



HAL
open science

Gestion Dynamique de Service de Bout en Bout dans un Contexte de Mobilité et d'Ubiquité: du Déploiement au Delivery des services

Soumia Kessal Ouanouche

► To cite this version:

Soumia Kessal Ouanouche. Gestion Dynamique de Service de Bout en Bout dans un Contexte de Mobilité et d'Ubiquité: du Déploiement au Delivery des services. Réseaux et télécommunications [cs.NI]. Télécom ParisTech, 2011. Français. NNT: . pastel-00679283

HAL Id: pastel-00679283

<https://pastel.hal.science/pastel-00679283>

Submitted on 15 Mar 2012

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.



Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

Télécom ParisTech
Spécialité “ Informatique et Réseaux ”

présentée et soutenue publiquement par

Soumia KESSAL OUANOUCHE

le 22 septembre 2011

**Gestion Dynamique de Service de Bout en Bout
dans un Contexte de Mobilité et d’Ubiquité :
du Déploiement au Delivery des services**

Directeur de thèse : **Noémie SIMONI**

Jury

Mme. Francine KRIEF, Professeur, ENSEIRB, LaBRI, Université de Bordeaux1
Mme. Michelle SIBILLA, Professeur, IRIT3, Université Paul Sabatier

M. Philippe COUDE, DGRé/DO/DPM, SFR
Mme. Gladys DIAZ, Maître de conférences, L2TI, Université Paris13
M. Samir TOHME, Professeur, PRISM, Université de Versailles

Mme. Noémie SIMONI, Professeur, INFRES, Télécom ParisTech

Rapporteur
Rapporteur

Examineur
Examineur
Examineur

Directeur de thèse

T
H
È
S
E

Télécom ParisTech

Grande école de l’Institut Télécom – membre fondateur de ParisTech

46, rue Barrault – 75634 Paris Cedex 13 – Tél. + 33 (0)1 45 81 77 77 – www.telecom-paristech.fr

Résumé

L'intensification de la concurrence et les exigences des clients pour des services toujours plus innovants et fiables conduisent l'opérateur à accélérer le renouvellement de son portefeuille de services et à vouloir devancer ses concurrents dans leur mise sur le marché. Mais dans le nouveau contexte NGN/NGS où l'utilisateur est nomade, change de terminal, change d'environnement et désire une continuité de service en tout lieu et selon ses préférences, la fourniture de services depuis la stratégie jusqu'à l'exploitation devient de plus en plus complexe et longue à maîtriser. Ainsi, avoir une gestion dynamique qui permet d'améliorer le TTM et le ROI devient un challenge pour les opérateurs. Dans cette thèse, nous nous intéressons à la résolution de problèmes de gestion du cycle de vie des services depuis le Déploiement jusqu'au Delivery dans un environnement ubiquitaire et mobile.

Pour ce faire, dans un premier temps, nous commençons par analyser les travaux existants en termes de gestion du cycle de vie des services, puis nous identifions plus précisément les besoins du nouveau contexte NGN/NGS du point de vue de l'utilisateur et de l'opérateur. Les verrous à lever concernent les différentes phases du cycle de vie des services. Comment les repenser afin d'avoir une continuité de service et un maintien de la QoS dans ce contexte de mobilité et d'ubiquité. Nos contributions sont au nombre de quatre.

Partant d'un travail en commun du groupe de recherche AIRS, qui a permis la définition d'un modèle de service et de ses propriétés intégrant les aspects fonctionnels et non fonctionnels (QoS) nous avons reconsidéré les différentes phases du cycle de vie.

Notre première contribution porte sur la phase de Déploiement. Elle vise à répondre au SLA de l'utilisateur à qui on attribuera le service dont la QoS offerte répond au mieux à celle demandée. En fait, chaque service possède un contrat de QoS configuré au moment du *déploiement*. Ceci pour tenir compte de la QoS offerte par tous les sous systèmes supportant le dit service. Chaque service s'auto-contrôle et notifie son "In" ou "Out" contrat.

Notre deuxième contribution porte sur la phase suivante qui couvre le Provisioning. Pour le service nous proposons un pré-provisioning pour ne pas mobiliser des ressources inutilement. Cela revient à sélectionner le service avec la bonne QoS au moment de l'ouverture de la session. Au moment de la demande de consommation, cette ressource service sera allouée dynamiquement en fonction de sa QoS et pourra être partagée entre plusieurs requêtes en fonction de cette QoS qu'elle offre. Le contrôle du respect du contrat de QoS de cette ressource service nous permet d'anticiper les dégradations et de re-router vers un autre composant de service ubiquitaire et ainsi de toujours mieux satisfaire le SLA de l'utilisateur.

Afin d'avoir une continuité de service notre troisième contribution propose un « Service Delivery » assuré par la couche service au dessus du « Media Delivery » assuré par la couche transport. Le processus proposé permet le suivi de la session de l'utilisateur en tenant compte de sa mobilité et des changements de l'environnement ambiant durant l'usage du service.

La dernière proposition couvre le maintien de la QoS de bout en bout. Les interactions avec les processus de Provisioning et de Management sont définies.

Deux champs d'application sont directement impactés par nos propositions et permettent de valoriser nos travaux.

Le premier se déduit de la gestion de la mobilité de la session de l'utilisateur par la QoS. Une gestion de cette session mobile par zones ambiantes est possible. Nous réalisons ainsi un Handover au niveau service qui permet de changer de service ubiquitaire par zone ambiante. Ce Handover Sémantique nous permet de garantir la continuité du service durant la mobilité et d'adapter la session des services au nouvel environnement ambiant de l'utilisateur, tout en tenant compte de ses préférences fonctionnelles et non fonctionnelles (QoS).

Le deuxième champ d'application est les standards télécom, et plus particulièrement eTOM afin de prendre en compte les exigences du contexte NGN/NGS. Les modèles définis et nos propositions basées sur les services et la QoS sont directement utilisables pour mieux tenir compte de l'usage de nos utilisateurs nomades d'aujourd'hui.

De façon plus concrète, nous présentons nos expérimentations sur une plate-forme composée de OpenIMSCore, le serveur d'application Glassfish V3, Oracle et des routeurs virtuels (VIRTUOR). Elles permettent de montrer la faisabilité de nos contributions.

En conclusion, les apports et les perspectives de cette thèse sont consignés.

*🌸 À ma mère, qui s'est sacrifiée toute sa vie pour nous permettre d'aller loin
dans nos études
À mon défunt père
À mon mari Lamir qui m'a soutenu tout au long de ma thèse et à mon fils Adam
À toute ma famille 🌸*

Remerciements

Je tiens tout d'abord à exprimer toute ma gratitude à ma directrice de thèse, Madame Noémie SIMONI, pour m'avoir offert l'opportunité de faire une thèse très enrichissante au sein de son équipe AIRS à Télécom ParisTech et qui m'a aidée à dépasser tous les obstacles et concilier ma vie professionnelle et mon rôle de maman tout au long de ma thèse. Je la remercie également pour son encadrement, son suivi, ses conseils, sa disponibilité et sa générosité.

Je remercie également Monsieur Antoine BOUTIGNON et Monsieur Philippe COUDE de la société SFR pour leur soutien, leur sympathie et leurs conseils.

Mes remerciements vont également à tous les membres de l'équipe AIRS dans laquelle j'ai évolué durant ces trois années de thèse et particulièrement, Mademoiselle Chunyang YIN pour toutes les explications et éclaircissements qu'elle m'a offert sur des points conceptuels et techniques concernant le sujet de mon stage.

Je remercie aussi les thésards et stagiaires pour leur esprit d'équipe et pour tous les bons moments et souvenirs inoubliables que nous avons eu ensemble.

Enfin, mes remerciements vont à Monsieur Samir TOHME, Membre du laboratoire PRiSM de l'université de Versailles Saint Quentin en Yvelines (UVSQ) pour son aide et ses conseils durant mon année de master.

Table des matières

RESUME	2
TABLE DES MATIERES	6
TABLE DES FIGURES.....	9
LISTE DES TABLEAUX.....	11
LISTE DES ABREVIATIONS	12
CHAPITRE I INTRODUCTION GENERALE.....	15
I.1 CONTEXTE.....	15
I.2 PROBLEMATIQUES	16
I.3 CONTRIBUTIONS DE LA THESE	17
I.3.1 <i>Déploiement Intelligent des éléments de service pour une E2E QoS efficace.....</i>	<i>17</i>
I.3.2 <i>Provisioning de service orienté QoS.....</i>	<i>18</i>
I.3.3 <i>Delivery de service.....</i>	<i>18</i>
I.3.4 <i>Gestion dynamique de la QoS sans couture.....</i>	<i>18</i>
I.3.5 <i>Valorisations</i>	<i>19</i>
I.4 ORGANISATION DU RAPPORT	19
CHAPITRE II LE CONTEXTE	21
II.1 INTRODUCTION	21
II.2 CYCLE DE VIE DE SERVICE	22
II.3 LE CONTEXTE NGN/NGS	23
II.3.1 <i>La convergence des réseaux.....</i>	<i>23</i>
II.3.2 <i>La convergence des services</i>	<i>23</i>
II.3.3 <i>La mobilité</i>	<i>24</i>
II.3.4 <i>Le réseau ambiant.....</i>	<i>25</i>
II.4 USER CENTRIC ET E2E QoS.....	25
II.5 PERIMETRE D'ETUDE DE LA THESE.....	25
CHAPITRE III ÉTAT DE L'ART	27
III.1 LES SOLUTIONS EXISTANTES POUR LE DEPLOIEMENT DE SERVICES	27
III.2 LES SOLUTIONS EXISTANTES POUR LE PROVISIONING DE SERVICES	30
III.2.1 <i>Introduction.....</i>	<i>30</i>
III.2.2 <i>Solutions de provisioning de services apportées par le SON</i>	<i>30</i>
III.2.3 <i>Solutions de provisioning de services apportées par SOA</i>	<i>31</i>
III.2.4 <i>Discussion</i>	<i>32</i>
III.3 LES SOLUTIONS EXISTANTES POUR LE DELIVERY.....	32
III.3.1 <i>Introduction.....</i>	<i>32</i>
III.3.2 <i>Travaux de recherche existants pour optimiser le SON</i>	<i>33</i>
III.3.3 <i>Exemple de Service Delivery.....</i>	<i>35</i>
III.3.4 <i>Discussion</i>	<i>36</i>
III.4 LES SOLUTIONS EXISTANTES POUR LE HANDOVER DANS LE RESEAU	37
III.4.1 <i>Introduction.....</i>	<i>37</i>
III.4.2 <i>Handover Horizontal de niveau 2 (L2).....</i>	<i>37</i>
III.4.3 <i>Handover Horizontal de niveau 3 (L3).....</i>	<i>40</i>
III.4.4 <i>Handover Vertical</i>	<i>40</i>

III.4.5	<i>Discussion</i>	41
III.5	GESTION DE SERVICES DANS LES STANDARDS	42
III.5.1	<i>Introduction</i>	42
III.5.2	<i>Évolution des architectures de gestion des télécommunications</i>	42
III.5.3	<i>Business Process Framework (eTOM)</i>	46
III.5.4	<i>Discussion</i>	52
III.6	CONCLUSION DE L'ETAT DE L'ART : LIMITES DES SOLUTIONS	53
CHAPITRE IV PROPOSITION : GESTION DU CYCLE DE VIE DANS UN CONTEXTE NGN/NGS		
54		
IV.1	DEPLOIEMENT INTELLIGENT D'ELEMENTS DE SERVICES POUR UNE QoS EFFICACE DE BOUT EN BOUT	54
IV.1.1	<i>Motivations</i>	54
IV.1.2	<i>Quoi déployer ? L'élément de service (SE)</i>	54
IV.1.3	<i>Comment déployer l'élément de service ?</i>	59
IV.1.4	<i>Quand déployer un élément de service ?</i>	62
IV.1.5	<i>Processus de Déploiement</i>	62
IV.1.6	<i>Redéploiement des éléments de service :</i>	65
IV.2	PROVISIONING DES ELEMENTS DE SERVICE ORIENTE QoS.....	66
IV.2.1	<i>Motivations</i>	66
IV.2.2	<i>VPxN et le Pré-Provisioning des éléments de service</i>	66
IV.2.3	<i>Provisioning de l'élément de service</i>	68
IV.2.4	<i>Autogestion de la ressource service</i>	69
IV.2.5	<i>Mécanisme de gestion de la file d'attente</i>	69
IV.2.6	<i>Algorithme de gestion de la file d'attente</i>	71
IV.3	SERVICE DELIVERY	72
IV.3.1	<i>Motivations</i>	73
IV.3.2	<i>Service Delivery dirigé par un modèle</i>	73
IV.3.3	<i>Processus de Service Delivery</i>	75
IV.3.4	<i>Prise en compte de la mobilité (VSC et ES Ubiquitaire)</i>	78
IV.4	GESTION DYNAMIQUE DE LA QoS SANS COUTURE.....	79
IV.4.1	<i>Introduction</i>	79
IV.4.2	<i>Contrat de QoS pour un cycle de vie cohérent</i>	79
IV.5	CONCLUSION DES PROPOSITIONS	80
CHAPITRE V VALORISATIONS.....		82
V.1	HANDOVER SEMANTIQUE POUR UNE CONTINUITÉ DE SERVICE SANS COUTURE DANS L'ENVIRONNEMENT AMBIANT	82
V.1.1	<i>Introduction</i>	82
V.1.2	<i>Scénario</i>	83
V.1.3	<i>Concept du Handover Sémantique</i>	85
V.1.4	<i>Initiateur du Handover</i>	86
V.1.5	<i>Décideur du Handover</i>	88
V.1.6	<i>Exécuteur du Handover</i>	90
V.1.7	<i>Conclusion</i>	90
V.2	PRISE EN COMPTE DE LA QoS DANS LE BUSINESS PROCESS FRAMEWORK	91
V.2.1	<i>Processus impliqués dans le Déploiement de Services dans eTOM</i>	91
V.2.2	<i>Processus de déploiement proposé</i>	93
V.2.3	<i>Processus proposés dans l'exploitation pour prendre en compte le contexte NGN/NGS</i>	95
V.2.4	<i>Processus de Provisioning proposé</i>	97
V.2.5	<i>Interaction entre les phases de Fullfilment, de Delivery proposé et d'Assurance</i>	98
V.2.6	<i>Conclusion</i>	100
CHAPITRE VI IMPLEMENTATION		102
VI.1	SCENARIO DE GESTION DE LA SESSION D'UN UTILISATEUR : DE L'ABONNEMENT JUSQU' A L'USAGE .	102
VI.1.1	<i>Abonnement : Création d'un « profil utilisateur » contenant tous les services abonnés par le client.</i>	102
VI.1.2	<i>Ouverture de session et Pré-provisioning : Création du VPSN</i>	104
VI.1.3	<i>Provisioning de QoS de l'élément de service durant l'usage</i>	105
VI.1.4	<i>Provisioning de QoS de la file d'attente durant l'usage</i>	106

VI.2	PLATE-FORME DE DEVELOPPEMENT ET DE TEST.....	106
VI.2.1	Élément de service de gestion « Queue_QoS Agent ».....	109
VI.2.2	Élément de service de gestion « VQC Maintenance ».....	112
VI.3	TEST DE FAISABILITE SUR LA FILE D'ATTENTE AVEC OPENMQ.....	112
VI.3.1	Scenario 1 : Provisioning du Service Par la QoS courante.....	112
VI.3.2	Scenario 2: Expiration du "Time out warning" des requêtes.....	113
CHAPITRE VII	CONCLUSION GENERALE.....	114
VII.1	CONTRIBUTIONS.....	114
VII.2	PERSPECTIVES.....	115
CHAPITRE VIII	REFERENCES BIBLIOGRAPHIQUES.....	117
LISTE DES PUBLICATIONS.....		121
CHAPITRE IX	ANNEXE.....	122

Table des Figures

Figure II.1-1 : Périmètre d'étude de la thèse	21
Figure II.5-1 : Cycle de vie du service.	26
Figure III.1-1: Descripteur de déploiement.....	29
Figure III.3-1 : Services Applicatifs et Services Réseaux dans l'architecture IPTV	35
Figure III.3-2: Service Delivery proposé au dessus du Media Delivery	36
Figure III.5-1: Le passage du network management au resource management.	43
Figure III.5-2 : Les blocs fonctionnels du NGNM.....	44
Figure III.5-3 : NGOSS	45
Figure III.5-4: eTOM de Niveau 1	47
Figure III.5-5 : Décomposition de niveau 2 de « Opérations ».....	48
Figure III.5-6 : Étapes de provisioning de ressources.....	49
Figure III.5-7: Flux de processus de Précommande dans eTOM.....	50
Figure III.5-8 : Flux de processus de Commande dans eTOM.	51
Figure III.5-9 : Flux de processus de Post Commande dans eTOM.	52
Figure IV.1-1: Profil d'équipement PDA N97.....	57
Figure IV.1-2: Profile d'Usage du service VoD.	59
Figure IV.1-3: Négociation de la QoS entre la ressource logique (service) à déployer et les ressources physiques.	60
Figure IV.1-4 : Profile d'Usage de la plate-forme A.	61
Figure IV.1-5 : Profil d'usage de la plate-forme B	61
Figure IV.1-6: Processus de déploiement	63
Figure IV.1-7: Diagramme de gestion des ressources service.	64
Figure IV.1-8: Diagramme de gestion de redéploiement de ressources	65
Figure IV.2-1 : Provisioning des ressources à tous les niveaux de visibilité.....	67
Figure IV.2-2 : “Resource Service” et exemple de Mutualisation de requêtes.....	68
Figure IV.2-3 : Autogestion par l'Agent QoS et VSC	69
Figure IV.2-4: Mécanisme de gestion de la file d'attente.....	70
Figure IV.2-5: autogestion (VSC, VQC).	71
Figure IV.2-6: Exemples des délais d'attente dans la file d'attente.....	72
Figure IV.3-1: Service Delivery avec une continuité de QoS selon la mobilité, les préférences de l'utilisateur et l'environnement ambiant	75
Figure IV.3-2: Création du VPSN.....	76
Figure IV.3-3 : Transaction.....	76
Figure IV.3-4: Processus Service Delivery et son interaction avec le Provisioning et le Management.	77
Figure IV.4-1: Contrat de QoS pour un cycle de vie cohérent.....	80
Figure V.1-1: Les ressources dans la zone ambiante.	83
Figure V.1-2 : Scénario de Handover au niveau service.....	85
Figure V.1-3: Déclencheur dans l'Infoware.....	87
Figure V.1-4 : Initiation du handover sémantique	87
Figure V.1-5 : Décision du handover sémantique.....	89
Figure V.1-6 : Exécution du handover sémantique.....	90
Figure V.2-1: Processus Développement et Retrait des services.	91
Figure V.2-2 : Décomposition du processus « Développement et Retrait des services ».....	92
Figure V.2-3 : Phase de déploiement proposée dans eTOM.....	92
Figure V.2-4 : Processus de déploiement.....	94

Figure V.2-5: Provisioning du service demandé par le client qu'elle que soit ses préférences et sa localisation	95
Figure V.2-6: Intégration de la QoS pour la prise en compte de la mobilité dans eTOM	97
Figure V.2-7: Flux du process Commande.	98
Figure V.2-8 : Interaction entre le processus d'assurance, de delivery et de provisioning....	100
Figure V.2-9: gestion dynamique de la mobilité dans le nouveau contexte NGN/NGS.....	101
Figure VI.1-1: Cas d'usage	103
Figure VI.1-2: Translation des services exposables et création du VPSN.....	104
Figure VI.1-3: Requête d'Alice envoyée vers la plate-forme de services.	105
Figure VI.1-4 : Requête d'Alice envoyée vers une file d'attente ubiquitaire.	106
Figure VI.2-1 : Différentes couches de visibilité.	107
Figure VI.2-2 : L'architecture du SERVICEWARE.....	108
Figure VI.2-3: Architecture d'un EJB (ES).	109
Figure VI.2-4: Automate d'un élément de service (Plan usage et Signalisation).	110
Figure VI.2-5 Automate d'un élément de service (Plan de gestion).....	111
Figure VI.3-1: Architecture JMS.	112
Figure VI.3-2 : Provisioning de service selon QoS_Capacity courante.....	113
Figure VI.3-3 : Expiration du time out, requête envoyé vers un SE ubiquitaire.....	113

Liste des Tableaux

Tableau III.6-1 : Limites des solutions existantes.....	53
Tableau IV.1-1 : Critères de QoS.....	57
Tableau IV.3-1: Table de QoS de la couche Service : Serveur1, Serveur2 et Serveur3	78
Tableau IV.3-2: Table de QoS de la couche transport : Serveur1 et Serveur2	78
Tableau IV.5-1 : Propositions et défi relevé dans le contexte NGN/NGS.....	81

Liste des Abréviations

3GPP	3rd Generation Partnership Project
API	Application Programming Interface
BER	Bit Error Ratio
BSC	Basic Station Controller
BSIC	Base Station Identity Code
CN	Core Network
CoA	Care of address
E2E	End-to-End
EDGE	Enhanced Data Rates for GSM Evolution
EJB	Enterprise Java Bean
ES	Element de Service
ETSI	European Telecommunication Standards Institute
eTOM	enhanced Telecommunication Operation Map
GAN	Generic Access Network
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
IMS	IP Multimedia Subsystem
IPTV	Internet Protocol Television
IT	Information Technology
ITIL	Information Technology Infrastructure Library
JMS	Java Message Service
JNDI	Java Naming and Directory Interface
LLA	Logical Level Architecture
LTE	Long Term Evolution
MDB	Message Driven Bean
MIPv4	Mobile IP version 4
MIPv6	Mobile IP version 6
MIH	Media Independent Handover
MN	Mobile Node
MS	Mobile Station
MSC	Mobile Switching Center

NASS	Network Attachment Sub-System
NGN	Next Generation Network
NGNM	NGN Management
NGOSS	New Generation Operations Systems and Software
NGS	Next Generation Service
NLR	Nœud Lien Réseau
OpenMQ	Open Message Queue
OSI	Open Systems Interconnection
PAN	Personal Area Network
PC	Personal Computer
PDA	Personal Digital Assistant
PEP	Policy Enforcement Point
QoS	Quality of Service
RACS	Resource and Admission Control Sub-System
RNC	Radio Network Controller
ROI	Return On Investment
RSCP	Received Signal Code Power
RSSI	Received Signal Strength Indicator
SDF	Service Delivery Function
SHS	Semantic Handover Service
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SOA	Service Oriented Architecture
SON	Service Overlay Network
SP	Service Platform
TMF	Tele Management Forum
TMN	Telecommunication Management Network
TTM	Time To Market
UBIS	User-Centric : uBiquité et Intégration de Services
UE	User Equipment
UMA	Generic Access Network
UMTS	Universal Mobile Telecommunications Systems
VEC	Virtual Equipment Community

VHO	Vertical HandOver
VM	Virtual Machine
VNC	Virtual Network Community
VoD	Video on Demand
VoIP	Voice over Internet Protocol
VPSN	Virtual private Service Network
VPxN	Virtual Private User/Service/Connectivity/Equipment Network
VQC	Virtual Queue Community
VSC	Virtual Service Community
WIMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
xDSL	x Digital Subscriber Line

Chapitre I Introduction Générale

I.1 Contexte

Aujourd'hui, les opérateurs doivent proposer des services de plus en plus rapidement et de plus en plus complexes, tout en garantissant une qualité de service (QoS) optimale. Ces fortes pressions du marché, et leur difficulté à maintenir leurs marges pour rester compétitifs, imposent aux opérateurs une meilleure maîtrise de leurs processus de gestion du cycle de vie de services depuis la stratégie jusqu'à l'exploitation. Avoir une cohérence entre les différents processus du cycle de vie permettra de maximiser leur RoI (Return on Investment) et de réduire considérablement leur TTM (Time to Market).

Le cycle de vie d'un service est composé de plusieurs phases qui commencent par la phase de Conception, qui permet de spécifier la fonctionnalité du service pour répondre aux demandes de la stratégie et de la planification responsable des prévisions sur les besoins futurs des clients actuels et potentiels. Puis vient la phase de Déploiement qui permet d'installer le service conçu, de le configurer, de le tester et de l'activer sur l'infrastructure hôte. Pendant l'exploitation du service, nous avons d'abord la phase de Provisioning qui permet la réservation et l'allocation des ressources à l'utilisateur en adaptant la configuration du service par rapport à sa commande et en allouant les ressources réseaux et équipements nécessaires à ce service. Puis nous avons la phase de Management permettant le suivi, l'analyse et le contrôle de la performance du service perçue par les utilisateurs pendant l'usage.

Ces différentes phases ne sont pas disjointes et nécessitent un échange dynamique dans un contexte NGN (*Next Generation Network*).

De plus ce contexte s'est complexifié, aujourd'hui l'utilisateur à un nouvel environnement composé de plusieurs réseaux d'accès, réseaux cœur, plates-formes de services, terminaux d'accès et mêmes d'opérateurs différents. Cet utilisateur est mobile, change de terminal, change d'environnement et désire une continuité de service en tout lieu et selon ses préférences. Une première réponse apportée à cet environnement de plus en plus complexe a été la convergence des réseaux qui permet de fournir les mêmes services à l'utilisateur à travers différents réseaux. Cette convergence de réseaux favorise la mobilité chez l'utilisateur et accroît ses besoins d'accéder à ses services, n'importe où, n'importe quand et par n'importe quel moyen (*Anywhere, Anytime and Anyhow*). Une autre réponse apportée à ces besoins est le réseau ambiant, un environnement qui permet à des utilisateurs d'accéder à des services au travers d'un ou plusieurs réseaux d'accès fixes ou sans fil (intégration des infrastructures filaires et non filaires). Le fait que ce réseau soit « ambiant » implique qu'il doit être capable de s'adapter aux préférences de l'utilisateur, à son emplacement courant, au contexte courant et aux capacités de son terminal. Quelque soit sa zone de présence, l'utilisateur accédera à des services ubiquitaires. Une nouvelle vision aussi est apparue au niveau "service" que l'on dénomme le NGS (*Next Generation Services*). Le NGS est basé sur les aspects de convergence et de personnalisation des services qui permettent d'avoir des services composés

(assemblages de différentes fonctionnalités, de différents fournisseurs) au sein d'une même session utilisateur.

Ainsi, ce nouveau contexte NGN/NGS introduit de nouveaux besoins qui sont :

- Offrir des services ubiquitaires dans différentes zones ambiantes.
- Tenir compte des préférences de l'utilisateur.
- Permettre des créations par composition dynamique de services

Avec le challenge majeur de prendre en compte dans ce contexte la gestion de la mobilité durant la session de l'utilisateur mobile. Ce qui introduit en plus les besoins suivants :

- Garantir la continuité de service quelque soit le type de mobilité.
- Maintenir la QoS quelque soit le type de mobilité.

I.2 Problématiques

Afin de satisfaire ces besoins NGN/NGS, les opérateurs doivent repenser leurs processus de gestion du cycle de vie des services.

Dans la phase de Conception, face au premier besoin d'avoir des services ubiquitaires:

- Nous pouvons déjà identifier un *premier verrou* : comment concevoir des services ubiquitaires? et quelles spécificités et propriétés doit on considérer.

En effet, Il nous faut repenser le service afin de pouvoir offrir les mêmes services ubiquitaires dans différentes zones ambiantes.

La question qui vient tout de suite après est : Quel impact ces services ubiquitaires ont ils sur la phase de Déploiement ? Comment déployer dans différentes zones ambiantes des services ubiquitaires ? Et peut-on redéployer dynamiquement pour faire face à la mobilité de l'utilisateur ? Et dans ce cas, quel est le rôle du déploiement dans la satisfaction du SLA de l'utilisateur ?

Cette dernière question constitue notre *deuxième verrou* que nous reformulons de la façon suivante : *Satisfaire un utilisateur c'est satisfaire son SLA, doit-on et comment peut-on considérer la QoS dès la phase de déploiement ?*

Ensuite vient la question de la phase d'exploitation, peut on se satisfaire d'un Provisioning de QoS statique (Fait au moment de l'ouverture de la session) quand il nous faut suivre les différents déplacements de notre utilisateur et les différents services sollicités par notre utilisateur (Ce que nous dénommons: durant l'usage).

- C'est là que se situe notre *troisième verrou* à lever : Comment faire le Provisioning des services ?

Il nous faut non seulement provisionner la QoS au niveau des ressources réseaux et équipements pour satisfaire une demande utilisateur mais prendre en compte dynamiquement la QoS du service applicatif pendant l'usage.

Mais que veut dire l'usage, en fait, c'est son besoin d'accéder aux mêmes services n'importe où et à travers n'importe quel terminal, ainsi que ses changements de préférences lors de ses déplacements.

- Nous avons là notre *quatrième verrou* : Comment tenir compte de la mobilité de l'utilisateur, de ces changements de préférences dans son environnement ambiant ?

Pour cela, il nous faut **considérer la couche service** selon la mobilité et les changements de préférences de l'utilisateur (**Service Delivery**¹), en plus, du **Média Delivery**² assuré par la couche transport, dans le but d'avoir une **continuité de service**.

Pour la phase de Management, c'est le temps de réaction posé par notre nouveau contexte qui nous intéresse. Surveiller la conformité de la QoS c'est bien, mais comment réagir à son dysfonctionnement pour répondre à l'ensemble des exigences de notre utilisateur ?

- Ce qui nous fait un *cinquième verrou* à lever : comment gérer dynamiquement la QoS pour que la solution permette le maximum de transparence à notre utilisateur (sans couture)?

Il nous faut avoir une **gestion au plus près du service ubiquitaire** pour effectuer son remplacement lors d'une dégradation de la QoS et **assurer ainsi la continuité de service**.

I.3 Contributions de la thèse

Les contributions de cette thèse constituent un ensemble de propositions de gestion de différentes phases du cycle de vie des services : la Conception, le Déploiement, le Provisioning, l'Usage et le Management afin de lever les verrous que nous venons d'identifier.

En ce qui concerne le premier verrou, c'est un travail en commun du groupe de recherche AIRS, qui a permis la définition d'un modèle de service et de ses propriétés (Stateless, Autonome, Auto gérable,...). L'autogestion repose sur un modèle de QoS dont l'agent d'exécution vérifie la conformité au contrat. Partant de cet élément de service intégrant les aspects fonctionnels et non fonctionnel nous avons repensé les différentes phases du cycle de vie et proposé des solutions pour lever les quatre autres verrous que nous venons d'identifier.

I.3.1 Déploiement Intelligent des éléments de service pour une E2E QoS efficace

Afin d'avoir une cohérence entre les différentes phases du cycle de vie du service et de faciliter les mises à jour lors des changements, nous pensons que toutes les ressources doivent être appréhendées à travers un même modèle qui régit l'ensemble. Ainsi, *la première contribution* de cette thèse qui répond au deuxième verrou que nous avons identifié porte sur un déploiement basé sur un même modèle de QoS et un même modèle de service.

¹ **Service Delivery** : utilise un ensemble de mécanismes pour contrôler et gérer la QoS au niveau service, ce qui offre une automatisation de la gestion du niveau service. Ce contrôle distribué et cette gestion dans chaque élément de service permet d'améliorer la délivrance des services, quelque soit le contexte, pour tous les éléments de service qui participent à une session de services. C'est en fait la gestion des deux extrémités (émetteur et récepteur) qui échangent le flux de données qui est géré par le "Media Delivery". Il prend donc en charge la mobilité des services qu'ils soient sur des serveurs ou dans les terminaux.

² **Media Delivery** : utilise un ensemble de mécanismes visant à fournir une mesure précise des paramètres de QoS au niveau réseau (IP) ce qui améliore la délivrance des données pour une application par session. C'est la gestion de la QoS des services de transfert d'un média à partir d'un point A à un point B.

Nous proposons un déploiement intelligent des éléments de services qui prend en compte le contrat de QoS dès cette phase. Notre but est qu'à partir de la phase de déploiement, nous répondons au SLA de l'utilisateur, en configurant le contrat que doit respecter chaque service. Ainsi, nous pouvons faciliter la phase de Management, qui, au lieu de surveiller le service et de remonter toutes les informations, va contrôler seulement la conformité de la QoS courante par rapport au contrat préétabli. Avec des éléments de service conçus autonomes, Stateless, qui s'autogèrent, composables, mutualisés et ubiquitaires, déployés dans différentes zones ambiantes, nous pouvons redéployer dynamiquement quand cela est nécessaire.

I.3.2 Provisioning de service orienté QoS

Le Provisioning de service permet la réservation des ressources réseaux et équipements selon leur QoS offerte afin de satisfaire une requête de l'utilisateur. Ces ressources étant partagées entre plusieurs requêtes, les processus de gestion du provisioning prennent en compte la QoS de ces ressources réseau et équipement avant de les allouer. Mais quand est-il de la ressource service applicatif elle-même ? Dans un contexte NGN/NGS les besoins de l'utilisateur sont de plus en plus complexes et doivent tenir compte de sa mobilité et de ses changements dans le comportement du service. Nous traduisons ce comportement par sa QoS.

Ainsi notre *deuxième contribution* qui répond au troisième verrou cité dans la problématique vise à traiter la ressource service à l'instar des ressources du réseau d'acheminement. Cette ressource service sera allouée dynamiquement en fonction de sa QoS et pourra être partagée entre plusieurs requêtes en fonction de cette QoS qu'elle offre. La gestion dynamique de cette ressource service nous permet d'anticiper les dégradations et ainsi de toujours mieux satisfaire le SLA de l'utilisateur.

I.3.3 Delivery de service

Pendant la session d'un utilisateur, des changements peuvent survenir causant des dégradations dans la QoS de bout en bout. Les approches existantes (Média Delivery) permettent la gestion du réseau de transport afin de trouver la meilleure solution qui maintient le service de transfert conformément au SLA de l'utilisateur. Mais la mobilité et l'hétérogénéité de l'environnement ambiant de l'utilisateur impose de plus en plus la nécessité d'assurer la continuité du service et le Média Delivery ne pourra pas, dans certain cas, garantir à lui seul une solution optimale.

Pour cela, notre *troisième contribution* qui répond à notre quatrième verrou vise à proposer un Service Delivery au dessus du Média Delivery afin de gérer la mobilité et les changements dans l'environnement ambiant de l'utilisateur au niveau service. Ce service Delivery est dirigé par un modèle nous permettant d'avoir un réseau de services qui s'autogère indépendamment des infrastructures de transport afin d'assurer la continuité de service tout en maintenant la QoS demandée.

I.3.4 Gestion dynamique de la QoS sans couture

L'ensemble de nos contributions de déploiement, de provisioning et de Delivery de services, va permettre de gérer le contrat de QoS et assurer ainsi une continuité de service avec le maximum de transparence. Ainsi, notre quatrième contribution qui répond à notre cinquième verrou vise à proposer une gestion dynamique de la QoS de bout en bout. La gestion est basée sur l'utilisation de nos modèles et concepts à chaque phase du cycle de vie

pour avoir un support homogène et une cohérence de réaction. Le but est d'assurer une continuité de service sans couture dans un contexte ubiquitaire et mobile.

I.3.5 Valorisations

En plus des contributions que nous venons de présenter, pour lesquelles, nous avons développé un certain nombre de composant de service, nous avons spécifié des nouveaux contextes de mise en œuvre qui valorisent nos travaux.

(a) Pour l'internet du futur

Nous proposons en plus de la gestion de la mobilité de la session de l'utilisateur par la QoS, une gestion de cette session mobile par zones ambiantes. Pour cela, nous proposons un Handover au niveau service qui permet de changer de service ubiquitaire par zone ambiante. Ce Handover Sémantique nous permet de garantir la continuité du service durant la mobilité et d'adapter la session des services au nouvel environnement ambiant de l'utilisateur, tout en tenant compte de ses préférences fonctionnelles et non fonctionnelles (QoS).

(b) Pour les standards télécom

La deuxième valorisation que nous proposons est l'intégration de la QoS dans les standards télécom. Nous nous sommes intéressés au Business Process Framework connu sous le nom eTOM (Enhanced Telecommunication Operation Map) qui est le cadre standard de description des processus, utilisé par les acteurs de l'industrie des télécoms, pour normaliser et harmoniser leurs activités. Dans l'eTOM, nous avons proposé d'intégrer la QoS dans les processus de Déploiement, de Provisioning et d'Assurance. Nous avons proposé aussi un nouveau processus de Delivery qui permet un Reprovisioning dynamique durant la mobilité.

L'ensemble des contributions que nous venons de présenter assurera la gestion de la continuité de service pendant l'usage tout en maintenant une E2E QoS du coté utilisateur, et une interaction dynamique entre les processus de gestion des services pour une E2E QoS du coté opérateur.

I.4 Organisation du rapport

Nous organisons le rapport de thèse en sept chapitres :

Le Chapitre I, est cette introduction générale qui a pour objectif de présenter les motivations et les problématiques de cette thèse. Après avoir identifié les besoins des utilisateurs d'aujourd'hui ainsi que ceux des opérateurs, nous avons identifié nos problématiques pour une gestion dynamique de bout en bout et les verrous qui s'y rapportent. Puis les contributions de cette thèse sont succinctement présentées.

Le Chapitre II, présente le contexte de nos travaux de recherche. Les processus de gestion du cycle de vie des services sont détaillés, afin de situer et de positionner les verrous qui restent à lever pour répondre au nouveau contexte NGN/NGS avec les défis de convergence des réseaux et des services, de mobilité et de réseau ambiant.

Le Chapitre III, donne une synthèse de notre étude bibliographique. Avant d'envisager les nouvelles solutions de gestion du cycle de vie des services, nous avons analysé les travaux

existants dans le Déploiement, le Provisioning et le Delivery de services, ainsi que les standards sur lesquels s'appuient les opérateurs télécom pour spécifier leur processus de gestion de services.

Le Chapitre IV, consigne nos propositions pour répondre à l'ensemble des besoins évoqués précédemment. La première (IV.1), concerne le déploiement de service où nous avons présenté comment prendre en compte le SLA de l'utilisateur dès cette phase. Comment déployer des services ubiquitaires, dans différentes zones ambiantes et comment redéployer dynamiquement, quand cela est nécessaire. Un processus de déploiement intelligent d'éléments de service basé sur la QoS est proposé. La deuxième proposition (IV.2) montre comment pré provisionner la QoS au niveau service afin de prendre en compte la mobilité et les changements de préférences de l'utilisateur. Dans la troisième proposition (IV.3), nous avons proposé un Service Delivery au dessus du Media Delivery qui permet le suivi de la session de l'utilisateur pendant la mobilité afin d'avoir une continuité de service. La quatrième proposition (IV.4) concerne un maintien de la QoS de bout en bout sans couture. Un processus de Service Delivery de bout en bout est proposé ainsi que ses interactions avec les processus de Provisioning et d'Assurance.

Le Chapitre V, consigne les différentes valorisations de nos travaux. Nos contributions pouvant aider dans l'internet du futur, nous avons proposé dans (V.1), en plus de la gestion de la session mobile de l'utilisateur selon la QoS, une gestion par zones ambiantes. Pour cela, nous avons proposé le concept du Handover Sémantique qui permet de changer d'éléments de services par zones ambiantes afin d'optimiser le bout en bout de la session dans l'environnement ambiant de l'utilisateur. Nous nous sommes aussi intéressés au Standard NGOSS (New Generation Operations Systems and Software), qui grâce à des activités collaboratives facilite l'implémentation réelle des services de télécommunications. Nous avons positionné nos résultats dans les processus de gestion des services des standards Télécom et nous avons analysé les avantages de leur intégration (V.2).

Le Chapitre VI, montre nos expérimentations sur une plate-forme composée de OpenIMSCore, le serveur d'application Glassfish V3, Oracle et des routeurs virtuels (VIRTUOR) pour présenter la faisabilité de nos propositions.

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

Chapitre II Le contexte

II.1 Introduction

Le Delivery des services de nouvelles générations (NGS) est devenu très complexe. Aujourd'hui les services convergent et sont riches en média. Ces services, qu'ils soient télécom, web et IT, utilisent des infrastructures réseaux très variées. Ils sont composés par des éléments de service qui peuvent être acquis auprès de tiers ou exposés vers des tiers. Pour être rentable dans cet environnement complexe, les fournisseurs de service doivent normaliser et automatiser les processus de délivrance de ces services à travers "leur cycle de vie" afin qu'ils puissent être créés et livrés aussi rapidement que les demandes croissantes du marché.

Par ailleurs, mis à part ces services complexes d'aujourd'hui, il faut tenir compte du fait que l'utilisateur est au centre des préoccupations des opérateurs. Dans l'environnement NGN/NGS, cet utilisateur est nomade et veut accéder à ses services sans se soucier de leur localisation ou des mécanismes mis en œuvre pour les fournir, quelque soit sa localisation et son contexte ambiant. Ces besoins de mobilité et de réseau ambiant doivent être introduits dans les processus de gestion du cycle de vie des services.

Pour compléter le paysage de la délivrance des services à l'utilisateur, l'opérateur doit maintenir le SLA contracté avec l'utilisateur en assurant une QoS de bout en bout dans la réservation des ressources "services applicatifs", des ressources "réseaux" et des ressources "équipements". Cet aspect de "User Centric" et de la QoS de bout en bout à maintenir durant la mobilité et les changements de préférences de l'utilisateur doit aussi être pris en compte dans le delivery des services.

C'est dans ce contexte que se situe la thèse. La Figure II.1-1, représente nos trois domaines avec comme objectif le " Service Delivery" correspondant à l'usage de l'utilisateur.

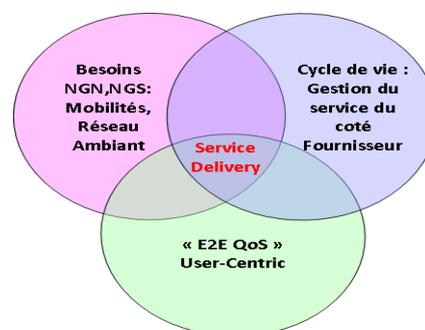


Figure II.1-1 : Périmètre d'étude de la thèse

Ainsi, nous présentons dans ce chapitre notre contexte avec le cycle de vie des services (§II.2) pour identifier les étapes clés que nous devons repenser en fonction des nouveaux besoins NGN/NGS, qui regroupent la mobilité et le réseau ambiant (§II.3), sans oublier l'aspect User-Centric et la QoS de bout en bout de la vision de l'utilisateur (§II.4). À la fin de ce chapitre et suite à l'analyse du contexte nous aurons le périmètre de cette thèse pour un delivery de service (§II.5).

II.2 Cycle de vie de service

ITIL [ITIL] définit un service comme étant « l'ensemble des moyens mis en œuvre pour apporter de la valeur pour un client, sans que celui-ci n'en supporte, ni les coûts, ni les risques spécifiques au service ».

En conséquence, un contrat client-fournisseur est établi. Afin de fournir ce service au client selon le contrat établi, toutes les phases de son cycle de vie qui vont de sa conception jusqu'à son exploitation doivent être bien définies et les objectifs bien différenciés.

Selon les méthodologies auxquelles on se réfère dans eTOM [TMF] [GB921a] et SDF [TR139], nous trouvons un découpage plus ou moins fin de ces phases du cycle de vie d'un service, nous retenons les cinq phases suivantes :

La phase de Stratégie, qui permet d'identifier le partenaire, de créer le modèle métier rentable et permet l'accès des partenaires aux ressources de l'opérateur en fonction du contrat business spécifique contenant des SLA, le partage de revenu, etc.

La phase de Développement, qui est essentiellement le temps de conception où les exigences reçues sont modélisées en services, testées et les artefacts développés. Cette phase utilise le catalogue de services pour les services réutilisables ainsi que les solutions appropriées pour inclure les préférences des clients et leurs profils, les règles métier, les moteurs de personnalisation et de conseils, et d'autres aspects qui seront évalués au moment de l'exécution du service.

La phase de Déploiement, où la transition des services est effectuée de l'environnement de test vers l'environnement de pré-production et enfin vers l'environnement de production lorsque l'acceptation conformément aux attentes de l'opérateur est validée.

La phase d'Exécution, qui est essentiellement l'*Usage* lorsque l'utilisateur final consomme les services déployés. Cette phase n'est pas traitée dans eTOM en tant que processus à part mais la gestion de la QoS est prise en compte dans la phase d'Assurance.

La phase d'Exploitation (regroupe dans eTOM la phase de Fullfilment, d'Assurance et de Billing), concomitante avec l'exécution, où les services sont gérés conformément à la QoS, facturés selon l'utilisation répartie entre tiers, selon les règles de partage dans le cas où les éléments de service dans la composition appartiennent à des tiers. Cette phase comprend aussi des rapports sur la QoS, l'*Usage* en corrélation avec l'utilisation des capacités de ressources réseau sous-jacentes pour fournir des informations utiles sur les coûts, revenue et qualité.

La phase de Retrait, qui couvre le retrait des services.

Dans ces différentes phases, quelles sont celles qui sont impactées par les nouveaux besoins NGN/NGS et comment la QoS E2E est-elle prise en compte ?

Pour répondre à ces questions, nous allons d'abord présenter ces nouveaux besoins.

II.3 Le contexte NGN/NGS

Le domaine des réseaux et services de nouvelle génération (NGN/NGS) est caractérisé par les évolutions suivantes:

- *La convergence* : La convergence de réseaux est le fait de pouvoir accéder à un même service par différents réseaux d'accès (§II.3.1). La convergence des services permet d'avoir des compositions flexibles et dynamiques de services hétérogènes dans une même session utilisateur (§II.3.2).

- *La mobilité* : qui recouvre tout type de déplacement (§II.3.3).

- *L'organisation en zone ambiante* : qui permet à l'utilisateur de profiter des ressources ambiantes pour optimiser la gestion de sa session (réseaux ambiants, services ambiants, terminaux ambiants) (§II.3.4).

II.3.1 La convergence des réseaux

Le NGN évolue de plus en plus ces dernières années avec différents types d'accès aux services (UMTS, xDSL, WLAN, Ethernet longue distance, etc.), différents réseaux cœur et différentes plates-formes de service. Une première réponse apportée pour répondre à la complexité de cet environnement NGN est la convergence des réseaux. Cette dernière, permet d'accéder à un même service par différents réseaux d'accès et un même réseau cœur.

Cette convergence permet le passage d'une architecture verticale (où le réseau d'accès, le réseau cœur et la plate-forme de service sont fortement couplés) vers une architecture horizontale [ETSI09a] [ETSI09b] [ETSI09c] (où le réseau et la plate-forme de service sont sur des couches indépendantes).

Dans un contexte mobile, la gestion de la session avec des changements de réseaux d'accès doit être prise en compte durant l'usage. Nous avons différentes techniques de handover au niveau du réseau, qui permettent de maintenir l'accès au service pendant les changements de réseaux.

II.3.2 La convergence des services

Il existe différents types de services, nous avons :

Les services IT (SOA), qui sont un support à l'offre de services de l'opérateur, ces services participent à une meilleure diffusion de l'offre (catalogue électronique, site web d'information), au développement de l'usage, à un meilleur suivi du service après vente (gestion des réclamations), au contrôle des coûts, etc.

Les services Web, qui constituent une infrastructure technologique distribuée faite pour assembler des systèmes hétérogènes en un seul et même modèle logique à travers un réseau internet. Les services web sont variés. Ils peuvent être un service spécifique (une simple requête telle que la conversion d'une monnaie suivant le taux de change), un processus métier, une application (la constitution d'un voyage organisé), une ressource du système d'information, etc.

Les Services Web et l'architecture orientée service (SOA) autorisent un nouveau type de pensée métier d'émerger. La logique métier globale par l'assemblage de composants logiciels utilisables par plusieurs utilisateurs (mutualisation).

Les services Telco, qui sont les services offerts par les opérateurs de télécommunication aux utilisateurs. Nous avons des services de télécom de base comme la téléphonie et des services de télécom à valeur ajoutée comme par exemple la visioconférence.

Aujourd'hui l'utilisateur veut une personnalisation de ses services, il veut accéder à des services avec des caractéristiques différentes dans une même session, que ces services soient IT, Web ou Telco. L'assemblage de différents types de services pose le défi d'avoir une convergence des services possible pour l'utilisateur.

II.3.3 La mobilité

Avec les réseaux et services de nouvelle génération, les utilisateurs peuvent se déplacer au grès de leurs besoins et de leurs préférences sans aucune limitation. Mais l'hétérogénéité des réseaux, des terminaux et des services fait que la continuité du service sans couture est de plus en plus complexe à gérer durant la mobilité. Nous analysons les différentes mobilités qui peuvent survenir durant la session de l'utilisateur et leur impact sur la continuité de la session. Nous prenons en compte trois types de mobilité qui sont la mobilité du terminal (§a), la mobilité de l'utilisateur (§b) et la mobilité du service (§c).

(a) La mobilité du terminal

La mobilité du terminal signifie qu'un terminal bouge et change de localisation. Pour gérer ce changement de localisation tout en maintenant la communication il existe deux solutions :

La première est le Handover qui permet de basculer le terminal d'un point d'attachement vers un autre point d'attachement ayant de meilleures conditions pour maintenir la communication. La deuxième est le Roaming qui permet au terminal mobile de passer automatiquement sur le réseau d'un opérateur partenaire de son opérateur d'origine quand ceci est nécessaire (par exemple changement de pays) et ainsi d'utiliser les services de cet opérateur selon les contrats établis.

Ces deux solutions peuvent entraîner des perturbations dans l'accès aux services et ne peuvent assurer une continuité sans couture.

(b) La mobilité de l'utilisateur

Les terminaux des utilisateurs aujourd'hui sont très diversifiés et la mobilité de l'utilisateur désigne le changement de terminal tout en maintenant la continuité de service. Le processus de reprovisioning durant cette mobilité doit adapter les services utilisés durant la session, sur le nouveau terminal utilisé, tout en tenant compte des préférences de l'utilisateur.

(c) La mobilité du service

La mobilité de service permet à l'utilisateur d'obtenir un service à partir d'une instance la plus appropriée, assurant la même fonctionnalité et le même contrat de QoS. Dans notre contexte, cette instance est un service ubiquitaire qui sera sélectionné selon la localisation, le réseau d'accès et le terminal de l'utilisateur. Les fournisseurs de service doivent déployer des services ubiquitaires assurant le provisioning des utilisateurs "anywhere, anyhow et anytime".

Ainsi, les fournisseurs doivent prendre en compte ces différentes mobilités dans leurs processus de gestion du cycle de vie afin de maintenir le contrat de QoS de bout en bout (dans toutes les étapes du cycle de vie) et de répondre ainsi aux nouveaux besoins NGN/NGS des utilisateurs.

II.3.4 Le réseau ambiant

La vision de réseaux ambiants signifie que les opérateurs et les utilisateurs peuvent exploiter conjointement les ressources réseaux disponibles pour un large éventail de services. Avoir accès aux ressources ambiantes peut avoir beaucoup d'avantages parmi lesquels l'optimisation de « la latence », le maintien du SLA de l'utilisateur, la possibilité pour l'utilisateur, d'avoir différentes préférences dans différents contextes (bureau, maison, vacances, etc.) et d'avoir plusieurs possibilités d'accès aux services liées à la localisation, à l'équipement ou au type d'opérateur. Cette thèse prend en considération ce contexte de ressources ambiantes, pour adapter la session de l'utilisateur. Ainsi, l'utilisation des ressources ambiantes nous permet d'éviter la latence réseau et de maintenir le SLA demandé.

II.4 User centric et E2E QoS

Le principe de « User-centric » signifie que le système complet est à l'écoute des besoins et des préférences de l'utilisateur. Bien que nous puissions penser que cela est toujours le cas, il n'en est rien ! En effet, dans la phase de l'analyse des besoins nous tenons compte, bien sûr, de ceux de l'utilisateur, mais nous concevons et réalisons ensuite des solutions selon d'autres approches. Nous trouvons l'approche « Système-centric » dans laquelle l'utilisateur doit se conformer aux contraintes de traitements du matériel, bien qu'aujourd'hui nous levons ces contraintes avec les machines virtuelles (VM). Nous avons aussi les approches « Réseau-centric » qui impliquent que le réseau soit au centre de l'architecture et conditionne toutes demandes de services. Et enfin, pour la grande majorité des solutions nous trouvons les approches « Application-centric » qui se concentrent sur l'application et la considèrent comme le point focal, avec des architectures "client - serveur".

Dans une approche « User-centric », l'utilisateur a une accessibilité de service par n'importe quel terminal (Adaptation + préférences), une personnalisation de service par la composition de service dynamique et flexible. Tout ceci durant une unique session, même si les services sont différents.

Durant l'usage, cet utilisateur se déplace et personnalise ses services sans aucune limite, ainsi sa session est dynamique avec des services ubiquitaires qui remplacent ceux faisant partie de la session, qui s'ajoutent à la session ou qui changent selon le contexte ambiant, les réseaux d'accès et les terminaux changent aussi durant cette session mobile, pour cela sa QoS E2E (SLA) durant sa session mobile doit être maintenue. Ayant un même modèle appliqué aux différents acteurs (équipement, réseau et services), qui ont des rôles complémentaires nous permet d'assurer la QoS E2E.

II.5 Périmètre d'étude de la thèse

Pour avoir un environnement de "Delivery de service" intelligent qui intègre toutes les fonctions nécessaires au cycle de vie d'un service, nous devons nous adapter aux évolutions du marché, aux préférences et à la mobilité des clients avec une flexibilité croissante.

Ainsi, notre périmètre d'étude dans cette thèse se situe à l'intersection des trois domaines que nous venons d'analyser (cercles montrés sur la Figure II.1-1) pour avoir ce "service delivery" qui satisfait aux utilisateurs et aux opérateurs.

L'analyse que nous venons de faire conduit à prendre en compte :

- Les différentes mobilités durant l'usage de service.

- La convergence des réseaux et des services.
- Les préférences de l'utilisateur et sa personnalisation du service.
- Considérer le contexte ambiant de l'utilisateur pour l'allocation des ressources.
- L'approche "User Centric".

Les phases du cycle de vie qui sont impactées par ces besoins sont le déploiement et l'exploitation (Figure II.5-1). Dans cette thèse, basée sur les travaux de recherche de notre groupe sur la phase de conception des services où un modèle de service a été défini, nous nous focalisons sur les phases de Déploiement et la phase d'Exploitation qui regroupe les étapes de Provisioning, d'Usage et de Gestion pour spécifier les processus de gestion dynamique dans le contexte NGN/NGS.

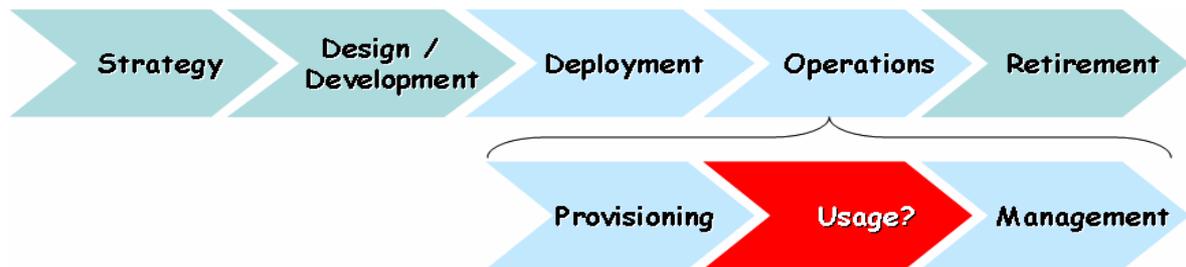


Figure II.5-1 : Cycle de vie du service.

Avant de présenter nos propositions, nous étudions dans le chapitre suivant les travaux existants sur les champs de recherche que nous venons d'identifier, à savoir :

Pour la gestion de services du côté fournisseur durant le cycle de vie, les solutions existantes dans le déploiement, le provisioning et le delivery des services ainsi que les standards de télécommunications (TMN au NGOSS).

Pour la gestion de la mobilité, les techniques existantes du handover et du roaming.

Chapitre III État de l'art

Dans le chapitre précédent, nous avons présenté le contexte dans lequel se situe notre recherche. L'objectif de cette thèse est d'avoir une gestion dynamique durant tout le cycle de vie des services, du déploiement jusqu'au Delivery. La considération de la mobilité et de l'ubiquité de service afin d'assurer la continuité de service et le maintien de la QoS dans un contexte NGN/NGS nous a conduit à repenser les processus de gestion de service. Nous présentons d'abord les travaux existants dans le déploiement de services (§III.1), dans le provisioning de services dans le SON, dans SOA (§III.2) et dans le Media Delivery (§III.3). Puis dans (§III.4) nous analysons les techniques de handover dans le réseau d'accès et le réseau cœur. Dans la section (§III.5) nous retraçons ce qui existent dans les standards de gestion des télécommunications en partant du TMN jusqu'au NGOSS. Enfin, une conclusion (§III.5.4) est faite pour résumer ce chapitre.

III.1 Les solutions existantes pour le Déploiement de services

Dans le but d'avoir une gestion efficace du déploiement, les fournisseurs de service doivent déployer leurs services dans la plate-forme qui satisfait à la demande du service et d'avoir assez d'éléments de service ubiquitaires pour satisfaire le SLA des utilisateurs nomades. Le déploiement de service consiste à trouver les ressources disponibles et accessibles, et de déployer et configurer le service sur ces ressources de sorte qu'ils puissent répondre à la QoS désirée. Nous présentons quelques uns des travaux existant dans le déploiement de service.

Projet OPUCE (Open Platform for User-centric service Creation and Execution) [JCY07] : est un projet intégré, 6^{ème} Framework Programme, visant à développer une plate-forme complète pour la création, le déploiement et l'exécution des services centrés sur l'utilisateur (User-Centric). L'objectif de la plate-forme OPUCE est de mettre l'utilisateur final au centre du cycle de vie entier d'un service, pour le laisser devenir en même temps le créateur de services personnalisés, leur déployeur, leur gestionnaire et leur consommateur. Dans OPUCE, le déploiement est basé sur « la description du service » qui contient toutes les données nécessaires pour déployer le service, ainsi le provisioning de service est effectué automatiquement en utilisant la description du service pour spécifier les détails de provisioning pour chaque service.

Avec cette approche, les concepts de service sont redéfinis. Le service est créé par l'utilisateur en orchestrant des services de base créés et déployés par l'opérateur. Ces services de base sont considérés comme des unités fonctionnelles déployées par l'opérateur ou des partenaires autorisés, disponibles dans la plate-forme OPUCE et offertes aux utilisateurs finaux à travers des interfaces web.

Dans ce projet les services sont composés dynamiquement selon le contexte et les préférences de l'utilisateur, par exemple l'utilisateur compose un service mail intelligent qui

transmet un mail entrant en fonction de la situation de l'utilisateur et du terminal utilisé, par exemple, si l'utilisateur est en train de conduire, le mail est transmis via un Text-to-Voice, si l'utilisateur n'est pas connecté à son PC de bureau, le service envoie un SMS contenant le sujet du mail et l'expéditeur. Les services de base utilisés ici par l'utilisateur sont un service de messagerie, service SMS et un service Text-to-Voice et un service de contexte utilisateur. Le déploiement est intéressant du fait qu'il soit basé sur une composition de service dynamique. C'est à dire que la composition peut changer selon les préférences de l'utilisateur et son contexte prédéfini. Mais quand l'utilisateur est mobile, cette composition ne change pas afin de maintenir la QoS demandée. Elle ne se recompose pas suivant les possibilités des zones ambiantes et la prise en compte des services de base proposés par les autres fournisseurs. Alors que les utilisateurs d'aujourd'hui sont mobiles et veulent une assurance de la continuité du service et un maintien de la QoS durant le déplacement. Ce maintien de la QoS ne peut être obtenu que si l'on prend en compte le comportement du service, traduit par sa QoS et de pouvoir dynamiquement changer les services faisant partie de la composition de service par d'autres services ubiquitaires.

Pour l'aspect mobilité et adaptation selon les événements, nous allons trouver une réponse avec la proposition de D. Ayed et al. qui ont proposé dans [DA08] le déploiement d'applications basées sur des composants placés dans un middleware sensible au contexte. Le service de déploiement sensible au contexte proposé permet l'installation et la reconfiguration d'une application selon le contexte. Nous avons, par exemple selon la mobilité de l'utilisateur (Détection d'une nouvelle localisation géographique de l'utilisateur) ou un changement de la bande passante dans le réseau, le service de déploiement va adapter la composition de l'application en ajoutant des composants ou en changeant des composants. Par exemple un utilisateur qui utilise son PC, aura un composant adapté à un large écran, puis si l'utilisateur utilise un PDA, ce composant va être changé par un autre plus adapté à cet équipement (à petit écran).

Le middleware permet l'interaction avec le contexte en fournissant un méta-modèle des informations du contexte et un ensemble de composants en mesure de collecter et d'analyser les informations du contexte. Le service de déploiement va décrire son contexte pertinent en utilisant le méta-modèle fourni par ce middleware. Lorsqu'un changement pertinent survient, le service de déploiement est notifié dans le but d'appliquer son processus d'adaptation.

La solution proposée pour déployer une application composée, utilise une composition dynamique qui change selon la mobilité de l'utilisateur, son changement de terminal ou de préférences. Ces informations sur tout type de changement sont récupérées par des agents, qui surveillent un type de changement et notifient après, le composant, qui a souscrit à ce type de changement.

L'aspect ubiquitaire nous le trouvons avec D. Hoareau and al. qui ont présenté dans [DH08], un support middleware pour le déploiement de composants logiciels ubiquitaires.

Les composants composés sont ubiquitaires dans le sens où ils sont mis à disposition sur un ensemble d'hôtes alors que chaque composant primitif est localisé sur un seul hôte. En outre, via la notion d'interface active, ils peuvent exécuter le composant dans un mode intégré pour prendre en compte la déconnexion du réseau sans pour autant avoir toute l'application inutilisable. Ils utilisent un langage descriptif pour spécifier le descripteur de déploiement qui permet un déploiement sensible au contexte (contexte aware). Le descripteur de déploiement (Figure III.1-1) permet la description des besoins en ressources du composant et de contraintes de placement.

Les composants sont déployés sur différentes plates-formes selon les contraintes en ressources et en localisation de l'application composée.

<pre> 5 <component name="DocumentSearch"> <component name="DocumentFinder"> <deployment-context> <resource-constraint> <cpu freq="1.2" unit="GHz" operator="min" /> </resource-constraint> 10 <location-constraint> <target varname="x"/> </location-constraint> </deployment-context> 15 </component> <component name="DocumentBuffer"> <deployment-context> <resource-constraint> 20 <memory free="200" unit="MB" operator="min" /> </resource-constraint> <location-constraint> <target varname="y"/> 25 </location-constraint> </deployment-context> </component> <deployment-context> 30 <location-constraint> <operator name="alldiff"> <arg varname="this.DocumentFinder.x" /> <arg varname="this.DocumentBuffer.y" /> </operator> 35 </location-constraint> </deployment-context> </component> </pre>	<pre> 40 <component name="PhotoApp"> <component name="DiapoMaker"> <deployment-context> <resource-constraint> <cpu freq="1.5" unit="GHz" operator="min" /> <memory free="50" unit="MB" directory="/home/" operator="min"/> 45 </resource-constraint> </deployment-context> </component> 50 <component name="PhotoRepository"> <deployment-context> <resource-constraint> <memory free="1" unit="GB" directory="/home/" operator="min" /> 55 </resource-constraint> </deployment-context> </component> 60 <component name="DocumentSearch"> <locationconstraint> <operator name="exclude"> <arg value="egilsay" /> <arg value="parvati" /> 65 </operator> </locationconstraint> </component> 70 <deployment-context> <locationconstraint> <target hostname="ambika"/> <target hostname="dakini"/> <target hostname="mafate"/> <target hostname="egilsay"/> <target hostname="parvati"/> 75 </locationconstraint> </deployment-context> </component> </pre>
(a)	(b)

Figure III.1-1: Descripteur de déploiement

Mais nous n'avons pas trouvé de proposition qui prenait en compte la QoS E2E et qui automatisait les redéploiements lors d'un dysfonctionnement de cette dernière. Or il nous semble capital de traiter la QoS dès la phase de déploiement et non pas seulement en phase d'exécution. Cela semble d'autant plus légitime, que nous manipulons des éléments de service et non pas des applications monolithiques.

III.2 Les solutions existantes pour le Provisioning de services

III.2.1 Introduction

L'un des défis à relever dans le traitement d'une demande utilisateur est la façon d'effectuer une réservation dynamique et efficace des ressources, et de pouvoir s'adapter aux changements de contexte de telle sorte que la QoS offerte peut toujours satisfaire les exigences de l'utilisateur. Le Provisioning de ressources est le processus qui recherche les ressources disponibles et accessibles, les réserve et les affectent afin de répondre aux besoins de la demande en cours. De nombreux travaux de recherche ont été effectués dans le provisioning des ressources réseaux et équipements (Media Delivery).

Dans le contexte NGN/NGS, la QoS est introduite dans tout type d'application et devient de plus en plus importante. Les exigences de l'utilisateur au niveau applicatif sont de plus en plus complexes. Ainsi, la gestion des ressources doit être plus flexible, ce qui fait que le provisioning des équipements et réseaux ne sont plus suffisants et que le provisioning des services est nécessaire pour faire face aux exigences des utilisateurs d'aujourd'hui. Les travaux de recherche ont proposés des approches dans les décisions de provisioning pour permettre aux fournisseurs de service d'effectuer une distribution dynamique des ressources réseaux et équipements entre les services et les SLAs. Cependant, afin d'offrir une continuité de service sans couture et une QoS stable durant tous types de mobilité, il est aussi nécessaire de provisionner les services applicatifs eux mêmes au niveau service, et de maintenir la QoS en la considérant au niveau applicatif.

Nous analysons les solutions apportées par le SON (§III.2.2) et par SOA (§III.2.3) dans le provisioning des services.

III.2.2 Solutions de provisioning de services apportées par le SON

Afin que l'utilisateur puisse accéder à ses mêmes services par différents réseaux hétérogènes, le Service Overlay Network (SON) [JWK10] a été proposé comme une couche intermédiaire qui fait converger les différents réseaux pour l'accès aux services.

Le SON permet d'apporter des solutions aux problèmes de QoS dans l'internet d'aujourd'hui et de faciliter la création et le déploiement de services à valeur ajoutée comme la VoIP, le VoD ainsi que d'autres services sensibles à la QoS. Le SON achète de la bande passante avec une certaine QoS garantie à partir de domaines de réseaux individuels via un SLA bilatérale pour construire un overlay de Media Delivery de bout en bout au dessus des réseaux de transport de données existants. Via le contrat de service, les utilisateurs payeront directement le SON pour l'utilisation des services fournis par lui.

Ainsi, la solution apportée par le SON permet le provisioning de la QoS pour un service donné dans le réseau comme par exemple la bande passante, mais pas la QoS des composants de service eux mêmes ni celle d'un service Delivery (Mobilité de service, par exemple).

Comme les réseaux de données génériques ont des limitations pour supporter certains types de services, SATO (Service-aware Adaptive Transport Overlay) [MS] permet le Provisioning de services à valeur ajoutée en formant un réseau overlay pour le transport de données avec un plan de signalisation et de control séparé. Quand une mobilité se produit, SATO gère une mobilité de session en réorganisant les ressources ambiantes pour adapter la

fourniture de service. Cependant, il manque de flexibilité et ne prend pas en compte tous les aspects (critères) de QoS dans la phase de Provisioning.

QSON (QoS-aware SON) [MS07] divise SATO en plusieurs sous-réseaux et chacun correspond à un niveau de QoS. Ainsi, la fourniture du service peut assurer une certaine QoS durant la mobilité. Cependant, la granularité du sous-ensemble de QoS équivalent n'est pas assez flexible pour avoir une adaptation dynamique quand une dégradation de QoS se produit.

DSO (Dynamic SON) [EKK09] vise à composer une plate-forme de Delivery de services spécifiques dynamiquement avec des composants de service distribués pour un provisioning de service fiable tout en permettant la mobilité et quel que soit la technologie de transport sous-jacente. Une session de services peut être créée dans le DSO et elle peut être gérée dynamiquement. En outre, le DSO est responsable du provisioning de la QoS. Cependant, la QoS ne peut être adaptée qu'à la couche réseau transport, il est dommage que dans un environnement de plusieurs fournisseurs de service, une QoS de services ubiquitaires n'est pas prise en compte et ne joue pas de rôle dans le provisioning dynamique.

III.2.3 Solutions de provisioning de services apportées par SOA

Le Service Oriented Architecture (SOA) [TP06] permet de composer des services distribués à couplage lâche indépendamment de la localisation. C'est néanmoins une architecture dont la gestion reste verticale et bien que le provisioning des ressources soit basé sur le SLA/QoS, il ne tient pas compte du service applicatif en tant que ressource à part entière.

M. Boniface et al. proposent dans [MB07] un provisioning de services dynamique en utilisant les SLAs pour maximiser l'utilisation des ressources. Trois questions essentielles sont abordées. La première est la fourniture de services multiples avec de multiples SLAs. Le fournisseur de services doit décider la façon de distribuer ou de partager des ressources, ce qui nécessite un système assez complexe. La seconde, est la stratégie de gérer les violations de SLA. Ils ont proposé de déployer dynamiquement les services applicatifs pour anticiper les violations de SLA. La dernière question est le mapping des paramètres de SLA en des paramètres mesurables. Ils proposent une définition de ce mapping et de stocker la définition dans une base globale. Cette solution est basée sur l'évolution prévue des caractéristiques de QoS de l'équipement et du réseau pour anticiper les violations de SLA, *mais elle ne tient pas compte de la QoS courante de l'élément de service*. L'un des défis à relever pour une gestion efficace des ressources est de pouvoir anticiper les violations de SLA par la surveillance dynamique de la QoS courante de tous les composants du système.

X. Li et al. proposent dans [XL08] une plate-forme de gestion de la QoS dirigée par le SLA, qui gère différentes ressources distribuées et s'adapte aux changements dynamiques de la disponibilité des ressources pour répondre aux besoins de QoS des utilisateurs. Le cycle de vie complet inclut la découverte de service, la négociation de QoS, la réservation et l'allocation de ressources, l'assurance et l'adaptation de la QoS. Cependant on ne trouve pas de reprovisioning dynamique pour prendre en compte le contexte NGN/NGS et pour assurer la continuité de service durant la mobilité.

S. Chen et al. proposent dans [SC08] un Context-Aware middleware de gestion de ressources dynamique pour les applications orientées service, qui vise à traiter la dynamique

inhérente aux équipements locaux et au réseau. Basé sur un environnement de coopération de services sécurisés, fourni en regroupant les services dans des communautés virtuelles, une architecture de surveillance inter-couches est désignée pour rassembler les statistiques de performance des services et du réseau. Le service de gestion des ressources est indépendant de la plate-forme. Cette solution prend en charge, de bout en bout le provisioning de la QoS, la coopération de service étendue, la confidentialité et le contrôle complet de chaque fournisseur de service sur ses ressources sous-jacentes. Mais là aussi, les ressources considérées sont seulement les ressources réseau et équipement. Est ce suffisant pour l'utilisateur ?

Pour terminer, nous avons trouvé les travaux de B. Urgaonkar et al intéressant. Ils proposent dans [BU08] un provisioning de capacité dynamique pour les applications avec des charges de travail qui varient dynamiquement. La solution emploie un modèle de file d'attente flexible afin de déterminer quelle quantité de ressources allouer à chaque niveau de l'application, et une combinaison de méthodes réactives et prédictives qui déterminent quand provisionner ces ressources. De cette façon, ils ont besoins de concevoir avec attention la façon de décrire complètement la QoS, ce qui n'est pas complètement spécifiée dans leur travail.

III.2.4 Discussion

Aujourd'hui, de plus en plus de chercheurs commencent à séparer les services applicatifs des services réseaux (la couche transport), dans le but d'être plus adapté à la dynamique de la QoS durant l'usage des services. Cependant, les architectures de services (SON ou SOA) sont encore verticales, et le provisioning est actuellement réalisé par le niveau réseau avec un respect du SLA demandé par le service. En fait, c'est le réseau qui est responsable de la fourniture d'une QoS conforme à la demande tout en prenant en compte les caractéristiques du service, mais il n'y a pas de réservation effective des éléments de services eux mêmes au niveau service. Actuellement, dans le contexte NGN/NGS, nous commençons à avoir des services ubiquitaires, c'est pourquoi ce type de réservation est en mesure d'être réalisé. Par ailleurs, dans un environnement multi fournisseurs de services, la possibilité de changer de plate-forme de services ouvre la trans-organisation à l'utilisateur. Les solutions existantes effectuent le reprovisioning en changeant le chemin du média Delivery pour adapter le changement de service mais, il n'y a pas de propositions de reprovisioning qui travaillent sur le niveau service et qui assurent la continuité du service sans couture pendant la mobilité. C'est pourquoi ce que nous proposons est d'avoir un Service Delivery au dessus du Media Delivery. Ce Service Delivery travaille dans la couche service et nous permet de provisionner la ressource service applicatif en fonction de sa QoS courante. Il nous permet également de reprovisionner dynamiquement pendant la mobilité ou d'autres changements dans l'environnement ambiant de l'utilisateur.

III.3 Les solutions existantes pour le Delivery

III.3.1 Introduction

Beaucoup de travaux de recherche existent pour traiter le Média Delivery dans la couche réseau. Mais aujourd'hui les besoins de plus en plus croissants d'avoir des services ubiquitaires performants et accessibles pour l'utilisateur quelque soit sa localisation et ses

préférences ne peuvent être complètement satisfait par les solutions apportées par la couche réseau. Ainsi, des solutions apportées par la couche service sont proposées pour répondre aux nouveaux besoins.

Comme nous venons de le voir, le SON est proposé comme solution et vise à accroître l'adaptation de l'application et la fiabilité du service. Toutefois, pour fournir les services aux utilisateurs mobiles quelque soit leur environnement ambiant, la gestion de la composition de service, le provisioning flexible des ressources et la garantie de la QoS restent des défis à relever. En fait, les ressources réservées dans le SON sont confinés à des éléments du réseau comme la bande passante, ou à un domaine intermédiaire qui ne peut fournir la QoS demandée [JWK10]. Dans cette section, nous analysons les travaux de recherche existants pour optimiser le SON (§III.3.2), afin de discuter des challenges actuels du SON et de ses lacunes pour la continuité de service (§III.3.4).

III.3.2 Travaux de recherche existants pour optimiser le SON

(a) Politiques de reconfiguration de topologie dans le SON

Comme la reconfiguration est l'une des caractéristiques les plus importantes du SON, J. Fan et al. proposent des politiques de reconfiguration optimales pour la conception de topologie du SON, afin d'améliorer la performance et de minimiser le coût des opérations de routage dynamique [JF06]. Puisque les exigences des communications changent, la topologie peut nécessiter une reconfiguration avec un chemin d'accès à moindre coût. En comparaison avec d'autres travaux, il est considéré dans le cas général que la topologie n'est pas en full-mèche mais bornée par degré : le nombre de voisins d'un nœud est limité. Ils ont analysé et identifié tous les types de coûts durant la phase de reconfiguration, ont observés la reconfiguration optimale de petits systèmes, et ensuite donné les politiques de reconfiguration pour de grands systèmes où le modèle de communication est dynamique.

Les politiques sont vérifiées pour avoir un moindre coût face à des changements des exigences de communication. Comme les nœuds dans le graphe de topologie sont à fixer et leurs nombres sont limités, il est difficile de découvrir des nœuds services dans un autre domaine (inconnu) en dehors du SON existant. Les politiques peuvent résoudre ainsi certaines mobilités mais dans une portée limitée.

(b) Un routage optimal basé sur la fédération de service

M. Wang et al. ont proposés un algorithme distribué [MW96] du graphe de flux de service basé sur la fédération de service, afin d'optimiser le routage dans le SON. La fédération de service permet à des services indépendants d'effectuer les tâches dans un ordre soit séquentiel, soit parallèle ou encore entrelacé en permettant d'avoir des chemins parallèles pour un service complexe.

Pour connecter l'exigence de service et le graphe overlay, ils ont défini un graphe de service abstrait, dans lequel un nœud de service abstrait est rempli avec les instances du service correspondant. Après l'application de l'algorithme proposé par Z. Wang et al. dans le graphe overlay, un graphe de service abstrait est construit en respectant les stratégies proposées, puis le même algorithme est appliqué à nouveau dans le graphe abstrait, de sorte que certains liens du chemin précédent peuvent être remplacés par d'autres plus optimaux.

Cet algorithme de fédération de service a prouvé sa capacité à construire des graphes de flux de service de haute qualité. Bien qu'il réponde dynamiquement dans la couche service, l'algorithme n'est pas assez souple face à l'environnement ambiant qui est très hétérogène et

mobile. Par ailleurs la continuité de service durant le processus de routage dynamique, n'est pas prise en compte. Or, c'est l'une des exigences de l'utilisateur dans le contexte NGN/NGS, supposée être comme une première priorité.

(c) Composition de service en assurant la QoS

X. Gu et al. ont présenté une infrastructure pour une composition de service efficace, en se concentrant sur l'assurance de la QoS [XG03], ce qui est en faveur de la fiabilité dans le SON. Ils ont proposé un algorithme avec une QoS assurée pour le mapping à partir d'un template de service, constitué à partir des exigences de service, pour un chemin de service instancié.

Pour l'initialisation de la composition de service, l'algorithme de base génère le graphe pondéré des candidats dans la première étape (à la place de rechercher le chemin), puis exécute l'algorithme de Dijkstra pour trouver le chemin le plus court en fonction des différentes contraintes de QoS. Afin d'examiner chaque contrainte individuelle de QoS, un algorithme amélioré est proposé en plus, qui peut changer dynamiquement le poids de différents facteurs. Enfin, les algorithmes de composition de service dynamique sont proposés pour faire face à la recomposition du chemin de service de manière complète ou d'une manière partielle, au moment de l'exécution lorsqu'une panne ou une dégradation importante se produit.

La proposition est dynamique et capable de fournir l'assurance de la QoS et le "Load Balancing" pour les services composés dans le SON. Comme l'algorithme considère les nœuds et les liens dans la topologie entière, il répond aux scénarios de mobilité. Pour la recomposition considérée du point de vue de l'optimisation de la topologie, par exemple, un modèle général peut être appliqué pour améliorer la flexibilité et l'efficacité sans augmenter la complexité.

(d) Auto-configuration dans le SON

R. Braynard et al. ont présenté Opus (Overlay Peer Utility Service), qui peut automatiquement configurer des Overlays de serveur réseau en exécutant des instances de ressources globales et suivant leurs statuts, dans le but d'améliorer la performance et en évitant des ré-implémentations inutiles [RB02]. Un Overlay de service fonctionne comme un « Backbone », à travers lequel Opus peut suivre et diffuser l'information. Pour chaque application, Opus crée un overlay, et adapte dynamiquement la topologie overlay selon la charge du système et les conditions réseau. Opus détermine l'allocation des ressources pour des applications concurrentes, par le maintien de la structure cellulaire, dans lequel, une cellule peut être un Opus entier présent ou une partie de celui-ci. Pour suivre les caractéristiques du système, ils ont appliqué à la bibliothèque de communication générique dans Opus, certains protocoles de routages existants qui ont les propriétés d'agrégation, de hiérarchie et d'approximation. Enfin pour assurer la fiabilité et la disponibilité, plusieurs approches sont suivies, telles que les flux restreints et la construction de chemins disjoints.

C'est une bonne idée de définir un module de gestion spécial pour implémenter l'auto-configuration. Cependant, il est très centralisé, le module a en charge de créer la composition de service, la réservation de ressources et même le maintien d'une structure cellulaire. Le mécanisme peut-il être plus distribué ? Et peut-il être plus indépendant des infrastructures ?

Nous avons préféré étudier des solutions non centralisées et considérer une autogestion au niveau de chaque composant.

- Les fonctions Media Delivery, Distribution and Storage : contient les services réseaux :
 - le Media Control Function (MCF) qui possède trois interfaces, une avec les services applicatifs (SCF), une avec les services réseaux (MDF) et une avec l'User Equipment (UE) pour la gestion de la télécommande.
 - Le Media Delivery Function (MDF) qui fournit le Media Delivery vers l'UE.

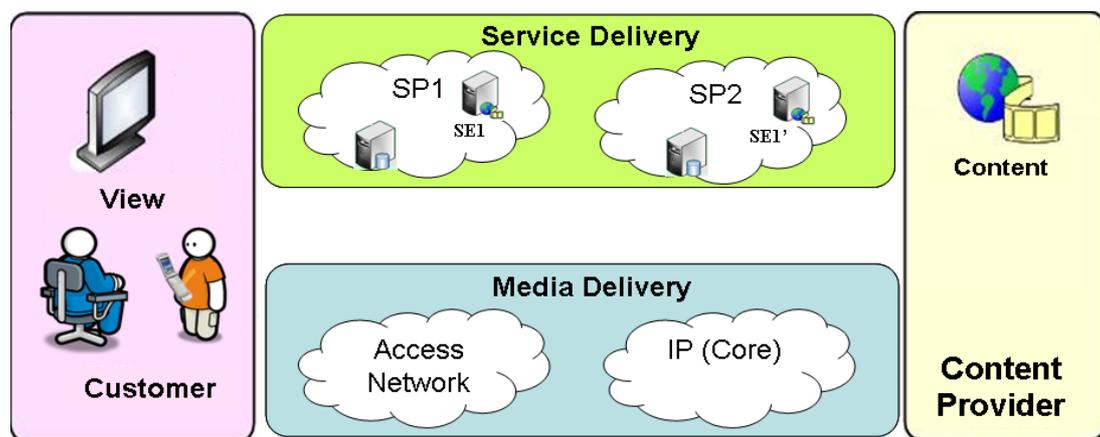
Dans cette architecture nous avons les services applicatifs (SSF, SDF, SCF et MCF) et les services réseaux (MCF et MDF). Les services applicatifs identifient les services générateurs de flux qui doivent être transmis à travers le réseau. De tels services peuvent changer pendant la transmission. Le MCF et le MDF sont concernés par le transport de média via tous les composants réseaux avec l'aide du NASS et le RACS, qui gèrent le Policy Enforcement Point (PEP) comme le policy control pour assurer la QoS.

Ainsi, nous avons un Service Delivery qui complète un Media Delivery pour ce service complexe de l'IPTV. Si cet IPTV était fourni sur des terminaux mobiles, le Service Delivery devra prendre en compte la gestion de cette mobilité pour la continuité de service.

III.3.4 Discussion

Dans les travaux de recherches que nous venons d'évoquer, nous avons trouvé quelques propositions pour le niveau de service, comme la composition de service et le routage pour la fédération de service. Mais la plupart des propositions relève du "Media Delivery" et permet d'assurer un « service réseau » à travers le transport du flux media d'une source vers une destination, tout en optimisant le chemin afin de maintenir la QoS demandée. Dans notre contexte mobile et ubiquitaire, cette source et cette destination peuvent changer pendant les déplacements de notre utilisateur. Le Media Delivery assure la continuité de la qualité du flux mais n'assure pas une continuité de service sans couture et une QoS de bout en bout.

Pour cela, nous proposons un Service Delivery (Figure III.3-2) au dessus du Media Delivery qui définit et gère la source et la destination, et sollicite le "media Delivery" pour transporter le contenu à partir de cette source choisie vers cette destination choisie en adéquation avec les préférences de l'utilisateur. Ceci nous offre une alternative pour adapter la QoS à la mobilité dans l'environnement ambiant.



SP : Service Platform

SE : Service Element

Figure III.3-2: Service Delivery proposé au dessus du Media Delivery

III.4 Les solutions existantes pour le Handover dans le réseau

La gestion de la mobilité du terminal est traitée par le processus du Handover, nous avons été motivée pour en étudier les tenants et aboutissants afin d'envisager un processus analogue pour la mobilité de service.

III.4.1 Introduction

Le handover est un processus fondamental dans les systèmes radio mobile car il permet indépendamment du temps de fournir une continuité de service à l'utilisateur nomade. La nécessité du handover peut s'expliquer soit pour l'opérateur, pour des raisons purement radio ou de trafic (équilibre de charge), soit pour l'utilisateur pour des raisons de sécurité, de services offerts non disponibles sur son système actuel ou de coût de connexion. Nous avons des techniques de handover horizontal au niveau du réseau d'accès (§III.4.2) et au niveau du réseau cœur (§III.4.3) et des techniques de handover vertical (§III.4.4). Dans notre analyse nous avons identifié que dans les différentes techniques de handover nous avons trois principaux rôles joués par un ou plusieurs acteurs. Nous avons le rôle d'initiateur, celui de décideur et celui d'exécuteur du handover. Nous allons décrire les différents acteurs dans les différents handover ainsi que le rôle joué par chacun d'eux.

III.4.2 Handover Horizontal de niveau 2 (L2)

(a) GSM (2G) et GPRS/EDGE (2,5G)

Le GSM (*Global System for Mobile communications*), le GPRS (*General Packet Radio Service*) et le EDGE (*Enhanced Data Rates for GSM Evolution*) sont trois types de réseaux cellulaires recommandés par le 3GPP. Ils gèrent la mobilité d'une manière équivalente, ils s'appuient sur un handover horizontal. Les acteurs impliqués dans ces types de handover sont le MS (*Mobile Station*), le BSC (*Basic Station Controller*) et le MSC (*Mobile Switching Center*) [GSM] [CO03] [GPRS] [EDGE]. Nous avons trois rôles dans le processus de handover :

- *L'initiateur* : ce rôle est joué par le MS. Il effectue des mesures sur le canal courant (le niveau de signal reçu noté RXLEV, la qualité du signal notée RXQUAL et le taux d'erreur bit (BER)) et sur les six cellules voisines ayant le niveau de puissance le plus fort (le niveau de signal reçu, la fréquence et le code couleur BSIC). Ces mesures sont ensuite transmises au BSC courant, ce qui lui permettra d'établir une liste de cellules candidates pour le handover.
- *Le Décideur* : ce rôle *peut être joué* soit par le BSC (cas du Handover intra-BSC) soit par le MSC (cas du Handover inter-BSC). Pour le premier cas, c'est le BSC qui choisit, parmi une liste de cellules candidates, la cellule cible la mieux adaptée pour des raisons, entre autre, de trafic. Pour le deuxième cas, c'est le MSC qui fait ce traitement. Il faut noter, qu'un Handover inter-MSC peut aussi apparaître.
- *L'exécuteur* : ce rôle est joué par le BSC. Il réserve un nouveau canal de trafic entre la cellule cible et le MS, puis demande au MS de basculer sur ce canal.

(b) UMTS (3G)

Afin de gérer la mobilité, l'UMTS (*Universal Mobile Telecommunications System*) se base sur un mécanisme de Handover horizontal [ETSI11] [ETSI07] qui se manifeste selon deux approches : le *Soft-Handover/Softer-Handover* et le *Hard-Handover*. Ces deux

techniques s'appuient sur trois acteurs essentiels : le UE (*User Equipment*), le Node-B et le RNC (*Radio Network Controller*).

Le *Soft handover* survient lorsque l'UE est dans une zone de couverture supportée par différents Node B. En effet, le *Soft-Handover* se produit lorsque l'UE est dans une zone de couverture supportée par différents Node-B. L'UE transmet et reçoit simultanément des données de ces différents Node-B. Les données utilisateur émises par l'UE sont envoyées à chaque Node-B qui les transmet au RNC où elles sont combinées et les données utilisateur émises par chaque Node-B simultanément sont combinées dans l'UE. Lorsque les Node-B sont contrôlés par différents RNCs, c'est le SRNC (*Serving RNC*) qui combine les données utilisateur.

L'UE mesure le RSCP (*Received Signal Code Power*) et le RSSI (*Received Signal Strength Indicator*) et renvoie ces informations vers les Node-B auxquels il est attaché (*Active Set*). A leurs tours, les Node-B renvoient les mesures vers le SRNC. Si ce dernier trouve que les mesures provenant d'un Node-B sont au dessous d'un certain niveau (noté As_{Th}) au moment où il y a un meilleur canal radio disponible, il prend la décision du *Soft-Handover*.

1. *L'initiateur* est joué par le SRNC.
2. *Le décideur* est joué par l'UE qui décide d'ajouter (*Radio Link Addition*), de relever (*Radio Link Removal*) ou de remplacer (*Combined Radio Link Addition and Removal*) un Node-B dans l'*Active Set*.
3. *L'exécuteur* est joué par le SRNC qui met à jour l'*Active Set* et envoie vers l'UE le message de signalisation *Active Cell Update Complete*.

Le *Hard-Handover* survient lorsqu'un lien radio d'un UE est enlevé avant l'établissement de nouveaux liens radio. Normalement, quand le UE passe dans une cellule où les fréquences diffèrent de celles qu'il vient de quitter, un *Hard-Handover* s'exécute. Nous parlons ainsi de *Handover inter fréquences*.

- [1] *L'initiateur* est joué par le SRNC, sur la base des mesures radio qui lui sont rapportées par l'UE.
- [2] Le décideur est joué par le SRNC qui choisit la cellule cible (Node-B cible).
- [3] *L'exécuteur* est joué par le SRNC qui exécute le handover en libérant le lien radio entre le UE et l'ancien Node-B, avant d'établir le lien entre cet UE et le Node-B cible.

(c) LTE (4G)

Pour gérer la mobilité, le LTE adopte une approche de *Handover horizontal* qui s'appuie sur trois acteurs essentiels qui sont l'UE, le eNode-B source et l'eNode-B cible [3GPP10] :

- *L'initiateur* est joué par l'UE en envoyant un rapport de mesure au eNode-B source. Ce dernier va déclencher le handover en se fondant sur le rapport reçu et sur les informations concernant la gestion des ressources radio.
- *Le décideur* est joué par l'eNode-B source qui envoie une requête de *Handover* (*HO Request*) vers l'eNode-B cible. Le eNode-B cible enregistre le contexte, prépare les couches basses pour le *Handover* et répond au eNode-B source par un acquittement (*HO Request Ack*).
- *L'exécuteur* est joué par le eNode-B source qui va transférer toutes les informations à l'UE, puis arrête d'envoyer et de recevoir des données avec lui et fait suivre les données au eNode-B cible. L'UE informe l'eNode-B cible du succès du *Handover*

avec un message de confirmation. Le eNode-B cible initie le changement de chemin de données en envoyant un « Handover Complete » aux passerelles pour que les données soient envoyées directement vers lui. Ces passerelles confirment le chemin par un message 'Handover Complete Ack'. A la réception de ce message le eNode-B cible envoie une indication 'Release Source' au eNode-B source pour qu'il libère définitivement la connexion avec l'UE.

(d) WIMAX (IEEE 802.16e)

Pour gérer la mobilité, le WIMAX (Worldwide Interoperability for Microwave Access) définit trois types de handover [IEEE09] : Le Hard Handover dans lequel le terminal se déconnecte de sa station de base avant de se connecter à une station de base cible. Il y a donc une coupure dans la communication. Par conséquent, ce mode de handover ne fonctionne que si la mobilité est lente. Le Fast Base Station Switching (FBSS) "optionnel" dans lequel les données du terminal sont reçues par toutes les stations de base du "jeu de diversité" (ensemble de stations de base actives pour le terminal), mais seule la station de base "ancree" va les interpréter. Cependant, les autres stations de base actives doivent être prêtes à envoyer des données au terminal. Notons que la station de base "ancree" peut changer au sein du "jeu de diversité". Et on assiste à une commutation rapide de station de base et ceci sans mettre en jeu la procédure de handover classique. Et enfin, nous avons la Macro Diversity Handover (MDHO) (soft handover) "optionnel" dans laquelle le terminal communique simultanément avec toutes les stations de base du "jeu de diversité". Au fil du temps et de ses déplacements, le terminal va modifier son "jeu de diversité". Dans le sens descendant, le terminal reçoit la même information, au même instant, en provenance de chacune des stations de base du "jeu de diversité". Le terminal va combiner les informations pour limiter les erreurs. Dans le sens montant, le trafic provenant du terminal est reçu par toutes les stations de base du jeu de diversité.

- L'initiateur est joué par le MS qui scanne son entourage dans des *Scanning Intervals* et fait des mesures sur les canaux de *Downlink* reçus. Il mesure le CINR (*Carrier to Interference-plus-Noise Ratio*), RSSI (*Receive Signal Strength Indicator*), le *Relative Delay* et le RTD (*Round Trip Delay*). Les mesures faites sont renvoyées ainsi au *Serving-BS* (ou le *Serving-Anchor-BS* dans le cas du FBSS), soit d'une manière périodique, soit suivant des événements déclenchés.
- Le décideur est joué par le *Serving-BS* (ou bien le *Serving-Anchor-BS*) qui sélectionne le meilleur *Target-BS* (ou bien *Target-Anchor-BS*) en se basant sur les paramètres reçus.
- L'exécuteur est joué par le *Serving-BS* (ou bien *Serving-Anchor-BS*) qui contacte le *Target-BS* pour qu'il exécute le Handover en ouvrant des canaux de communications avec le MS. Une fois la connexion est établie entre le *Target-BS* (ou bien *Target-Anchor-BS*) et le MS, la connexion pourra être relâchée entre ce MS et l'ancien *Serving-BS* (Ancien *Serving-Anchor-BS*).

III.4.3 Handover Horizontal de niveau 3 (L3)

(a) Mobile IP (MIPv4 et MIPv6)

Le MIP (*Mobile IP*) [JX07], a été proposé comme un handover au niveau du réseau cœur, il a été conçu afin de permettre à l'utilisateur de se déplacer à travers différents réseaux l'accès avec son application en cours tout en maintenant son adresse IP permanente. Pour chaque réseau visité, l'utilisateur obtient une adresse IP temporaire qu'on appelle CoA (*Care of Address*) qui est différente de son adresse permanente HoA (*Home Address*). Afin de gérer la mobilité, trois entités interagissent : le MN (*Mobile Node*), le HA (*Home Agent*) qui est un routeur sur le réseau mère et le FA (*Foreign Agent*) qui est un routeur sur le réseau visité par le MN. En effet, le FA se charge de l'attribution de la CoA. Cette adresse sera enregistrée au près du HA qui se charge d'établir une table de correspondance entre le CoA et le HoA. Notons que l'adresse temporaire peut être délivrée soit par le nouveau FA (on parle de FA-CoA) soit dynamiquement par DHCP (*Collocated CoA*). Au niveau de la communication, tous les paquets à destination du MN passent par le HA qui les encapsule en mettant la CoA comme adresse destinataire. Ensuite, le FA les envoie au MN. Cependant, les paquets envoyés par le MN passent par le FA sans passer par le HA.

1. Le MN joue le rôle d'initiateur, il se rattache à un nouveau FA qui se trouve dans sa nouvelle localisation. En effet, un MN pourra se connecter à plusieurs FA simultanément.
2. Le FA attribue un CoA au MN.
3. Le MN enregistre la nouvelle CoA ainsi que l'adresse du nouveau FA au sein de son HA. Par suite le HA transmet toutes les requêtes à direction du MN (à direction de l'ancien FA) vers le nouveau FA.

III.4.4 Handover Vertical

Suivant la diversité des technologies et leur inévitable cohabitation on fait appel au handover vertical.

(a) UMA/GAN

UMA/GAN (*Unlicensed Mobile Access/Generic Access Network*) assure la continuité des communications entre un réseau GSM/GPRS et un réseau local sans fil tel le WiFi ou le Bluetooth et inversement, offrant ainsi les mêmes applications métiers aux utilisateurs en dehors du réseau de l'entreprise. Il s'agit ici d'utiliser un terminal bi-mode pouvant fonctionner avec une licence dans un réseau GSM et sans licence dans un réseau Wifi et Bluetooth étant dans une zone de couverture GAN [GAN].

Avec UMA/GAN les utilisateurs peuvent utiliser toutes les applications offertes par les réseaux GSM/GPRS sans soucis de déconnexion.

Exemple de handover : Handover du GSM vers WiFi [AA07] :

Article I. *Initiateur* : Le MS envoie un rapport de mesure au BSC contenant les caractéristiques des cellules voisines ayant le niveau de signal le plus puissant y compris celui de la cellule UMA.

Article II. *Décideur* : Après analyse du rapport, le BSC envoie au CN un message « Handover Required » avec comme cellule cible la cellule UMA. Le CN va demander au UmaNC d'allouer un nouveau canal radio au mobile, ce qu'il va faire et en informe le CN. Le CN enverra ces informations au BSC. Puis le BSC va signifier à la MS qu'elle peut commencer le « handover » vers l'UMA.

Article III. *Exécuteur* : Le MS (qui décide du moment du handover) contacte l'UmaNC et lui fournit les données devant être utilisées pour le HO reçu du BSC. Après vérification, l'UmaNC met en place le tunnel pour allouer les ressources à la MS. Ensuite, l'UmaNC indique au CN qu'il a bien détecté la MS. Dès que l'UmaNC signale au CN que le handover est complètement réalisé, il va libérer les ressources allouées au BSC, qui va à son tour, confirmer la libération des ressources allouées.

(b) **MIH (IEEE 802.21)**

Le MIH (Media Independent Handover) [MIH] élabore une architecture qui permet le passage sans coupure d'un réseau à un autre (que ces réseaux soient homogène ou hétérogènes) lors d'une connexion active et ce indépendamment du media. Pour cela on assiste à une création d'une nouvelle fonction appelée MIHF (coté mobile et réseau et qui permet la coopération entre le mobile et le réseau) dans la pile protocolaire de gestion de mobilité MME (*Mobility Management Entity*). La fonction MIHF est constituée de trois services :

Media Independent Event Service (MIES): génère des événements (qui se transmettent d'une couche basse à une couche haute) qui peuvent indiquer ou prévoir un changement dans l'état ou le comportement de transmission des couches physique et liaison.

Media Independent Command Service (MICS): il définit les décisions prises par les couches supérieures afin de contrôler les couches basses. Il facilite l'initiation et la préparation du Handover .En effet ces décisions permettent entre autre: de déterminer l'état des liens, de contrôler les performances du terminal multi-mode et de renseigner dynamiquement sur le débit du lien, la puissance du signal etc.

Media Independent Information Service (MIIS) : Le service d'information aide le nœud mobile à la préparation du handover (avec la possibilité de découvrir les réseaux environnants, et de sélectionner le réseau adéquat).

On distingue 3 types de handover [] :

- Initialisé et décidé par le terminal (utilise une partie des services MIH)
- Initialisé par le terminal et décidé par le réseau (utilisation des services MIH)
- Initialisé et décidé par le réseau (le réseau utilise des événements MIH)

Le MIH ne gère pas l'exécution des handover, il ne fait qu'apporter son aide aux couches supérieures afin qu'elles puissent initier et préparer ces derniers.

III.4.5 **Discussion**

Dans cette analyse des techniques existantes dans la gestion de la mobilité, nous avons vu que nous avons des techniques de handover au niveau du réseau d'accès qui permettent de gérer la mobilité du terminal par le basculement du réseau d'accès vers un autre réseau d'accès, des techniques de handover au niveau du réseau cœur qui permettent d'avoir une adresse IP permanente durant le déplacement entre plusieurs réseaux d'accès et enfin des techniques qui permettent de basculer d'une technologie réseau vers une autre technologie réseau. Toutes les techniques analysées se basent sur trois rôles principaux qui sont l'initiateur, le décideur et l'exécuteur. Chacun de ces trois rôles peut être joué par un acteur ou différents acteurs.

Avec le handover réseau nous pouvons ainsi profiter des réseaux ambiants à l'utilisateur mobile pour maintenir sa QoS et assurer son service, la question que nous nous sommes posé est : est ce que ce concept de handover peut être appliqué au niveau service ? Et peut on profiter de services équivalents qui se trouvent dans la zone ambiante à l'utilisateur pour

maintenir sa QoS et assurer la continuité du service? Pour répondre à ces questions, nous avons proposé le handover sémantique qui traite la continuité de service durant la mobilité par zones ambiantes.

III.5 Gestion de services dans les Standards

III.5.1 Introduction

Afin de définir les processus de gestion du cycle de vie des services dans le nouveau contexte NGN/NGS nous nous sommes intéressés à ce qui existe dans les normes afin d'étudier les solutions actuelles et d'analyser ce qui manque pour la prise en compte de nos besoins de gestion dynamique. Ceci nous a conduit au Business Process Framework : eTOM (enhanced Telecommunication Operation Map) qui est le cadre standard de description des processus, utilisé par les acteurs de l'industrie des télécoms, pour normaliser et harmoniser leurs activités. Avant de parler du Business Process Framework nous allons retracer l'état et l'évolution de l'existant en termes de gestion dans les télécommunications en partant du TMN vers le NGOSS.

III.5.2 Évolution des architectures de gestion des télécommunications

Dans cette partie, nous présentons l'évolution de l'architecture de gestion des télécommunications, nous commençons par présenter le TMN :

(a) Telecommunication Management Network (TMN M3010)

Le TMN a été développé pour gérer les équipements, les réseaux et les services de télécommunications offerts par les opérateurs. Il favorise l'interconnexion des systèmes de gestion et les équipements de télécommunications à gérer.

La recommandation M.3010 [TMN] définit les concepts généraux de gestion du TMN et présente plusieurs architectures de gestion à différents niveaux d'abstraction :

- Une architecture fonctionnelle : qui définit cinq blocs fonctionnels ainsi que cinq points de référence.
- Une architecture physique : qui définit comment ces fonctions de gestion peuvent être implémenté dans l'équipement physique ainsi que les interfaces les reliant.
- Une architecture informationnelle : qui décrit l'ensemble des composants selon une structuration préconisée par la gestion OSI.
- Une architecture en couches logique (LLA) : qui montre comment la gestion peut être structurée selon différentes responsabilités.

Les différentes couches logiques (Figure III.5-1) sont :

- Couche gestion commerciale (BML): qui comprend la gestion de la planification, des budgets et des objectifs de l'entreprise
- Couche gestion de services (SML): qui s'occupe de tout ce qui touche à la gestion des services utilisateurs.
- Couche gestion de réseaux (NML): qui s'occupe des équipements en termes de configuration des réseaux.

- Couche gestion des éléments (EML): qui s'occupe de toute la gestion des éléments ou groupes d'éléments comme leur charge, leur état, mais aussi de la gestion logique en terme de configuration.
- Couche éléments de réseaux (NEL) : qui contient les éléments de réseaux eux mêmes.

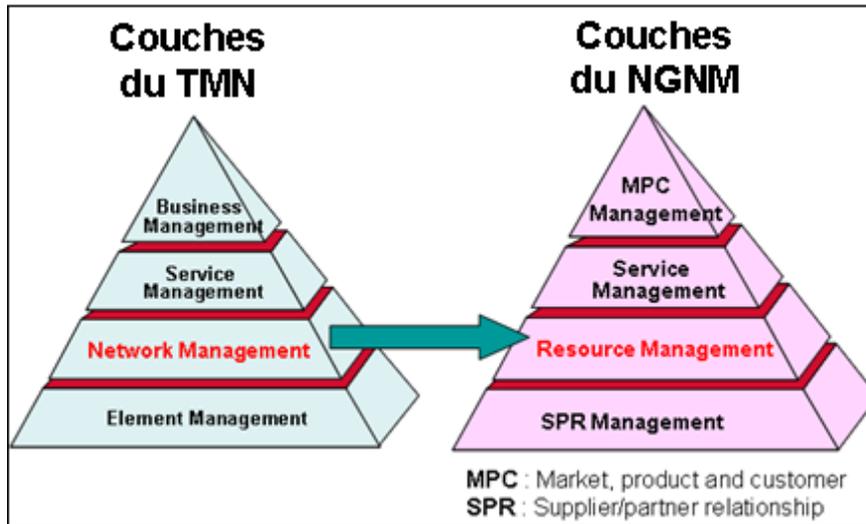


Figure III.5-1: Le passage du network management au resource management.

Ces différentes couches contiennent des blocs fonctionnels qui sont reliés entre eux grâce aux points de référence, appelés aussi interfaces lorsque les blocs fonctionnels connectés sont situés sur des systèmes distincts.

Les points de référence permettent de définir les informations qui doivent être échangées entre les blocs fonctionnels ainsi que les protocoles à utiliser.

Ces blocs fonctionnels sont :

- Operations System Function (OSF): qui représente le manager et offre l'ensemble des activités de gestion comme le monitoring, la coordination et le contrôle des entités du TMN. Parmi les activités nous retrouvons celles définies dans le cadre de la gestion OSI comme la gestion des fautes, de la configuration, de la comptabilité, des performances et de la sécurité.
- Mediation Function (MF): qui s'occupe du transfert et du prétraitement des informations entre le système d'exploitation et les éléments réseau qui ne sont pas TMN natif.
- Workstation Function (WSF): qui est chargée de présenter les informations provenant des différents composants actifs du système de gestion TMN.
- Data Communications Networks : qui sont en charge du transport des informations entre les entités TMN.
- Network Elements Function (NEF) : qui représente l'agent et qui est implémentée dans les équipements de réseaux eux-mêmes.

La Recommandation M3010 a défini le réseau de gestion des télécommunications qui sert à gérer des réseaux de télécommunication traditionnels.

Ce réseau offre une architecture transparente aux technologies ainsi il peut gérer les réseaux de nouvelles générations (NGN) tout en introduisant certaines modifications qui sont principalement une meilleure prise en charge des services et des processus d'ingénierie commerciale. Ces modifications apportées ont été définies dans la recommandation M3060 [NGNM].

(b) Gestion des Réseaux NGN (NGNM M3060)

La gestion des réseaux NGN prend en charge la planification, le déploiement, l'approvisionnement, l'exécution et la maintenance des services et des ressources NGN.

Nous avons noté que le passage du TMN vers le NGNM prend en charge la gestion de toutes les ressources (Figure III.5-1). Les ressources à gérer sont divisées en ressources de transport et ressources de service.

En plus des points de vue fonctionnel, informationnel, physique et de l'architecture en couches logiques (LLA), nous avons un nouveau point de vue relatif aux processus d'ingénierie commerciale, fondé sur le plan eTOM.

Dans l'architecture fonctionnelle, nous avons de nouveaux blocs fonctionnels (Figure III.5-2) : bloc de gestion de ressources de service (SRMF), bloc de gestion de ressources de transport (TRMF), bloc d'élément de service (SEF) et bloc d'élément de transport (TEF).

De nouveaux blocs fonctionnels (Figure III.5-2) sont introduits concernant la gestion des entreprises (EpMF), des fournisseurs (SPRMF), des marchés, produits et clients (MPCMF).

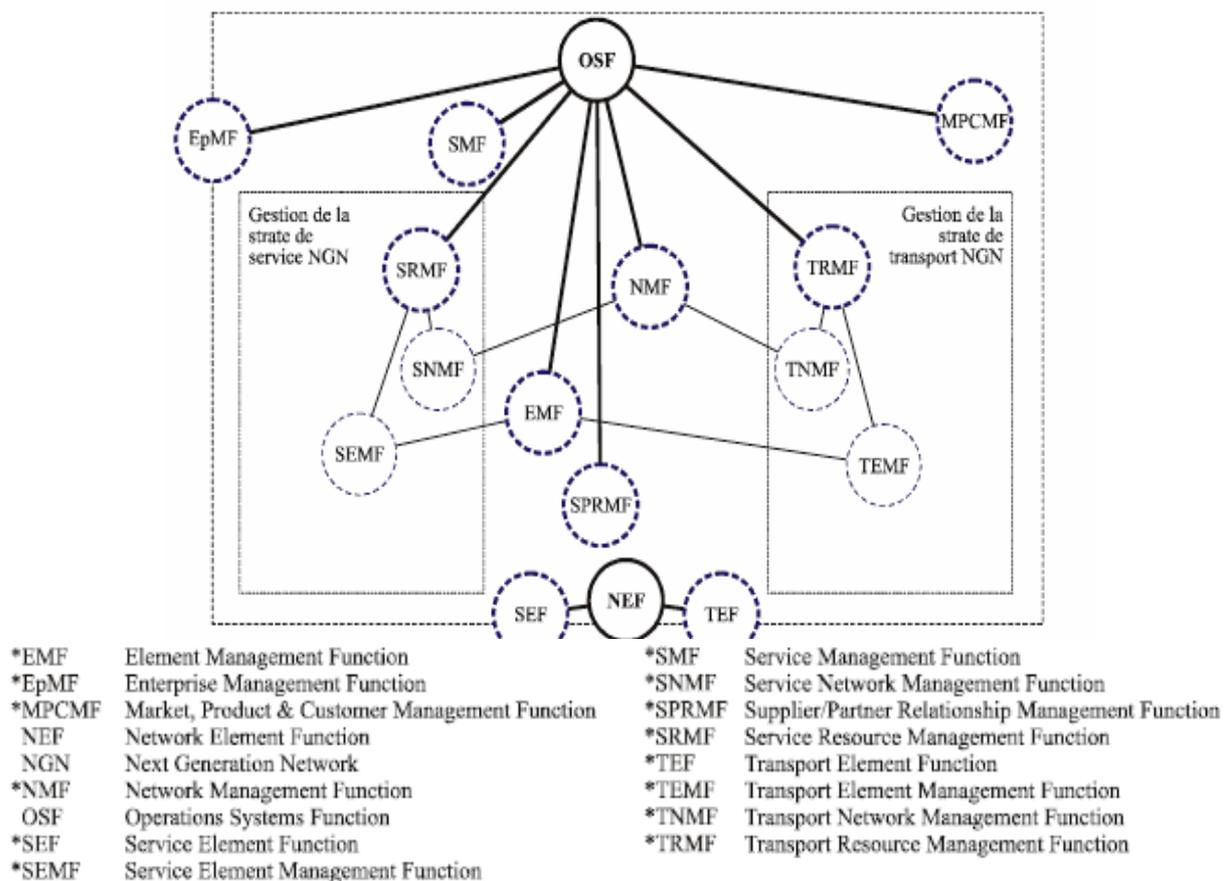


Figure III.5-2 : Les blocs fonctionnels du NGNM.

(c) Le NGOSS (New Generation Operations Systems and Software)

La Solution Frameworks NGOSS est un programme intégré innovant, pour le développement et le déploiement de systèmes de gestion. Il a été introduit par le Tele Management Forum (TMF) qui est un consortium privé à but non lucratif qui fournit une stratégie ainsi que des solutions pratiques pour améliorer la gestion et le fonctionnement des services de communication et d'information.

Le TMF liste plusieurs membres comprenant des fournisseurs de services, des équipementiers, des sociétés de services ou des intégrateurs, et ce du monde entier.

Le NGOSS (Figure III.5-3) [TMF] propose une approche intégrant les axes suivants:

- Une analyse des processus et flux opératoires de l'opérateur, cette étape utilise le Business Process Framework : *eTOM (Enhanced Telecom Operations Map)* qui sera détaillé plus loin.
- L'analyse et la conception du système, qui est basée sur le *SIM (System Integration Map)*.
- L'analyse et la conception du référentiel informationnel (données utilisées dans les architectures NGOSS), basée sur le Information Framework (IF) anciennement dénommée *SID (Shared Information Data)*.
- TMF Application Framework (TAM), fournit un modèle pour regrouper les processus et leurs informations associées en des applications reconnaissables. Il fournit également un langage commun et un système d'identification entre l'acheteur et le fournisseur pour tous les domaines d'application.
- Integration Framework, définit comment les processus et les informations peuvent être automatisés en définissant des interfaces basées sur un SOA standardisé appelées les Business Services (NGOSS Contracts).

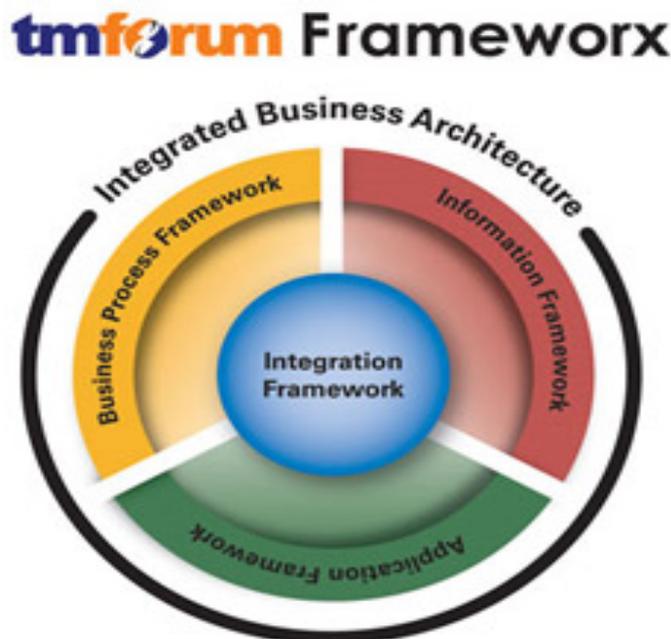


Figure III.5-3 : NGOSS

III.5.3 Business Process Framework (eTOM)

Défini par le TMF pour fournir un modèle business et une architecture destinés aux fournisseurs de services et acteurs du secteur des télécommunications. Il décrit l'ensemble des processus d'entreprise utilisés par un fournisseur de services et l'analyse selon différents niveaux de détail selon leur importance et leur priorité. L'eTOM [GB921b] se décompose en 4 niveaux hiérarchiques (de zéro à trois) :

Au niveau zéro, il comprend trois principaux domaines de processus :

Stratégie, infrastructure et produits (SIP) : qui comprend des processus d'élaboration de stratégies et d'engagements à respecter au sein de l'entreprise; des processus de planification et de gestion de la fourniture et de l'amélioration des infrastructures et des services; et des processus de développement, déploiement et gestion de la chaîne d'approvisionnement des services, c'est dans cette partie que nous avons identifié les processus impliqués dans le déploiement de services.

Opérations (OPS) : qui comprend tous les processus opérationnels d'exploitation des services et de gestion des clients et du réseau. Ce domaine inclut également la gestion des ventes et la gestion des relations avec les fournisseurs/partenaires.

Gestion de l'entreprise : qui comprend les processus permettant la définition et la réalisation d'objectifs stratégiques de l'entreprise ainsi que la fourniture de services de prise en charge requis dans toute l'entreprise (processus de gestion financière, de gestion des ressources humaines, etc.).

Dans les deux domaines SIP et OPS, l'eTOM représente quatre niveaux de visibilité (sous forme de processus fonctionnels horizontaux) :

Marchés, Produits et Clients : qui comprend des processus de gestion des ventes, de commercialisation, de produits et de l'offre ; des processus opérationnels tels que la gestion de l'interface client, le traitement des commandes, le traitement des problèmes, la gestion des accords SLA et la facturation ;

Services : qui sont les processus de gestion de capacité de développement, de fourniture de services (traduction entre les services catalogues et les services réels), de configuration des services, de gestion des problèmes de service et d'analyse de la qualité de service ;

Ressources : qui sont les processus de gestion du développement et fourniture d'une infrastructure de ressources (réseaux, applications (services) et équipements), de gestion opérationnelle de ces ressources (approvisionnement, gestion des anomalies, gestion de la performance, etc.).

Fournisseurs / partenaires : processus de gestion des interactions de l'entreprise avec ses fournisseurs et ses partenaires. Il s'agit des processus de développement et de gestion de la chaîne d'approvisionnement sur laquelle l'infrastructure et les produits sont fondés, ainsi que des processus assurant l'interface opérationnelle avec les fournisseurs et les partenaires.

Verticalement l'eTOM représente sept processus de bout en bout (Figure III.5-4) qui sont :

Stratégie et engagement (strategy and commit): qui permet d'élaborer des stratégies assurant la prise en charge des processus de cycle de vie de l'infrastructure et de cycle de vie des produits et services. Il doit également permettre d'élaborer des engagements au sein de l'entreprise en soutien à ces stratégies. Pour cela, il fait intervenir tous les niveaux, depuis les marchés, les clients et les produits ainsi que les services et les ressources dont ils dépendent et jusqu'à la participation des fournisseurs et des partenaires.

Gestion du cycle de vie de l'infrastructure : qui concerne le développement et le déploiement d'une nouvelle infrastructure, l'évaluation de la performance de l'infrastructure et la prise de mesures destinées à satisfaire aux engagements pris en matière de performance.

Gestion du cycle de vie des produits : qui porte sur l'introduction de nouveaux produits (sous la forme de services fournis aux clients), de leur déploiement, de l'évaluation de leur qualité et sur la prise de mesures relatives à cette qualité.

Exécution (Fullfilment): qui permet de fournir aux clients les services demandés en temps voulu et de façon appropriée. Ce groupement de processus de bout en bout informe le client de l'avancement de sa commande, garantit la réalisation de la commande en temps voulu et assure la satisfaction pleine et entière du client.

Assurance: qui permet d'exécuter des activités de maintenance pour faire en sorte que les services fournis aux clients soient continuellement disponibles et conformes aux accords SLA ou aux niveaux de qualité de service requis. Ce groupement effectue une surveillance continue de l'état des ressources et de la performance pour détecter d'éventuelles défaillances. Il recueille des données de performance et les analyse afin d'identifier d'éventuels problèmes et de les résoudre sans incidence pour le client. Il gère les accords SLA et rend compte au client de la performance du service. Il reçoit des comptes rendus d'anomalies provenant du client, informe ce dernier de l'état de ces anomalies et les corrige pour une satisfaction pleine et entière du client.

Facturation (Billing) : qui permet de générer les factures, de fournir aux clients des informations d'utilisation avant facture puis les factures, de recueillir les paiements et de procéder à leur traitement. Il gère en outre les demandes des clients relatives aux factures, indique l'état de ces demandes et assure la résolution des problèmes de facturation en temps voulu et à la satisfaction du client.

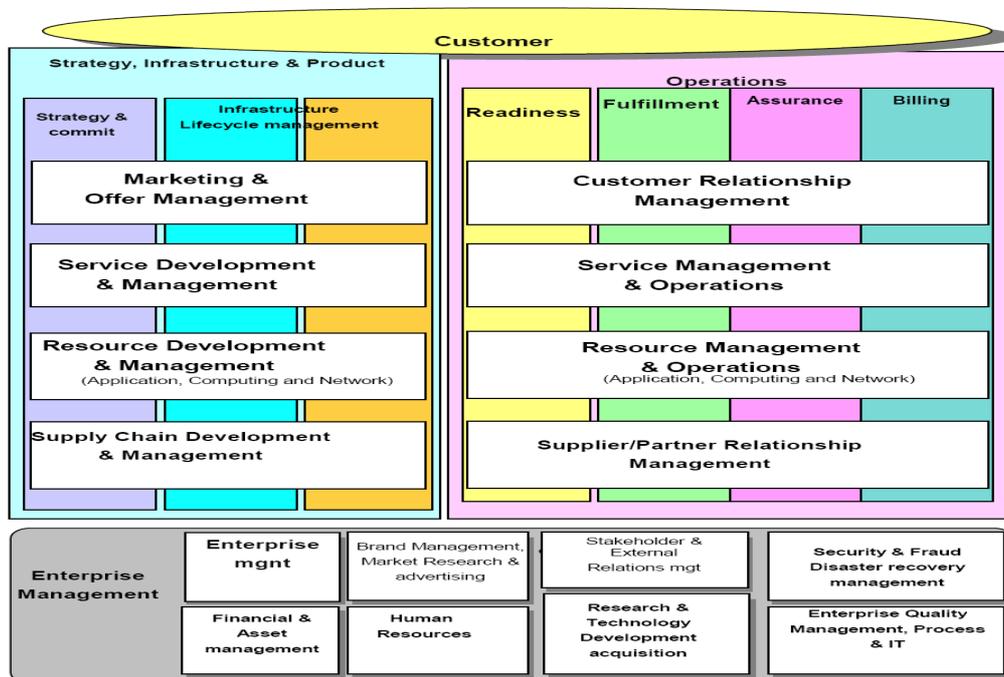
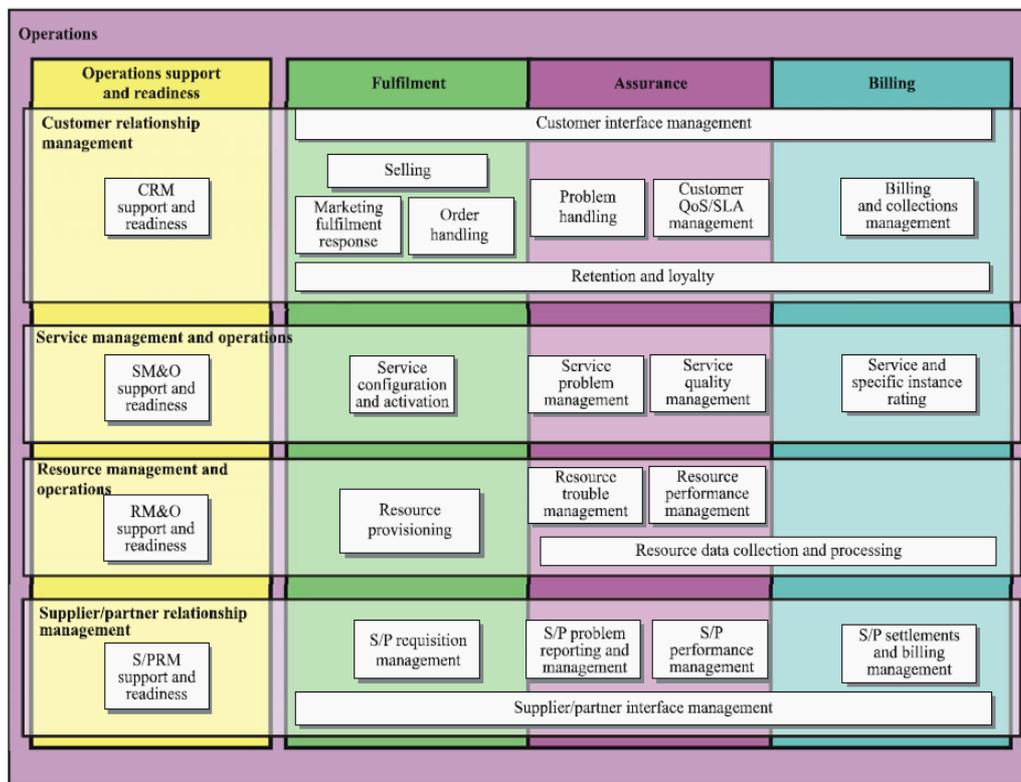


Figure III.5-4: eTOM de Niveau 1

Le niveau 2 de la partie Operations de l'eTOM est représenté sur la Figure III.5-5.



M.3050Suppl.4(07)_F6-4

Figure III.5-5 : Décomposition de niveau 2 de « Opérations ».

(a) Provisioning de service dans eTOM

Le processus de provisioning dans eTOM se situe dans la partie Operations de la cartographie (processus Fulfilment), c'est un processus vertical qui passe par les quatre couches horizontales d'eTOM. Le processus est divisé en trois phases : précommande, commande et post commande, nous détaillons chaque étape en fournissant le process flow de bout en bout.

Flux de processus de Précommande : est l'étape où un client contacte un fournisseur de service pour demander un ou plusieurs services, dans cette phase nous avons une Réserve de ressources pour la commande faite par le client, car avant d'avoir le service, le client passe par quatre étapes principales concernant son service (Figure III.5-6) :

L'étape de réservation des ressources où le client contacte un fournisseur de service et donne ses besoins et le fournisseur contrôle la faisabilité de la demande et lui propose une proposition de vente.

L'étape d'allocation des ressources où le fournisseur de service alloue les ressources déjà réservées par le client.

L'étape de provisioning des ressources allouées après acceptation de ce que lui propose le fournisseur.

L'étape d'activation des ressources où le fournisseur informe le client que le service est prêt à être utilisé.

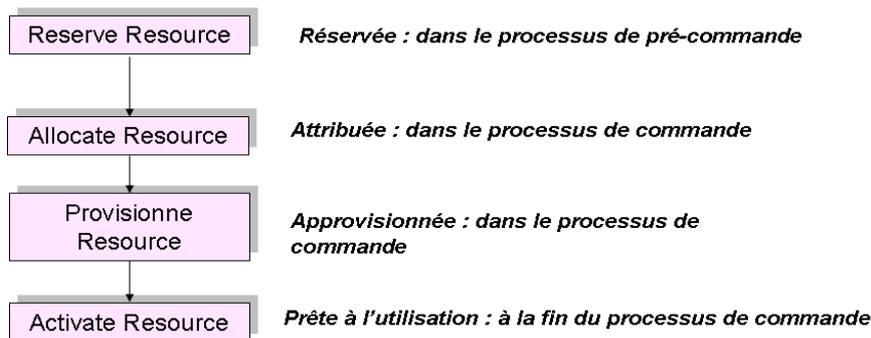


Figure III.5-6 : Étapes de provisioning de ressources.

Sur la Figure III.5-7, nous avons un exemple de Flux de processus de réservation des ressources [GB921c]:

Etape 1 : Un client contacte le fournisseur sur des services.

Etape 2 : Demande des renseignements sur la vente envoyée vers le processus Selling tout en prenant en compte le profil du client fourni par le processus de fidélisation (Retention and loyalty).

Etape 3 : Le processus de vente envoie une demande de clarification sur les besoins du client.

Le client fournit ses besoins.

Etape 4 : Si le fournisseur ne peut répondre exactement à ces besoins, une solution alternative peut être proposée au client.

Etape 5 : Le processus de Selling va envoyer une demande de Faisabilité au processus de traitement des commandes (Order Handling).

Etape 6 : Le processus Order Handling envoie une demande de choix de Technology et de Design au processus Service Configuration & Activation qui en s'appuyant sur la base de données des services (Service Inventory) va proposer une configuration du service selon la demande d'utilisateur.

Etape 7 : Le processus Service Configuration & Activation va aussi demander à la couche ressource et plus précisément au processus Resource Provisioning de réserver les ressources nécessaires à ce service.

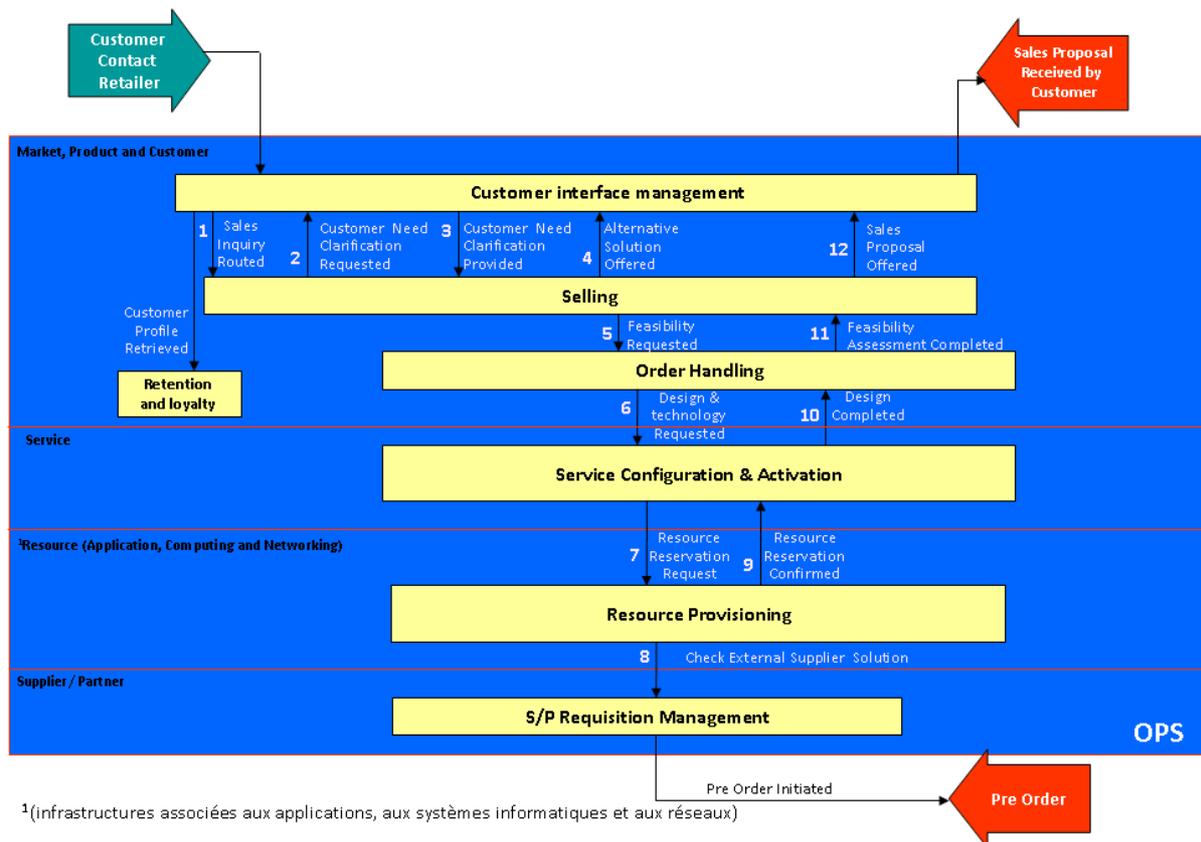
Etape 8 : Dans cet exemple nous avons des ressources externes donc le processus Resource Provisioning va vérifier la solution du fournisseur externe en interagissant avec le processus S/P Requisition Management et lancer la précommande.

Etape 9 : Le processus Resource Provisioning va confirmer la réservation des ressources externes et internes vers la couche service.

Etape 10 : La couche service va envoyer la réponse du choix de conception et de technologie vers le processus de traitement des commandes.

Etape 11 : Le processus Order Handling va notifier le processus Selling de la fin d'évaluation de faisabilité.

Etape 12 : Le processus Selling va offrir au client une proposition de vente.



¹(infrastructures associées aux applications, aux systèmes informatiques et aux réseaux)

Figure III.5-7: Flux de processus de Précommande dans eTOM.

Flux de processus de Commande : nous avons l'allocation, le provisioning et l'activation des ressources déjà réservées par le client dans la phase précommande, une fois ces trois étapes établies le service est prêt à être utilisé.

La Figure III.5-8, montre un exemple de Flux de processus d'allocation et d'activation des ressources [GB921c] :

Etape 1 : Le processus commence par le lancement d'une commande par le client.

Etape 2 : La commande est reçue par le Selling.

Etape 3 : Le Selling envoie une demande de priorité du client au processus Retention & Loyalty.

Etape 4 : Le processus Retention & Loyalty notifie la priorité du client.

Etape 5 : Le Selling va envoyer une confirmation de la commande au client et lance cette commande en l'envoyant vers le Order Handling.

Etape 6 : Le Order Handling va envoyer une demande de Design vers le processus Service Configuration & Activation.

Etape 7 : Le processus de configuration et d'activation des services va envoyer les requêtes pour l'Allocation des ressources internes et externes, pour les ressources externes la requête est envoyée vers la couche Supplier / Partner.

Etape 8 : Le processus Resource Provisioning va interagir avec la base de données des ressources pour allouer les ressources déjà réservées dans la phase de précommande.

Une fois les ressources allouées, le statut de ces ressources sera changé dans le Resource Inventory du statut Reserved à Allocated.

Etape 9 : Le processus Resource Provisioning va notifier à la couche service que les ressources sont allouées.

Etape 10 : La couche service va notifier au processus Order Handling la fin de la partie conception.

Etape 11 : Le processus Order Handling va envoyer une demande de confirmation de la conception.

Etape 12 : Le client répond par l'acceptation de la conception.

Etape 13 : Le processus Order Handling lance la commande de service vers la couche service.

Etape 14 : Le processus Service Configuration & Activation lance le processus de Provisioning des ressources internes.

Etape 15 : Le processus Resource Provisioning va notifier la fin du Provisioning vers la couche service.

Etape 16 : Le processus Resource Provisioning va Activer les ressources internes, les ressources externes seront activées par le processus S/P Requisition Management.

Etape 17 : Le processus Service Configuration & Activation va notifier à la couche au dessus que le service est activé et va envoyer les informations sur le service vers les processus d'Assurance et de Billing.

Etape 18 : Le processus Order Handling va notifier le client de la fin de traitement de sa commande.

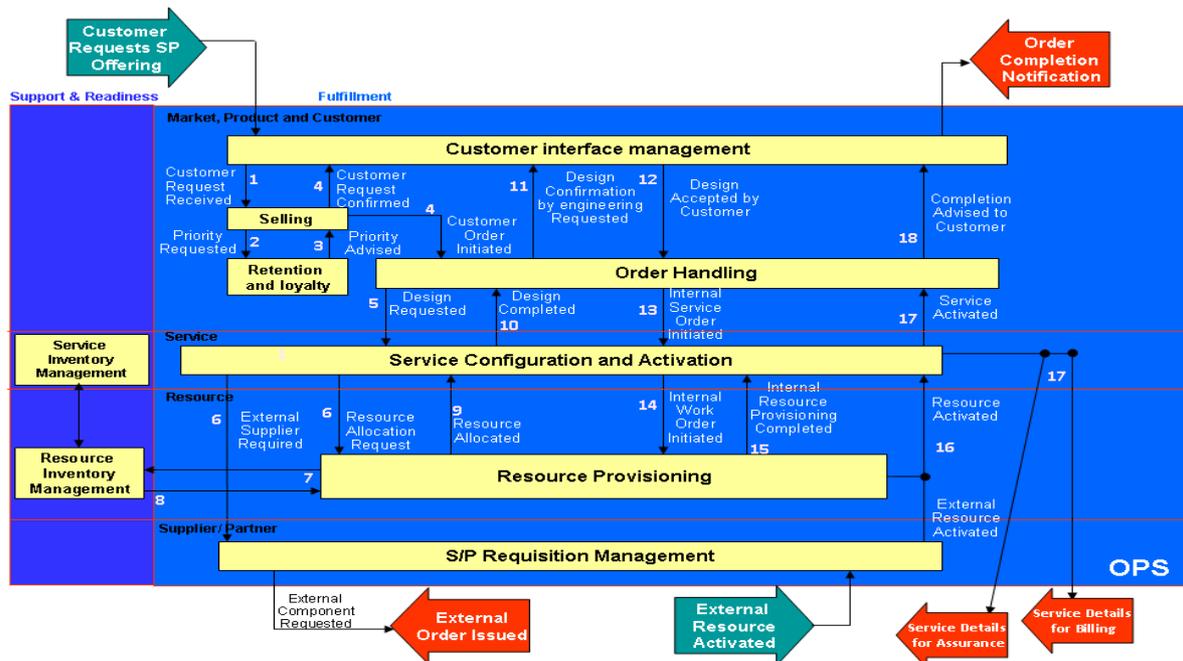


Figure III.5-8 : Flux de processus de Commande dans eTOM.

Flux de processus de Post Commande [GB921c] : Dans cette phase le processus Order Handling va envoyer une requête vers le client pour savoir s'il est satisfait, et dans ce cas là, il va envoyer les informations vers le processus d'Assurance et de Billing (Figure III.5-9).

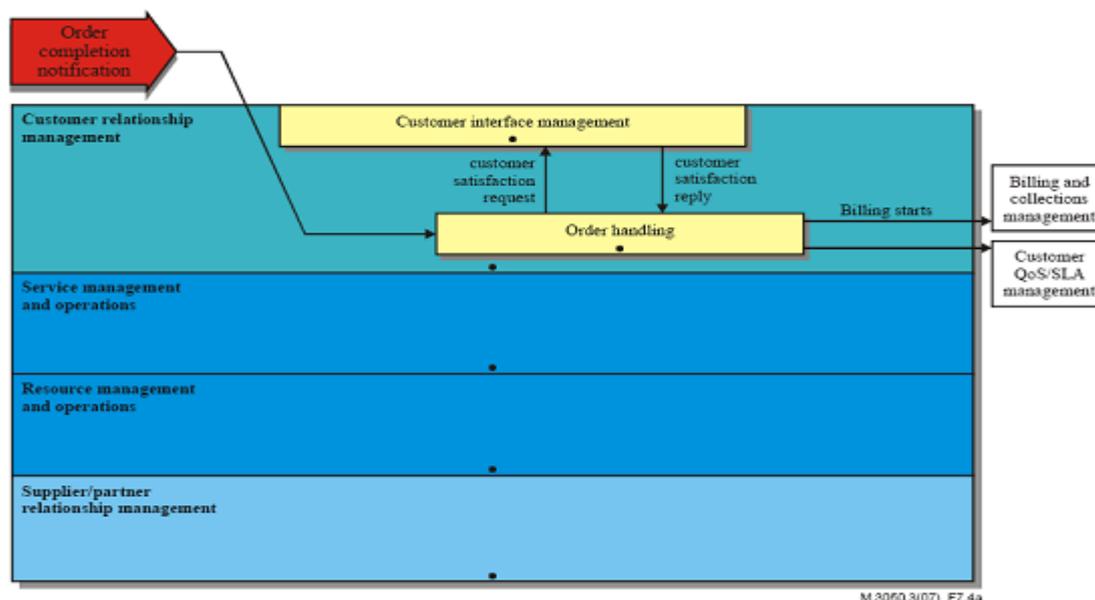


Figure III.5-9 : Flux de processus de Post Commande dans eTOM.

Nous constatons dans ce dernier processus de Post Commande l'assurance et le billing sont initialisés une fois pour toute, c'est supposé que le service est de nature statique. Or le contexte NGN suppose que nous avons des utilisateurs qui sont mobiles, qui ont des préférences en fonction de leur localisation, qui vont changer leur composition de services élémentaires, etc. C'est dire que les services sont loin d'être statiques et sont donc plutôt dynamique, c'est-à-dire qu'en cours d'exploitation des changements doivent être pris en compte et que nous aurons besoin de réapprovisionner nos ressources dynamiquement.

III.5.4 Discussion

Dans eTOM, la couche service s'occupe du *passage de paramètre de services* conformément à la demande envoyée par la couche au dessus et selon ce qui existe dans le Service Inventory. *C'est-à-dire que la couche service traduit les paramètres du service au catalogue (Offre) en paramètres (voir en services élémentaires) des ressources réelles (équipement, réseau et service).*

Cette couche ne fait pas le provisioning de la ressource service. Cette couche n'est donc pas la couche service représentant les ressources "Service". C'est la couche ressource qui s'occupe du provisioning des ressources, que ce soient celles des services (applications), celles des réseaux, ou celles des équipements. Elle ne suit pas la même identification que celle donnée par notre modèle (abstraction du monde réel). Dans notre modélisation nous proposons des « services applicatifs » à la place des applications et « des services de base » à la place des ressources Computing afin d'avoir une structuration, voire une architecture de services. Dans cette structuration les services applicatifs sont les services exposables présents dans le catalogue et les services de base sont ceux qui supportent la réalisation des services exposables. Une application client s'obtient alors par une composition de services.

III.6 Conclusion de l'état de l'art : limites des solutions

En conclusion aux travaux existants et leurs limitations pour une gestion dynamique du cycle de vie des services dans le nouveau contexte NGN/NGS, le Tableau III.6-1 résume les défis à relever par rapport à l'existant.

Tableau III.6-1 : Limites des solutions existantes.

	Les nouveaux besoins	Etat de l'art	Défis
Déploiement	Des applications centralisées aux services ubiquitaires	Services ubiquitaires, composition de services basés infrastructure fournisseur	Déploiement dynamique pour réagir rapidement à un dysfonctionnement de QoS durant l'usage
Provisioning	SLA = QoS E2E donc y compris le niveau service	Configuration du service, Provisioning de QoS au niveau réseau	Comment provisionner le niveau service
Usage	Transparence E2E pour l'utilisateur durant les changements (mobilité et préférences)	Assuré seulement par la couche transport (Media Delivery) en prenant en compte le type de service	Continuité de service durant l'usage Peut-on standardiser cet aspect?
Handover	Ressources ambiantes	Handover horizontal et vertical	Handover au niveau services

Chapitre IV Proposition : Gestion du cycle de vie dans un contexte NGN/NGS

Après avoir analysé le contexte et rapporté les principaux résultats de l'existant, nous allons présenter nos propositions pour une gestion dynamique du cycle de vie des services dans un contexte NGN/NGS. Elles se subdivisent en trois parties. Nous présentons comment repenser les phases de Déploiement (§IV.1), de Provisioning (§IV.2) et de Delivery (§IV.3) afin de répondre aux nouveaux besoins NGN/NGS.

IV.1 Déploiement intelligent d'éléments de services pour une QoS efficace de bout en bout

IV.1.1 Motivations

Le déploiement d'un service passe par différentes étapes allant de l'installation du service (ou de sa mise à jour), à sa configuration, à son test et jusqu'à son activation sur la plate-forme.

Pour déployer des services, les fournisseurs aujourd'hui installent les services sur leurs plates-formes puis effectuent les tests afin de s'assurer de la bonne exécution du service sur la plate-forme, une fois les tests réussis (le service s'exécute et effectue le traitement demandé), ils configurent ces services et enfin les activent (prêt à être utilisé par les clients).

Pour s'assurer que le service à installer va satisfaire son contrat de QoS et ainsi le SLA demandé par le client, n'est il pas mieux de garantir ce contrat dès son installation ? C'est à dire qu'il soit installé sur la plate-forme qui répond à ses besoins et que le transfert des données se fasse sur les réseaux qui assurent cette QoS. Ce challenge d'installer les services là où ils auront la bonne QoS a motivé notre proposition d'avoir un mapping entre la QoS demandée par le service et la QoS offerte par les ressources physiques durant le processus de déploiement. En effet, la plate-forme rend "un service d'environnement d'exécution" avec une "QoS offerte" que le processus de déploiement sélectionne en fonction de la QoS demandée. Ceci va permettre au service déployé d'assurer son contrat durant l'utilisation.

Dans notre contexte NGN/NGS, mobile et ubiquitaire, les utilisateurs utilisent les services sous forme de compositions d'éléments de service ubiquitaires quelque que soit leur localisation, ainsi avant de présenter le processus de déploiement, nous allons d'abord détailler ce que nous allons déployer (l'élément de service) et son modèle de service, son modèle de QoS et le modèle informationnel utilisé pour le déployer.

IV.1.2 Quoi déployer ? L'élément de service (SE)

Notre contexte NGS est caractérisé par une architecture de service où à priori la majorité des composants déployés répondent à un modèle pour leur aspect fonctionnel : le modèle de service (§a), incluant un agent de QoS pour son autogestion et son aspect non fonctionnel qui

répond à un modèle de QoS (§b) conformément au profil de service (§c) relevant du modèle informationnel.

(a) Modèle de service :

Dans la nouvelle génération de service (NGS), le SOA nous permet d'avoir des compositions de services distribués à couplage lâche, de manière indépendante de la localisation géographique. Les éléments de service pouvant être partagés entre différents utilisateurs et sollicités dans différentes localisations, différents contextes et par différents terminaux, ils doivent être flexibles et composables dynamiquement tout en garantissant la QoS demandée pour chaque utilisateur de ces services, ainsi nous avons redéfinis les éléments de service et avons un modèle de service, qui est conforme aux caractéristiques suivantes :

- *Stateless* : afin de pouvoir mutualiser un élément de service pour plusieurs utilisateurs, il doit réaliser les mêmes tâches (même opérations) pour tous les utilisateurs et sans conserver les données ou l'état spécifique à chacun d'eux, pour cela nous avons défini des ESs stateless qui peuvent réaliser des tâches génériques pour tous les utilisateurs sans prendre en compte leurs contextes et sans maintenir leurs données spécifiques. Cette caractéristique garantit la capacité de remplacement dynamique d'un ES par un autre.
- *Autonomie* : dans notre contexte NGS, les services sont constitués d'une composition d'éléments de service dynamique et flexible, ainsi un élément de service doit être autonome. Cette propriété permet de déployer un ES indépendamment des autres éléments et il peut s'exécuter si ses besoins sont satisfaits indépendamment de la façon dont ils le sont. Elle permet aussi de remplacer un élément de service dégradé par un autre sans perturber le fonctionnement des autres ESs.
- *Autogestion* : nous avons défini dans chaque élément de service un agent QoS qui nous permet de surveiller et de contrôler les paramètres de QoS courante de l'ES tout en vérifiant si son comportement est conforme à celui négocié dans le contrat de QoS préétabli.
- *Interfonctionnement* : l'interfonctionnement consiste à assurer l'interopérabilité et l'interconnexion entre les ESs. En effet, la composition du service global est basée essentiellement sur la propriété d'interopérabilité des ESs. Cette approche permet la coopération entre plusieurs ESs tout en évitant les conflits de traitement et de deadlocks. L'interconnexion entre les ESs nécessite de définir des interfaces et des liens génériques.
- *Exposable* : le service vu par le client (catalogue) peut être composé de plusieurs éléments de service, ce service à une valeur marchande et peut faire partie de plusieurs offres différentes. Ainsi nous avons défini l'exposabilité des services qui permet de donner la granularité d'un ES suivant une justification business.
- *Mutualisable* : pour prendre en compte le déploiement de service de grande capacité (installé dans de puissantes plates-formes de service), nous avons défini les éléments de service partageable. Ainsi, nous avons proposé des mécanismes de gestion de cette mutualisation.
- *Ubiquitaire* : avoir des ESs ubiquitaires permet de répondre aux demandes des utilisateurs quelque soit leur localisation, ainsi l'élément de service est toujours disponible à une demande utilisateur (Accès au service demandé dans sa zone ambiante). Ceci facilite le

dimensionnement dans le contexte NGN/NGS. Pour cela, nous proposons de déployer des services ubiquitaires dans différentes zones de localisation afin de couvrir la demande des abonnés et des utilisateurs mobiles. Afin de gérer cette ubiquité, nous avons regroupé les éléments de service selon leur même fonctionnalité et QoS dans une même communauté d'intérêt, nommée VSC (Virtual Service Community).

Pendant la phase de Delivery (Usage), l'intérêt des VSC est de garantir la continuité de service tout en maintenant la QoS demandée par l'utilisateur de bout en bout. Ceci est effectué en remplaçant un ES qui ne continue plus à assurer son contrat par un autre équivalent de la même VSC. Dans le cas où la VSC ne peut assurer ce remplacement, elle envoie une requête de Redéploiement Dynamique d'un ES équivalent vers le Gestionnaire de Déploiement de Service.

Les caractéristiques du modèle de service proposées nous permettent d'affiner les propriétés du SOA et d'avoir ainsi un même modèle de service pour tout type de ressource (ensemble flexible d'éléments de service applicatifs, d'élément de service réseaux et d'éléments de service équipements).

En effet, dans notre proposition, chaque ressource rend un service à travers sa QoS. Ainsi, nous avons :

- *Le Service rendu par l'Équipement* : qui est le traitement de l'algorithme d'allocation qui fournit la CPU et la mémoire à tous les logiciels qui partagent le composant.
- *Le Service rendu par le Réseau* : qui est le traitement de l'algorithme protocolaire qui effectue la délivrance du flux qui répond à la QoS demandée par le protocole.

Le Service rendu par l'application : qui est le traitement de l'algorithme applicatif qui fournit le service selon les conditions spatiale et temporelle de l'environnement ainsi que les préférences des utilisateurs.

Cependant, nous ne considérons pas seulement cet aspect fonctionnel du service, mais nous prenons aussi en compte l'aspect non-fonctionnel du service, à savoir la qualité de service, pour définir notre architecture et de guider les différentes phases du cycle de vie d'un service.

(b) **Modèle de QoS :**

La QoS exprime le comportement d'une ressource, elle définit l'aspect non fonctionnel de cette ressource et représente le SLA entre un fournisseur et un client.

Nous avons modélisé le comportement d'une ressource quelque soit son type pour avoir une représentation homogène. Nous avons retenu les quatre critères nécessaires et suffisants [FB06] [NS07] qui sont :

- *La Disponibilité* : représente l'accessibilité d'une ressource en tenant compte des conditions contractuelles temporelles et spatiales.
- *La Fiabilité* : représente l'aptitude d'une ressource à s'exécuter sans détériorer l'information traitée tout en respectant les demandes et les conditions contractuelles. Elle indique le taux involontaire de modification d'information concernant les nœuds et des liens pendant leur traitement.
- *Le Délai* : représente l'aptitude d'une ressource à s'exécuter en respectant le délai précisé dans les demandes. Il indique la durée de traitement des nœuds et la durée d'attente moyenne passée dans la ressource lien.
- *La Capacité* : indique la charge maximum des nœuds et la largeur de bande utilisable des liens.

Ces quatre critères de QoS, s'évaluent à partir de différents paramètres mesurables, le Tableau IV.1-1 représente la vision opérateur et la vision utilisateur de ces quatre critères.

Tableau IV.1-1 : Critères de QoS

Critère	vision Opérateur	vision Utilisateur
Capacité	Bande passante, débit	Débit du service (requête/s)
Délai	Latence, Gigue, Temps de propagation	Temps de réponse demandé
Fiabilité	Pertes, Erreurs	Taux d'altération pour les requêtes erronées
Disponibilité	CAC, Taux de rejet	Taux d'indisponibilité, Taux d'accessibilité, Délai de rétablissement

Chacun de ces quatre critères est évalué à différents niveaux de visibilité (utilisateur, équipement, réseau et service). Pour évaluer un critère, nous avons des paramètres mesurables selon le niveau de visibilité. Par exemple le critère de Capacité représente la CPU et la mémoire pour un équipement. Dans ce qui suit, nous avons un exemple d'un profil d'équipement PDA N97, contenant toutes les informations de QoS (les quatre critères mesurés par des paramètres).

Profil d'un équipement : Il est composé de quatre parties (Identité, Hardware, Software et E/S) (Figure IV.1-1).

La partie Hardware décrit la QoS offerte à travers les caractéristiques suivantes :

- Disponibilité : mesuré par le type d'alimentation ou niveau batterie.
- Fiabilité : mesuré par le nombre d'instructions erronées par seconde.
- Délai : temps de traversée de l'équipement.
- Capacité : mesuré par les MIPS (vitesse de traitement du processeur) et la mémoire disponible.

```

- <EquipmentProfile>
- <Equipment_id type="1">
  <EquipmentName>PDA</EquipmentName>
  <EquipmentType>Terminal</EquipmentType>
- <Characteristics>
- <Hardware>
  - <QoSCapabilities>
    - <Availability>
      <Memory>128Mo</Memory>
    </Availability>
    <Reliability>0.998888</Reliability>
    <Delay>3000ms</Delay>
  - <Capacity>
    <Processor>434MHz</Processor>
  </Capacity>
  </QoSCapabilities>
</Hardware>
- <Software>
  <OS>Symbian</OS>
  <OSversion>9.4</OSversion>
- <Application>
  <browser>Oui</browser>
</Application>
</Software>
- <IHM>
  <device>screen</device>
  <ScreenSize>640 x 360</ScreenSize>
  <ScreenType>Tactile</ScreenType>
  <device>Keyboard</device>
  <KeyboardType>AZERTY</KeyboardType>
</IHM>
- <Network>
  <BearerService>UMTS</BearerService>
  <BearerSupported>WIFI ,GSM,EDGE,Bluetooth
  ,UMTS,HSDPA </BearerSupported>
</Network>
</Characteristics>
</Equipment_id>
</EquipmentProfile>

```

Figure IV.1-1: Profil d'équipement PDA N97.

Ainsi ces valeurs de QoS dans le profil d'équipement vont servir à instancier la QoS Offerte de cet équipement dans le profil d'usage utilisé pour le déploiement.

En appliquant le modèle de QoS cité ci-dessus, nous pouvons avoir trois types de valeurs pour chaque critère:

1. La **valeur de conception** : déterminée au moment de la conception d'une ressource, traduit ses possibilités maximales. Elle intervient lors de la planification et dimensionnement des services. *Cette valeur va servir à instancier la QoS Offerte d'un service dans notre Profil d'Usage en s'adaptant au contexte de l'environnement de déploiement. Cette valeur est utilisée par le processus de déploiement pour créer le profil d'usage puis sélectionner la plate-forme et les ressources réseaux où sera déployé le service. La sélection se fait en comparant la QoS demandée du service à celle Offerte des ressources (plate-forme et réseau).*
 - La **valeur courante** : indique le comportement courant d'une ressource. *Cette valeur est à surveiller durant la phase d'exploitation par rapport à la valeur seuil à ne pas dépasser.* Nous avons intégré dans chaque élément de service un Agent QoS, qui surveille la valeur courante de QoS, dans le cas où cette valeur dépasse la valeur seuil, il envoie un Out Contrat aux autres éléments de service dans la communauté afin de déclencher le processus de gestion des communautés (VSC).
 - La **valeur seuil** : indique la limite du fonctionnement normal d'une ressource. *Cette valeur représente le seuil alerte utilisé pour déclencher les réactions adéquates et les processus d'autogestion pendant la phase d'exploitation.*

(c) Modèle informationnel

Le modèle informationnel permet la description de n'importe quelle entité du monde réel, (Nœud, Lien ou Réseau : NLR), sur un niveau de visibilité donné « V », suivant le concept abstrait et générique : Network Element(V) (NE(V)) [NS97] [ZBD04], en précisant son architecture matérielle et logicielle ainsi que ses services offerts.

En se basant sur la modélisation NLR et le modèle de QoS défini, nous avons différents profils sollicités aux différentes phases du cycle de vie afin de prendre en compte le contrat de QoS durant toutes ces phases.

2. Pour la phase de conception, nous avons le *profil de ressource*, qui contient les valeurs de conception des critères de QoS,
3. Pour la phase de déploiement, nous proposons un *profil d'usage de ressource* qui sera utilisé pour déployer au bon endroit. En effet, il nous faut connaître les conditions (QoS demandée + contraintes) pour pouvoir offrir la QoS de conception (QoS offerte) adaptée à l'environnement de déploiement,
4. Pour la phase d'exploitation nous avons le *profil real time* qui sera instancié durant l'usage des services à partir des autres profils. Durant cette phase d'exploitation nous aurons un contrôle de la QoS courante par rapport à la QoS seuil à ne pas dépasser.

Profil de ressource : conçu par le fournisseur de la ressource et indépendant de son environnement d'exécution, instancié pour fournir l'information de QoS de conception d'une ressource « QoS Capabilities » qui représente le maximum qu'une ressource peut fournir. Voir exemple de profil d'équipement PDA N97 (Figure IV.1-1).

IV.1.3 Comment déployer l'élément de service ?

Notre but est de s'assurer dès la phase de déploiement, que notre composant a, à sa disposition toutes les ressources nécessaires, en configurant le contrat que doit respecter chaque service. Ainsi, nous pouvons faciliter la phase de gestion, qui, au lieu de surveiller le service et de remonter toutes les informations, va contrôler la conformité du composant à son contrat. Pour ce faire nous comparons la QoS courante par rapport au contrat préétabli et ainsi, nous ne traiterons que les contrats non conformes. Pour effectuer le déploiement nous utilisons un profil d'usage (§a) pour effectuer le mapping de QoS (§b).

(a) Profil d'Usage pour le déploiement

Notre objectif est de déployer l'élément de service sur les ressources adéquates (celles qui garantissent la QoS Demandée). Pour cela, nous avons le « Profil d'Usage de ressource » créé pour la phase de déploiement. Les informations nécessaires au déploiement contenues dans ce profil sont les suivantes :

- *La QoS Demandée* : représente les exigences de l'élément de service envers l'environnement de déploiement afin qu'il fonctionne correctement.
- *La QoS Offerte* : Cette valeur est instanciée à partir de la QoS de Conception adaptée au contexte de déploiement.
- *Les Contraintes* : représente les restrictions de chaque niveau de visibilité (Utilisateur, Service, Réseau et Equipement) sur les possibilités offertes.

La Figure IV.1-2, montre un exemple d'un Profile d'Usage du service VoD avec les informations nécessaires au déploiement.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <ServiceUsageProfile>
  <version>1.0.0.0</version>
  - <Identity>
    <ServiceName>VoD-SFR</ServiceName>
    <ServiceID>1</ServiceID>
    <ServiceType>streaming</ServiceType>
  </Identity>
  - <OfferedQoS>
    <Availability>0.998888</Availability>
    <Reliability>0.998888</Reliability>
    <Delay>3600ms</Delay>
  - <Capacity>
    <UserMax>300</UserMax>
  </Capacity>
</OfferedQoS>
  - <DemandedQoS>
    <Availability>0.998888</Availability>
    <Reliability>0.998888</Reliability>
    <Delay>3000ms</Delay>
    <Capacity>0.5M</Capacity>
  </DemandedQoS>
  - <Constraints>
    - <User>
      <Group>AnyBody</Group>
      <Language>English</Language>
    </User>
    <AccessCost>"0.20/m"</AccessCost>
  </Constraints>
  - <SoftwareReq>
    - <JVM>
      <Edition>Personal Java</Edition>
      <Version>1.2</Version>
    </JVM>
  </SoftwareReq>
  - <Service>
    <State />
    <Preference />
  </Service>
  - <Network>
    <ProtocolRelated>IPV6</ProtocolRelated>
    <ConnectionExtern>TRUE</ConnectionExtern>
  </Network>
  - <Equipement>
    <Processor>2.5GHz</Processor>
    <Memory>2Go</Memory>
    <Application>MediaPlayer</Application>
  </Equipement>
</Constraints>
</ServiceUsageProfile>

```

Figure IV.1-2: Profile d'Usage du service VoD.

Dans ce profil d'usage de service, nous avons quatre parties (Identité, QoS Offerte, QoS demandée et Contraintes).

La partie identité : contient toutes les informations nécessaires à identifier un service comme son numéro identifiant unique, son nom, son type, sa version, etc. Il est à noter que le nœud et le lien rendent un service indépendant, chacun avec sa QoS.

La partie QoS (Offerte ou Demandée) contient :

- La valeur de disponibilité du service qui représente le taux d'accessibilité au service (control d'admission). Le déploiement d'éléments de service ubiquitaire nous permet d'assurer un taux de disponibilité performant et ainsi de contribuer à maintenir la QoS demandée par les utilisateurs. Voir §IV.2.4 pour la gestion des VSC (nœud) et VQC (lien). En effet, nous avons proposé dans la phase de provisioning, un contrôle des requêtes acceptées dans le SE par l'Agent de QoS qui détecte les requêtes qui risquent de ne pas être servies par le nœud (SE) et les envoie vers un autre lien (file d'attente attachée au SE) ubiquitaire.
- La valeur de Fiabilité représente le taux de réponses erronées (nœud) ou de requêtes erronées (lien).
- La valeur de Capacité représente le nombre moyen de requêtes traitées par le SE (nœud) ou acceptées par la file d'attente du SE (lien).
- La valeur de Délai représente le temps de traitement moyen par le SE (nœud) ou par la file d'attente (lien).

La partie Contraintes contient :

- Les contraintes utilisateur, par exemple le coût d'accès au service,
- Les contraintes réseaux, par exemple le Protocole supporté,
- Les contraintes services,
- Les contraintes équipement, par exemple les versions de logiciel.

(b) Mapping de QoS

Le profil d'usage est utilisé pour le déploiement en comparant la QoS demandée par la ressource logique (le service à déployer) et celle offerte par les ressources physiques (plateforme et réseaux) (Figure IV.1-3).

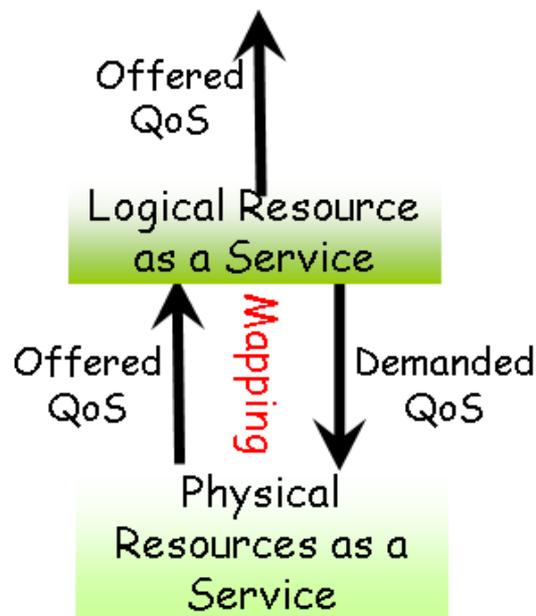


Figure IV.1-3: Négociation de la QoS entre la ressource logique (service) à déployer et les ressources physiques.

Nous avons un exemple de mapping entre la QoS demandée par le service VoD (Figure IV.1-2) et la QoS offerte par deux plates-formes différentes A (Figure IV.1-4) et B (Figure IV.1-5) candidates pour le déploiement.

```

Platform A Usage Profile
<?xml version="1.0" encoding="UTF-8" ?>
- <EquipmentUsageProfile>
  <EquipmentName>VoD Server A</EquipmentName>
  <EquipmentType>Server</EquipmentType>
- <QoSOffered>
  - <Availability>
    <Memory>1Go</Memory>
  </Availability>
  <Reliability>0.88999</Reliability>
  <Delay>700ms</Delay>
- <Capacity>
  <Processor>1.5GHz</Processor>
</Capacity>
</QoSOffered>
- <Constraints>
- <User>
  <AccessPassword />
</User>
  .
- <Service>
- <Software>
  <OS>Windows XP</OS>
  <Application>MediaPlayer</Application>
</Software>
- <Hardware>
  <Device>Screen</Device>
  <Device>Speaker</Device>
</Hardware>
</Service>
- <Network>
  <ProtocoleRelated>IPv4</ProtocoleRelated>
- <NetworkCard>
  <DataRate>80M/s</DataRate>
</NetworkCard>
</Network>
- <Equipment>
  <State />
</Equipment>
</Constraints>
</EquipmentUsageProfile>

```

Figure IV.1-4 : Profil d'Usage de la plate-forme A.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <EquipmentUsageProfile>
  <EquipmentName>VoD Server B</EquipmentName>
  <EquipmentType>Server</EquipmentType>
- <QoSOffered>
  - <Availability>
    <Memory>2Go</Memory>
  </Availability>
  <Reliability>0.998888</Reliability>
  <Delay>2000ms</Delay>
- <Capacity>
  <Processor>2.5GHz</Processor>
</Capacity>
</QoSOffered>
- <Constraints>
- <User>
  <AccessPassword />
</User>
- <Service>
- <Software>
  <OS>Windows XP</OS>
  <Application>MediaPlayer</Application>
</Software>
- <Hardware>
  <Device>Screen</Device>
  <Device>Speaker</Device>
</Hardware>
</Service>
- <Network>
  <ProtocoleRelated>IPv6</ProtocoleRelated>
- <NetworkCard>
  <DataRate>100M/s</DataRate>
</NetworkCard>
</Network>
- <Equipment>
  <State />
</Equipment>
</Constraints>
</EquipmentUsageProfile>

```

Figure IV.1-5 : Profil d'usage de la plate-forme B

Les informations de QoS mappées concernant les deux plates-formes sont la QoS offerte (valeur courante) en termes de capacité de la CPU, de mémoire disponible, de délai courant pour l'exécution du service et de la fiabilité courante de la plate-forme. Les informations concernant le service à déployer sont la QoS demandée du service ainsi que les contraintes d'équipements et de réseaux.

Ce mapping de la QoS, entre le service à déployer et les ressources où sera déployé le service, est facilité du fait que nous avons le même modèle de QoS pour toutes les ressources

(Service, Réseau et Equipement). Ainsi la plate-forme B est choisie car elle répond aux demandes du service.

Le résultat de l'adaptation entre la QoS Demandée par le service et la QoS Offerte par la plate-forme B va initialiser le Contrat de QoS de l'élément de service (QoS Offerte du service déployé).

IV.1.4 **Quand déployer un élément de service ?**

La requête de déploiement d'un service peut venir de :

5. De la phase de Stratégie : qui analyse les plaques cibles des services ubiquitaires qui seront déployés, ainsi que le nombre de prospect, puis envoie une requête à la Gestion du Déploiement des Services Exposables pour déployer de nouveaux services dans différentes zones stratégiques (services ubiquitaires) pour répondre à la demande.
 6. De la phase de Conception / Développement : qui récupère les services à déployer qui sont soit développés par le fournisseur de service, soit achetés chez d'autres fournisseurs. Elle obtient leurs profils afin de composer les services Exposables et de mettre à jour le "real time profile". Puis elle envoie la requête de déploiement avec le « code du service » ainsi que son « Profil de service».
- De la phase d'Exploitation : (et plus spécialement durant les étapes de Gestion) qui envoie une requête de redéploiement dans le cas où les services déployés ne suffisent pas à satisfaire la demande.

IV.1.5 **Processus de Déploiement**

Plusieurs services de gestion sont impliqués dans le processus de déploiement (Figure IV.1-6). Nous avons, ceux qui gèrent les services exposables, ceux qui font la correspondance entre les services exposables et les services de base et ceux qui gèrent les ressources. Dans ce qui suit, nous présentons le rôle de chacun d'eux dans le processus de déploiement.

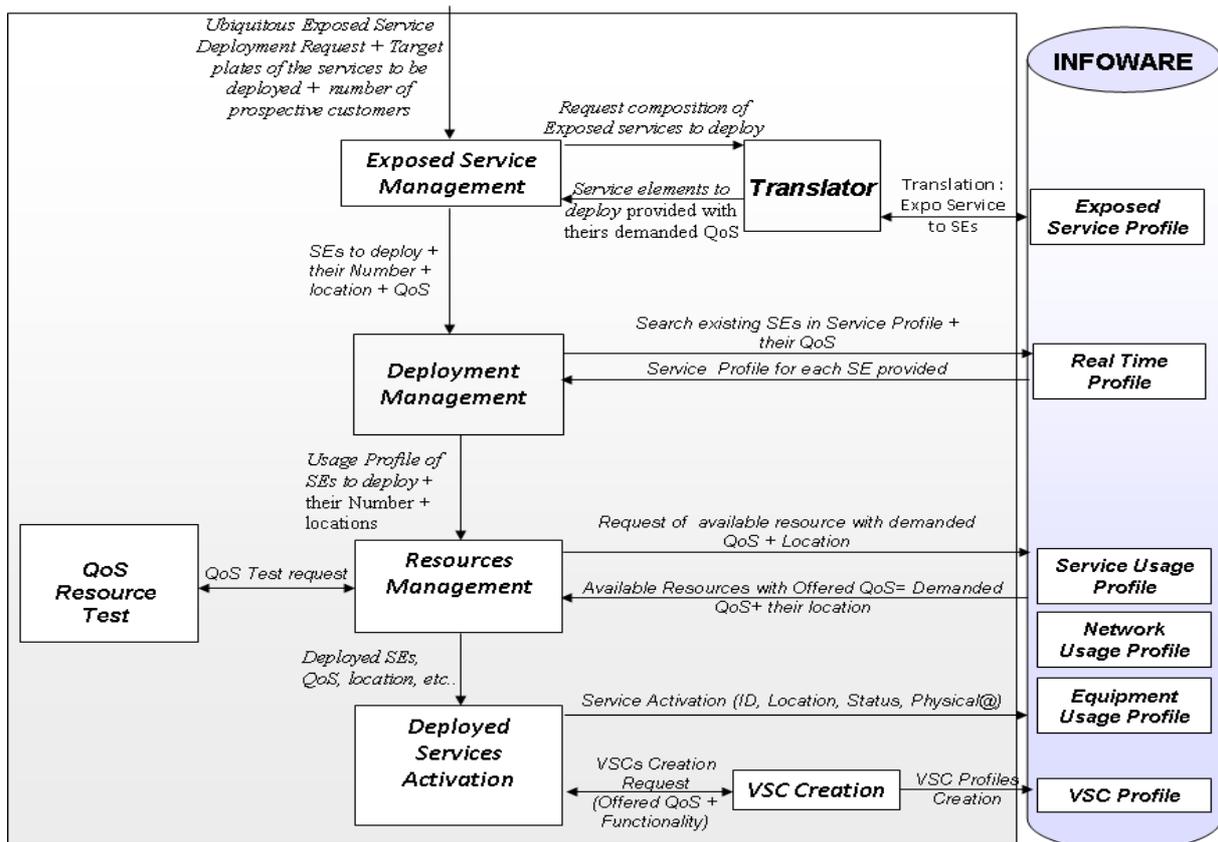


Figure IV.1-6: Processus de déploiement

Service de Gestion des Services Exposables : à la réception d'une requête de déploiement contenant des services exposables à déployer, leurs plaques cibles et le nombre de prospect, il envoie une demande vers le service de gestion « Translator » (proposé par notre groupe de travail) qui traduit un service exposable avec une QoS demandée en une composition d'éléments de service avec une QoS demandée pour chacun. Le « Translator » utilise le profil des services exposables dans l'INFOWARE pour la translation puis envoie la réponse vers le service de gestion des services exposables.

Service de Gestion du Déploiement : le rôle de ce service de gestion est de rechercher dans la base informationnelle l'existence des éléments de service de base et d'étudier leur capacité à supporter les nouveaux services exposables. Si la réponse est oui, il n'y a pas de déploiement à faire. Sinon, il récupère le profil d'usage afin de pouvoir demander le déploiement au niveau des ressources.

Une fois le Usage profile récupéré, il envoie une requête de déploiement à la Gestion du Déploiement des Ressources, avec la liste des éléments de service avec leurs QoS demandées et le nombre des services ubiquitaires à déployer et leurs zones de déploiement.

Service de Gestion du déploiement des Ressources : Le rôle principal de ce service est de mapper la QoS Demandée des éléments de service à déployer et celle Offerte par les ressources qui peuvent accueillir chaque élément de service et ainsi de sélectionner les ressources où sera déployé l'élément de service. Le fait d'avoir un même modèle de QoS et l'existence d'un profil d'usage pour tous les types de ressources va rendre cette correspondance flexible et rapide. Les informations récupérées par ce service de gestion sont en temps réel.

Ainsi ce service va récupérer les ressources ayant un état disponible et ayant suffisamment de QoS offerte pour satisfaire la QoS demandée par le service. Puis il sélectionne dans les candidats reçus de la base informationnelle les ressources choisies. La sélection est faite selon différents critères. Par exemple, la localisation de la plate-forme, où s'il prend la décision pour plusieurs ESs dans une même plate-forme, il mappe entre différents paramètres de QoS comme la CPU, la Mémoire et le nombre maximum d'ESs à déployer et fait la correspondance avec ce que demande l'ES. Enfin, il envoie une requête de test de QoS de bout en bout afin de tester l'intégration du service déployé sur les ressources, avant d'envoyer la demande d'activation vers le service d'activation afin d'activer et de créer les communautés de ces ESs.

Test de QoS de Ressources : le rôle de ce service est juste d'effectuer un test d'intégration sur les ressources sélectionnées, en vérifiant si la QoS de bout en bout (service, réseau et équipement) est conforme au contrat de QoS du service préétabli.

Activation de Service : ce service de gestion reçoit les ESs déployés et sélectionnés par la Gestion de Déploiement des Ressources pour être activés. Puis envoie une requête à la création de VSC pour créer les VSCs aux nouveaux éléments de service déployés.

Enfin il met à jour dans l'Infoware toute l'information de configuration des ESs déployés (ID, QoS, @physique, état, ID de la communauté, etc.)

L'activation n'est pas automatique pour tous les éléments de service déployés car nous proposons de déployer plusieurs éléments de services ubiquitaires afin de prévenir toute demande d'utilisation par les abonnés et les utilisateurs mobiles. Puis afin d'optimiser l'utilisation des ressources, un certain nombre de ESs sera activé et le reste sera activé selon le besoin (les ESs activés ne suffisent plus à couvrir la demande).

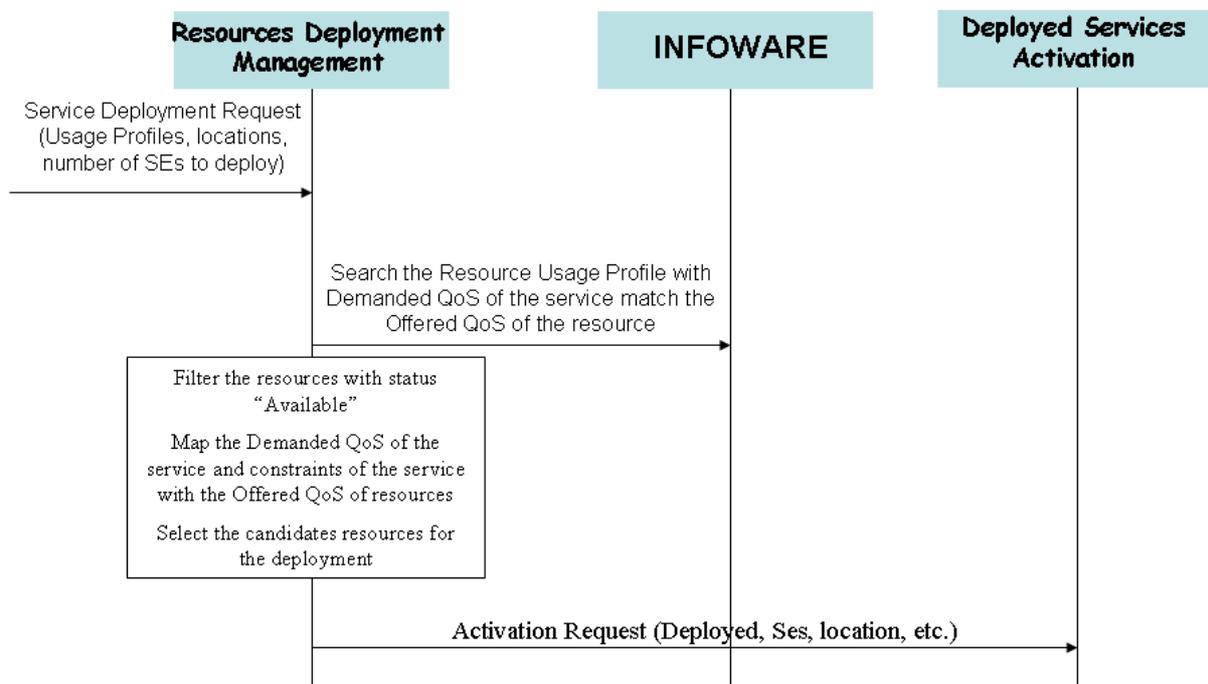


Figure IV.1-7: Diagramme de gestion des ressources service.

Création de Communauté Virtuelle de Service (VSC) : Le dimensionnement spécifie le nombre d'éléments de service ubiquitaires à déployer et leur localisation. Ainsi, selon le nombre d'abonnés et d'utilisateurs mobiles potentiels dans une zone ambiante, nous

déployons un ensemble d'ESs pour satisfaire statistiquement à la demande. Puis, nous activons un certain nombre d'ESs pour l'utilisation et les autres ESs déployés seront activés selon le besoin. D'où l'intérêt d'avoir des communautés de service équivalentes.

Ce service de gestion va regrouper les nouveaux ESs par fonctionnalité et QoS équivalente et créer une VSC pour chaque groupe. Cette VSC n'est pas centralisée, mais existe pour chaque élément de service, ainsi pour la création de la VSC, pour chaque élément de service nous avons un Profil VSC qui contient un identifiant de cette VSC et tous les identifiants des ESs qui lui sont naturellement rattachés.

INFOWARE : Contient toutes les informations sur les ressources, leurs profils, leur état, leurs localisations qui nous permettent de déployer les services. Toutes les informations mises à jour après le déploiement seront utilisées par les processus de Provisioning, Delivery et Management.

IV.1.6 Redéploiement des éléments de service :

La capacité à délivrer de nouveaux services rapidement et de manière économique aide les fournisseurs de services à préserver leur avance sur leurs concurrents et à fournir une réelle valeur ajoutée à leurs clients.

Ainsi, simplifier et écourter le redéploiement de service, peut permettre aux opérateurs et fournisseurs de services de réduire les délais de mise sur le marché des services, afin d'accroître le potentiel de croissance des revenus et des activités.

Durant l'exploitation des services, si la phase de gestion des communautés se rend compte que les services ubiquitaires déployés ne suffisent pas à satisfaire la demande (Provisioning et Delivery), et qu'il n'existe plus de ESs ubiquitaire à activer, elle envoie une requête de redéploiement dynamique vers le processus de gestion du déploiement. Ceci est facilité du fait que nous avons séparés le traitement de l'élément de services, des données et que nous avons des services avec les caractéristiques d'autonomie, de stateless, etc. ce qui veut dire que nous avons le même service assurant la même fonctionnalité et le même contrat déployé dans différentes localisations.

La phase de gestion va envoyer la requête de redéploiement avec l'information sur la localisation où le service sera déployé et son identifiant vers le service de gestion de déploiement de services. Ce dernier, va retrouver le code, le profil d'usage afin de lancer un redéploiement dynamique et rapide.

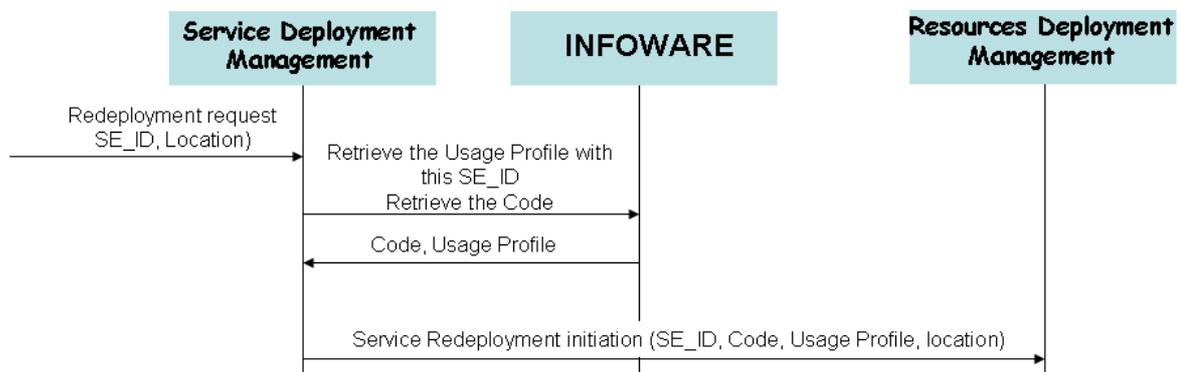


Figure IV.1-8: Diagramme de gestion de redéploiement de ressources

IV.2 Provisioning des éléments de service orienté QoS

Après avoir présenté notre contribution pour prendre en compte le contrat de QoS dès la phase de Déploiement, nous allons consigner dans cette section notre contribution de prendre en compte la QoS pendant la phase de Provisioning *afin d'assurer la continuité du service, de maintenir la QoS de bout en bout et d'optimiser l'utilisation de la ressource service elle-même.*

IV.2.1 Motivations

Le terme « ressource » couvre une large palette de notions en télécommunication, allant de la bande passante aux adresses réseaux en passant par l'espace disque ou mémoire d'un serveur. Toutes ces ressources équipements (puissance de calcul, capacité mémoire ou disque, capacité d'affichage, etc.) et réseaux (capacités d'aiguillage des nœuds, capacités d'écoulement des flux sur les liens ou bandes passante, adresses réseaux etc.) nécessitent d'être provisionnés afin de pouvoir délivrer le service à l'utilisateur. Mais quand est-il de la ressource "Service Applicatif" ? Avec les nouveaux services de plus en plus complexes le Provisioning au niveau réseau et équipement ne suffit plus à toujours assurer la continuité de la session et de maintenir toujours la QoS de bout en bout. Ainsi nous proposons d'avoir un Provisioning au niveau de la ressource service.

Les ressources sont étroitement liées aux paramètres de QoS. En effet, une qualité de service donnée sera respectée si et seulement si les ressources influant sur les paramètres de QoS (débit, perte, délai et disponibilité) sont bien réservées. Et un service ou une application devant offrir une excellente qualité de service se doit d'être fiable, robuste et tolérante aux pannes avant même de se préoccuper du bon acheminement des données au niveau réseau. Notre Provisioning de service proposé est orienté QoS, il permet de provisionner la ressource « service applicatif » selon sa QoS afin d'optimiser son utilisation.

Les éléments de service dans notre vision sont mutualisables entre plusieurs requêtes utilisateur, ainsi ce partage de l'ES est effectué selon sa QoS courante en termes de Disponibilité, de Fiabilité, de Délai et de Capacité (modèle de QoS).

IV.2.2 VPxN et le Pré-Provisioning des éléments de service

Pour l'aspect architectural de notre proposition, nous proposons une abstraction des ressources à provisionner pour la session d'un utilisateur à tous les niveaux de visibilité en Réseaux Privés Virtuels (Virtual Private Network : VPxN, x : Service, Réseau ou Equipement) [ZBD06] (Figure IV.2-1).

Le service exposable demandé par l'utilisateur est parfois composé de plusieurs éléments de service. Ces éléments de service sont liés par une logique de service. Pour répondre à la demande de l'utilisateur quand il initialise sa session, nous constituons d'abord cette logique de service en prenant en compte la mobilité et les préférences de l'utilisateur (préférence d'un service, d'un réseau d'accès, d'un terminal, d'un opérateur ou d'un prix par rapport à un autre).

Ainsi, à l'initialisation de la session, suivant la logique de service de l'utilisateur, nous sélectionnons les éléments de service demandés par l'utilisateur pour être utilisés durant sa session. Ces éléments de service vont constituer son VPSN. La transition d'un élément de

service vers un autre passe par un réseau de transport, ainsi nous sélectionnons aussi les ressources réseaux (VPCN) et équipements (VPEN) nécessaires à cette logique de service.

L'avantage d'avoir ces VPxN est que nous n'allons pas mobiliser des ressources inutilement. Nous ne faisons que sélectionner la ressource avec la bonne QoS au moment de l'ouverture de la session. Puis, Au moment de la demande de consommation, cette ressource service sera allouée dynamiquement en fonction de sa QoS et pourra être partagée entre plusieurs requêtes en fonction de la QoS qu'elle offre. Un autre avantage d'avoir ces VPxN est que pendant la mobilité ou changement de préférences, chaque ressource dans le VPxN peut être remplacée par une autre ressource ubiquitaire pour maintenir toujours le SLA de l'utilisateur.

Ainsi, nous avons :

Le VPSN : Réseau Privé Virtuel des éléments de services sélectionnés pour être provisionnés (logique de service) dynamiquement durant l'usage (transaction de l'utilisateur).

Le VPCN : réseau privé virtuel des connectivités sélectionnés pour être provisionnés (Virtual Path, Virtual Channel, Routing machines, etc.) dynamiquement durant l'usage.

Le VPEN : réseau privé virtuel des équipements sélectionnés pour être provisionnés (Terminaux, Point d'accès, Routeurs, Edges, Serveurs, etc.) durant l'usage.

La sélection des ressources faisant partie du VPxN se fait selon la QoS Offerte (QoS Prévisionnelle). Puis, quand l'utilisateur envoie sa requête pour utiliser des services faisant partie de sa session (VPxN), nous avons un Provisioning dynamique de toutes les ressources (à tous les niveaux de visibilité) selon leur QoS courante. Nous provisionnons dynamiquement la requête, car entre l'initialisation de la session et l'usage, la QoS courante de chaque ressource peut changer où se dégrader.

La mobilité nous impose de partitionner le réseau global en zones ambiantes successives. C'est pourquoi, nous aurons une re-sélection dynamique pour maintenir la QoS de bout en bout durant la mobilité et optimiser l'utilisation des ressources locales (la ressource ubiquitaire la plus proche). Puis, selon la localisation de l'utilisateur et grâce aux ressources ubiquitaires, nous sélectionnons la ressource qui répond aux préférences de l'utilisateur et qui satisfait la QoS requise.

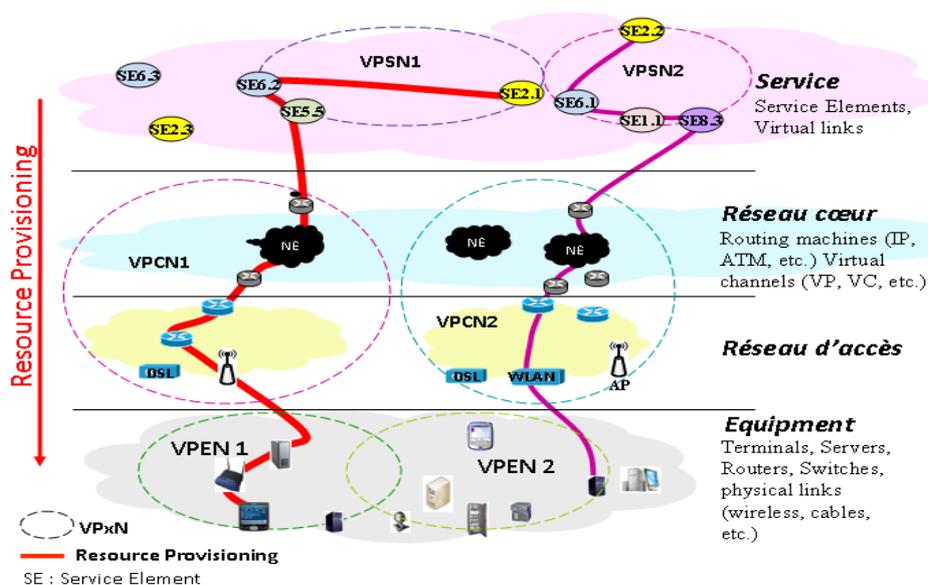


Figure IV.2-1 : Provisioning des ressources à tous les niveaux de visibilité

En conclusion, ce concept de VPxN, nous permet d’avoir une continuité de service durant la mobilité en sélectionnant dynamiquement les ressources selon leur QoS courante dans la zone ambiante de l’utilisateur.

IV.2.3 Provisioning de l’élément de service

Pendant l’Usage, nous avons une réservation dynamique des ressources sélectionnées (phase de Pré-Provisioning) dans le VPxN selon leur QoS Courante.

Notre ressource service étant mutualisée entre plusieurs requêtes, nous devons l’allouer dynamiquement selon sa QoS Courante afin d’optimiser son utilisation. Cette QoS agrège les quatre critères de notre modèle de QoS (Disponibilité, Fiabilité, Délai et Capacité).

Pour gérer cette mutualisation (Figure IV.2-2 : **“Resource Service” et exemple de Mutualisation de requêtes.**Figure IV.2-2) nous proposons d’associer une File d’attente à chaque élément de service qui contiendra toutes les requêtes acceptées par cet ES.

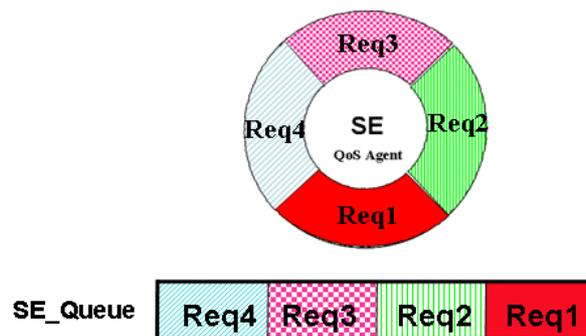


Figure IV.2-2 : “Resource Service” et exemple de Mutualisation de requêtes.

Les requêtes sont acceptées dans la file selon la QoS en terme de :

Disponibilité/Accessibilité : qui représente le taux de requêtes non acceptées par l’ES. *Pour maximiser la disponibilité du ES, nous déployons un nombre d’éléments ubiquitaires pour couvrir la demande (Dimensionnement).*

Fiabilité : qui représente le taux d’erreurs des requêtes adressées à l’ES.

Capacité : qui représente la capacité de traitement moyenne durant une unité de temps.

Délai : qui représente le temps de traitement moyen.

Pendant l’Usage, les requêtes acceptées par un ES selon la QoS demandée sont insérées dans sa file d’attente. L’avantage de notre proposition est que nous prenons en compte tous les critères de QoS pour le provisioning.

IV.2.4 Autogestion de la ressource service

Nous avons intégré un Agent QoS dans chaque élément de service. Cet Agent QoS permet de contrôler la QoS courante de l'ES et dans le cas où cette QoS se dégrade, il sera remplacé dans les sessions actives (VPSNs actifs) par un autre ES ubiquitaire.

Pour gérer cette ubiquité, nous utilisons le concept des VSC (Virtual Service Community) [NS09]. Chaque ES fait partie d'une VSC regroupant un certain nombre d'éléments fonctionnellement et QoS équivalents.

L'Agent QoS compare la valeur courante de QoS de l'ES à sa valeur seuil (à ne pas dépasser). Dans le cas où cette valeur reste inférieure à la valeur seuil il notifie par un « IN Contrat » les autres éléments de service de sa VSC. Dans le cas où cette valeur est supérieure à la valeur seuil, il notifie par un « OUT Contrat » les autres membres de sa communauté (Figure IV.2-3) afin qu'il soit remplacé dans les sessions actives.

Cette notification peut se faire ou bien en multicast vers tous les membres de la communauté et le choix du remplaçant se fera par le premier qui répond à l'ES dégradé ou en peer-to-peer, chacun notifiant son voisin.



Figure IV.2-3 : Autogestion par l'Agent QoS et VSC

Puisque chaque ES fait partie d'une VSC et possède sa propre file d'attente, nous proposons de gérer ces files d'attentes appartenant aux ES ubiquitaires (même VSC) dans des communautés de files d'attente ubiquitaires (Virtual Queue Communities : VQC). Ces VQCs nous permettent de maintenir le SLA de chaque requête dans la file d'attente. L'avantage des VSCs étant de maintenir la QoS de l'ES dans tous les VPSNs actifs, l'avantage d'avoir ces VQC est de pouvoir envoyer immédiatement les requêtes non servies par l'ES vers la file d'attente disponible et ayant la capacité de recevoir cette requête (respect des SLAs des requêtes).

Ainsi, ces deux concepts de VSC et de VQC nous permettent d'avoir une autogestion permettant de maintenir la QoS et d'anticiper les violations des SLAs.

Dans ce qui suit, nous allons détailler le mécanisme de gestion de la QoS au niveau de la file d'attente.

IV.2.5 Mécanisme de gestion de la file d'attente

Nous avons intégré pour chaque file d'attente un Agent QoS qui control la QoS de la file selon les quatre critères de QoS en terme de :

Disponibilité/Accessibilité : elle représente le taux de requêtes rejetées par la file.

Pour maximiser la disponibilité de la file, nous déployons un nombre de files ubiquitaires pour couvrir la demande (Dimensionnement).

Fiabilité : elle représente le taux d'erreurs généré par la file.

Capacité : elle représente la taille de la file.

Délai : il représente le temps d'attente moyen pour chaque requête.

L'Agent QoS de la File d'attente permet d'assurer le respect des délais des requêtes dans la file. Car toutes les requêtes dans la file d'attente ont une QoS de bout en bout à respecter et un délai à ne pas dépasser. Si le délai d'attente dans la file dépasse un délai seuil, la requête sera envoyée vers une file ubiquitaire qui peut satisfaire cette requête.

Cet Agent contrôle le temps d'attente des requêtes et dans le cas où il détecte un Out Contrat il notifie le VQC qui va exécuter son algorithme afin d'envoyer celle dont le délai va expirer vers une autre file ubiquitaire.

Le mécanisme de gestion (Figure IV.2-4) est comme suit :

- L'agent QoS de la file d'attente contrôle dynamiquement la QoS de la file en comparant la valeur courante à la valeur seuil à ne pas dépasser (ces valeurs dépendent fortement du délai d'attente des requêtes). La QoS de la file dépend de la QoS de l'ES. par exemple si l'ES traite dans un délai moyen donné et une capacité d'un nombre de requêtes par unité de temps, ces valeurs vont nous déduire le temps d'attente moyen pour chaque dans la file.
- Dans le cas où la valeur courante dépasse la valeur seuil (signifiant qu'une requête a prit plus de temps de traitement par exemple générant un retard d'attente pour la requête suivante dans la file), l'agent QoS va notifier le « VQC Management » par un Out contrat.
- Le « VQC Management » va déclencher son algorithme de gestion et calculer l'estimation du temps que va passer toutes les requêtes qui sont provisionnées après cette requête qui a pris du retard. Il va ainsi décider lesquelles risquent d'expirer.
- Le « VQC Maintenance » interagit avec notre base informationnelle « INFOWARE » afin de trouver une file d'attente ubiquitaire disponible (VQC Profile) pour chaque requête sélectionnée (ID_Request, ID_VPSN de la requête).
- Pour chaque ID d'une requête sélectionnée comme risquant de ne pas maintenir son SLA, le VQC envoie toutes celles ayant le même VPSN_ID vers une même file d'attente ayant la capacité de prendre ces requêtes. Ceci est nécessaire car dans une file d'attente nous pouvons avoir plusieurs requêtes d'un même utilisateur (même VPSN) et nous devons garder la traçabilité de la session et de la transaction.

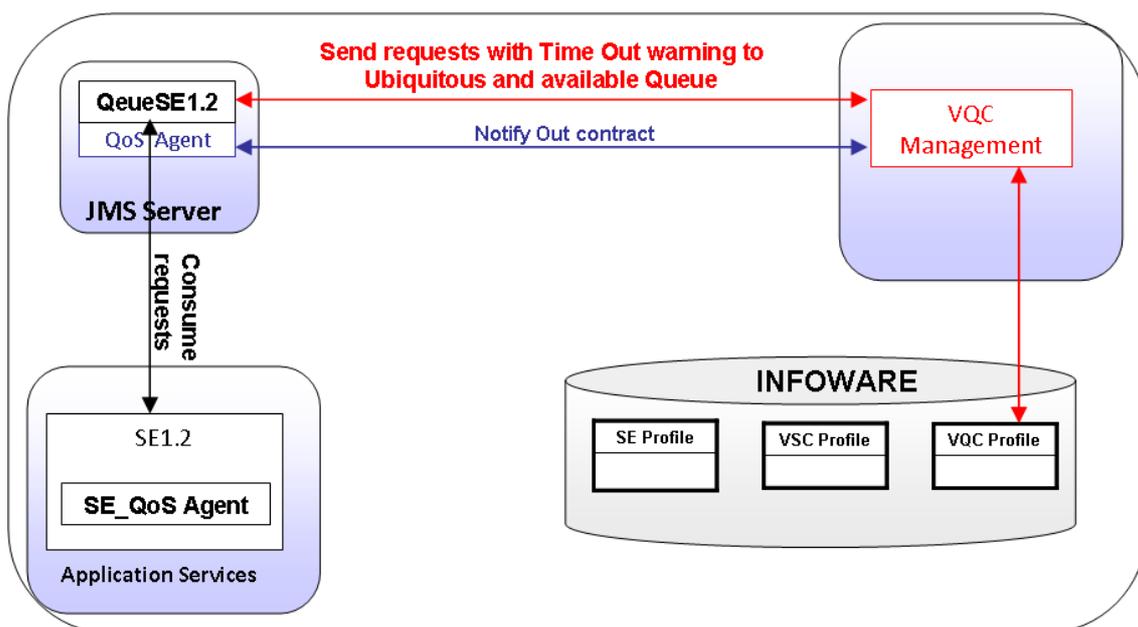


Figure IV.2-4: Mécanisme de gestion de la file d'attente.

Dans la Figure IV.2-5, nous avons un exemple où la requête « Req3 » est dans la file d'attente de l'élément de service SE5.5 car sa QoS accepte cette requête. La file contient aussi trois autres requêtes (Req0, Req1 et Req2). Pendant l'Usage, la requête « Req0 » va prendre plus de temps de traitement sur SE5.5 que ne prévoit la moyenne. Ceci est dû, par exemple, à un grand nombre de données à traiter ou à des accès plus long vers les bases données et non pas à un dysfonctionnement de l'ES (sinon l'agent QoS du nœud l'aurait signalé et c'est aux VSC de gérer cette dégradation). L'agent QoS de la file d'attente va notifier le VQC d'un « Out contrat » afin contrôler le temps d'attente de chaque requête dans la file. Dans ce scénario, le temps d'attente de la requête « Req1 » va dépasser le temps d'attente moyen, le VQC_Management va déclencher son algorithme de calcul. Ainsi le VQC trouve que « Req3 » risque de dépasser son temps d'attente dans cette file et prend la décision d'envoyer cette requête vers une file d'attente qui répond à sa demande. Dans cet exemple la requête est envoyée vers la file d'attente de SE5.4.

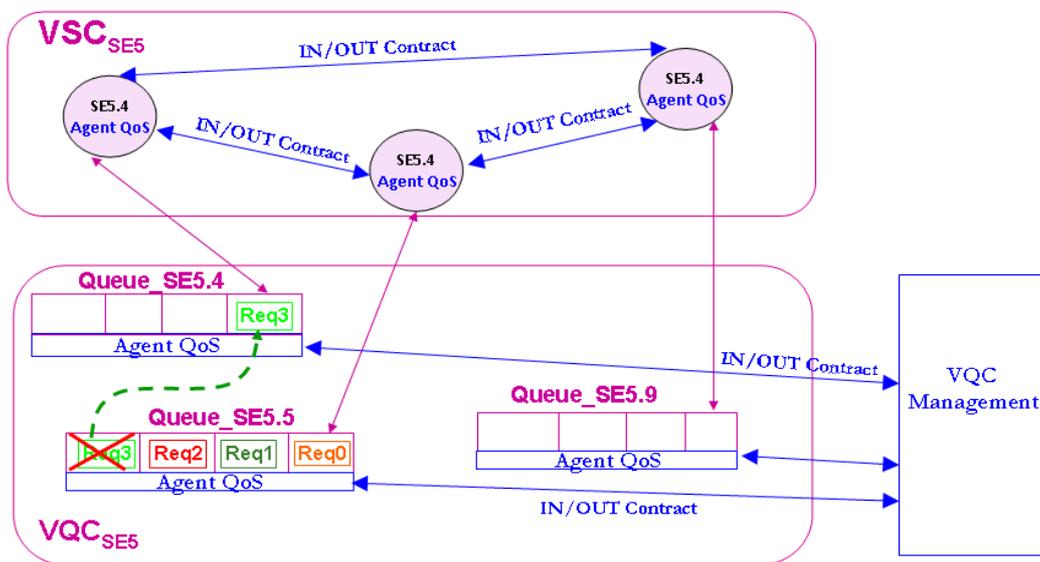


Figure IV.2-5: autogestion (VSC, VQC).

IV.2.6 Algorithme de gestion de la file d'attente

Cet algorithme est déclenché par le «VQC Management» à la suite d'une notification « Out contrat » reçue par l'Agent QoS. L'agent QoS envoie la notification si le délai d'attente moyen d'une requête dans la file dépasse un certain seuil. La valeur seuil permet de ne pas déclencher l'algorithme tout le temps mais d'attendre d'atteindre un seuil qui nécessite de contrôler les autres requêtes par le VQC.

L'algorithme du VQC permet de détecter et sélectionner les requêtes qui seraient susceptibles d'être en retard et de les envoyer vers une file d'attente ubiquitaire avant que ce délai n'expire. La Figure IV.2-6, montre un exemple de quatre requêtes dans la file avec un temps moyen d'attente dans la file et un temps d'attente en temps réel, ainsi l'algorithme va détecter que la requête 4 risque de ne pas respecter le délai d'attente moyen et va ainsi l'envoyer vers une file d'attente ubiquitaire.

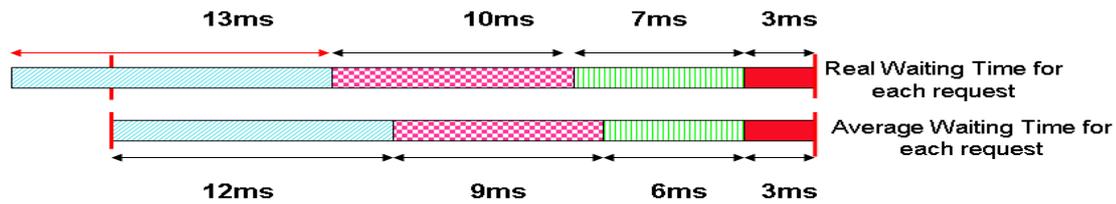


Figure IV.2-6: Exemples des délais d'attente dans la file d'attente

Algorithme

Délai_Queue : délai d'attente de la dernière requête

Capacité_Queue = nombre de requêtes dans la file

Pour $i=0, i < \text{Capacité_Queue}$ {

Temps d'attente moyen [i];

Temps d'attente réel [i] = System.currentTime - JMSTimeStamp ;}

Control :

Pour $i=0, i > \text{Capacité_Queue}$ {

si (Temps d'attente réel [i] > Temps d'attente moyen [i])

{

Pour $j= i+1, j \leq \text{Capacité_Queue}$

{

Si (Temps d'attente moyen [j] + (Temps d'attente réel [i] - Temps d'attente moyen [i]) >

Délai_Queue)

{ID_Request sélectionnée pour être envoyé vers file ubiquitaire}

}}}

IV.3 Service Delivery

Notre proposition a pour but de fournir un support et de définir les processus pour tenir compte de l'usage dans un contexte NGN/NGS c.-à-d. essentiellement un contexte mobile orienté User Centric. En effet entre le service commandé par l'utilisateur et son utilisation qui dans ce contexte est dynamique, il nous faut non seulement provisionner le service au moment de son activation, mais réapprovisionner en continu en fonction du lieu et du terminal utilisé.

De plus le service présenté sur un catalogue, que nous avons nommé service exposable (par exemple le Triple Play) sollicite en termes de ressources plusieurs composants de service de base (Voix, Données, TV) que nous devons intégrer dans notre gestion dynamique (Provisioning, Reprovisioning et Assurance).

Dans cette partie, nous commençons d'abord par expliquer dans (§IV.3.1) nos motivations sur les besoins impactés par les différentes mobilités (User, Terminal, Réseau et Service) et les réseaux ambiants proposant des services ubiquitaires afin de répondre au contexte NGN/NGS. Pour gérer cette dynamique pendant l'usage de service (pendant le déplacement), nous proposons de nous baser sur le modèle NLR [NS97] [ZBD04], qui nous permet d'avoir une traçabilité nous permettant de suivre la session dynamique entre les différents niveaux (Service, Réseau et Equipement) (§IV.3.2). Dans (§IV.3.3), nous présentons le processus service Delivery de bout en bout qui tient compte de l'usage et qui complète le média Delivery. Puis nous détaillons les modèles (VSC et ES Ubiquitaire) qui gèrent la mobilité dans le processus Delivery dans (§IV.3.4).

IV.3.1 Motivations

Dans le but de réaliser le concept de Service Delivery, il est nécessaire de se concentrer sur la gestion de la mobilité, le Provisioning flexible des ressources et la garantie de la continuité de services et la QoS dynamique de bout en bout pendant la mobilité.

Tout d'abord, l'environnement ambiant est actuellement un environnement très hétérogène composé de différentes technologies d'accès, différents services et réseaux. La compétition et la coopération de différents acteurs du marché sont facilitées par la définition d'interfaces, qui permettent la négociation des contrats. Ce nouvel environnement permet aux services d'être omniprésents, i.e. Les mêmes services à partir de différents fournisseurs sont visibles à l'utilisateur.

En conséquence, ce nouveau contexte ambiant nous met dans le défi de :

- *D'offrir des services ubiquitaires.* Ceci vise à fournir les mêmes services dans l'environnement ambiant de l'utilisateur durant la mobilité.
- *De prendre en compte les préférences de l'utilisateur.* Ceci peut influencer le choix d'un service, d'un réseau d'accès, d'un terminal, d'un opérateur ou d'un prix par rapport à un autre.
- De fournir une personnalisation à travers une *composition de services hétérogènes* pendant sa session ce qui nécessite une *gestion horizontale indépendante* des différents niveaux architecturaux (Service, Network et Equipment).

Un autre challenge majeur est la gestion des différentes mobilités (Terminal, Réseau, Service, Utilisateur). Ce contexte mobile nous impose de :

- *De garantir la continuité de service* à travers tous les types de mobilités.
- *D'assurer le Re provisioning des ressources* durant la mobilité. Les nouveaux usages induisent la considération des préférences et de la mobilité de l'utilisateur.

Notre but est de répondre à ces besoins en proposant un processus de Service Delivery de bout en bout dirigé par nos modèles NLR, de service et de QoS. Ce Service Delivery prend en charge les changements impactés par les différents types de mobilités sur la plate-forme de service.

IV.3.2 Service Delivery dirigé par un modèle

Notre modèle consiste en plusieurs concepts afin de répondre aux besoins identifiés dans la section précédente.

- L'élément de service (SE) : nous le modélisons selon les propriétés d'ubiquité et de mutualisation (partageable par plusieurs utilisateurs). Son ubiquité répond essentiellement à l'environnement ambiant, les éléments de service sont équivalents en fonctionnalité et en QoS et installés sur une ou plusieurs plates-formes. La mutualisation permet le partage entre plusieurs utilisateurs, ce qui par conséquent ne permet pas d'avoir des options ou des méthodes. C'est pourquoi pour la personnalisation des services nous avons recours à la composition de services.
- Le model NLR : nous permet de modéliser le monde réel hétérogène à travers des nœuds et des liens. L'ensemble des nœuds et des liens de même nature va constituer un réseau virtuel. Ces réseaux virtuels auront naturellement une gestion horizontale par construction architecturale.

- Le concept VPxN : nous permet le Provisioning des ressources dans chaque niveau de visibilité. Grâce aux VPxN, pendant la mobilité nous pouvons avoir l'environnement ambiant de l'utilisateur (toutes les ressources équipements, réseaux et services qui permettent la continuité de la session dans la nouvelle localisation).
- Le concept VxC : nous permet la gestion de la session (Agrégation des trois niveaux de visibilité) pendant la mobilité. Les VxC vont contenir les ressources fonctionnellement et QoS équivalentes. Grâce aux VSCs, durant la mobilité nous pouvons anticiper les dégradations en remplaçant la ressource dégradée par une autre ressource ubiquitaire.

Avec l'application de ces modèles, nous avons notre architecture globale comme suit :

- Le VPEN qui regroupe tous les équipements ambiants à l'utilisateur (terminaux, équipements réseaux et plate-forme de services).
- Le VPCN qui est constitué du réseau d'accès et réseau cœur. Il regroupe toutes les connectivités ambiantes à l'utilisateur (accès wifi, accès BTS, accès ADSL, cœur IMS,...).
- Le VPSN qui constitue le réseau logique de tous les services auxquels l'utilisateur a droit. La gestion de ces services se fait horizontalement (sans dépendre d'une infrastructure réseau particulière).

Chaque niveau de visibilité (VPEN, VPCN et VPSN) est dynamique et se gère horizontalement. Puis c'est la gestion de la session qui prend en compte les différents niveaux de visibilités pour le calcul de la QoS de bout en bout. Par exemple pour une préférence d'un terminal de l'utilisateur, voir quel composant ubiquitaire prendre, par quel réseau passer, afin de garantir la continuité de la QoS.

Enfin, pour répondre aux besoins de gestion dynamique des éléments de service durant la mobilité, les VxCs surveillent le comportement de chaque ressource et procèdent à son remplacement dans chaque VPxN.

Ainsi, nous avons (Figure IV.3-1) trois niveaux de visibilité dans l'architecture :

Dans le niveau équipement nous avons représenté tous les équipements faisant partie du PAN de l'utilisateur, les équipements de son réseau d'accès et de son réseau cœur et les serveurs sur lesquels sont installés les éléments de services auxquels il a droit.

Dans le niveau réseau nous avons représenté le réseau d'accès et réseau cœur choisi pour sa session.

Dans le niveau service nous avons représenté le réseau logique (VPSN) de tous les services choisis pour sa session et auxquels il a droit.

A partir de ces VPxN, dans une localisation donnée, nous pouvons avoir ainsi tout l'environnement ambiant à l'utilisateur. Par exemple l'utilisateur dans sa première localisation il a choisi d'utiliser son téléphone portable comme terminal pendant sa session, il passe par le réseau d'accès et réseau cœur (VPCN1) choisi et il est susceptible d'utiliser les éléments de service (SE2.1, SE1.2 et SE6.2) pendant sa session. Ces éléments de services sont sur deux plates-formes différentes (SP1 et SP2) et sont ubiquitaires (de même couleur sur la Figure IV.3-1).

Pendant cette session, nous avons une mobilité de l'utilisateur (changement de terminal d'un « téléphone » vers un « ordinateur »), ce changement peut nécessiter un changement de réseau d'accès et de réseau cœur (VPCN2 sur la figure). Ceci peut également nécessiter un changement de composant de service (par exemple le composant adapté au nouveau terminal), ou de garder le même composant sur la même plate-forme tout en changeant de réseau et de terminal. Dans l'exemple l'utilisateur a un service ubiquitaire (SE2.3) qui remplace SE2.1 sur une autre plate-forme (SP2). Ce changement peut être fait pendant la transaction, suite au

calcul de la QoS de bout en bout. En effet, si, il y a dégradation, le choix d'un autre composant ubiquitaire sera fait afin de garantir la QoS demandée. Après mobilité, l'utilisateur a aussi deux nouveaux éléments (SE7.1 et SE5.5) qui sont adaptés à son terminal et remplacent le SE1.2. La gestion de VSC prend en charge le remplacement efficace des éléments de service par anticipation.

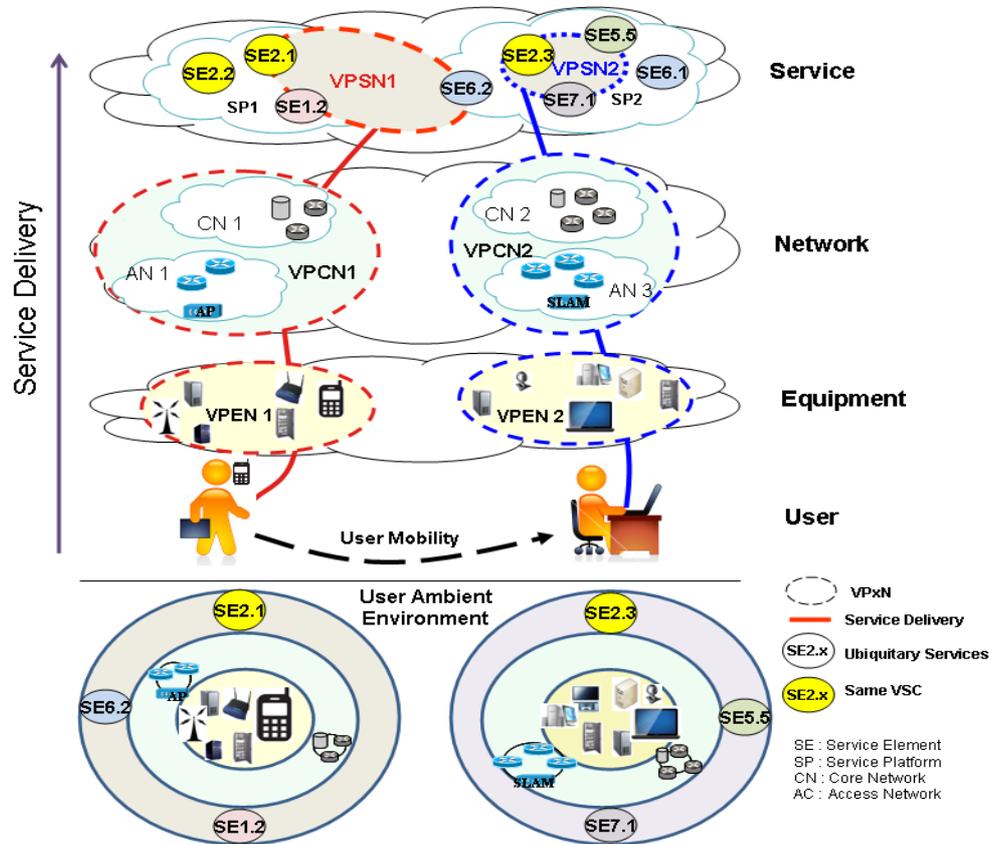


Figure IV.3-1: Service Delivery avec une continuité de QoS selon la mobilité, les préférences de l'utilisateur et l'environnement ambiant

Nous venons de synthétiser l'utilisation de nos modèles dans tout le système, pour montrer la portée de la récursivité. Dans le développement de cette thèse nous avons surtout travaillé au niveau service, mais dans un souci de fournir une approche de bout en bout. Nous présentons le processus de Service Delivery de bout en bout dans la section suivante.

IV.3.3 Processus de Service Delivery

Lorsque l'utilisateur commande des services, le fournisseur de services va lui créer un Profil Utilisateur. Ce Profil Utilisateur va contenir tous les services exposables souscrits par l'utilisateur ainsi que les préférences exprimées par l'utilisateur (Réseaux d'accès, terminaux, services, horaires, Agenda, localisation, etc.).

À l'ouverture de la session (Figure IV.3-2), suivant sa logique de service, l'utilisateur va invoquer à travers une requête de type SIP ses services exposables qu'il sera susceptible d'utiliser tout au long de sa session, la

traduction de ces services exposables en éléments de service constituera son « VPSN ». Chaque élément de service dans le VPSN fait partie d'une VSC qui le gère dynamiquement pendant les mobilités ou suite à n'importe quel changement.

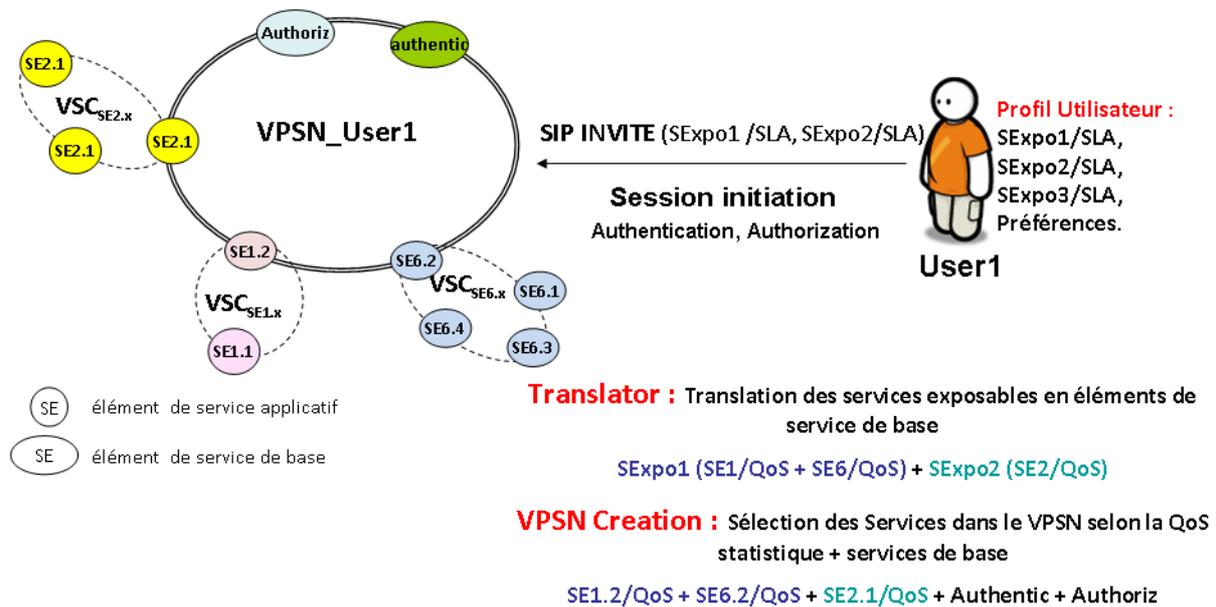


Figure IV.3-2: Création du VPSN

Durant la transaction (Figure IV.3-3), l'utilisateur envoie sa requête de consommation des services selon sa logique de service.

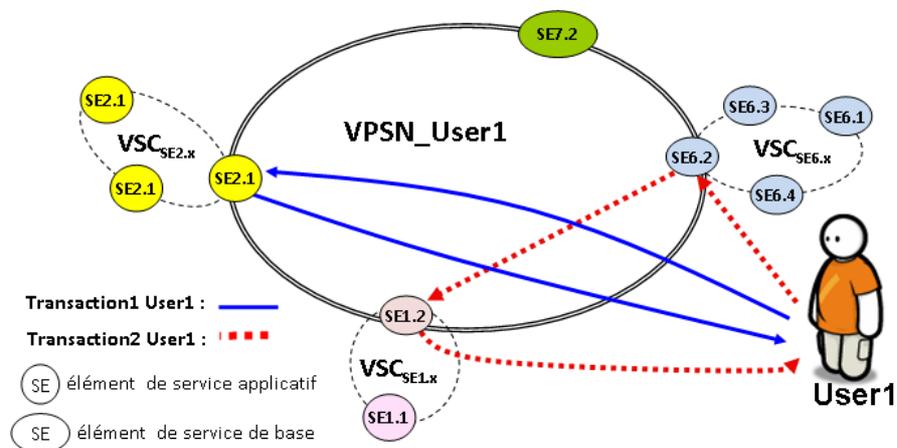


Figure IV.3-3 : Transaction.

Dans ce qui suit nous allons détailler le processus durant l'usage, qui représente la valeur ajoutée du "service Delivery".

A l'ouverture de la session, l'utilisateur indique les services exposables et leur SLA qu'il désire solliciter durant sa session. Ces services exposables sont traduits en éléments de services de base avec les QoS associées par le service de gestion « Translator ». Comme nous l'avons présenté dans le chapitre sur le provisioning, il y aura création du VPSN avec sélection des composants de base qui constituent la composition de service. En fait cela se fait à tous les niveaux de visibilité, ainsi nous avons le processus (VPSN Provisioning) qui va

provisionner (sélectionner) la ressource service dans le VPSN. Ce processus va envoyer la demande vers le processus (VPCN Provisioning) afin de sélectionner les ressources réseaux appropriés aux services commandés. Puis la requête est envoyée vers le processus VPEN Provisioning pour sélectionner les équipements utilisateur, réseaux et services appropriés à la demande. Ces trois processus communiquent avec la base informationnelle pour avoir les ressources disponibles, leurs QoS statistiques, leurs adresses, etc.

Durant l'usage, nous avons le processus Service Provisioning & Delivery (Figure IV.3-4). Ce processus va gérer la mobilité et les changements dans l'environnement ambiant de l'utilisateur. Nous avons deux phases dans le processus :

Dans la phase Provisioning, nous avons 'Service Composition' qui s'occupe de gérer la logique de service de la transaction ainsi que le routage sémantique entre les éléments de service sollicités dans la transaction.

Dans la phase de Delivery, nous avons les processus de Re provisioning des VPSN, VPCN et VPEN. Chaque VPxN s'autogère dynamiquement et re provisionne les ressources pendant la mobilité, un changement de préférences ou une dégradation d'une ressource. C'est la session qui s'occupe d'agrèger les trois niveaux afin de rendre le service demandé avec un maintien de QoS. Ainsi le processus 'Session Management' regroupe les trois processus de Re provisioning et de gestion des communautés qui en s'autogérant assurent une continuité de la session.

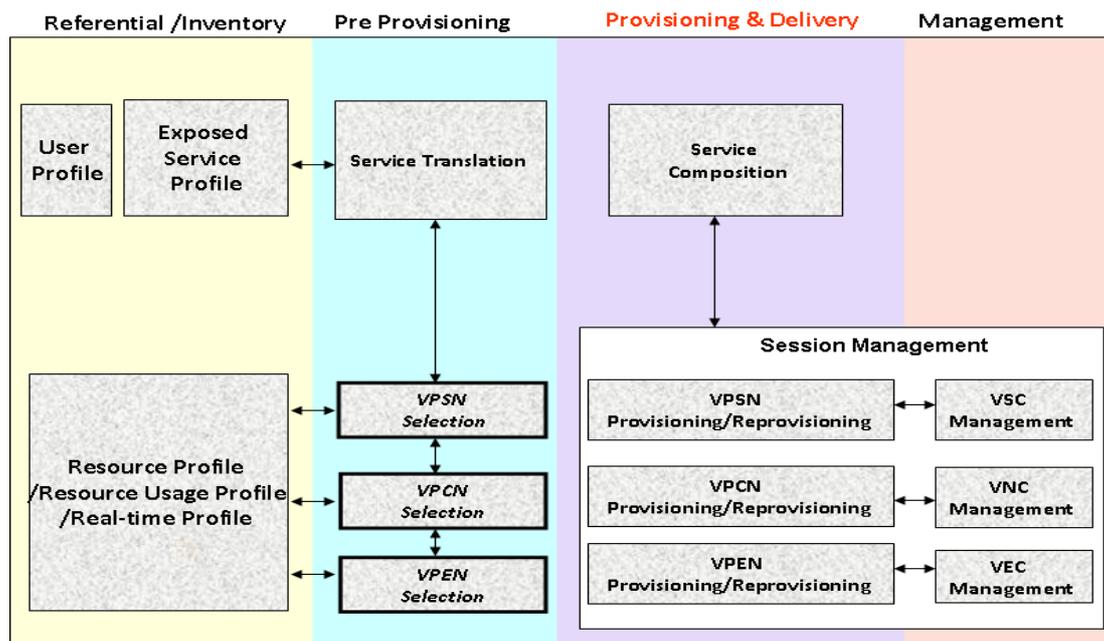


Figure IV.3-4: Processus Service Delivery et son interaction avec le Provisioning et le Management.

Durant les changements (mobilité, préférences), nous avons le processus de Management (Figure IV.3-4), où nous avons les composants de gestion des VSC, VNC et VEC. Ces composants surveillent le comportement des ressources. Chaque ressource annonce sa conformité ou sa non-conformité au contrat avec les autres membres de sa communauté. Dans le cas de non-conformité dû par exemple à une dégradation, la ressource envoie un Out Contrat aux autres membres de la communauté et elle est remplacée dans chaque VPSN, VPCN ou VPEN associé, par une autre fonctionnellement et QoS équivalents dans sa communauté.

La gestion de la mobilité est traitée par les VSCs et les éléments de service ubiquitaires. Nous la détaillons ci-dessous.

IV.3.4 Prise en compte de la mobilité (VSC et ES Ubiquitaire)

Les VSCs sont construites au moment du déploiement des services. Nous avons vu que quand un service est déployé sa communauté des éléments de services ubiquitaires avec une QoS et un fonctionnement équivalent est créée. Cette communauté aura pour finalité d'autogérer les dysfonctionnements ou les dégradations de QoS. Le but d'avoir ces communautés est que lorsqu'un élément de service est dégradé et ne remplit plus son contrat il sera remplacé par un ES équivalent dans le VPSN. Ainsi avec ce concept nous pouvons anticiper les dégradations afin de permettre une continuité de services.

Pendant la mobilité, supposons que nous ayons une dégradation dans un élément de service (par exemple SE1.2), la dégradation est alertée par la communauté de SE1.2 qui va procéder au remplacement de cet élément de service par un autre équivalent dans le VPSN et permettre ainsi la continuité de service. Mais avant de remplacer cet élément, nous avons d'abord le calcul de la QoS du lien (couche transport) entre ce nouvel élément de service à remplacer (par exemple SE1.3) et l'élément de service suivant et précédent dans le VPSN (par exemple SE6.3 et SE7.3).

A partir du tableau de QoS des éléments de service sur le Serveur1, Serveur3 et Serveur2, nous avons la QoS de SE1.2, SE6.3, SE7.3 et SE1.3.

SE	Server 1	SE	Server 2	SE	Server 3
SE1.2	QoS _{SE1.2}	SE1.3	QoS _{SE1.3}	SE1.5	QoS _{SE1.5}
SE2.3	QoS _{SE2.3}	SE6.3	QoS _{SE6.2}	SE7.3	QoS _{SE7.3}
SE7.5	QoS _{SE7.5}	SE6.5	QoS _{SE6.9}	SE6.5	QoS _{SE6.5}

Tableau IV.3-1: Table de QoS de la couche Service : Serveur1, Serveur2 et Serveur3

A partir du tableau de QoS de la couche transport nous avons la QoS de tous les liens (réseaux) possibles entre le Serveur1 et le Serveur2, et entre le Serveur3 et le Serveur2.

Ainsi pour remplacer SE1.2 installé sur le Serveur1 par SE1.3 installé sur le Serveur2, nous avons {QoS_{SE1.2}, QoS_{(1-3)'}, QoS_{(1-3)''}} et après remplacement nous avons {QoS_{SE1.3}, QoS₍₂₋₃₎, QoS_{(2-3)''}} qui permet de maintenir la QoS de bout en bout pendant la transaction en cours.

	Server 2	Server 3	Server 4		Server 1	Server 3	Server 4
Link	QoS ₍₁₋₂₎	QoS ₍₁₋₃₎	QoS ₍₁₋₄₎	Link	QoS ₍₂₋₁₎	QoS ₍₂₋₃₎	QoS ₍₂₋₄₎
Link'	QoS _{(1-2)'}	QoS _{(1-3)'}	QoS _{(1-4)'}	Link'	QoS _{(2-1)'}	QoS _{(2-3)'}	QoS _{(2-4)'}
Link''	QoS _{(1-2)''}	QoS _{(1-3)''}	QoS _{(1-4)''}	Link''	QoS _{(2-1)''}	QoS _{(2-3)''}	QoS _{(2-4)''}
Link'''	QoS _{(1-2)'''}	QoS _{(1-3)'''}	QoS _{(1-4)'''}	Link'''	QoS _{(2-1)'''}	QoS _{(2-3)'''}	QoS _{(2-4)'''}

Tableau IV.3-2: Table de QoS de la couche transport : Serveur1 et Serveur2

IV.4 Gestion dynamique de la QoS sans couture

IV.4.1 Introduction

Dans notre modélisation, nous avons des éléments de service ubiquitaires qui s'autogèrent.

Cette autogestion permet à l'élément de service de surveiller ses propres paramètres de QoS courante tout en vérifiant si son comportement est conforme à celui négocié dans le contrat de QoS préétabli. Dans cette section nous détaillons cette autogestion de QoS dans l'élément de service et son utilité durant la mobilité ou changement de préférences de l'utilisateur pour la continuité de service.

IV.4.2 Contrat de QoS pour un cycle de vie cohérent

Nous avons intégré dans chaque élément de service un Agent QoS. Ce dernier notifie si l'élément de service est toujours en contrat ou s'il est hors contrat. Dans le cas où le ES est hors contrat, l'Agent QoS notifie le service de gestion des communautés (VSC Management) afin de le remplacer dans toutes les sessions courantes desquelles il fait partie (VPSNs Actifs).

Selon notre modèle de QoS, nous avons quatre critères de QoS, ainsi l'Agent de QoS contrôle la valeur courante des quatre critères de QoS et la compare à la valeur seuil de QoS de l'élément de service, dans le cas où au moins trois des quatre critères se dégradent (valeur courante dépasse la valeur seuil), l'Agent QoS envoie un Out Contrat vers les autres éléments de service faisant partie de sa communauté (VSC), ainsi l'ES qui a suffisamment de QoS pour le remplacer va faire partie de tous les VPSN auxquels participe l'ES dégradé. Cette autogestion nous permet ainsi d'assurer la continuité de service et de maintenir le contrat de QoS.

Nous avons vu dans la proposition sur le provisioning de services que l'élément de service est attaché à une file d'attente qui contient les requêtes acceptées selon sa QoS courante. Dans le cas où l'ES se dégrade, il sera remplacé. Mais la QoS de l'ES peut différer de ce qui est prévu en moyenne, par exemple l'ES a pris plus de délai dans le traitement d'une requête, ce qui peut éventuellement causer un retard pour une ou plusieurs requêtes déjà acceptées par lui. Ainsi, prendre en compte la QoS du nœud (élément de service) ne suffit pas à maintenir la QoS des requêtes acceptées par l'élément de service, pour cela nous proposons de contrôler la QoS du lien (file d'attente proposée attachée à l'élément de service) par un QoS Agent qui notifie un gestionnaire de file d'attente ubiquitaire (VQC Management) pour envoyer les requêtes qui risquent de dépasser leur délai. Ceci va anticiper les violations de SLA de chaque requête et assurer une bonne Disponibilité et Fiabilité de l'ES.

La Figure IV.4-1, résume l'ensemble de nos contributions qui prend en compte le contrat de QoS à partir du Déploiement jusqu'au Management. Ainsi, basé sur le modèle NLR (Nœud, Lien, Réseau), de service, de QoS et informationnel (profil de ressource) définis dans la phase de conception, nous avons un mapping du contrat de QoS dès la phase de déploiement en utilisant un profil de déploiement (profil d'usage), ce mapping prend en compte la QoS de bout en bout (service, réseau et équipement). Une fois les ESs déployés,

nous créons leurs VSCs selon leurs fonctionnalités et QoS. Durant la phase de provisioning, nous provisionnons les requêtes selon la QoS statistique de l'ES (sélection des ESs dans le VPSN), puis durant la phase d'usage, nous avons un provisioning dynamique selon la QoS courante de l'ES (ajout des requêtes dans la file d'attente), ce provisioning prend en compte tout changement dans la session de l'utilisateur (mobilité, changement de préférences) dynamiquement par un contrôle de QoS de bout en bout effectué par les VSCs. La phase de Management est ainsi facilitée par l'autogestion de l'ES qui contrôle juste la conformité du contrat de QoS préétabli. Dans le cas de non respect du contrat, il y a le Reprovisioning dynamiquement. Dans le cas où les ESs ubiquitaires déployés et activés ne suffisent plus à satisfaire la demande, il faut activer ou redéployer dynamiquement.

La phase de redéploiement est d'autant facilitée du fait d'avoir un même modèle de service et de QoS de chaque ES ubiquitaire.

Toute cette gestion dynamique de la QoS durant les phases du cycle de vie nous assure ainsi une continuité de service et une E2E QoS pour les fournisseurs.

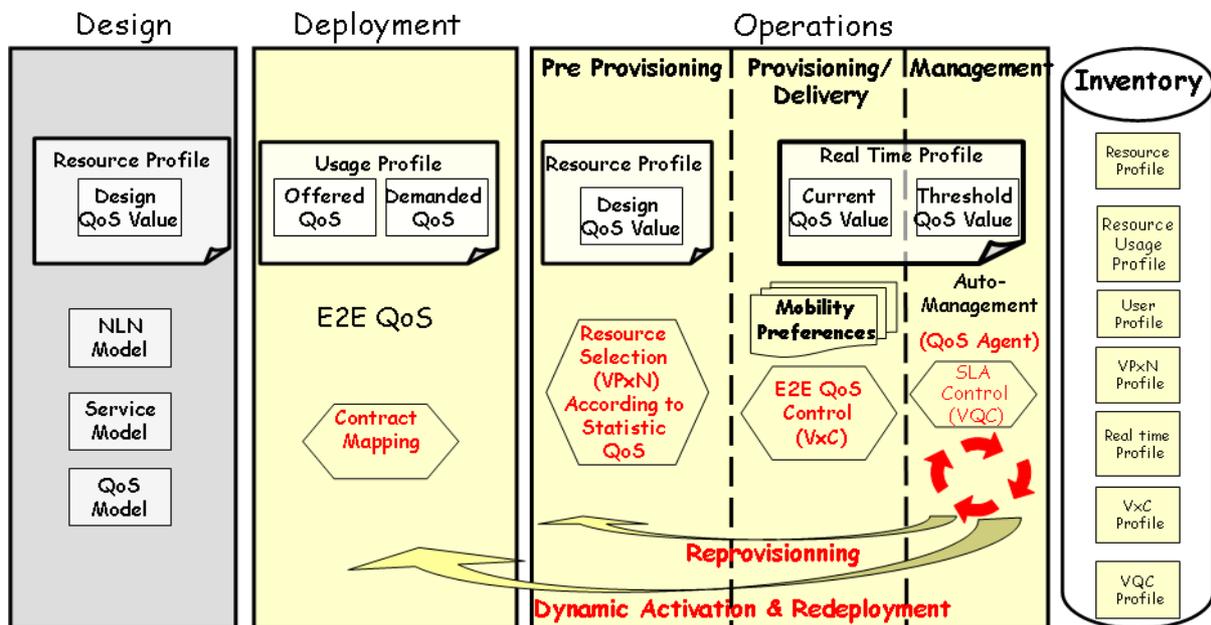


Figure IV.4-1: Contrat de QoS pour un cycle de vie cohérent.

IV.5 Conclusion des propositions

Dans ce chapitre, nous avons consigné nos propositions de repenser les phases du cycle de vie des services dans le nouveau contexte NGN/NGS mobile et ubiquitaire. Un ensemble de « services de gestion » a été préconisé.

Basé sur les travaux de notre groupe de recherche sur la conception d'éléments de services ubiquitaires en prenant en compte non seulement leur aspect fonctionnel mais aussi leur aspect non fonctionnels (QoS), nous avons d'abord proposé de déployer des services ubiquitaires basés sur cet aspect non fonctionnel. Ceci, nous assurera de répondre au contrat du service dès son déploiement, puis d'activer les ESs selon les besoins courant afin d'optimiser l'utilisation des ressources.

Pour maximiser l'utilisation d'un élément de service tout en le mutualisant entre plusieurs requêtes utilisateurs, nous avons proposé une "sélection" de l'ES durant le prè-provisioning (Session VPSN), puis durant l'usage de provisionner en temps réel l'élément de service selon sa QoS courante. Notre proposition prend en compte non seulement le provisioning de la QoS du nœud (SE) mais prend en compte aussi la QoS du lien (proposition d'une file d'attente attaché à chaque ES et contrôlée par un agent QoS), tous ceci nous permet de maintenir la fiabilité, la disponibilité, la capacité et le délai des éléments de services.

Pour maintenir la QoS et assurer une continuité de service durant la mobilité, nous avons proposé un Service Delivery au dessus du Media Delivery qui prend en charge tout changement. Enfin la QoS est contrôlée dynamiquement durant toute la session mobile de l'utilisateur, ainsi nous avons proposé une autogestion dans l'élément de service pour gérer sa QoS et des communautés d'intérêt permettant d'anticiper les violations de SLA et de remplacer les éléments de service Out Contrat.

Le Tableau IV.5-1, résume nos principales propositions en précisant l'existant et le défi à relever dans le nouveau contexte NGN/NGS.

Tableau IV.5-1 : Propositions et défi relevé dans le contexte NGN/NGS.

Phase du cycle de vie	Existant	Verrous/Défis	Propositions
Conception	Aspect Fonctionnel	Composants Ubiquitaires	Aspect Fonctionnel + Aspect non Fonctionnel (QoS)
Déploiement	Basé infrastructure fournisseur	Comment choisir le bon endroit	Basé sur la QoS (QoS offerte, QoS demandée), trace des services ubiquitaires déployés (Création de VSC)
Provisioning	Configuration du service, Provisioning de QoS au niveau réseau	Comment provisionner le niveau service	sélection des ESs dans la session (VPSN), provisioning de l'ES dynamiquement et Mutualisation des ESs
Usage	Assuré seulement par la couche transport (Media Delivery) en prenant en compte le type de service	Continuité de service	Service Delivery pour gérer les changements
Management (Assurance)	Monitoring des paramètres de QoS	Continuité de QoS	Autogestion, VSC/VQC , suivre la Session mobile

Chapitre V Valorisations

Deux champs d'application sont directement impactés par nos propositions et permettent de valoriser nos travaux.

Le premier se déduit de la gestion de la mobilité de la session de l'utilisateur par la QoS. En effet, nous avons traité la QoS de bout en bout en proposant de remplacer automatiquement les composants qui causaient les dégradations. Une gestion de cette session mobile par zones ambiantes est aussi possible. Nous réalisons ainsi un Handover au niveau service qui permet de changer de service ubiquitaire par zone ambiante (§V.1). Ce Handover Sémantique nous permet de garantir la continuité du service durant la mobilité et d'adapter la session des services au nouvel environnement ambiant de l'utilisateur, tout en tenant compte de ses préférences fonctionnelles et non fonctionnelles (QoS).

Le deuxième champ d'application impacté par nos propositions est les standards télécom, et plus particulièrement le Business Process Framework (eTOM) (§V.2). Les modèles définis et nos propositions basées sur les services et la QoS sont directement utilisables pour mieux tenir compte de l'usage de nos utilisateurs nomades d'aujourd'hui.

V.1 Handover Sémantique pour une continuité de service sans couture dans l'environnement ambiant

V.1.1 Introduction

La mobilité est l'un des challenges que nous avons identifié dans le nouveau contexte NGN/NGS. Nous avons vu dans les travaux précédents de notre groupe de travail comment gérer cette mobilité par la QoS. Le concept utilisé est les communautés d'intérêts nommés VSCs, où chaque élément de service appartient à une VSC regroupant des ESs fonctionnellement et QoS équivalents, ainsi dans le cas où la QoS d'un ES se dégrade la gestion des VSCs permet de changer l'ES par un autre fonctionnellement et QoS équivalents. Tout ceci permet de maintenir la QoS offerte par l'élément de service, mais quand est il de la QoS de bout en bout ? Afin d'optimiser le bout en bout, toute ressource ambiante équipement, réseau ou service dans la localisation de l'utilisateur peut être utilisée.

La mobilité engendre un changement de localisation et peut ainsi nécessiter une adaptation, au niveau du réseau d'accès et du réseau cœur, apportée par les solutions de Handover Réseau selon la zone ambiante dans laquelle se trouve l'utilisateur. Ce Handover permettra de maintenir la QoS demandée par l'utilisateur mais ne peut assurer dans la plupart des cas une continuité de service sans couture.

Afin d'assurer la continuité de service pendant la mobilité tout en maintenant la QoS de bout en bout, nous proposons dans cette thèse d'avoir en plus du Handover au niveau réseau d'accès et réseau cœur, un Handover au niveau service. Ce Handover nommé Handover Sémantique va permettre de changer d'éléments de service dans la session de l'utilisateur par zones ambiantes. Le changement de l'élément de service peut être dû au changement de localisation, au changement de terminal ou au changement du contexte ambiant (par exemple les préférences d'un opérateur ou d'un prix de l'ES par rapport au contexte).

Dans ce qui suit nous allons présenter comment fonctionne le Handover Sémantique en l'illustrant d'abord par un scénario.

V.1.2 Scénario

Sur la Figure V.1-2, nous avons un utilisateur qui a initialisé sa session dans sa maison, composé de trois éléments de service (Création de VPSN1) : SE1b =Mail, SE5a=VoD et SE6a=VoIP, sur son Ordinateur. L'accès aux services se fait par un réseau d'accès DSL et un réseau cœur IP.

$VPSN1 = SE1b, SE5a \text{ et } SE6a.$

Cet utilisateur se trouve dans une zone géographique délimitée par un rayon R1 par rapport à sa position géographique (Longitude et Latitude).

$Zone1 = User_Longitude1, User_Latitude1 \text{ et } R1.$

Cet utilisateur est nomade et peut changer de localisation, changer de terminal ou de préférences. Notre but est qu'en plus du Handover qui se fait au niveau du réseau d'accès et du réseau cœur d'effectuer un Handover au niveau service qui se déclencherait suite à certains événements reçus.

Pour les éléments de service, nous avons des éléments de service ubiquitaires déployés dans différentes zones ambiantes (Figure V.1-1). Ces ESs peuvent être de différents opérateurs, leurs prix peuvent différer d'une zone de localisation à une autre et l'autorisation d'accès à ces services peut différer d'un utilisateur à un autre.

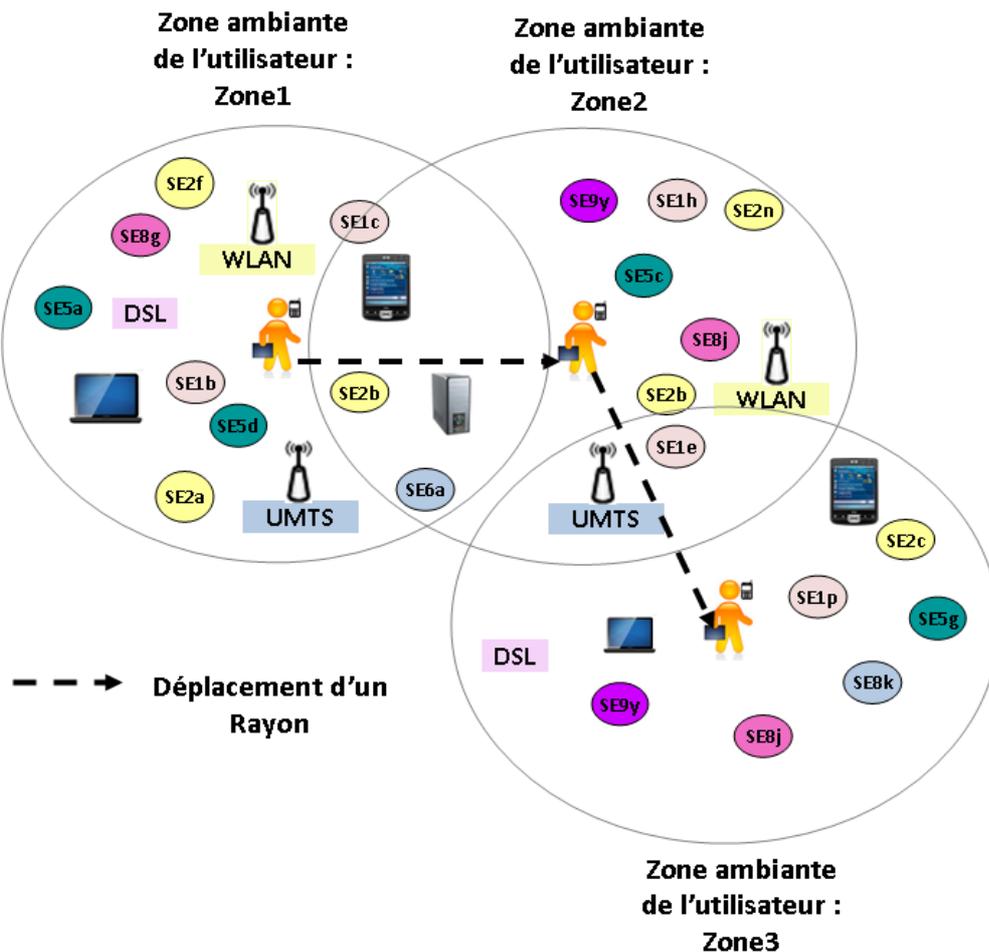


Figure V.1-1: Les ressources dans la zone ambiante.

- **Cas 1 (Mobilité de l'utilisateur):** Cet utilisateur décide de sortir et va basculer sa session sur un autre terminal, par exemple sur son iphone. L'adaptation pour le réseau d'accès et le réseau cœur se fait et la session est accessible ainsi par le réseau d'accès UMTS et le réseau cœur IP.

Pour le Handover au niveau Service, nous allons remplacer l'élément de service SE5a (VoD) par un autre élément de service équivalent sur le iphone et accessible par l'UMTS dans cette zone ambiante (Zone1), ainsi le Service de Gestion du Handover Sémantique (SHS1) qui se trouve sur la même plate-forme (SP1) que SE5a va le remplacer par SE5d.

Le VPSN2 = SE1b, SE5d et SE6a.

- **Cas 2 (Mobilité du terminal) :** L'utilisateur veut rejoindre son bureau et se déplace donc avec son terminal. Ainsi, le Handover au niveau du réseau d'accès et du réseau cœur est déclenché suivant cette mobilité. Le Handover au niveau du réseau d'accès se fait selon la couverture de sa 3G, quand le signal s'affaiblit dans une zone de localisation, le réseau va basculer sur un autre point d'accès.

Pour effectuer un Handover au niveau service, nous considérons les déplacements de l'utilisateur par zone ambiante d'un rayon R1. Donc le Handover Sémantique se déclenche une fois que le terminal de l'utilisateur s'est déplacé d'un rayon R1 ce qui veut dire qu'il a changé de zone.

Notre but est qu'une fois que l'utilisateur rentre dans la nouvelle zone ambiante (déplacement d'un rayon R1 par rapport à sa position), un Handover est effectué au niveau des éléments de service qui ne sont pas localisés dans la nouvelle zone ambiante de l'utilisateur.

Ainsi, pour l'élément de service SE1b qui se trouve sur la plate-forme de service SP4, le SHS qui se trouve dans cette même plate-forme (SHS4) va regarder si cet ES est dans la nouvelle zone ambiante de l'utilisateur, dans ce cas là il ne fait rien, dans le cas contraire il va rechercher des éléments de service ubiquitaires (fonctionnellement et QoS équivalents) dans la nouvelle zone ambiante afin d'effectuer le Handover. Dans notre exemple, le SHS4 va sélectionner SE1e pour remplacer SE1b. Pour SE5d, c'est le SHS1 qui se trouve sur la plate-forme SP1 qui va décider de son Handover et va sélectionner SE5c pour le remplacer dans cette nouvelle zone. Le SHS5 qui se trouve sur la même plate-forme que SE6a va décider de ne pas effectuer de Handover sur lui car il se trouve dans la même zone ambiante que l'utilisateur. Ainsi, la session (VPSN) de l'utilisateur dans la nouvelle zone ambiante est composée des éléments de service *SE1e, SE5c et SE6a*.

VPSN3 = SE1e, SE5c et SE6a.

Zone2 = User_Longitude2, User_Latitude2 et R1.

- **Cas 3 (Préférences de l'utilisateur par rapport au contexte ambiant) :** L'utilisateur va sélectionner et ajouter de nouveaux ESs ou remplacer des ESs par d'autres (découverts dans cette nouvelle zone) à sa session (VPSN) par rapport au contexte ambiant dans lequel il se trouve, dans l'exemple nous avons un nouvel élément de service (SE2c) découvert dans la zone ambiante Zone3, ajouté au VPSN de l'utilisateur et le remplacement (Handover Sémantique) de l'élément de service SE1e par SE1g pris chez un autre opérateur qui permet de ne pas payer plus chers (Préférence de l'utilisateur prise en compte) dans cette zone de localisation (Zone3). Après le Handover effectué par SHS2 pour remplacer SE1e par SE1g, le VPSN4 de l'utilisateur est composé de SE5c, SE6a, SE2c et SE1g.

VPSN4 = SE5c, SE6a, SE2c et SE1g.

$Zone3 = User_Longitud3, User_Latitude3 \text{ et } R1.$

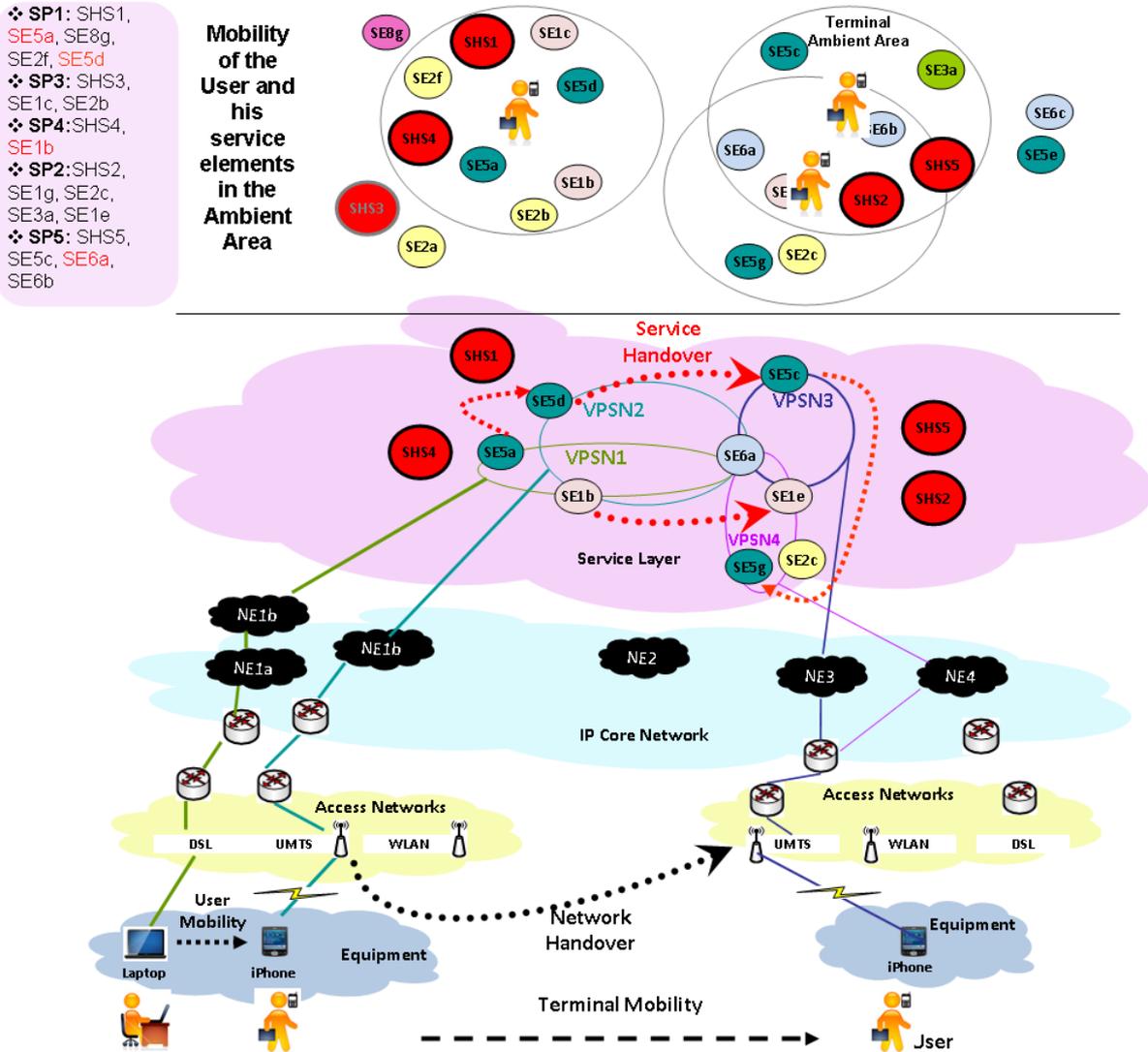


Figure V.1-2 : Scénario de Handover au niveau service.

Nous définissons ce concept de Handover Sémantique et détaillons son fonctionnement dans la section suivante.

V.1.3 Concept du Handover Sémantique

Par analogie aux Handover dans le réseau d'accès et le réseau cœur, nous proposons un Handover au niveau service. Nous avons vu dans l'état de l'art les différentes solutions existantes de Handover au niveau réseau et nous avons constaté que nous avons trois rôles principaux, dans le processus de Handover réseau, qui sont l'Initiateur, le Décideur et l'Exécuteur du Handover. Chaque rôle peut être joué par un même ou différent acteur (BSC, MSC,...).

Pour le Handover au niveau service nous avons défini trois rôles :

L'Initiateur du Handover : représente l'entité qui détecte un changement et qui initie le Handover en envoyant une notification d'initiation du Handover vers le décideur.

Le Décideur du Handover : représente l'entité qui reçoit l'événement de Handover et qui décide d'effectuer un Handover sémantique ou pas en se basant sur les informations récupérées en interagissant avec d'autres entités de l'architecture.

L'Exécuteur du Handover : représente l'entité qui exécute la décision prise par le décideur.

Pour chacun des trois rôles nous définissons des acteurs, ainsi nous avons :

Pour le rôle d'initiateur, nous avons le Terminal Userware [SR07], ou la base informationnelle Infoware qui peut détecter un changement ou un événement nécessitant d'effectuer un Handover.

Pour le rôle de décideur, nous proposons un service de gestion du Handover Sémantique dénommé : Semantic Handover Service (SHS). Ce Service déclenche le processus du Handover suite à une mobilité du terminal, à une mobilité de l'utilisateur ou à un contexte ambiant. Nous proposons de déployer un SHS dans chaque plate-forme de service afin qu'il prend en charge les ESs qui sont déployés dans sa plate-forme.

Pour le rôle d'Exécuteur, nous avons le service de gestion du VPSN, dénommé le VPSN Maintenance qui s'occupe de maintenir la session de l'utilisateur qui va exécuter la décision prise par le SHS.

V.1.4 **Initiateur du Handover**

L'initiation du Handover peut être due aux événements suivants : Mobilité du Terminal, Mobilité de l'utilisateur ou Préférences de l'utilisateur.

(a) **Initiation due à une mobilité du terminal**

Durant la mobilité, l'utilisateur se déplace et les informations sur sa position géographique sont envoyées à partir de son terminal vers la base informationnelle qui enregistre ces informations (Figure V.1-4). Quand le réseau détecte ces changements de localisation et qu'il constate un affaiblissement du signal de la communication du terminal il initie le Handover au niveau réseau. Dans le réseau, l'initiateur du Handover peut être le même acteur que celui qui va décider d'effectuer le Handover ou un acteur différent.

L'initiation du Handover au niveau service se fait selon le déplacement par zone délimitée par la localisation géographique de l'utilisateur (sa longitude et sa latitude) et un rayon R_x prédéfini par l'opérateur.

Les informations sur la localisation géographique de l'utilisateur (User_Longitude, User_Latitude) sont enregistrées dans sa base informationnelle INFOSPHERE [NS08], si ces deux paramètres changent d'un Rayon R_x , un déclencheur réagit dans l'Infosphere qui envoie une notification d'un changement de zone vers l'INFOWARE : Notify Area Change.

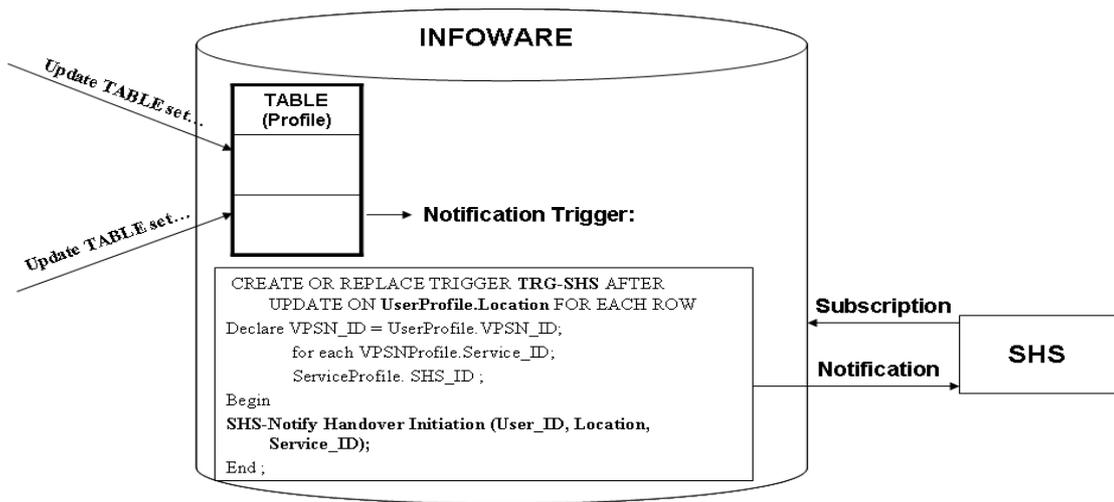


Figure V.1-3: Déclencheur dans l'Infoware.

Notre base informationnelle INFOWARE est événementielle et inférente. Tout changement dans cette base peut déclencher des changements ou des notifications vers d'autres acteurs. La mise à jour (modification) de la localisation de l'utilisateur dans son profil (User Profile) va induire un « Déclencheur » (Figure V.1-3), dans lequel pour chaque ligne du profil VPSN (chaque Service_ID), il va envoyer une notification vers son SHS_ID (Pris du Service Profile de ce Service_ID) afin d'initier le Handover Sémantique : Notify Mobility Handover (User_ID, Location, Service_ID).

Ainsi le Handover est initié dans chaque SHS qui a la charge d'un ou plusieurs Services faisant partie de la session (VPSN) de l'utilisateur.

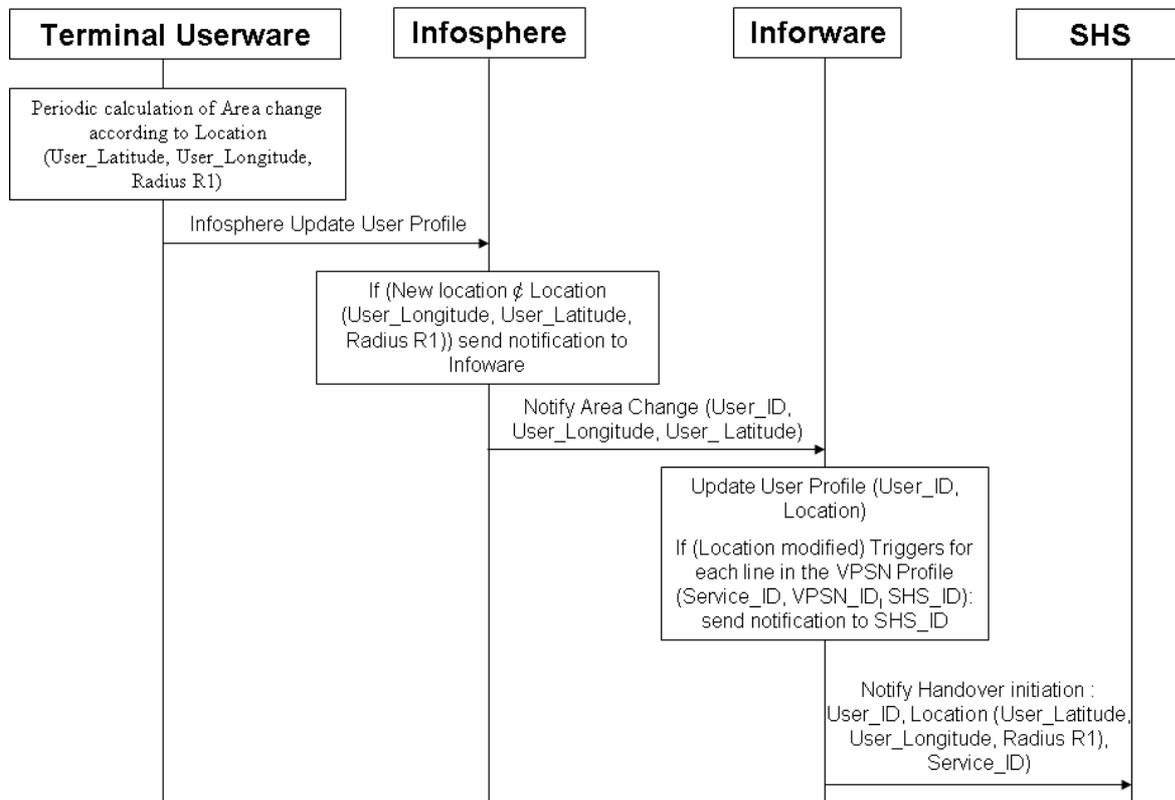


Figure V.1-4 : Initiation du handover sémantique

(b) **Initiation due à une mobilité de l'utilisateur**

La mobilité de l'utilisateur signifie qu'il change de terminal. Afin de maintenir la session de l'utilisateur durant ce changement, nous avons un basculement de l'attachement du VPSN de l'ancien terminal vers le nouveau terminal, mais les éléments de service faisant partie de ce VPSN peuvent ne pas être adaptés au nouveau terminal ou aux préférences prédéfinies de l'utilisateur, pour cela le Handover Sémantique peut être nécessaire pour remplacer des éléments de service par d'autres adaptés au contexte ambiant de l'utilisateur.

Ainsi l'événement qui montre une mobilité de l'utilisateur est l'identification de l'utilisateur sur un nouveau terminal et sa requête d'attachement sur sa session active (son VPSN). Ces informations sont mises à jour sur l'Infosphere (UserProfile) et notifiées vers l'Infoware. Ainsi, si nous avons une mise à jour d'une identification de l'utilisateur sur un nouveau terminal dans l'Infoware, cette dernière par inférence va mettre à jour d'autres informations appropriées selon l'événement reçu et peut déclencher des « déclencheurs » de Handover Sémantique quand c'est nécessaire.

(c) **Initiation due à un changement de préférences selon le contexte ambiant**

L'utilisateur peut avoir prédéfini des préférences d'utilisation des services selon le contexte ambiant dans lequel il se trouve, par exemple, il préfère utiliser un service fourni par un tel opérateur à tel localisation géographique ou dans tel contexte ambiant. Dans ce cas là, le Handover Sémantique est initié quand un tel événement se déclenche pour remplacer l'élément de service concerné par celui qui est approprié.

V.1.5 **Décideur du Handover**

Dans le Handover réseau, le rôle de décideur peut être joué par différents acteurs. Par exemple dans le GSM le rôle de décideur peut être joué soit par le BSC (cas de Handover intra-BSC) ou par le MSC (cas de Handover inter-BSC). Dans notre concept de Handover Sémantique, le rôle de décideur est joué par un seul acteur qui est le SHS dans chaque plateforme de services.

Quand le SHS reçoit une notification d'une mobilité du terminal et une liste de services dont il a la charge, il déclenche son processus de préparation de Handover que nous détaillons dans les étapes suivantes :

- (a) Le SHS reçoit une notification pour initier le Handover concernant les services dont il a la charge : Notify Handover Initiation (User_ID, User_Longitude, User_Latitude, List of Service_IDs).
- (b) Le SHS va rechercher dans la zone ambiante de l'utilisateur qui est délimitée par sa Latitude et sa Longitude et un Rayon R1 prédéfini, si chaque Service_ID reçu se trouve dans cette zone. Si c'est le cas alors il ne fait rien (pas de Handover déclenché). Dans le cas contraire il va déclencher le Handover sémantique : search if Geo-Location of Service_IDs € Location (User_Longitude, User_Latitude, R1).
- (c) Le SHS va interagir avec l'INFOWARE afin de lui demander la fonctionnalité et la QoS pour chaque Service_ID qui n'est pas dans la zone ambiante (Service Profile).
- (d) Le SHS va interagir avec le service de gestion de découverte DBS afin de rechercher les services ubiquitaires qui ont la même fonctionnalité et la même QoS pour chaque Service_ID dans la zone fournie (User_Latitude, User_Longitude et R1).
- (e) Le service de gestion DBS va fournir la liste des ESs ubiquitaires.

- (f) Le SHS va interagir avec le service de gestion de Présence PBS afin de connaître l'état³ (Disponible, Indisponible, Activable ou Activé) de chaque ES fourni par le DBS.
- (g) Le SHS va filtrer les services en excluant ceux qui sont indisponibles, puis va choisir pour chaque Service_ID un service ubiquitaire qui peut le remplacer. Le choix peut se faire selon le premier de la liste, celui qui se rapproche le plus dans son intervalle de QoS, celui appartenant à un certain fournisseur ou selon un autre critère prédéfini selon la politique de l'opérateur.
- (h) Une fois le SHS a décidé des services (New Service_IDs) qui vont remplacer ceux pour lesquels il a déclenché le Handover (Old Service_IDs), il va envoyer les deux catégories de services vers l'Exécuteur du Handover : Notify VPSN Maintenance (User_ID, List of Old Service_ID, List of New Service_ID).

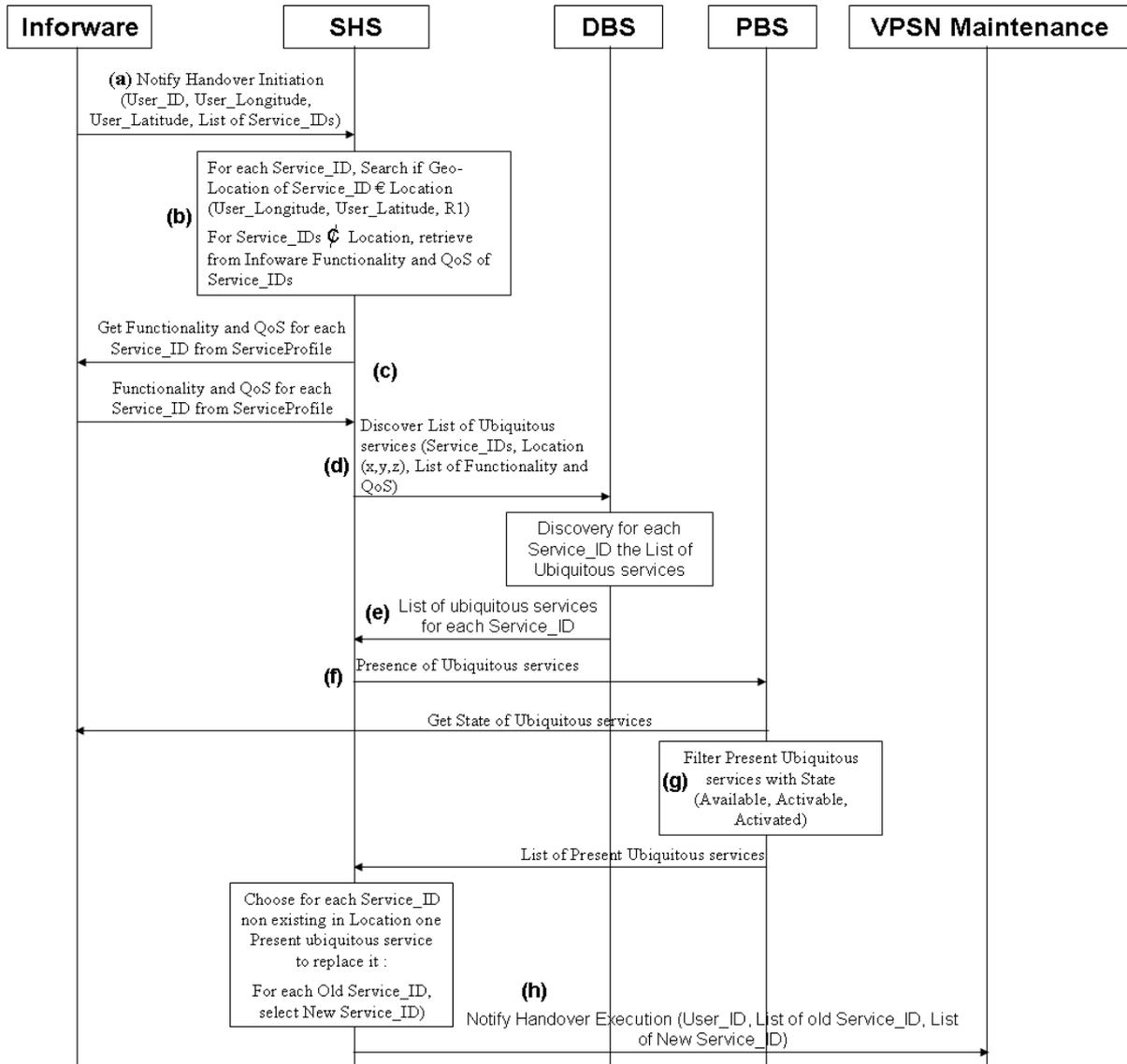


Figure V.1-5 : Décision du handover sémantique

³ L'état de SE est défini dans le plan de gestion du SE

V.1.6 Exécuteur du Handover

L'Exécuteur du Handover peut être le même acteur qui décide du Handover ou un acteur différent. La session de l'utilisateur constitue son VPSN et elle est maintenue par un service de gestion qui est le "VPSN Maintenance", ainsi ce dernier constitue notre Exécuteur de Handover. Le rôle du "VPSN Maintenance" est de maintenir tous les services choisis par l'utilisateur pour faire partie de sa session, et d'adapter ces services dans le cas d'une dégradation ou de tout autre changement (mobilité, préférences,...). Le "VPSN Maintenance" reçoit une notification de la part du SHS afin d'exécuter le Handover Sémantique.

Le "VPSN Maintenance" va attacher chaque nouveau service sélectionné par le SHS. L'attachement est effectué en ajoutant les Nouveaux Service_ID dans le VPSN Profile dans l'INFOWARE. Une fois l'attachement effectué, puis le "VPSN Maintenance" va détacher chaque service remplacé dans le VPSN Profile. Notre base informationnelle n'étant pas centralisée, une mise à jour par inférence est réalisée entre les différents fragments d'INFOWARE sur les autres plates-formes.

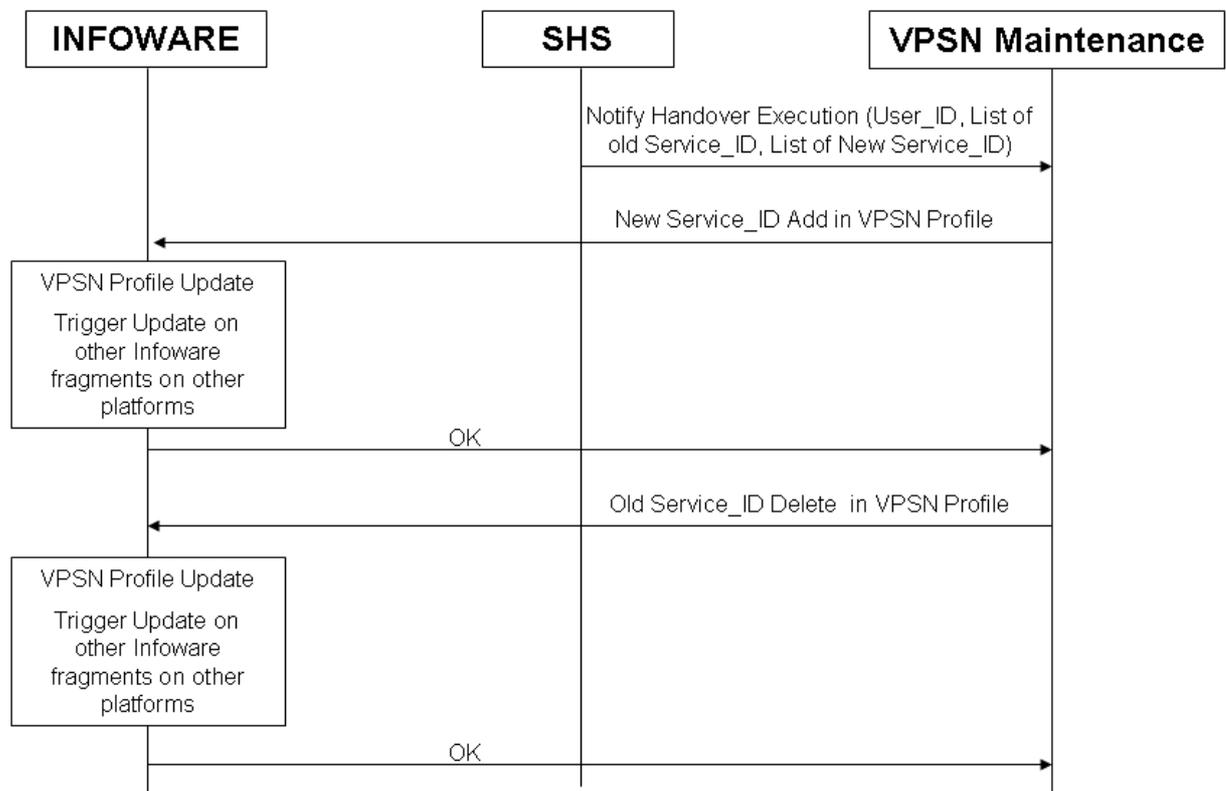


Figure V.1-6 : Exécution du handover sémantique.

V.1.7 Conclusion

En plus de la gestion de la mobilité par la QoS proposée dans notre groupe de recherche AIRS, dans cette partie nous avons montré comment gérer la mobilité par zones ambiantes. Pour cela nous avons proposé et spécifié un *handover sémantique* au niveau service qui permet de changer d'élément de service quand l'utilisateur s'est déplacé d'un rayon donné. Par analogie aux concepts de handover au niveau du réseau d'accès et du réseau cœur, le handover sémantique proposé est basé sur trois rôles, l'initiateur, le décideur et l'exécuteur.

L'initiateur, permet de détecter le déplacement par un rayon donné et d'initier donc le handover sémantique. Le décideur (SHS) permet de sélectionner les éléments de service dans la zone délimitée pour le handover de service. L'exécuteur, permet d'exécuter la décision (choix des ESs déployés dans la zone ambiante) prise par le décideur. En utilisant ce concept de handover sémantique nous pouvons garantir la continuité de services lors des mobilités et d'adapter la session de services (VPSN) au nouveau contexte ambiant de l'utilisateur, tout en tenant compte de ses préférences fonctionnelles et non-fonctionnelles (critères de QoS).

V.2 Prise en compte de la QoS dans le Business Process Framework

V.2.1 Processus impliqués dans le Déploiement de Services dans eTOM

Le déploiement fait partie du cycle de vie des services, ainsi les processus que nous avons identifiés concernant le déploiement sont dans le processus vertical de bout en bout du cycle de vie des produits. Dans le niveau service, nous avons le « *Service Development and Retirement* » [GB921b], qui regroupe sept processus de niveau trois de l'eTOM (Figure V.2-1). Parmi ces processus nous avons ceux de gestion du développement, du déploiement et du retrait des services.

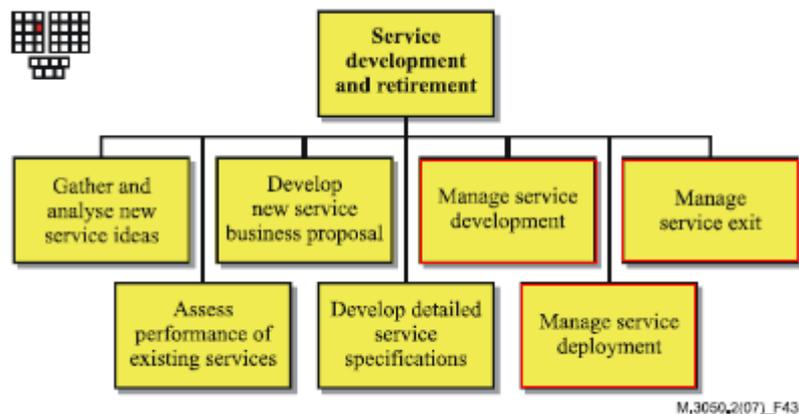


Figure V.2-1: Processus Développement et Retrait des services.

Déployer un service n'implique pas toujours son développement et le développement d'un service n'implique pas son déploiement immédiat ainsi ces deux processus sont indépendants et ne sont pas toujours sollicités en duo. Dans notre analyse nous avons dissociés les processus de développement et ceux de déploiement car nous pensons que les séparer nous est utile dans ce nouveau contexte trans-organisationnel où les opérateurs peuvent déployer des services qu'ils n'ont pas développés eux même et qu'ils ont acheté chez différents fournisseurs de service. Cette dissociation entre les processus nous permet ainsi de spécifier le processus de déploiement de bout en bout en identifiant les blocs qui sont invoqués dans les différentes couches (service exposable, service, ressource et chaîne d'approvisionnement) ainsi que les informations utilisés par ces processus (composition/décomposition, profil d'usage, localisation, statut, etc.).

Pour cela, nous avons décomposé le processus *Service Development & Retirement* en trois processus indépendants (Figure V.2-2) : *Service Development*, *Service Deployment* & *Service Retirement*.

Cette dissociation nous donne le découpage suivant :

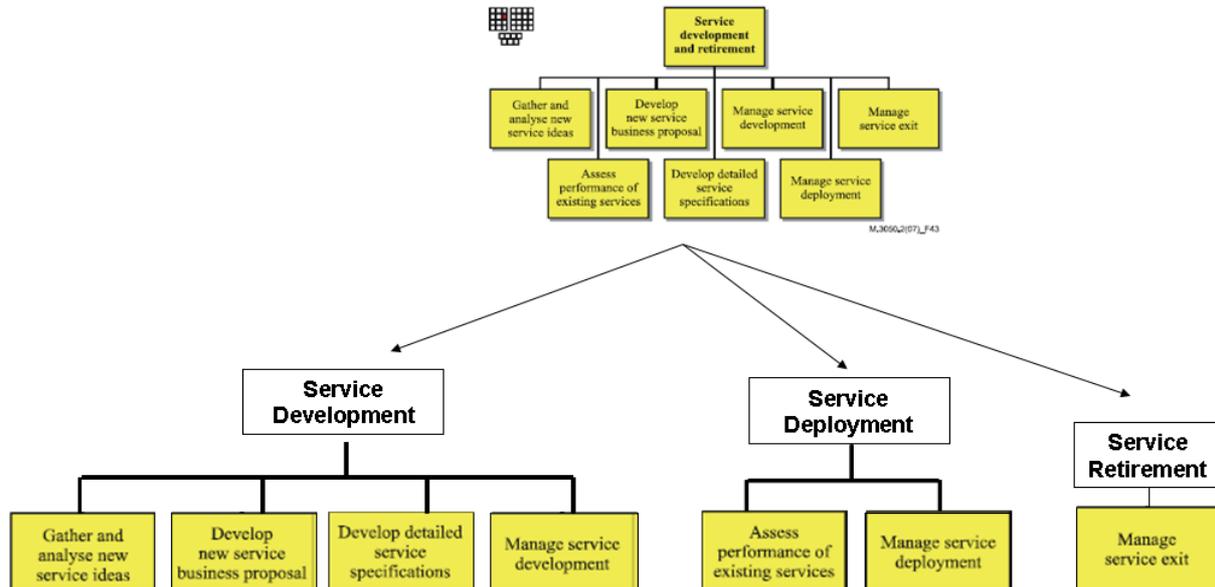


Figure V.2-2 : Décomposition du processus « Développement et Retrait des services ».

Ainsi les processus proposés pour le déploiement (Figure V.2-3) sont : Manage Exposed Service Deployment, Manage Service Deployment, Manage Resource Deployment, Test QoS Resource et Manage Community.

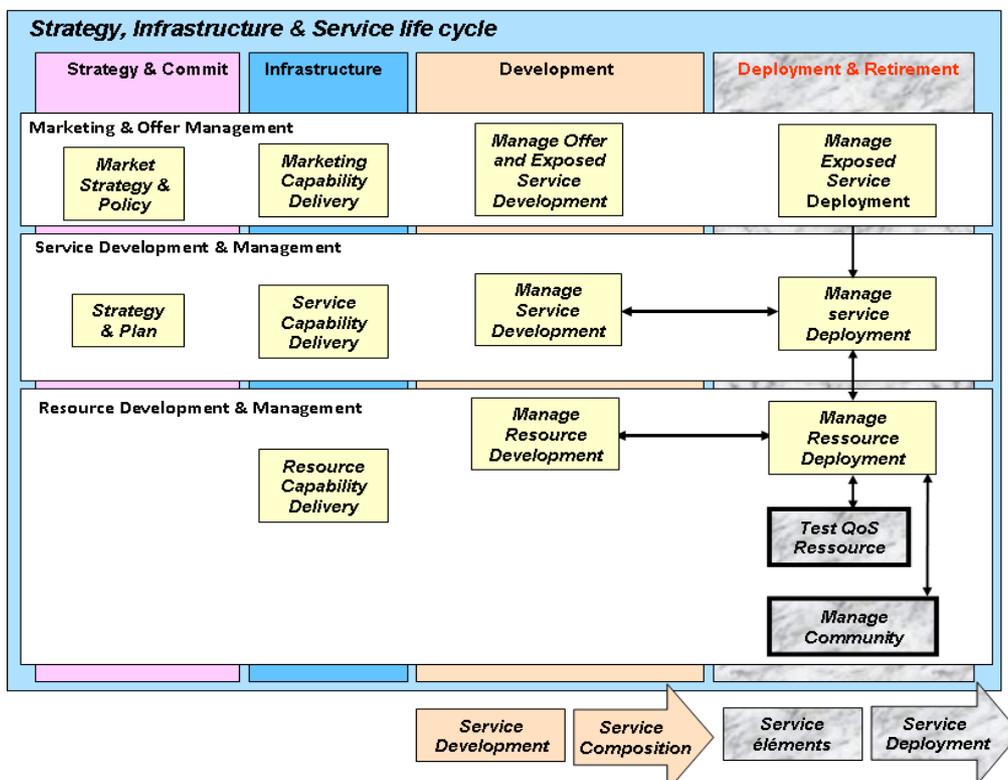


Figure V.2-3 : Phase de déploiement proposée dans eTOM.

V.2.2 Processus de déploiement proposé

Dans cette partie nous détaillons le processus de déploiement en se basant sur les processus de l'eTOM, mais en y intégrant le modèle de QoS, la création des communautés des services une fois ils sont déployés ainsi que le profil de déploiement de services qui représente le profil d'usage de ressource. Les étapes de déploiement sont (Figure V.2-4) :

1. Une demande de déploiement arrive de :
 - a. Processus de *Strategie & Plan* de l'eTOM qui demande de déployer un nouveau service dans plusieurs zones géographiques nécessitant ce service (Service Ubiquitaire).
 - b. Processus de *Market Strategy and Policy* de l'eTOM qui a pour rôle d'analyser les plaques cibles des services ubiquitaires à déployer ainsi que leurs nombres de prospect.
2. La demande est reçue par le processus de niveau 2 de l'eTOM : *Product Deployment & Retirement* et plus spécifiquement par son processus de niveau 3 : *Manage Product Deployment*
3. Le processus *Manage Product Deployment* va l'envoyer à son tour vers le processus de déploiement de service : *Service Deployment & Retirement* tout en ajoutant les informations concernant les plaques cibles de déploiement.
4. Le processus *Manage Service Deployment* va demander au processus *Service Development* la composition de ce service et le Profil d'Usage de chaque composant ainsi que le profil d'usage du service global.
5. Le processus *Service Development* va envoyer les Composants de Services constituant ce service ainsi que le profil d'usage de ce service (QoS demandée par ce service).
6. Le processus *Manage Service Deployment* va envoyer une recherche d'existence de chaque élément de service dans le répertoire de service vers le processus de niveau 2 du domaine *Opérations* de l'eTOM : *SM&O Support and Readiness* et plus spécifiquement à son processus de niveau 3 : *Manage Service Inventory* qui gère la base de données des éléments de services déployés, leur configuration, leur état (Disponible, Indisponible, Activable ou Activé) ainsi que le lien avec les ressources associés à ces ESs. Ce processus va demander aussi le profil d'Usage des éléments existants dans la base de données.
7. Le processus *Manage Service Inventory* va envoyer les *composants de service existants* dans la base de données avec leurs *QoS Offerte* qui va servir à connaître la QoS globale offerte par le service global déployé.
8. Pour les éléments de services non existants dans la base de données, le processus *Manage Service Deployment* va envoyer la *QoS demandée* de chaque élément vers le processus *Manage Ressource Deployment* qui est le cœur du déploiement de services car c'est ce processus qui va affecter les ressources appropriées à chaque élément à déployer, il va lui envoyer aussi la QoS offerte par les ESs existants pour voir au niveau des ressources si on a ce qu'il faut (QoS courante).
9. Le processus *Manage Ressource Deployment* va envoyer au processus de niveau 1 du domaine *Operations* : *RM&O Support and Readiness* et plus spécifiquement son processus de niveau 2 : *Manage Ressource Inventory*, la demande des Ressources disponibles des quatre types de ressources dont a besoin l'élément de service à déployer (*Ressource équipement, ressource réseau, ressource service et ressource logiciels*) satisfaisant la QoS demandée (QoS demandée = QoS offerte).
10. Le processus *Manage Ressource Inventory* va répondre par les *Ressources Disponibles* et la *QoS Offerte Agrégée* de ces quatre types de ressources.
11. Les ressources disponibles et leur QoS offerte sont reçus par le processus *Manage Ressource Deployment*, ce dernier va envoyer une demande de test de QoS vers le processus *Test Ressource QoS*.

12. Une fois que le Test de QoS est effectué avec succès le processus *Manage Ressource Deployment* va envoyer une demande de *création de communauté* pour la *Ressource de service* en envoyant la *QoS offerte* reçue du processus *Manage Ressource Inventory* ainsi que la *fonctionnalité* de cette ressource de service vers le processus *Service Community Management*.

13. Le processus *Service Community Management* va créer la communauté de ce composant de service et envoyer la notification de création de communauté vers le processus *Manage Ressource Deployment* qui va envoyer les informations de mise à jour du répertoire des ressources puis envoyer la notification de déploiement des ESs vers le processus *Manage Service Deployment*.

14. Une fois que tous les éléments de service, qui composent le service global, sont déployés le processus *Manage Service Deployment* va envoyer une demande de Test de QoS globale du service vers le processus *Test QoS Service (Test QoS (SD= CS1+CS2+CS3))*.

15. Si le test de QoS est effectué avec succès, le processus *Manage Service Deployment* va transférer les informations de configuration au domaine Operations (processus *Enable service configuration and activation*) et mettre à jour le répertoire des ESs, leur état et leur profil d'usage (*processus Manage Service Inventory*).

16. Le processus *Manage Service Deployment* va envoyer au processus *Manage Exposed Service Deployment* que le service est déployé, ce dernier va envoyer les informations de tarification, promotions au domaine opérations et mettre à jour le répertoire des service exposables.

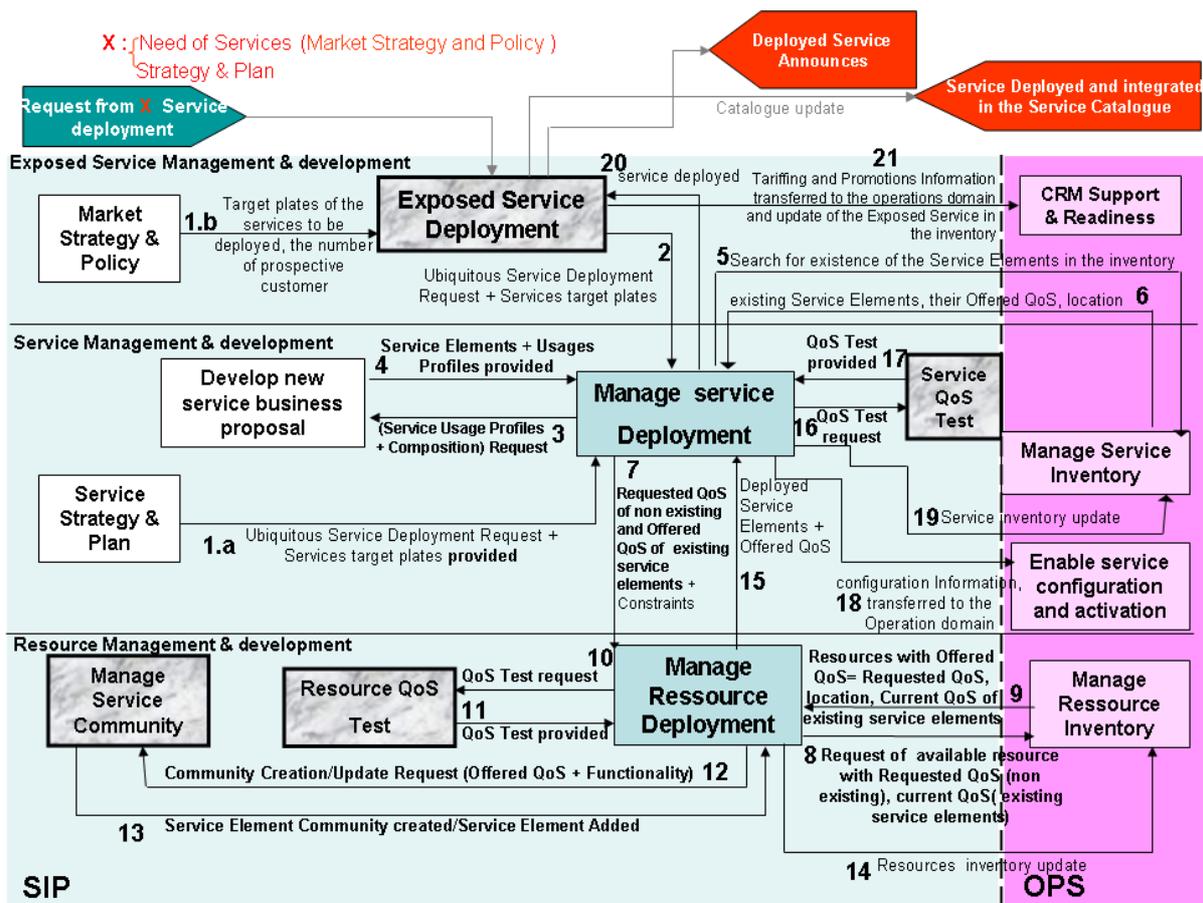


Figure V.2-4 : Processus de déploiement.

V.2.3 Processus proposés dans l'exploitation pour prendre en compte le contexte NGN/NGS

Avant d'intégrer nos modèles et services de gestion proposés pour prendre en compte notre contexte NGN/NGS où la plate-forme de service est provisionnée avant même le choix du réseau et indépendamment de ce dernier, nous avons d'abord commencé par analyser les processus de provisioning de services dans eTOM. Ce qui nous a amené à la conclusion que cet approvisionnement (lors de la commande) est une combinaison de configuration et d'activation de service qui se fait dans la couche service et d'un provisioning réseau qui se fait dans la couche ressource, ainsi nous avons vu que les ressources services ne sont pas prises en compte.

Sur la Figure V.2-6 suivante nous montrons les différentes couches de l'eTOM sollicitées pendant le processus d'approvisionnement de service lors de la commande d'un client, ainsi nous avons :

- dans la couche Customer : la Commande du client (Customer Order).
- dans la couche Product : les produits offerts au client sur le catalogue de l'opérateur / Fournisseur de service.
- dans la couche Service : le(s) service(s) qui constituent un produit, où nous avons les paramétrages des services par rapport à *une demande client (catalogue technique)*.
- dans la couche Ressource : *nous proposons une structuration des ressources conformément à notre modélisation (niveaux de visibilité). ainsi nous avons les ressources services, réseaux et équipements.*

D'un autre coté, nous avons remarqué que le processus de provisioning dans eTOM (1) est statique et ne prend pas en compte la mobilité et les préférences de l'utilisateur. Pour prendre en compte la mobilité et la gestion des préférences de l'utilisateur selon sa localisation (2), nous proposons d'intégrer notre processus de *gestion dynamique pendant l'exploitation de service*, ce processus que nous avons nommé « Delivery » permet de gérer toute la session mobile de l'utilisateur.

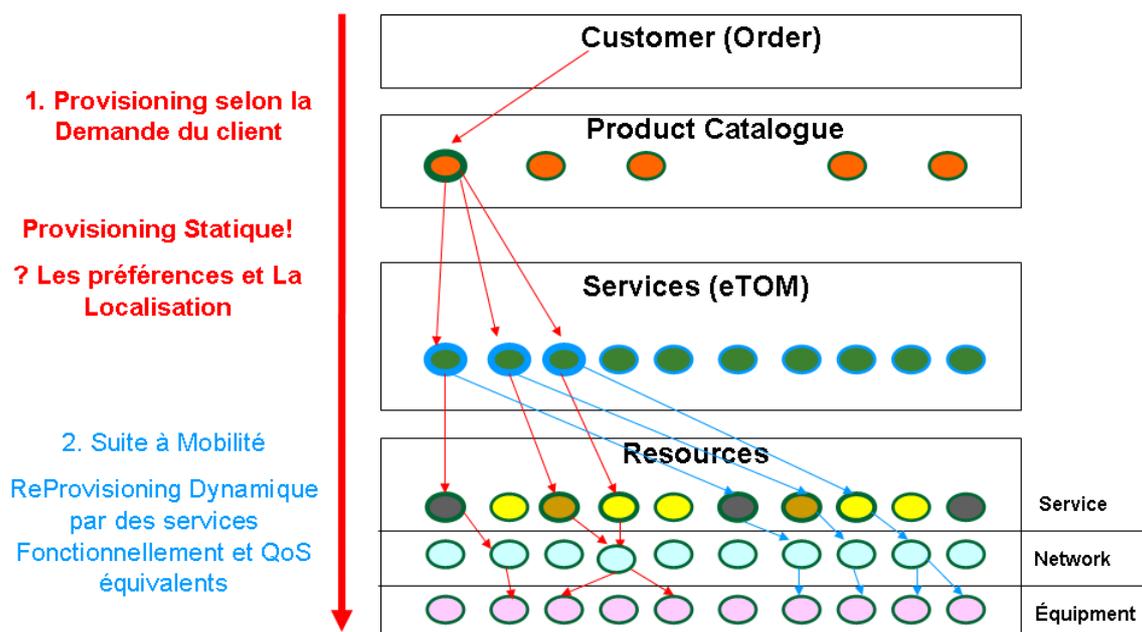
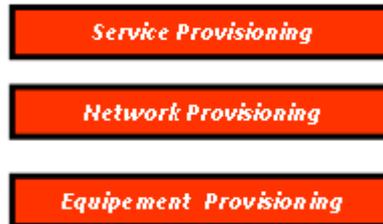


Figure V.2-5: Provisioning du service demandé par le client qu'elle que soit ses préférences et sa localisation

Ainsi nous proposons des processus qui prennent en compte la ressource « service » dans la couche ressource d'eTOM. Le processus d'approvisionnement est donc une combinaison « de configuration et d'activation de service » qui se fait dans la couche Service, d'un « service resource provisioning » qui se fait dans la couche Ressource avec un « contrôle d'admission des requêtes » pouvant utiliser ce service pour gérer la mutualisation de ce dernier et d'un « network resource provisioning » qui se fait dans la couche Ressource (provisioning des ressources réseaux requises pour ce service).

Resource Provisioning:

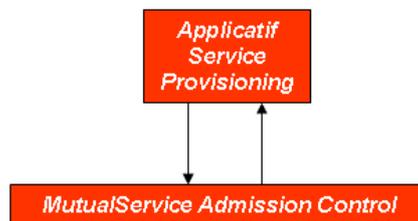


Service Provisioning (VPSN) : s'occupe du Provisioning des services Applicatifs.

Network Provisioning (VPCN) : s'occupe de provisionner les ressources réseaux demandées par le service applicatif comme par exemple la réservation d'un VC (Virtual Circuit) dans le cas d'ATM.

Equipment Provisioning (VPEN) : s'occupe du Provisioning des équipements (Terminaux, Plate-forme de service, Routeurs, Edges, ...).

Mutual Service Admission Control :



Permet de gérer le nombre de requêtes acceptées pour l'utilisation partagée de l'élément de service pendant la session, invoqué par le processus de Provisioning des Services Applicatifs.

Service Composition Management:

Ce processus s'occupe de la gestion des éléments de service faisant partie du VPSN de l'utilisateur où chaque service exposable demandé par l'utilisateur est décomposé en éléments de service de base et sélectionnés dans le VPSN, les informations sur ces services exposables et leurs compositions sont stockées dans le « Exposed Services Inventory Management ».

Exposed Services Inventory Management

Cette base de données contient tous les services exposables ainsi que leurs compositions (les éléments de service), leurs QoS, etc.

VSC Management:

Le processus Virtual Service Community maintient les communautés fonctionnellement et QoS équivalents. Il s'occupe de remplacer un élément dégradé dans le VPSN par un autre équivalent.

VNC Management:

Le processus Virtual Network Community maintient les communautés des connectivités fonctionnellement et QoS équivalentes.

VEC Management:

Le processus Virtual Equipment Management maintient les communautés des équipements fonctionnellement et QoS équivalent.

La Figure V.2-6, montre l'intégration de ces nouveaux processus dans la cartographie d'eTOM (partie Opérations).

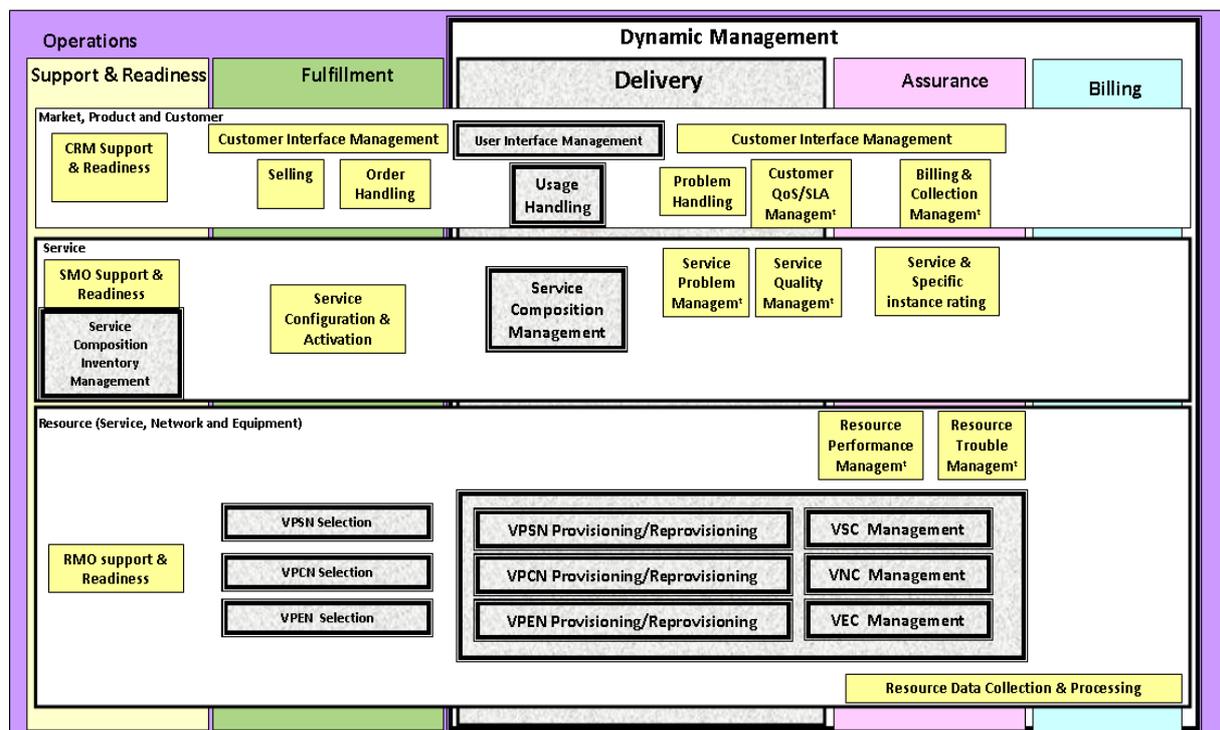


Figure V.2-6: Intégration de la QoS pour la prise en compte de la mobilité dans eTOM

V.2.4 Processus de Provisioning proposé

La partie commande est la partie où l'on alloue, provisionne et active les ressources déjà réservées, donc dans le flux de la Figure V.2-7, nous avons d'abord une allocation de toutes les ressources (Service, Réseau, Equipment) puis une fois les ressources allouées nous avons un provisioning de ces ressources et à la fin l'activation de ces ressources. Une fois que les ressources sont activées, le processus Service Configuration & Activation va envoyer les informations sur le service vers les processus d'Assurance et de Billing et va notifier à la couche « Market, Product & Customer » que le service est activé donc qu'il est prêt à être utilisé.

Le flux du processus de provisioning est détaillé sur la Figure V.2-7.

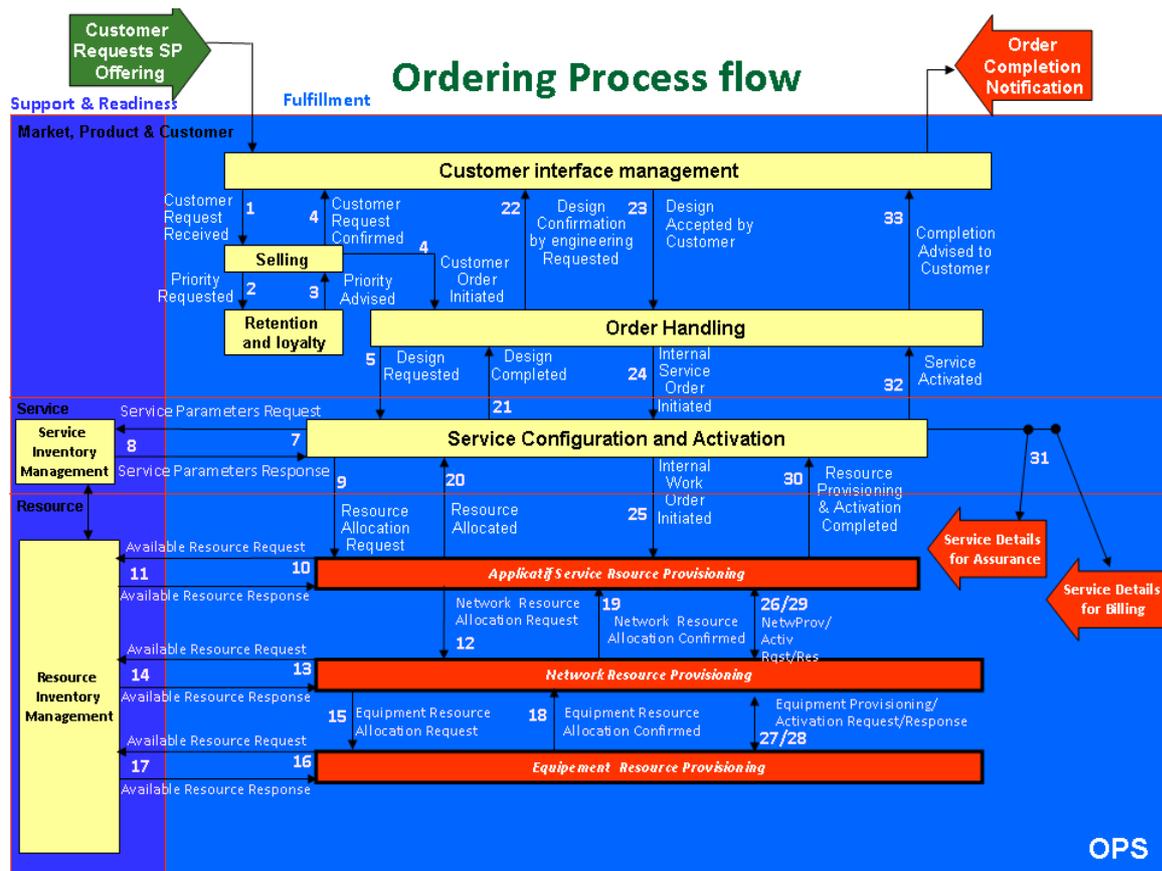


Figure V.2-7: Flux du process Commande.

V.2.5 Interaction entre les phases de Fulfilment, de Delivery proposé et d'Assurance

Pendant l'exploitation du service nous avons une surveillance de bout en bout de toutes les ressources utilisées pour fournir le service à l'utilisateur.

Le processus vertical Assurance dans eTOM s'occupe de collecter des données sur la performance à partir des ressources afin de les analyser, puis pour déterminer la qualité de service globale et enfin pour vérifier le respect de l'accord SLA avec le client.

Ce processus d'Assurance interagit avec le processus vertical de Billing pendant l'exploitation du service, par exemple, lorsqu'il faut résoudre en temps réel une transgression d'accord SLA. Dans ce cas, le client notifie un problème qui est résolu mais qui entraîne une transgression d'accord SLA avec le client de sorte qu'un rabais est offert au niveau de la facturation.

Pour nos éléments de service qui sont ubiquitaires, lorsqu'un ES ne rend pas la QoS demandée pendant l'utilisation nous le remplaçons dans la session (VPSN) par un autre ES fonctionnellement et QoS équivalents qui appartient à la VSC. Ainsi pendant le processus d'Assurance nous avons un contrôle dans les processus VSC, VNC et VEC de toutes les ressources fonctionnellement et QoS équivalentes. Dans le cas d'une dégradation de l'une des ressources ou de la QoS de bout en bout, ces processus vont remplacer la ressource dégradée par une autre dans le VPSN, VPCN ou VPEN, la ressource responsable de la dégradation va intégrer une autre communauté qui correspond à sa QoS ou une nouvelle communauté dans le cas où elle n'existe pas. Dans le cas où nous n'avons pas d'ES disponible pour celui qui est

dégradé, le processus qui gère les compositions de service (Service Composition Management) va envoyer une demande de provisioning vers le processus Service Configuration & Activation qui va Réserver, Allouer, Provisionner et Activer cet ES, tout en mettant à jour les bases de données des ressources et des services.

Dans eTOM nous avons aussi des processus qui récoltent les données de performance sur les ressources et envoient une notification vers les processus de la couche service pour réapprovisionner les ressources dégradées.

Un exemple du flux du processus d'Assurance est présenté sur la Figure V.2-8 :

Étape 1. Les processus de gestion des communautés d'équipements (VEC), de connectivité (VNC) et de service (VSC) surveillent le comportement des ressources (ce comportement se traduit par la QoS). Dans le cas d'une dégradation ces processus remplacent les ressources dégradées dans les VPSN, VPCN ou VPEN associés par d'autres fonctionnellement et QoS équivalent.

Étape 2. Le processus Resource Data Collection & Processing s'occupe des notifications collectées en permanence à partir de l'infrastructure de fourniture des services. Ce processus va envoyer une alarme de défaillance qui traduit la défaillance d'un élément ayant une incidence sur le service offert à un ou plusieurs clients vers le processus « Resource Trouble Management ».

Étape 3. La nous avons deux cas possibles, ou bien le processus Resource Trouble Management va résoudre l'incident par une restauration automatique, ou dans le cas où il ne peut pas le faire envoyer des rapports d'alarme au processus « Service Problem Management », en indiquant l'heure à laquelle l'interruption s'est produite et sa durée potentielle afin de permettre à ce processus de déterminer d'autres mesures susceptibles de minimiser l'incidence sur le service.

Étape 4. Le processus « Service Problem Management » envoie le rapport d'alarme au processus « Session Management » afin de prendre les mesures correctives qui s'imposent.

Étape 5. Le processus « Session Management » va envoyer une demande de Re-provisioning de l'élément de service dégradé vers le processus Service Composition Management.

Étape 6. Le processus Service Composition Management va réserver les ressources appropriées pour répondre à la demande (VPSN, VPCN et VPEN re-provisioning).

Étape 7. Dans le cas où il n'y a pas de ressources approvisionnées le processus Service Composition Management va envoyer la demande vers le processus « Service Configuration & Activation » pour réapprovisionner le service dégradé. Cette demande est envoyée au processus ressources provisioning.

Étape 8. Le processus ressources provisioning apporte les modifications requises relatives à la configuration des ressources afin de garantir que les ressources sont conformes aux indicateurs KQI pour le service.

Étape 17. Le processus Resource Provisioning envoie les mises à jour au processus Resource Inventory Management

Étape 18. Le processus Resource Provisioning notifie les résultats des modifications ainsi que le temps mis pour procéder aux modifications et tous les autres paramètres d'infrastructure et d'exploitation au processus « Service Configuration & Activation ».

Étape 19. Le processus « Service Configuration & Activation » envoie les mises à jour au processus Service Inventory Management.

Étape 20. Le processus « Service Configuration & Activation » notifie les mesures prises au processus Service Composition Management et au processus « Service Problem Management ».

Etape 22. Le processus « Service Problem Management » envoie des détails concernant les mesures correctives au processus « Service Quality Management » afin de les intégrer dans la surveillance et la gestion permanentes de la qualité de service.

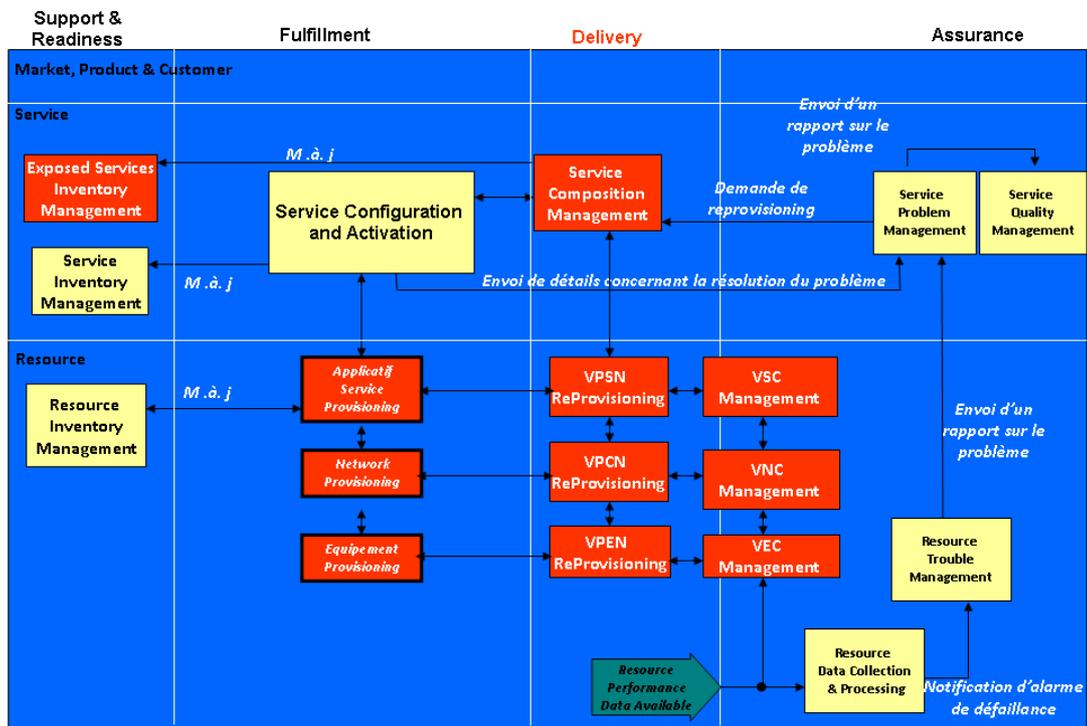


Figure V.2-8 : Interaction entre le processus d'assurance, de delivery et de provisioning.

V.2.6 Conclusion

Dans un environnement NGN/NGS, l'utilisateur est nomade et veut un maintien de sa QoS et une assurance du service durant ses déplacements et ses changements de préférences, nous avons vu que l'eTOM ne prend pas en charge cette mobilité dans ses processus de gestion du cycle de vie, ainsi, nous avons proposé d'intégrer la gestion de la QoS durant les différentes phases du cycle de vie et d'avoir ainsi un cycle de vie cohérent et une gestion dynamique de la session mobile pendant le provisioning, le delivery et l'assurance des services.

En utilisant nos modèles de service et de QoS, nous avons commencé par un déploiement intelligent basé sur la QoS (profil d'usage, modèle de QoS), dans lequel nous déployons le service dans la plate-forme qui va lui permettre d'assurer son contrat, une fois le service déployé, sa communauté est créée (VSC Creation). Cette dernière permettra une gestion des services ubiquitaires selon leur QoS et leur fonctionnalité durant le processus d'assurance. Durant le processus de provisioning, nous avons une sélection des ressources selon leurs QoS statistiques (VPSN Creation), puis durant le processus de delivery proposé, nous avons une affectation dynamique du service selon sa QoS courante (VPSN Provisioning). La Figure V.2-9 résume nos propositions intégrées dans l'eTOM pour une gestion dynamique de la mobilité et des changements de préférences dans le contexte NGN/NGS.

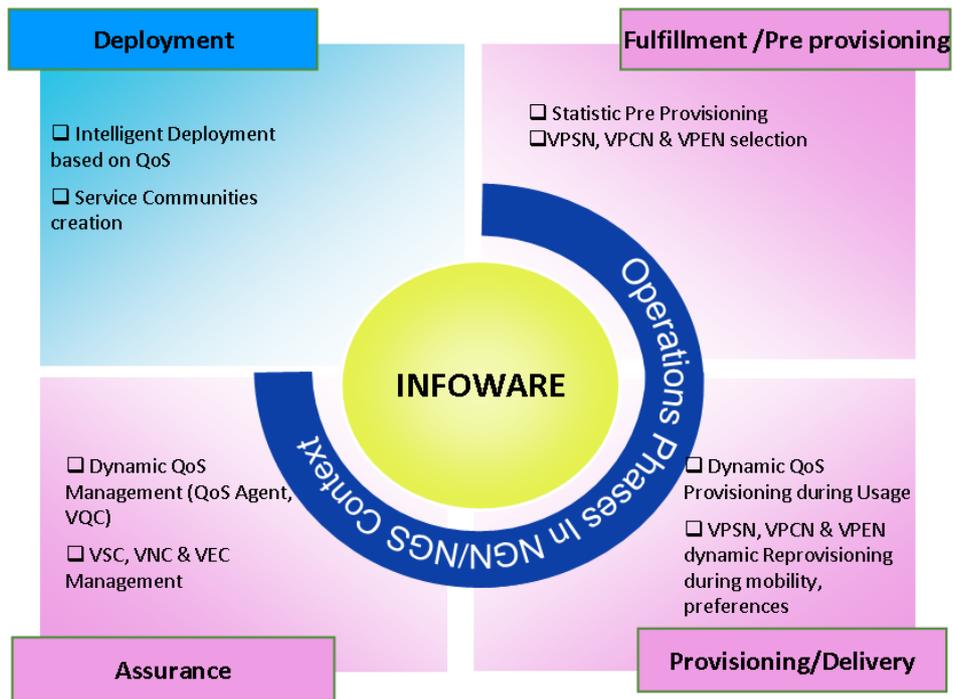


Figure V.2-9: gestion dynamique de la mobilité dans le nouveau contexte NGN/NGS.

Chapitre VI Implémentation

Dans cette partie, nous présentons d'abord un scénario qui montre toute la gestion de la session d'un utilisateur à partir de son abonnement chez l'opérateur jusqu'à son utilisation des services et la gestion de tout type de changement dans sa session (§VI.1). Puis dans (§VI.2) nous détaillons notre plate-forme UBIS de développement des services applicatifs ainsi que des services de gestion proposés pour gérer les différentes phases du cycle de vie (§VI.3). Enfin, nous montrons les testes effectués pour la gestion de la file d'attente attachée à l'élément de service durant l'usage et la prise en compte de la QoS.

VI.1 Scénario de gestion de la session d'un utilisateur : de l'abonnement jusqu'à l'usage

Nous présentons dans cette section un cas d'usage (Figure VI.1-1) de la gestion d'une session utilisateur dans le nouveau contexte NGN/NGS mobile et ubiquitaire, en partant de l'abonnement de l'utilisateur aux services jusqu'à leurs usage et leurs gestion.

SERVICEWARE : représente la couche service qui contient des éléments de service ubiquitaires (déployés dans différentes plates-formes, différentes zones géographiques) gérés par différents opérateurs. Nous avons des éléments de service de type Telco, des éléments de service de type web ainsi que tout les services de gestion définit dans nos travaux qui permettent de gérer les services applicatifs.

INFOWARE : représente la base informationnelle contenant tout les profils et les tables définit dans nos travaux (Profil d'usage pour le déploiement, Profil de ressource, Profil real time, Profil utilisateur, etc.).

USERWARE : représente la plate-forme de service définit sur le terminal de l'utilisateur, avec des services applicatifs (web et telco) et des services de gestion.

INFOSPHERE : représente la base informationnelle attachée au USERWARE, contient les informations nécessaires à la gestion dans le USERWARE. Cette base informationnelle peut se synchroniser avec l'INFOWARE.

VI.1.1 Abonnement : Création d'un « profil utilisateur » contenant tous les services abonnés par le client.

Alice est une cliente qui veut s'abonner à des services UBIS, l'abonnement s'effectue par la consultation du catalogue des services exposables. L'abonnement peut aussi s'effectuer sur des services exposables découverts par son terminal. Ainsi Alice consulte le catalogue pour voir les offres sur les services qu'elle veut à savoir : *un service d'échange de maison pour les vacances (House Swap), un service VoD et un service appel- visio.*

Service Exposable	Offre	Tarif	Qualité	Opérateur
VoD1_SFR	1 mois	49€	High Quality	SFR
VoD2_SFR	3 mois	29€/mois	High Quality	SFR
VoD3_ORANGE	6mois	15€/mois	High Quality	Orange
FaceTime_SFR	12mois	30€/mois		SFR
Swap to go	3mois	19,90€		SFR
House Swap	1mois	9€		

Sur le catalogue Alice trouve différentes offres concernant les services qu’elle veut qui sont chez différents opérateurs, avec différentes qualités, différentes offres et différents tarifs. Alice va choisir ses services exposables selon ses préférences.

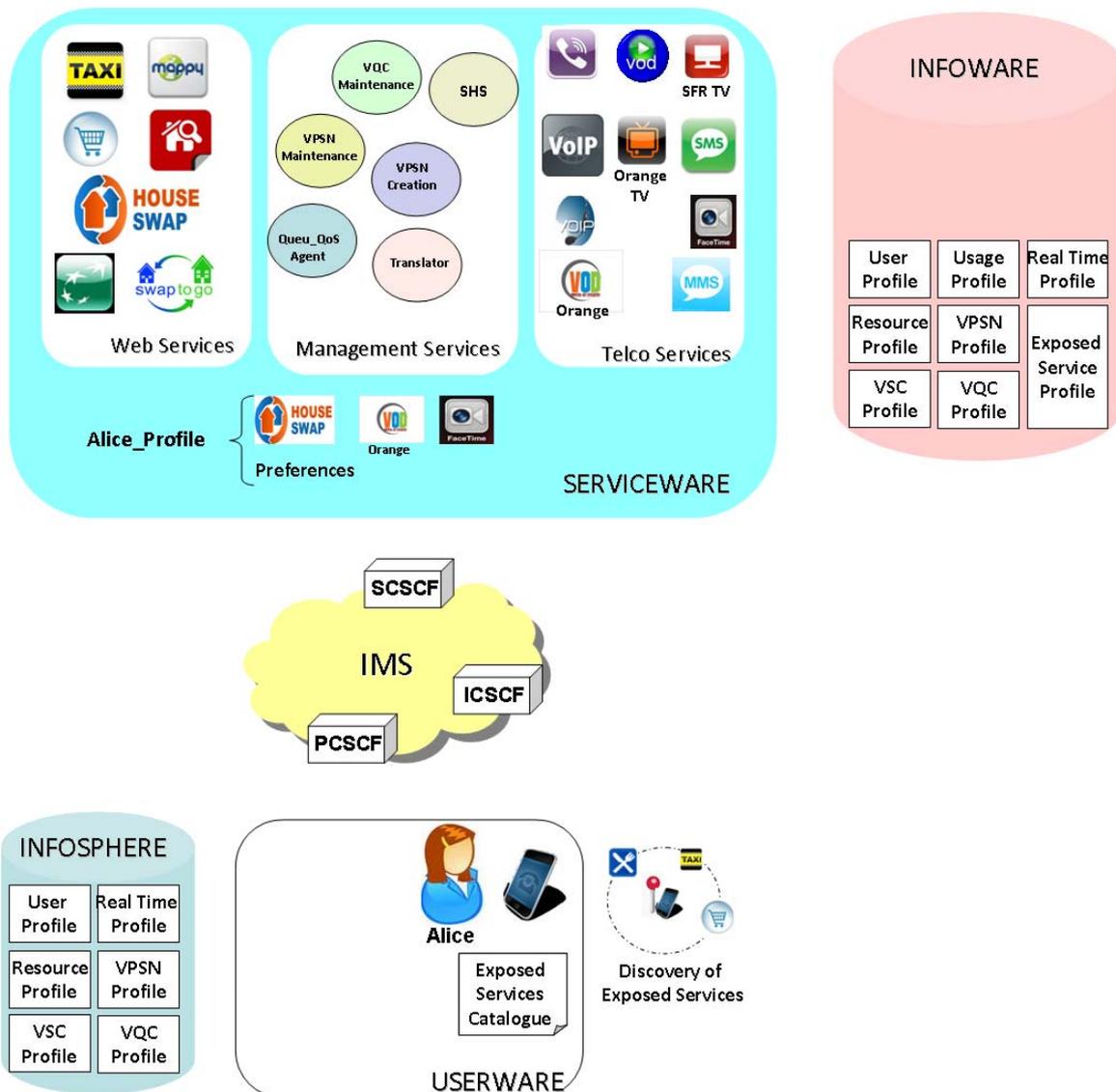


Figure VI.1-1: Cas d’usage

Elle choisit un service web d’échange de maison : « **Swap to go** » qui permet de consulter des annonces selon différents critères (localisation, période, style de maison, etc.), d’échanger les photos et/ou vidéo en live de la maison, de contact par SMS ou Mail. Alice choisit aussi

un service « **VoD3_Orange** », car le prix, la qualité et l'offre lui convient. Elle choisit le service d'appel visio « **FaceTime_SFR** » chez SFR. Ainsi, un Profil Utilisateur est créé pour Alice dans la base informationnelle INFOWARE contenant les services exposables auquel elle a souscrit. Profil_Alice : « **SwapToGo** » + « **VoD3_Orange** » + « **FaceTime_SFR** ». Son profil utilisateur va contenir aussi toutes ses préférences.

VI.1.2 Ouverture de session et Pré-provisioning : Création du VPSN

Alice veut échanger sa maison avec une autre personne habitant à Nice pour ses vacances, étant abonnée à un service web « House Swap », elle va l'utiliser pour sélectionner quelques annonces et contacts de gens qui échangent leur maison pour les vacances à Nice, puis pour visualiser les photos, échanger les photos de sa maison avec son contact ou visualiser par vidéo la maison.

Pour utiliser ses services, Alice initialise sa session en choisissant les services qu'elle veut utiliser dans cette session : House Swap + FaceTime_SFR.

Une requête SIP INVITE contenant la liste des services exposables que veut utiliser Alice durant sa session ainsi que leur $QoS_{SE_{Exp0}}$ demandée est envoyée via le cœur IMS vers la plateforme de service. « House Swap » et « FaceTime_SFR » sont des services exposables décomposables en éléments de service.

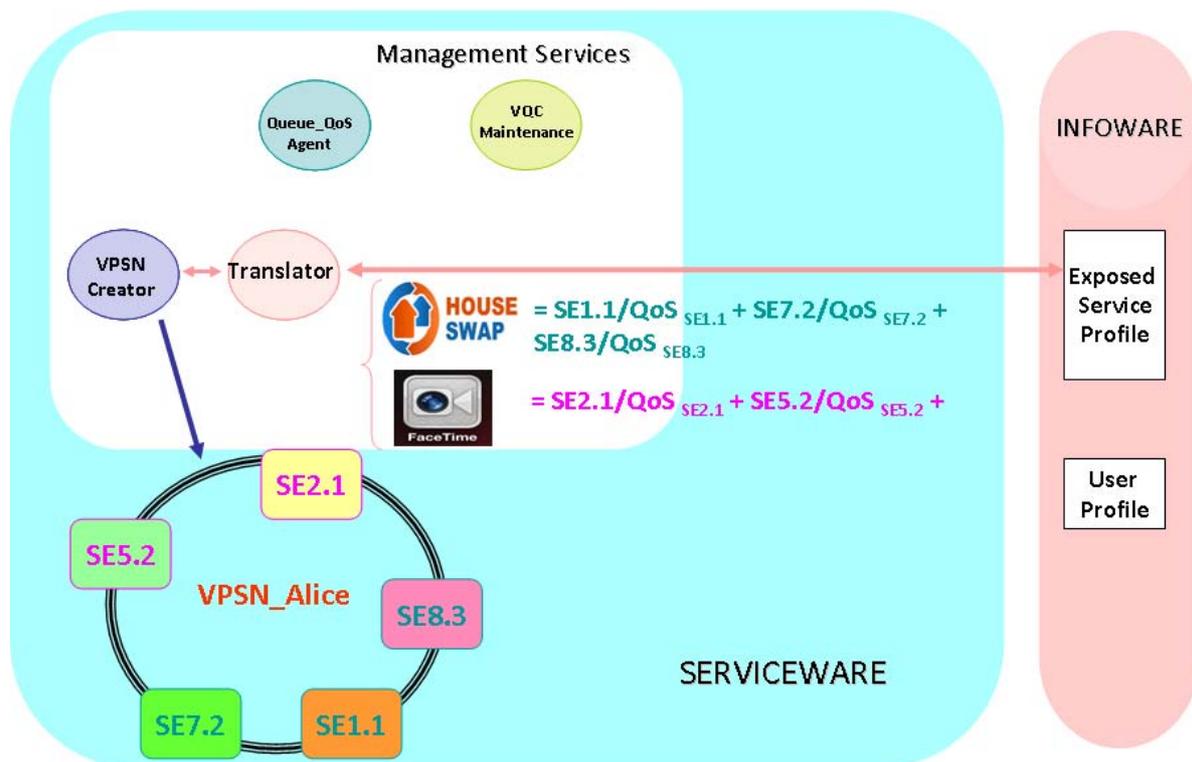


Figure VI.1-2: Translation des services exposables et création du VPSN.

La requête est traitée par le service de gestion « Translator » qui traduit les services exposables reçus en éléments de service de base. Ce service va récupérer du « Profil des services exposables » dans l'INFOWARE la composition de « House Swap » et de « Facetime_SFR » ainsi que la QoS de chaque ES de la composition, House Swap se décompose en : Annonce + Gallery + Contact. FaceTime_SFR se décompose en : Voix + Vidéo.

Le service de gestion « Translator » va solliciter le service de gestion « VPSN Creation » qui va créer le VPSN de Alice en effectuant tous les contrôles de sécurité comme par exemple l'autorisation et récupération de l'ID du VPSN à partir du service de gestion de la sécurité. Les cinq SE seront ainsi Pré-provisionnés (sélectionnés) pour être ajoutés dans le VPSN de Alice. Ayant un agent QoS dans chaque élément de service, l'ajout d'un élément de service dans le VPSN se fait par le « Attach_VPSN » de l'Agent QoS qui va attacher l'ES au VPSN en mettant l'ID de ce VPSN dans le ES et en changeant son statut à « Activé » dans le cas où il ne l'est pas déjà. Le choix d'un élément de service ubiquitaire peut se faire selon la localisation d'Alice, pour choisir un élément de service dans son environnement ambiant.

VPSN_Alice : Annonce (SE8.3) + Gallery (SE7.2) + Contact (SE1.1) + Voix (SE2.1) + Vidéo (SE5.2) (Figure VI.1-2).

VI.1.3 Provisioning de QoS de l'élément de service durant l'usage

Durant l'usage, Alice envoie une requête pour utiliser son service SE1.1. Alice sera provisionnée par l'élément de service SE1.1 qui a été sélectionné dans son VPSN.

La sélection de SE1.1 à été faite selon sa QoS statistique. La requête d'Alice est envoyée vers la file d'attente de SE1.1, puis dans le cas où cet élément de service ne peut satisfaire la requête (selon sa QoS courante), l'Agent QoS de l'élément de service va notifier le VQC et elle sera routée vers une autre file d'attente ubiquitaire par ce service de gestion des files d'attentes ubiquitaires « VQC Maintenance ».

Ainsi, la requête « Req3 » est envoyée vers la plate-forme de service, en trouvant dans l'annuaire JNDI le nom de la file d'attente associée à SE1.1, elle sera mise dans SE1.1_Queue (Figure VI.1-3).

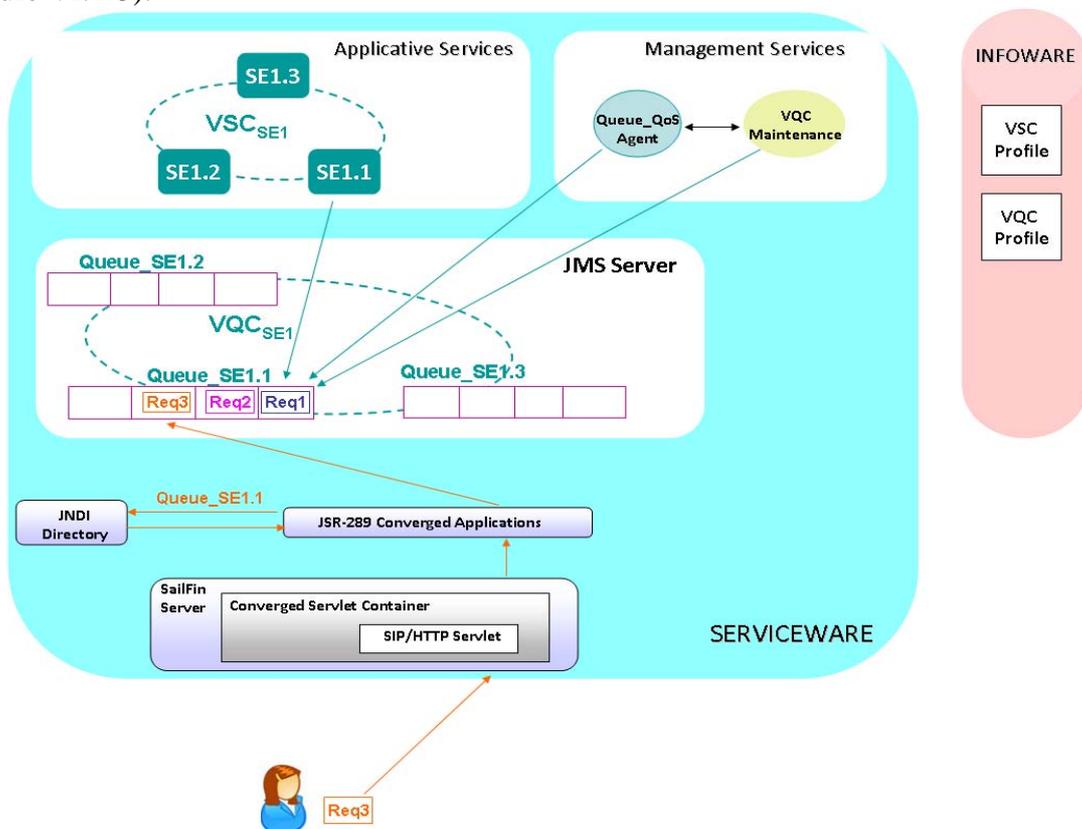


Figure VI.1-3: Requête d'Alice envoyée vers la plate-forme de services.

VI.1.4 Provisioning de QoS de la file d'attente durant l'usage

La file d'attente de SE1.1, dans laquelle se trouve « Req3 » d'Alice contient déjà trois requêtes selon sa QoS (Capacité, Délai, Fiabilité et Disponibilité),
 Le service de gestion « Queue_QoS Agent » contrôle si la QoS courante de la file d'attente ne dépasse pas la QoS seuil, s'il y'a un dépassement (par exemple dans notre cas délai de traitement plus long au niveau de l'élément de service qui signifie un délai d'attente plus long pour les requêtes dans la file), il va notifier le VQC Maintenance.
 Chaque requête dans la file ne doit pas dépasser son temps d'attente moyen (son SLA). Le « VQC Maintenance » en comparant le délai courant de la requête « Req1 » par rapport au délai moyen, trouve qu'il est dépassé, ainsi il va déclencher son algorithme qui calcul pour chaque requêtes après « Req1 » si SE1.1 peut le traiter dans le délai demandé. Le VQC trouve que la requête « Req3 » va dépasser son délai d'attente moyen dans la file et donc va envoyer la requête « Req3 » vers une file d'attente ubiquitaire, dans notre exemple vers Queue_SE1.3 (Figure VI.1-4).

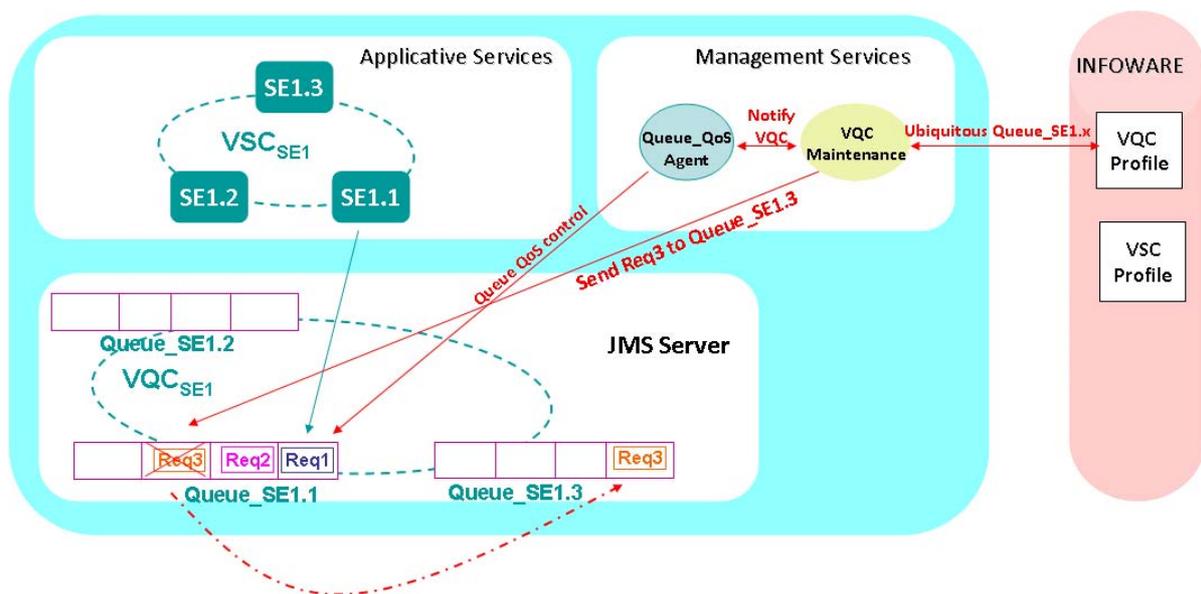


Figure VI.1-4 : Requête d'Alice envoyée vers une file d'attente ubiquitaire.

VI.2 Plate-forme de développement et de test

L'ensemble de nos propositions rentre dans le cadre du projet UBIS. Dans ce projet nous avons une plate-forme complète regroupant les différentes couches de visibilité : utilisateur, terminal, réseau d'accès, réseau cœur et service (Figure VI.2-1).

Pour l'utilisateur : selon l'approche user-centric que nous adoptons, nous avons un profil utilisateur qui sera stocké dans la base de connaissances INFOWARE et INFOSPHERE qui contient toutes les exigences et les préférences de l'utilisateur ainsi que ses abonnements.

Pour le terminal : nous avons une architecture USERWARE pour l'intégration entre l'usage et la personnalisation de services. Ce USERWARE fournit les fonctionnalités et les capacités nécessaires pour permettre une personnalisation, et un accès orienté usage et transparent de bout-en-bout aux services voulus.

Pour le réseau de transport : nous utilisons la solution VIRTUOR qui commercialise une machine physique acceptant de nombreuses entités virtuelles de différents types : routeurs IPv4 et IPv6, point d'accès virtuel, serveur SIP, etc. Cette solution permet la mise en place de plusieurs réseaux virtuels, spécifiques aux applications supportées, qui sont vus de la part des utilisateurs comme des réseaux physiques distincts.

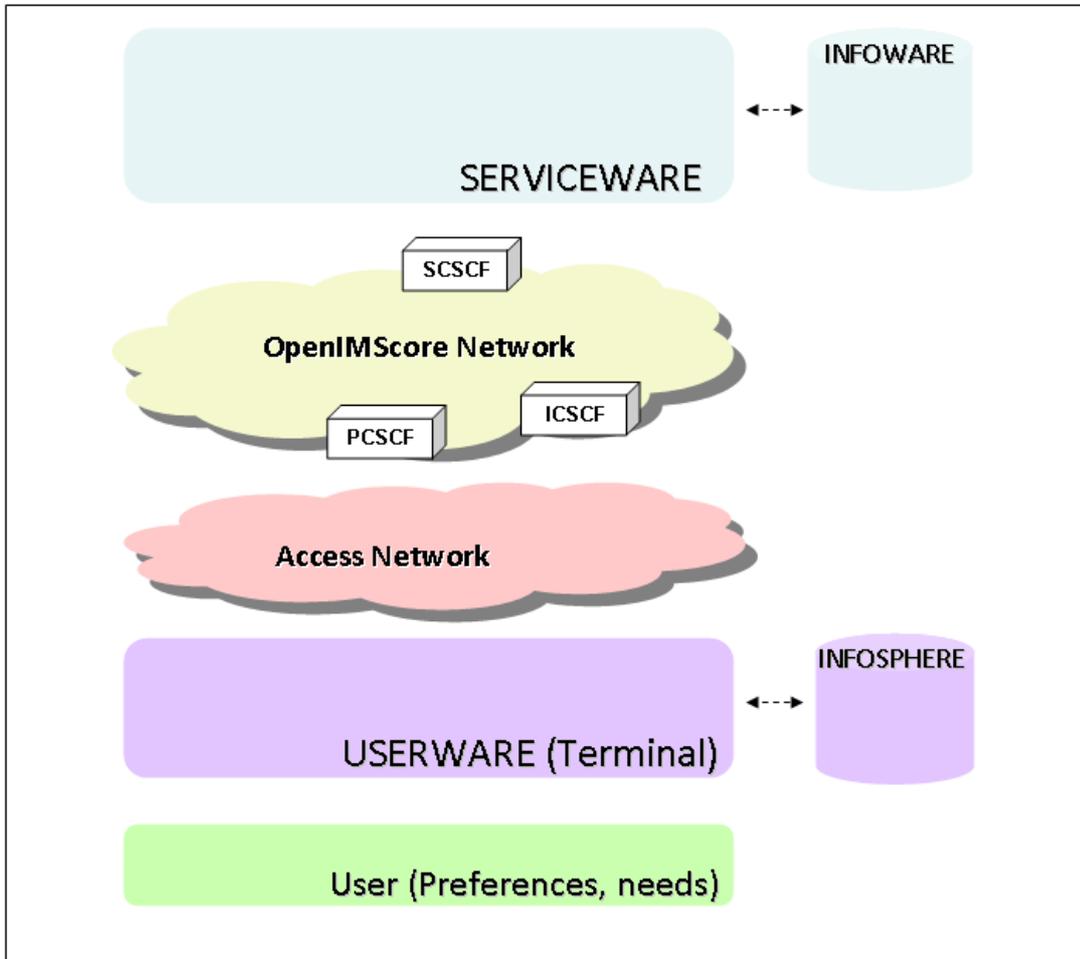


Figure VI.2-1 : Différentes couches de visibilité.

Pour le réseau cœur : nous utilisons le OpenIMS de FOKUS [OpenIMScore] comme plan de contrôle.

Pour les informations : nous avons une base de connaissance INFOWARE du côté fournisseur et INFOSPHERE du côté utilisateur/terminal.

Pour les services : nous avons une architecture de service, appelée SERVICEWARE, qui est mise en œuvre par notre groupe de travail afin de tenir compte des défis au niveau NGN (mobilité, hétérogénéité et aspect user-centric) et NGS (Composition de service, hétérogénéité des services et ubiquité).

Dans le SERVICEWARE nous avons développé des services applicatifs (web et telco) et des services de gestion (QoS Agent, VQC Maintenance, VPSN Creation, VSC Maintenance, Translator, Compositor, etc.).

Du point de vue implémentation, nous avons GlassFish, un serveur d'application Java. GlassFish V3 [Glassfish] est une plate-forme open source standard, facilement utilisée pour construire et déployer des services de nouvelle génération (NGS) basés sur SOA. A ce serveur d'application s'ajoute Sailfin comme serveur pour la signalisation. Au niveau signalisation, une version SIP améliorée est proposée par notre groupe de travail (SIP+).

Pour valider nos propositions, chaque service de gestion et service applicatif est développé comme un EJB [RPS06] indépendant. Les EJBs permettent la création de différents composants faiblement couplés et autonomes. Les EJBs sont déployés sur un serveur d'application Glassfish. Glassfish est certifié JEE6 et en conséquence supporte différentes APIs comme JMS, JNDI et JDBC.

Les requêtes envoyées par l'utilisateur sont des requêtes SIP pour la signalisation et le provisioning et des requêtes HTTP pour l'usage. Pour l'interface entre SIP/HTTP et nos éléments de service EJB, nous avons le JSR Converged Applications.

Le Converged Applications est un ensemble de *SipServlet* et *HttpServlet* géré par le même conteneur permettant l'interfaçage entre le monde SIP /HTTP et les composants (En EJB ou Web Services).

Pour la gestion des noms des files d'attente attachée à chaque ES, nous utilisons le JNDI Directory.

Le JNDI fournit les fonctionnalités de nommage et d'annuaire aux applications écrites en Java. Il permet d'associer des noms à des objets et de retrouver ces objets grâce à leurs noms.

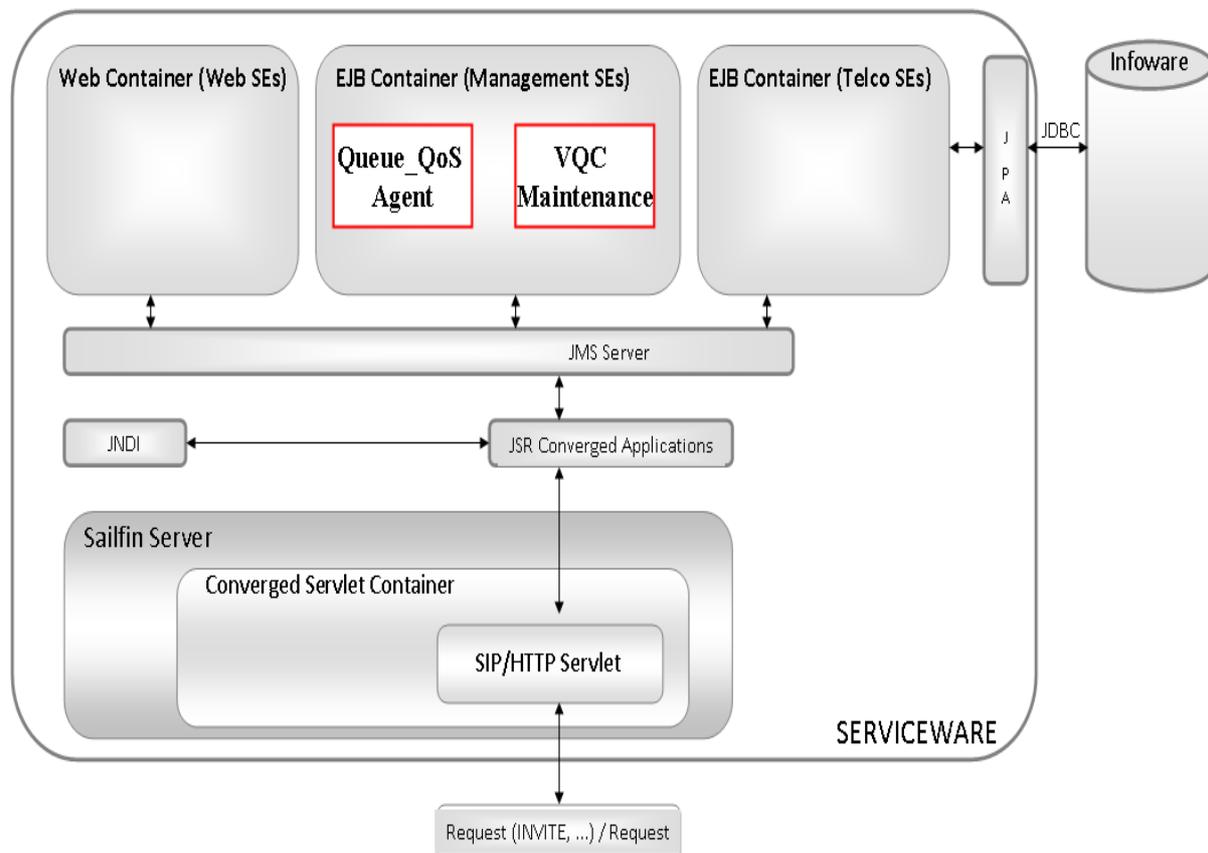


Figure VI.2-2 : L'architecture du SERVICEWARE.

Figure VI.2-2, montre notre plate-forme de service, dans laquelle nous avons un Web Container pour les services web, un EJB Container pour les services telco et un EJB Container pour les services de gestion.

Pour se connecter à la base informationnelle, nous utilisons la technologie JPA (Java Persistence API) qui fait la correspondance entre l'objet (EJB) et le modèle relationnel. Pour cela l'Entity Manager de l'EJB utilise les informations de configuration contenues dans un fichier appelé unité de persistance qui contient une référence sur le connecteur JDBC à utiliser en tant que source de données.

La Figure VI.2-3, représente l'architecture d'un EJB, nous utilisons des EJB Stateless (sans état). Le MDB est à l'écoute de la file d'attente et réagit à l'arrivée d'une requête. Il extrait cette dernière de la file d'attente, en récupère le contenu puis exécute un traitement.

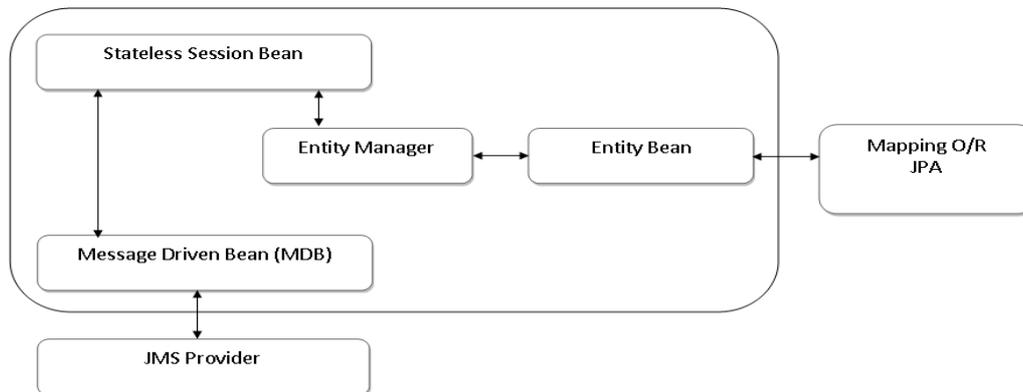


Figure VI.2-3: Architecture d'un EJB (ES).

Pour la gestion de la file d'attente associée à un élément de service (EJB), nous avons développés deux éléments de service de gestion « Queue_QoS Agent » et « VQC Maintenance ». Nous présentons le service de gestion « Queue_QoS Agent ».

VI.2.1 Élément de service de gestion « Queue_QoS Agent »

Les éléments de service spécifiés et développés par notre groupe de travail pour prendre en compte les nouveaux besoins NGN/NGS ont des caractéristiques d'autonomie, d'ubiquité, d'autogestion, de stateless, d'interopérabilité, de mutualisation, etc. Pour l'autogestion, nous avons un Agent QoS dans chaque élément de service qui permet de contrôler si cet élément de service est conforme à son contrat ou pas (envoi IN/OUT Contrat). Pour cet élément de service (Figure VI.2-4) :

Dans le plan de Signalisation : reçoit :

un **VPSN Attach**, à partir du VPSN Creation :

Incrémente la variable Attach à +1, pour avoir une trace du nombre de VPSN auquel il est attaché.

Vérifie la QoS statistique pour décider s'il peut accepter la demande et répond en sortie par OK/NOK.

Dans le cas d'un NOK, mettre la variable Attach à -1, il n'est pas attaché à ce VPSN.

Dans le cas d'un OK, inscrire ce VPSN dans la table du SE et changer l'état du SE à l'état Activable dans le cas où il faisait partie d'aucun VPSN (si la variable Attach est égale à 1), dans le cas où Attach est différente de 1, ignorer (ne pas changer l'état).

Une « Transaction_j » :

Incrémente la variable j à $+1$, pour avoir une trace du nombre de transactions auquel il participe.

Vérifie la QoS dynamique pour voir s'il peut accepter la transaction dans la file d'attente et répond en sortie par OK/NOK.

Dans le cas d'un NOK, diminuer la variable j à -1

Dans le cas d'un OK, provisionner la requête dans la file d'attente de l'élément de service et changer l'état du SE à l'état Activé dans le cas où il n'existe aucune requête dans la file auparavant (le cas où $j=1$).

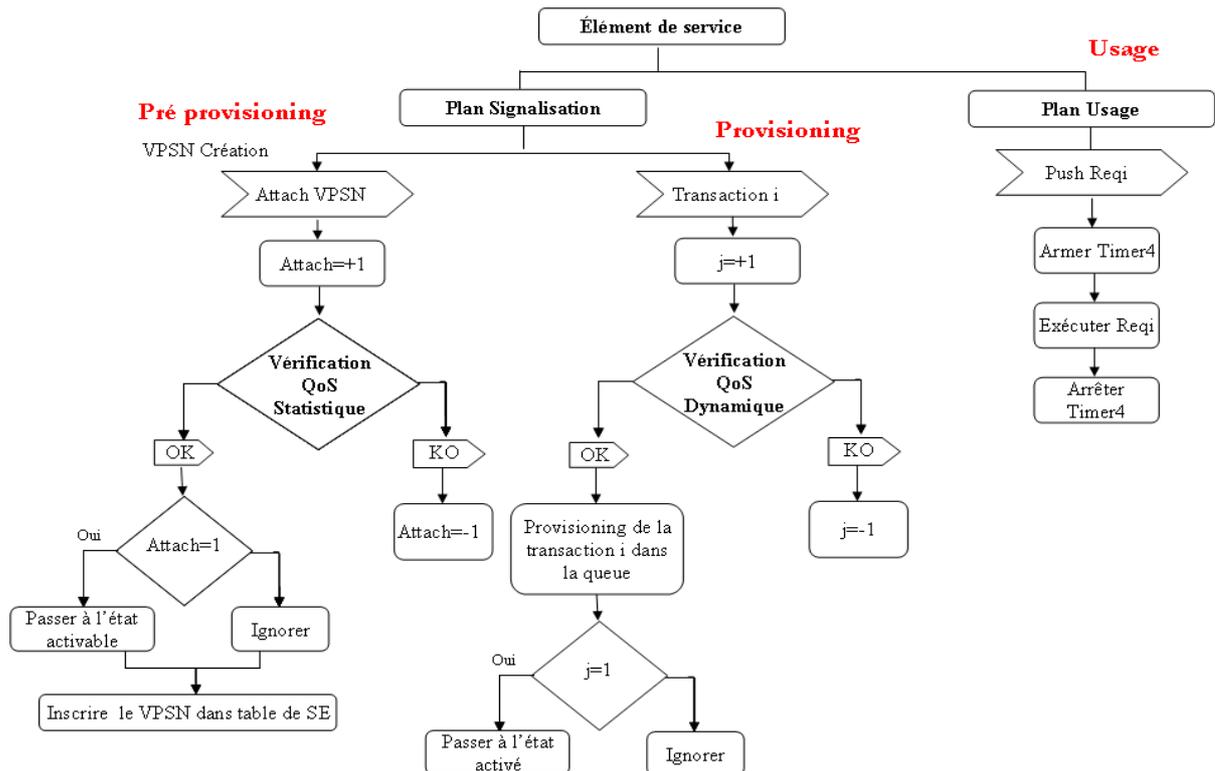


Figure VI.2-4: Automate d'un élément de service (Plan usage et Signalisation).

Dans le plan d'Usage :

Récupère la requête de la file d'attente

Arme un Timer4 du temps d'exécution moyen d'une requête selon la QoS du SE

Exécute la requête

A la fin d'exécution (exécution se termine bien), arrêter le Timer4

Dans le plan de Gestion :

Dans le cas où le Timer armé (Timer4) à expiré, par exemple la requête à prit plus de temps de traitement, il va notifier le gestionnaire VQC pour qu'il envoie la requête vers une file d'attente ubiquitaire.

L'Agent QoS (de la file d'attente ou de l'élément de service) compare la valeur courante de QoS à la valeur seuil :

À la fin d'une période de temps (Fin Timer5) compare les valeurs courantes de QoS aux valeurs seuils,

Si les valeurs courantes sont inférieure aux valeurs seuils envoyer un IN Contrat, puis armer Timer5 (pour re comparer les valeurs à la fin du Timer5)

Si les valeurs courantes sont supérieures aux valeurs seuils (au moins deux valeurs: par exemple le Délai +Capacité) envoyer un Out Contrat, armer un premier Timer2, puis un deuxième Timer3 plus grand que le premier, puis envoyer notification au VSC pour demander un changement de communauté. Une fois le traitement effectué Arrêter le Timer2, puis armer le Timer5 afin d'attendre la comparaison des valeurs de QoS.

Pour la réception, si le SE reçoit un Out Contract, il met un FlagOut à 1 (pour spécifier qu'il a reçu un Out Contrat) et envoi une notification au VSC afin d'exclure le SE dégradé de la communauté.

Si le SE reçoit un IN Contrat, il va arrêter le Timer1 qui à été armé en attendant de recevoir un IN Contrat, puis il va armer le Timer1 pour attendre le IN Contrat suivant.

Si le Timer1 expire (Timer1 out), ça veut dire que le SE n'a pas reçu de IN Contrat, il va d'abord vérifier le FlagOut s'il est à 1 cela veut dire que le SE à reçu un Out Contrat, sinon (le SE n'a reçu ni IN ni Out Contrat) donc notifier le VSC afin de gérer ce SE (communication défectueuse).

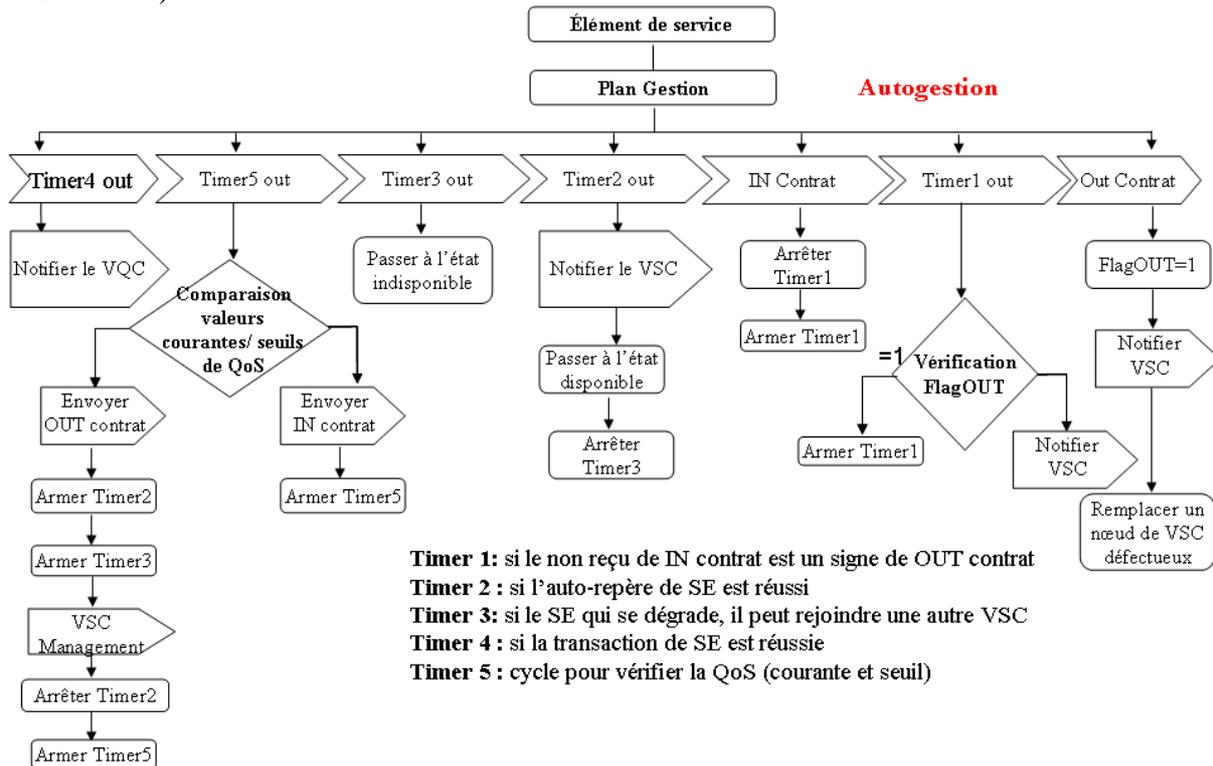


Figure VI.2-5 Automate d'un élément de service (Plan de gestion).

Notre travail se situe dans le plan signalisation pour le provisioning des transactions selon la QoS dynamique de l'ES et de la file d'attente et dans le plan de gestion, pour le traitement des outs contrats de la file d'attente par la notification du VQC.

VI.2.2 Élément de service de gestion « VQC Maintenance »

Lorsque le « VQC Maintenance » reçoit un « Out contrat » de la part du Queue_QoS Agent », Agent QoS de la file d'attente, il exécute son algorithme afin de décider des requêtes qui risquent d'expirer dans la file (requêtes qui ne vont pas respecter leurs SLA) et les envois vers une autre file d'attente ubiquitaire.

VI.3 Test de faisabilité sur la file d'attente avec OpenMQ

Dans notre plate-forme de développement, chaque élément de service a une file d'attente attachée qui reçoit les requêtes acceptées par lui selon sa QoS courante. Les files d'attentes sont sur un serveur JMS [JMS]. Ce dernier, offre une API Java permettant aux applications de créer, envoyer et lire des requêtes à travers d'une file d'attente gérée. Dans l'architecture JMS, nous avons des entités qui jouent le rôle de Producteur, de Fournisseur et de Consommateur.

Le Producteur est un client qui envoie des requêtes.

Le Fournisseur est un système qui prend en charge la gestion des files d'attente.

Le Consommateur est un élément de service qui reçoit et consomme les requêtes.

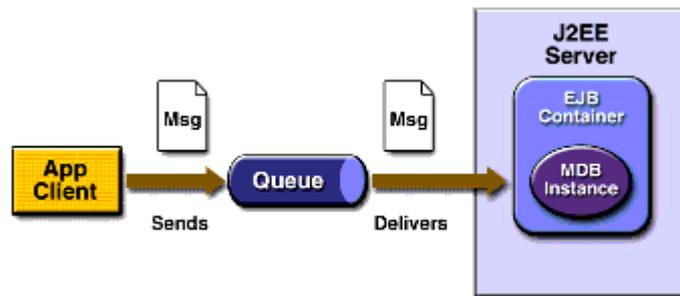


Figure VI.3-1: Architecture JMS.

Nous utilisons Open Message Queue (OpenMQ) [OpenMQ] comme notre système de gestion des files d'attentes qui fournit une implémentation JMS complète. Un OpenMQ est lancé avec chaque élément de service au même moment qu'est lancé le SE.

Chaque élément de service est implémenté comme un EJB. Dans chaque EJB, nous avons une file attachée et un Message Driven Bean (MDB) qui écoute sur les requêtes rentrant dans sa file d'attente attachée. Puis consomme les requêtes de cette file d'attente.

L'acceptation des requêtes dans la file d'attente se fait selon les quatre paramètres de QoS courante de l'ES. Par exemple si le ES est disponible et peut consommer quatre requêtes par une unité de temps. La cinquième requête qui arrive va être routé vers un ES ubiquitaire.

Nous avons un Queue Agent qui traite les « outs contrats » de chaque file d'attente et notifie un service de gestion « VQC Maintenance » pour contrôler le temps d'attente de chaque requête dans la file.

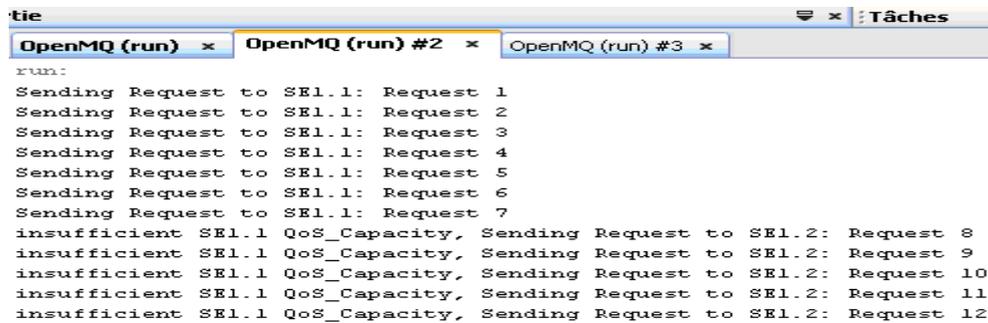
Chaque ES fait partie d'une VSC groupée par la même fonctionnalité et QoS, nous avons un service de gestion « VSC Maintenance » qui gère les ESs ubiquitaires.

Chaque file d'attente fait partie d'une VQC, nous avons un service de gestion « VQC Maintenance » qui gère les files d'attentes ubiquitaires.

VI.3.1 Scenario 1 : Provisioning du Service Par la QoS courante

Quand une requête arrive, le service de provisioning dans le ES va voir si sa QoS courante peut accepter la requête dans la file. Par exemple, si SE1.1 peut traiter 8 requêtes en moyenne

par une unité de temps selon sa QoS_Capacity, dans un délai QoS_Delay et qu'il existe déjà huit requêtes dans la file pour ce délai, la requête qui arrive sera transférée vers un autre ES ubiquitaire dans notre exemple vers SE1.2. Si la requête peut être acceptée un message JMS sera créé selon le contenu de la requête. La figure montre le résultat d'un exemple d'acceptation des requêtes selon la capacité de SE1.1 dans OpenMQ. Dans le cas d'une capacité insuffisante, les requêtes sont envoyées vers SE1.2.



```

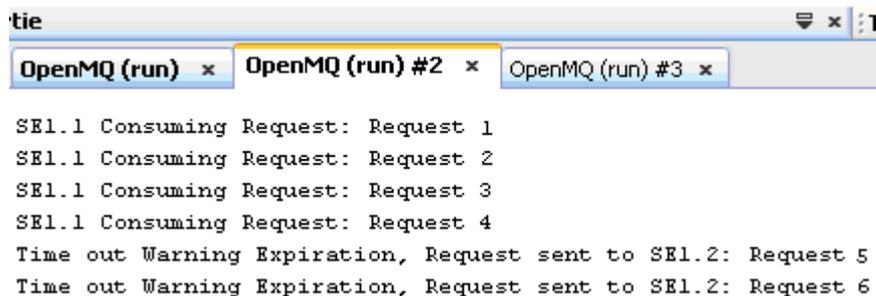
tie
OpenMQ (run) x OpenMQ (run) #2 x OpenMQ (run) #3 x
run:
Sending Request to SE1.1: Request 1
Sending Request to SE1.1: Request 2
Sending Request to SE1.1: Request 3
Sending Request to SE1.1: Request 4
Sending Request to SE1.1: Request 5
Sending Request to SE1.1: Request 6
Sending Request to SE1.1: Request 7
insufficient SE1.1 QoS_Capacity, Sending Request to SE1.2: Request 8
insufficient SE1.1 QoS_Capacity, Sending Request to SE1.2: Request 9
insufficient SE1.1 QoS_Capacity, Sending Request to SE1.2: Request 10
insufficient SE1.1 QoS_Capacity, Sending Request to SE1.2: Request 11
insufficient SE1.1 QoS_Capacity, Sending Request to SE1.2: Request 12

```

Figure VI.3-2 : Provisioning de service selon QoS_Capacity courante.

VI.3.2 Scenario 2: Expiration du “Time out warning” des requêtes

Le service de gestion VQC Maintenance permet d'anticiper les expirations d'attente d'une requête dans la file d'attente, par exemple dans le cas où le SE a pris plus de temps pour traiter la requête (Request 4), la requête qui vient derrière sera dans les temps, tandis que les requêtes 5 et 6 risquent de dépasser le délai d'attente moyen, ainsi notre VQC qui contrôle la file va décider d'envoyer ces deux requêtes vers une autre file d'attente ubiquitaire de SE1.2.



```

tie
OpenMQ (run) x OpenMQ (run) #2 x OpenMQ (run) #3 x
SE1.1 Consuming Request: Request 1
SE1.1 Consuming Request: Request 2
SE1.1 Consuming Request: Request 3
SE1.1 Consuming Request: Request 4
Time out Warning Expiration, Request sent to SE1.2: Request 5
Time out Warning Expiration, Request sent to SE1.2: Request 6

```

Figure VI.3-3 : Expiration du time out, requête envoyé vers un SE ubiquitaire.

Le VQC Maintenance contrôle le temps moyen d'attente de la première requête dans la file (AverageTime) et le compare au temps d'attente réel (System.currentTimeMillis - JMSTimeStamp). Dans le cas où ce temps dépasse le temps d'attente réel, il exécute son algorithme qui sélectionne les requêtes qui risquent d'expirer.

Chapitre VII Conclusion Générale

Cette thèse avait pour objectif d'apporter une contribution dans la gestion du cycle de vie des services dans un contexte NGN/NGS mobile et ubiquitaire en proposant des concepts et des processus qui permettent d'avoir un service delivery qui satisfait l'utilisateur et l'opérateur. Dans cette conclusion, nous allons faire la synthèse de nos contributions et valorisations (§VII.1) et formuler nos perspectives (§VII.2).

VII.1 Contributions

Notre objectif était d'assurer une continuité de service pendant l'usage tout en maintenant une E2E QoS du côté utilisateur, et une interaction dynamique entre les processus de gestion des services pour une E2E QoS du côté opérateur dans le nouveau contexte NGN/NGS. Pour cela, nous avons commencé par délimiter ce nouveau contexte en nous focalisant sur la mobilité, l'ubiquité et l'environnement ambiant et analyser leurs impacts sur les processus de gestion du cycle de vie des services pour un "delivery de service" dans ce nouveau contexte. Notre analyse nous a permis d'identifier les verrous suivants :

1) *Offrir les mêmes services ubiquitaires dans différentes zones ambiantes.*

Ce point nous a conduit à repenser les services pour avoir des services avec les caractéristiques d'autonomie, d'interopérabilité, d'autogestion, de stateless, de mutualisation, et d'autonomie.

2) *Déployer dans différentes zones ambiantes des services ubiquitaires et redéployer dynamiquement pour faire face à la mobilité et à la QoS.*

Ce point nous a conduit à proposer un déploiement intelligent des éléments de services ubiquitaires basé sur un même modèle de service et de QoS. Le déploiement proposé prend en compte le contrat de QoS pour l'installation de l'élément de service sur la plateforme appropriée par l'utilisation d'un profil de déploiement (profil d'usage). Le redéploiement ainsi sera très rapide et flexible. Ainsi, à partir de cette phase de déploiement, nous répondons au SLA de l'utilisateur, en configurant le contrat que doit respecter chaque service. Parallèlement, nous avons créé des VSC pour tracer et gérer les communautés de services ubiquitaires.

3) *Provisionner les services en suivant les utilisateurs mobiles.*

Ce point nous a conduit à proposer de provisionner la ressource « service applicatif » selon sa QoS afin d'optimiser son utilisation et sa disponibilité pour les utilisateurs mobiles. Nous avons proposé un pré provisioning du service applicatif selon sa QoS statistique, puis durant l'usage nous tenons compte de sa QoS courante pour provisionner les requêtes. Les éléments de service étant mutualisables, nous avons proposé d'intégrer un agent de QoS à la file d'attente de chaque élément de service afin de gérer les requêtes acceptées par l'ES. Pour gérer les files d'attentes des éléments de service ubiquitaires nous avons proposé des

communautés de files d'attentes (VQC) qui permettent en plus de la QoS du nœud (ES) de maintenir la QoS du lien entre les éléments de services.

4) Tenir compte de la mobilité de l'utilisateur, de ces changements de préférences dans son environnement ambiant pour le delivery du service à l'utilisateur.

Dans la délivrance du service à l'utilisateur, nous avons le Media Delivery dont le rôle est de trouver le chemin optimal qui maintient la QoS pour transporter le flux entre une source et une destination. Mais cette destination et cette source risque de changer durant l'usage suite aux mobilités. C'est ce point qui nous a conduit à proposer un Service Delivery au dessus du Media Delivery qui permet de gérer la mobilité et les changements de préférences, ainsi que la gestion du changement de la source ou de la destination entre l'utilisateur et sa composition de service. Le Service Delivery proposé est dirigé par un modèle nous permettant d'avoir un réseau de services (VPSN) qui s'autogère indépendamment des infrastructures de transport afin d'assurer la continuité de service tout en maintenant la QoS demandée.

5) Gérer dynamiquement la QoS pour que la solution permette la continuité du service.

L'ensemble de nos contributions nous a conduit à proposer une gestion dynamique de la QoS de bout en bout. Cette gestion est basée sur :

- a) un même modèle de service avec des éléments de service ubiquitaires déployés dans différentes zones ambiantes et gérés par des VSC selon leur QoS et leur fonctionnalité.
- b) un même modèle de QoS basé sur les quatre critères de QoS (Disponibilité, Fiabilité, Capacité et Délai) et des valeurs mesurables.
- c) un même modèle informationnel (profil de ressource, profil d'usage, profil de VPSN, profil de VSC, profil de VQC).
- d) une autogestion dans chaque élément de service réalisé par un Agent QoS qui control et l'élément de service et sa file d'attente.

Comme valorisation à nos contributions, nous avons proposé :

D'étendre la gestion de la mobilité par la QoS (VSC) à une gestion par zones ambiantes (SHS) afin de profiter des services ambiants à l'utilisateur et optimiser ainsi l'accès aux services (Latency). Pour cela, nous avons proposé un Handover au niveau service par analogie aux handover au niveau du réseau d'accès et du réseau cœur, ce handover permet de changer d'élément de service ubiquitaires par zones ambiantes.

D'intégrer nos concepts, nos modèles de QoS, de Service et informationnel dans les standards de gestion du cycle de vie des services dans les télécom pour répondre au nouveau contexte NGN/NGS.

VII.2 Perspectives

Notre travail de thèse se fait en parallèles avec d'autres projets (INFOWARE pour la base informationnelle, USERWARE pour le terminal, Sécurité) et Intégration, il rentre dans le cadre du projet UBIS (User-centric uBiquité et Intégration des Services), dans lequel nous avons développé des éléments de services selon notre modèle de service et de QoS, ainsi il faudrait maintenant tester la faisabilité et les performances de nos propositions par leur intégration dans le projet.

Comme perspectives à nos travaux, nous proposons aussi d'intégrer des éléments de service de tarification et de facturation dans notre session VPSN et gérer dynamiquement cette tarification intra et inter fournisseurs.

En plus de la virtualisation proposée au niveau de la couche service, nous proposons aussi une virtualisation de bout en bout et ainsi une évolution vers le cloud computing.

Chapitre VIII Références Bibliographiques

- [3GPP10] 3rd Generation Partnership Project (3GPP). 3gpp ts 36.300 version 9.4.0, release 9: Evolved universal terrestrial radio access (e-utra) and evolved universal terrestrial radio access network (e-utran). Jun. 2010.
- [AA07] A. Arjona, H. Verkasalo, “Unlicensed mobile access (UMA) handover and packet data performance analysis”, In *Proc. of ICDT*, pages 9–9, Barcelona, Spain, Jul. 2007.
- [BU08] B. Urgaonkar, P. Shenoy, A. Chandra, P. Goyal and T. Wood, “Agile Dynamic Provisioning Of Multi- Tier Internet Applications”, *ACM Transactions on Autonomous and Adaptive Systems TAAS 2008*, Volume 3 Issue 1, March 2008, doi: 10.1145/1342171.1342172
- [CO03] C. Oliveira Appendix i of deliverable 4.2 of the agile project: GSM system -handover. Technical report, Agile Alliance, 2003.
- [DA08] Deployment of Component-Based Applications on Top of a Context-Aware Middleware
- [DH08] Didier Hoareau and Yves Mahéo, Middleware support for the deployment of ubiquitous software components,
- [EDGE] 3rd Generation Partnership Project, “Technical Specification Group GSM/EDGE Radio Access Network”, 3GPP TS 44.060 V9.2.0.
- [EKK09] E. Kyu Kim, Y. Kim and I. Chong, “Architectural model and service scenario of dynamic service overlay network (DSON)”, *First International Conference on Ubiquitous and Future Networks, ICUFN 2009*, pp. x – xiii, doi: 10.1109/ICUFN.2009.5174273
- [ETSI07] European Telecommunications Standards Institute (ETSI). 3GPP TR 25.922 version
- [ETSI09a] ETSI DTR/USER-00029-1 (TR 102 805-1), “User Group; End-to-end QoS management at the Network Interfaces; Part 1: User's E2E QoS - Analysis of the NGN interfaces (user case), Control plan solution: QoS signalling”, 2009.
- [ETSI09b] ETSI DTR/USER-00029-2 (TR 102 805-2), “User Group; End-to-end QoS management at the Network Interfaces; Part 2: Control and management planes solution - QoS continuity, Management plan solution: QoS Inter-

- working”, 2009.
- [ETSI09c] ETSI DTR/USER-00029-3 (TR 102 805-3), “User Group, End-to-end QoS management at the Network Interfaces; Part 3: QoS informational structure”, 2010.
- [ETSI11] 3rd Generation Partnership Project (3GPP). 3gpp ts 25.832 version 4.0.0, release 4 : Manifestations of handover and srns relocation. Mar. 2011.
- [FB06] F. Bennani, Z. Benhamed Daho, N.Simoni, Chunyang.Yin: “An Informational Framework for Autonomic Networking”, GRES 2006
- [GAN] European Telecommunications Standards Institute (ETSI). 3GPP TS 43.318 version 10.1.0, release 10: Generic access network (GAN) stage 2. Apr. 2011.
- [GB921a] TMF-GB921, “Business Process Framework (eTOM): Concepts and principles, R8.0 Release.
- [GB921b] TMF-GB921, “Business Process Framework (eTOM): Business process framework decompositions and descriptions”, release 8.0.
- [GB921c] TMF-GB921, “Business Process Framework (eTOM): Representative process flows”, release 8.0.
- [Glassfish] SUN Microsystems, <https://glassfish.dev.java.net>
- [GPRS] European Telecommunications Standards Institute (ETSI). 3gpp ts 23.060, version 10.3.0, release 10: General packet radio service (gprs). Mar. 2011.
- [GSM] European Telecommunications Standards Institute (ETSI). GSM TS 03.09, version 5.1.0: Handover procedures. Aug. 1997.
- [IEEE09] IEEE Computer Society, IEEE Microwave Theory, and Techniques Society. IEEE standard for local and metropolitan area networks part 16: Air interface for broadband wireless access systems. In *IEEE Std 802.16-2009 (Revision of IEEE Std 802.16-2004)*. IEEE, May 2009.
- [IPTV] ETSI TS 182 027, V2.0.0, “Telecommunications and internet converged services and protocols for advanced networking (TISPAN), IPTV Architecture”, Technical specification, 2008.
- [ITIL] Information Technology Infrastructure Library, <http://www.itilfrance.com/>
- [JCY07] User-Driven Service Lifecycle Management – Adopting Internet Paradigms in Telecom Services
- [JF06] J Fan, M Ammar, “Dynamic Topology Configuration in Service Overlay Networks: A Study of Reconfiguration Policies”, - Proc. IEEE INFOCOM, 2006

- [JWK10] J.W. Kim, S.W. Han, D.H. Yi, N. Kim, and C.C.J. Kuo, “Media-Oriented Service Composition with Service Overlay Networks: Challenges, Approaches and Future Trends”, *Journal of Communications*, Vol. 5, No 5. (2010), pp. 374-389, May 2010. doi:10.4304/jcm.5.5.374-389.
- [JX07] J. Xie, I. Howitt, I. Shibeika IEEE 802.11-based mobile IP fast handoff latency analysis. In *Proc. of ICC*, pages 6055–6060, Glasgow, Scotland, Jun. 2007.
- [MB07] M. Boniface, S. Phillips, A. Sanchez-Macian and M. Surridge, “Dynamic Service Provisioning Using GRIA SLAs”, *Service-Oriented Computing, NFPSLA-SOC'07*, Sept 2007, Vienna
- [MIH] IEEE Computer Society. Ieee standard for local and metropolitan area networks part 21: Media independent handover services. In *IEEE Std 802.21-2008*. IEEE, Jan. 2009.
- [MS] M. Stiermerling et al., "System Design of SATO & ASI", IST Project Ambient Networks, D12, http://www.ambientnetworks.org/Files/deliverables/D12-F.1_PU.pdf
- [MS07] M. Song, and B.Mathieu, “QSON: QoS-aware Service Overlay Network”, *CHINACOM 2007*, pp. 739 – 746, doi: 10.1109/CHINACOM.2007.4469494
- [MW96] M Wang, B Li, Z Li, “sFlow: Towards Resource-Efficient and Agile Service Federation in Service Overlay Networks”, *Quality-of-Service Routing for Supporting Multimedia Applications. IEEE Journal of Selected Areas in Communications*, 14(7):1228–1234, 1996
- [NGNM] ITU-T M.3060/Y.2401, “Principles for the Management of Next Generation Networks”, Mars 2006.
- [NGOSS] TMF-GB930, “The NGOSS Approach to Business Solutions”, Release 1.0
- [NS07] N. Simoni, B. Mathieu, C.Yin, M. Song, « Autogestion de service par la QoS dans un Réseau Overlay », *GRES'07*, Hammamet Tunisia, Nov. 2007
- [NS08] N.Simoni, C.Yin, G. Du Chene, « An intelligent user centric middleware for NGN: Infosphere and AmbientGrid”, In *Proc. of COMSWARE*, pages 599–606, Bangalore, India, Jan. 2008.
- [NS09] Noémie SIMONI, Xiaofei XIONG, Chunyang YIN Virtual Community for the Dynamic Management of NGN Mobility ICAS'09, Avril, 09, Valencia, Spain
- [NS97] N.Simoni, S.Znaty, “*Gestion de réseau et de service – similitude des concepts, spécificité des solutions*”, InterEditions, 1997.
- [OpenIMSCore] Open IMS Core, FOKUS, <http://www.openimscore.org/>

- [OpenMQ] Open MQ, <https://mq.dev.java.net/>
- [RB02] R Braynard, D Kostic, A Rodriguez, J Chase, A Vahdat, “Opus: an Overlay Peer Utility Service”, Proceedings of the 5th International Conference on Open Architectures and Network Programming (OPENARCH), 2002
- [RPS06] R.P. Sriganesh, G. Brose and M. Silverman: Mastering Enterprise JavaBeans 3.0. Wiley Publishing Inc. 2006
- [SC08] S. Chen, J. Lukkien, R. Verhoeven, P. Vullers and G. Petrovic, “Context-Aware Resource Management for End-to-End QoS Provision in Service Oriented Applications”, IEEE GLOBECOM Workshop 2008, pp. 1-6, doi: 10.1109/GLOCOMW.2008.ECP.50
- [SR07] S.Rostambeik, N.Simoni et A.Boutignon, “Userware: A framework for next generation personalized services”, *Computer Communications*, 30(3): 619–629, Feb.2007.
- [TMF] Tele Management Forum, <http://www.tmforum.org/>
- [TMN] ITU-T M.3010, « Principles for a telecommunications management network », 2005.
- [TP06] T. Pollet, G. Maas, J. Marien and A. Wambecq, “Telecom Services Delivery in a SOA”, 20th IEEE/ International Conference on Advanced Information Networking and Applications 2006, doi: 10.1109/AINA.2006.322.
- [TR139] Service Delivery Framework Overview, V1.0, Nov 2007.
- [XG03] X Gu, K Nahrstedt, RN Chang, C Ward, “QoS-Assured Service Composition in Managed Service Overlay Networks”, 2003
- [XL08] X. Li, H. M. Chan, T. Hung, K. H. Tong and S. J. Turner, “Design of an SLA-driven QoS Management Platform for Provisioning Multimedia Personalized Services”, AINAW '08 Proceedings of the 22nd International Conference on Advanced Information Networking and Applications, doi: 10.1109/WAINA.2008.256
- [ZBD04] Z.Benhamed 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.
- [ZBD06] Z. Benahmed Daho, « Vers une Nouvelle Génération de Service pour une Autogestion de bout en bout », 2006.

Liste des publications

Soumia KESSAL, Noémie SIMONI, “Service Provisioning oriented QoS”, CNSM 2011, Octobre 2011, Paris.

Rachad NASSAR, Soumia KESSAL, Noémie SIMONI, “Semantic Handover for Seamless Service Continuity within User Ambient Context”, NGMAST 2011, Septembre 2011, Cardiff.

Soumia KESSAL, Noémie SIMONI, “A deployment of service elements based on QoS”, SERVICES 2011, Juillet 2011, Washington.

Soumia KESSAL, Noémie SIMONI, Xiaofei XIONG, Chunyang YIN, “Model-Driven Dynamic Service Delivery in Mobility and Ambient Environment”, ICIW'11, Mars 2011, St. Maarten.

Soumia KESSAL, Noémie SIMONI, Philippe COUDE, « Déploiement intelligent des éléments de service basé sur la QoS », 1^{er} Délivrable DME2S pour partenaire SFR, 2009.

Soumia KESSAL, Noémie SIMONI, Philippe COUDE, « Provisioning dynamique de service dans un contexte de mobilité et d'ubiquité », 2^{ème} Délivrable DME2S pour partenaire SFR, 2009.

Chapitre IX Annexe

Dans cette section nous présentons le code source de l'agent QoS.

```

public class QosControlClass {

    private static String response = null;

    public static String QosControl(List CurrentQosList, List MaxThrsldQosList, List
MinThrsldQosList) {

        boolean a = false, b = false, c = false, d = false;
        for (int j = 0; j < CurrentQosList.size(); j++) {

            if (Float.valueOf(CurrentQosList.get(j + 0).toString()) <
Float.valueOf(MaxThrsldQosList.get(j + 0).toString())
                && Float.valueOf(CurrentQosList.get(j + 0).toString()) >
Float.valueOf(MinThrsldQosList.get(j + 0).toString())) {
                System.out.println("---" + Float.valueOf(MinThrsldQosList.get(j + 0).toString())
                    + "----" + Float.valueOf(CurrentQosList.get(j + 0).toString())
                    + "----" + Float.valueOf(MaxThrsldQosList.get(j + 0).toString()));
                a = true;
            }
            if (Float.valueOf(CurrentQosList.get(j + 1).toString()) <
Float.valueOf(MaxThrsldQosList.get(j + 1).toString())
                && Float.valueOf(CurrentQosList.get(j + 1).toString()) >
Float.valueOf(MinThrsldQosList.get(j + 1).toString())) {

                System.out.println("---" + Float.valueOf(MinThrsldQosList.get(j + 1).toString()) +
"----" + Float.valueOf(CurrentQosList.get(j + 1).toString())
                    + "----" + Float.valueOf(MaxThrsldQosList.get(j + 1).toString()));

                b = true;
            }
            if (Double.parseDouble(CurrentQosList.get(j + 2).toString()) -
Double.parseDouble(MaxThrsldQosList.get(j + 2).toString()) < 500
                && Double.parseDouble(CurrentQosList.get(j + 2).toString()) -
Double.parseDouble(MinThrsldQosList.get(j + 2).toString()) > 500) {
                System.out.println("---" + Float.valueOf(MinThrsldQosList.get(j + 2).toString())
                    + "----" + Float.valueOf(CurrentQosList.get(j + 2).toString())
                    + "----" + Float.valueOf(MaxThrsldQosList.get(j + 2).toString()));

                c = true;
            }
        }
    }
}

```

```

        if (Double.valueOf(CurrentQosList.get(j + 3).toString()) <
Double.valueOf(MaxThrsldQosList.get(j + 3).toString())
        && Double.valueOf(CurrentQosList.get(j + 3).toString()) >
Double.valueOf(MinThrsldQosList.get(j + 3).toString())) {
            System.out.println("---" + MinThrsldQosList.get(j + 3).toString()
                + "----" + CurrentQosList.get(j + 3).toString()
                + "----" + MaxThrsldQosList.get(j + 3).toString());
            d = true;
        }
        System.out.println("boolean response " + a + ":" + b + ":" + c + ":" + d);
        j = j + 3;
        if (a == true && b == true && c == true && d == true) {
            response = "In_Contrat";
        }
        if (a == false || b == false || c == false || d == false) {
            response = "Out_Contrat";
        }
    }
    return response;
}

```

public class AgentTimer2Class {

```

    public static void timer2(String idService,String response, long delay, long period) {
        final String Tresponse = response;
        final String TidServ = idService;
        final long Tdelay = delay;
        final long Tperiod = period;
        Timer timer = new Timer();
        timer.schedule(new TimerTask() {

            public void run() {
                System.out.println("Timer 3\n");
                AgentTimer3Class.timer3(TidServ,Tresponse, Tdelay, Tperiod);
                cancel();
            }
        }, delay, period);
    }
}

```

public class AgentTimer5Class {

```

    public static void timer5(String idService,String stringContrat,long delay,long period){
        final String[] tab = null;
        final String string= stringContrat;
        final String idServ = idService;
        final long Tdelay = delay;

```



```
        ServInterfaceMgtRemote service =
(ServInterfaceMgtRemote)ctx.lookup(ServInterfaceMgtRemote.class.getName());
        service.updateServiceStatus("serv_interface_mgt", TidServ, "Unavailable");
        cancel();
    } catch (NamingException ex) {
        Logger.getLogger(AgentTimer3Class.class.getName()).log(Level.SEVERE, null,
ex);
    }
    System.out.println("Fin Timer3");
}
}, delay, period);
}
}
```