilité, la dégradation de QoS, ou la fin de la composition de service fonctionnant. La transition d'*Activé* vers *Activable* est faite lorsque l'ES a fini de fournir le service pour lequel il a été demandé, mais reste dans le VPSN pour une utilisation future. Deux autres sous états sont définis pour un ES *Activé* comme montré dans la Figure V.5:

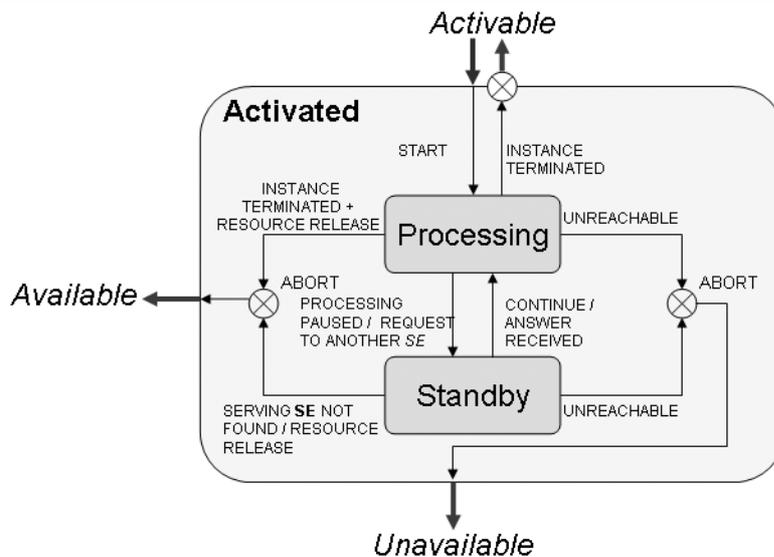


Figure V.5: Etat Activé du ES

- Traitement: L'ES est en opération, des fonctions de traitement sont nécessaires pour offrir ce service.
- En veille/Standby: L'ES se met en pause du traitement afin d'attendre une réponse demandée à un autre ES ou d'accomplir des fonctions de gestion (c'est-à-dire de choisir et de demander un service d'un autre ES).

Lorsque l'ES termine le traitement attendu, il passera de l'état *Activé* à l'état *Activé-Traitement*. Si le service global a pris fin et que l'ES n'est plus nécessaire dans le VPSN, cela changera son état à *Activable* et les ressources seront libérées. Sinon, l'ES passera à l'état *Activable*.

L'ES peut passer directement de n'importe quel sous état de l'état *Activé* à l'état *Non disponible*, lorsqu'il devient soudainement inaccessible.

Un ES *Activé* peut être en veille attendant une réponse d'un ES servant ou dans le cadre d'une procédure de gestion.

Les Compteurs sont activés lorsque un ES est en attente d'une réponse d'un autre ES. Si aucune réponse n'a été reçue à temps, l'ES en déduira une dégradation de la QoS du ES servant et l'ES cherchera un remplaçant. Dans le cas où aucun nouveau ES servant n'est trouvé l'ES ne sera pas en mesure d'offrir le service demandé et devra également être remplacé dans le VPSN. Dans ce cas là ou dans tout autre cas où l'ES doit être changé en raison du processus de gestion tout en étant *Activé*, l'ES change son état à *Disponible* et ses ressources seront libérées.

Interfaces et Composition d'un ES

Pour compléter la modélisation du ES, la séparation des fonctions exercées dans le ES et leurs interfaces avec d'autres ESs est essentielle. La séparation des fonctions exercées par un ES est faite à travers un modèle à trois couches comme est montré dans la Figure V.6.

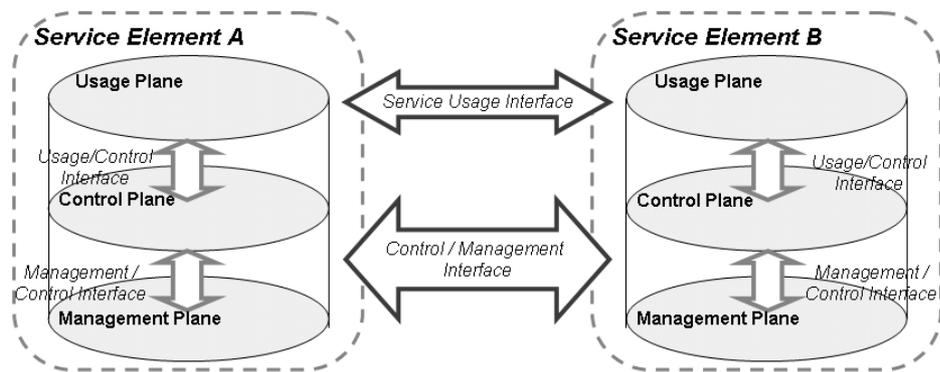


Figure V.6: Vue en couches pour les interfaces ES

- Plan usage: Il comprend les principales fonctions exercées par le ES, afin d'offrir le service ;
- Plan contrôle: Il comprend la signalisation nécessaire pour l'approvisionnement, le routage, synchronisation ; ...
- Plan Gestion: Il comprend l'autogestion et la conception des fonctions (surveillance QoS, QoS mapping-agrégation, gestion communautaire d'un ES, gestion des contrats, le déploiement de service, traçabilité (Accounting)).

Les interfaces internes entre ces couches sont nécessaires pour l'échange d'informations entre l'interaction des fonctions qui appartiennent à différentes couches. Une interface de conception devra également être incluse dans le modèle pour exercer des fonctions de gestion nécessaires dans la phase de déploiement.

Pour être considéré comme autonome, le monde extérieur devrait voir le ES comme une "boîte noire" qui effectue un service, en se concentrant uniquement sur le résultat sans demander comment cet objectif est atteint par le ES. Une définition claire des interfaces du couplage lâche entre chacun des plans d'une ES (Usage, Contrôle/Gestion des interfaces) permettra la communication entre n'importe quel ES, sans regarder le service qu'ils offrent, ainsi que d'un traitement souple de la composition. Pour permettre un contrôle de la gestion dynamique, des interfaces de contrôle et de gestion sont intégrées ensemble dans une interface commune. Une fois la généricité du ES et de leurs interfaces est atteint, nous serons en mesure de déployer une composition de service souple et dynamique.

Conception architecturale

Après avoir une composition d'un ES et leur façon d'interfacer, nous allons proposer comment les ESs coexistent dans une vision architecturale.

Nous essayons d'abord de classifier les types génériques des ESs. Nous nous concentrons dans la couche de réseau overlay de service en fonction du Meta modèle «<Nœud, Lien, Réseau> (§ Background). Nous rappelons que nos composants de service sont mutualisables, réutilisables et autonomiques. Nous obtenons donc trois types des composants de service (§ Introduction) :

- *SCA (Service Component for Application)* qui représente les services pertinents proposé par niveau de service ;

- *SCN (Service Component for Network)* qui représente les services offerts par le niveau réseau, par exemple, le transport ;
- *SCU (Service Component for User)* qui représente les services personnalisés et demandés par l'utilisateur.

Fondée sur ce modèle référentiel, d'une vision « User-Centric », n'importe quelle application demandée par l'utilisateur (Figure V.7 -1) peut être décomposée par les instances de ces trios types de composants (SCU, SCA et SCN) et leurs liens virtuels, qui sont en fait les logiques de services. Chaque service orienté utilisateur impose sur un réseau overlay de service (Figure V.7 -2) qui associé, dans une vue organisationnelle, plusieurs fournisseurs de service.

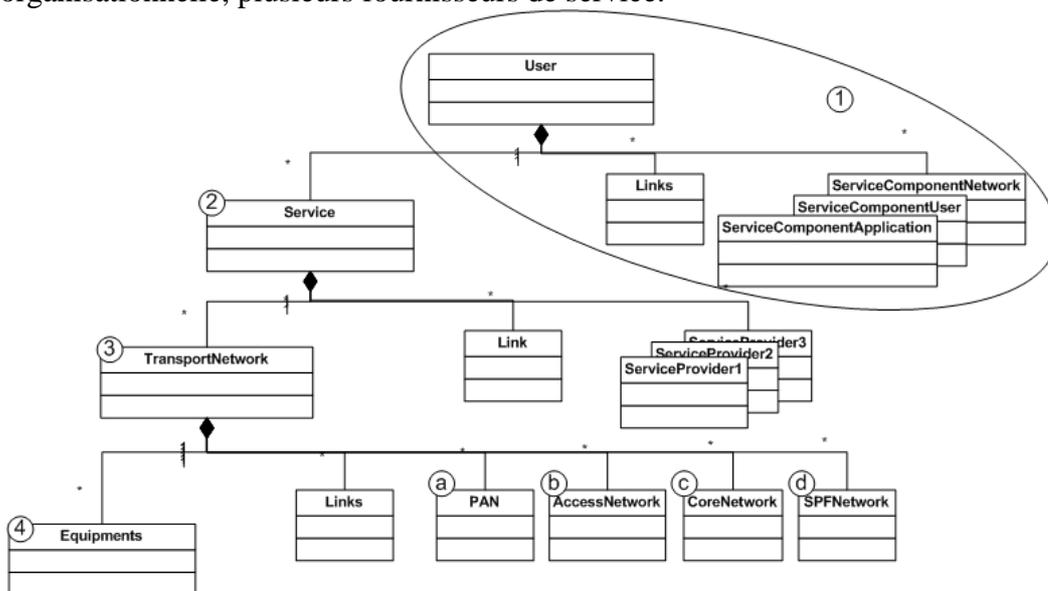


Figure V.7: Modèle architectural

De plus, la caractéristique de mutualisation de composant de service et un environnement de Peer-to-Peer nous permettent d'avoir les mêmes composants dans différentes sessions des utilisateurs. Toutes ces sessions sont mobiles et reposent sur un réseau de transport (Figure V.7 -3) le plus près et le plus désiré du utilisateur. Pour couvrir les mobilités que nous avons examiner (§ Chapitre II), nous proposons un nœud « PAN » (Personal Access Network) (Figure V.7 -a) pour envisager la possibilité du changement de terminal qui est en effet, la mobilité de l'utilisateur. Les différentes accès sont alors présentés par les nœuds « AccessNetwork » (Figure V.7 -b) pour couvrir la mobilité de terminal. Ce dernier peut coopérer avec le nœud « CoreNetwork » (Figure V.7 -c) pour fournir une continuité sur la connectivité réseau, qui est la mobilité de réseau. Finalement, le changement de composant de service, par fois si causé le changement de fournisseur de service est pris en compte par l'ajoute d'un nœud « SPN (Service Provider Network) » (Figure V.7 -d). Tous ces quatre nœuds (Figure V.7 – a, b, c et d) avec leurs liens (les routages réseaux) en temps réel, supportent la connectivité de bout en bout à travers un réseau des équipements (Figure V.7 -4).

V.2.2 Le VPSN

Nous avons déjà une vision omniprésente de la composition de service grâce à notre modèle organisationnel, pensons que les composants de service sont partout

dans chaque niveau de visibilité, une proposition générique est donc possible pour traiter un même type de l'événement.

Nous avons résumé les propositions évoluées de la composition de service, nous rappelons que pour avoir une continuité de service *sans couture*, c'est-à-dire, une omniprésence de composition de service, nous préférons la gestion à travers le réseau overlay de service dans une architecture de P2P.

Tant que le déploiement de services autonomes et auto gérables consomment temps et argent, la mutualisation du ES fournira un moyen d'optimiser les ressources en partageant de manière efficace entre les différents utilisateurs. A travers la mutualisation, chacun des ES des utilisateurs lui sera assigné une tranche du ES, qui comprend seulement les ressources nécessaires à cet utilisateur en particulier et demandé dans son contrat, donc, le ES peut être mutualisés parmi plusieurs VPSN et utilisés dans différents contextes (Figure V.8). Toutes les ressources non utilisées dans le ES seront libres d'être assignés à de nouveaux utilisateurs à la demande. En outre, plusieurs tranches du même ES peut être trouvé dans le même VPSN, chacun d'eux servant à remplir la même fonction, mais avec des cibles différentes, avec ses propres exigences et contrat.

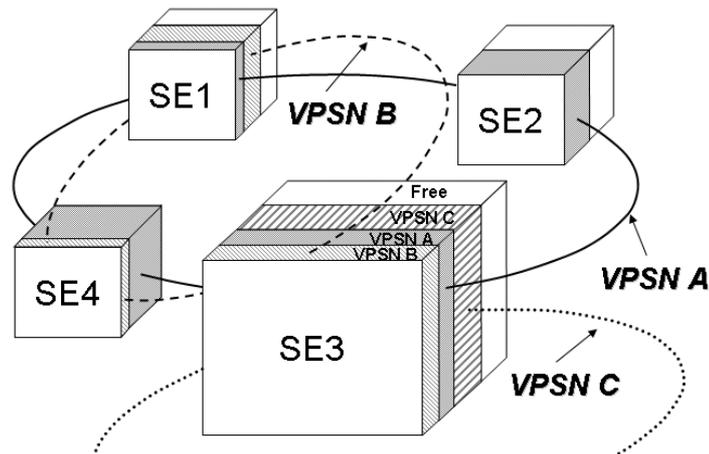


Figure V.8: Mutualisation d'un ES pour différente Composition de Service

Afin de visualiser la flexibilité de la composition de service, nous proposons un scénario (Figure V.9) qui prend cas de deux utilisateurs, chacun d'eux avec sa propre session, donc ils disposent des VPSN indépendants.

L'utilisateur A veut afficher un film offert par son fournisseur de VoD chez des amis. L'utilisateur A ouvre sa session, et accède à l'interface de la vidéo à la demande du fournisseur qui lui permet de choisir le film qu'il souhaite regarder de la base de données. Pour autoriser l'accès à l'utilisateur A, une authentification (authentification ES) sera faite par le fournisseur de vidéo à la demande afin de vérifier son identité. Le service qui permettra le choix du film sera fait par le choix du contenu du ES. Une fois le film choisi, une nouvelle instance de l'authentification du ES sera nécessaire, cette fois, pour authentifier l'utilisateur à sa compte afin de charger le coût du film demandé. L'ES autorisation sera également demandé afin de vérifier si le film choisi n'est pas interdit à cet utilisateur particulier (d'après sa politique de contenu, si une scène d'action s'applique dans le film). Notez que l'authentification et l'autorisation des services sera effectuée d'une manière transparente, sans l'intervention de l'utilisateur A. Une fois le film est choisi et approuvé, l'ES Localisation sera utilisé pour localiser

l'utilisateur A et les terminaux susceptibles à être utilisés pour l'affichage du film. Après avoir analysé toutes les possibilités, les meilleurs équipements sont un vidéo projecteur connecté à l'ordinateur de l'utilisateur A pour afficher l'image et pour le son, il y a un équipement avec des haut-parleurs de bonne qualité avec Hi-fi. L'ES streaming, transcodage, séparation de l'image/et Son, affichage de l'image et affichage du son travailleront ensemble pour fournir le service demandé par l'utilisateur A avec les meilleures performances possibles.

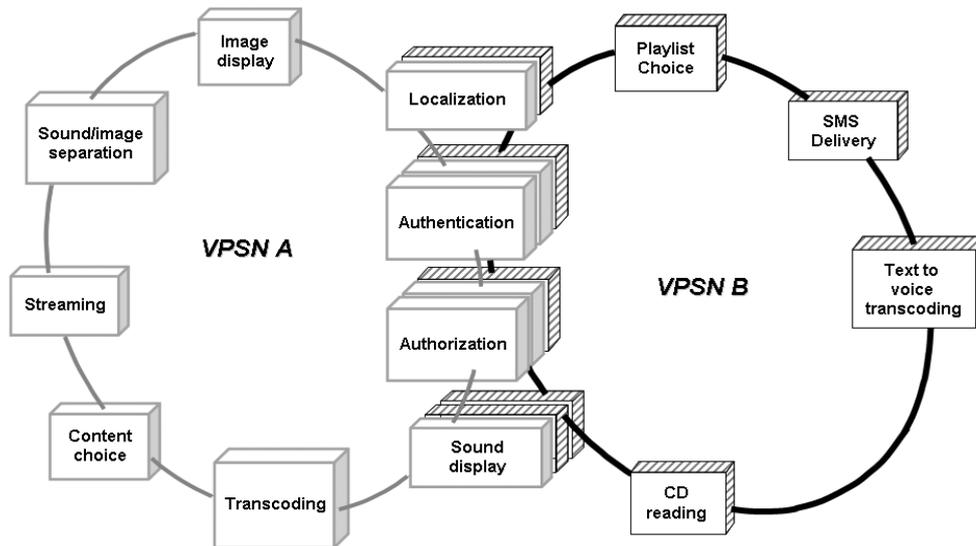


Figure V.9: Scénario de la mutualisation d'un ES et Composition de Service

Notre deuxième cas est situé dans un contexte complètement différent. Un utilisateur B est en vacances et conduit sa voiture pendant quelques heures à la plage la plus proche. Il a ouvert sa session avant de quitter son domicile et choisi parmi une liste la musique qu'il veut écouter au cours de son trajet (ES playliste choix). Selon les préférences de l'utilisateur spécifié dans son contrat, l'utilisateur B veut recevoir tous ses messages SMS comme SMS-vocaux pendant qu'il conduit sa voiture. L'utilisateur B quitte sa maison en écoutant la musique choisie précédemment (ES lecture de CD et affichage du son pour adapter le son au type de musique joué). Alors qu'il est sur la route, il reçoit un SMS. L'ES Localisation sera appelé pour localiser l'utilisateur afin de trouver la meilleure façon de délivrer le SMS. Une fois localisé, l'authentification et autorisation du réseau sont nécessaires. Le service de lecture de musique est en pause tout en offrant la communication vocale-SMS et ES: SMS de livraison, transcodage texte à voix et affichage du son, sont utilisés pour y parvenir. Le service de jouer la musique sera repris après que la livraison des SMS soit terminée.

On peut observer dans ces deux cas, comment le même ES peut être mutualisés parmi les différents utilisateurs et comment le même ES peut être utilisé dans des contextes différents, grâce à la généricité du ES. Cela rendra la composition souple et permettra aux utilisateurs de personnaliser leurs services et d'en créer de nouveaux en fonction de leurs besoins.

Dans la deuxième partie de notre étude de cas, nous présentons la description fonctionnelle pour un ES "une seule authentification". Le processus opérationnel de ce ES peut génériquement être atteint à travers 5 opérations liées ensemble dans une logique définie montrée dans la Figure V.10. En associant deux acteurs, le suppliant dans une partie (qui veut être authentifié) et celui qui contrôle

(authenticator) dans l'autre côté, la structure du ES est accessible par deux entrées et délivre un résultat à la sortie qui met fin à la partie de l'authentification.

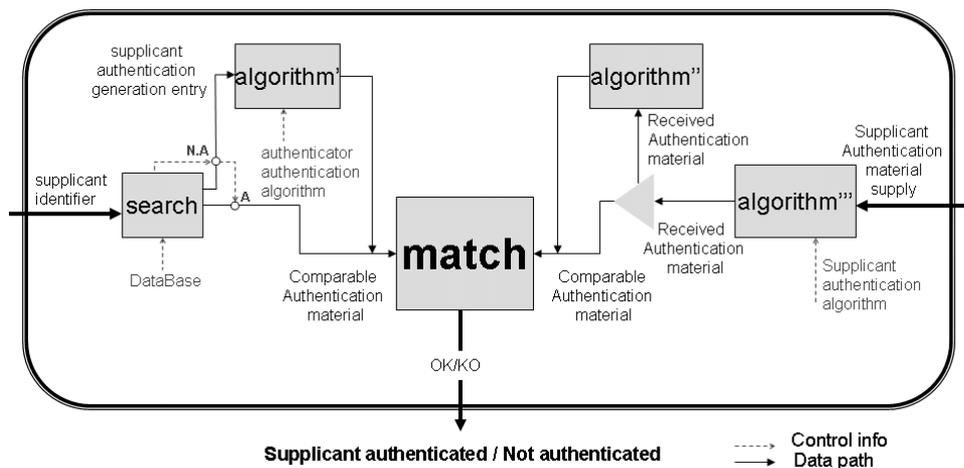


Figure V.10: Description fonctionnelle du ES “une seule authentification”.

Les cinq opérations structurant la description fonctionnelle du ES sont expédiés comme suit: *recherche*, *algorithme*, *l'algorithme''* et *correspondance* sont effectués par le contrôleur (authenticator), le suppliant couvrira seulement une opération (*algorithme'''*). Le suppliant transmet à l'authenticateur ses matériaux d'authentification obtenus après l'exécution d'un algorithme d'authentification défini par le suppliant. Du côté authenticateur, une opération de recherche est réalisée à l'aide d'un identifiant du suppliant (un nom d'utilisateur par exemple) à travers une base de données pour récupérer ce qu'il attend de l'utilisateur de faire l'authentification réussie, c'est-à-dire matériaux d'authentification comparables pris comme référence. Si la recherche n'a pas abouti, le processus est redirigé par le biais de l'opération de *l'algorithme'* pour générer les paramètres nécessaires. Lors on obtient du suppliant les matériaux d'authentification, la poursuite du traitement sont nécessaires à travers l'opération de *l'algorithme''* afin de les rendre prêts à être comparés. Le processus de ES est terminé à travers une opération de comparaison pour vérifier si du côté du suppliant les matériaux d'authentification comparables de référence correspondent à l'authentification du matériel délivré par le contrôleur.

V.2.3 « Overlay Serviceware »

Basé sur les propositions sur la composition de service et la fourniture de service à travers le VPSN, nous pouvons maintenant préciser le bloc fonctionnel de l' « Overlay Serviceware » comme la suite (Figure V.11) :

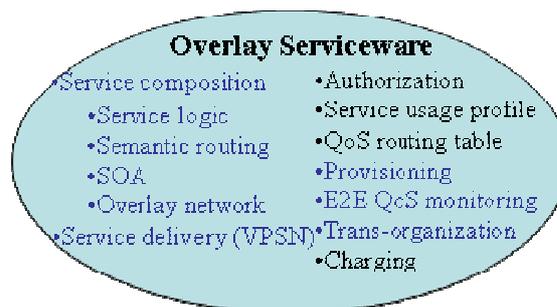


Figure V.11: Overlay Serviceware

Nous expliquons les fonctions importantes :

- Service Composition

Cette fonction doit effectuer toutes les spécifications que nous venons de proposer.

- Service delivery (VPSN)

La livraison de service concerne la gestion d'une session continue basée sur un tableau d'entrée/sortie de l'agrégation de service (QoS routing table). Ce tableau est maintenu par la fonction de continuité dans chacune des communautés. La construction dynamique du VPSN permet une session de service sans couture et sans coupure.

- E2E QoS management

Cette fonction nous permet d'avoir une continuité de QoS de bout en bout.

- Charging

Ces trois fonctions nous permettent d'enchaîner tous les composants de service des différents fournisseurs de service et offrir à l'utilisateur un charging global et une seule facture.

V.3. Gestion des mobilités

En considérant les mobilités qui sont les causes principalement des discontinuités de service, nous pouvons retrouver les solutions traditionnelles de la gestion de mobilité pour le réseau tout IP : *Mobile IP* et *SIP Mobility*. Partant de la première réponse à notre problématique obtenue, pour avoir une gestion de la mobilité à la façon générique, nous nous intéressons à la gestion de mobilité dans un environnement P2P.

Nous rappelons que pour avoir une continuité de service respectant la QoS, tous les dysfonctionnements de la QoS (§V.3.1) sont considérés comme les événements qui lanceront une gestion. La première étape est d'essayer d'identifier les acteurs (§V.3.2) qui invoqueront la coupure de la fourniture de service à cause des mobilités. Nous proposons une gestion autonome et dynamique par les communautés virtuelles pour gérer ces mobilités. Nous démarrons du concept de communauté d'intérêt (§V.3.3) pour donner un règlement d'organisation. L'autogestion de la communauté (§V.3.4) est alors proposée.

V.3.1 Dysfonctionnements de la QoS du composant

Nous avons choisi quatre critères pour exprimer les caractéristiques de service. Comme nous l'avons exprimé, la QoS traduit le comportement du service. Or ce comportement peut se modifier durant les différents déplacements. Nous allons recenser les raisons du dysfonctionnement de la QoS du composant lors des mobilités à travers les critères.

La Disponibilité

Au-delà des indisponibilités dues aux pannes du composant, nous pouvons avoir, lors de la mobilité du terminal, qui entraîne des changements de réseau d'accès, des dépassements de seuil de temps de réponse ou des coupures de connectivité.

- Dépassement de seuil du temps

Par exemple, on prend un composant de service. Pour ce composant, le dépassement de seuil du temps peut être à cause de l'expiration de crédit du côté utilisateur, l'autorisation pour accéder à ce composant n'est plus valable. Ainsi que l'utilisation du composant comme un dépassement d'une mise à jour ou d'un changement des paramètres.

- Coupures de connectivité

Elles se produisent suite à des pannes de matériel ou de logiciel. Une panne du hard indique une faute dans les équipements par exemple les serveurs, les câbles, les terminaux, etc. Celle du soft peut être la congestion du réseau ou des fautes / time out dans les protocoles de la communication.

De plus, si on se trouve dans un contexte 'transorganisationnel' on peut également avoir des coupures lors du changement de fournisseur de service ou d'une remise à jour de la configuration à cause du changement de location.

Délai

Là aussi la mobilité du terminal, entraîne des temps de traversé plus long, qui impactent le temps de réponse demandé initialement par l'utilisateur dans son SLA. Dans notre contexte ubiquitaire, on aura des possibilités de préserver les temps de réponse.

Capacité

Il s'agit ici d'un manque de ressource pour les traitements aussi bien au niveau du terminal, que du réseau ou du composant de service. Les fonctions de provisioning (terminal, réseau, composant de service) n'ont pu aboutir lors de la mobilité de l'utilisateur (changement de terminal) ou lors de la mobilité du terminal (changement de réseau d'accès) ou lors du changement du composant de service.

Fiabilité

Nous trouvons sous ce critère les dysfonctionnements dus à la perte de données lors des différents handovers qui interviennent lors des déplacements.

Le dysfonctionnement de ces quatre critères traduit le fait que la QoS demandée n'est plus garantie par les composants de service lors des déplacements de l'utilisateur. Tous ces dysfonctionnements sont considérés comme des événements qui déclancheront la gestion.

V.3.2 Des acteurs

Après le recensement des dysfonctionnements de la QoS, nous allons identifier les acteurs extérieurs et les blocs fonctionnels correspondants pour traiter les différents types de mobilité.

D'après les exigences de NGN et les besoins de l'utilisateur centric, on trouve donc (Figure V.12):

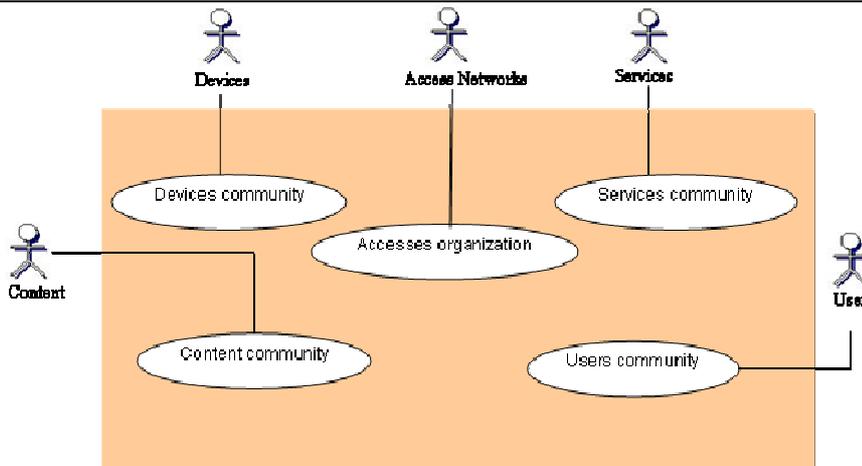


Figure V.12: les cas d'utilisation de la gestion de mobilité

Devices, qui représentent les terminaux autour d'un utilisateur, ces terminaux peuvent être ceux qui sont utilisables et ceux qui sont prêts à être utilisés.

Access Networks, qui représentent les différents types de technologies d'accès, ces réseaux d'accès peuvent être homogènes ou hétérogènes.

Services, qui représentent les services variés supportés par les fournisseurs de services.

Users, qui représentent les utilisateurs avec une identification unique.

Contents, qui représentent les contenus de service comme textes, images, audio, vidéos et animations. Ils ont un rôle important dans le format de service.

V.3.3 Concept de communauté d'intérêt

Pour pallier les dysfonctionnements de la QoS concernant les composants, on propose une organisation fondée sur "les communautés d'intérêt".

Nous disposons d'une définition de Hillery (1955): «la communauté est composée de personnes avec des interactions sociales au sein d'une zone géographique et ayant un ou plusieurs autres liens.»

Une autre définition de la «communauté d'intérêt» (CoI), est donnée par US DoD: "Un groupe de personnes connectées les unes aux autres par un besoin de résoudre des problèmes communs, de développer les compétences et de partager des pratiques communes "

[4] propose une définition pour le concept de "Communauté d'intérêt" dans le contexte de gouvernement local, qui est appliqué à un groupe de personnes dans une location résidentielle et ayant au moins une des trois dimensions suivantes:

1. la dimension Perception : Un sentiment d'appartenance à une région ou à une localité qui peut être clairement définie.
2. la dimension Fonctionnelle : la capacité de coordonner les services demandés avec les exigences de la communauté.
3. la dimension Politique: la capacité à l'élection d'une représentativité pour exprimer l'intérêt de la communauté et concilier les conflits de tous ses membres.

C'est important de mentionner que cette définition inclue deux dimensions de la notion de communauté d'intérêt. Elles sont identifiables en tant que dimension subjective (perception) et objective (fonctionnelle) et de ce fait, chacune se rapporte à différentes mesures et qualités. La dimension politique implique à la fois l'aspect subjectif et objectif.

La CoI définit son endroit géographique et les relations sociales et économiques, sans oublier l'indication de l'identité, du besoin et de la priorité entre les membres de la communauté.

Les mesures fonctionnelles de la communauté d'intérêts devraient également analyser l'étendue de l'administration locale comme un fournisseur de services surtout dans le cadre de la gestion. Il peut aussi être utile d'évaluer l'existence ou la tendance de la coopération entre différentes communautés d'intérêts. La différence des communautés peut aussi se faire selon les activités et les services fournis.

L'évaluation des caractéristiques politiques peut être la direction, la distribution de l'information et le réseau de gestion.

En conclusion, plusieurs approches peuvent exister pour concevoir une CoI :

- Une approche organisationnelle ou structurelle : la CoI a un caractère organique et stable dans le temps. Elle ne dépend pas des missions/objectifs attribuées. Par exemple, la communauté politique que nous venons de mentionner.
- Une approche fonctionnelle : la CoI est constituée pour la durée de vie d'une application. Par exemple, une équipe de construction, un groupe de soldats pour un convoi, celle de position,.
- Une approche informationnelle : la CoI permet d'organiser des constructions d'ensembles pour faciliter la mise en oeuvre du partage de l'information. Ses membres sont chargés de rendre l'information visible, accessible, compréhensible, et de promouvoir la confiance - ces facteurs contribuent à l'interopérabilité nécessaires des données pour avoir l'efficacité du partage de l'information.
- Une approche collaborative : la CoI est formée pour regrouper des acteurs complémentaires. L'intérêt majeur est d'associer les capacités des membres participés et d'obtenir ce qu'il y a de mieux avec les ressources disponibles.
- Une approche coopérative : la CoI regroupe des acteurs soumis à des règles communes de comportement.

Le but des communautés d'intérêts est d'acquérir des accords sémantiques et structuraux sur l'information partagée. Pour que la communauté d'intérêt soit efficace, il faut que leur champ d'application, c'est-à-dire la sphère de leurs accords, soit aussi restreinte que raisonnable.

On s'appuie sur cette notion de communauté d'intérêt pour gérer la composition et la recomposition aussi bien de services que des réseaux, lors des différentes mobilités. Nous précisons que dans notre contexte, l'intérêt commun, la mission, est de fournir la continuité de service quelque soit la mobilité considérée.

Nous considérons d'abord la dimension organisationnelle de la CoI. L'organisation de CoI doit être au sein d'un cadre commun, c'est le cadre qui formalise l'existence commune des composants comme une entité. Il protège également ses membres pour que les données et les services intérieurs ne soient pas proposés à l'extérieur. De plus, en considérant la partie collaborative, ce premier filtrage nous permet de réunir un ensemble d'acteur qui, au lieu de

d'ajouter leur ressources, les multiplient, c'est-à-dire, plusieurs membres peuvent profiter du même service.

Ensuite, la CoI met en commun des acteurs jouant des rôles. Les rôles sont instanciés par des acteurs. Il est souhaitable de proposer les supports permettant le travail collaboratif, c'est-à-dire, le partage de l'objectif soit porte non seulement sur les informations, mais aussi sur les services, de façon transverse, dans le cadre d'une mission, indépendamment de l'organisation hiérarchique. Ainsi, une entité peut avoir plusieurs représentations dans ce plan : appartenance à plusieurs CoI simultanément.

Finalement, la CoI possède les moyens procurant des capacités opérationnelles. Cette dimension permet de proposer ou de souscrire à des ressources d'après les comportements des composants de service. Les ressources ont des capacités qui sont en fait les services offerts.

Donc, les communautés d'un même type d'organisation peuvent être précisées par leur dimensions fonctionnelles et dimensions opérationnelles.

Nous appliquons ces propriétés (cadre- rôle- capacité) pour obtenir la définition formelle de la CoI. Avec l'architecture proposée, chaque mobilité relève d'un niveau de visibilité. Nous proposons de créer à chaque niveau des communautés, chacune anticipant les actions à prendre pour la mobilité dont elle est responsable. Nous trouvons donc les acteurs extérieurs mentionnés :

- **Communauté des utilisateurs (Users community)**

Organisationnellement, cette communauté a d'abord pour but de gérer un groupe d'utilisateurs qui sont accessibles par une identification personnelle unique.

De plus, d'après les rôles joués par chaque utilisateur, ceux qui ont le même intérêt ou bien participent à la même activité peuvent être placés dans une même communauté d'utilisateur. L'objectif pour les mettre dans une même communauté est de partager au maximum des ressources ou/et des services.

A travers la vision opérationnelle, chaque l'utilisateur apporte des comportements différents pour des activités différentes. Un utilisateur peut être un demandeur d'information, mais il peut en être aussi un fournisseur si il possède des services intéressants d'autres utilisateurs. Dans ce cas là, ceux qui ont les mêmes comportements doivent être mis dans la même communauté.

- **Communauté des équipements (Devices community)**

Cette communauté est identifiée pour gérer les équipements en répondant différents besoins.

Pour un utilisateur donné, une seule communauté est donnée pour gérer son groupe des équipements. Cette communauté est donc conçue pour être responsable de la gestion des équipements dans le PAN (Personal Area Network). La propriété des équipements est donc protégée.

Par contre, fonctionnellement, le travail principal de cette communauté est de gérer la mobilité des utilisateurs, chaque type de équipement a ses propres services offerts mais aussi contraintes comme la batterie, le poids, le coût d'utilisation, en considérant les préférences de l'utilisateur sur les équipements, toutes ces informations doivent être bien organisées. Un équipement participe

donc à plusieurs communautés selon ses services possibles et ses contraintes techniques. La mutualisation de ressources est donc garantie.

De plus, si nous considérons la capacité opérationnelle, les comportements de chaque service offert par un équipement doivent être pris en compte. Dès qu'un composant de service est installé pour un déploiement de service d'une session « User-Centric », sa QoS devient également un critère pour la construction de la communauté des équipements.

- **Organisation des accès (Accesses organization)**

Cette communauté est nommée 'organisation' car elle gère tous les réseaux d'accès utilisables ou potentiellement utilisables en considérant un terminal mobile. A cause du basculement du terminal entre différents domaines d'administratifs, cette communauté doit aider la gestion du handover et la gestion de localisation. L'objectif de la gestion du handover est de garder toujours la connexion malgré le mouvement du terminal. L'objectif de la gestion de la localisation est de garder la trace du terminal même sous l'état de veille pour être prêt à fournir les services demandés. La gestion est prise en compte par les fournisseurs de réseau, nous avons considéré que la plupart des solutions actuelles répondaient à cette mobilité du terminal. La mobilité de réseau doit également être supportée par cette organisation.

- **Communauté des services (Services community)**

La mobilité de service doit être supporté grâce à l'auto gestion de cette communauté. La communauté organise les services qui ont la même caractéristique organisationnelle, par exemple, l'endroit où se trouvent les composants de service. De plus, les services dans une même communauté peuvent être fournis par différents fournisseurs.

Principalement, on peut avoir plusieurs communautés de service simultanément en fonction de différentes exigences fonctionnelles. Cette communauté gère tous les composants de service fonctionnellement équivalents.

Pour que l'aspect non fonctionnel soit respecté, les composants de service qui sont QoS équivalents doivent être mis dans une même communauté de service.

- **Communauté du contenu (Content community)**

Cette communauté a pour but de gérer la migration du contenu lors des différents types de mobilité. Par exemple, la fonction de choisir les différents codages pour un même film a besoin du support de la gestion de contenu pour avoir tous les formats de fichiers disponibles pour le transcodeur. Nous proposons de gérer également les contenus à travers la gestion de communauté appropriée car ce dernier est très important dans le cycle de vie d'un service. La gestion d'un système distribuée et le problème de sécurité peuvent être gérés par l'auto gestion de cette communauté. Cette communauté n'est pas dans le périmètre d'étude de cette thèse.

V.3.4 Autogestion de communautés

Pour satisfaire le besoin de *flexibilité* demandé par « User-Centric », nous avons d'abord proposé de répondre aux différentes mobilités par la gestion d'un réseau représenté à travers les enchaînements de composants de même natures

dénommés VPXN (X=Utilisateur, Service, Connectivité, Equipement). Nous rappelons que les communautés d'intérêt définies ici ont pour l'objectif la continuité de service, et sont proposées comme les « services supports » de chacun de réseau virtuel (VPXN). La gestion de ces éléments en communauté et leurs interfonctionnements doivent être donc autonomiques et dynamiques.

Nous définissons la gestion de communauté comme un contrôle fonctionnelle du système par la communauté virtuelle en fonction de leurs comportements communs et/ou leurs caractéristiques communes. Elle s'auto gère automatiquement en fonction des mobilités en temps réel. Leurs comportements communs peuvent être : la même localisation, la même organisation, les mêmes besoins et les mêmes préférences, etc.

Nous organisons les utilisateurs qui ont la même caractéristique dans une même Communauté Virtuelle (CV), ce qui permet à un utilisateur de recevoir les informations intéressant le groupe ou de remplacer l'un et l'autre suivant leur activité sociale (VPUN).

Le VPSN est alors construit et maintenu par sélection des ES appropriés des différentes Communautés Virtuelles de Services (CVS). Les remplacements des ES doivent être faits dans chacune des CVS de façon dynamique et autonome. Quant à VPCN, les NEs (Network Element) sélectionnés doivent pouvoir être remplacés par les NEs qui ont la même fonctionnalité. Les CV de Réseau (CVR) sont responsables de l'organisation des noeuds de type backup.

Finalement, les EEs (Element d'équipement) qui ont la même caractéristique doivent être regroupés dans une même CV de Equipement (CVE) pour garantir le bon fonctionnement du VPEN.

De plus, la construction et l'autogestion de CV dépend non seulement des fonctionnements équivalents mais aussi de la partie non fonctionnelle : les besoins de QoS de bout en bout. Le VPXN correspondant doit sélectionner le composant qui répond au mieux au SLA (Service Level Agreement) et maintenir la QoS de bout en bout en cas de dégradation de QoS causée par les différentes mobilités pendant l'exploitation.

A travers ces exigences que nous avons retenues, nous essayons de préciser les fonctions génériques de gestion selon les différentes phases.

Création d'une communauté

Nous nous concentrons dans la couche de service. Nous prenons un ES comme un Peer. Quand un ES est installé dans la plate-forme de service, il déclare tout de suite le service qu'il peut offrir, ce qui permet aux autres ESs d'y accéder. Il annonce également sa fonctionnalité, ce qui permet de l'associer au groupe des Peers adéquat, c'est-à-dire au bon CVS. Par exemple (Figure V.13), une CVS est construite en fonction de la fonctionnalité équivalente (Fun1).

Quand un utilisateur demande une application, un VPSN est créé pour gérer les composants de service répondant à son besoin. Pour la fonction Fun1, le VPSN consulte le CVS avec une QoS1 afin de décider quel composant prendre. Le CVS est donc filtré d'abord selon les ES ayant la QoS demandée et met les candidatures dans une communauté. Nous avons alors une communauté où se trouve le composant A avec ses ES fonctionnellement et QoS équivalentes. Dans l'exemple, c'est le composant A qui est finalement sélectionné par le VPSN.

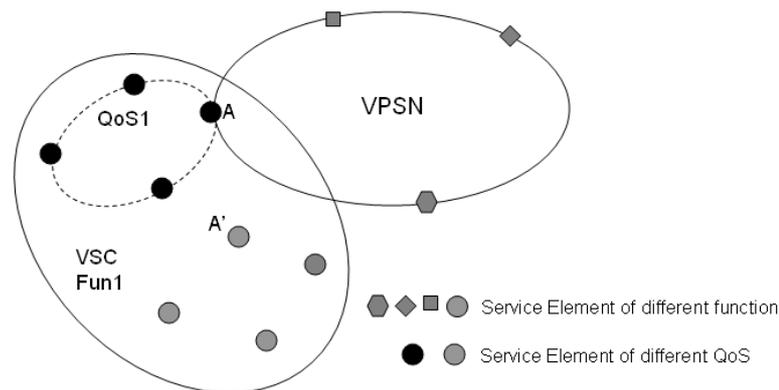


Figure V.13: Création de communauté

Les fonctionnalités que nous pensons nécessaires pour la création de la communauté virtuelle sont:

- **Authentification**

L'authentification est la procédure qui consiste, pour un système informatique, à vérifier l'identité d'une entité (personne, ordinateur...), afin d'autoriser l'accès de cette entité à des ressources (systèmes, réseaux, applications...). L'authentification permet donc de valider l'authenticité de l'entité en question. Elle est nécessaire à tout accès et celle du coût (Cost) nécessaire à tout choix.

- **Cost**

Cette fonction a pour but de calculer le montant total des dépenses pour réaliser un service. Le coût peut être décomposé en frais de base et en coût de transaction demandée par le fournisseur.

- **Profile & usage profile**

Pour avoir une représentation homogène de tous les composants, on adopte la QoS comme le dénominateur commun. Les critères de QoS sont sélectionnés pour construire les profils qui caractérisent un membre d'une communauté. Les informations de la capacité QoS (le valeur conception de la QoS maximum d'un composant peut offrir) sont dans le profil de ce composant, l'usage de ces composants se trouve dans le profil de usage. L'usage est relié avec l'environnement d'exécution d'un composant, donc le profil d'usage doit comprendre les informations pertinentes pour décider comment utiliser ce composant. On propose de le structurer à travers la QoS demandée, la QoS offerte et les contraintes.

L'exploitation d'une communauté

Dans la phase d'exploitation du CV, nous devons tenir compte non seulement les exigences spatiales mais aussi ceux de nature temporelle car le composant doit être toujours connecté malgré son changement de localisation ou d'activité. Or, les éléments associés à localisation et à l'activité doivent être accessibles en temps réel avec les contraintes.

Avec ces besoins, nous identifions les services de base comme les suivantes :

- **LBS (Location Based Service)**

Ce service de base a pour but de déterminer l'endroit géographique d'un composant de la communauté en fonction de l'information donnée. Ce qui peut être l'adresse ou la position de ce composant. LBS peut servir de trouver un composant avec une fonction et une QoS précise pour l'ajouter à la communauté.

- **PBS (Presence Based Service)**

Ce service de base a pour but de trouver les composants qui sont à proximité d'une position connue d'un composant avec des critères. Les résultats de PBS peuvent servir à donner les possibilités concernant une demande.

- **DBS (Discovery Based Service)**

Ce service de base a pour but de découvrir les nouveaux membres qui sont qualifiés pour entrer dans la communauté actuelle.

- **ABS (Activity Based Service)**

Ce service de base est pour identifier l'activité commune des composants dans un même groupe.

Ces fonctions sont les fonctions de base pour gérer tous les membres de la communauté en temps réel. En considérant un dysfonctionnement d'un membre de la communauté, grâce aux services de base, on peut offrir les réactions correctes et faire les mises à jour des membres de la communauté. Nous allons détailler des cas d'usage représentatifs pour montrer comment ces services de base peuvent être utilisées afin d'avoir la gestion de communauté. De plus, avec cette gestion, la mobilité correspondante peut être résolue.

- Gestion de communautés des utilisateurs

Supposant que (Figure V.14) Alice veut acheter un cadeau Noël samedi et compte rejoindre Bob aux galeries Lafayette. Elle passe d'abord par le magasin du Printemps, et se retrouve avec tous ceux qui sont présents autour de ce dernier (UPBS). Grâce à son agenda, l'UABS déclare automatiquement que l'activité d'Alice est l'achat du cadeau. Donc elle rejoint la communauté des consommateurs (Communauté 1) pour recevoir les bonnes informations. Des bons de réduction électronique Printemps sont automatiquement envoyés par SE5 selon les préférences qu'Alice a indiquées dans son profil d'utilisateur. Les conseillers qui offrent les bons de réduction sont ceux qui sont retrouvés (UDBS) autour d'Alice et qui sont réunis dans la Communauté 2. Quand elle arrive près des galeries Lafayette, les informations de Lafayette sont diffusées par SE6 et sont aussi reçues par Alice car grâce à la localisation (ULBS), Alice est ainsi incluse dans la communauté 3 de 'Lafayette'. Bob est supposé arriver un peu plus tard qu'Alice, il sera alors inclus dans la communauté 3 de 'Lafayette', Il lance alors UDBS pour trouver Alice.

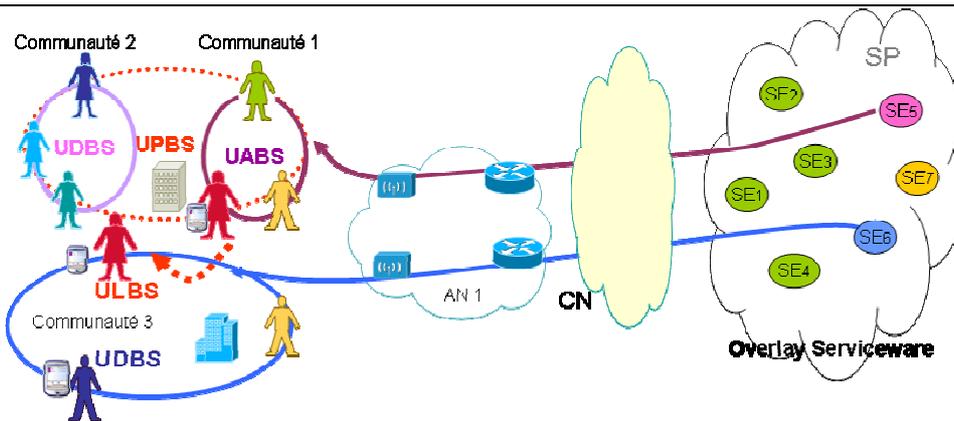


Figure V.14: Exemple de la gestion de la communauté des utilisateurs

Notre approche facilite les diffusions et le partage d'information grâce à cette construction dynamique de communautés d'utilisateur. Ce genre des communautés a pour objectif de réunir les utilisateurs ayant les mêmes caractéristiques, ce qui intéresse les fournisseurs de contenu. Aujourd'hui, les caractéristiques de l'utilisateur sont enregistrés dans la base de données du côté l'opérateur, le diffuseur de l'information peut la consulter avant d'envoyer les publications. L'objectif est de permettre aux diffuseurs d'informations d'envoyer des informations sans connaître spécifiquement chaque user et d'augmenter leurs chances de toucher plus de personne et d'avoir l'effet « longue traîne ».

Avec les services génériques proposés avant, nous avons ici la gestion de la communauté des utilisateurs comme suit (Figure V.15) :



Figure V.15: Gestion de la communauté des utilisateurs

- Gestion de communautés des équipements

La gestion de cette communauté peut supporter deux types de mobilité : la mobilité de terminal et la mobilité de l'utilisateur. Nous regardons d'abord la mobilité de terminal.

Un utilisateur souhaite connaître les terminaux utilisables dans un contexte mobile, dynamique et en temps réel. Terminaux autour d'un utilisateur sont organisés de manière autonome, un nouveau PAN doit être prêt dès l'arrivée de l'utilisateur a un endroit donné. Aujourd'hui, un utilisateur possède plusieurs terminaux hétérogènes. C'est l'opérateur qui conserve l'information de ces terminaux concernant un utilisateur. Une base de donnée centralisée et statique est employée. Supposant que (Figure V.16) Alice est localisée dans son bureau (DLBS) grâce à son terminal (portable), le service de base de DPBS l'aide de savoir tous les équipements autour quand elle est dans son bureau. De plus, la fonction DDBS avec sa propre droit d'accès organise les terminaux utilisables qui lui appartenant pour construire son PAN (PC, Phone et PDA).les ressources fonctionnellement équivalents sont ainsi réunies par le DPBS dans leurs communautés pour les

différentes activités (scanner ou imprimer). Ces communautés des équipements préparent des ressources pour une utilisation sans coupure et fiable en temps réel. Quand elle se déplace dans une salle de réunion (DLBS), à travers du service de base de DABS, son activité de conférence sur son agenda lui permet de joindre la communauté de la conférence, là où se présentent (DPBS) les équipements des autres utilisateurs, les équipements sur place (IP phone et projecteur). Supposant que en cours de réunion, les deux participants ont allumé leur projecteurs apportés, ceux deux nouveaux membres sont alors découvertes (DDBS par la fonction « projeter ») et regroupés avec le projecteur actuel utilisé dans la communauté de projecteur.

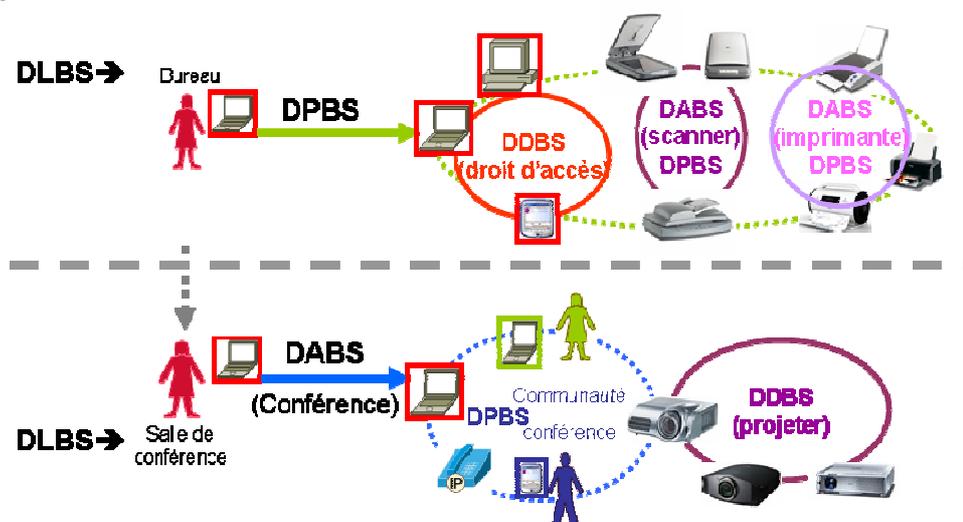


Figure V.16: Exemple de la gestion de la communauté des équipements 1

Notre approche est « User centric ». Selon cette vue, tous les équipements doivent être organisés autour de l'utilisateur. De plus, ils concernent un utilisateur dédié. Communauté de équipements gère, dynamiquement et automatiquement, les terminaux autour d'un utilisateur pour que l'utilisateur puisse toujours avoir une bonne sélection de terminal répondant différents services. Dans cette vision, la mobilité de l'utilisateur peut aussi être supportée.

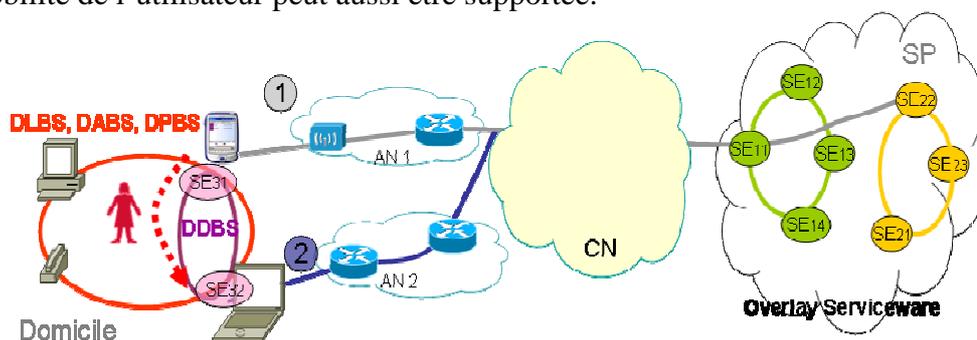


Figure V.17: Exemple de la gestion de la communauté des équipements 2

Alice est en train (Figure V.17) de regarder une vidéo dans son PDA. L'ordinateur fixe, IP phone, PDA et le portable sont dans la communauté des équipements domicile (DLBS, DABS et DPBS). D'après sa préférence, un écran plus grand est préféré pour ce type de service grâce à la communauté des équipements fonctionnellement équivalents (SE31=SE32= service d'affichage), selon le DDBS, SE32 installé sur son portable est trouvé comme celui ayant la meilleure QoS par

rapport au SE31 installé dans le PDA. Le transport est alors changé à travers AN2. Tous ces traitements sont transparents à l'utilisateur grâce à l'autogestion de la communauté des équipements Fonctionnellement équivalents (service affichage). En conclusion, cette communauté s'auto gère grâce aux fonctions qui sont des services de base :

- Location est pour identifier où se trouve le terminal;
- Découverte est pour avoir des terminaux disponibles selon un critère sélectionné.
- Activité est pour donner les contraintes et les possibilités pendant la construction d'une communauté avec les préférences de l'utilisateur.

Avec les services génériques proposés avant, nous avons ici la gestion de la communauté des équipements comme suit (Figure V.18) :



Figure V.18: Gestion de la communauté des équipements

- Gestion de communautés des services

Cette fois-ci, la problématique est de comment assurer une session service de bout en bout dans une manière sans coupure et sans couture quand le terminal se déplace. De plus, l'utilisateur n'a pas conscience des traitements.

La session de service ne doit pas être coupée quand un meilleur composant de service est trouvé et remplacé automatiquement. Aujourd'hui, l'utilisateur est obligé de couper la session pour choisir un fournisseur qui offre cette fonction avec la QoS souhaitée. Dans notre approche, les éléments des services sont organisés dans une communauté de service fonctionnellement et QoS équivalents. L'autogestion des communautés de service va assurer la mobilité de service.

Alice se déplace (Figure V.19) avec son PDA tout en regardant un film fournit par SE22 et synchronisé par SE11. Après son arrive à AN 1', la QoS de service offerte par SE22 se dégrade, grâce à l'autogestion de la communauté de service SE2x (SLBS, SPBS, SDBS), SE21 est automatiquement sélectionné pour une session de service sans coupure.

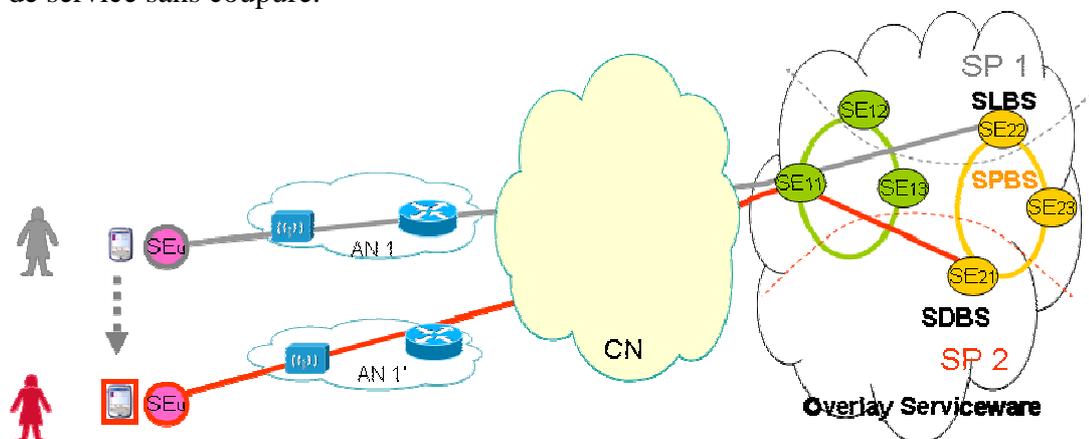


Figure V.19: Exemple de la gestion de la communauté des services

Avec les services génériques proposés avant, nous avons ici la gestion de la communauté des services comme suit (Figure V.20) :



Figure V.20: Gestion de la communauté de services

L'autogestion de communauté par la QoS

Nous prenons en compte les impacts des mobilités qui causeront les dégradations de QoS durant la fourniture de service « User-Centric ». Nous nous situons toujours sur la couche de service.

Supposant que dans la Figure V.21, la mobilité de terminal ou la mobilité de l'utilisateur introduit un non respect de la QoS sur le composant A. La gestion de la mobilité doit maintenant chercher un autre composant A' pour le remplacer. Mais avec le remplacement, ce n'est pas que la QoS du composant qui doit être respecté, mais aussi la QoS de bout en bout (passage de M à N à travers A') qui doit être maintenue.

Donc, à la suite de notre proposition dans la phase de la création de communauté, maintenant, nous cherchons une solution qui permet d'avoir également la QoS du lien.

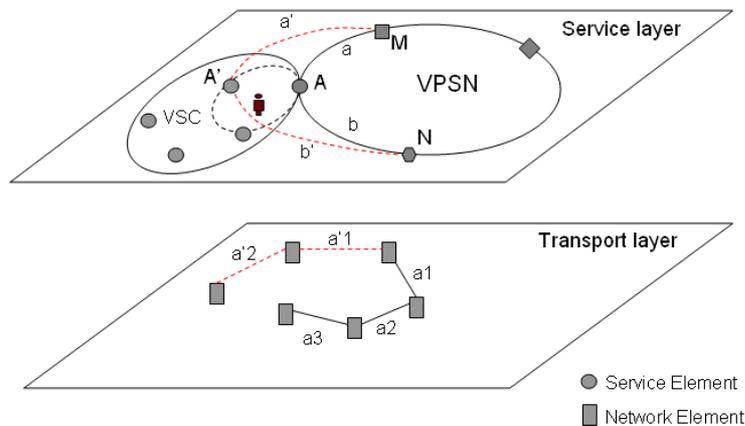


Figure V.21 : Autogestion de communauté

Tout d'abord, le CVS reçoit la mise à jour de la localisation de l'utilisateur. Grâce à LBS, nous pouvons d'abord obtenir l'adresse physique en fonction de son adresse logique. Le CVS invoque le PBS pour connaître tous les composants disponibles autour de l'utilisateur. Finalement, le CVS compare et filtre la QoS des candidats avec celles de leurs liens.

Par exemple, nous comparons la QoS (a, A, b) avec la QoS (a', A', b'). Dans la table 1, on consigne tous les serveurs au niveau du Réseau, des composants de service s'exécutant sur eux et des QoS offertes. Dans notre cas, on suppose qu'il y a 4 serveurs, sur lesquels A, A', M et N sont lancés.

Composant de service	Serveur 1	Serveur 2	Serveur 3	Serveur 4
A	QoS ₁	-	-	-
A'	-	QoS ₂		
M	-	-	QoS ₃	-
N	-	-	-	QoS ₄

Tableau V-1 : QoS du mapping entre le niveau de Service et de Réseau

Le niveau de Réseau tient une table 2 pour des QoS des liens entre les serveurs. Si deux serveurs sont connectés, on calcule la QoS du chemin d'après la table de routage au niveau de Réseau et on la consigne dans cette table.

Composant de service	Serveur 1	Serveur 2	Serveur 3	Serveur 4
Serveur 1	-	-	QoS _(1,3)	QoS _(1,4)
Serveur 2	-	-	QoS _(2,3)	QoS _(2,4)
Serveur 3	QoS _(1,3)	QoS _(2,3)	-	QoS _(3,4)
Serveur 4	QoS _(1,4)	QoS _(2,4)	QoS _(3,4)	-

Tableau V-2 : QoS au niveau de Réseau

La VSC a trouvé trois nœuds fonctionnellement équivalents et QoS équivalentes, qui sont dans le chemin de M à N. On prend les deux nœuds A et A' comme un exemple. Pour choisir un nœud plus approprié, la VSC compare de QoS de M-A-N et de M-A'-N.

D'après table 1, on trouve que M est lancé sur Serveur 3, et A est lancé sur Serveur 1. La QoS du lien M-A est ainsi exactement la QoS du Serveur 3 au Serveur 1, qui est égale à QoS(1,3) d'après table 2. De la même façon, on obtient la QoS du Serveur 1 au Serveur 4 - QoS(1,4). Donc la QoS (M-A-N) égale {QoS1, QoS(1,3), QoS(1,4)}. On peut de même trouver que la QoS (M-A'-N) égale {QoS2, QoS(2,3), QoS(2,4)} selon les deux tables. Ça c'est le processus de Requête de QoS.

Composant de service	{M, N}
A	{QoS ₁ , QoS _(1,3) , QoS _(1,4) }
A'	{QoS ₂ , QoS _(2,3) , QoS _(2,4) }

Tableau V-3 : Résultat de la Requête de QoS

Nous obtenons les fonctions suivantes:

- **QoS Management (VXC) (X= device, service...)**

Nous pouvons faire la gestion selon les critères de la QoS en construisant des communautés, des communautés virtuelles VXC (X= device, service...). Cette fonction fait sa propre autogestion pour avoir des membres fonctionnellement et QoS équivalents.

- **QoS Continuity (par composant)**

La continuité de la QoS par composant se fait par la fonction de "OoS continuity" qui réagit aux pannes et aux changements de QoS.

- **Selecting**

Nous avons la « sélection » des nouveaux membres de la communauté (Selecting). Cette fonction a pour but d'augmenter la qualité de service à travers l'auto gestion de la communauté. Le membre le plus adapté est choisi pour construire ou maintenir la communauté sur demande.

V.4. Continuité de service intégrée : Sessionware

Pour assurer la continuité de service de bout en bout, nous proposons d'abord une session mobile et unique orientée utilisateur (§V.4.1). A travers cette session, les gestions de différentes mobilités peuvent être prises en compte par le Binding (§ V.4.2) dynamique des quatre niveaux architecturaux sans oublier les préférences de l'utilisateur. Finalement, les « services supports », dénommé « sessionware » (§V.4.3) traitent l'agrégation et la fourniture finale de service demandé. La session tient également compte de la partie personnalisation grâce aux préférences fournies par « Seamless Userware » (§V.4.4).

V.4.1 Concept de session « User-Centric »

Les demandes d'utilisateur, telle que la réalisation d'un plan de voyage, ont besoin de plusieurs services coordonnés ensemble. Aujourd'hui, l'utilisateur est obligé de gérer ce processus de coordination lui-même. Il doit d'abord ouvrir le premier site pour réserver des tickets, et après le deuxième pour l'hôtel. En terme de technique, il doit ouvrir deux sessions différentes séparément. Plus généralement, nous pouvons dire que aujourd'hui nous sommes « application centric », c'est-à-dire que suivant les contraintes des applications, l'utilisateur est amené à ouvrir autant de sessions particulières que d'applications.

Pour palier cet inconvénient, nous proposons de mettre en œuvre une seule session de bout en bout regroupant toutes les applications demandées. L'objectif étant que la session soit orientée utilisateur, basée sur des blocs fonctionnels qui seront en plus ubiquitaires pour supporter la mobilité.

Cette proposition de la session « User-Centric » vise à fournir les services personnalisés automatiquement au travers d'une demande utilisateur. L'utilisateur suivant sa logique de service, formule sa demande une seule fois au début et il obtient le déroulement des opérations de façon automatisée et transparente. La session « User-Centric » est responsable de la composition de service dynamique, ainsi que son support en terme de la connectivité de bout en bout. Il combine tous les services séparés ensemble et génère une session unique (Figure V.22).

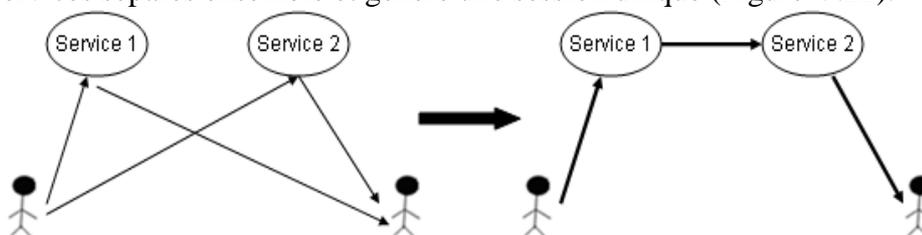


Figure V.22:Sessions Multiples vs. Session Unique

Tous les échanges de données, l'établissement et la destruction de la session, ou toutes les autres activités comme modification, se passeront dans cette session spéciale.

Nous ne traitons pas la sécurité dans cette thèse, mais un des objectifs de cette session est que l'utilisateur ne soit obligé de s'authentifier qu'une seule fois. Le

système devant prendre en charge la relation avec les différents services sollicités par la session.

V.4.2 Le Binding de la session

Basé sur le modèle VPSN, la session « User-Centric » suit aussi le principe du service de routage. La demande de l'utilisateur invoque le premier service. C'est au premier service de trouver le deuxième service et d'invoquer son successeur automatiquement. De cette façon, une chaîne de services peut être générée pendant ce processus. Ces services sont activés un après l'autre pour réaliser l'application demandée par l'utilisateur.

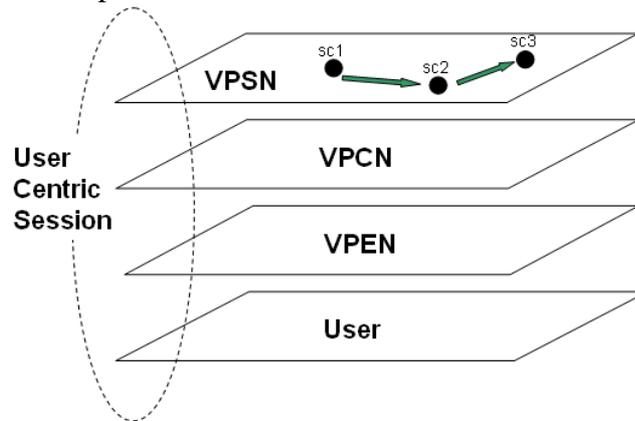


Figure V.23: Concept de la session « User Centric »

Comme le montre la Figure V.23, grâce à notre modèle architectural VPXN, la session « User-Centric » a des services supports dans chacune des couches. Donc la session elle-même doit s'adapter à toutes les modifications selon la mobilité ou l'environnement ambiant. Nous introduirons un automate pour réaliser la gestion automatique et dynamique de cette session. C'est cette session « User-Centric » qui garantit une mise en relation complète, pratique et continue pour les utilisateurs. Donc, nous proposons de raffiner sa gestion automatique selon les différentes mobilités.

Notre approche est de type « User-Centric », la session « User-Centric » est proposée pour un utilisateur dans la couche d'utilisateur traversant toutes les couches de VPSN, VPCN, et VPEN, les changements dans chaque couche peuvent générer des mobilités. Nous rappelons que nous avons principalement quatre types de mobilités : la mobilité d'utilisateur causée par le mouvement de l'utilisateur ; la mobilité de terminal causée par le changement de l'équipement ; la mobilité de réseau causée par l'environnement ambiant du réseau ; et la mobilité de service causée par la disponibilité de composants de service. La session « User-Centric » doit s'adapter à toutes ces mobilités.

Supposons que l'utilisateur A est en train d'utiliser une chaîne de service sc1, sc2 et sc3. Ces trois composants de service sont supportés par trois composants de réseau cc1, cc2 et cc3 de la couche VPCN respectivement. Dans la couche VPEN, l'utilisateur envoie et reçoit son application à travers un terminal mobile. A un moment donné, l'utilisateur A bouge, la performance de cc1 se dégrade et ne peut pas fournir la qualité de service nécessaire à sc1 non plus. Le processus d'adaptation commence, comme le montre la Figure V.24.

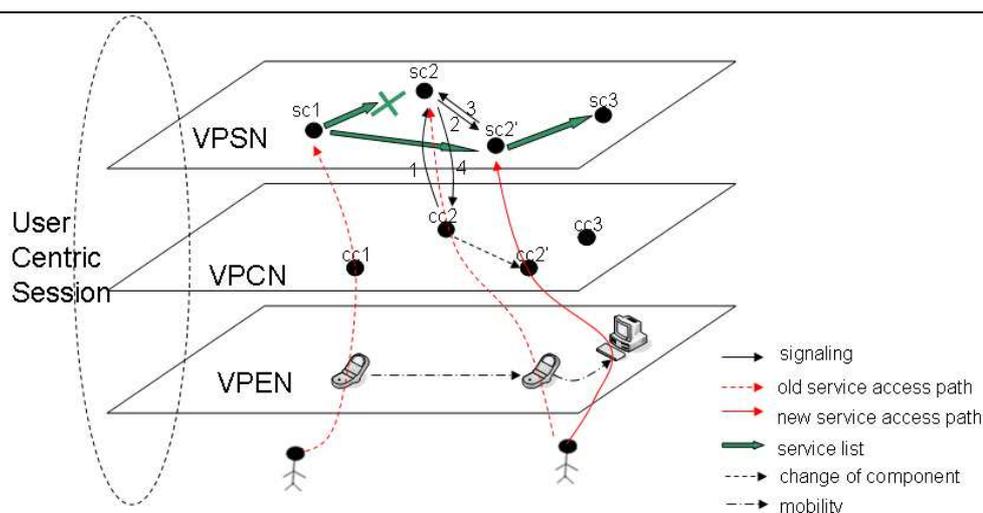


Figure V.24: Gestion automatique selon la mobilité d'utilisateur

Une fois que cc2 estime la dégradation de performance à cause du déplacement de terminal, il envoie un message 1 de signalisation à sc2. Grâce à l'autogestion de CVS (§V.3.4), il y a toujours des candidats fonctionnellement et QoS équivalents pour un certain type de service. Donc sc2 commence à chercher son remplaçant immédiatement. Une fois sc1 trouve le substitut sc2', il invoque sc2' par l'envoi du message 2. Sc2' répond à cette demande par un ACK (message 3). Jusqu'à maintenant, sc2' a réussi à prendre la place de sc2. La mobilité de service est alors supportée. Ensuite, sc2' invoque sc2 et sc2 invoque sc3 pour continuer la chaîne des services. Enfin, un message 4 est envoyé de sc2 à cc2, pour l'informer du changement du composant de service. A cause de la mobilité de l'utilisateur, il faut transférer la connexion à un autre composant réseau. cc2' est alors sélectionné pour associer le nouveau terminal avec le réseau.

De cette façon, la continuité de cette session est maintenue en fonction de mobilité.

Toutes ces opérations sont transparentes pour l'utilisateur et sont effectuées automatiquement dans la session « User-Centric » elle-même.

V.4.3 « Sessionware »

Après avoir envisagé les autogestions des communautés virtuelles et la proposition de la session « User-Centric », pour répondre les cas d'usage couvrant chacune de mobilité, Nous préconisons un Binding de bout en bout dans un déroulement sous la même forme (Figure V.25).

Comme nous avons indiqué, les dysfonctionnements de la continuité de service sont principalement à cause des mobilités. Grâce à notre architecture à quatre visibilité, nous pouvons gérer la mobilité de session dans une dimension temporelle de bout en bout.

Le Binding de la session est décomposé en Binding architecturalement horizontal (Terminal Binding, Bearer Binding, Connectivity Binding et Service Binding), chaque sous Binding est responsable de liaison de composant choisi dans la chaîne de la session. Les informations de la QoS peuvent être obtenues et mise à jour au travers de l'Infoware. Les préférences de l'utilisateur peuvent être prises en compte à travers le « Seamless Userware ».

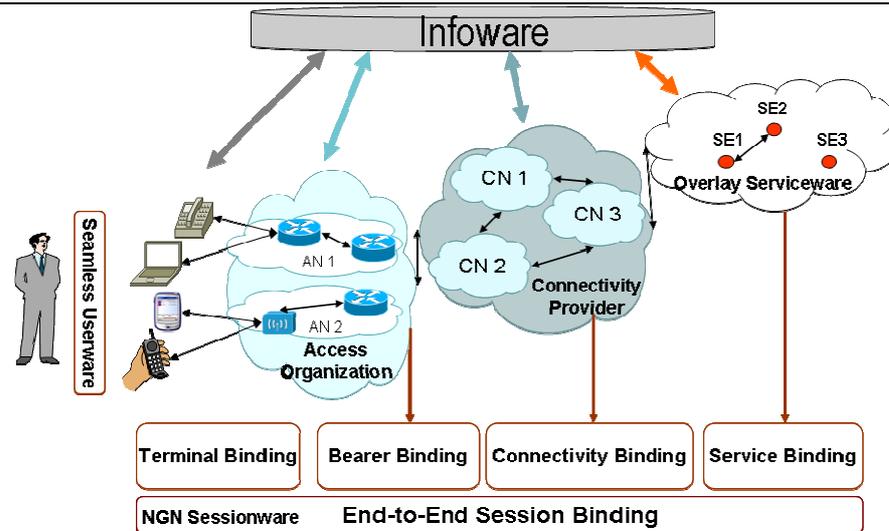


Figure V.25: Binding de la session de bout en bout

Pour répondre aux fonctionnalités demandées afin de construire et de maintenir la session « User-Centric », nous identifions maintenant, comment la gestion de la session coopère avec les autres propositions que nous venons de définir. Dans la topologie fonctionnelle (§V.1), nous avons défini EDA comme un des points avantageux de nos propositions. Un automate (Figure V.26) est proposée pour prendre en compte tous les événements possibles qui invoqueront le Binding ou le ReBinding de la session « User-Centric ».

Dans notre proposition, nous considérons l'utilisateur comme le point d'extrémité de la session mobile. Donc les événements qui nous intéressent sont les changements à cause des mobilités et changements au niveau de service (QoS ou composant). Nous pensons qu'avant la fourniture de service global, pour une session d'utilisateur, la gestion de la session attend les constructions des VPxNs pour provisionning. Nous rappelons que dans notre concept, chaque VPxN est responsable d'un type de mobilité dans son niveau de visibilité. La session « User-Centric » est donc en fait composée par la maintenance de ces trois VPxN simultanément.

Une fois le provisionning fini, le Binding est responsable de les agréger et de donner une fourniture du service final. Une session « User-Centric » est alors obtenue. Notre objectif est de maintenir cette session en temps réel en fonction des mobilités. Donc nous proposons d'abord d'identifier les événements qui causeront la modification la de session. Le changement de la location géographique et le changement des activités de l'utilisateur sont les événements extérieurs qui déclencheront la modification de la session. Par contre, les dégradations au niveau de la QoS des entités ont besoin des mises à jour des VPxNs, ce dernier est identifié donc en tant que l'événement interne. La modification de session doit être transparente pour être sans couture, ce qui demande une manière du type MBB (Make Before Break). MBB veut dire qu'avant de supprimer et lâcher l'ancien composant, il faut déjà une mise en relation avec le nouveau composant dans la chaîne. Le MBB a pour objectif de rendre le maximum de transparence à l'utilisateur.

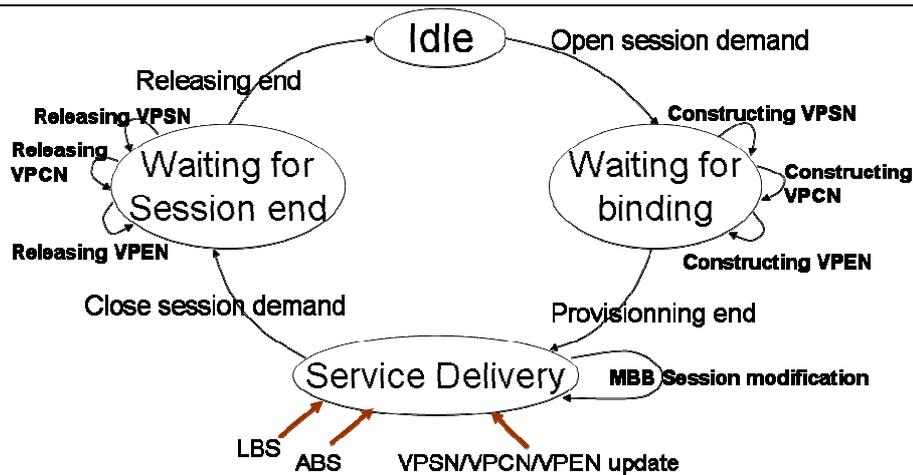


Figure V.26: automate de sessionware

Quand l'utilisateur a décidé de finir sa session, la session correspondante demande une clôture de la session, tous les VPXNs sont alors lâchés.

En fait, une vision système du Binding de la session est que ce dernier est supporté par les entités de gestion, une base de connaissance « Infoware », un « Seamless Userware » et le « NGN Sessionware » (Figure V.27).

Comme nous avons identifié les événements de LBS et ABS comme des déclencheurs extérieurs, les autogestions des communautés correspondantes sont donc activées pour maintenir leurs VPXNs. Puisque la session est orientée utilisateur, ses préférences sont aussi prises en compte grâce au « Seamless Userware » qui travaille d'une façon dynamique et automatique. Chaque VPXN gère le Binding de son niveau, c'est au « NGN Sessionware » d'être responsable du respect de la QoS de bout en bout.

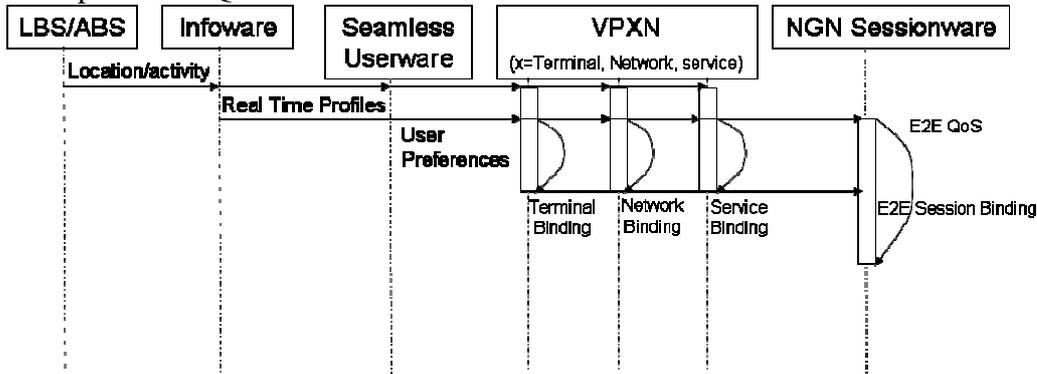


Figure V.27: Diagramme séquence du Binding

Les fonctions dans le «NGN Sessionware» (Figure V.28) sont donc identifiées:

- Service agrégation

Au delà du réseau de service overlay. L'agrégation de service est la mise en relation dynamique de tous les services offerts par les composants de niveau différent (service, réseau, et équipement).

- Continuous service delivery

Cette fonction responsable de la fourniture des services pour l'application demandée par l'utilisateur.

- E2E QoS management

Cette fonction nous permet d'avoir une continuité de QoS de bout en bout.

- Authorization
- Charging aggregation
- Billing (one bill)

Ces trois fonctions nous permettent d'enchaîner tous les composants de chaque visibilité et d'offrir à l'utilisateur un charging global et une seule facture.

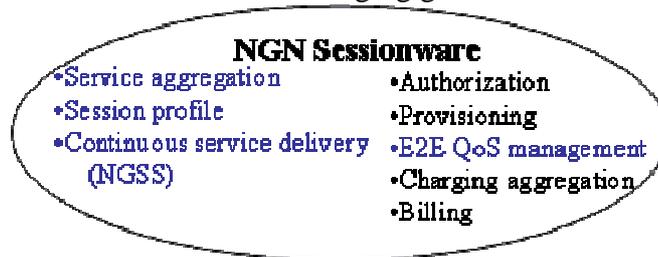


Figure V.28 : « NGN Sessionware »

V.4.4 « Seamless Userware »

Nous pouvons apercevoir que pour avoir une personnalisation de service, nous avons adopté non seulement la vision « User-Centric », mais aussi un filtrage des préférences de l'utilisateur pendant le Binding de la session (Figure V.25).

Nous proposons un « Seamless Userware » (Figure V.29) avec l'utilisateur pour interférer. Les préférences sont mises en fonction de la localisation des ressources, et/ou suivant son activité à un instant donné. Nous devons traduire les préférences en fonction de sa location et son activité temporelle. Le filtrage par les préférences nous permet de déterminer la rentabilité des ressources ambiantes. La priorité de l'utilisation de chaque ressource ambiante peut être déterminé se basant sur un mécanisme de calcul de base sur les poids [1] Après l'application des préférences d'utilisateur, l'information sur des ressources disponibles (« Real Time Profile » dans l'Infoware Figure V.25) pour la maintenance de la session est ressortie.



Figure V.29 : « Seamless Userware »

V.5. Synthèse

Une analyse plus en détaillée nous permet d'identifier les interfaces qui existent entre les blocs fonctionnels dans une carte fonctionnelle (Figure V.30).

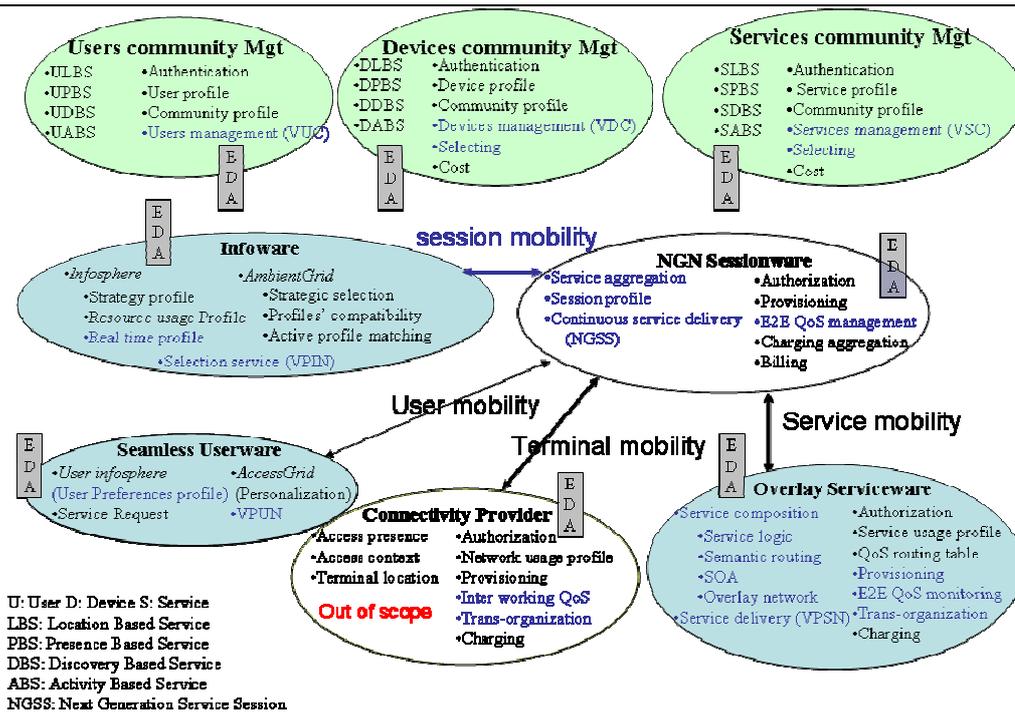


Figure V.30 : Carte fonctionnelle

Les besoins de la gestion en temps réel et ceux de faire coopérer les « communautés » nous imposent une autogestion en fonction de mobilités. Finalement, les handovers horizontaux et les handovers verticaux qui existent dans le réseau de transport nous demande un « offreur de connectivité » selon la meilleure connectivité.

Pour répondre à ces besoins, nous identifions respectivement les usecases suivants : « seamless userware », « service session provider » et « connectivity provider », pour offrir la mobilité de session et la gestion de la QoS de bout en bout.

Nous détaillons que le « seamless useware » est la version évoluée de l'usware du projet VTC2e, qui rajoute un profil de temps réel (real time profile) pour avoir l'information concernant un utilisateur lors de ses déplacements. L'inférence correspondante est la sélection de l'offreur (Provider selection), pour construire un réseau virtuel de l'utilisateur correspondant à une demande spécifique (VPUN).

Pour la composition de service qui intègre le SOA pour avoir une architecture horizontale de service du type réseau overlay. La seconde est pour un enchaînement dynamique et flexible des composants dans un VPSN.

Nous rappelons que toutes les entités dans notre architecture sont mutualisables et partageables. Les blocs fonctionnels sont tous du type EDA, ce qui rendre la flexibilité au système.

Dans le « Overlay Serviceware » proposé, la gestion se fait dans deux parties : une partie est la gestion de la composition de service dans une architecture horizontale pour offrir un réseau overlay de services. Une autre partie est la gestion de logique de service pour une session de services.

Nous reviendrons ensuite sur la QoS, les entités ont leurs propres gestions de performance pour contrôler et optimiser leur comportement. Cette gestion locale est faite par monitoring leurs caractéristiques non fonctionnelles, c'est-à-dire, leurs QoS.

En effet, durant l'exploitation du service, sa QoS reflétera si ce composant fonctionne conformément au SLA ou pas. Une QoS de bout en bout est souhaité quand nous considérons une vraie application de services. Assurer les applications globales à travers la construction d'une session virtuelle et crosslayer est notre but final. Donc les aspects QoS doivent être intégrés dès le début de la phase conception jusqu'au bout.

Pour avoir une vision plus claire de toutes nos propositions, nous proposons de les mettre dans un scénario final (Figure V.31).

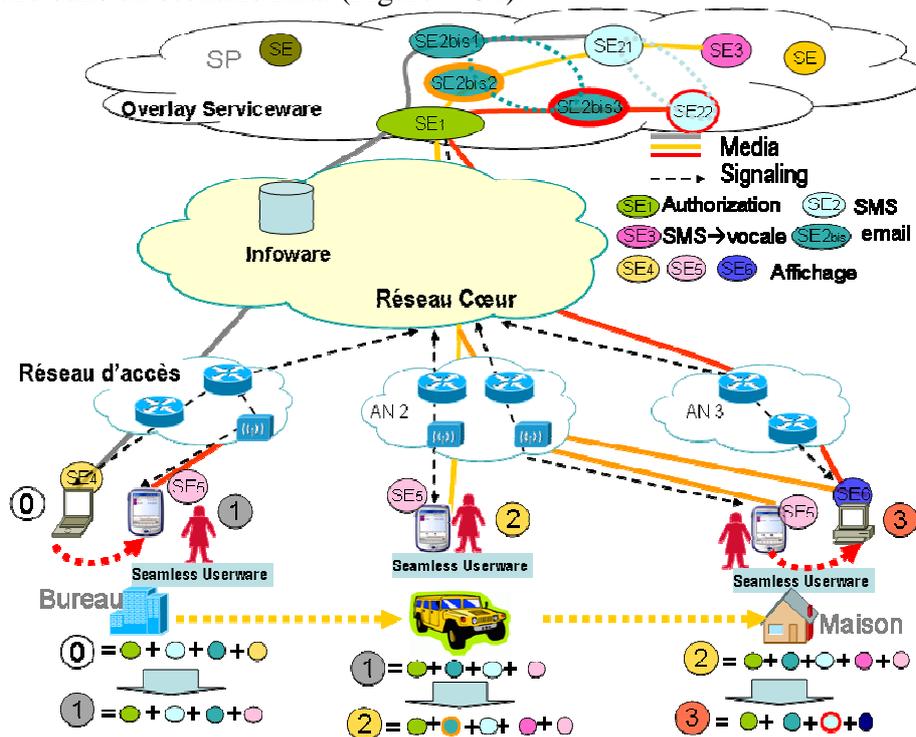


Figure V.31: Scénario final

Un utilisateur emploie son ordinateur portable quand elle est dans son bureau, le SE4 qui est en fait un composant applicatif, par exemple l'affichage, du côté terminal est en train de fonctionner. Supposant qu'elle souhaite également recevoir les emails (SE2bis) et SMS (SE2) à travers un même terminal, la première application « User-Centric » est alors construite par une logique de service (0) :

$$LS_0 = SE_1 \text{ (authorization)} + SE_2 + SE_{2bis1} + SE_4. \tag{0}$$

Dans son bureau, le PAN (1) d'Alice représente tous les équipements qui lui appartiennent :

$$PAN_{\text{Bureau}} = \{\text{laptop, PDA}\} \tag{1}$$

Il est 18H00, elle est supposée sortir du bureau. Mais suivant son agenda, elle attend toujours un dernier email de confirmation de son client (UABS). Car il n'est pas pratique d'utiliser l'ordinateur portable pendant la route, Grâce à sa communauté des équipements PAN au bureau, son PDA est trouvé (DABS, DPBS) disponible pour l'activité de « travail » pour avoir tous les services. Avec la logique de service (0), SE4 est identifié comme la fonction souhaitée. Le

composant de service SE5 est trouvé (DDBS) équivalent de SE4, donc le PDA est choisi (DSelecting) pour continuer la session d'application d'Alice. La session répond à la mobilité de l'utilisateur et un changement de SE4 à SE5 peut se faire, une LS1 (Figure V.31 - 1) est l'application actuelle (2) :

$$LS_2 = SE_1 + SE_2 + SE_{2bis1} + SE_5. \quad (2)$$

Alice se déplace dans sa voiture, un deuxième PAN (3) est alors automatiquement construit.

$$PAN_{voiture} = \{PDA\} \quad (3)$$

Puisqu'elle n'a plus de choix de terminal, elle reste donc avec le PDA. Grâce à son agenda, nous pouvons prévoir son trajet de 18H10- 19H30 avec sa voiture (UABS), elle ne peut pas voir ses email pendant son trajet en voiture, une modification sur la logique de service (LS3) (Figure V.31 - 2) est souhaitée par les préférences de l'utilisateur : elle souhaite recevoir ses SMS en vocal. La session actuelle inclut automatiquement le SE3 qui est SMS → vocal. Puisque pendant le déplacement d'Alice, le fournisseur de réseaux mobiles est plus capable d'offrir le service email à travers la 3G (AN2) grâce à sa couverture plus fine que le WiFi, la communauté de service (4) remplace automatiquement SE2bis1 avec SE2bis2.

$$\text{Communauté de Service (email)} = \{SE_{2bis1}, SE_{2bis2}\} \quad (4)$$

Le changement de la composition de service notifie dynamique à VPSN. Le réseau support (VPCN) s'autogère au niveau du fournisseur des réseaux mobiles pour offrir une meilleure connectivité. Grâce au « NGN Sessionware », la session des services est toujours sans couture avec la logique de service (5) :

$$LS_3 = SE_1 + SE_2 + SE_{2bis2} + SE_3 + SE_5. \quad (5)$$

Quand Alice arrive chez elle, son PAN (6) indique qu'elle a son ordinateur fixe disponible.

$$PAN_{Maison} = \{PDA, \text{Ordinateur fixe}\}. \quad (6)$$

Grâce à la connexion ADSL (AN3), le fournisseur de ce type de réseau permet l'utilisateur d'avoir plus de possibilité au niveau d'un service dédié sur les différentes plates-formes. Les communautés de services sont donc automatiquement et dynamiquement construites :

$$\text{Communauté de Service (email)} = \{SE_{2bis1}, SE_{2bis2}, SE_{2bis3}\} \quad (7)$$

$$\text{Communauté de Service (SMS)} = \{SE_{21}, SE_{22}\}$$

L'arrivée à la maison notifie également un changement au niveau de la Logique de Service, Alice n'a plus besoin du service de SMS → Vocal car elle peut voir les messages elle-même maintenant. Grâce à son PAN(6), l'ordinateur fixe est choisi car il a une meilleure qualité d'affichage (SE6), découverte par le DDBS. La modification se fait non seulement au niveau du terminal mais aussi au niveau du transport et du service car les anciens composants ne conviennent plus pour la QoS de bout en bout demandé par la session « User Centric ». Le Binding se fait par le « NGN Sessionware » et nous obtenons la session actuelle (Figure V.31 - 3) :

$$LS_4 = SE_1 + SE_{22} + SE_{2bis3} + SE_6. \quad (8)$$

Dans ce scénario, nous nous apercevons que la communauté des équipements est mise à jour en fonction de la localisation et de l'activité (1)(3)(6). Les communautés de services (4) (7) s'auto gèrent également pour soutenir leur propre VPXN. La fourniture de service est toujours garantie d'être délivrée en continue grâce au « NGN Sessionware » qui offre le Binding cross layer.

Nous concluons avec les événements déclencheurs et les réactions dans notre système. Nous obtenons la table de décision comme suit :

Cette table est la base essentielle pour notre déploiement de la map fonctionnelle.

Acteur Événement	Communauté des utilisateurs	Communauté des équipements	Communauté des services	VPXN	Sessionware
Changement de localisation		DLBS → Nouvelle Location L		logique de service LS={ $F_n n \in N$ }	QoS de bout en bout Q_{E2E} Préférence de l'utilisateur P
Changement de l'activité	UABS → Nouvelle activité A			logique de service LS={ $F_n n \in N$ }	QoS de bout en bout Q_{E2E} Préférence de l'utilisateur P
Changement de la QoS d'un composant		Device profile → EE (F_m, QoS_x)	Service profile → ES (F_m, QoS_x)	logique de service LS={ $F_n n \in N$ }	QoS de bout en bout Q_{E2E} Préférence de l'utilisateur P
$LS = \{ F_n n \in N \}, Q_{E2E}$		EQoS management → CE _m (F_m, QoS_{seuil})	SQoS management → CS _m (F_m, QoS_{seuil})		
L, A, CE/CS_m (F_m, QoS_{seuil})	UPBS → CU (L/A)	DPBS → CE _m ($F_m, QoS_{offerte}$)	SPBS → CS_m ($F_m, QoS_{offerte}$)		
EE/ES (F_x, QoS_x), CE/S _m (F_m, QoS_{seuil})		DQoS continuity → CE _m ($F_m, QoS_{offerte}$)	SQoS continuity → CS _m ($F_m, QoS_{offerte}$)		
CE/CS_m ($F_m, QoS_{offerte}$), P		DSelecting → EE _m ($F_m, QoS_{offerte}$)	SSelecting → ES_m ($F_m, QoS_{offerte}$)		
EE/ES _m = null		DDBS (EE _m) → CE _m ($F_m, QoS_{offerte}$)	SDBS (ES _m) → CS _m ($F_m, QoS_{offerte}$)		
EE/ES _m != null				Mise à jour	
Modification de VPXN					Binding

Tableau V-4 : La table de décision

V.6. Conclusion

Dans ce chapitre, nous avons consigné nos propositions pour assurer la continuité de service dans un contexte mobile. Un ensemble de « services supports » a été préconisé.

Face au besoin de NGS, nous proposons d'abord une composition de service pour satisfaire tous les critères demandés : interopérabilité, réutilisabilité, mutualisation, et autonomie (l'autogestion). Tous ces composants sont enchaînés dynamiquement dans un VPSN qui est l'application demandée par l'utilisateur. Grâce à nos méta modèle et modèle référentiel, le réseau de connectivité et le groupe des équipements utilisé par la même application orientée utilisateur sont aussi représentés par le VPCN et le VPEN. Nous proposons ensuite la gestion de la mobilité par la construction dynamique de communautés virtuelles qui sont en fait le support de l'autogestion de chaque VPXN. La communauté peut se construire et se maintenir selon différents critères en temps réel. C'est notre proposition de « sessionware » qui nous permet d'associer les VPXNs par un Binding pour avoir une session temporelle orientée utilisateur, la fourniture de la continuité de l'application est assurée par un reBinding dynamique des résultats de niveaux architecturaux en tenant compte des préférences de l'utilisateur.

De plus, nous mettons la QoS en tant que critère important de l'autogestion de la communauté, la QoS de bout en bout de la session est ainsi respectée grâce à sa modification du type MBB.

Nous proposons de rassembler toutes nos propositions dans un scénario final pour mieux les comprendre. Ce dernier est celui que nous réaliserons dans nos implémentations.

V.7. Références

- [1] Sasan ROSTAM BEIKTAFRESHI, Thèse de doctorat ENST, Paris: Userware: de l'intégration de l'usage à l'interfonctionnement des services, 2006
- [2] Projet VTC2E,
- [3] ISO 20000 / ITIL International Standard
- [4] Helen Fulcher, THE CONCEPT OF COMMUNITY OF INTEREST, 1991

116 Vers l'autogestion pour une continuité de service intégrée et « sans couture»

Chapitre VI L'implémentation

Pour évaluer la faisabilité de nos propositions, nous allons d'abord, examiner le concept de la communauté virtuelle à travers un prototype (§VI.1) en profitant du mécanisme de Peer-to-Peer offert par JXTA. Ensuite, nous testons la réalisation et la gestion de la session de services dans un environnement basé sur IMS (IP Multimedia Subsystem) (§VI.2).

VI.1 Prototype

Nous avons proposé de gérer les compositions de services à travers des communautés virtuelles. Dans chacune des communautés, nous introduisons les services de base génériques pour recevoir les événements déclenchés par les différentes mobilités. Nous profitons du mécanisme de Peer-to-Peer proposé par JXTA (§VI.1.1) pour organiser et gérer automatiquement et dynamiquement les composants de services grâce aux fonctions simulées (§VI.1.2). Une conclusion (§VI.1.4) est présentée pour tenir compte ce qui a testé pour ce prototype.

VI.1.1 Plate-forme JXTA

JXTA [1] est un projet Open Source lancé par Sun Microsystems en avril 2001. Le but de JXTA est de pouvoir interconnecter n'importe quel système sur n'importe quel réseau avec un mécanisme peer-to-peer basé sur JAVA. Il permet de créer un réseau au-dessus des autres. JXTA crée un réseau virtuel ou Groupe de pairs, dont les membres (Peers) peuvent communiquer.

Les groupes de pairs peuvent se chevaucher sans restriction, en d'autres termes, les pairs peuvent appartenir à autant de groupes de pairs que besoin. Les spécifications de JXTA n'imposent pas la façon de former des groupes de pairs. Dans un réseau JXTA, un groupe de pairs est une collection de pairs qui partagent des ressources et des services. En accord avec la philosophie de conception de JXTA, un groupe de pairs est spécifié pour être sans contrainte et aussi générique que possible.

JXTA spécifie seulement le minimum nécessaire pour maintenir un groupe, sans indiquer comment il le fait. Cette gestion du groupe est une partie du cœur des services de JXTA, mais elle pourrait prendre de nombreuses formes - elle pourrait être soit reposée sur une base de données ou un service répertoire, par exemple, et pourrait être basée sur une application centralisée ou distribuée.

Les services rendus par la gestion de ce Peer Group sont les suivants :

- Diffusion et découverte des ressources par les pairs, PeerGroup, Service et diffusion par tuyaux
- communiquer les uns avec les autres par les tuyaux
- coopérer les uns avec les autres dynamiquement pour construire des groupes de pairs

Les protocoles du projet JXTA (Figure VI.1) sont au nombre de six et sont divisés en deux catégories:

- Core Specification Protocols
- Standard Service Protocols

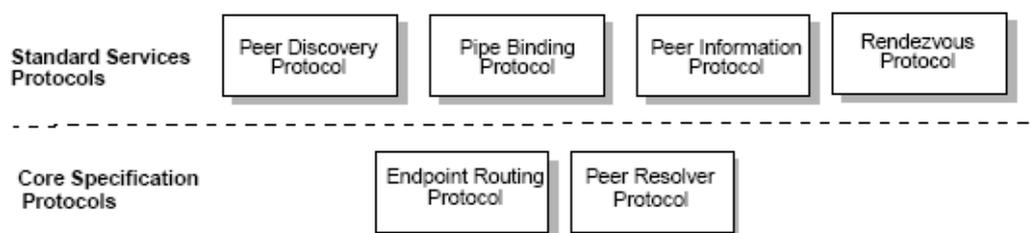


Figure VI.1 : Les protocoles JXTA

Les protocoles JXTA ne spécifient pas comment la recherche des annonces est effectuée mais fournissent un “resolver” générique avec une politique par défaut qui peut être réécrite. Les développeurs peuvent implémenter le “resolver” de façon centralisée, décentralisée ou hybride suivant les besoins de leurs applications. L’infrastructure « resolver » qui est au cœur de JXTA permet d’envoyer et de propager les requêtes et de recevoir les réponses. Elle permet l’authentification et le rejet des messages non valides.

La politique par défaut de JXTA est basée sur les *Rendezvous* super-peers. Les Rendezvous sont des pairs qui ont accepté de stocker les « *advertisement indices* » (c’est à dire des pointeurs vers les « *edge peers* » qui possèdent l’annonce correspondante). Ces Rendezvous correspondent conceptuellement à des localisations bien connues utilisées pour indexer et localiser les annonces.

Les “edge peers” maintiennent une relation spéciale avec leurs pairs rendezvous. N’importe quel pair peut devenir un rendezvous s’il a les bonnes capacités. Les peergroups sécurisés peuvent restreindre le choix d’edge peers qui peuvent se comporter en rendezvous.

Un changement fondamental dans JXTA 2.0 est que les rendezvous ne stockent plus les annonces des “edge peers”. Ils maintiennent un index concernant les annonces publiées par leurs « *edge peers* ». Ceci permet à l’architecture rendezvous d’être plus scalable et réduit le problème de la mémorisation des annonces qui ont expiré. Le service *Shared Resource Distributed Index (SRDI)* est utilisé par les “edge peers” pour indexer leurs annonces sur les rendezvous.

Les “edge peers” utilisent SRDI pour envoyer les index des annonces à leur rendezvous à la publication de nouvelles annonces.

Quand une requête d'annonce est issue du pair A concernant une annonce stockée sur le pair B, la requête est envoyée au pair rendez vous de A, RDv1. ce dernier va chercher s'il possède un index de cette annonce. S'il n'en trouve pas, il propage la requête au rendezvous suivant RDv2. Quand la requête atteint RDv2, ce dernier trouve l'index de l'annonce et achemine la requête à B. ceci permet de s'assurer que la dernière copie de l'annonce stockée en B est envoyée au pair A.

Seuls les rendezvous peers sont impliqués dans la propagation des requêtes. Une requête n'est acheminée vers un edge peer que si un index correspondant a été trouvé. Ceci diminue de manière significative le trafic réseau résultant des recherches d'annonces.

VI.1.2 Fonctions simulées

Tous les dysfonctionnements de la QoS en fonction de différentes mobilités sont considérés comme les événements qui lanceront une gestion d'un réseau VPXN (X=Utilisateur, Service, Connectivité, Equipement) qui représente les enchaînements de composants de même natures. Nous rappelons que les communautés d'intérêt ont pour l'objectif la continuité de service, et sont proposées comme les « services supports » de chacun de réseau virtuel (VPXN). Les services de base sont des fonctions qui feront les mises à jour des membres de la communauté. Nous allons mettre en oeuvre les cas d'usage dans la plate forme de JXTA pour montrer comment ces services de base peuvent être utilisés afin d'avoir une gestion des communautés qui réponde aux mobilités.

Dans JXTA, il existe plusieurs types de pairs. Dans notre cas, les éléments des services sont des pairs en bordure, qui mettent en oeuvre les fonctionnalités désirées. Le pair bordure publie une annonce de sa description de service sous la forme XML qui a pour objectif de décrire les méta-données. Le pair de Rendezvous maintient l'index de tous ceux qui ont enregistré. Ce type de pairs peut aider à découvrir et à publier les annonces et les itinéraires, de maintenir une carte de la topologie des infrastructures d'autres pairs, qui ensuite utilisée pour le contrôle de propagation, et l'entretien de la table distribuée de « hach ».

Nous allons utiliser deux types des annonces dans notre prototype :

- Annonce par les pairs: elle donne le cahier des charges du service fourni par l'ES.
- Annonce par le groupe : elle décrit un groupe de pairs tels que CVS ou CVE.

Nous allons d'abord créer une communauté virtuelle dans la couche de service, qui contient des éléments de services fonctionnellement équivalent. Quand une QoS demandée arrive, le CVS filtre ses ESs par le SLA de l'utilisateur et renvoie les ESs correspondant à l'exigence. ES publie une annonce (Figure VI.2) y compris ses caractéristiques, telles que la fonction fournie, la qualité de service actuelle, l'emplacement actuel.

```

<?xml version="1.0"?>
<!DOCTYPE jxta:PeerAdvertisement >
<jxta:PeerAdvertisement xmlns:jxta="http://jxta.org">
<Id>urn:jxta:uuid-...</Id>
<Type>JxtaUnicast</Type>
<Name>A</Name>
<Layer>Service</Layer>
<Function>Fun1</Function>
<QoS>
  <Delay>1000ms</Delay>
  <Reliability>98%</Reliability >
  <Availability>99%</Availability >
  <Capacity>5000</Capacity>
</QoS>
<Location>LogicAddress</Location>
</jxta:PeerAdvertisement>

```

Figure VI.2 : Annonce de pair A

Tous les pairs qui ont pour fonction nommée 'Fun1' vont publier leurs annonces au même Rendez-vous, en se fondant sur la CVS construite. Quand une CVS reçoit une QoS demandée, elle choisira ceux qui répondent au SLA. Ce travail peut être accompli par la fonction de découverte fournie par JXTA.

Nous allons maintenant examiner le support de la mobilité de terminal. Le EE correspondant fera annonce (Figure VI.3) à son CVE, qui va publier une annonce du groupe des pairs à la CVS correspondante pour informer qu'un filtrage doit s'appliquer pour le changement de localisation.

```

<?xml version="1.0"?>
<!DOCTYPE jxta:PeerGroupAdvertisement >
<jxta:PeerGroupAdvertisement xmlns:jxta="http://jxta.org">
<Id>urn:jxta:uuid-...</Id>
<Type>JxtaUnicast</Type>
<Name>VE|C1</Name>
<Layer>Terminal</Layer>
<Filter>
  <Name>Location</Name>
  <Value>Domain1</Value>
</Filter>
<Node>
  ...
</Node>
</jxta:PeerGroupAdvertisement>

```

Figure VI.3:Annonce de la Communauté des équipements

Lorsque le CVS reçoit la nouvelle localisation, elle invoquera le service de base de présence pour trouver les ESs géographiquement accessibles autour de l'utilisateur. Elle doit choisir également un ES approprié afin de mieux répondre à l'utilisateur de la SLA. Pour un élément choisi, nous devons comparer (Figure V.21) la QoS (a, A, b) avec la QoS (a', A', b'). Elle publie donc une annonce du groupe de pairs (Figure VI.4) à la CVR (Communauté virtuelle de réseaux) correspondant au nœud précédent, le nœud candidat et le nœud prochain dans le but d'interroger la qualité de service de chaque lien. La CVR va calculer la qualité de service demandée en utilisant sa table de routage et la logique de services, et retourner les résultats à la CVS. La CVS va décider quel nœud remplacer en fonction de la comparaison.

```

<?xml version="1.0"?>
<!DOCTYPE jxta:PeerGroupAdvertisement >
<jxta:PeerGroupAdvertisement xmlns:jxta="http://jxta.org">
<Id>urn:jxta:uuid-...</Id>
<Type>JxtaUnicast</Type>
<Name>VSC1</Name>
<Layer>Service</Layer>
<Function>Fun1</Function>
<Node>
  <ID>urn:jxta:uuid-...</ID>
  <Name>A</Name>
</Node>
<Node>
  <ID>urn:jxta:uuid-...</ID>
  <Name>A'</Name>
</Node>
<Node>
  ...
</Node>
<PreNode>M</PreNode>
<NextNode>N</NextNode>
</jxta:PeerGroupAdvertisement>

```

Figure VI.4: Annonce de la Communauté des services

Nous pouvons voir que les CVs dans chaque couche peuvent être construites et peuvent communiquer les unes avec les autres par l'annonce publicitaire par le biais d'un canal P2P. Cette chaîne virtuelle peut être créée dans une couche, ou au travers des couches.

VI.1.3 Scénarii simulés

Nous allons essayer de situer les fonctions simulées dans les scénarii plus précis suivant la construction d'une communauté des équipements (), les différents types de mobilités.

Scénario 1 : Création d'une Communauté

Dans ce scénario, nous supposons que l'utilisateur entre son bureau, et son PANBureau est construit selon le profil de l'utilisateur. Chaque terminal dans le PAN possède un profil d'équipement qui le décrit par le maintien des caractères de l'appareil. Par exemple, un ordinateur portable et un téléphone portable sont disponibles pour l'utilisateur actuel (Figure VI.5).

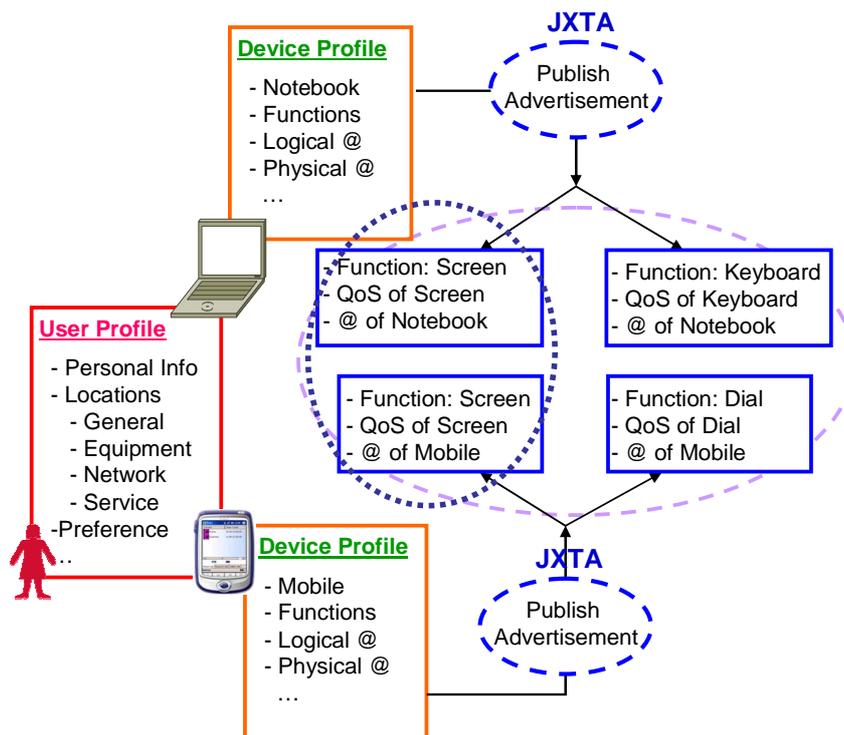


Figure VI.5: L'avisement publié par un EE

Chaque équipement publie un avisement afin de faire valoir ses fonctions et des caractères correspondants dans le réseau. Pour la publication et détection d'avisement, on importe des packages de JXTA (Figure VI.6).

```
import net.jxta.pipe.PipeMsgEvent;
import net.jxta.pipe.PipeMsgListener;
import net.jxta.pipe.PipeService;
import net.jxta.discovery.DiscoveryEvent;
import net.jxta.discovery.DiscoveryListener;
import net.jxta.discovery.DiscoveryService;
```

Figure VI.6: Packages importés de JXTA

Figure VI.7 montre un paragraphe de code faux. La méthode « run » publie l'avisement du mobile, alors que la méthode « discoveryEvent » écoute des événements d'arriver de l'avisement à partir de l'autres. Car les équipements travaillent de façon P2P, ils peuvent échanger des informations entre eux, ce qui signifie que chacun peut recevoir les avisements des autres.

```

Class MobilePeer {
  public void run ()
  {...
    net.jxta.discovery.DiscoveryService.publish
      (Advertisement adv)
    net.jxta.pipe.pipeService.createInputPipe
      (Advertisement adv, PipeMsgListener listener);
    Thread.sleep(time);
    ...
  }
  public void discoveryEvent (DiscoveryEvent event)
  {...
    updateCommunityProfile() {...}
    ...
  }
}

```

Figure VI.7: Code faux d'un pair

Alors que trois VECs peuvent être construits (Figure VI.8). Une VEC offrant la fonction de l'écran inclut deux EEs : le lap top et le mobile ; la deuxième soutenant la fonction du clavier inclut un EE : le lap top ; la dernière incluant le mobile offre la fonction de l'appel. Chaque EE tient une copie du profil de communauté de chacune VEC qu'il appartient.

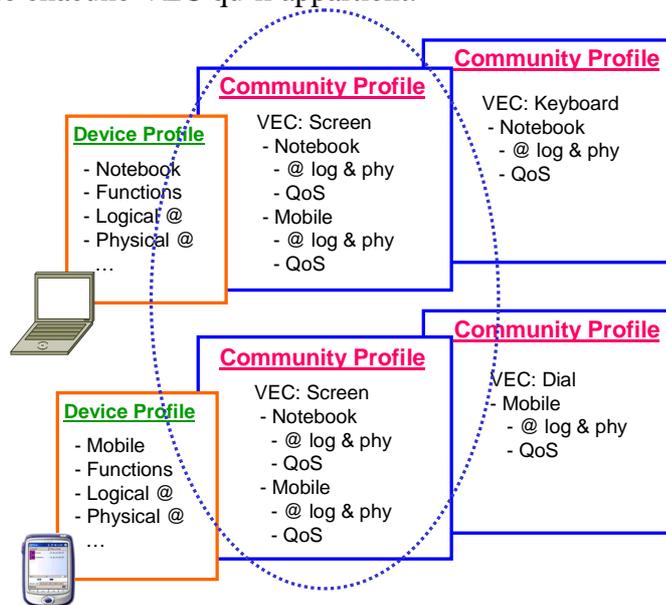


Figure VI.8: Création de VEC

Nous avons relié chacune des mobilités à un niveau de notre architecture (Utilisateur, Service, Réseau et Equipement), en leur associant une communauté virtuelle autogérée à travers un protocole Peer-to-Peer. On pourrait construire de même façon l'autre VC sur chaque niveau avec l'aide de JXTA.

Scénario2 : Mobilité de l'utilisateur

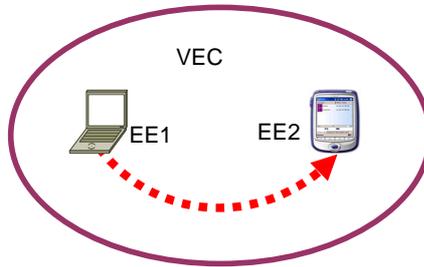


Figure VI.9: Mobilité de l'utilisateur

Quand l'utilisateur change son terminal au cours d'un service, la mobilité de l'utilisateur induit des changements des besoins du NGN. Dans ce scénario, on concerne la situation que la VEC ne change pas. Ça veut dire que l'utilisateur est toujours dans son bureau (Figure VI.9). Nous obtenons la table de décision qui précise les réactions d'après des événements (Tableau VI-1).

Acteur / Événement	Équipement (EE 1)	Équipement (EE 2)	Communauté des équipements	Communauté des services
Changement de l'équipement	Device profile → Mise à jour l'état	Device profile → Mise à jour l'état	--	--

Tableau VI-1: Table de décision de la mobilité de l'utilisateur

Quand un changement de l'équipement se passe, les deux équipements font la mise à jour l'état de leur propre profil. Le lap top est « inactive » maintenant, alors que le mobile est « active » pour la même fonction.

Description / Classe	Dérivé de	Méthode	Input	Output	Action
EE	Composant de base	LBS	@ Logique de la Location L	@ Physique de la Location L	
Device Profile	Profile	Update	l'état		Mise à jour l'état de certaine fonction

Tableau VI-2: Table de classe de la mobilité de l'utilisateur

Par conséquent, nous accomplissons la table de classe (Tableau VI-2) à des fins de l'implémentation de la mobilité de l'utilisateur.

Scénario 3 : Mobilité de terminal (service)

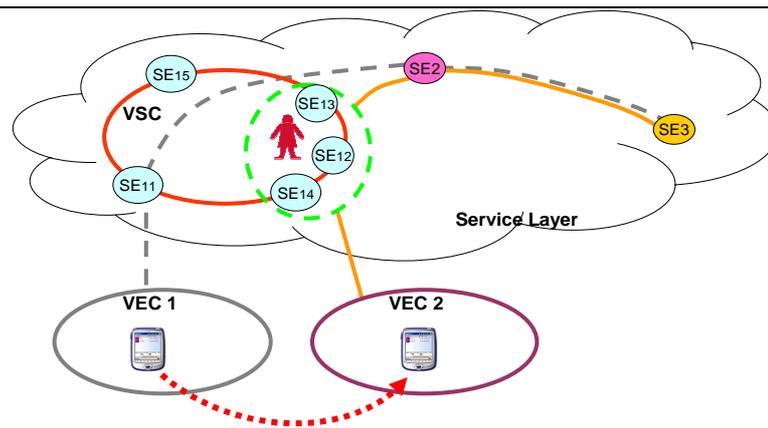


Figure VI.10: Mobilité de terminal

Dans ce scénario, on suppose que dans la couche, une VSC est déjà construite, et l'utilisateur déplace avec son mobile de la VEC 1 à la VEC 2, ce qui on l'appel la mobilité de terminal (Figure VI.10). Chaque SE publie un avertissement à intervalles réguliers, comme ce que l'EE fait dans Figure VI.5. Cet avertissement est diffusé par JXTA en tant que fichier XML (Figure VI.11).

```
<?xml version="1.0"?>
<!DOCTYPE jxta:PeerAdvertisement >
<jxta:PeerAdvertisement xmlns:jxta="http://jxta.org">
<Id>urn:jxta:uuid-280984B767FF4B80980DE4586287229305</Id>
<Type>JxtaUnicast</Type>
<Name>SE12</Name>
<Layer>Service</Layer>
<Function>SendVoiceMessage</Function>
  <Delay>1000ms</Delay>
  <Reliability>98%</Reliability >
  <Availability>99%</Availability >
  <Capacity>300</Capacity>
<Location>L</Location>
</jxta:PeerAdvertisement>
```

Figure VI.11: Avertissement de SE

Dans ce cas-là, nous mettons en œuvre la gestion dynamique des éléments de la VSC existante. L'utilisateur dans son bureau est en train d'envoyer un message vocal par son mobile. Il sort de son bureau sans d'arrêter l'émission. Le mobile, en tant qu'un EE, informe le réseau entier sa nouvelle location « L » et met à jour son propre profil.

La relation entre les événements et les actions est exprimée dans Tableau VI-3. Lorsque le VSC est conscient de la mobilité, il demande PBS pour trouver les SEs disponibles autour de l'utilisateur. PBS est mise en œuvre par le Peer Discovery Service (PDS) fourni par JXTA, qui réalise la recherche rapide au sein d'un peer groupe. Dans notre cas, le input est « Location=L », et le output est un sous-ensemble de VSC : {SE12, SE13, SE14}.

Acteur / Événement	Équipement	Communauté des équipements (VEC 1)	Communauté des équipements (VEC 2)	Communauté des services (trans-organisationnelle)
Changement de localisation	LBS → @Physique de la Nouvelle Location L Envoyer L			
L	Device profile → Mise à jour @Physique de l'équipement et VEC	Community profile → Supprimer l'équipement	Community profile → Ajouter l'équipement	PBS (L) → CS _m (F _m , QoS _{offerte}) If PBS fail: DBS (ES _m) → CS _m (F _m , QoS _{offerte})

Tableau VI-3: Table de décision de la mobilité de terminal

Si le résultat de PBS est nulle, DBS est invoqué à effectuer des recherches des SEs fonctionnellement équivalents et QoS équivalente dehors la VSC existante. DBS est également implémenté par le PDS, alors que la condition de la découverte est la fonction et la QoS demandées.

Description / Classe	Dérivé de	Méthode	Input	Output	Action
EE	Composant de base	LBS	@ Logique de la Location L	@ Physique de la Location L	
		SendLocation	@ Physique de la Location L		Envoyer @ Physique de la Location L au réseau
Device Profile	Profile	Update	@ Physique de la Location L; ID de nouvelle VEC		Mise à jour @ Physique de l'équipement et VEC
Community Profile de VEC	Profile	Delete	EE		Supprimer l'équipement
		Add	EE		Ajouter l'équipement
VSC	VC	PBS	L	CS _m (F _m , QoS _{offerte})	
		DBS	ES _m	CS _m (F _m , QoS _{offerte})	

Tableau VI-4: Table de classe de la mobilité de terminal

D'après la table de décision, on conçoit la table de classe (Tableau VI-4) et implémente les méthodes correspondantes. Figure VI.12 montre un exemple de l'usage de service de découverte afin de PBS.

```

discovery.getRemoteAdvertisements(
    null, // No specific peer
    DiscoveryService.PEER, // Type is peer
    "Location", "L",
    5, // Max result is 5
    this); // Query listener is the current class
    
```

Figure VI.12 L'usage de service de découverte de PBS

VI.1.4 Conclusion

Le prototype que nous venons d'implémenter confirme la faisabilité de VPSN basé sur la gestion de communauté virtuelle. Une partie d'évaluation [2] est faite en reliant le protocole SIP. Nous percevons que avec la gestion de communauté, le VPSN est construit plus efficacement.

VI.2 Environnement IMS+

Afin de mettre en exergue les avantages de nos propositions sur la session et l'architecture de service, nous avons travaillé sur une autre plate-forme, celle supportant le système IMS. Nous allons d'abord consigner l'architecture d'IMS (§VI.2.1), ses entités (§VI.2.2), sa structure de service (§VI.2.3), ainsi que les protocoles et les interfaces (§VI.2.4) pour ensuite positionner nos propositions (§VI.2.5). La plate-forme (§VI.2.6) utilisée pour les tests est celle proposée par FOKUS. Finalement, nous concluons les résultats obtenus.

VI.2.1 Architecture de l'IMS

L'introduction de l'IMS [3][4] dans les réseaux fixe et mobile représente un changement fondamental dans les réseaux de télécommunication de type voix. Les nouvelles capacités des réseaux et des terminaux, le mariage entre l'Internet et la voix, le contenu et la mobilité donnent naissance à des nouveaux modèles de réseaux et surtout offrent un formidable potentiel pour développer de nouveaux services. Dans cet objectif, l'IMS est conçu pour offrir aux utilisateurs la possibilité d'établir des sessions multimédia en utilisant tout accès haut débit et une commutation de paquets IP.

Un ensemble de besoins ont été défini lors de la conception de l'IMS :

- **Indépendance par rapport à l'accès** : L'IMS a été conçu pour être indépendant de l'accès afin que les services IMS puissent être fournis à partir de n'importe quel type d'accès connecté à un réseau IP (e.g. GPRS, UMTS, WLAN, xDSL, câble, etc).
- **Garantie QoS des services multimédia** : Sur Internet, le type de QoS fourni est best effort. Cela ne sera pas le cas avec l'IMS. Les réseaux d'accès et de transport de l'IMS fournissent la QoS de bout-en-bout. A travers l'IMS, le terminal négocie ses capacités et exprime ses exigences de QoS durant la phase d'établissement de la session avec le protocole SIP. En parallèle le terminal réserve les ressources nécessaires dans le réseau d'accès en utilisant un protocole de réseau de ressources (e.g., RSVP, SM/GTP1, etc).
- **Support des mobilités**: L'utilisateur peut accéder à ses services IMS depuis n'importe quel réseau IMS visité. La mobilité de l'utilisateur sans coupure et de ses services sont pris en compte.
- **Interfonctionnement avec d'autres réseaux** : L'IMS ne sera pas déployé partout au même moment. Il est donc nécessaire de prévoir des passerelles entre les réseaux RTC/GSM et le réseau IMS.
- **Contrôle de service** : L'IMS fournit tous les éléments permettant de connaître les services souscrits par l'abonné et de les invoquer pour toute session sortante ou entrante.
- **Développement de service** : L'IMS fournit les APIs permettant le développement de services multimédia. Parmi les APIs déjà considérées figurent la présence, la messagerie instantanée, le push to talk, la conférence et le « chat ».

L'IMS fournit un réseau IP multiservice, multi-accès:

- Multiservices : tout type de services délivrés par un réseau cœur supportant différents niveaux de QoS pourront être offerts à l'utilisateur.
- Multi-accès: Tout réseau d'accès large bande, fixe et mobile pourra s'interfacer à l'IMS.

L'IMS est un « enabler » pour les fournisseurs de service afin d'offrir :

- Des services de communication non temps-réel, pseudo temps-réel et temps réel suivant une configuration client-serveur ou entre entités paires
- La mobilité des services / Mobilité de l'utilisateur sans coupure
- Plusieurs sessions et services simultanément sur la même connexion réseau

L'architecture IMS (Figure VI.13) peut être structurée en couches. Quatre couches importantes sont identifiées :

- La couche **ACCES** peut représenter tout accès haut débit tel que : **UTRAN** (UMTS Terrestrial Radio Access Network), **CDMA2000** (technologie d'accès large bande utilisée dans les réseaux mobiles aux Etats-Unis), **xDSL**, **réseau câble**, **Wireless IP**, **WiFi**, etc.
- La couche **TRANSPORT** représente un réseau IP. Ce réseau IP pourra intégrer des mécanismes de QoS avec MPLS, Diffserv, RSVP, etc. La couche transport consiste donc en des routeurs (edge router à l'accès et en core router en transit) reliés par un réseau de transmission. Différentes piles de transmission peuvent être considérées pour le réseau IP: IP/ATM/SDH, IP/Ethernet, IP/SDH, etc.
- La couche **CONTROLE** consiste en des contrôleurs de session responsables du routage de la signalisation entre usagers et de l'invocation des services. Ces nœuds s'appellent des CSCF (Call State Control Function). L'IMS introduit donc un environnement de contrôle de session sur le domaine paquet.
- La couche **APPLICATION** introduit les applications (services à valeur ajoutée) proposées aux usagers. L'opérateur peut se positionner grâce à sa couche CONTRÔLE en tant qu'agrégateur de services offerts par l'opérateur lui-même ou par des tiers. La couche application consiste en des serveurs d'application (AS, Application Server) et des MRF (Multimedia resource function) que les fournisseurs appellent serveurs de média IP (IP MS, IP Media Server).

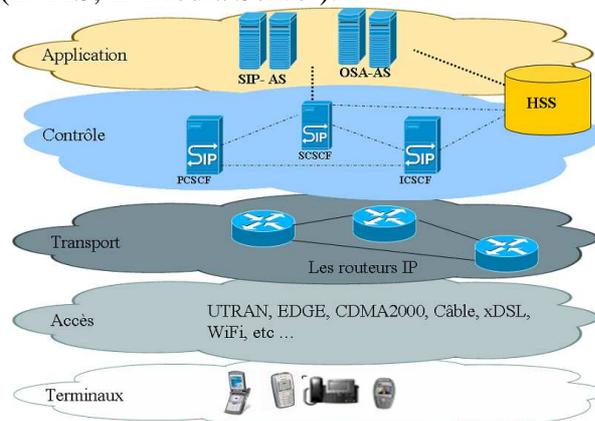


Figure VI.13 : Architecture en couche du Réseau IMS

VI.2.2 Les entités du Réseau IMS

Dans cette partie on présente les différentes entités [3][4] constituant le réseau IMS pour pouvoir comprendre par la suite comment ces entités inter fonctionnent afin de réaliser la session de service dans l'IMS.

- Terminal IMS

Il s'agit d'une application sur un équipement de l'utilisateur qui émet et reçoit des requêtes SIP. Il se matérialise par un logiciel installé sur un PC, sur un téléphone IP ou sur une station mobile UMTS (UE, User Equipment).

- HSS (Home Subscriber Server)

L'entité HSS (Home Subscriber Server) est la principale base de stockage des données des usagers et des services auxquels ils ont souscrit. Les principales données stockées nécessaires à un utilisateur pour établir une session multimédia sont ses identités privées et publiques, ses informations de sécurité (son vecteur d'authentification), son état d'enregistrement, l'adresse du SCSCF qui lui a été alloué, sa localisation et les informations permettant l'invocation de ses services (son profil de service). L'entité HSS interagit avec les entités du réseau à travers le protocole Diameter et est en fait une évolution du HLR dans le réseau GSM.

- Call Session Control Function (CSCF)

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 : PCSCF (Proxy CSCF), I-CSCF (Interrogating CSCF) et S-CSCF (Serving-CSCF).

- P-CSCF (Proxy Call Session Control Function)

Le PCSCF est le premier point de contact entre les terminaux et le réseau IMS. L'adresse du PCSCF est découverte par le terminal lors de l'activation d'un contexte PDP pour l'échange de messages de signalisation SIP. Il est donc alloué à un utilisateur IMS au moment de l'enregistrement et ne change pas durant toute la période d'enregistrement. Toute la signalisation SIP du UE et vers le UE passe via le P-CSCF. Le P-CSCF se comporte comme un Proxy Server SIP lorsqu'il relaye les messages SIP vers le destinataire approprié et comme un User Agent SIP lorsqu'il termine l'appel (e.g. suite à une erreur dans le message SIP reçu).

Les fonctions réalisées par l'entité P-CSCF comprennent :

- L'acheminement de la méthode SIP REGISTER émise par le terminal à l'entité I-CSCF à partir du nom du domaine nominal.
- L'acheminement des méthodes SIP émises par le terminal au S-CSCF dont le nom a été obtenu dans la réponse à la procédure d'enregistrement.
- Le contrôle d'admission des appels
- Le routage des méthodes SIP ou réponses SIP au terminal.
- La compression / décompression des messages SIP.

Un réseau IMS peut inclure plusieurs PCSCF pour assurer une scalability et une redondance. Chaque PCSCF peut servir un certain nombre de terminaux IMS dépendamment de la capacité du nœud. Le PCSCF est toujours localisé avec le GGSN dans le réseau mère ou visité.

- I-CSCF (Interrogating CSCF)

L'ICSCF est un SIP proxy constituant le point d'entrée au réseau IMS pour les réseaux tiers IMS. Il s'interface avec le SLF et le HSS en utilisant le protocole DIAMETER et se sert des informations sur la localisation de l'utilisateur pour router les requêtes SIP vers le S-CSCF adéquat. Un réseau IMS peut inclure plusieurs ICSCF pour assurer une scalabilité et une redondance. L'ICSCF est localisé dans le réseau mère.

- S-CSCF (Serving CSCF)

Le SCSCF est le nœud central du plan de signalisation qui assure principalement l'enregistrement et l'authentification des utilisateurs. En plus de son rôle de serveur SIP, le SCSCF fait le contrôle de session et joue le rôle du SIP registrar en maintenant une association entre la localisation de l'utilisateur (adresse IP) et l'identité publique de l'utilisateur PUI (Public User Identity). Il implémente une interface DIAMETER avec le HSS pour télécharger les vecteurs d'authentification de l'utilisateur qui veut accéder à l'IMS et pour informer le HSS des terminaux qu'ils lui ont été alloués. En plus, il télécharge du HSS le profil de service de l'utilisateur constitué d'une série d'initial filter criteria (IFC) et envoie les requêtes SIP satisfaisant ces critères de déclenchement de service vers les serveurs d'applications correspondant pour offrir le service demandé. Le S-CSCF est localisé toujours dans le réseau mère et peut servir plusieurs terminaux IMS selon la capacité du nœud.

VI.2.3 L'architecture de service en IMS

L'architecture de service IMS [4] consiste en un ensemble de serveurs d'application interagissant avec le réseau IMS (i.e., S-CSCF) à travers l'interface ISC (IP Multimedia Service Control) supportée par le protocole SIP (Figure VI.14). Le serveur d'appel SIP appelé S-CSCF (Serving - Call State Control Function) joue le rôle de point depuis lequel un service peut être invoqué. Il dispose du profil de service de l'abonné qui lui indique les services souscrits par l'abonné et sous quelle condition invoquer ces services.

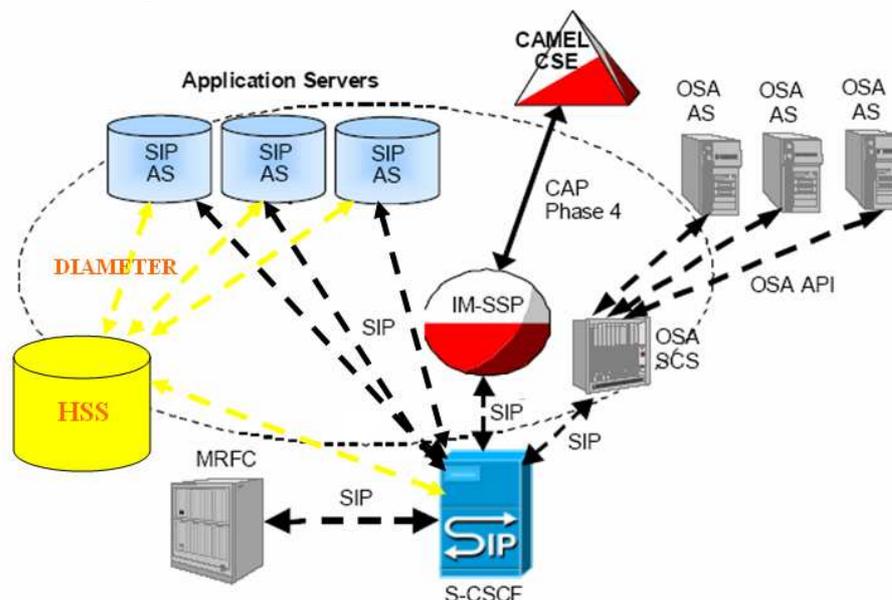


Figure VI.14 : Architecture de la plateforme de services

- Entités de l'architecture de service IMS

Les serveurs d'application constituant la plate-forme de service en IMS peuvent se trouver sous différents types :

- Les serveurs d'application SIP (SIP AS) qui hébergent et exécutent des services IP multimedia à valeur ajoutée basé sur SIP (e.g., Push To Talk, Présence, Prépaïd, Instant messaging, vidéoconférence, Unified messaging, etc.) et qui peuvent influencer le déroulement de la session à la demande du service. Les AS SIP peuvent opérer en SIP proxy mode: SIP proxy server, SIP redirect server, SIP User Agent, SIP B2B UA afin d'exécuter des services.
- La passerelle OSA (OSA SCS, OSA Service Capability Server) qui est un type particulier de serveur d'application SIP qui termine la signalisation SIP sur l'interface ISC et qui interagit avec des serveurs d'application OSA en utilisant l'API OSA vers le OSA-AS. Elle permet à des serveurs d'application tiers d'accéder au réseau IMS d'une façon sécurisée pour fournir des services aux utilisateurs.
- L'IP Multimedia Service Switching Function (IM-SSF) qui est un point de commutation au service et un type particulier de serveur d'application SIP qui termine la signalisation SIP sur l'interface ISC d'une part et qui joue le rôle de SSP RI/CAMEL d'autre part (i.e., il dispose des modèles d'appel O-IM-BCSM et T-IM-BCSM, des points de détection RI/CAMEL et du protocole INAP/CAP) pour interagir avec les SCP RI/CSE CAMEL. Ce serveur d'application permet de réutiliser les services CAMEL, développés pour le réseau GSM, dans l'IMS.

Ces trois types de serveur d'application se comportent comme des serveurs SIP par rapport au réseau IMS. En plus de l'interface SIP, l'AS peut aussi avoir optionnellement une interface avec le HSS basée sur le protocole DIAMETER. Cette interface est utilisée pour télécharger/mettre à jour des données du/dans le HSS liées à un utilisateur. L'AS peut être localisé ou bien dans le réseau mère ou alors dans un réseau tiers avec lequel le réseau mère maintient un accord de service. Mais dans le cas où l'AS est à l'extérieur du réseau mère, il n'aura pas d'interface avec le HSS.

Un type spécialisé de serveur d'application SIP appelé gestionnaire d'interaction de service (**SCIM, Service Capability Interaction Manager**) permet la gestion des interactions entre les serveurs d'application SIP. Le SCIM est introduit par 3GPP TS 23.002 [5], il s'occupe de l'orchestration de l'interaction entre les "capacités" dans le plus grand réseau qui sont représentés comme les instances de serveurs de l'application SIP.

En plus des serveurs d'application, il existe un serveur de média appelé MRF (Multimedia Resource Function). Il offre des fonctions relatives aux flux medias par exemple ; Il établit des conférences multimédias, joue des annonces vocales ou multimédia, mélange des media Stream, fait un transcodage entre différents codecs.

L'entité MRF est décomposée en deux fonctions :

- La fonction **MRFP** (MRF Processor) qui traite le média à travers le transport RTP/UDP/IP

- La fonction **MRFC** (MRF Controller) qui traite la signalisation.

Nous allons maintenant détailler l'interfaçage entre la plateforme de service IMS et le réseau cœur IMS (Figure VI.15).

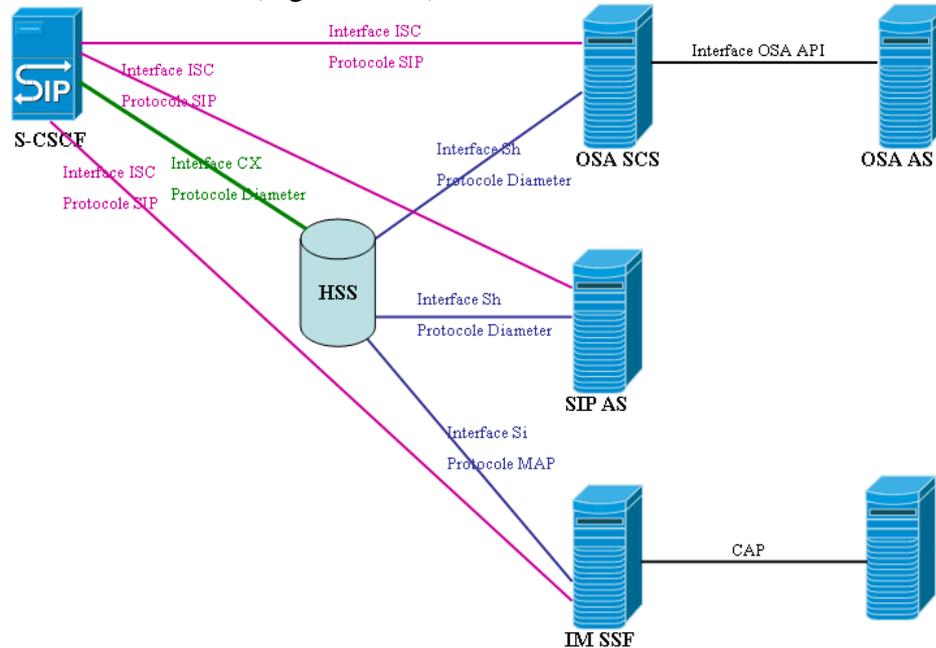


Figure VI.15 : Interfaces et protocoles de l'architecture de service IMS

Le SCSCF est d'abord lié à la base de données HSS du cœur IMS via l'interface Cx. Elle permet, pendant l'enregistrement de l'utilisateur dans le réseau IMS, de télécharger son profil du HSS dans le SCSCF à l'aide des messages DIAMETER (SAR/SAA).

Le SCSCF s'interface en SIP via l'interface ISC avec les serveurs d'application. Le SCSCF évalue la requête SIP qu'il reçoit d'un abonné et selon son profil de service, il décide ou non de relayer le message SIP au serveur correspondant pour réaliser un certain service.

Enfin les AS s'interfaçent en DIAMETER sur l'interface Sh avec la base informationnelle HSS pour s'informer des modifications éventuelles sur le profil d'un certain abonné ou bien pour mettre à jour les informations du HSS sur les profils des abonnés.

- Profil de service en IMS [6]

Le profil de service, est une collection d'information spécifique à l'utilisateur, stocké en permanence dans le HSS. Il est transféré du HSS à un S-CSCF alloué à l'aide de deux opérations « user datahandling operations » – Server-Assignment-Answer (SAA) and Push-Profile-Request (PPR). Il est défini dans un document XML dans un AVP Diameter.

Le profil de service (Figure VI.16) est divisé en trois parties :

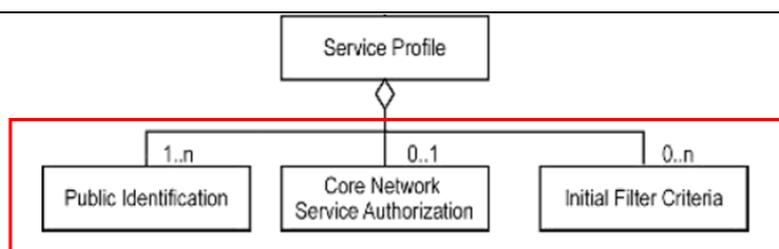


Figure VI.16 : Profil de service des utilisateurs

L'identification publique indique l'identité SIP URI et le Tel URI de l'abonné. Le profil de service de l'abonné peut définir une ou plusieurs identités publiques.

L'autorisation de service dans le réseau cœur contient les informations sur les règles de politique sur les types de media. En fait il contient un entier qui identifie un profil de media souscrit dans le S-CSCF (ex: paramètres de SDP permis). Cette information permet aux opérateurs de définir différents profils de souscription dans leurs réseaux IMS et différentes classes (ex Or, Argent et bronze). Le transfert de la valeur de l'entier entre le HSS et le S-CSCF économise de l'espace mémoire dans le HSS et optimise l'usage de l'interface Cx. Le S-CSCF a besoin d'une base de données statique qui fait correspondre l'entier et le profil de media souscrit mais il est à noter que la signification de la valeur de l'entier n'est pas standardisé (spécifique aux opérateurs)

Les critères de filtrage initiaux (Service-triggering information ou IFC) (Figure VI.17) décrivent quand un message SIP est routé vers un AS spécifique.

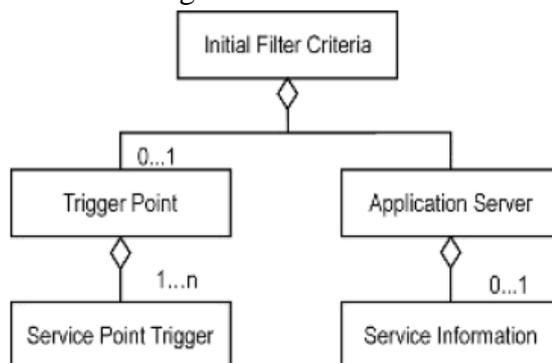


Figure VI.17 : Critères de filtrage initiaux (IFCs)

L'IFC associe un point de déclenchement (trigger point) constitué de points de déclenchements de service (service point trigger) à un serveur d'application hébergeant le service demandé.

Le point de déclenchement (Trigger point) décrit les conditions à vérifier pour savoir si l'AS indiqué doit être contacté. Il contient une ou plusieurs instances du **Service Point Trigger** qui peuvent être reliées avec des expressions logiques (AND, OR, NOT)

Le Service Point Trigger (Figure VI.18) indique la valeur des champs du message SIP à être vérifiés. Un SPT peut être basé sur par exemple la présence ou l'absence ou le contenu de n'importe lequel des champs suivant constituant le message SIP.

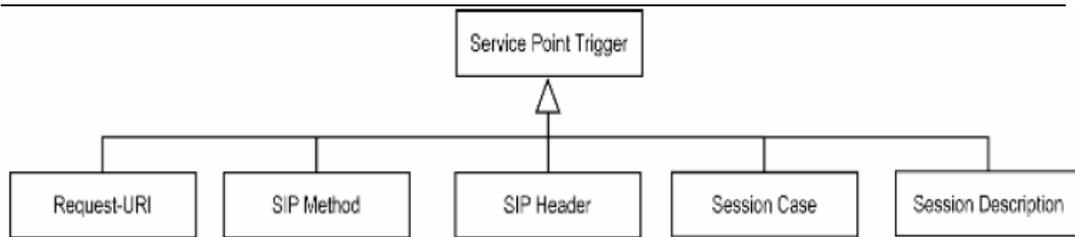


Figure VI.18 : Points de déclenchement des services (SPTs)

- Request-URI – identifie une ressource que la requête adresse (ex:sport@ims.example.com).
- SIP Method – indique le type de requête (ex: INVITE or MESSAGE).
- SIP Header – contient les informations concernant la requête.
- Session Case – peut être une des trois valeurs Originating, Terminating or Terminating_Unregistered, qui indique si le filtre doit être utilisé par le S-CSCF qui est en train de gérer le service originating, le service terminating ou terminating pour un service d'utilisateur non enregistré.
- Session Description – définit un SPT suivant le contenu de n'importe quel champ du SDP dans le corps de la méthode SIP.

Le profil de service d'un abonné peut donc être constitué de plusieurs IFCs classés par ordre de priorité afin de réaliser des services en séquence.

VI.2.4 Les protocoles et les interfaces

Les protocoles de signalisation les plus utilisés dans le réseau IMS sont SIP et Diameter. Ces deux protocoles représentent le langage que parlent les entités IMS pour communiquer entre elles afin de réaliser une session de service.

- Session Initiated Protocol - SIP

SIP est un protocole de signalisation [7] défini par l'IETF (Internet Engineering Task Force) permettant l'établissement, la libération et la modification de sessions multimédias. 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. L'adressage utilise le concept d'URL SIP (Uniform Resource Locator) qui ressemble à une adresse E-mail. Chaque participant dans un réseau SIP est donc adressable par une URL SIP. Par ailleurs, les requêtes SIP sont acquittées par des réponses identifiées par un code numérique. D'ailleurs, la plupart des codes de réponses SIP ont été empruntés au protocole HTTP. Par exemple, lorsque le destinataire n'est pas localisé, un code de réponse « 404 Not Found » est retourné. Une requête SIP est constituée de headers comme une commande SMTP. Enfin SIP comme SMTP est un protocole textuel.

Le *Session Description Protocol* – **SDP** est un élément d'un message SIP qui décrit les caractéristiques de qualité de service échangées entre 2 extrémités à l'établissement d'une session IMS. Les paramètres SDP incluent le type de média

(voix, audio, vidéo, etc...), les codecs (G.711, etc.) et la bande passante demandée.

- **DIAMETER**

Le protocole Diameter, l'évolution du RADIUS, est le protocole AAA dans IMS (authentification, autorisation et Accounting).

Le protocole Diameter fournit [7] les fonctionnalités suivantes:

- Délivre les AVPs (attribute value pairs)
- Possède une capacité de négociation
- Notifie sur les erreurs
- Extensible grâce à la possibilité d'ajouter des nouvelles commandes et des avps.
- Permet de faire des services basiques nécessaires pour les applications comme la manipulation des sessions des utilisateurs ou la comptabilité

Les AVPs sont utilisés par le protocole DIAMETER pour :

- Transporter les informations d'authentification et d'autorisation
- Echanger des informations qui peuvent servir pour la comptabilité
- Envoyer les informations sur le profil des services des utilisateurs.

Les différentes entités, qu'on avait citées dans le paragraphe précédent, sont reliées entre elles à l'aide d'interfaces et parlent entre elles en utilisant les protocoles décrits précédemment. Le tableau suivant nous donne un aperçu des différentes interfaces et des protocoles qu'elles supportent.

Interface	Rôle	Protocole
Gm	Echange de messages entre l'UE et les PCSCFs	SIP
Mw	Echange de messages entre les CSCFs	SIP
Cx	Communication entre I-CSCF/S-CSCF et le HSS	Diameter
Dx	Utiliser par I-CSCF/S-CSCF (pour communiquer avec le SLF) afin de trouver le bon HSS dans le cas d'un réseau qui contient plusieurs HSS.	Diameter
Mm	Echange de messages entre l'IMS et un réseau IP externe.	
Mg	Transite la signalisation ISUP transformé en signalisation SIP par la MGCF, à l'I-CSCF	SIP
Mi	Echange de messages entre le S-CSCF et le BGCF	SIP
Mj	Transite les messages entre le BGCF et le MGCF qui sont dans le même réseau.	SIP
Mk	Utiliser pour échanger des messages entre différentes BGCFs, qui n'appartiennent pas forcément au même réseau IMS.	SIP

Tableau VI-5: Les interfaces de l'IMS

VI.2.5 Vers IMS+

Le réseau IMS a été conçu pour offrir la possibilité aux utilisateurs d'accéder à leurs services en utilisant n'importe quel terminal et en passant par n'importe quel type de réseau d'accès fixe ou mobile. Cette accessibilité aux services est rendue possible par la convergence des réseaux de télécommunications qu'assure l'IMS. Mais la convergence suppose aussi la capacité d'assurer une continuité de session au cours des différentes mobilités qui peuvent survenir durant la session de service.

- La mise en œuvre d'une session de service IMS

Dans une session de service IMS un abonné demande un (ou plusieurs) service. Un service peut être un appel normal /conférence voix /Visio, un service de vidéo sur demande ou un autre service multimédia auquel il s'est souscrit. L'abonné ayant établi un contrat de service IMS avec son opérateur doit passer par la phase d'attachement au réseau d'accès puis l'enregistrement au réseau IMS avant de pouvoir établir n'importe quelle session de service (Figure VI.19).

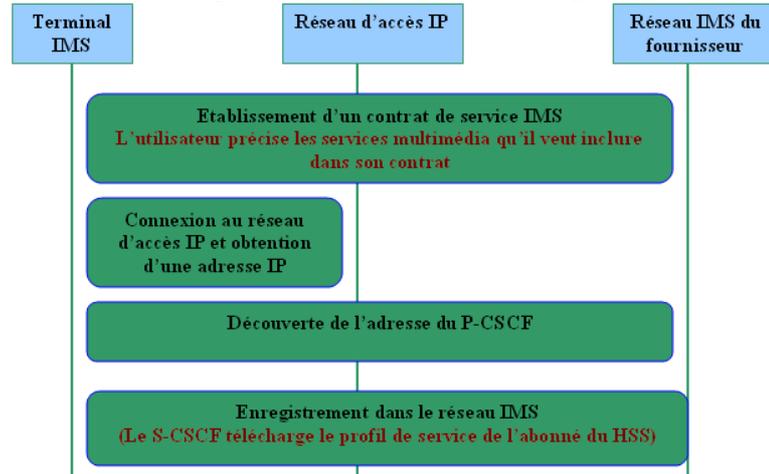


Figure VI.19 : Procédure générale pour accéder aux services IMS

La procédure d'enregistrement au réseau IMS permet à un terminal de se déclarer joignable du point de vue service IMS, le terminal sera authentifié par le réseau IMS et son profil de service sera chargé du HSS au S-CSCF nominal avec les messages SAR et SAA comme on peut le voir dans le diagramme (Figure VI.20) suivant :

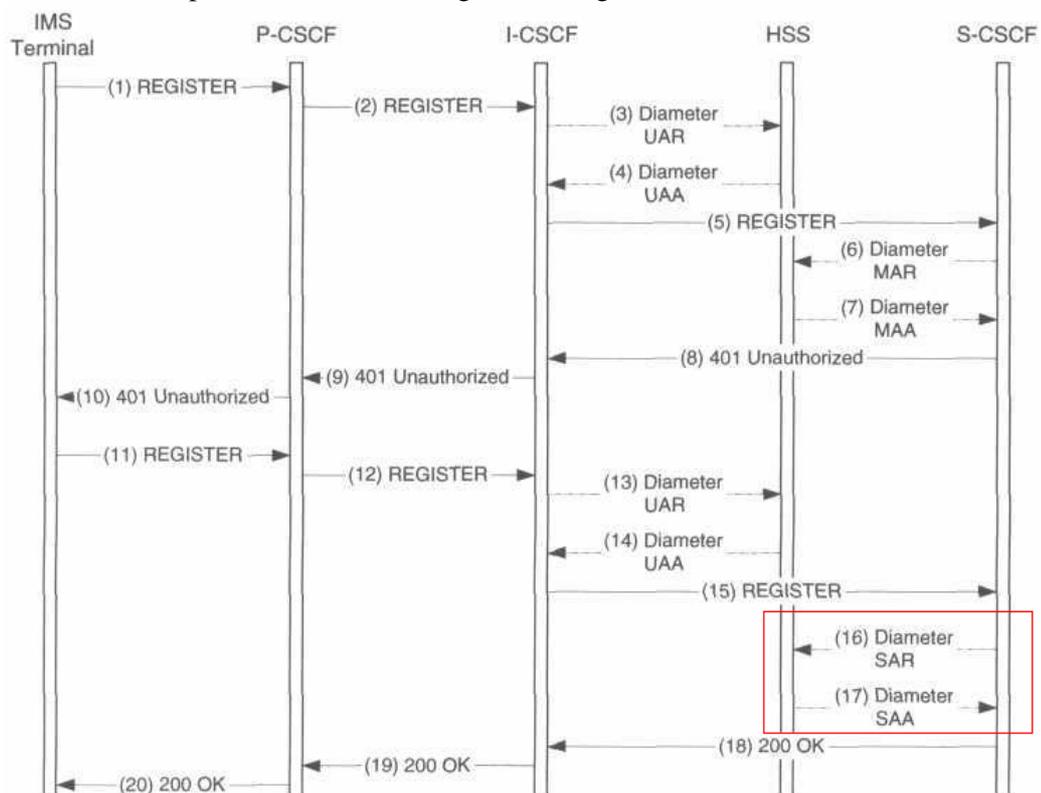


Figure VI.20 : Enregistrement d'un abonné et téléchargement de son profil de service

En fait après que l'UE s'est authentifié auprès du réseau IMS et qu'un SCSCF lui a été alloué. Ce SCSCF envoie un message diameter SAR (Server Assignment Request) pour demander à la base de données de l'assigner à l'identité publique de l'UE en question, et de lui envoyer les informations correspondantes à cette identité publique. Le HSS envoie alors en réponse le message SAA (Server Assignment Answer) au SCSCF contenant le profil de service de l'abonné sous forme d'un fichier XML. Enfin le HSS accepte l'enregistrement de l'UE en lui envoyant une réponse 200 OK. Une fois que l'UE est autorisé, il est capable d'initialiser et de recevoir des sessions.

- Invocation des services dans l'IMS [8]

Pour expliquer la méthode actuelle adoptée pour réaliser la session de service dans l'IMS, supposons qu'un UE a un profil de service constitué de plusieurs IFCs classés par ordre de priorité. Il envoie un message SIP demandant des services. Ce message SIP correspond en fait à plusieurs IFCs définis dans le profil de service de cet UE. Quand le SCSCF reçoit le message SIP, il doit le comparer avec tous les IFCs dans le profil de l'UE par ordre de priorité. Et à chaque fois qu'il trouve une correspondance avec le Trigger point (TP) d'un IFC il doit envoyer le message SIP vers l'AS associé à ce TP. Ensuite quand l'AS reçoit le message SIP et termine son traitement, le message est retourné au SCSCF pour que ce dernier continue sa comparaison des IFCs suivant afin de retrouver un nouveau IFC qui matchera avec les champs du message SIP induisant aussi à l'envoi du même message SIP vers l'AS associé au TP de ce nouveau IFC et ainsi de suite (Figure VI.21).

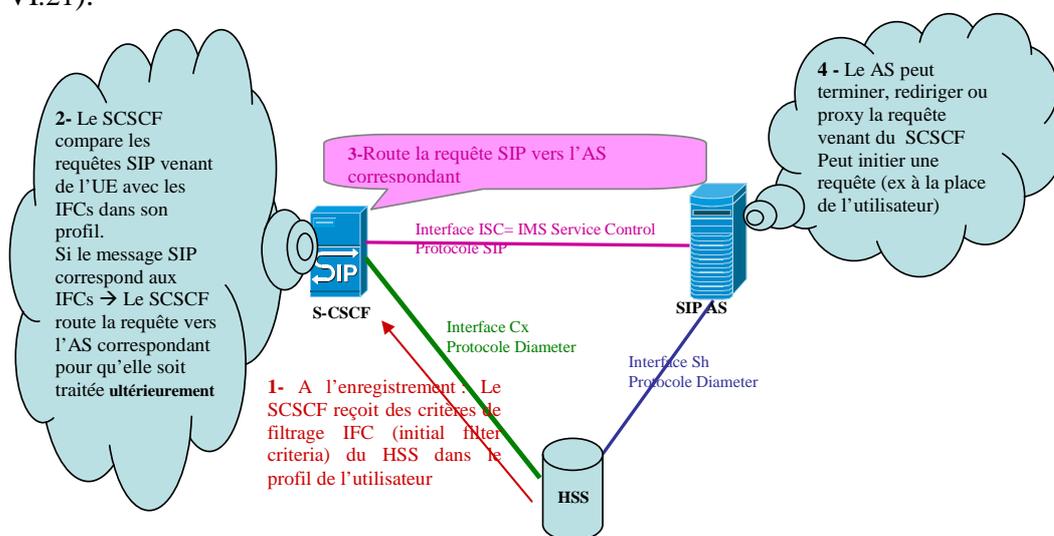


Figure VI.21 : Méthode adoptée pour invoquer services dans l'IMS

Nous allons dénommer IMS+, l'architecture qui tiendrait compte de nos propositions (Figure VI.22). Elle contient une base de données HSS évoluée pour offrir des informations complètes mise à jour dynamiquement à temps réel (par exemple les préférences des utilisateurs, les terminaux des utilisateurs, les réseaux auxquels avaient souscrit les utilisateurs, les services etc...). Un middleware NGN a été proposé pour composer, sélectionner d'une manière optimale et invoquer les éléments de services (ESs). La proposition de la logique de service et de la table de routage de Qualité de service et le protocole SIP avancé (SIP+) rendent la session de service dynamique de bout en bout.. Le middleware NGN (MNGN)

propose des services basiques pour supporter le réseau overlay de service : le framework du MNGN est présenté.

Dans notre vision, les services doivent être accessibles de n'importe quelle localisation et quel moment. On a proposé d'avoir une vision orientée service au lieu de celle orientée application. Pour cela, on a proposé un middleware NGN qui peut être considéré comme un AS évolué de l'IMS. Le middleware offre les services basiques qui permettent :

- La composition de service en éléments de services (SE - Service Element) dépendamment de la logique de service.
- La session de service suivant la table de routage QoS (Service QoS Management)
- La mobilité et gestion des services par la gestion des communautés de services (Service Community Management)
- La gestion de l'utilisateur à travers la gestion de la communauté qui gère la localisation des utilisateurs (User Location)

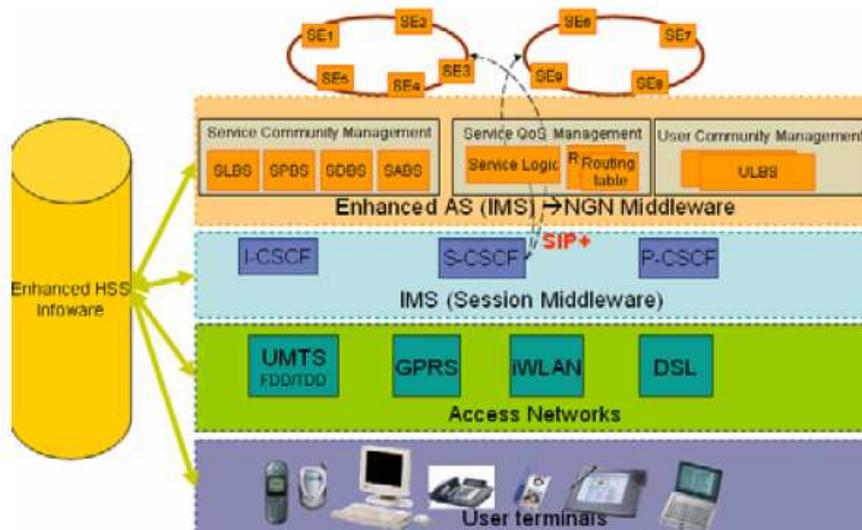


Figure VI.22 : Un NGN Middleware et un HSS évolué

Le middleware NGN assure la continuité de la session en contrôlant le lien sémantique entre les éléments de service suivant la table de routage de QoS. Il remplace la logique d'orchestration entre les ASs réalisé par le SCSCF ou le SCIM. Donc notre MNGN sélectionne et invoque d'une manière optimale (dépendamment des états des SEs) des éléments de services à la réception d'une requête SIP de la session middleware IMS. Nous regardons la diagramme de séquence pour une signalisation de SIP.

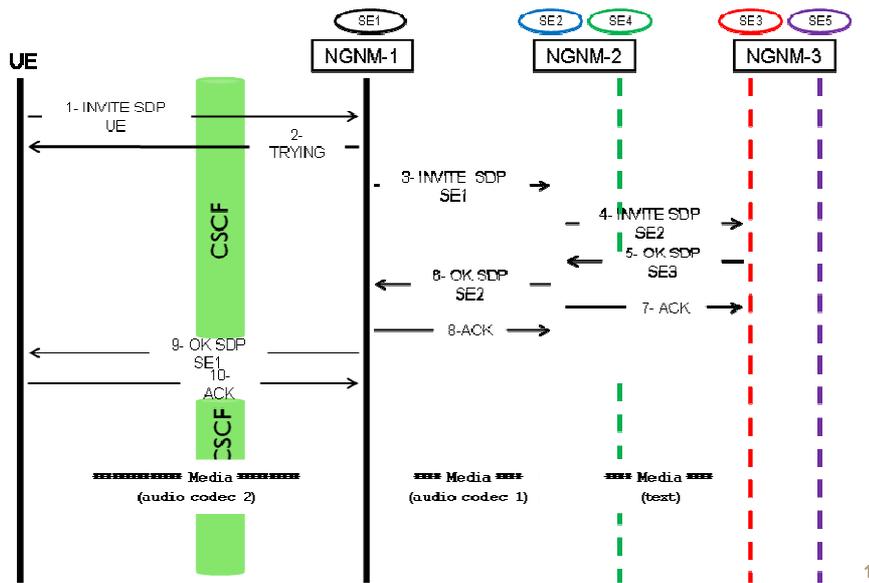


Figure VI.23: Diagramme de séquence de SIP

Dans notre vision et pour pouvoir réaliser le service global distribué, le protocole SIP, langage utilisé par les MNGNs, doit être modifié pour supporter les informations, sur la session de service, fournies par la table de routage de QoS (Figure VI.24).

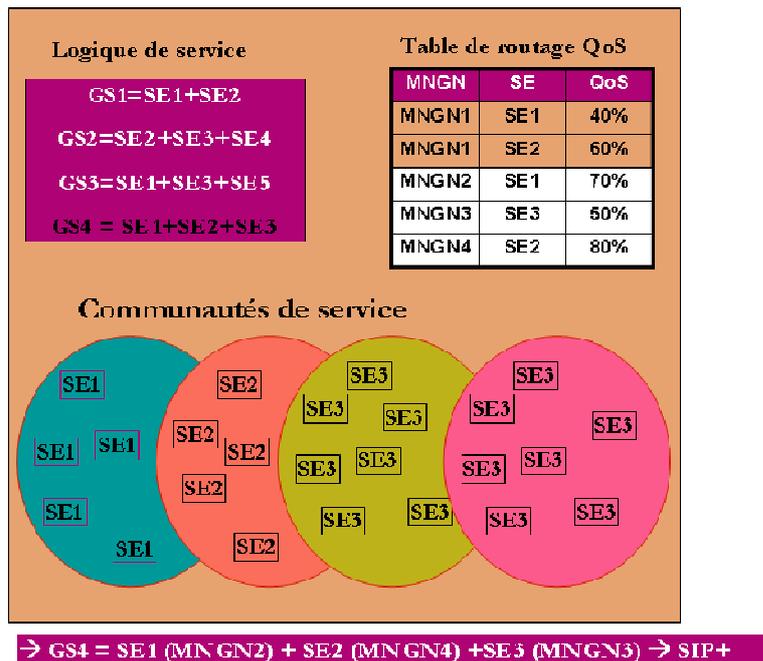


Figure VI.24: Construction d'un GS

On a nommé ce protocole SIP auquel des champs doivent être ajoutés « SIP+ ». Ces champs vont informer les MNGNs de la session de service complète constituée d'une séquence de SEs. Il est à noter que dépendamment du dimensionnement, un ES peut être déployé dans un ou plusieurs MNGNs avec une QoS appropriée. Différentes instances de ce ES peuvent s'exécuter sur de multiples MNGNs (Tableau VI-6).

Noms	ID	MNGN-address	MNGN-Localisation	MNGN-QoS
MNGN1	1	MNGN1@domain.com	Glacière	40%
MNGN2	2	MNGN2@domain.com	Barrault	80%
MNGN3	3	MNGN3@domain.com	Montparnasse	60%

Tableau VI-6: Multiples NGNs

Nous essayons d'appliquer le SIP+ sur la séquence que nous venons de donner. Dans le message de l'invitation (1-INVITE SDP UE), nous rajoutons les informations sur la QoS demandée et la logique de services de l'utilisateur (Figure VI.25 - a). Pour le deuxième message de l'invitation (3-INVITE SDP SE1), NGNM-1 découvre que l'application «SG» demandée par John est composée de trois éléments de service SE1, SE2 et SE3. Il transfère donc la logique de service avec ses résultats de raisonnement, et note que le SE1 est déjà installé sur lui-même (Figure VI.25 - b).

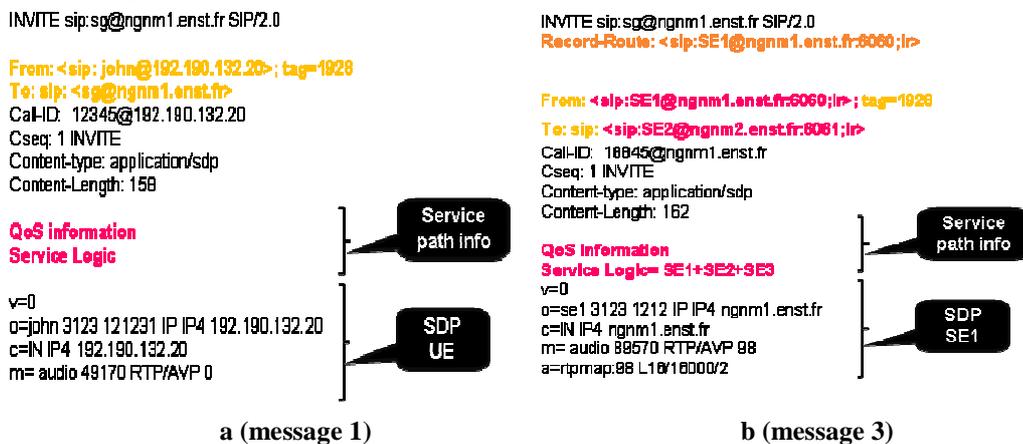


Figure VI.25: Message SIP+ (a) et (b)

Quant au troisième message de l'invitation (4 INVITE SDP SE2), le NGNM 3 fait le provisionning de ES3 comme indiqué dans la logique de service (Figure VI.26 - c). Quand la session est complète, un message de SIP+ OK (Figure VI.26 - d) est retourné grâce au trajet enregistré.

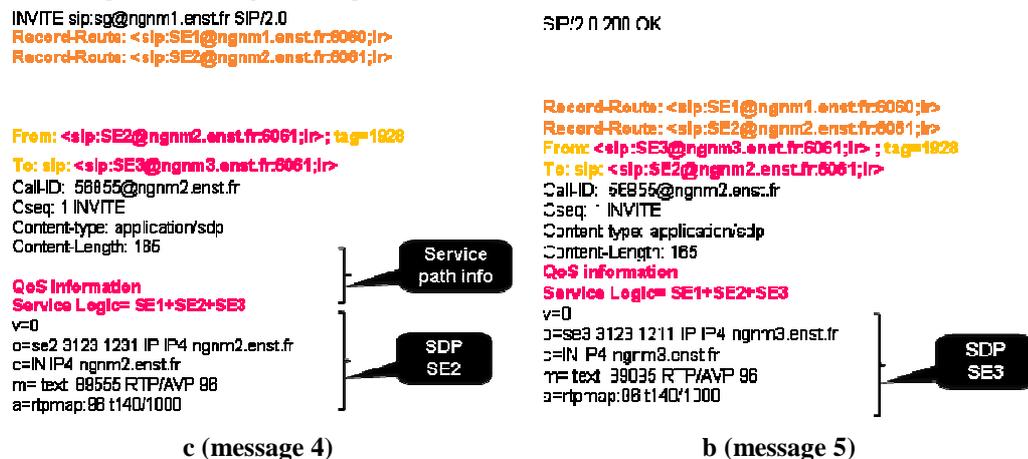


Figure VI.26: Message SIP+ (c) et (d)

En plus, on a proposé une base de données HSS évoluées qui joue le rôle d'Infoware et offre des informations complètes mises à jour d'une manière

automatique et plus dynamique que l'HSS existant. Dans ce HSS évolué, la QoS de bout en bout est aussi prise en considération. Pour cela l'utilisateur pourra se connecter au réseau à travers des terminaux sophistiqués et aura la possibilité de sélectionner des services ayant différentes options de Qualité de service. Ceci grâce aux informations complètes dans notre HSS évolué sur les préférences de l'utilisateur, les terminaux de l'utilisateur, les réseaux auxquels il s'est souscrit, la session IMS peut ainsi être enchaînée depuis le terminal de l'utilisateur, le bearer, le réseau cœur IMS jusqu'à la session de service (MNGNs).

VI.2.6 Plate-forme de tests OpenCoreIMS

Il est très important et nécessaire de tester l'IMS existant pour comprendre concrètement l'inter fonctionnement et l'interaction entre les différentes entités du cœur IMS et de la plateforme de service. Ces tests nous ont permis de voir les limitations dans les architectures de services actuelles afin de proposer des améliorations dans la réalisation de la session de service pour optimiser et assurer l'accès aux services d'une manière transparente et continue.

Le projet « Open Core IMS » de Fokus est un projet dédié à être une référence de base pour les tests d'implémentation de la technologie IMS et une application prototype pour le domaine de recherche sur IMS. Dans cette partie plusieurs aspects du cœur open source de Fokus ont été présentés et détaillés. D'abord on a présenté l'architecture et des différentes entités constituant le cœur IMS du projet de fokus, puis on a cité les étapes à suivre pour l'installer ensuite on a montré les différentes fonctionnalités offertes par son interface graphique d'administration FHoSS. En plus, on a parlé plus en détail de la plateforme applicative de service qui est maintenant en cours de développement. Les différentes entités (S-CSCF, HSS et AS), les interfaces (ISC, Sh et Cx) et les protocoles (SIP et DIAMETER) impliqués pour offrir les services aux utilisateurs ont été étudiés de près et plusieurs tests ont été faits. Puisque la plateforme de service n'existait pas dans la version actuelle du cœur open source IMS, on était amené à ajouter un serveur d'application pour tester son intégration et son inter fonctionnement avec le cœur existant.

L'**Open Source Core IMS** est une implémentation de la spécification IP Multimedia Subsystem de 3GPP déployée par FOKUS (institut allemand des communications). **Les entités constituant le cœur IMS sont basés sur le SIP Express Router (SER) qui est un Proxy SIP** mais avec des fonctionnalités avancées leur permettant d'être intégrés aux projets avancés de VoIP.

Les modules constituant l'Open Source Core IMS sont [9]:

- **FHoSS** qui constitue la base de données HSS dans l'IMS.
- **CDiameterPeer** ou **JavaDiameterPeer** constituant le protocole DIAMETER employé par différentes interfaces comme Sh et Cx reliant les entités HSS, S-CSCF, I-CSCF et AS.
- **Ser-ims** constituant les 3 entités du CSCF (P, I et SCSCF). Chaque entité est implémentée 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 indiqués par l'appelant ou l'appelé et basé sur ses

réponses route les messages vers le S-CSCF approprié, CDiameterPeer (cdp) est utilisé pour la communication avec le HSS sur l'interface Cx d'IMS (**SER + icsf + cdp**).

- **Module Serving-CSCF (scscf)** : Communique également avec le HSS en utilisant le protocole diameter sur l'interface Cx pour télécharger le profile de l'utilisateur et décider vers quel serveur d'application (AS) le message sera expédié à travers l'interface ISC (**SER + scscf + cdp + isc**).

Le cœur open source IMS est constitué alors du CSCF et de la base de données HSS mais plusieurs entités n'existe pas encore ou sont en cours de développement (Figure VI.27) :

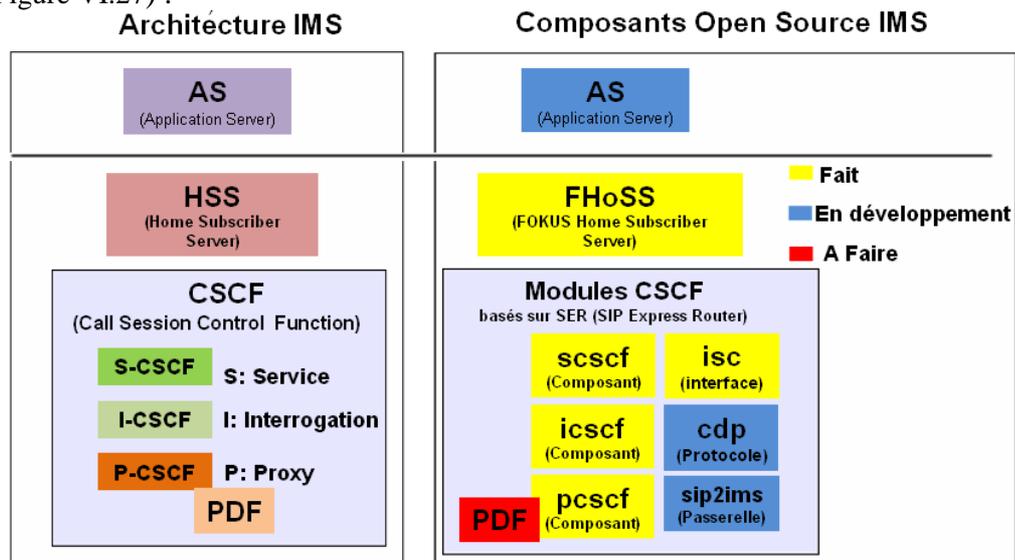


Figure VI.27 : Comparaison entre l'architecture d'IMS et celle du cœur Open source

VI.2.7 Conclusion

Grâce aux tests et l'implémentation que nous avons effectués, nous avons pu voir la faisabilité de nos propositions sur un middleware de NGN basé sur environnement IMS+.

Mais il nous manque quelques éléments dans la plate-forme finale, l'évaluation est donc ne peut pas se faire pour l'instant.

VI.3 Références

- [1] Bernard Traversat, Ahkil Arora, Mohamed Abdelaziz, Mike Duigou, Carl Haywood, Jean-Christophe Hugly, Eric Pouyoul, Bill Yeager, "Project JXTA 2.0 Super-Peer Virtual Network", Sun Microsystems, Inc.
- [2] Emmanuel Lavinal, Noémie Simoni, "Dynamic and adaptive composition of SIP-based services", ICC'08, Beijing, China, 2008
- [3] GONZALO CAMARILLO, MIGUEL A. GARCIA-MARTIN – The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds, WILEY, 2ème Edition, 2006
- [4] Simon ZNATY et Jean-Louis DAUPHIN, IP Multimedia Subsystem :Principes et Architecture, EFORT
- [5] 3GPP TS 23.002 V8.3.0 "Technical Specification Group Services and Systems Aspects"; (Release 8), 2008

- [6] 3GPP TS 23.218: "IP Multimedia (IM) session handling; IM call model; Stage 2 (Release 7)
- [7] RFC 3261 – SIP : Session Initiation Protocol
- [8] Miika Poikselkä, Georg Mayer, Hisham Khartabil, Aki Niemi - The IMS: IP Multimedia Concepts and Services, 2nd edition, , Wiley, 2006
- [9] <http://www.fokus.fraunhofer.de/home/>

Chapitre VII Conclusion générale

L'objectif de cette thèse était d'apporter une contribution dans le nouveau paysage des télécommunications en proposant une nouvelle orientation, celle de « l'User Centric » pour enfin aboutir à la convergence complète, c'est-à-dire, non seulement celle du fixe et du mobile, mais aussi celle du monde des télécommunications et du monde de l'informatique. Nous avons donc contribué au niveau des architectures des services (NGS) pour répondre aux besoins des services de la NGN et au niveau de la session pour supporter toutes les mobilités. Dans cette conclusion, nous allons faire la synthèse de nos contributions (§VII.1) et formuler nos perspectives (§VII.2).

VII.1 Contributions

La thèse avait pour objectif le souci d'offrir une continuité de service sans couture et intégrée. Nous avons commencé par délimiter le contexte NGN/NGS en nous focalisant sur les domaines qui pouvaient impacter la continuité de service, c'est-à-dire, l'hétérogénéité et la mobilité. Puis, il semblait important de changer d'orientation et d'avoir une approche « User-Centric » pour mieux répondre le besoin de l'utilisateur d'aujourd'hui sans oublier le ROI maximum et un TTM minimum des fournisseurs de services.

Cette analyse nous a permis d'identifier quatre verrous prioritaires à lever :

- **Personnalisation du service**

Ce point nous a conduit à proposer des solutions pour la composition de service avec les critères d'interopérabilité, de réutilisabilité, de mutualisation, et d'autonomie (l'autogestion). La gestion dynamique et autonome se faisant dans une architecture pertinente.

- **Une seule session utilisateur pour les différents services sollicités**

Ce point est la problématique capitale qui nous a inspiré pour chercher à spécifier un ensemble des « services supports » pour maintenir la continuité de service dans une session orientée utilisateur.

- Accessibilité au service par n'importe quel moyen

Ce point nous a conduit à proposer une session mobile pour supporter les différentes mobilités.

- Une QoS de bout en bout

Ce dernier point est le noyau fédérateur car pour une demande de QoS de bout en bout de la session nous devons considérer la QoS à tous les niveaux.

Devant ce contexte très complexe nous avons d'abord proposé, un modèle organisationnel qui réponde à l'environnement de mobilité NGN. Un des points avantageux de ce modèle est de considérer la couche de service à part entière. C'est en fait une couche de réseau overlay de composants de service.

Ensuite, nous avons proposé des règles de composition de service afin de couvrir tous les besoins que nous avons identifiés. Chaque composant de service est défini comme un élément générique basé sur des opérations atomiques, accessible pour tous les acteurs (l'utilisateur, le contrôle et la gestion). De plus, nous avons choisi une communication Peer-to-Peer.

Pour les mobilités, nous proposons de les gérer à travers des communautés d'intérêt, qui sont en fait des groupes de ressources ayant des caractéristiques communes (même appartenance, même fonction et même QoS), ainsi que le même rôle pour l'utilisateur. La gestion de ces communautés a pour objectif de remplacer en temps réel un composant automatiquement et dynamiquement au moindre dysfonctionnement. Cette intervention autonome se fait au niveau de chaqueVPXN qui est le résultat de notre modélisation.

Pour associer les modifications de chacun des niveaux, nous proposons un « NGN Sessionware » pour opérer un Binding des quatre niveaux architecturaux. C'est la coopération avec l' « Infoware » qui permet d'avoir les bonnes informations décisionnelles. C'est le filtrage en fonction des préférences données par le « Seamless Userware » qui permet d'avoir une session personnalisée.

Nous avons essentiellement travaillé sur l'axe fonctionnel, dont nous avons évalué la faisabilité sur deux types de plates-formes. Pour nos propositions de gestion de mobilités c'est à travers des prototypes sur une plate-forme JXTA et pour la composition de services dynamique et la gestion de la mobilité de session, c'est dans un environnement IMS.

VII.2 Perspectives

Tout le long de notre travail de recherche sur l'axe fonctionnel, nous avons bien sûr collaboré à l'avancement des autres axes, il faudrait maintenant rassembler l'ensemble des contributions et avoir un démonstrateur non seulement sur la faisabilité mais aussi sur les performances. Les autres axes auxquels nous faisons référence sont ceux de :

1. La dimension informationnelle :
 - Une base de connaissances qui représentent les ressources hétérogènes et l'intégration de l'usage et ses fonctionnalités d'inférences.
2. La dimension protocolaire avec convergence des interfaces :
 - QoS signaling
 - QoS continuity

Liste des publications

Noémie SIMONI, Chunyang YIN, Ken CHEN
E2E Service Delivery through user mobile session management
Soumis à NOVATICA, Spain

Jing CHI, Chunyang YIN, Meina SONG, Junde SONG, Mei SONG
Generic Service Composition Platform for Pervasive E-Commerce
Soumis à Wireless Communications and Mobile Computing, UK

Noémie SIMONI, Xiaofei XIONG, Chunyang YIN
Virtual Community for the Dynamic Management of NGN Mobility
ICAS'09, Avril, 09, Valencia, Spain

Wen GUO, Noémie SIMONI, Chunyang YIN
"Automated Management of User Centric Session in NGN",
DANMS'08 Nov. 2008, New Orleans, LA, USA

Noémie SIMONI, Chunyang YIN, Ghislain Du Chéné
"Service continuity management through an E2E dynamic session in NGN",
NOMS'08, Avril. 2008, Salvador, Bahia, Brezil

Noémie Simoni, Chunyang Yin, Ghislain du Chéné,
"An Intelligent user centric middleware in NGN: Infosphere and AmbientGrid",
COMSWARE'08, Jan. 2008, Bangalore, India

Noémie Simoni, Chunyang Yin, Rhéa Berberi, Ghislain du Chéné,
"An NGN middleware based on an enhanced IMS",
MNCNA.07, Nov. 2007, Newport, California, USA

Noémie Simoni, Bertrand Mathieu, Chunyang Yin, Meng Song,
"Autogestion de service par la QoS dans un Réseau Overlay",
GRES.07, Nov. 2007, Hammamet, Tunisia

Chunyang Yin, Noémie Simoni, Ghislain du Chéné,
"A personalization and mobility aware service enabler for a service continuity in
heterogeneous networks",
MSPE.06, Nov. 2006, Aachen, Germany

F. Bennani, Z. Benahmed Daho, N.Simoni, Chunyang Yin,
"An Informational Framework for Autonomic Networking",
GRES.06, May, 2006, Bordeaux, France

Cahier des Charges de l'aire fonctionnelle « continuité de service »